Technische Universität München
TUM School of Engineering and Design

Dynamic Robot Walking on Unknown Terrain:

Stabilization and Multi-Contact Control of Biped Robots in Uncertain Environments

Felix Simon Sygulla

Vollständiger Abdruck der von der TUM School of Engineering and Design der Technischen Universität München zur Erlangung eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz: Prof. Dr.-Ing. Boris Lohmann

Prüfer*innen der Dissertation:

1. Prof. dr.ir. Daniel J. Rixen
2. Prof. Abderrahmane Kheddar

Die Dissertation wurde am 24. Juni 2021 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 29. November 2021 angenommen.

Abstract

The main advantage of anthropomorphic robots is their adaptability to environments built for humans. For walking robots, robustness to uneven terrain is essential to exploit these advantages. Although current approaches often use the full multi-body dynamics, the executed motions over uneven terrain are often only quasi-static, i.e., correspondingly slow. This thesis deals with reduced control approaches to answer which level of detail of the models is necessary for dynamic walking over uneven ground.

In this thesis, a tactile sensor prototype for measuring contact surfaces is proposed and experimentally evaluated. Also, this work proposes contributions for balance control strategies that consider the robot's structural eigenfrequencies and centroidal dynamics. Furthermore, force-control concepts for contacts with the environment are presented, which exhibit robust stability to uncertain mechanical contact properties. On the one hand, the approaches tolerate uncertainty in the timing, stiffness, and geometry of the contacts. On the other hand, they also adapt to unexpected parameters based on the measured contact surface and contact forces. Both hand and foot contacts are considered to support multi-contact locomotion on rugged terrain. Other parts of this thesis deal with software modules and architectures for biped robots and a hardware layer design. In addition, a machine-learning-based tool for intelligent robot system testing is presented.

All methods are evaluated on the bipedal robot LOLA both for normal walking and multi-contact locomotion experiments. The results show high robustness to undetected uneven terrain, unknown ground-height changes, and unexpectedly soft ground using the proposed control schemes with reduced dynamic models. The achieved walking speed on uneven and soft terrain is significantly higher than for state-of-the-art bipeds. Moreover, the results show that high-quality hardware and software/hardware integration in humanoid robotics are essential to achieve high overall system performance.

Acknowledgments

This thesis results from my research work at the Chair of Applied Mechanics, Technical University of Munich between 2015 and 2021. I want to express my gratitude to the many people who supported, inspired, and assisted me during the project.

First of all, I thank my supervisor Prof. dr. ir. Daniel Rixen for his trust and for giving me the great opportunity of working on Lola. I am grateful for the freedom he gave me in my research and his constant interest and support. The good working atmosphere at the Chair is mainly due to his uncomplicated and cooperative working style.

During my time at the Chair, I have worked with many highly talented and motivated individuals. The very collegial and friendly spirit was a true enrichment. I want to thank all colleagues for their support, the numerous professional and personal conversations, and the friendships that extended well beyond life at university.

I was lucky to work with and learn from three generations of Lola Ph.D. students during my time. I'll never forget the fruitful discussions, various hours in the lab, and the ups and downs of the project we have experienced together. It is virtually impossible to work alone on a machine as complex as Lola, which is why I am particularly grateful for the ever-reliable support of the Lola team members. My thanks go to Robert Wittmann, Arne-Christoph Hildebrandt, Daniel Wahrmann, Philipp Seiwald, Nora-Sophie Staufenberg and Moritz Sattler. Special thanks are devoted to Philipp Seiwald, with whom I spent most of the time maintaining and improving Lola. This thesis is unthinkable without his support and great commitment to the project. This is also the place to thank the German Research Foundation (DFG) for funding the project "Adaptive Walking through Multi-Contact Stabilization", grant #407378162; it also supported the research presented in this thesis.

I want to thank Christoph Schütz, Tobias Berninger, and Felix Ellensohn from the Chair's robotics research group for the fruitful discussions and joint research. Furthermore, I am grateful to my mentor Thomas Buschmann, who always had time for technical discussions on Lola and beyond.

The electrical and mechanical workshops at the Chair are especially valuable for the production and maintenance of robots. I want to thank Georg Mayr and Andreas Köstler for designing numerous electronic components, laying and repairing many cables, and always being ready to help with electrical problems of any kind. I further thank Georg König for building the great and challenging walls and terrain boards used in this thesis and Simon Gerer for manufacturing the required spare parts.

Numerous student projects supported the research of the Lola project. I want to thank all of my students for the commitment to the project and the fruitful scientific discourse. Finally, I would like to thank Philipp, Moritz, Nora-Sophie, and Kira for proofreading this thesis and giving valuable advice.

Contents

Glossary

Acronyms

SVM	Support Vector Machine
FT	Force/Torque
CoM	Center of Mass
IMU	Inertial Measurement Unit
CAN	Controller Area Network
ZMP	Zero Moment Point
FZMP	Fictious Zero Moment Point
FRI	Foot Rotation Indicator
DoF	Degree of Freedom
ADC	Analog-to-digital converter
TCP	Tool Center Point
ZRAM	Zero Rate of Angular Momentum
CMP	Centroidal Moment Pivot
LIPM	Linear Inverted Pendulum Model
CoP	Center of Pressure
DCM	Divergent Component of Motion
BoS	Base of Support
FoR	Frame of Reference
QP	Quadratic Program
DSCB	Distributed Sensor Control Board
WPG	Walking Pattern Generator
SIK	Stabilization and Inverse Kinematics
RF	Right Foot
LF	Left Foot
RH	Right Hand
LH	Left Hand
B	Base (Feet, Torso, Head)
FB	Floating Base
ASC	Automatic Supervisory Control
IK	Inverse Kinematics
DK	Direct Kinematics
J	Joints
DARPA	Defense Advanced Research Projects Agency
DRC	DARPA Robotics Challenge
MPC	Model Predictive Control
DDP	Differential Dynamic Programming
MIMO	Multiple-Input Multiple-Output
MRAC	Model-Reference Adaptive Control

SDO	Service Data Object
PDO	Process Data Object
API	Application Programming Interface
PSD	Power Spectral Density
PC	Passivity Controller
CAD	Computer Aided Design
VHIP	Variable-Height Inverted Pendulum
RMS	Root Mean Square
HWL	Hardware Layer
CI	Continuous Integration
PWM	Pulse-Width Modulation
FFT	Fast Fourier Transform
RBF	Radial Basis Function
DCT	Discrete Cosine Transform
C-SVC	C Support Vector Classification
EC	EtherCAT
DC	Distributed Clocks
ROS	Robot Operating System
YARP	Yet Another Robot Platform
CP	Capture Point
PCB	Printed Circuit Board

Symbols

${}_k\boldsymbol{r}_{i,j}$	Position vector from i to j written in k FoR
${}_k\boldsymbol{\omega}_{i,j}$	Angular velocity of j relative to i written in k FoR
${}_j\boldsymbol{\vartheta}_{j,i}$	Rotation vector describing the frame rotation from i to j
${}_j\boldsymbol{A}_i$	Frame-transforming rotation matrix from FoR i to FoR j
${}_j\boldsymbol{s}_i$	Frame-transforming quaternion from FoR i to FoR j in vectorial form
$\boldsymbol{q}$	Configuration-space vector
$\boldsymbol{J}$	Jacobian matrix
$\boldsymbol{e}_x$	3D unit vector along the x-axis
$\boldsymbol{e}_y$	3D unit vector along the y-axis
$\boldsymbol{e}_z$	3D unit vector along the z-axis
g	Gravity constant
n	Current time step: $t = n\,\Delta t_{\mathrm{cont}}$
$\boldsymbol{\Lambda}^P$	Total wrench on the robot at point P
$\boldsymbol{\lambda}^P$	A wrench consisting of forces and torques, e.g. $\boldsymbol{\lambda}^P = \begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{T}^P \end{bmatrix}$ at point P
$\boldsymbol{F}$	Force vector
$\boldsymbol{T}^P$	Torque vector at point P
ξ	Task-space selection factor $\xi \in [0,1]$; a 1 denotes the hand position is in task-space
$\boldsymbol{\xi}$	All task-space selection factors $[\xi_{\mathrm{RH}}, \xi_{\mathrm{LH}}]^T$
$\boldsymbol{x}$	Task-space vector; upper case indicates the complete vector, lower case a subvector
$\boldsymbol{v}$	Absolute task-space velocity; upper case indicates the complete vector, lower case a subvector
I'	FoR, which differs from I only in rotations around I_z. The I'_x axis always points in walking direction of the robot

I	Real world inertial frame of reference
W	Ideal world inertial FoR
U	Upper-body attached FoR
E	Generic end-effector FoR, see e
$\boldsymbol{C}(s)$	Laplace transformation of a linear controller
$\boldsymbol{P}(s)$	Laplace transformation of a linear plant
$\boldsymbol{U}$	Contact model matrix
$\boldsymbol{S}$	Binary selection vector or matrix
γ	A load factor $\gamma \in [0,1]$, $\gamma_{\mathrm{RF}} + \gamma_{\mathrm{LF}} = 1$
$\boldsymbol{\gamma}$	All load factors γ_e as vector
β	A control-mode blending factor $\beta \in [0,1]$; a value of 1 means force-controlled
$\boldsymbol{\beta}$	All control-mode blending factors β_e as vector
${}_{I'}\boldsymbol{\vartheta}_{\mathbf{IMU}}$	IMU-based rotation vector ${}_{I'}\boldsymbol{\vartheta}_{I',U}$ for frame rotation from I' to U, written in I'
${}_{U}\boldsymbol{\omega}_{\mathbf{IMU}}$	IMU-based angular velocity ${}_{U}\boldsymbol{\omega}_{I,U}$ for frame rotation from I to U, written in U
${}_{U}\boldsymbol{\vartheta}_{\mathbf{FB}}$	Rotation vector ${}_{U}\boldsymbol{\vartheta}_{\mathrm{id,m}}$ for floating-base frame rotation of measured U to ideal U, written in U
${}_{U}\boldsymbol{\omega}_{\mathbf{FB}}$	Angular velocity ${}_{U}\boldsymbol{\omega}_{I',W}$ for floating-base frame rotation from I' to W, written in U
ϵ	Early contact state $\epsilon \in \{0,1\}$
$\boldsymbol{\epsilon}$	All early contact states ϵ_e as vector
ι	Late contact state $\iota \in \{0,1\}$
$\boldsymbol{\iota}$	All late contact states ι_e as vector
κ	Actual contact factor $\kappa \in \{0,1\}$
$\boldsymbol{\kappa}$	All actual contact factors κ_e as vector
G	Feedback control gain
$\boldsymbol{\chi}$	Binary contact state data from discrete switches (one vector element per pad) or tactile sensor (binary matrix)
$\boldsymbol{D}$	Torque and friction limit matrix
$\boldsymbol{n}_e$	Contact surface normal for an end effector e in the end-effector FoR
$\boldsymbol{N}$	Contact surface normals for all end effectors $[\boldsymbol{n}_{\mathrm{RF}}, \boldsymbol{n}_{\mathrm{LF}}, \boldsymbol{n}_{\mathrm{RH}}, \boldsymbol{n}_{\mathrm{LH}}]$

Indices

$(\cdot)_{\mathbf{id}}$	An ideal (planned) quantity
$(\cdot)_{\mathbf{ad}}$	A quantity already modified by the trajectory adaptation module, see Figure 3.1
$(\cdot)_{\mathbf{bl}}$	A quantity already modified by the balance control module, see Figure 3.1
$(\cdot)_{\mathbf{d}}$	A desired quantity
$(\cdot)_e$	A quantity related to end effector $e = \{\mathrm{RF, LF, LH, RH}\}$
$(\cdot)_{\check{e}}$	A quantity related to the associated end effector of e. For the right hand, this is the left hand. For the left foot, this is the right foot, ...
$(\cdot)_f$	A quantity related to a foot $f = \{\mathrm{RF, LF}\}$
$(\cdot)_h$	A quantity related to a hand $h = \{\mathrm{RH, LH}\}$
$(\cdot)_{\mathbf{m}}$	A measured quantity
$(\cdot)_*$	An optimal quantity
$(\cdot)_{\mathbf{cont}}$	A quantity related to the main control cycle
$(\cdot)_{\mathbf{bus}}$	A quantity related to the bus update cycle
$(\cdot)_{\mathbf{ref}}$	A quantity describing a reference

Operators

$\tilde{(\cdot)}$	Tilde operator, see Equation (A.7)
wrap	Wraps the given rotation vector to the range $[-\pi, \pi)$, see Appendix A.2
$\odot$	Element-wise product (Hadamard / Schur product)
$\hat{(\cdot)}$	Element-wise inverse of a matrix or vector (Hadamard inverse)
$(\cdot)^{\#}$	The Moore-Penrose pseudoinverse of a matrix
ddt	Discrete-time derivative, see Appendix A.3
int	Discrete-time integrator, see Appendix A.3
lpf1	First-order low pass, see Appendix A.3
lpf2	Second-order low pass, see Appendix A.3
notch	Notch filter, see Appendix A.3
rotMat	Transforms quaternion to rotation matrix, see Equation (A.20)
$\otimes$	Hamiltonian product of two quaternions, see Equation (A.11)
quat	Converts a rotation vector to a corresponding quaternion, see Equation (A.23)
rotVec	Converts a quaternion to a corresponding rotation vector, see Equation (A.23)
diag	Diagonal matrix from vector
deltaX	Calculates the difference between two task-space vectors, see Appendix A.4
modify	End effector task-space modifier, see Appendix A.4
modifyAll	Apply modify to multiple end effectors in a chain, see Section 3.4.3
hybridCtl	Hybrid control scheme taking $(\beta, \boldsymbol{u}, \boldsymbol{x})$ as arguments, see Equation (A.40)
posCtl	Position control scheme taking $(\boldsymbol{u}, \boldsymbol{x})$ as arguments, see Equation (A.39)
intA	Integrator for vectors in moving FoR, see Equation (3.12)
$(\cdot)^*$	Quaternion conjugate
$\overset{\circ}{(\cdot)}$	Absolute derivative for quantities in a moving FoR, ${}_E\overset{\circ}{\boldsymbol{r}}_i = {}_E\dot{\boldsymbol{r}}_i + {}_E\boldsymbol{\omega}_i \times {}_E\boldsymbol{r}_i$

Chapter 1

Introduction

Since the industrial revolution, the importance of machines for human society has continuously increased. The first industrial machines drastically changed the economy and human society of that time, although these machines were relatively simple and well specialized on a particular problem or purpose. In the 1950s, the first industrial robots, i.e., programmable multi-purpose manipulators, were developed. Today, industrial robots automate the production of a large amount of high-tech goods. These robots are, when compared to humans, still very specialized in their specific tasks. Most notably, the robots, in most cases, are not mobile but are fixed in a very controlled environment and perform mostly repetitive tasks.

Mobility drastically increases robotic systems' complexity and leads to a new range of applications in complex non-industrial scenarios. Today, wheels are the preferred way of locomotion for robots in warehouses [170, 191], service robots [276], or cleaning robots at home. Still, wheeled robots may fail to overcome stairs, high obstacles, or uneven terrain — a restriction that does not hold for legged robots. Hybrids between legged and wheeled robots combine both worlds' strengths at the cost of higher system complexity [22, 148]. While the use of four-legged robots is — due to the higher stability — in most cases beneficial, narrow environments require biped robots with a smaller footprint. With their anthropomorphic design, biped robots are also ideal for the operation in environments built for humans. The initially high costs for humanoid robots favor applications that put high physical or psychological stress on humans. The list of applications is potentially huge, ranging from disaster recovery, search and rescue, inspection and monitoring of machinery in a dangerous environment to health- and elderly care. Moreover, the technology behind these robots can push the development of active prostheses or exoskeletons.

The unilateral contact between the robot's feet and the ground and the discrete steps used for continuous locomotion make bipedal walking challenging. The most significant problem is not the formulation of a comprehensive mathematical model for the involved physics. Instead, it is a problem of computational power to generate valid solutions in real-time and identify all the required parameters — some are hard to measure. One common approach is the use of a specific model in conjunction with a comprehensive optimization problem. While this strategy is elegant and generalizing from a mathematical perspective, its performance depends on the model's accuracy, and its optimization parameters are often sensitive to the targeted robotic hardware. A different approach — which is used throughout this thesis — is based on reduced models and simple control policies, which are easy to relate with physical effects and may be ported to different robots using only minimal information on the system characteristics. This approach further explains why a specific control policy has benefits in certain situations, which may show links to human walking strategies.

There has been considerable research on the dynamics and control of biped walking robots since the development of the first biped in 1973 [149]. Nevertheless, biped robots are still not competitive compared to wheeled robots, mostly because of their limited robustness on uneven or undetected terrain. Typically, the environment is perceived by cameras or similar sensors on the robot. Safe footholds are then computed based on a generated environment model [77, 93, 114, 146, 213, 317]. Disturbances and model inaccuracies are typically mitigated via feedback control loops. Still, uneven terrains are only traversed at low walking speeds [77, 146, 156, 195, 257, 277, 330] to keep the perception errors low and maintain stability. However, the machines must perform motions at a speed comparable to humans to be competitive — even in the presence of disturbances or unexpected uneven terrain.

This work attempts to identify control policies to improve the stabilization of biped robots on undetected, uneven terrain without using visual information on the terrain. It further tries to generate an understanding of why specific methods work and how they compare to different approaches. The thesis further deals with robot balancing in multi-contact scenarios, i.e., when the feet and arms are used for stabilization. In the following, an overview of the state of art is given, and the objectives and outline of this thesis are described.

1.1 State of the Art: An Overview

Due to the vast field of biped robot research, only works most relevant to this thesis's scope are summarized in the following. A comprehensive literature survey on work related to the proposed methods is part of the thesis's corresponding chapters. For a more in-depth review of vision-based planning and navigation for biped robots, refer to [111, 318].

Waseda University The first biped robot *WABOT-1* was developed at Waseda University in 1973 [149]. Since then, several impressive humanoid robots have been developed at the university's Humanoid Robotics Institute HRI. The latest biped *WABIAN-2R* [220] is able to walk over soft ground [104] at a walking speed of 0.22 m/s and realizes a quite human-like gait pattern with heel-strike and toe-off motions [221]. The robot is able to use slippery ground for quicker turning on the spot [102]. Recent research deals with developing human-inspired control of a running robot [224, 225].

Tokyo University - Jouhou System Kougaku Laboratory (JSK) At the JSK laboratory of the Tokyo University, several biped robots were developed, including the humanoids H6 and H7 [212, 218]. These robots reached a high level of autonomy and enabled early work on online 3D vision and motion planning for bipeds [133]. More recently, bipeds with water-cooled high-performance electrical drives were developed at JSK [131, 310]. These machines showed an impressive performance for step recovery in case of external disturbances although having position-controlled joints [311]. The team later founded the company *SCHAFT Inc.* to participate in the DARPA Robotics Challenge (DARPA) 2013 [58, 128] and won the first round of the "Trials" before being acquired by Google. The used robot was based on Urata and Nakanishi's work, who pronounced the importance of high-performance hardware for humanoid robots [128]. Recent work investigates the use of joint torque control without dedicated torque sensors [288] and robot control in multi-contact scenarios [116].

Humanoid Robotics Project / AIST The Humanoid Robotics Project (HRP) is a joint project of Japan's National Institute of Advanced Industrial Science and Technology (AIST), several universities, and Japanese industry partners. The general aim of the project is the development of domestic helper robots. Since 1998, several humanoid robots have been developed in the scope of the project [6, 134, 141–144] with the *HRP-5P* being the latest prototype [145]. At AIST, ground-breaking research on the stabilization of biped robots based on the Zero Moment Point (ZMP) was conducted [136, 137, 214]. Furthermore, adjustments to the ZMP trajectories [203, 216] combined with ground force control, and early-/late-contact strategies [201] enabled HRP robots to walk over uneven ground (±2 cm) at walking speeds up to 0.31 m/s. The used control schemes support partial contacts with the feet when the Base of Support (BoS) — often called support polygon due to the common approximation via polygons — is known beforehand [204]. Recent research with *HRP-5P* focuses on multi-contact planning of arm contacts [171].

Honda The Honda Motor Company is one of the pioneers in biped robot research. It initiated research on domestic robots in 1986 and developed several biped robots in total secrecy [117]. The first publicly released prototype *P2* already used ZMP and ground reaction force control and was able to walk on flat surfaces and stairs [115]. The gained experience with *P2* and the smaller *P3* resulted in the development of the biped *ASIMO* in 2002 [253]. *ASIMO* was continuously improved and is able to walk, hop, and run at a speed of 2.8 m/s [302]. A later developed experimental humanoid robot based on *ASIMO* is able to switch between different gait schemes dynamically in the presence of external disturbances [140]. Honda published few papers on *ASIMO*'s control algorithms [140, 300–303], which, however, do not fully disclose the secrets behind *ASIMO*'s hardware and software design. Nevertheless, Honda's approaches greatly shaped the design of humanoid robots worldwide.

Later work concentrated on the development of a legged robot for disaster response [332]. Recently, Honda published work on a mechanical leg design, which renders the mechanical properties of a biped close to those of a spring-loaded inverted pendulum, potentially simplifying control effort for biped walking [272].

KAIST / Seoul National University The Korea Institute of Science and Technology (KAIST) developed several humanoid robots from *KHR-1* over *KHR-3 (Hubo)* [231] to *Hubo-2* [107]. The robots are able to walk over uneven terrain (±4 mm) at speeds < 0.2 m/s [156] and run with a short flight phase [48]. For walking and running, the BoS of the feet overlap in forward direction, which allows to use a static stability criterion for motion planning. KAIST won the finals of the DRC [58] with the robot *DRC-Hubo+* [12], which can switch to wheeled locomotion to overcome debris. The *Hubo* robots were commercialized and are sold by the Korean *Rainbow Robotics* company [249].

The Seoul National University participated at the DRC Finals with their robot *THORMANG* [159] with a focus on the driving and manipulation tasks. Recent work deals with balance control of a biped with torque-controlled joints [176] and detecting partial contacts in uneven terrain by active exploration [177]. The motions are executed quasi-statically.

German Aerospace Center (DLR) The German Aerospace Center presented its first biped robot in 2010 [227]. The mechanical design is based on the DLR lightweight robot arm [7], enabling torque control at the joint level. The DLR Biped was later extended to the full humanoid robot *TORO* with 25 torque-controllable joints and two position-controlled joints [70].

The balance control scheme is based on impedance control for the Center of Mass (CoM) position [226]. Computed net wrenches from the CoM- and posture impedance controller are distributed to the feet; the desired joint torques are computed via inverse dynamics. The approach was later extended to multi-contact scenarios [106]. It showed impressive compliance of the robot in various balancing tasks while the robot stood in one place. The control approach was, however, not applied to walking. The walking control of *TORO* is based on the related concepts of Capture Point (CP) — a point where a robot can step to come to rest — [241] and the Divergent Component of Motion (DCM) [301], which were extended to arbitrary CoM height trajectories [67]. Although having torque-controllable joints, *TORO* was at first only walking based on joint position control with the DCM control scheme [69] due to stability problems [66]. Static walking — the feet overlap in forward direction — on torque-level was achieved later using a specialized combination of a rate limiter and filter [66]. It is unclear if the observed limiting structural vibrations are caused by methodological limitations or hardware and integration limitations. Recent work achieved dynamic walking of *TORO* at a speed of 0.37 m/s using joint torque control by extending the passivity-based balance control scheme with DCM tracking [195]. *TORO* is able to walk over grass at a walking speed of 0.15 m/s and can traverse a soft mattress with 0.05 m/s [195].

Centre National de la Recherche Scientifique (CNRS) The CNRS operates a Joint Robotics Laboratory (JRL) with Japan's research institute AIST, using the HRP robot platform. The JRL lab's research focus is the generation of complex motion plans for multi-contact scenarios [29, 30, 71, 178]. The methods facilitate the use of complex models and constraint sets, leading to planning times in the range of minutes [29] or hours [178]. More recent research is directed towards real-time capability, describing a combination of stabilization methods for stair climbing [43], or real-time static balancing in multi-contact scenarios [254]. Experiments with HRP-4 to walk on gravel with a soft sole have been conducted in [228]. The robot quasi-statically walks over the uneven ground at a speed of 0.011 m/s.

Italian Institute of Technology (IIT) Several biped robots have been developed at the Italian Institute of Technology [179, 198, 306], some of them containing serial elastic actuators with a relatively low intrinsic stiffness [179, 306]. The *WALK-MAN* robot can overcome uneven terrain based on visual data [146]. Only a single footstep is located on rough terrain, and the walking speed is unspecified; an approximation from the attached video yields ≈ 0.2 m/s. The contact with the uneven terrain significantly disturbs the robot's state. Experiments without prior knowledge on the environment or vision-based data have not been conducted to this author's best knowledge.

Boston Dynamics In 1979, Marc Raibert started with his research on one-legged hopping machines to study legged locomotion fundamentals. The pneumatically driven machines with a low leg-to-base mass ratio were able to hop and locomote in 3D using relatively simple control structures [246]. Raibert further argues "that the trotting quadruped is like a biped, that a biped is like a one-legged machine, and that control of one-legged machines is a solved problem." [246, p. 22]. Conceptually, the approach is related to the control of the instantaneous CP [241]. By defining a *Virtual Leg*, the control schemes were later transferred to the control of a quadruped [247, 248].

In 1992, Raibert founded the company *Boston Dynamics Inc.* to build the hydraulically powered quadruped *BigDog* with funding from Defense Advanced Research Projects Agency (DARPA). The machine uses control based on the virtual leg concept and is able to traverse uneven terrain carrying high payloads [245]. From 2009 to 2012, the company developed the bipeds *PETProto* and *PETMAN* based on hardware and control of *BigDog* [208]. These machines are able to walk at a speed of up to 1.96 m/s on flat ground. Later, the *Atlas* series extended the capabilities of *PETMAN* with arms to make contact with the environment or balance using angular momentum control [207]. The *Atlas* robots are able to cross challenging terrain at high walking speeds and were provided as a platform to teams participating in the DRC [58].

There is no substantial information on the control schemes used by Boston Dynamics to control these machines. One supposed big advantage to other machines is the high total mass of the robots (95-182 kg) in conjunction with the very powerful hydraulically actuated legs, which make these robots particularly well suited for stabilization by step modification.

For safety reasons, these robots' hydraulic actuation does not seem suitable for a broader range of applications, though. Boston Dynamics' first commercially available product — the quadruped *Spot* — is electrically actuated [27]. In 2016, Boston Dynamics revealed the next generation *Atlas*. To date, only videos of executed motions are available [26].

Massachusetts Institute of Technology (MIT) / Institute for Human and Machine Cognition (IHMC) At MIT, a team of researchers worked on a control framework for biped robots based on a hierarchy of Quadratic Programs (QPs) [169]. A motion planner QP computes valid footholds and trajectories based on visual information with a computation time in the range of several minutes. A second, real-time capable QP then tracks the plan while considering constraints and the robot's dynamics [168]. Experiments are carried out on Boston Dynamic's *Atlas* platform, which was provided to several US institutes, and universities in the context of the DRC [58]. The QP based framework is tested for visual-guided walking on uneven terrain. The terrain-blind traversal of uneven terrain has — to the author's best knowledge — not been tested.

In 2006, Pratt, Carff, Drakunov, and Goswami [241] described the concept of the CP — the point a robot needs to step to bring its orbital energy to zero — for the stabilization of biped robots. The approach was experimentally validated on the *Yobotics-IHMC* robot [242], is equivalent to the Divergent Component of Motion approach used in the Honda robots [299, 301], and uses a linear model for the robot's dynamics.

In the context of the DRC, a team from IHMC and MIT developed a control framework based on CP and momentum control [166]. The instantaneous CP is used to generate the CoM trajectories and the desired momentum rate of change. A subsequent QP solves for the joint accelerations and contact wrenches using the relation between centroidal momentum and wrenches on the robot. The QP combines wrench distribution, momentum control, and motion tasks in one problem, which is solved in real-time. Experimental results with *Atlas* walking over undetected rubble were shown. The walking speed is not specified; analysis of the photos of the experiment indicate an approximate speed of 0.2 m/s over the undetected uneven terrain [166]. By combining the momentum-based controller with a Center of Pressure (CoP) based contact estimator, IHMC's *Atlas* is able to balance on partial/line contacts [324]. Recent work includes disturbance rejection by step modification [91] and footstep planning for uneven terrain based on visual data [93].

Oregon State / Agility Robotics The semi-passive biped robot *ATRIAS* was developed at the Oregon State University in 2016 to study the application of the biomechanically inspired spring-mass model. The robot is equipped with two point-feet, and most of its weight is located in the upper body. This enables *ATRIAS* to walk in 3D with relatively simple heuristics for the next step location [125].

Jonathan Hurst and Damion Shelton later founded the robotics spin-off *Agility Robotics* to commercialize the research on *ATRIAS* with the biped *Cassie* [4]. *Cassie* has no upper body, and each of its legs is driven by five actuated and two passive degrees of freedom. Just as *ATRIAS*, it uses four-bar linkage mechanisms with leaf springs for the passive joints and is technically walking on its toes with a small support area. The advanced mechanical design lumps most of the mass in its upper body and enables the control of *Cassie* with simple heuristics [87]. While *Cassie* can go upstairs quite well, it is unclear if going down is possible with its inverted knee kinematics. *Agility Robotics* has recently announced a new product *Digit*, which has a full torso, perception, and arms to manipulate objects [4]. It uses the same leg kinematics as *Cassie* with larger feet. The arms have four Degrees of Freedom (DoFs) each and are used to hold an object by clamping it between the two arms (no fingers).

Others At Carnegie Mellon University, Stephens and Atkeson [283] worked on push recovery and momentum-based balancing of compliant robots [284]. The methods were experimentally validated by applying external disturbances on a Sarcos Primus hydraulic robot standing on flat ground. On the same hardware, Herzog, Righetti, Grimminger, Pastor, and Schaal [108] from the Max Planck Institute for Intelligent Systems demonstrated the effectiveness of cascaded QPs for the implementation of a similar momentum-based controller. The approach considers a full set of physical constraints and the robot's full centroidal dynamics while still being fast enough for a 1 kHz control loop. Experimental validation is carried out in several balancing scenarios while the robot stands still.

For participation in the DRC Finals, the humanoid robot *ESCHER* was built at Virginia Tech [122]. The robot with torque-controlled joints is able to overcome undetected uneven and soft terrain at a walking speed of 0.075 m/s.

Quadrupeds The control of quadruped robots poses, in general, very similar problems compared to biped robot control. Some differences to biped robots are the point feet, a larger support area, and a drastically smaller leg-to-base mass ratio. Notable research quadrupeds are *ANYmal* [126] from the ETH Zürich, MIT's Cheetah [61, 210, 230] and Mini-Cheetah [150], and *HyQ* [268] from the IIT. At ETH Zürich, methods based on Model Predictive Control with the full dynamics of the robot are used on *ANYmal* for disturbance rejection [78] or walking on soft ground [90]. Current research with the Mini Cheetah is about the extraction of simple heuristics from offline optimizations to reduce the computational power required on the robot and simplify parameter tuning [23]. At IIT, current research deals with quadruped locomotion over terrain with different compliance using a contact-consistent whole-body control scheme [75].

Quadrupeds are increasingly commercially available, e.g., Spot from *Boston Dynamics* [27], ANYmal from *ANYbotics* [10], or the A1 from *Unitree Robotics* [309].

Chair for Applied Mechanics - Technical University of Munich In the 1990s, the six-legged robot *Max* and eight-legged robot *Moritz* were designed at the Chair for Applied Mechanics using a control design based on neurobiological findings [236]. The first biped robot JOHNNIE was able to autonomously walk over known obstacles using onboard navigation and reached a maximum speed of 0.66 m/s on flat ground [186, 187]. The experience with JOHNNIE led to the design of its successor LOLA [189], which was presented to the general public in 2010. LOLA was shown to reach a maximum walking speed of 0.93 m/s on flat ground [36] and overcome simple undetected obstacles of several centimeters in height [35]. Furthermore, real-time autonomous navigation methods [112], and step modifications for disturbance rejection in the presence of obstacles [114, 328] were investigated on this platform. The LOLA biped has been continuously improved over the years and is the hardware used for this thesis's investigations.

Summary There are some remarkable particularities when looking at the state of the art in the field of biped robots: First, most high-performance bipeds are either designed by companies or private research institutes. Consequently, one could reason that system integration and the particularly time-consuming hardware design seem to influence system performance significantly. On the other hand, the control schemes and their complexity are incredibly diverse in biped robot control. Lacking hardware performance or integration may, in some cases, be compensated with more complex control algorithms. Different approaches are seldom evaluated on the same hardware, making it hard to separate problems related to the core nature of bipedal walking from problems caused by poor hardware performance or system integration. Third, most control approaches use dynamic models, but the experimental validations often show quasi-static motions only.

1.2 Objectives and Outline

The goal of this thesis is to improve the robustness of fast biped walking over undetected uneven terrain with contributions to the following biped walking research areas:

- proprioceptive contact sensors,
- hardware and software integration,
- balance and multi-contact force control algorithms.

The dynamics and state of contacts play an essential role in the control of biped walking robots. It is difficult to create precise models of the contacts made between the robot and the ground, making the measurement of contact information especially valuable for control.

Enhanced hardware and software integration improves the robot's overall performance and reduces the coupling between effects caused by unsatisfactory system performance and effects caused by the control schemes. This author hypothesizes that the higher performance of the robot on hardware and software level enables less complicated control schemes; its weakest part always limits the whole system's performance. This work tries to answer how far the limits of robustness can be pushed using elementary control policies with reduced dynamic models, making hardware and software integration especially important. The balance and force control algorithms proposed in this thesis

- rely on fast feedback loops with reduced dynamical models,
- are local in time (non-predictive) and therefore efficient,
- do not require vision-based or other prior knowledge on the environment,
- support stabilization in multi-contact situations, i.e., with additional use of the robot's hands,
- also work at relatively high walking speeds $\geq 0.5\,{}^{\mathrm{m}}\!/\!_{\mathrm{s}}$.

The approaches do not require a full multi-body dynamics model of the biped. Because identifying a precise multi-body model is time-consuming, this requirement reduces the overall engineering effort for the application to a robot. Time-locality ensures efficiency and makes it easy to use the approaches in global optimal control schemes on higher control-levels, e.g., [328].

The developed algorithms are experimentally tested on the biped robot LOLA in several uneven terrain scenarios. In all experiments, the control software assumes perfectly flat ground. To evaluate the control algorithms' performance in the context of high uncertainty, the robot's vision system is not used. Nevertheless, the approaches described in this thesis may be combined with a vision-based planner, which reduces the terrain uncertainties to the perception errors of the vision system.

The outline of this thesis is visualized in Figure 1.1. In Chapter 2, an overview of the hardware platform LOLA is given first. Furthermore, the proposed hardware control architecture and tactile contact sensor are described and experimentally evaluated. Chapter 3 gives an overview of the planning and control approach of LOLA and deals with the connection

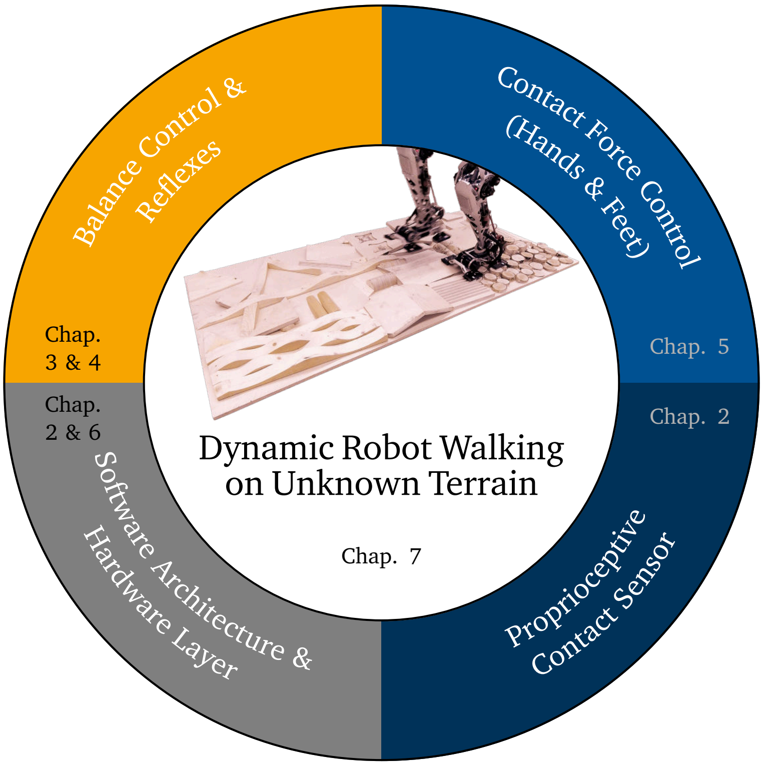


Figure 1.1: Visualization of the objective and outline of this thesis.

between feedback control loops and trajectory adaptation strategies. In Chapter 4, concepts for stabilizing the floating base in the presence of disturbances are presented, discussed, and experimentally validated. Furthermore, strategies for the distribution of stabilizing CoM wrenches to the different contacts with the environment are described and analyzed. Chapter 5 deals with proposed approaches for a contact force control scheme at the contact locations, which uses sensor-based contact information and adaptation strategies. The concepts are experimentally evaluated and enable LOLA to walk over undetected uneven terrain in partial contact situations. Details on the control concepts' software implementation are presented in Chapter 6. The software for the control of biped robots is a complex system, and despite simulative unit and system tests, errors are often only discovered in experiments. A machine-learning-based analysis tool for simulative system tests, which leverages error detection, is presented. Experimental results achieved with the combined proposed control approaches in a set of uneven terrain scenarios are shown in Chapter 7. In Chapter 8, the contributions of this thesis, conclusions, and recommendations for future work are presented.

1.3 Notation and Videos

Required fundamentals and the used notation are described in Appendix A. All symbols, indices, operators, and acronyms are hyperlinks in the electronic version of this document, directing to the respective glossary entry. The time is considered discrete, i.e., $t = n\Delta t$. For clarity of the equations, the time step is omitted when it equals the current time step n, i.e., $a = a[n]$ for any time-dependent variable a. The order in which methods are described follows the flow of data from input to output. When higher-level methods are validated, the default for the low-level modules is used unless noted differently. Links to videos of the experiments are provided at the beginning of the corresponding result sections.

Chapter 2

Platform Mechatronics

This chapter describes the platform mechatronics of the humanoid robot LOLA, i.e., every piece of platform-specific software, electronics, and hardware required to move the robot's joints and read its state. This includes the *low-level* software for joint control, communication, and data pre-/post-processing. The interface to the control software is defined by the *Hardware Layer (HWL)*, which provides general data interfaces to the *high-level* control software modules.

The platform mechatronics is extremely important for the whole robotic system's performance: All high-level modules, i.e., the Walking Pattern Generator (WPG) and Stabilization and Inverse Kinematics (SIK), rely on the performance of this combined piece of hardware and software. If seen as a cascaded structure, it becomes clear that the platform mechatronics must meet the highest requirements for update rates, latency, safety, and reliability. This chapter presents the proposed hardware control architecture for improved hardware integration and a tactile sensor concept for biped robots.

2.1 Mechatronic Design Overview

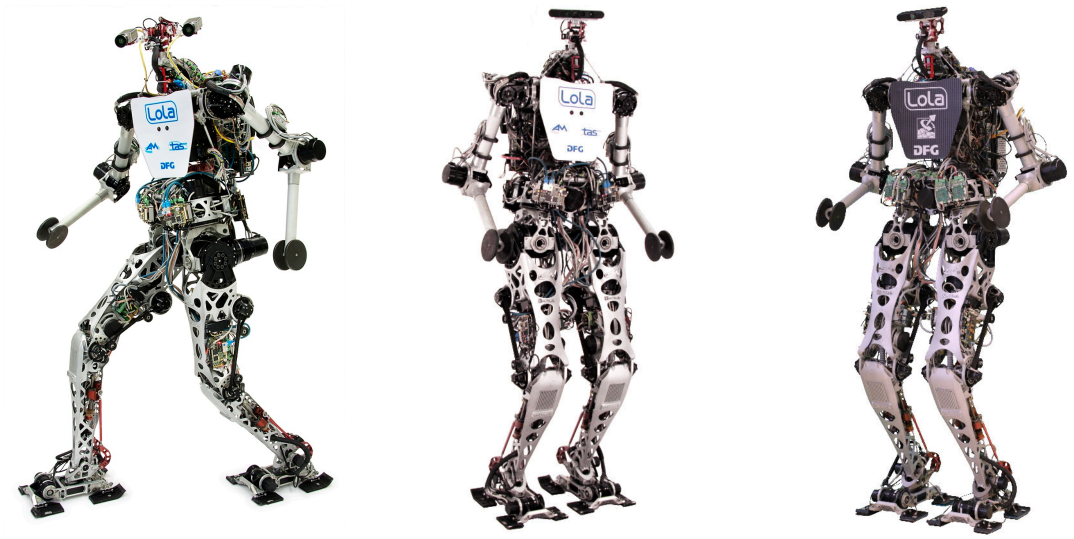
This section gives a short overview of the different design revisions of LOLA and describes the current design. Note that the original design is only described in detail where necessary for the understanding of later improvements.

2.1.1 Design Revisions of LOLA

Since its first stable design in 2010, several improvements were made to the mechatronic architecture of LOLA, see Figure 2.1. In the following, the changes leading to different hardware revisions and corresponding lessons learned are briefly summarized.

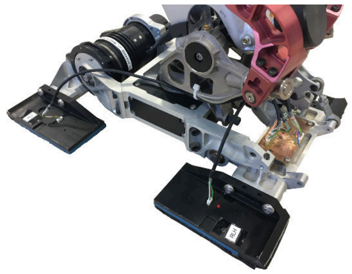
2010: Original Design The original mechatronic design of the humanoid robot LOLA [33, 81, 188] is shown in Figure 2.1a. By that time, the robot was equipped with a stereo camera head from a project partner with 3 degrees of freedom [39]. This resulted in a total of 25 degrees of freedom for the whole robot. Due to the high computational load, the vision data processing was done on computers in an external rack. However, the trajectory generation and control already ran on a central control unit on LOLA's back.

The communication and actuation structure of LOLA consisted of 9 Distributed Sensor Control Boards (DSCBs) [80]. Each board could control up to three joints via dedicated *Elmo Motion Control* servo drives [65], which were connected to a DSCB via a Controller Area Network (CAN) bus. Sensor data from the joint encoders was read in; target positions for the joints were sent to the servo drives. Additional sensor data — for example from the



(a) Lola from 2010–2014 [326] ©2014 IEEE. **(b)** Lola from 2014–2016 [327] ©2015 IEEE. **(c)** Lola from 2016–2020 [266] ©2019 IEEE.

Figure 2.1: The history of Lola.



Contact Switch Bit Encoding	
Pad	Bitmask
Right Heel	0x01
Right Toe	0x02
Left Heel	0x04
Left Toe	0x08

Figure 2.2: Lola's right foot with contact switches in the pads.

Force/Torque (FT) sensors — was directly processed on individual DSCBs. The boards also acted as slaves in a Sercos-III bus system [270] to communicate with a central control unit at a cycle time of 1.5 ms. The software processes on this control computer used the received sensor data to calculate new trajectories for every joint and sent the corresponding target positions to the distributed sensor control boards at an update rate of $666.\bar{6}$ Hz. The real-time operating system *QNX* Neutrino 6.5 was used in this setup.

2013: Contact Switches Lola's feet consist of four footpads each, making the actual contact with the ground. The original pads did not contain any sensors or electronics. In 2013, these pads were extended with an internal discrete switch to detect if ground contact was made [74]. The discrete information on the pads with closed contact is encoded bitwise in one byte for each foot, see Figure 2.2. The switches' force threshold is ≈ 30 N per pad.

2014: New Head Design The stereo camera head system with 3 degrees of freedom was replaced by a low-cost ASUS Xtion RGB-D sensor on a pan/tilt unit. This was mostly because the collaboration with the project partners for Lola's vision ended and because low-cost structured light sensors became available on the market.

2016: New Hardware Control System The original communication and joint control system had several drawbacks:

- **Unnecessary Complexity:** The design of the DSCBs was very complex. This complexity was built to enable advanced distributed control concepts — concepts that were never implemented.
- **Lacking Reliability:** Probably due to their high complexity, the control boards were unreliable. After a certain time of operation, the boards repeatably failed to boot-up and needed to be replaced. This process was repeated until the circuit components were no longer available. The problems with hardware and software caused much downtime of the robot and made it hard to concentrate on developing new methods.
- **Low Control Bandwidth:** Although equipped with an Ethernet-based Sercos-III bus, the servo drives were still connected via CAN to the DSCBs. The update cycle time was limited to 1.5 ms. Furthermore, the joints' position control loop was closed on the control boards and over the CAN. This led to additional latencies and jitter in the control loop timing.

Based on the experiences with the original control system, a new design was created. Furthermore, the real-time operating system was upgraded to *QNX* Neutrino 6.6. The new hardware control system's detailed design and characteristics are described in Section 2.2 of this thesis.

2020: Redesigned Upper Body and Arms Earlier versions of LOLA used the arms only for compensating the angular momentum of the feet during walking. In the context of a new DFG research project[1], LOLA's arms were redesigned to enable balancing in multi-contact scenarios. By adding one degree of freedom to each arm, the kinematically reachable space was extended. Also, FT sensors were added to the hands to enable feedback control of hand contact forces [267]. Hand contacts are made via spherical end effectors with rubber coating. Note that only unilateral multi-contact scenarios are considered.

The weak connection between the lower- and upper body in LOLA's original design caused structural resonances near the mounting point of the Inertial Measurement Unit (IMU). This mechanical design flaw limited the balancing controller's bandwidth and is discussed in [16]. The upper body has thus been redesigned with the intent to shift the structural eigenfrequencies to higher values. A comparison of the eigenfrequencies of both designs can be found in [17].

Additional hardware enhancements include increased computational power of the onboard computers and an updated vision system. A comprehensive in-detail description of all changes is presented in [267].

2020: New Foot-Sole Material The compliant material on the footpads was replaced with a two-layer design. The inner layer is made of the compliant shoe material *Nora* Lunasoft SLW (6mm) to absorb impacts with the ground. The compliance of this material is additionally important for the design of the contact force control, see Chapter 5. The outer layer is made of the shoe-sole material *Nora* Astral Crepe (1.8mm) and provides the necessary rigidity for contacts with rough surfaces. Furthermore, this layer provides an excellent static friction coefficient $\mu = 0.8$ [285].

2.1.2 Current Design

The humanoid robot platform LOLA is 1.76 m tall, weighs approximately 67 kg, and is actuated by 26 electric joint drives. The kinematic structure is shown in Figure 2.3. LOLA is equipped with an active toe joint, which allows to rotate the forefoot separately.

[1] German Research Foundation, grant number 407378162

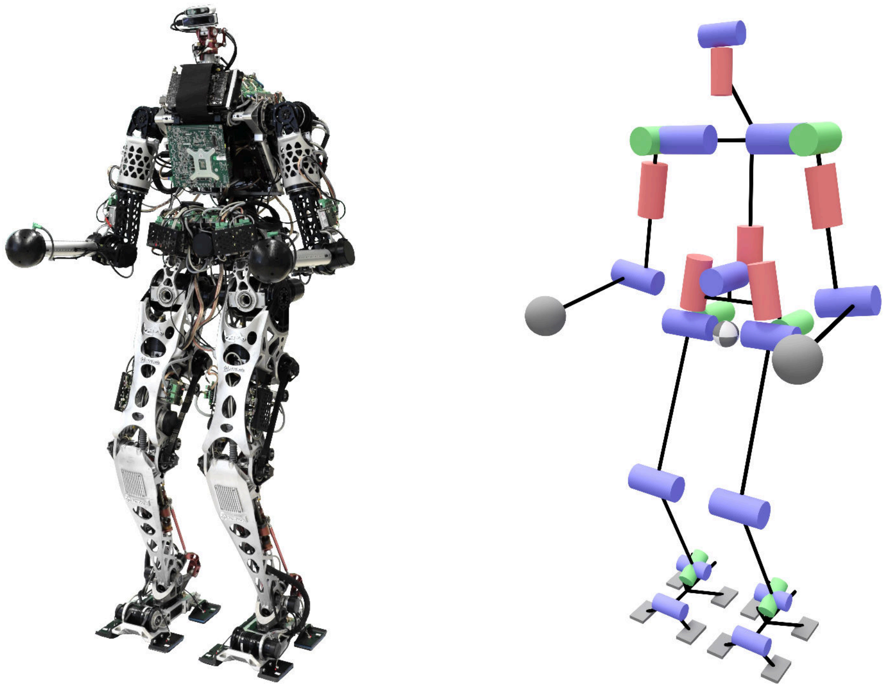


Figure 2.3: The design of LOLA and its kinematic structure at the time of writing. The current design features 26 DoFs and a redesign of the upper body / the arms.

Drive Modules Every joint drive module consists of a brushless DC motor, encoders, and a servo drive for decentralized and fast control of joint position and velocity. The servo drives are commercial Gold Line drives from *Elmo Motion Control* [64]. The distributed joint control architecture enables high local sample rates with 20 kHz for the current feedback, and 10 kHz for velocity and position. The servo controller implements a PPI cascade and uses the incremental encoder on motor-side for position and velocity feedback.

Most drive modules are equipped with stiff high-ratio Harmonic Drive transmission gears. An exception are the ankle and knee joints: The knee joint is designed as linear actuator via planetary roller screws; the ankle joints are actuated via a parallel kinematics using two spatial slider crank mechanisms and belt drives. All joints are back-drivable. Every drive module is equipped with an incremental encoder on motor side and an absolute encoder on joint side, i.e., after the gearbox or linear actuator kinematics. The schematics of a typical drive module is shown in Figure 2.4. Details on the joint drive properties are described in Appendix B. Information on the mechanical design of the drive modules can be found in [188].

Sensors The robot is equipped with the commercial IMU iVRU-FC-C167 from *iMAR Navigation* [129]. The device is mounted at the torso of the robot. It contains three fibre-optic gyroscopes, three Micro-Electro-Mechanical accelerometers, and a micro controller for internal sensor data fusion, temperature correction, and communication. The system provides the orientation, its derivative, and the linear accelerations via CAN at a data rate of 200 Hz.

LOLA is further equipped with custom 6-axis FT sensors at the feet, because there are — to date — no commercial sensors available on the market, which combine the requirements of low weight and size with the high torque specifications (±120 Nm, 1000 N, overload-safe). Measurement data is supplied via CAN at a data rate of 1 kHz. Details on the mechanical design are described in [188]. Efforts for a redesign of the sensor have been made to increase the accuracy and bandwidth while keeping the low weight of 400 g [237].

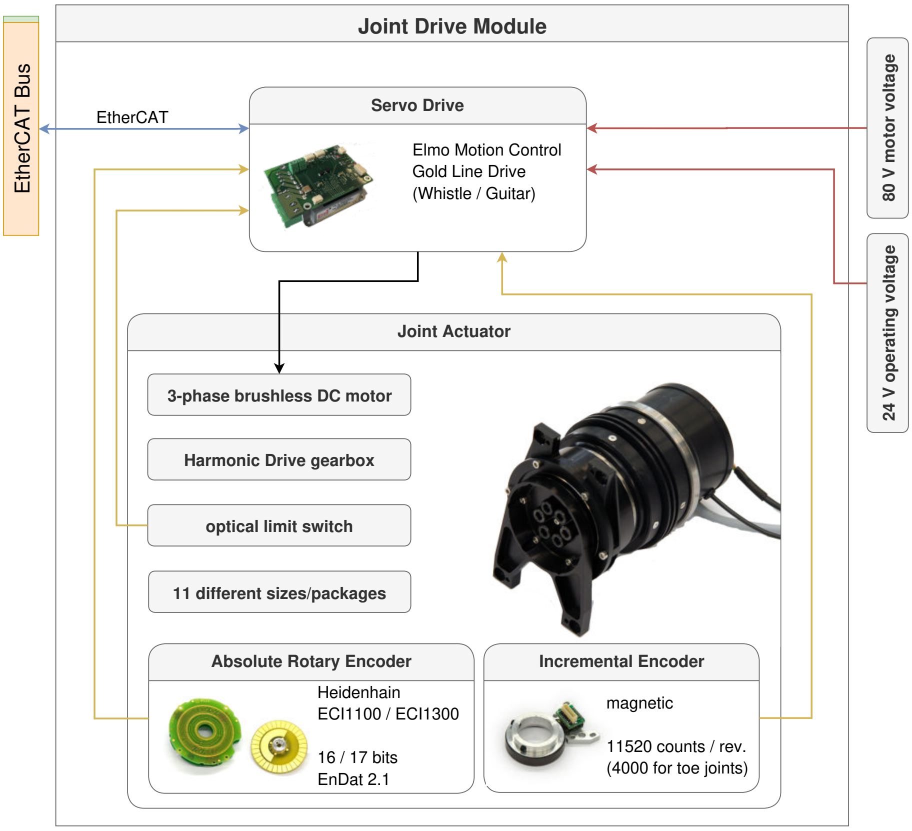


Figure 2.4: Simplified schematics for a typical joint drive module of LOLA.

The hands of LOLA feature FTE-AXIA80-DUAL SI-200-8/SI-500-20 6-axis FT sensors from *Schunk GmbH & Co. KG* with a maximum contact force of 900 N and a maximum contact moment of 20 Nm. These AXIA FT sensors are directly connected to the EtherCAT (EC) bus and support a maximum update rate of 4 kHz.

Onboard Computing & Communication The distributed servo drives and sensors communicate via EC [73] with LOLA's central control computer, see Figure 2.5. This control unit, which is located at the back of the robot, runs the real-time operating system *QNX* Neutrino 7.0 and executes all computations required to plan and control the motion of the humanoid. The onboard computer is built from an *Advantech* AIMB-276 mainboard with an Intel i7-8700 hexa-core CPU and 32 GB RAM. A second onboard computer at the front, with identical specifications, but additional Nvidia Quadro P2000 graphics card, runs a standard linux distribution and is connected to the control unit via Gigabit Ethernet. This vision processing unit and the connected Intel RealSense cameras at LOLA's head are not used throughout this thesis.

The EC bus topology is based on three lines — one for each arm and one for the rest of the slaves. The separated lines for the arms are necessary due to missing second ports on the commercial FT sensors of the arms. CAN-based data is integrated via a CAN-EC gateway [72] with 1 Mbit/s bandwidth. The EC bus runs with an update frequency of 4 kHz and uses the Distributed Clocks feature of the EC technology. In Table B.4, a detailed description of the bus topology is presented.

Power Network LOLA does not have an onboard power supply; it is operated tethered with a 24 V supply line for the electric components, and a 80 V supply line for the motors (except head joints). The power supplies are located in a nearby rack.

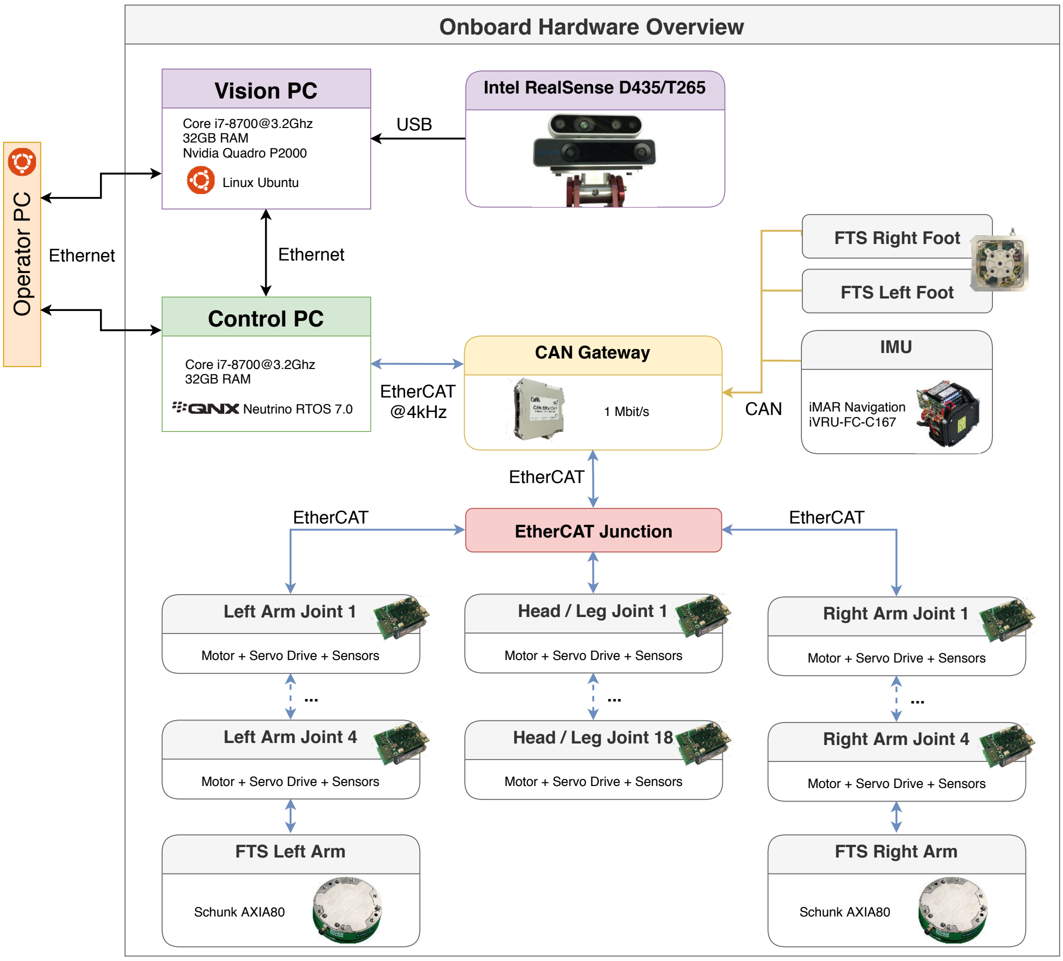


Figure 2.5: Overview on the computing and communication components onboard of LOLA.

Operator Commands Both onboard computers are connected to an operator PC via Gigabit Ethernet. The connection is used to send high-level control commands and parameters to the software running on the control unit via a publish–subscribe system. Furthermore, logfiles and executables are transferred via Ethernet to the onboard computers.

Hardware Layer Application The *Hardware Layer Application* runs the HWL in the software architecture of LOLA. It is responsible for all low-level communication and safety measures, handles hardware errors, implements device abstractions to represent their state, and provides a shared-memory based sensor and target data interface to the high-level control modules. The module is further described in the following section.

2.2 EtherCAT-based Real-Time Hardware Layer

The content in this section has previously been published in [296] ©2018 IEEE.

The following section describes the proposed EC-based hardware control software for LOLA to improve hardware integration and overall performance. This control software is responsible for executing and controlling target motions for the robot. Furthermore, sensor data is collected and sent to the higher-level planning and stabilization modules of the robot. The performance, i.e., control bandwidth and latency, of this hardware-near control structure is essential because it represents the inner control loop for all higher-level software and, therefore, directly limits the overall system's performance. The key features of the new hardware control system design are — in the order of priority:

- **Reliability:** More commercial components and less hardware-near custom code. By reducing the complexity and deploying widely used technology, the risk of failure is minimized.
- **Bandwidth:** The EC field-bus technology and commercial servo-drives with high local update rates allow high bandwidth of the hardware control system. The control loop and all involved software components are designed for hard real-time constraints with low communication latency and low jitter, i.e., a low standard deviation of the control loop update rate. The communication system allows update rates $> 1\,\mathrm{kHz}$.
- **Compatibility:** The EC technology is a wide-spread standard and allows easy integration of many commercial sensors and actors in the future.
- **Portability:** The hardware layer tries to be as generic as possible. It implements a real-time bus middleware to decouple the field-bus characteristics from those of the connected devices and the higher-level modules. This makes it easier to change technologies (devices or the bus itself) in the future.

In the following, the components of the real-time capable HWL are described. This includes data pre-/post-processing, the real-time bus middleware, device drivers, and safety measures. Subsequently, the software's performance, the EC bus, and the overall system are analyzed. Parts of the HWL code are available open-source, see Chapter 6.

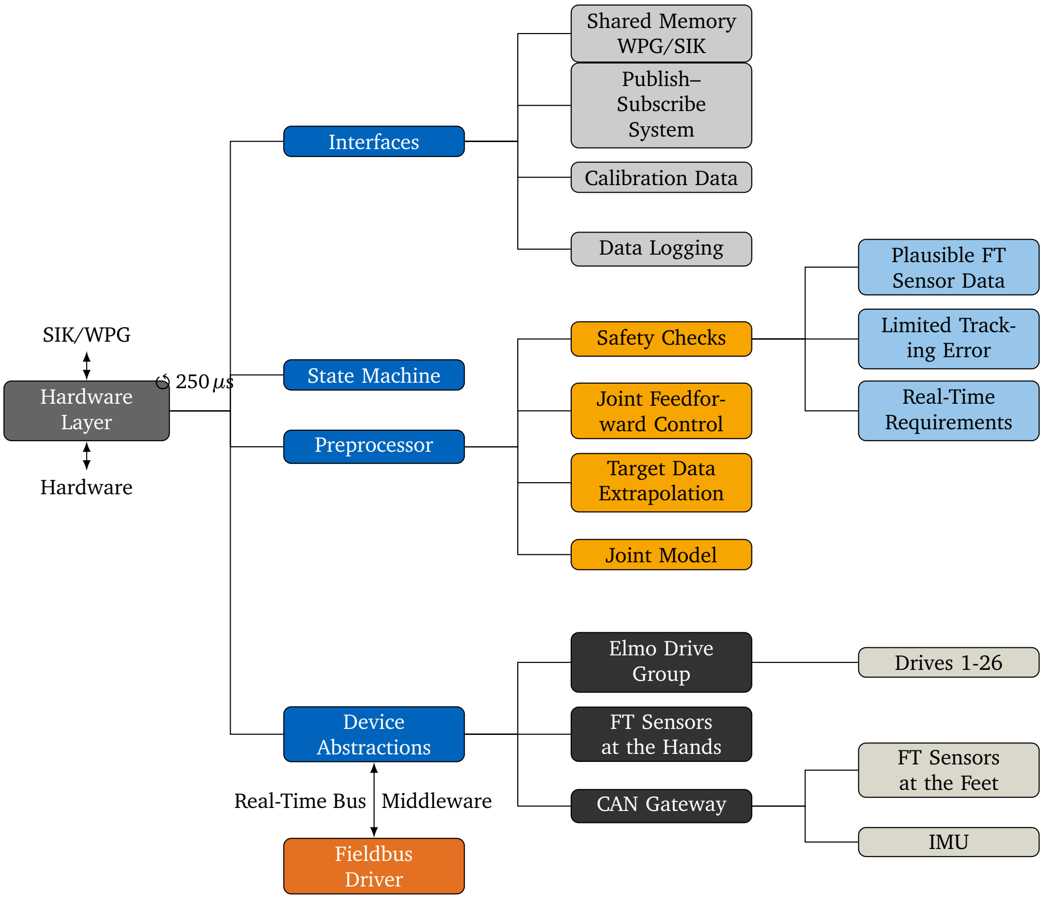


Figure 2.6: Overview of the architecture of the HWL application with major components.

2.2.1 Software Architecture

An overview of the HWL's software components is visualized in Figure 2.6.

Interfaces and Data Preprocessing The HWL receives target data from high-level robot control processes and, at the same time, sends sensor data to these processes. The communication is done via a synchronized shared memory interface [33]. All higher-level modules wait for new sensor data from the HWL, generate new target data, and finally copy it to the shared memory region. This means the HWL defines the timing of all software processes — including the high-level modules. While the HWL blocks the robot control processes until new sensor data is ready, the HWL itself is never blocked to ensure high timing accuracy and safety. Instead, it always uses the newest available target data in the shared memory region.

With the new EC bus system, communication cycle times as fast as $\Delta t_{\text{bus}} = 250\,\mu s$ can be attained for Lola. However, the high-level control modules typically can, because of the computational complexity of the control algorithms, not keep up with these low cycle times. Therefore, the HWL publishes sensor data just every nth bus cycle to trigger high-level module execution at the higher control cycle time $\Delta t_{\text{cont}} = n \Delta t_{\text{bus}}$. The last target data in the shared memory region is linearly extrapolated for all minor bus cycles $C_m = \{i \,|\, i \mod n \neq 0\}$ based on the target data gradients (e.g. velocities).

In addition, target and sensor data preprocessing includes converting motor to joint angles and vice-versa and converting raw sensor data to physical units. These conversions include the resolution of the knee- and ankle-joint kinematics with a non-constant gear ratio. Moreover, a feed-forward control strategy on velocity-level is implemented to further reduce joint position tracking errors [296]. The technique uses a learned feed-forward gain to compensate for typical velocity-dependent position errors, e.g., caused by friction.

Parameters and operator commands are received via network using a publish–subscribe system. The robot's kinematics can be calibrated using a specialized calibration rig. The HWL stores relevant joint calibration data, which is used when homing the joint drives.

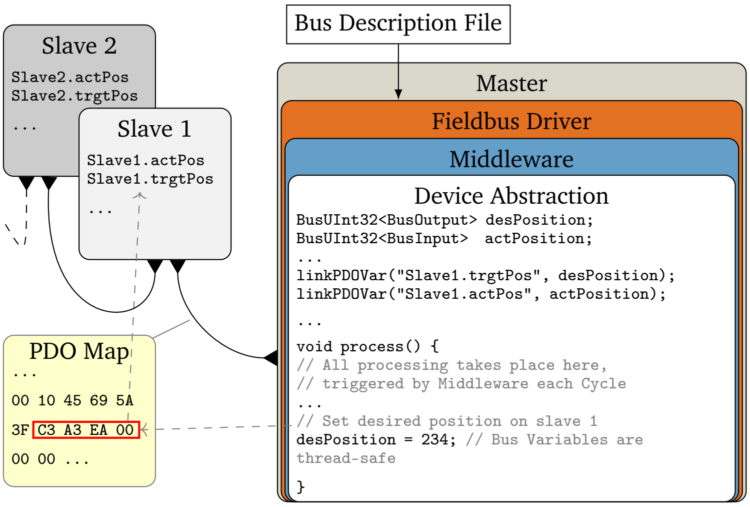
State Machine and Safety The HWL contains a state machine to represent the state of the communication bus and all attached devices. If a device is entering an invalid or unsafe state or experiencing communication errors, the software tries to bring the robot to a safe halt. Once the machine is in such a fault state, the operator has to explicitly acknowledge continued operation before the joint drives can be reactivated.

Also, safety checks are part of the data preprocessor. When FT sensor data is not plausible or the tracking error on a joint is above a specified limit, the HWL state machine assumes a fault and puts all devices in a safe state. Also, every sensor data frame sent to SIK/WPG carries a working counter number, which increases every cycle. The working counter from the sensor data is sent as part of the planner data from WPG to SIK, and as part of the target data from SIK back to the HWL. For every sensor data frame sent to the higher-level modules with a certain working counter, the HWL expects a target data frame from SIK with exactly the same working counter in the next cycle. If the counters do not match, either SIK or WPG were too slow and violated the real-time requirements. A working counter mismatch does trigger a fault when it happens at two consequent cycles.

Real-Time Bus Middleware To abstract the communication between nodes in a Fieldbus, two common CANopen [40] definitions are used: Process Data Objects (PDOs) and Service Data Objects (SDOs). These concepts are also widely used in modern Fieldbus technology such as EC and define the name, type, and size of objects, which can be received from and/or sent to a slave. PDOs define data objects, which are cyclically sent and received in real-time; SDOs are used for asynchronous communication based on a request-response pattern and do typically not meet hard real-time constraints. Furthermore, the following definition of the Fieldbus' state based on the EC standard is used in the middleware:

- **Init**: Initial state of a slave.
- **Pre-Op**: Initialization done; SDO but no PDO communication yet.
- **Safe-Op**: PDO data is exchanged; PDO output data is not yet applied, i.e., physical outputs remain in a safe state.
- **Op**: PDO data is exchanged; PDO output data is applied on the slave, i.e., physical outputs are changed according to the PDO.

In the following, a new approach to decouple the Fieldbus communication from the application software is proposed. A middleware software layer makes the bus completely transparent for the application software. It makes device-specific implementations, e.g., interpreting sensor data or abstracting the device's state, independent from the Fieldbus protocol and implementation. This has advantages in handling the software (and system) complexity and improves maintainability and safety. The core idea behind the abstraction is the proposal of *Bus Variables* to make PDOs and SDOs available to the HWL application. Basically, a Bus Variable is an instance of a special class representing a variable of a certain predefined primitive data type (int, float, char, etc.); it may be used as any standard variable in the application code. However, the Bus Variable can be linked to a PDO of a slave by telling the middleware the name and slave identifier of the PDO object. The Bus Variable then serves as a



```
BusUInt32<BusOutput> desPosition;
BusUInt32<BusInput>  actPosition;
...
linkPDOVar("Slave1.trgtPos", desPosition);
linkPDOVar("Slave1.actPos", actPosition);

...
void process() {
// All processing takes place here,
// triggered by Middleware each Cycle
...
// Set desired position on slave 1
desPosition = 234; // Bus Variables are
thread-safe
}
```

Figure 2.7: The proposed middleware decouples the device abstraction layer (application software) from the Fieldbus driver and Fieldbus type by using Bus Variables. These variables are linked to PDOs in the PDO map via the name and type information in the bus description file. Adapted from [296] ©2018 IEEE.

fully transparent representation of the distributed slave objects. If an Output Bus Variable's value is changed, the middleware automatically sends the new data to the corresponding slave. Equivalently, the data in Input Bus Variables is updated every time new PDO data is received from the bus slaves. Bus Variables are thread-safe and implement an automated data-type checking during run time. Furthermore, Bus Variables can be used for SDO-based communication. The concept is visualized in Figure 2.7.

The middleware software layer defines an abstract data interface to the Fieldbus driver, which then translates changes in the PDO/SDO data to data packets on the bus. Note that one can use different implementations for different Fieldbus types or driver implementations without changing code on the device abstraction or application side. For LOLA, a commercial EC master stack is used as Fieldbus driver [3]. The connection between the Bus Variables in the application code and the variables on the slaves is made through the respective slave and variable names defined in the EC Network Information File (ENI), which is the bus description file for EC.

The framework for Bus Variables is available open-source as part of a header-only C++ library, see Section 6.2.

Device Abstraction Layer On top of the middleware layer, all devices on the bus are represented by device abstraction classes. The class instances map the internal logic and physical behavior of the slaves to the software. The device abstraction classes are derived from a general BusDevice class provided by the middleware, which makes the use of Bus Variables in these implementations straightforward.

For LOLA, there are device classes for the *Elmo Motion Control* servo drives, the CAN gateway, the IMU, and the FT sensors. Every device class implements a state-machine, specialized error handling and provides an Application Programming Interface (API) to control the corresponding device. For example, the FT sensor device class provides methods that return the current force and torque readings; the servo drive device class accordingly provides methods to change the joint controllers' set points. These API methods are called from the data processing module of the HWL, see Section 2.2.1. Some common device abstraction implementations based on Bus Variables are available open-source, see Section 6.2.

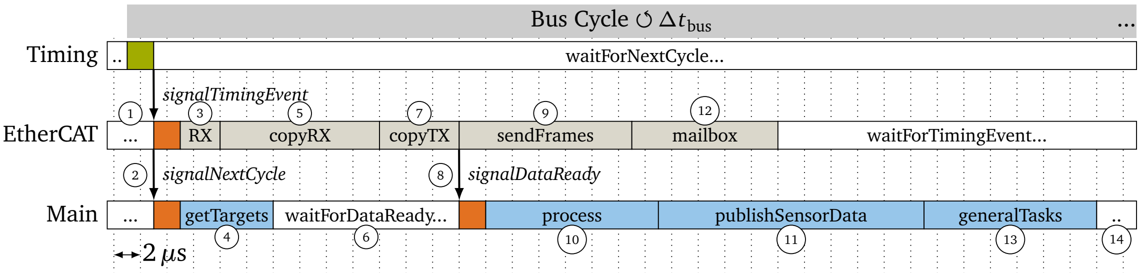


Figure 2.8: Timing and synchronization of the three main threads in the HWL. The timing thread unblocks with Δt_{bus}. Orange sections describe CPU-time used by the *QNX* Neutrino scheduler. The sections are described in Tables 2.1 and 2.2. The figure is to scale with the given legend. Adapted from [296] ©2018 IEEE.

ID	Name	Description
↶		
1	waitForTimingEvent	Wait for an event from the timing thread
2	signalNextCycle	Unblocks the main thread
3	RX	Process the frames received in the last cycle (EC stack)
5	copyRX	Copy incoming data from the EC stack to the Bus Variables
7	copyTX	Copy outgoing data from the Bus Variables to the EC stack
8	signalDataReady	Signal main thread new incoming data is available and Output Bus Variables may be modified
9	sendFrames	Queue cyclic EC frames to be sent on the bus
12	mailbox	Send acyclic frames and execute administrative tasks of the EC stack
↷		

Table 2.1: Execution sections called in the EC-thread of the HWL. Adapted from [296] ©2018 IEEE.

2.2.2 Software Integration and Timing

The HWL contains both the device abstraction layer and the middleware. It spawns several child threads for the bus communication, logging, operator interfaces, and inter-process communication with the other two control framework processes (SIK/WPG). The main thread runs the data processing, the state machine, and error handling in a cyclic loop, see Section 2.2.1. It is synchronized to a higher-priority EC-thread to ensure minimum latency of input and output data. Within the EC-thread, methods of the bus middleware and the EC master stack are executed. This includes copying data between the Bus Variables and the actual EC PDO map and sending all cyclic and acyclic EC frames. To keep the timing between consecutive EC cycles precise, a separate timing task with the highest priority is used to trigger the EC-thread execution.

In Figure 2.8, the timing and synchronization of the three threads is shown. All important execution sections of the EC- and main thread are described in Tables 2.1 and 2.2. A particularity of the approach is the interleaving of sections that access the same Bus Variables. Section 4 and 7 both operate on the target data Bus Variables and, in theory, simultaneous access may occur when section 4 has a higher execution time than section 3 and 5. Although Bus Variables are thread-safe, the process data may become invalid when data in different Bus Variables of the same device becomes inconsistent. There is more sensor data than target data to process in practice, at least for humanoid robots. This means section 5 very likely has a higher execution time — more Bus Variables need to be processed — than section 4. This concept seems unsafe at first, but it reduces the latency of the whole control loop — i.e., from sensor data over the high-level modules to target data on the devices — by one bus cycle Δt_{bus}. Furthermore, the execution times are deterministic and can be analyzed by high precision measurements with a trace logging tool to prove the concept's viability.

ID	Name	Description
↶		
4	getTargets	Get new target values from high-level control modules (if available)
6	waitForDataReady	Block the thread until new input/output data has been processed in EC-thread
10	process	Process raw input data in Device Abstractions
11	publishSensorData	Push processed input data to shared memory. This triggers a new calculation of target data in the high-level modules and is skipped for all minor bus cycles C_m
13	generalTasks	Executes state-machine logic, error handling, and communication with the operator computer
14	waitForNextCycle	Blocks the thread until the EC Cycle Time has elapsed
↷		

Table 2.2: Execution sections called in the main thread of the HWL. Adapted from [296] ©2018 IEEE.

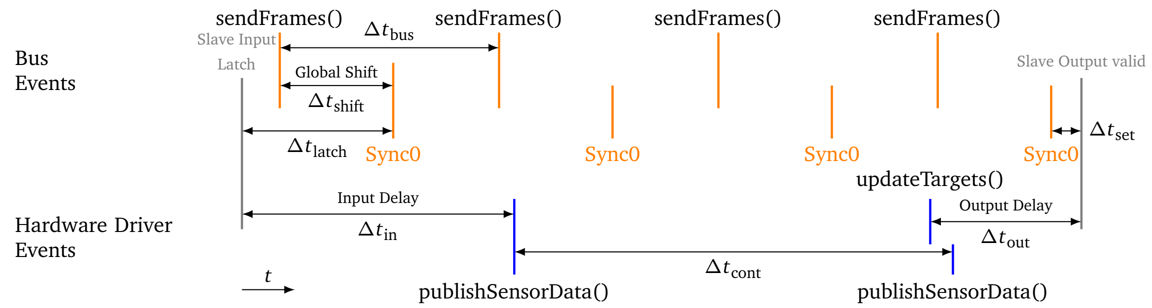


Figure 2.9: Visualizes the input- (Δt_{in}) and output delay (Δt_{out}) for slave input and output data in the HWL. All slaves and the master are synchronized to the *Sync0* event of the EC distributed clocks functionality. The control delays depend on the settings for the distributed clocks master synchronization (Δt_{shift}), and the slave's input data latch (Δt_{latch}) / output set (Δt_{set}) delays. The visualization is simplified to a bus with identical slaves and $\Delta t_{\text{cont}} = 2\,\Delta t_{\text{bus}}$.

2.2.3 Communication with the Bus Slaves

The slaves in an EC bus are organized in a logical ring — independently from the physical topology of the slaves. When an EC frame with new data is sent from the bus master, it passes every slave and is finally sent back to the master by the last logical slave. Each slave reads and writes data to the EC frame while passing the device, i.e., "on the fly". This enables short delays, resulting in low minimum cycle times, and avoids collisions by design. Furthermore, bandwidth usage is optimized as input- and output data belonging to different slaves may share the same space in the EC frame. Due to the logical ring and the propagation delays of the physical transport medium, however, an EC frame passes every slave at a different time. This means that the slaves' input- and output data is not synchronized and may represent the system's state at different points in time. To overcome this problem, master and slaves can be synchronized with the EC Distributed Clocks (DC) functionality. The synchronization of the master operates in *BusShift* mode, i.e., the time of the slaves is changed depending on the timing of the bus master. All slaves and the master are synchronized to a global-in-time event called *Sync0*, see Figure 2.9. The *Sync0* event is shifted relative to the sendFrames() operation of the bus master with the *Global Shift Time* Δt_{shift}. The hardware driver's input- (Δt_{in}) and output delay (Δt_{out}) depends on the slave time required for latching/processing a measured input Δt_{latch}, the slave time required to set an output value Δt_{set}, the bus cycle time Δt_{bus}, and the global shift time Δt_{shift}. Note that for a bus with identical slaves, $\Delta t_{\text{latch}} > \Delta t_{\text{shift}}$ and $\Delta t_{\text{in}} > \Delta t_{\text{bus}}$ holds.

The global shift time Δt_{shift} is subject to tuning. It needs to be as low as possible to reduce the output delay Δt_{out} and as high as required to allow all slaves to receive the EC

Quantity	Mean or norm value	Standard deviation	Data Source
Input PDO data size	822 bytes	-	ENI
Output PDO data size	526 bytes	-	ENI
Total PDO data size	1348 bytes	-	ENI
EC datagram size	974 bytes	-	ENI
Global Shift Δt_{shift}	140 μs	-	-
Propagation Delay	114 μs	6 μs	Wireshark measurement
Minimum input data delay $\Delta t_{\text{in,min}}$	$\Delta t_{bus} + 35\,\mu$s	n.a.	tracelogger
Minimum output data delay $\Delta t_{\text{out,min}}$	$\Delta t_{\text{shift}} + 20.5\,\mu$s	n.a.	tracelogger
Minimum total control latency	$\Delta t_{\text{in,min}} + \Delta t_{\text{out,min}} + \Delta t_{\text{cont}} - 56\,\mu$s	n.a.	tracelogger
Bus cycle time Δt_{bus}	249.3 μs	2.69 μs	High Performance Event Timer
Master Sync0 Timing Error	147 ns	840 ns	master stack
Slaves Sync0 Timing Error	1.5 ns	94 ns	master stack

Table 2.3: Parameters and performance quantities for the HWL and the EC bus system.

frame before the scheduled *Sync0* event. This means the propagation delay of the EC frame to the last logical slave defines the absolute minimum for the global shift time. To optimize the global shift time, the propagation delay is measured using an Ethernet switch between the master and the first slave and a connected measurement computer with the open-source software Wireshark [305]. The time delay between the outgoing (coming from the master) and incoming EC datagram belonging to one bus cycle is a conservative measurement of the propagation delay. Additional latencies from the switch and the measurement computer are added. The mean value of the propagation delay over a period of ≈ 13 seconds is 114 μs with a standard deviation of 6 μs. For robust operation, the global shift time is set to 140 μs.

2.2.4 System Performance

The performance of the HWL in terms of data processing and communication is evaluated in the following. For all further tests, the bus runs at an update rate $\Delta t_{\text{bus}} = 250\,\mu$s, with a high-level control update rate $\Delta t_{\text{cont}} = 1\,$ms. Higher bus update rates are not possible with the current control unit's hardware. The limited accuracy of the real-time clock does not allow BusShift operation of the DC at higher update rates. The parameters and measured performance values are summarized in Table 2.3.

The system's software-side performance is measured on the real-time control unit using *QNX* Neutrino's trace logging features. This enables to get precise information on the timing of individual program sections. The exact execution times for all code sections in one EC bus cycle are visualized in Figure 2.8. Furthermore, these measurements are the basis for the calculation of the input- and output data delay. The installed joint drive modules do not provide information on the actual latch- Δt_{latch} or set-delays Δt_{set} via the slave's object dictionary. Therefore, only the minimum delays, usually only a few microseconds less than the actual ones, can be calculated from the trace logging data.

For the chosen bus cycle time, the total latency caused by the hardware control system is only 389.5 μs. The total control latency, including the high-level control modules' computation time, results in 1389.5 μs.

The timing error of the global Sync0 events on the master and the slaves — the system is synchronized via the DC feature — is measured by the master stack; statistical information on the error is given in Table 2.3. In 99.99 % of the time, the error of the master timing is less than 1.4 % of the bus cycle time. The timing of the slaves is even more accurate.

The accuracy of the timing thread (bus cycle time) is measured online with a high-precision CPU-internal timer. As the bus cycle is the data source for all high-level control modules, errors in this timing primarily affect the correctness of these modules' calculations. Relative to the control module cycle time $\Delta t_{\text{cont}} = 1000\,\mu s$, the error is less than 1.2 % in 99.99 % of the time.

Additional data on the performance of the joint controllers itself can be found in [296].

2.2.5 Related Work

In the following, the proposed hardware control system is discussed in the context of related work. First, real-time software frameworks in the context of robotics are compared with the proposed real-time bus middleware. Second, the achieved performance of the combined hardware and software concept is benchmarked to state-of-the-art robots.

Software Frameworks Commonly used frameworks for the control of robots, e.g., the Robot Operating System (ROS) [244] or Yet Another Robot Platform (YARP) [197], do in general not support hard real-time constraints. While this is not a problem for high-level communication and planning, the frameworks are not suited for low-level motion control. Using ROS or YARP in such a context is, of course, possible but will introduce jitter in the timing of the control loops and unnecessarily limit the minimum reachable sample time.

To mitigate this problem, several (not so commonly used) frameworks provide real-time capable communication. The OpenRTM-aist Software supports real-time control threads, provides a graphical user interface for the design and connection of components, and uses code generation to speed up the development process [9]. Unfortunately, the timing and internal latency of the software framework are not analyzed. Compared to the hardware control system presented in this work, OpenRTM-aist provides a more general software generation approach for robots.

Another robot platform software, XBotCore, provides a lightweight shared-memory-based communication structure directly connected to an open-source EC master stack implementation [205]. The interface between the EC bus, the device abstractions, and high-level software is defined by typical robot devices, such as joints, IMU, and FT sensors and not on PDO/SDO level. Data showing the control period for the robot WALK-MAN using XBotCore is presented in [205]. While the control period is below the target of 1 ms, there is significant jitter in the timing.

To the author's best knowledge, the concept of Bus Variables has not been presented in literature so far. In contrast to related work, the interface between the communication bus and the device implementations is at variable-level, effectively separating device-specific code from the bus technology and master implementation. Despite this fine-grained interface, the presented hardware layer concept supports hard real-time requirements and minimal control loop latency.

Hardware Layer Performance Although modern real-time bus technologies have been available for quite some time, only a few humanoids are equipped with high-performance communication systems. In contrast to distributed control systems with a digital bus communication system, central control concepts were used, e.g., in the Honda humanoid robot [115], HRP-2 [141], or Wabian-2 [220]. All sensors and actuators are directly attached to I/O interface boards in the central control computer in these architectures. As an advantage, high update rates are possible as no communication bus is needed. However, the system's complexity is high, as all peripheral sensors and actuators must be connected directly to the central control unit. Due to their high complexity, centralized systems are more error-prone than distributed systems, where some error checking and handling are already executed on intelligent distributed slaves. With decentralized control concepts, parts of the computational effort can be off-loaded to the intelligent actuator controls. This also allows for very high update rates of the local control cycles (e.g., 20 kHz current control).

Because of its easiness and reliability, CAN is a prevalent technique for communication in distributed control systems of humanoid robots. It is used in popular robots such as the HRP robots version 3 and 4 [142, 144], Hubo-2 [48] and the iCub [198] to send and receive data to and from the distributed joint controllers. However, the maximum bandwidth of CAN is relatively low (1 Mbit/s) and communication is only partially deterministic (for high priority messages). Therefore, multiple CAN networks are used in parallel for robots with a high number of DoFs. Still, the maximum achievable update rates for joint-controller set-points are considerably low. The DRC-Hubo uses CAN for communication and is limited to a control rate of 200 Hz [333]. With the use of four parallel CAN-Buses, CHIMP reaches an update rate of 500 Hz [282]. Also, the CAN protocol does not allow to compensate for the transmission delays, i.e., allow synchronous execution of commands on the distributed joint controllers.

Several different approaches were used in literature to overcome the drawbacks of CAN. [310] proposes the use of multiple RS422 connections with an effective data rate of 6 Mbits per connection and a central control system with ART Linux operating system. However, it is unclear what kind of protocol and media access control is used in the daisy-chain setup of the RS422 interface. In [6], a real-time communication system based on the Ethernet protocol is developed for the HRP-3P (prototype). It uses a custom protocol to link several bus nodes, which operate an ART-Linux real-time operating system. While originally designed to replace the unreliable central control system of the HRP-2, the AIST group later switched to CAN for the final HRP-3 humanoid "to improve reliability and maintenance of the system". [142, p.2476]. PETMAN [208] uses a modified CAN bus to reach an update frequency of 1 kHz, details have however not been published so far. Unfortunately, there is only little information on the hardware of ATLAS (1 kHz update rate) [169] and no information on the inner structure of Honda's ASIMO. For the robot TORO [70], a Sercos-II bus with a bandwidth of up to 16 Mbit/s is used. Although this enables a 1 kHz control rate, the bitrate would probably not allow for much higher update rates. The former hardware design of the humanoid robot LOLA, see Section 2.1.1, used a Sercos-III bus based on 100 Mbit/s-Ethernet [81]. However, the distributed I/O boards and interfaces to the actual joint drives were in-house made and complex. As this introduces another source of errors, the reliability of the whole solution was limited.

For the design of newer robots, the EC bus became increasingly popular, as it is fast, reliable, and widely-used. In-house-made electronics are often used for the actual joint control, and fault handling in such systems [83, 139], which may reduce such solutions' reliability and persistence. Furthermore, often non-real-time capable software frameworks are combined with the EC technology, giving away its advantages of low communication delays [83]. The latency of the communication or the control loop is in general not analyzed.

The RoboSimian robot from the Jet Propulsion Laboratory features a similar hardware structure using an EC bus and the same commercial joint servo drives [147]. The control loop runs at 1 kHz on a non-real-time operating system. There is no data given on the latencies of the control loop. The biped robot TALOS from *PAL Robotics* uses an EC bus for its communication network with update rates beyond 1 kHz. The hardware layer interface is based on a ROS node, and there is no data given on the latency of the communication or control loop [280]. The recently presented robots HRP-5P [145], and CENTAURO [148] use an EC network. There is no information given on the used control rates or latency.

Compared to the systems found in literature, the proposed control architecture operates

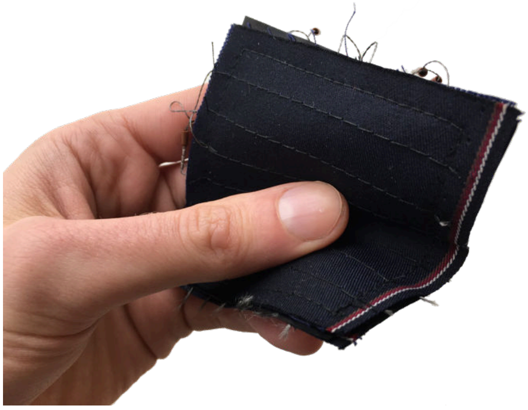


Figure 2.10: Picture of the proposed tactile sensor (prototype *C*) [291] ©2017 IEEE.

at an update rate of 4 kHz. The system has explicitly been designed and evaluated for low control loop latency. The approach uses reliable and available hardware modules. The only other example found in literature specifies a 2 ms latency at a 1 kHz bus update rate for the humanoid robot TORO [70]. However, it is not specified how the latency is measured and what is included in the value. The total control latency of the proposed architecture currently is < 1.4 ms, which is mostly limited by the high-level control modules.

2.3 A Flexible and Low-Cost Tactile Sensor for Robots

The content in this section is published in [291] ©2017 IEEE.

While today's industrial robots present potential harm and are thus caged with safety fences, the next generation of robots will have to provide intrinsic safety without external measures. This paradigm change enables operation in human environments, and an expansion of robotic applications is expected, e.g., to health-care and service robotics or human-robot cooperation. In these scenarios, the robots must handle unstructured, unknown, or changing environments, requiring them to safely interact with the environment or detect contacts if they occur.

There are many approaches to achieve safe motion planning for robots in human-centered environments. Typically, these use visual information, prior knowledge on the environment, or FT sensors, e.g., in the robot's joints. However, these methods generally do not perform well when the line of sight is obscured or objects with unknown mechanical properties are encountered. The common use of FT sensors provides only limited information on the pressure distribution, i.e., the contact location and surface. Moreover, contacts with the environment are generally avoided. The sense of touch is crucial to explore unknown and unstructured environments, as contacts with the environment can be accurately detected and localized.

In the following, the development of a tactile sensor for robots is described. It uses a set of tactile pixels — commonly called taxels — to sense the pressure on a sensor surface. Three different designs are analyzed and compared. The final design, see Figure 2.10, is selected based on experimental results. The sensor is generally suitable for a wide range of applications, not only for humanoid robots. The presented designs are motivated by the experience gained from preliminary studies [100].

2.3.1 Design Overview

In this section, the design criteria, and materials for the proposed sensor are described.

Design Goals The tactile sensor requirements are defined with several use cases in mind: First, tactile sensing on robotic fingers with a high spatial resolution (2–3 mm). Second, a flexible tactile skin may cover a humanoid robot's whole body to detect collisions with the environment and the precise contact location. Third, a tactile foot sole on the feet of a humanoid robot to detect the contact state. Based on these applications, the design goals for the tactile sensor are defined. The tactile skin has been developed due to the lack of commercial alternatives which meet the following resulting base requirements for a robotic application:

- High spatial resolution
- High resolution and linearity of a calibrated taxel (for spatial interpolation)
- Mechanical robustness and flexibility
- Low cost
- Easy to replicate
- High update rates $\geq 1\,\text{kHz}$, low mass, and low energy consumption

The main objective of the sensor is to detect accurate positions of contacts on the sensor surface. Interpolation algorithms may be used on the pressure readings to improve the overall spatial resolution for a given amount of taxels. Interpolation requires an acceptable accuracy and linearity of the sensed pressure on each taxel. Any calibration of the sensor should be easy and ideally only be carried out once. For good integration on arbitrarily shaped surfaces, the material must be flexible, and it must be possible to customize the shape and spatial resolution of the sensor. Further essential requirements are low cost and the ability to replicate the sensor element with commonly available tools and materials. This enables the use of the sensor concept as a tool in research projects.

General Concept and Materials The selection of the materials largely influences conformance with the requirements. In the following, the general concept and materials are motivated and described. The sensor is based on the piezoresistive effect, enabling the measurement of both static and dynamic pressures. If pressure is applied to such a piezoresistive material, its electrical resistance decreases. The resistance is measured via conductive electrodes on both sides of the sensor layer. Many taxels are arranged next to each other to achieve a high spatial resolution. In order to keep the number of necessary connecting wires low, the commonly-known matrix structure is used. All electrodes on a vertical line are connected to one wire, and all electrodes on a horizontal line are connected to another common wire. Each taxel's resistance in the matrix can be measured by selecting the right combination of vertical and horizontal wires.

Based on the design goals, the sensor design should consist of commonly available and low-priced materials. Retail prices are used to estimate the overall costs. The search led to the choice of Velostat™ 4540 EVA Copolymer (≈ 50€/m²), which is produced by 3M and consists of a polymeric foil (11 μm thickness) impregnated with carbon black. While this material is intended for the packaging of electrostatic-sensitive devices, it is known for its piezoresistivity. For the electrodes, which also serve as connecting wires, a commonly available stainless steel thread (2-ply and 3-ply) at costs of ≈ 0.26€/m is used. An outer protection layer is made of a cotton fabric (≈ 3€/m²), or neoprene[2](≈ 25€/m²).

[2] SEDO Chemicals Neoprene LS, 2 mm thickness

Prototype Description	Drawbacks
Similar to prototype *A*, but with a silicone cast instead of the neoprene cover	High pressure thresholds High bending radius
Similar to prototype *B*, but laminated with insulating foil instead of the neoprene cover	Air between the contacts High bending radius
Similar to prototype *C*, but with the threads directly attached to the polymer using an adhesive foil and neoprene cover	Difficult to manufacture

Table 2.4: A list of earlier design prototypes and their drawbacks [291] ©2017 IEEE.

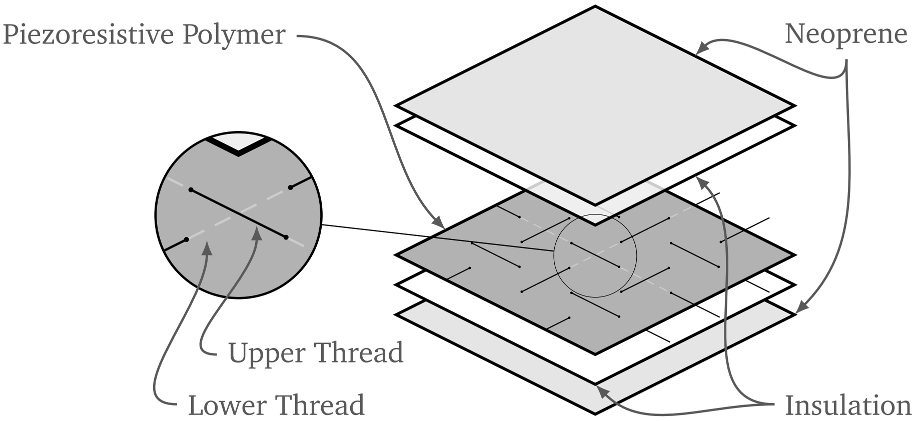


Figure 2.11: Assembly of the sensor prototypes *A* and *B* with neoprene cover. For prototype B, the adhesive insulation layer is not present and all layers are sewed together on the border [291] ©2017 IEEE.

2.3.2 Sensor Prototypes

Based on the selected materials, several prototypes for the overall sensor element assembly were built and evaluated. This section describes the three best designs concerning the accuracy, manufacturability, and low force thresholds. All other tested prototypes and their drawbacks are described shortly in Table 2.4.

Woven Matrix with Glued Neoprene Cover (A) The structural composition of this prototype is depicted in Figure 2.11 for a 3 × 3 taxels matrix. Stainless steel threads (3-ply) are woven into the polymeric foil to build a grid. Shimojo, Namiki, Ishikawa, Makino, and Mabuchi [271] previously used this technique for a silicone-rubber-based matrix. Each vertical and horizontal thread intersects at exactly one point, where one of the threads is on the upper side of the polymer and the other on the lower side. On the left side of Figure 2.11, one taxel on the matrix is magnified. The sewing process is carried out manually. The sensor element is wrapped in a neoprene cover to protect it from mechanical damage. The cover is glued to the inner layers using a silicone-based adhesive[3] to preserve its flexibility. Before the cover is glued, a thin and flexible adhesive insulating foil is attached to the polymer to protect the electrical contacts from the silicone adhesive, which may break the electrical contact between the polymeric foil and the stainless steel thread. This design allows a bending radius $r_A \approx 10\,\mathrm{mm}$. The radius is determined by clamping two ends of the sensor together and measuring the resulting radius in the middle of the patch.

[3] Dow Corning 3140 MIL-A-46146 RTV

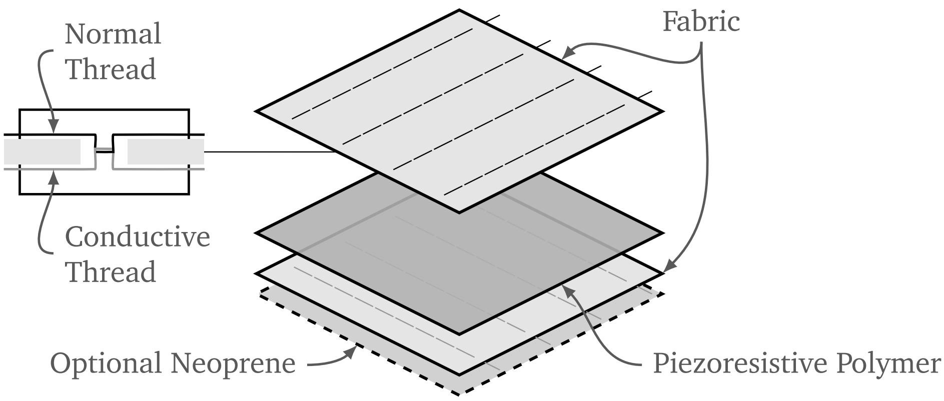


Figure 2.12: Assembly of the sensor prototype *C* with machine-sewed matrix and fabric cover [291] ©2017 IEEE.

Woven Matrix with Hand-Sewed Neoprene Cover (B) This prototype is similar to prototype *A*. Instead of gluing the neoprene cover, all layers are sewed together manually on the sensor patch's outer rim. Thus, there are no adhesive tangential forces between the single layers, even if the sensor is bent or deformed in any way. Without adhesive, the insulation layer is not necessary, and a lower bending radius $r_B \approx 6\,\text{mm}$ is reached.

Machine-Sewed Matrix with Fabric Cover (C) The assembly of prototype *C* is depicted in Figure 2.12 for a 4 × 4 taxels matrix. In this case, the conductive threads are machine-sewed into the protective cotton fabric using a lockstitch with a normal sewing thread and a stainless steel thread (2-ply). The non-conductive thread electrically insulates the polymer / steel thread from the environment; the conductive thread is accessible only from the inner sides of the fabrics, touching the polymer. All layers are machine-sewed together at the outer rim of the sensor patch. The result is a very flexible sensor prototype. Due to the machine-based sewing process, higher spatial resolutions and accuracies can be reached compared to prototypes *A* and *B*. Furthermore, this prototype can be cut to fit arbitrary shapes after manufacturing, as long as the conductive threads can be connected. The taxels' position can be designed arbitrarily (curves or other shapes) and even non-uniformly throughout the patch. The sewing process is easy and fast. Due to the higher robustness of the sensing layers, it is unnecessary to wrap the elements with neoprene. However, an additional neoprene layer may be sewed to the bottom of the sensor patch to achieve mechanical compliance. This prototype allows a bending radius of $r_C \approx 5\,\text{mm}$.

2.3.3 Experimental Results

https://youtu.be/HLdVkaF9ZR4/

The proposed concepts are evaluated on three different prototypes with a matrix size of 6 × 6 taxels and 7 mm taxel distance. Analyzed metrics are the accuracy of the pressure measurement and the threshold for the detection of contacts. Prototype *C* is measured with the additional neoprene layer at the bottom.

Calibration Model A calibration model is identified to reconstruct a corresponding pressure value from the resistance of a taxel. Due to the nature of the piezoresistive effect, the relation between resistance and pressure is highly nonlinear. The following equations were considered possible candidates for the sensor's calibration:

$$p = a\ \exp(bR) \tag{2.1}$$

$$p = a\ \exp(bR) + c\ \exp(dR) \tag{2.2}$$

$$p = aR^2 + bR + c, \tag{2.3}$$

with the sensed pressure p, the resistance R and the parameters a, b, c, d. Experimental results show the lowest fitting errors for model Equation (2.2). This model is used in two different ways to describe the sensor behavior:

Distinct Model The distinct model uses a separate parameter set for each taxel, which leads to the highest accuracy but requires taxel-wise calibration.

Universal Model The universal model uses only one parameter set for all taxels. As all taxels have slightly different properties, higher errors occur. However, this significantly reduces the calibration effort to the measurement of one taxel. Different units of the same sensor type may even use a single parameter set. In the following, the universal model is generated from a measurement on all taxels. However, the concept can similarly be applied by using measurements from a few taxels only.

Test Setup The test bench for the prototypes consists of a manual force test stand IMADA HV-500N II combined with a Sauter FL-20 force gauge, see Figure 2.13. The reference force sensor accuracy is ±0.05 N, with a resolution of 0.01 N. The flat circular test probe has a contact area of $A_p = 3.0\,\mathrm{cm}^2$. A custom electronics board with a voltage divider and several analog multiplexers converts the taxels' resistance to a voltage signal. The row and column are selected via the address lines of the used multiplexer modules. The output voltage reflects the resistance of the selected taxel. A scanning algorithm is implemented on a data acquisition system to read the complete matrix taxel-per-taxel with a sample time of 0.001 s per taxel. The update rate for data from the entire sensor is relatively low with $\approx 28\,\mathrm{Hz}$. However, this is a limitation of the test setup readout electronics and not the sensor itself.

Evaluation of the Sensor Patches First, a single taxel is loaded with a continuously increasing pressure until 33.33 kPa (10 N) are reached. Then, the force is decreased to zero. The corresponding taxel's resistance is measured during the whole process, which is repeated for each taxel on the matrix. Figure 2.13 shows the result for one taxel of prototype *A*. For low loads, the resistance of the taxel increases unexpectedly. This effect has been previously described in [287] and is caused by the different elasticity of the stainless steel thread and the piezoresistive polymeric foil. When pressure is applied on a taxel, the steel thread with a higher elastic modulus is deformed less than the polymeric foil (and neoprene). Consequently, the contacting steel threads lift off the foil, and the contact resistance increases. As a result, pressures below a specific pressure threshold (p_{th}) cannot be detected reliably. The pressure value at the first drop below the unloaded resistance value is defined as p_{th}, see Figure 2.13.

The original intention behind prototype *B* is less influence from tangential forces when the whole sensor is bent or placed on a curved surface. However, without the adhesive, the thread-lifting effect is also clearly reduced, and lower pressures can be detected. As horizontal movement between the layers is no longer constrained, the viscoelastic materials may deform horizontally, which possibly reduces the vertical deformation and, therefore, the lift-off of the steel threads.

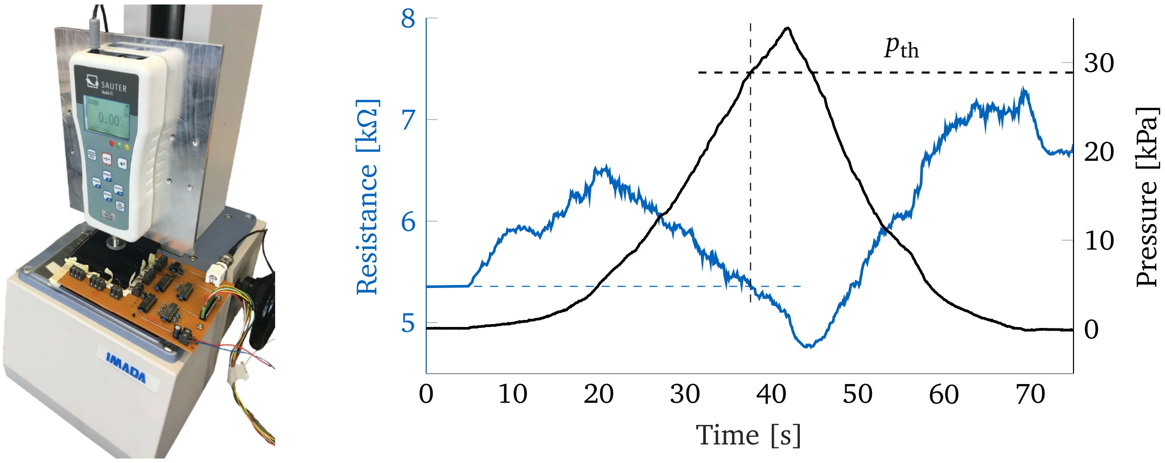


Figure 2.13: On the left: the used IMADA HV-500N II test stand with Sauter FL-20 force gauge, tactile skin, and multiplexer electronics board. On the right: Resistance and reference pressure over time for a taxel of prototype *A*.

	A	*B*	*C*
Pressure threshold max(p_{th}) [kPa]	27.9	3.9	0.7
Distinct model error median $e_{\text{med,dist}}$ [kPa]	0.5	0.5	0.3
Distinct model standard deviation $\sigma_{15,\text{dist}}$ [Ω]	552	344	286
Universal model error median $e_{\text{med,univ}}$ [kPa]	4.1	2.7	1.5

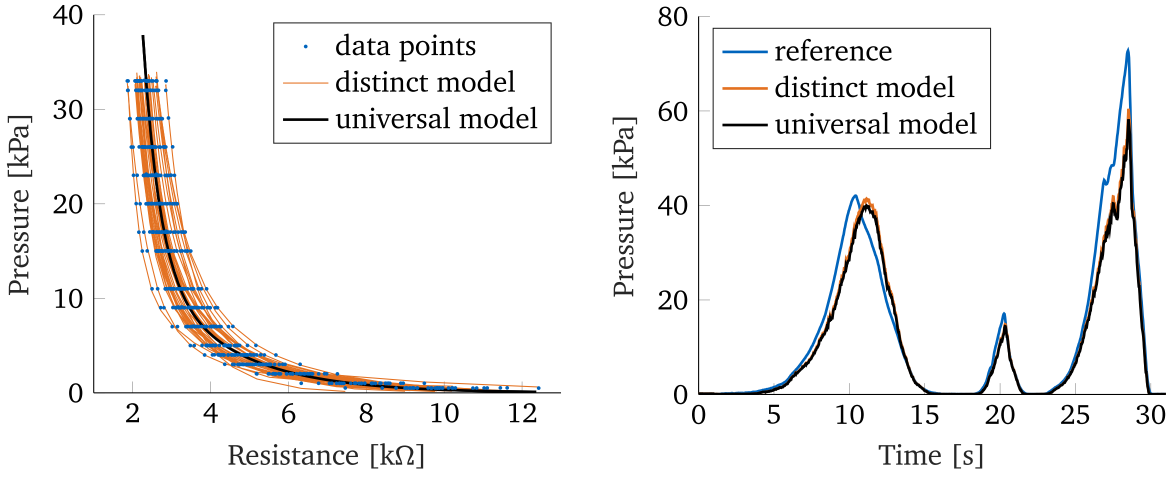
Table 2.5: Pressure thresholds and model errors for all prototypes.

For prototype *C*, even lower minimum pressures are reached. On the one hand, this may again result from the loose coupling between the layers. On the other hand, the steel thread is sewed into the fabrics with significantly higher thread tensions. Thus, the steel threads are additionally constrained in their movements, making the combined fabrics less inhomogeneous. Presumably, the elastic modulus of the combined fabrics (cotton & steel threads) better matches the polymer's elastic modulus, effectively reducing the lift-off effect. Therefore, prototype *C* is selected as the final prototype and is considered in the following experiments. The maximum pressure threshold values max(p_{th}) over all taxels of the prototypes are summarized in Table 2.5.

Calibration and Pressure Resolution Calibration curves are identified from the measurements described above to reconstruct pressure signals from each taxel's resistance. For each taxel, 17 data points (reference pressure and resistance) between 0.5 kPa and 33 kPa are selected from one data set (pressure increase over time). Data points with a resistance value above the unloaded resistance of the taxel are discarded. Via nonlinear least-squares, the parameters for each taxel's distinct calibration model are identified from this data.

The median of each parameter over all taxel-distinct models builds the universal model for all taxels. This procedure results in a fit, which may have higher errors for outliers than a new least-squares fit over all data points but has lower errors for most taxels (typical taxels). Moreover, a subset of all taxel measurements generally leads to the same universal model, which reduces the calibration effort.

Results on the calibration models of the prototypes are shown in Table 2.5. The analysis uses the absolute error between the distinct model and the data points for each taxel. The median of all these errors $e_{\text{med,dist}}$ is quite low for all three prototypes. The standard devi-



(a)

(b)

Figure 2.14: (a) Data points as well as distinct models and universal model of prototype *C*. For the measurement of the data points, the pressure is increased step by step. (b) Comparison of reference and sensed pressure for distinct and universal model of prototype *C*. Adapted from [291] ©2017 IEEE.

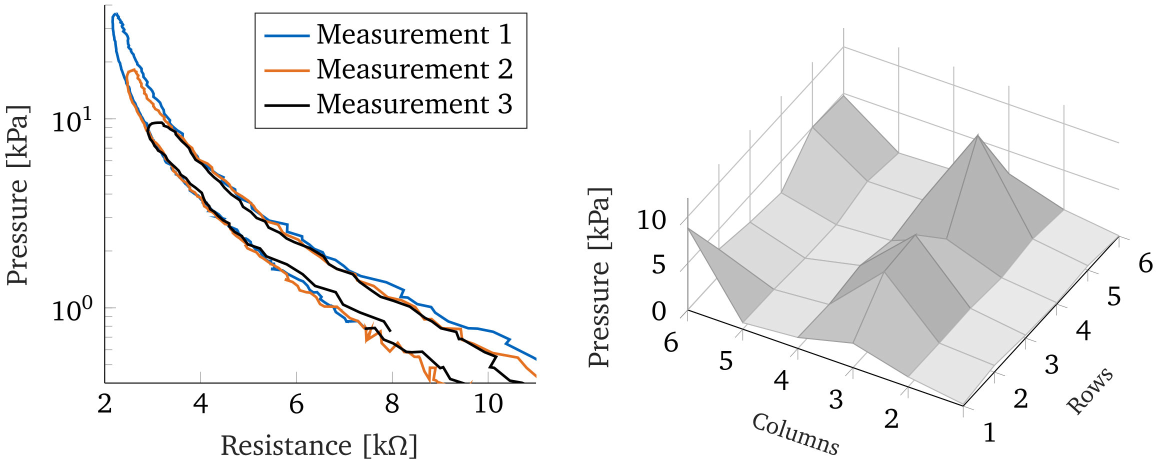
	Typical Taxel [kPa]		Worst Taxel [kPa]	
	max(e_{dist})	max(e_{univ})	max(e_{dist})	max(e_{univ})
0 - 5 kPa	0.9	1.0	1.4	1.6
5 - 10 kPa	2.2	2.4	3.7	4.7
10 - 20 kPa	4.4	4.8	7.9	11.8
20 - 40 kPa	9.9	10.9	14.4	24.8

Table 2.6: Maximum errors between sensor value and reference pressure for a typical and for the worst taxel of prototype *C*.

ation $\sigma_{15,\mathrm{dist}}$ of the resistance value at 15 kPa indicates the extent of differences between a prototype's taxels in the calibration curves. The distinct calibration models of prototype *A* differ more from each other than those of prototype *B*. The lowest value for $\sigma_{15,\mathrm{dist}}$ is reached with prototype *C*. This also reflects in the median model error for a universal parameter set, $e_{\mathrm{med,univ}}$, where prototype *C* is best.

The data points for every taxel, the distinct models, and the universal model over all taxels for prototype *C* are shown in Figure 2.14a. Due to the nonlinear response of the material, the universal model's error is best at low pressures with increasing (absolute) uncertainty for higher pressures.

Figure 2.14b shows the sensed pressure signal with distinct- and universal model for prototype *C* on a typical taxel, i.e., a taxel with a parameter set near the median of all taxels. Note that this is based on a measurement that is not part of the calibration data and contains different loading speeds and loads (0–74 kPa) significantly above the calibration range (0–33 kPa). Especially for low pressures, the sensor values represent meaningful information about the intensity of contact with the environment. Besides, there is no significant dependency on the loading speed, and the calibration model also yields reasonably accurate estimates outside the calibration range. The maximum sensor errors for a typical and for the worst taxel of prototype *C* are shown in Table 2.6. Using the universal model with just one fit function still provides low sensor errors. However, sensor errors can be higher for single outlier taxels.



(a)

(b)

Figure 2.15: (a) Hysteresis of prototype *C*. Note that the upper end of the hysteresis curve is rounded and not a sharp edge due to superimposed drift effects of the material for constant pressure [291] ©2017 IEEE. (b) Sensor output of prototype *C* for a four-point load.

Hysteresis and Drift Several load cycles at different maximum pressure were applied to one taxel of prototype *C* to analyze the piezoresistive material's hysteresis effects. The result of the measurements is shown in Figure 2.15a. There is a significant hysteresis effect, especially for higher pressures > 15 kPa. The sensor model may be extended to include different calibration curves for loading and unloading the taxels if required for the targeted purpose. For the applications described in this thesis, the effect is only of minor importance because the tactile sensor data is typically only used during foot loading.

Prototype *C* is loaded with a weight to ensure constant loads to analyze the drift over time. The median drift (of all taxels) of the sensed pressure is 0.26 kPa/s. The drift effect reduces the absolute accuracy for long-time static loads. However, the effect is equal for all taxels (standard deviation of 0.08 kPa/s), i.e., spatial localization of contacts is still accurate.

Spatial Resolution In general, the spatial resolution is defined by the number of taxels per area and the taxels' arrangement on the sensor surface. Depending on the skills of the worker who sews the steel threads into the fabric, high resolutions < 5 mm can be achieved. An example of a sensor output of prototype *C* for a four-point contact is shown in Figure 2.15b. A video of the sensor output for the tactile prototype *C* at different loads and contact scenarios is available at https://youtu.be/HLdVkaF9ZR4/. For some applications, interpolation methods with adequate models may be used to enhance the spatial resolution of the sensor output. Both the four-point load example in Figure 2.15b and the video show some crosstalk between the taxels. A further analysis of the effect and the resulting spatial resolution has been conducted by Hirschmann [118].

2.3.4 Related Work

Tactile sensing has been of increasing scientific interest in the last years. Numerous tactile sensor skins have been proposed [41, 262, 308]. Work has been done in the field of multi-modal and modular tactile skins [57, 199] and on the problems of electric wiring and scalability [14, 320]. However, these tactile sensors are stiff, and their application to arbitrarily shaped geometries is limited. Therefore, several 3D-shaped sensor elements have been designed [31, 164]. While these sensors can be manufactured to fit arbitrary shapes, they are not inherently flexible. A flexible sensor simplifies integration and allows adaptation to any form of external contact. A completely flexible, multi-modal skin based on a capacitive measurement principle is proposed in [120] — the design yields high accuracy and low cost. However, the spatial resolution may be limited. In [19, 32, 123] the taxels are also based on larger contact areas. While these sensors provide excellent results for specific applications, the theoretically reachable spatial resolution is limited.

Micro-mechanical taxels or a matrix structure of the connecting wires are used to reach higher spatial resolutions [49, 63, 158, 174, 234, 286]. Most of these designs are based on flexible Printed Circuit Boards (PCBs) or injection-molded elastomers. Either the structure is complex, or the manufacturing process targets high volumes. For low volumes and custom pieces, as needed in, e.g., research labs, manufacturing effort, and costs are usually high.

Some flexible sensor designs described in the literature are easy to reproduce. In [307], a robust and low-cost tactile skin is proposed, based on a capacitive measurement principle. In general, this design can be built quickly without the need for special machinery. However, it requires a dedicated circuit for each taxel, which complicates the overall design. Although the sensor could be made flexible, no flexible version has been published so far to the author's knowledge. [154] presents an approach based on a screen-printed piezoresistive ink and a flexible PCB. It is unclear how accurate the pressure measurements are and how easy the manufacturing process is. Another approach is based on electrical wires stitched into piezoresistive rubber [271]. The design is easy to build and flexible. There are, however, no details on the accuracy of the sensor patch given.

Recently, after the publication of the presented design in [291], Kim, Vu, and Kim [155] proposed a single layer piezoresistive tactile sensor based on woven fabric with a mixture of carbon-impregnated conductive yarn and standard wool yarn. The sensor is flexible, easy to build, and its design enables a high spatial resolution. The sensor is experimentally evaluated for pressures up to 1000 kPa. The sensitivity at low pressures has not been analyzed. In [173], a promising design based on a tactile matrix is proposed. It consists of carbon-black-filled silicone and conductive fabric. Manufacturing is easy given the mix of carbon-black and silicone is known. The sensor's accuracy is high; however, it is unclear if higher spatial resolutions can be reached easily. Another approach is presented in [269], based on conductive ink on a PET film. While this sensor is easy to manufacture, only discrete pressure values are detected. In [223], a highly scalable tactile skin is presented, based on photo-reflectors covered by urethane foam. The design is low-cost and relatively easy to build, but the photo reflector size limits the spatial resolution. Another approach based on a piezoresistive foil and a fabric-based conductive matrix is proposed in [59]. In contrast to the work presented, the sensor consists of more layers, and its spatial resolution is limited by the large conductive fabric rows and columns. Day, Penaloza, Santos, and Killpack [59] additionally proposed methods to mitigate crosstalk effects between taxels.

The proposed sensor concept is especially targeted to low-cost, flexibility, and a simple manufacturing process. Its primary application is detecting the contact points or the contact area and not the precise reconstruction of contact forces.

2.3.5 Discussion & Summary

In the evaluation, prototype *C* (Figure 2.12), which is made out of a cotton fabric and machine-sewed steel threads, showed the lowest threshold pressure and highest accuracy. Prototypes *A* and *B* (Figure 2.11) can be made by hand, but the manufacturing effort is high for large areas and accuracies are lower. As prototype *C* only requires a sewing machine and can be easily manufactured, it is selected as the final prototype. The sensor patch has a maximum activation threshold (over all taxels) of only 0.7 kPa, which is significantly below the values found in literature. The thresholds for comparable flexible sensors are 10–20 kPa in [286], 15 kPa in [234], 15.8–158 kPa in [63] as well as ≈ 3–4 kPa in [32].

The relationship between the sensor accuracy and the number of taxels was not evaluated; however, similar results for a higher number of taxels are expected. Furthermore, the influence of parameters such as the polymeric foil's thickness, the thread tension, and thickness was not analyzed. Even better designs may be reached with further experiments in this direction. Although larger contact areas are generally possible, the matrix design results in a high wiring effort for a high number of taxels compared to decentralized approaches, e.g., [199] (which do, however, not support high spatial resolutions).

The calibration model with distinct parameter sets leads to low errors in the sensed pressure as it considers local differences between the taxels. A universal calibration model generated from all distinct models leads to almost the same accuracy for a typical taxel on the sensor patch. The universal model may be general enough for a group of sensor patches without or with a very simple re-calibration per sensor. The detected hysteresis leads to errors for higher pressures. An extended inverse hysteresis model might improve this. Furthermore, the sensor values drift over time. As the drift is equal for all taxels, localization of the contact areas should not be affected. In general, the sensor is more suitable for measuring contact points and the contact area than for accurate contact forces.

With the described properties, this sensor fits general tactile sensing applications in robotics. Due to the low force threshold, it may be used for collision detection or even on a robotic hand's fingertips. The sensor is made of widely available and low-cost materials and can be manufactured with low effort. Due to the high flexibility, a bending radius of 5 mm is feasible. The material costs are $60-100$€/m², depending on the spatial resolution. Despite the low-cost design, pressure sensing on the taxels is accurate in the range of 0–40 kPa using a simple calibration model. Pressures of up to 70 kPa can be detected reliably. As the sensor patch consists of fabric with stainless steel threads, the spatial resolution can be customized, and the sensor may be cut to arbitrary shapes.

The manufacturing process for an exemplary sensor patch is shown in Appendix D. Further work dealing with data extraction from a tactile sensor of this type is presented in [118]. Moreover, a comprehensive model of the tactile sensor considering hysteresis, drift, nonlinearity, and taxel crosstalk is described in [250].

2.4 A Tactile Foot Design

Biped robots typically sense the ground reaction forces and use these measurements in balance control concepts. In some cases, joint torque sensors are used; in other cases, 6D FT sensors are attached to the end effectors — hands and arms — of the robot. Generally, only the resulting ground reaction forces are measured and not the pressure distribution of the contact. However, especially in partial ground contact situations, the knowledge of the precise contact area helps to improve humanoid robots' balancing capabilities. In the following, a foot design for LOLA with an integrated tactile sensor array is presented. Besides the integration of the sensor, proposed mechanical improvements for the traversal of uneven terrain are described. The design is a prototype proposal, which has not been integrated on LOLA yet.

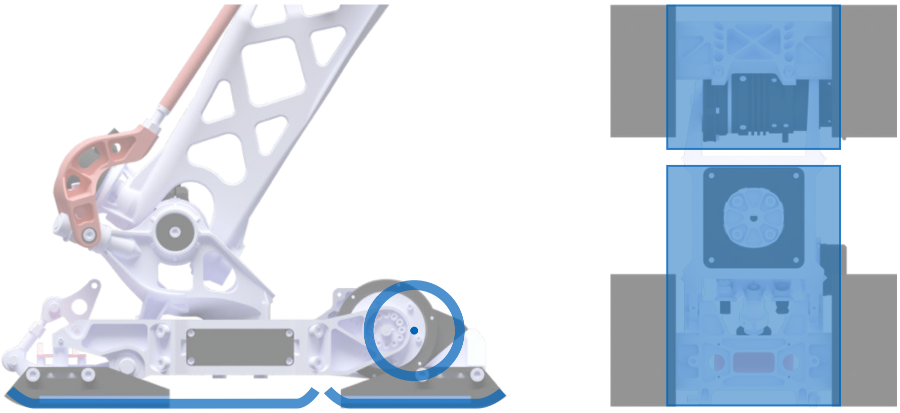


Figure 2.16: Pad-based foot design with contact switches and the coarse design (blue) for the new toe and heel element from side and bottom view.

2.4.1 Mechanical Concept

The original foot concept for LOLA consists of four footpads making contact with the environment, see Figure 2.2 — a design going back to the feet of its predecessor JOHNNIE [33]. This approach allows a lightweight design of the foot and reduces the leg's effective inertia, thus enabling faster walking. The pads are covered with a compliant, thin material to absorb shocks from impacts; the normal stiffness of one pad (with compliant material) is relatively high with approx. $80\,\mathrm{N/mm}$ [85] to enable contact force control at high bandwidth. In the presence of small and high objects, this design may lead to unwanted contact points between the four pads.

A schematic comparison of the original and proposed mechanical foot design is shown in Figure 2.16. The active toe joint is kept, and the foot is separated in a closed toe- and heel element. Alternative actuation possibilities for the toe joint, which reduce the leg's inertia, were investigated in [172]. The solutions are, however, very complex and require comprehensive changes to the robot's leg design. Thus, the toe actuator is kept in place, and only the axis of the toe joint is shifted forward to the middle of the toe element to provide equal torque reserves in both directions. When standing on the toe element only, it should not make a difference for the toe joint torque requirements wether the robot stands on the front or rear edge of the element. Furthermore, the width of the foot is reduced to make walking in narrower environments possible.

2.4.2 Sensor Concept and Contact Material

For the contact sensor array, the tactile sensor described in Section 2.3 is used. Hirschmann [118] successfully evaluated its suitability for the detection, parametrization, and classification of individual contacting objects. Furthermore, the convex BoS can be calculated from the tactile sensor data.

A separate sensor patch is used for the toe and heel area; both sensors feature a resolution of $1\,\mathrm{taxel/cm^2}$, resulting in 8 × 14 and 20 × 14 taxels per patch, see Appendix D. The conductive threads are connected to separate row and column multiplexers for each patch, which in turn route the sensor signals to a PCB with the Infineon XMC4300 microcontroller, see Figure 2.17. Both sensor patches are scanned taxel-by-taxel and simultaneously by the two Analog-to-digital converters (ADCs) of the microcontroller. The raw resistance values for all taxels are then sent to the control computer via EC.

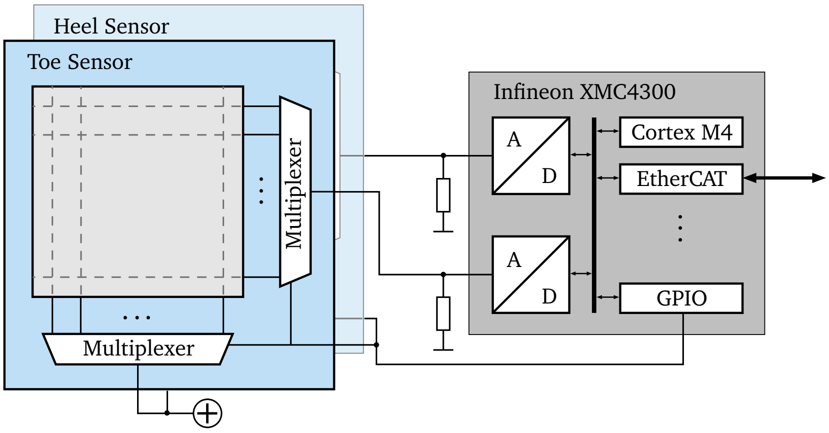


Figure 2.17: Proposed wiring diagram for the readout electronics of the tactile sensor arrays of toe and heel element.

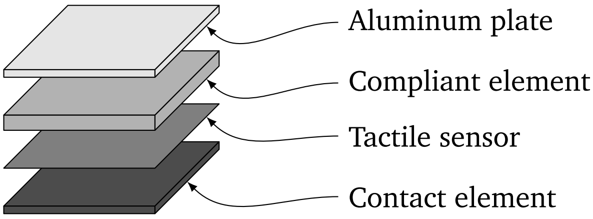


Figure 2.18: Proposed composition of the different foot sole layers.

On the one hand, the tactile sensor patches must be protected from wear and sharp objects on the ground. On the other hand, the sensor should be as close as possible to the ground to avoid the spatial distribution of ground pressure to a larger sensor patch surface. A structure consisting of four layers is proposed to resolve this conflict of objectives as far as possible, see Figure 2.18. The sensor is protected by a hard contact layer, while a thicker layer over the sensor ensures compliance. Only materials withstanding the high loads occurring from walking over uneven terrain are used. A detailed analysis of several mechanical designs for the foot sole layers is described in [285].

2.4.3 Design Proposal

A drawing of the foot sole prototype with two tactile sensor elements — one for the pad and one for the heel — is depicted in Figure 2.19. The design is inspired by the preliminary works developed in [285]. The compliant sole-layer is intended to be glued to the aluminum plate with an universal leather glue adhesive. The tactile sensor sits in a small pouch of the compliant material and is loosely clamped by the contact element (not shown) to avoid tangential stresses on the sensor fabric. The design uses an early form of multiplexer electronics boards at the foot's side, where the individual sensor wires are connected. The isolated row and column line wires are routed through small drill holes in the aluminum plate and connected to the multiplexer boards. In this early prototype, the clamping bars and multiplexer boards stand over the foot's support boundary. Future designs may integrate parts of the wiring in the sole layers. The foot also contains the regular FT sensor, which is more accurate in measuring the total ground reaction forces than the tactile skin. The electronics board with an interface to the EC bus is located at the heel.

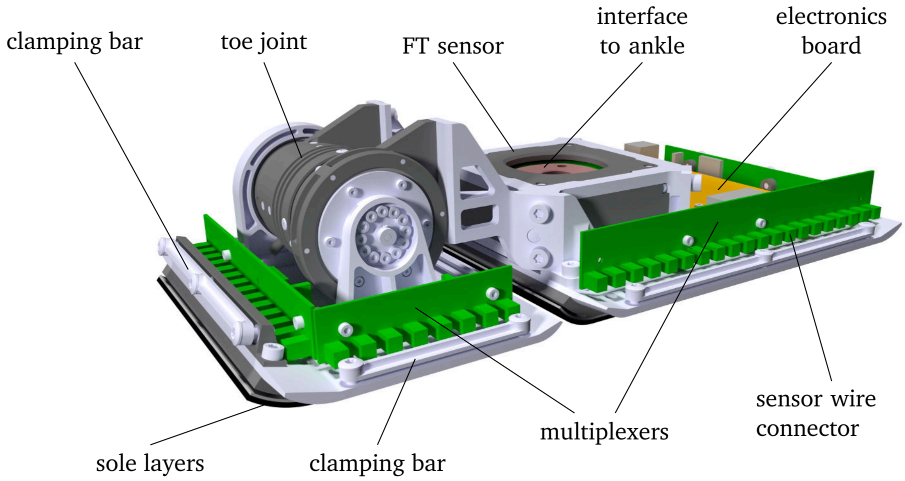


Figure 2.19: Computer Aided Design (CAD) drawing of the tactile foot sole prototype proposal. The sole is attached to the ankle via the interface at the FT sensor.

2.4.4 Related Work

A tactile foot sole on a humanoid robot was first proposed by Kinoshita, Kimura, and Shimojo [161] in 2003. The approach used a conductive rubber between row and column electrodes forming 14 × 24 taxels and was used to measure the actual CoP [162]. A similar design for the H7 humanoid based on a conductive rubber pad and a PCB board with 32 × 32 (3.3 taxel/cm^2) taxels is described in [298]. In [76], a flexible foot sole with integrated capacitive sensor patches is proposed to retrieve proximity data in different foot areas.

The presented approaches concentrate on designing the sensor foot sole and not on control schemes using the additional data. In many cases [161, 162, 298] only the calculation of the actual CoP from the pressure data is described. However, the used sensing techniques can generally not keep up with the precision and accuracy of FT sensors — either in the joints or the end effectors. The use of the contact area is not discussed.

In [252], the multi-modal tactile skin described in [199] is used on the foot sole of a humanoid robot. It is shown that actual CoP and the support BoS can be estimated from the measurements of the 42 taxels (0.15 taxel/cm^2) per foot. Furthermore, the contact area at the initial touchdown is used to decide if the contact is safe; when the area is below a fixed threshold, the footstep location is changed. The movements are executed quasi-statically.

Compared to related work, the proposed tactile foot design is specifically targeting the measurement of the contact area rather than performing a second measurement of the contact forces, which are anyway acquired from the FT sensor in each foot. The contact area information can, especially for dynamic walking on uneven terrain with unplanned partial contacts, significantly increase the contact force controllers' performance, see Chapter 5. The contact area data is primarily of interest in the first 100 ms after initial contact, which justifies the high readout sample rate and related engineering effort.

2.5 Chapter Summary

This chapter gives an overview on the hardware of the biped robot platform LOLA. Furthermore, a hardware control software based on the new concept of Bus Variables is proposed. Although the hardware-near software is designed for maximum flexibility, special emphasis is put on low latency and a high update rate of 4 kHz. All higher-level control modules described in the following chapters rely on the performance of the low-level hardware control, which makes it especially important for the operation of the whole system.

An easy-to-manufacture fabric tactile sensor design is proposed and analysed experimentally. It may generally be used for a wide range of applications where contacts and their locations must be traced on an arbitrary surface. This information can, for example, be used to avoid collisions with obstacles in real-time, see [261, 294].

To improve the performance of contact control methods described in this thesis, a foot design prototype with integrated tactile fabric is proposed. The concept aims to provide accurate information on the actual contact surface with the ground. Control schemes described in Chapter 5 are able to use this data to improve the robot's robustness to unexpected contact conditions.

Chapter 3

Real-Time Walking Control

The following chapter briefly discusses the dynamic properties of biped robots and consequent implications for their control. Furthermore, it gives an overview of LOLA's planning and control schemes. It sets the different control modules into context and draws the connection between the high-level and low-level control modules. Important frames of reference and the task-space definition are discussed. Besides, an overview of the WPG used throughout this thesis is given. In addition, this chapter describes strategies for the event-based adaptation of trajectories. It also presents a method to identify relevant control system plants using only onboard sensors and actors. The proposed control structure supports multi-contact scenarios, i.e., intended support with the feet and at least one hand.

3.1 The Dynamics of Biped Robots

A high number of degrees of freedom is characteristic for the dynamics of humanoid robots. Furthermore, these machines are not fixed to their environment but are standing on the ground with unilateral contact. The nature of bipedal walking requires to make and break contact with the environment in a rhythmic pattern, combining discrete events with continuous dynamics. The control of biped robots has been (and continues to be) a research topic for several decades because of these special properties of the involved dynamics.

To illustrate the implications on the control of such systems, let us write the equations of motions for a biped with n joint coordinates $\boldsymbol{q}_{\mathrm{J}} \in \mathbb{R}^n$, and the additional DoFs $\boldsymbol{q}_U \in \mathbb{R}^6$, which define the orientation and position of the upper body in the world:

$$\boldsymbol{M}(\boldsymbol{q})\,\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{\tau}_{\mathrm{J}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{J}_{\mathrm{FB,RF}}^T \\ \boldsymbol{J}_{\mathrm{J,RF}}^T \end{bmatrix} \boldsymbol{\lambda}_{\mathrm{RF}} + \begin{bmatrix} \boldsymbol{J}_{\mathrm{FB,LF}}^T \\ \boldsymbol{J}_{\mathrm{J,LF}}^T \end{bmatrix} \boldsymbol{\lambda}_{\mathrm{LF}} \tag{3.1}$$

with

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_U \\ \boldsymbol{q}_{\mathrm{J}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}_{\mathrm{FB}} + \boldsymbol{q}_{U,\mathrm{d}} \\ \boldsymbol{q}_{\mathrm{J}} \end{bmatrix} \quad \boldsymbol{M} = \begin{bmatrix} \boldsymbol{M}_{\mathrm{FB}} & \boldsymbol{M}_{\mathrm{FBJ}} \\ \boldsymbol{M}_{\mathrm{FBJ}} & \boldsymbol{M}_{\mathrm{J}} \end{bmatrix} \quad \boldsymbol{h} = \begin{bmatrix} \boldsymbol{h}_{\mathrm{FB}} \\ \boldsymbol{h}_{\mathrm{J}} \end{bmatrix}. \tag{3.2}$$

$\boldsymbol{M}$ denotes the mass matrix, and $\boldsymbol{h}$ contains the gyroscopic, centrifugal and gravitational forces on the robot. The right-hand side consists of the joint actuator torques $\boldsymbol{\tau}_{\mathrm{J}}$, and the contact wrenches (forces, torques) for the right ($\boldsymbol{\lambda}_{\mathrm{RF}}$) and left ($\boldsymbol{\lambda}_{\mathrm{LF}}$) foot, which are projected to the coordinates $\boldsymbol{q}$ via corresponding Jacobians. The floating-base DoFs $\boldsymbol{q}_{\mathrm{FB}}$ are defined to be the delta between the actual upper-body pose $\boldsymbol{q}_U$ and the desired pose $\boldsymbol{q}_{U,\mathrm{d}}$[1]. For simplicity, $\boldsymbol{q}_{U,\mathrm{d}} = \text{const.}$ is further assumed in this section.

[1] The precise definition of $\boldsymbol{q}_{\mathrm{FB}}$ and $\boldsymbol{q}_U$ is not required for the following considerations, but it is assumed that their choice does not lead to singularities in practice.

Equation (3.1) shows the *underactuation* of the system: Only $\ddot{\boldsymbol{q}}_\mathrm{J}$ can directly be influenced by the actuator torques $\boldsymbol{\tau}_\mathrm{J}$. This limitation is especially problematic because $\boldsymbol{q}_\mathrm{FB}$ contains DoFs with unstable (labile) dynamics. Even when sufficient friction in the ground plane is assumed, rotations around the edges of the feet in the sagittal and frontal plane are labile. When there is low friction, even more floating-base components require stabilization.

One indirect possibility for the floating-base stabilization is the induction of inertial force via the acceleration of joints. This becomes clearer when the floating-base dynamics are written for position-controlled joints. In this case, the control inputs are desired joint angles $\boldsymbol{q}_\mathrm{J,d}$, and it is assumed that the decentralized joint control achieves $\boldsymbol{q}_\mathrm{J} \approx \boldsymbol{q}_\mathrm{J,d}$:

$$\boldsymbol{M}_\mathrm{FB}\, \ddot{\boldsymbol{q}}_\mathrm{FB} + \boldsymbol{h}_\mathrm{FB} = \underbrace{-\boldsymbol{M}_\mathrm{FBJ}\, \ddot{\boldsymbol{q}}_\mathrm{J,d} + \boldsymbol{J}^T_\mathrm{FB,RF}\, \boldsymbol{\lambda}_\mathrm{RF}(\boldsymbol{q}_\mathrm{J,d}) + \boldsymbol{J}^T_\mathrm{FB,LF}\, \boldsymbol{\lambda}_\mathrm{LF}(\boldsymbol{q}_\mathrm{J,d})}_{\boldsymbol{\Lambda}_\mathrm{FB}} \tag{3.3}$$

The right hand side of Equation (3.3) describes the wrench $\boldsymbol{\Lambda}_\mathrm{FB}$ acting on the floating-base dynamics. The term $-\boldsymbol{M}_\mathrm{FBJ}\, \ddot{\boldsymbol{q}}_\mathrm{J,d}$ describes a linear dependency of $\boldsymbol{\Lambda}_\mathrm{FB}$ on the desired accelerations of the joints, which may be used to stabilize the robot's floating base.

A second and more powerful possibility concerning control input limitations is the control of the external contact forces, which act on the dynamics of the floating-base DoFs, see the last two terms of the right hand side of Equation (3.3). The corresponding Jacobians $\boldsymbol{J}_\mathrm{FB,RF}$, $\boldsymbol{J}_\mathrm{FB,LF}$ typically are full rank for surface contact (at least one foot must be on the ground). Furthermore, the robot's desired configuration $\boldsymbol{q}_\mathrm{J,d}$ influences the contact wrenches when the contact is compliant, which is valid for all real-world contacts. The contact wrenches are limited by the *unilateral contact* to the ground, which complicates the control of the floating-base dynamics.

The control of dynamic systems described by Equation (3.3) involves the generation of trajectories for the robot's joints, which consider the discrete nature of footsteps, are feasible with the unilateral contact constraints including friction, and stabilize the underactuated floating-base dynamics.

3.1.1 Feasibility of Motions

The unilateral contact between the ground and the feet is one of the essential constraints in biped walking. In combination with the unilateral contact restrictions, the discrete footholds lead to a changing BoS for the robot in the ground plane. A planned motion (floating base and joints) for a biped robot is considered feasible when it is dynamically consistent with the unilateral contact constraints. In this case, the underactuated floating-base DoFs experience zero acceleration when the joint motion is applied to an arbitrary (often reduced) dynamics model.

Feasibility does not guarantee the motion's stability because it does not consider a possibly disturbed state of the robot with nonzero floating-base angular momentum [240], nor does it necessarily consider an exact dynamical model of the robot. Feasibility is *required* for a biped's stability in the static case (no movement). However, feasibility is not required for a biped's stability in general because a motion can be stable even if it does not result in the planned floating-base motion (there are multiple valid solutions). Still, from a control perspective, it does not make sense to violate the feasibility criterion deliberately because this increases the uncertainty in the system due to a (higher) deviation of the floating-base DoFs from their expected trajectories.

The Zero Moment Point The ZMP was proposed over 50 years ago and is defined as that point on the ground *within* the BoS of the feet, at which the total moment of inertial and gravitational forces has no component along the horizontal axes [314]. Differently phrased, the existence of the ZMP (within the BoS) means that the forces corresponding to a dynamic motion of the biped are compatible with the unilateral contact constraint under the assumption of sufficient friction. Note that the direction of allowed unilateral forces is only implicitly considered because, by definition of the ZMP, only forces in one of the two directions are possible. When used as a synthesis tool, the direction of ground-reaction forces is defined a priori. When used for the analysis of motions, an additional check for the vertical force component is required.

Vukobratović and Borovac [314] term the system "dynamically balanced" when the ZMP exists. Furthermore, the ZMP always coincides with the CoP on the ground; the CoP is, however, not automatically the ZMP, because the latter may not exist. Vukobratović and Borovac [314] develop the ZMP equations from the dynamics of the foot only, with the result of an undesired foot rotation when the ZMP does not exist. This description is commonly used, e.g., for the FRI [88]. Still, this definition makes it hard to differentiate between undesired and intentional rotations of the foot. The meaning of the ZMP becomes clearer when the robot is considered as one system with all gravitational and inertial forces being written at one point (the CoM for example) and the ground-reaction forces described in the ground plane. With this description, it becomes clear that a non-existing ZMP does lead to a rotational acceleration of the whole robot around the edge of the foot. Because this rotation is not contained in the bodies' intentional (planned) motion, it shows up as a floating-base rotation in the model. Consequently, when the ZMP exists, the *floating-base* angular acceleration is zero (for the used dynamics model). This logic matches the definition of feasibility above and makes it a suitable criterion for synthesizing biped walking motions.

Strictly speaking, the ZMP is only defined for contacts in one ground plane. However, the concept may be ported to foot locations on different heights or uneven terrain [256].

A typical trajectory synthesis method for bipeds is the ZMP trajectory: the contact forces are defined by a specified ZMP trajectory, which automatically ensures compatibility with the unilateral contact. Usually, the ZMP trajectory is generated first, then a compatible CoM motion is calculated using the system's dynamics or a reduced-order model. In theory, the ZMP trajectory can be designed arbitrarily, and this method does not impose limitations beyond the constraints of the unilateral contact. The distances between the ZMP and the edges of the BoS describe margins of the contact forces to the infeasibility of the planned motion. The margins limit the foot torque reserves, which might be needed by a balance controller relative to the system's nominal motion. Thus, it is advisable to keep these margins high, although they are not (directly) related to the system's stability.

The ZMP criterion is fundamental and only related to the feasibility of planned motions. In particular, it does not enforce a zero rate of angular momentum on the total system dynamics nor does it consider friction limitations. Besides, measurements of human walking on a flat ground show a margin of the ZMP trajectory to the BoS [238], similar to the standard gait synthesis method for LOLA.

The Fictious Zero Moment Point / Foot Rotation Indicator To detect if the currently executed motion is compatible with the BoS, Goswami [88] proposed the Foot Rotation Indicator (FRI). Vukobratović and Borovac [314] term this point the *Fictious Zero Moment Point (FZMP)*. It is defined identically to the ZMP, except that it may leave the BoS. Compliance of a motion with the contact constraints can not be evaluated using the contact forces because the CoP can not leave the support and a location on the edge of the BoS is indifferent concerning feasibility. The FZMP needs to be calculated from the biped's motion to evaluate if the current motion is valid. If its location is within the BoS (including the edges), the unilateral constraint is met.

Friction Cone The description of the contact via the ZMP assumes sufficient friction in the horizontal ground plane. To lift this limitation, the additional constraint

$$||\boldsymbol{F}_t|| \leq \mu||\boldsymbol{F}_n|| \tag{3.4}$$

for the tangential $\boldsymbol{F}_t$ and normal $\boldsymbol{F}_n$ total ground reaction force must be considered. This limits the direction of the total ground reaction force to a friction cone at the ZMP/CoP described by the friction coefficient μ.

Because of the nonlinearity of the constraint Equation (3.4), the friction cone is often approximated using polyhedra, or more specifically regular pyramids [119, 166, 169]. This allows the description of valid forces via a set of linear basis vectors and makes related optimization problems convex [166, 169]. For multiple contacts in different planes, volumetric friction polyhedrons are used to define the feasible range of forces on the robot [2, 11].

Note that the consideration of friction coefficients in the generation of biped robot trajectories may cause drawbacks. It is generally difficult to know the exact mechanical properties of the environment beforehand. When the believed friction coefficients are lower than the actual ones, conservative (slow) movements result from the constraints, making the application of biped robots unappealing. Therefore, choosing relatively high values by default and treating unexpected slipping as disturbance seems to be a better alternative.

3.1.2 Stability of Motions

The generation of motions for biped robots, which lead to a dynamically stable gait, is a key problem in biped robotics. The motions must be compatible with the contact constraints and stabilize the unstable parts of the underactuated floating-base dynamics. They must match discrete footsteps constrained by the environment and kinematic capabilities of the robot. In the following, several approaches for trajectory generation and stability criteria are summarized. Note that some of the techniques may be combined to improve balancing in different scenarios.

Hybrid Zero Dynamics / Virtual Constraints Westervelt, Grizzle, and Koditschek [322] proposed an interesting framework for generating asymptotically stable motions for biped robots. The control concept is based on a set of holonomic constraints parametrized by the state of the robot. These position-based constraints are used to define the desired motions of the robot. They are called *virtual constraints* because they are enforced by controllers acting on the actuated DoFs of the robot. Instead of explicitly generating trajectories for the joints over time, the approach looks for parameters of the virtual constraints (between joints) leading to stable walking.

By choosing corresponding system outputs for the dynamics, which are zero when the constraints are fulfilled, control loops for the virtual constraints can be designed with low effort. Having the system outputs driven to zero via these controllers, only the (internal) zero dynamics of the robot determines the overall system's stability. The zero dynamics are the system's internal dynamics, which are not visible on the output (in the constraints). Because bipeds are a hybrid system with discrete and continuous states, it is called *hybrid zero dynamics* in this case. An offline optimization problem on the parameters of the virtual constraints is used to achieve stable zero dynamics. This concept has been successfully applied to a series of semi-passive bipeds [87, 229, 279]. The approach has, to the author's knowledge, only been applied to robots with small (point) feet. The application to different walking scenarios (speed, step parameters) requires the generation of a gait library with different virtual constraint parametrizations.

Zero Rate of Angular Momentum / Centroidal Moment Pivot Goswami and Kallem [89] introduce the concept of the Zero Rate of Angular Momentum (ZRAM) point as a stability criterion for biped robots. The ZRAM is the point on the ground surface where the total ground reaction force would have to act to reach a zero rate of change of the biped's total angular momentum [89]. In an independent investigation, Popovic, Goswami, and Herr [238] describe the same point as Centroidal Moment Pivot (CMP). When a biped falls, the distance between ZRAM/CMP and the CoP increases, effectively leading to a high angular momentum. However, the criterion of a zero rate of angular momentum is stringent for the generation of biped robot trajectories. A motion with a non-zero rate of angular momentum may still be stable, e.g., for a periodic and symmetric modulation. The difference to the ZMP is that the ZRAM enforces zero rate of the total angular momentum of the robot, while the ZMP only enforces that the directions of $\ddot{\boldsymbol{q}}_{\mathrm{FB}}$ corresponding to the sagittal and frontal inclinations are zero for a certain considered dynamics.

Reference Trajectory Modification A prevalent method for controlling biped robots is the generation of feasible — not necessarily stable — reference trajectories for the CoM and the contact forces. A subsequent stabilization module then uses sensor data to balance the robot by modifying the reference. The generation of reference trajectories is generally a complex problem with a high number of effects and parameters to consider. Using an optimization problem for the generation, many relations can be considered exactly [29, 169, 178]. However, this also requires the knowledge of often uncertain parameters, and leads to high computation times in the range of minutes or hours. Moreover, the high dimensionality of the search space may trap the optimizers in local minima, leading to often counterintuitive solutions. To achieve higher autonomy of the machine, the planner must be real-time capable, i.e., have a planning time lower than one footstep duration. These online planners typically use simplified models for the robot's dynamics; in most cases, the Linear Inverted Pendulum Model (LIPM) is used [50, 99, 136, 202]. In some cases, the model is extended to directly describe dynamic effects of the torso and leg motion [37, 241, 266, 301]. The CoM trajectory is usually obtained by solving the dynamic model for a predefined ZMP reference trajectory. This typically involves approximating the solution or modifying the ZMP, as the underlying problem is overdetermined [37, 136, 266].

The stabilization of the reference trajectories is done via feedback loops. There are three different basic mechanisms to balance a biped robot. Two of them are continuous and directly visible in Equation (3.3): (1) modifying the contact wrench, (2) accelerating the CoM. The third mechanism is discrete: (3) a modification of the BoS, i.e., modifying the next footstep location. While this is the most efficient approach, it is constrained by kinematic limits, obstacles, and the robot's stepping speed. The first two mechanisms are related since the acceleration of the CoM changes the contact forces and vice versa. The third one can be interpreted as a change in the feasible contact wrench. Most control approaches stabilize the upper-body orientation or position of the CoM via contact forces [38, 135, 166, 187, 201, 300]. For robots with torque-controlled joints, this usually requires control of the CoM position based on the robot's estimated state [106, 127, 166, 175]. In contrast, additional force control feedback based on FT sensor data is usually needed for robots with position-controlled joints to track desired ground reaction forces [38, 86, 201]. An acceleration of the CoM is typically applied when the contact forces are saturated [166, 303]. Step modifications based on heuristics are described by [303]. Further stabilization schemes based on a modification of the next footstep location are described in the following paragraphs.

Capturability Pratt, Carff, Drakunov, and Goswami [241] introduced the concept of the Capture Point (CP) — the point a robot needs to step to bring its orbital energy to zero.

When multiple CPs are possible — and kinematically reachable — the set of these points is called *Capture Region*. For the calculation of CPs, the robot is modeled as a linear inverted pendulum with additional rotational inertia in the sagittal plane. The ability to bring the robot's orbital energy to zero within N footsteps is called N-step *Capturability*. This approach is used as a stability criterion and control scheme for step recovery [203, 242].

The CP concept is equivalent to the 2D Divergent Component of Motion (DCM) described earlier by Takenaka [299] and Takenaka, Matsumoto, and Yoshiike [301]. The main idea behind this approach is the separation of the CoM dynamics in a stable (convergent) and unstable (divergent) component. A stable gait trajectory for the CoM is obtained by specifying the divergent component dynamics. The method has later been extended to 3D with arbitrary trajectories for the CoM height [67]. The DCM-based CoM planner framework described in [68] was successfully implemented on different robots traversing various terrain [166, 195]. The concept exploits the linearity of the floating-base CoM dynamics of the robot. However, it does not directly include dynamic effects caused by the acceleration of the legs and arms. Furthermore, a zero rate of angular momentum is assumed in the planning process, limiting the set of possible motions (see above).

Model Prediction Due to the nonlinear dynamics, Model Predictive Control (MPC) schemes are often used for the stabilization of biped robots. Nishiwaki and Kagami [214] and Tajima, Honda, and Suga [297] facilitate a high frequency replanning of the CoM trajectories based on the LIPM. Wieber [323] extends ZMP preview control [136] to linear model predictive control to generate optimal CoM trajectories online. Stephens and Atkeson [283] use linear MPC to calculate step modifications in case of disturbances on the robot. Urata et al. [311] combine fast replanning with an optimal preview controller for the LIPM to determine foot step locations and foot step timing online, achieving impressing disturbance rejection. The MPC concept is also used for step recovery with nonlinear reduced dynamic models [328]. With advances in computational power and by using parallelization techniques, MPC considering the full whole-body dynamics and constraints was successfully deployed to a biped [163] and quadruped [78]. Both implementations are based on Differential Dynamic Programming (DDP).

MPC schemes do not consider the effect of unforeseen mechanical properties of the environment or obstacles which were not detected by a vision system. Therefore, these concepts must be paired with performant local control schemes to increase disturbance rejection on uneven terrain.

3.2 Control Structure Overview

Figure 3.1 visualizes the hierarchical control structure of LOLA's walking controller. An operator sends high-level user commands via a network to the WPG. The operator interface allows the execution of pre-defined walking sequences or the control of the robot via joystick. Using a reduced dynamic model of the robot, the WPG module calculates an ideal walking pattern with trajectories $\boldsymbol{X}_{\text{id}}, \boldsymbol{V}_{\text{id}}$ for the center of mass, all four end effectors, and the corresponding ideal ground reaction wrench $\Lambda_{\text{id}}^{\text{CoM}} = [\boldsymbol{F}_{\text{id}}^T, \boldsymbol{T}_{\text{id}}^T]^T$ at the CoM[2]. In addition, the information on the desired contact states γ_{id} (open/closed in a continuous fashion) is generated by the WPG. The trajectories are ideal in the sense of being calculated from a reduced dynamic model and not considering a possibly disturbed state of the robot. Vision-based data on the environment is the only sensory information used by the WPG module.

[2] Due to the task-space definition and the planner's internals, the CoM is a natural choice.

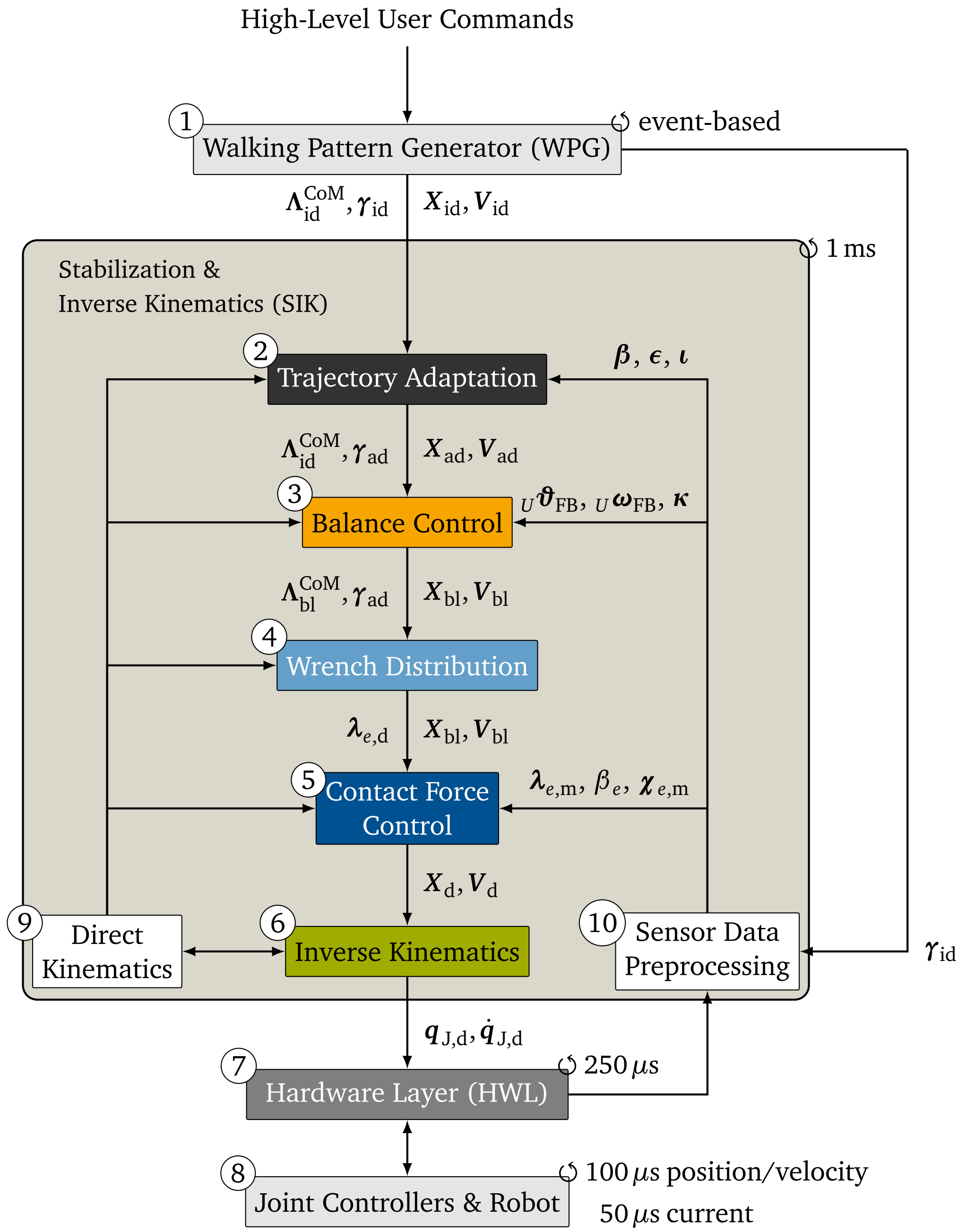

High-Level User Commands
1
Walking Pattern Generator (WPG)
event-based
$\Lambda_{\mathrm{id}}^{\mathrm{CoM}}, \gamma_{\mathrm{id}}$
$X_{\mathrm{id}}, V_{\mathrm{id}}$
1 ms
Stabilization & Inverse Kinematics (SIK)
2
Trajectory Adaptation
β, ϵ, ι
$\Lambda_{\mathrm{id}}^{\mathrm{CoM}}, \gamma_{\mathrm{ad}}$
$X_{\mathrm{ad}}, V_{\mathrm{ad}}$
3
Balance Control
${}_{U}\vartheta_{\mathrm{FB}}, {}_{U}\omega_{\mathrm{FB}}, \kappa$
$\Lambda_{\mathrm{bl}}^{\mathrm{CoM}}, \gamma_{\mathrm{ad}}$
$X_{\mathrm{bl}}, V_{\mathrm{bl}}$
4
Wrench Distribution
$\lambda_{e,\mathrm{d}}$
$X_{\mathrm{bl}}, V_{\mathrm{bl}}$
5
Contact Force Control
$\lambda_{e,\mathrm{m}}, \beta_{e}, \chi_{e,\mathrm{m}}$
$X_{\mathrm{d}}, V_{\mathrm{d}}$
9
Direct Kinematics
6
Inverse Kinematics
10
Sensor Data Preprocessing
γ_{id}
$q_{\mathrm{J,d}}, \dot{q}_{\mathrm{J,d}}$
7
Hardware Layer (HWL)
250 µs
8
Joint Controllers & Robot
100 µs position/velocity
50 µs current

Figure 3.1: Overview of the hierarchical walking controller with signal flow and simplified control loops.

The generated plan contains several footsteps and always ends with a safe state for the robot. The (re-)planning of the walking pattern is triggered event-based, e.g., by a new target position or commanded velocity; however, the currently executed step is not changed to ensure smooth trajectories.

All planned task-space trajectories are evaluated and sent to the SIK module every $\Delta t_{\mathrm{cont}} = 500\,\mu s$ via a shared memory interface. The primary purpose of SIK is the compensation of inaccuracies in the model and the environment and the mitigation of a caused disturbed state by modifying the ideal task-space trajectories from the WPG module.

The reactive trajectory adaptation module is designed for the mitigation of discrete events, such as unexpected impacts on the environment. It uses contact event data of early contacts $\boldsymbol{\epsilon}$, and late contacts $\boldsymbol{\iota}$. Furthermore, the currently executed control mode (force or position), stored in β_e for every end effector e, is utilized. A subsequent balance control module modifies the desired contact wrench and the CoM trajectories to reject disturbances on the measured floating-base inclination ${}_U\boldsymbol{\vartheta}_{\mathrm{FB}}$, and angular velocity ${}_U\boldsymbol{\omega}_{\mathrm{FB}}$ of the robot. A vertical CoM acceleration module additionally uses the actual contact factors $\boldsymbol{\kappa}$.

The resulting total desired wrench $\boldsymbol{\Lambda}_{\mathrm{bl}}^{\mathrm{CoM}}$ is distributed to specific end-effector wrenches $\boldsymbol{\lambda}_{e,\mathrm{d}}$ for each end effector e. The resulting desired wrenches for each end effector are then fed into corresponding contact force controller instances. These modules track the desired contact wrenches based on the measured forces $\boldsymbol{\lambda}_{\mathrm{m}}$ and contact state $\boldsymbol{\chi}_{e,\mathrm{m}}$ by modification of the task-space trajectories. These final trajectories $\boldsymbol{X}_{\mathrm{d}}, \boldsymbol{V}_{\mathrm{d}}$ are passed to a velocity-level Inverse Kinematics (IK) before being sent to the hardware control software via a shared memory interface.

This work deals with continuous and local control schemes. The inclination ${}_U\boldsymbol{\vartheta}_{\mathrm{FB}}$ and inclination rate ${}_U\boldsymbol{\omega}_{\mathrm{FB}}$ of the floating base are used as quality factors to quantify the performance of different control schemes regarding the stability of the system. The goal of all described stabilization schemes is the minimization of these deviations. The floating base inclination is measured via the IMU using the torso's known inclination relative to the base, see Section 3.6.1.

3.3 Previous Work on LOLA

This section briefly summarizes relevant previous work on LOLA in the context of this chapter. More detailed comparisons are part of the corresponding chapters dealing with individual control submodules. The hierarchical control structure for LOLA has been proposed in [33] and is adopted for this work. The original planner used a collocation approach with cubic splines to generate CoM trajectories for the robot [37]. A subsequent inertial stabilization and force control method modified the planned trajectories in the presence of disturbances [38]. The wrench distribution to the two feet used a heuristic, which does not consider additional hand contacts with the environment. An angular momentum compensation in nullspace (mainly using the arms) was proposed in [264].

Later work on LOLA deals with the generation and optimization of stepping motions over previously unseen but detected obstacles [109, 113] and the navigation in the presence of obstacles [112]. Furthermore, a vertical torso trajectory generation for LOLA based on an optimization problem was investigated in [110]. Due to the integration of a simplified kinematic robot model, the approach enabled the successful execution of dynamic stepping motions near the kinematic limits of LOLA. In terms of disturbance rejection, a reduced nonlinear model was used for step recovery in a model-predictive-control fashion [327, 328]. The corresponding control module was located between the planner and stabilization module, with an update rate of 50 Hz. These global approaches are not used in the scope of this thesis.

Frames of Reference and Task-Space Definition Based on the experience with JOHNNIE, a jumping reference frame was used to describe task-space trajectories [33]. The frame was inertial on the current stance leg position, which simplifies calculating an ideal walking pattern. This approach required additional variables for the biped's location in the world, especially for vision-related implementations.

The end-effector trajectories were defined relative to the CoM position, which was defined in the jumping reference frame. The orientation of the torso and the feet were described in a custom angle representation proposed by LÖFFLER [33]. While these representations allow the independent description of two of the three rotations, their use makes a purposive modification of foot orientations complicated.

Note that the CoM position was part of the task-space for the used IK solution. Furthermore, the arm joints were entirely in the nullspace, determined only by secondary objectives, e.g., angular momentum compensation and the CoM location.

Event-Based Walking Control Event-based walking is motivated by how many animals and humans adapt their walking patterns once contact with the environment occurs. The WPG of LOLA was previously enhanced to enable event-based transitions from single-support to double-support phase [35]. These transitions are necessary when the robot steps on an undetected obstacle — leading to an early contact of the swing foot. The findings indicate that keeping the original step timing is beneficial because this extends the practical double-support phase in the presence of such disturbances. The transitions were implemented via an additional impact state, in which the trajectory of the swing foot was replanned to accomplish a premature halt of the swing foot.

However, practical experience with this algorithm revealed several drawbacks: (1) The algorithm is implemented in the WPG — and not in the generally faster SIK module. This leads to additional delays until the changed trajectory becomes effective. Furthermore, a parametrized implementation in the form of a curve makes the implementation more complicated than necessary. (2) After the early-contact impact on the obstacle, the swing foot moves too far away from the new ground, effectively leading to a subsequent late-contact situation. Thus, a different method for early-contact situations (which still keeps the original step timing) is proposed in this chapter.

3.4 Frames of Reference and Task-Space Definition

In general, there are many possible valid choices for the components used in the task-space definition and their respective Frames of Reference (FoRs). Compared to previous work on LOLA, the end effectors' motions and the CoM are described relative to a fixed (non-jumping) FoR. This simplifies calculations in the SIK module and gives a more intuitive description of the motions. In contrast to previous work on LOLA, double-precision floating-point arithmetic is used. Thus, the numeric error caused by an accumulation of large values in the task-space entries is no (longer) significant relative to the machine's total accuracy. Furthermore, the location of the reference FoR can be reset from time to time to avoid numerical inaccuracies. Internally, the WPG still uses a jumping reference frame, which significantly simplifies the planning effort for feasible trajectories.

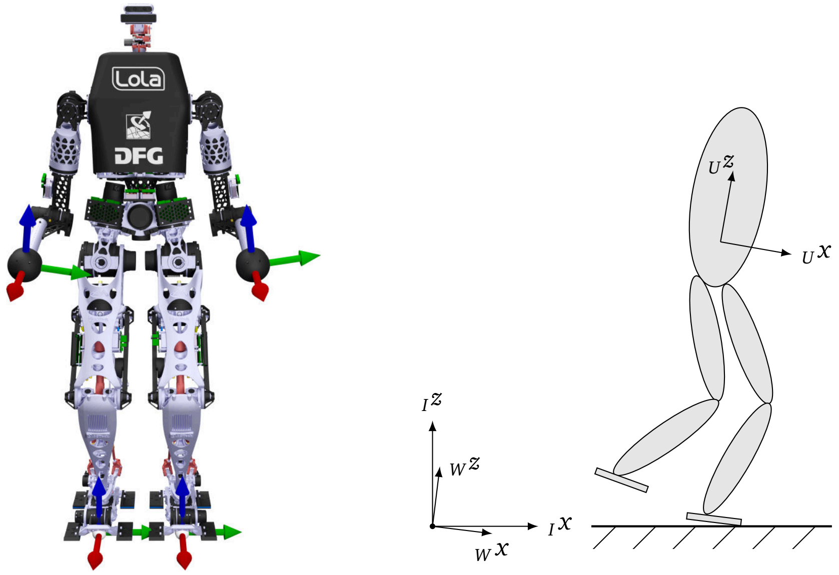


Figure 3.2: On the left: the end-effector FoRs for LOLA with visualization of x- (red), y- (green), and z-axis (blue). On the right: Visualization of inertial FoR I, the ideal-world FoR W, and the upper-body FoR U.

3.4.1 Frames of Reference

The planned trajectories from the WPG are described in the world FoR W. This frame is inertial in the ideal world of the WPG; its z-axis points upwards. The origin of the FoR is located on the ground level; the exact location, however, is not essential. The planner has no notion of a possibly disturbed state of the robot and the planned trajectories are always expressed in W. However, the actual robot state may be disturbed, and W is thus not a real inertial FoR. For example, when the robot tilts, W rotates with respect to the real-world inertial frame I. The definition of I allows SIK to differentiate between trajectories for the ideal world and the robot's actual orientation. The transformation between I and W equals the floating-base DoFs introduced in Section 3.1.

Another important FoR, the upper body frame U, is fixed to the torso body of LOLA. Its z-axis faces upwards when the upper body is vertical, its x-axis is directed forwards. The frame is attached to the torso body's origin; only its orientation is relevant. A visualization of the FoRs is depicted in Figure 3.2. The additional intermediate FoR I' differs from I in a rotation around its z-axis; the I'_x-axis and the projection of the U_x-axis onto the I'_x, I'_y plane are parallel. This FoR is required for the representation of yaw-free orientations measured by the IMU, see Section 3.6.1.

Each end effector (feet, hands) has a body-attached frame $E = \{\mathrm{RF}, \mathrm{LF}, \mathrm{LH}, \mathrm{RH}\}$, see Figure 3.2. The frames are located at the respective Tool Center Point (TCP) of the end effector. For the feet, this is the geometric center of the toe contact surface on ground-level; the z-axis is orthogonal to the contact surface and faces upwards, the x-axis points in forward walking direction. For the hands, the TCP is located in the geometric middle of the contact sphere.

3.4.2 Task-Space Definition

The components used for the task-space definition of LOLA are summarized in Table 3.1. The rotation vectors are a form of angle-axis representation and can be transformed from and to quaternion descriptions, see Appendix A. Using this parametrization instead of quater-

Symbol	Description
${}_{W}\boldsymbol{r}_{W,\mathrm{CoM}}$	Position (x, y, z) of the CoM relative to the world FoR in the world FoR
${}_{W}\boldsymbol{\vartheta}_{W,U}$	Rotation vector from upper body FoR to world FoR
${}_{W}\boldsymbol{r}_{W,\mathrm{RF}}$	Position (x, y, z) of the right foot TCP relative to the world FoR in the world FoR
${}_{W}\boldsymbol{\vartheta}_{W,\mathrm{RF}}$	Rotation vector from right foot end effector FoR to the world FoR
φ_{RT}	Right toe angle
${}_{W}\boldsymbol{r}_{W,\mathrm{LF}}$	Position (x, y, z) of the left foot TCP relative to the world FoR in the world FoR
${}_{W}\boldsymbol{\vartheta}_{W,\mathrm{LF}}$	Rotation vector from left foot end effector FoR to the world FoR
φ_{LT}	Left toe angle
φ_{P}	Head pan angle
φ_{T}	Head tilt angle
${}_{W}\boldsymbol{r}_{W,\mathrm{RH}}$	Position (x, y, z) of the right hand TCP relative to the world FoR in the world FoR
${}_{W}\boldsymbol{r}_{W,\mathrm{LH}}$	Position (x, y, z) of the left hand TCP relative to the world FoR in the world FoR

Table 3.1: Components of the task-space definition of LOLA.

nion coefficients in the task-space simplifies the modification of orientations in the control approaches and reduces the task-space size.

Using the described components, the following three task-space vectors are defined. The position of the base — that is the upper body and the legs — is described by

$$ {}_{W}\boldsymbol{x}_{\mathrm{B}}^{T} := \begin{bmatrix} {}_{W}\boldsymbol{r}_{W,\mathrm{CoM}}^{T} & {}_{W}\boldsymbol{\vartheta}_{W,U}^{T} & {}_{W}\boldsymbol{r}_{W,\mathrm{RF}}^{T} & {}_{W}\boldsymbol{\vartheta}_{W,\mathrm{RF}}^{T} & \varphi_{\mathrm{RT}} & {}_{W}\boldsymbol{r}_{W,\mathrm{LF}}^{T} & {}_{W}\boldsymbol{\vartheta}_{W,\mathrm{LF}}^{T} & \varphi_{\mathrm{LT}} & \varphi_{\mathrm{P}} & \varphi_{\mathrm{T}} \end{bmatrix}, \tag{3.5} $$

with the corresponding absolute velocity using angular velocities and not $\dot{\boldsymbol{\vartheta}} \neq \boldsymbol{\omega}$:

$$ {}_{W}\boldsymbol{v}_{\mathrm{B}}^{T} := \begin{bmatrix} {}_{W}\dot{\boldsymbol{r}}_{W,\mathrm{CoM}}^{T} & {}_{W}\boldsymbol{\omega}_{W,U}^{T} & {}_{W}\dot{\boldsymbol{r}}_{W,\mathrm{RF}}^{T} & {}_{W}\boldsymbol{\omega}_{W,\mathrm{RF}}^{T} & \dot{\varphi}_{\mathrm{RT}} & {}_{W}\dot{\boldsymbol{r}}_{W,\mathrm{LF}}^{T} & {}_{W}\boldsymbol{\omega}_{W,\mathrm{LF}}^{T} & \dot{\varphi}_{\mathrm{LT}} & \dot{\varphi}_{\mathrm{P}} & \dot{\varphi}_{\mathrm{T}} \end{bmatrix}. \tag{3.6} $$

Additionally, separate vectors for the two hands $h = \{\mathrm{LH},\mathrm{RH}\}$ are defined:

$$ {}_{W}\boldsymbol{x}_{h} := {}_{W}\boldsymbol{r}_{W,h}, \qquad {}_{W}\boldsymbol{v}_{h} := {}_{W}\dot{\boldsymbol{r}}_{W,h}, \tag{3.7} $$

$$ {}_{E}\boldsymbol{x}_{h} := {}_{E}\boldsymbol{r}_{W,h}, \qquad {}_{E}\boldsymbol{v}_{h} := {}_{E}\overset{\circ}{\boldsymbol{r}}_{W,h} \,. \tag{3.8} $$

Via the selection factors ξ, the task priorities in the IK can be changed online. A value of $\xi = 1$ denotes the position of a hand can be specified in task-space, while $\xi = 0$ makes the arms available for secondary objectives, see Section 3.9. Depending on the task-space choice, the number of redundant DoFs changes, see Table 3.2. The complete task-space vector is described by

$$ \boldsymbol{X}^{T} := \begin{bmatrix} {}_{W}\boldsymbol{x}_{\mathrm{B}}^{T} & {}_{W}\boldsymbol{x}_{\mathrm{RH}}^{T} & {}_{W}\boldsymbol{x}_{\mathrm{LH}}^{T} \end{bmatrix} \in \mathbb{R}^{28}. \tag{3.9} $$

Task-Space Choice	Task-Space Size	Redundant DoFs
Base	22	10
One Arm & Base	25	7
Both Arms & Base	28	4

Table 3.2: The task-space size and redundancy for the different selectable task-space choices. LOLA has 26 mechanical DoFs plus 6 unactuated DoFs for the floating base.

The position and orientation of the feet are key for the stabilization and control of the robot. For convenience, the 6D pose of a foot $f = \{\mathrm{LF}, \mathrm{RF}\}$ is additionally described with

$$ {}_W\boldsymbol{x}_f^T := \begin{bmatrix} {}_W\boldsymbol{r}_{W,f}^T & {}_W\boldsymbol{\vartheta}_{W,f}^T \end{bmatrix}, \qquad {}_W\boldsymbol{v}_f^T := \begin{bmatrix} {}_W\dot{\boldsymbol{r}}_{W,f}^T & {}_W\boldsymbol{\omega}_{W,f}^T \end{bmatrix}, \tag{3.10} $$

$$ {}_E\boldsymbol{x}_f^T := \begin{bmatrix} {}_E\boldsymbol{r}_{W,f}^T & {}_E\boldsymbol{\vartheta}_{W,f}^T \end{bmatrix}, \qquad {}_E\boldsymbol{v}_f^T := \begin{bmatrix} {}_E\overset{\circ}{\boldsymbol{r}}{}_{W,f}^T & {}_E\boldsymbol{\omega}_{W,f}^T \end{bmatrix}. \tag{3.11} $$

3.4.3 Task-Space Trajectory Modification

Reference trajectories from the planner need to be modified to stabilize the robot in uneven terrain. For example, several control modules modify the pose of the feet relative to an input task-space vector, see Figure 3.1. In this section, several operators and helper algorithms are defined, which are used for the purpose of modifying a reference trajectory online.

Different task-space vectors must not be added directly due to the description of foot orientations via the coupled 3D rotation vector. Thus, the operators $\mathrm{modify}_{\mathrm{RF}}$, $\mathrm{modify}_{\mathrm{LF}}$ are introduced to define the superposition of a task-space trajectory $\boldsymbol{X}, \boldsymbol{V}$ with a differential foot trajectory $\Delta\boldsymbol{x}_f, \Delta\boldsymbol{v}_f$, see Appendix A.4. The alias ${}_W\,\mathrm{modify}_f$, ${}_E\,\mathrm{modify}_f$ are used to describe the operators for modifying the trajectory of a foot f in world or end-effector FoR. Modifiers for the hand positions of Lola are defined analogously to $\mathrm{modify}_{\mathrm{RH}}$, $\mathrm{modify}_{\mathrm{LH}}$, with the alias ${}_W\,\mathrm{modify}_h$, ${}_E\,\mathrm{modify}_h$. Algorithm 1 describes a cascade to modify all specified end effectors using a specified expression for the modifications.

```
Function $(\boldsymbol{X}', \boldsymbol{V}') = {}_{\mathrm{FoR}}\,\mathrm{modifyAll}_{\mathtt{Set}}\,(\boldsymbol{X},\ \boldsymbol{V},\ \mathsf{expression})$
    $(\boldsymbol{X}', \boldsymbol{V}') \leftarrow (\boldsymbol{X}, \boldsymbol{V})$;
    for all i in Set do
        $(\boldsymbol{X}', \boldsymbol{V}') \leftarrow {}_{\mathrm{FoR}}\,\mathrm{modify}_i(\boldsymbol{X}', \boldsymbol{V}', \mathsf{expression}_i)$;
    end
end
```

Algorithm 1: Applies the modify operator to a set of end effectors by chaining the calls to the operator.

Time-Integration Helpers Instantaneous velocities need to be integrated to translational or rotational delta modifications before being added to a reference trajectory. By default, the discrete-time integrator int() described in Appendix A.3 is used for integration from $t_0 = 0\,\mathrm{s}$ to current time $t = n\,\Delta t_{\mathrm{cont}}$. In some cases, the modifying (absolute) velocities are described in moving FoRs, for example the end-effector FoR. However, when the requested velocities are zero, the modifications stored in the integral should be constant in an inertial FoR, not a moving one. Thus, the velocities are transformed to an inertial FoR before integration. For this purpose, the special integrator

$$ \begin{aligned} \mathrm{intA} \colon \mathbb{R}^k \times \mathbb{R}^{k\times k} &\to \mathbb{R}^k \\ (\boldsymbol{z}, \boldsymbol{A}) &\mapsto \boldsymbol{A}^T[n]\,\mathrm{int}(\boldsymbol{A}[n]\,\boldsymbol{z}[n]) \end{aligned} \tag{3.12} $$

takes a time-dependent transformation matrix $\boldsymbol{A}$ for this purpose. The result of the integrator is, however, always returned in the FoR of the input vector $\boldsymbol{z}$. The transformation matrix usually is based on the reference trajectory the module wants to modify. Note that this special measure is not a question of mathematical correctness but a question of modification policy. Either the modification is constant in the moving FoR or in the inertial FoR for zero velocity.

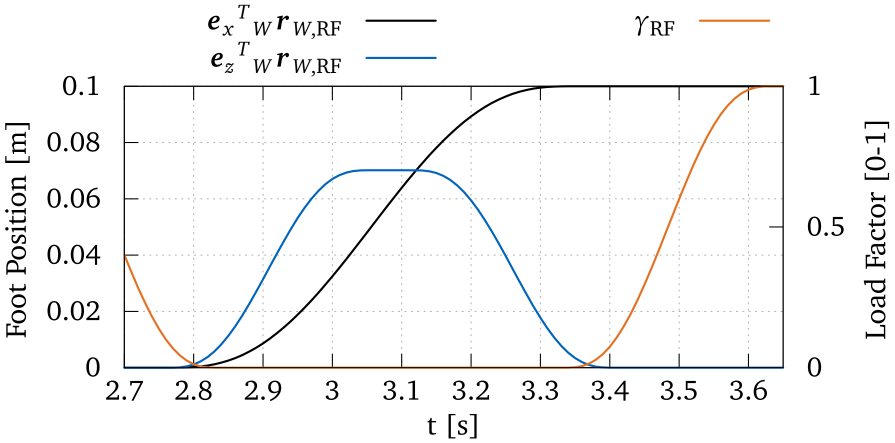


Figure 3.3: Typical foot trajectory for one step of the right foot on flat ground, with a step duration of 0.8 s.

3.5 Walking Pattern Generation

This section gives an overview of the WPG (1 in Figure 3.1) used for experiments in this thesis. The WPG's design is outside of this thesis's scope[3], and details on the implementation are consequently only given where necessary to understand the overall system. For an in-depth explanation of the planner, refer to [33, 266]. The WPG is currently being extended to generate feasible trajectories for multi-contact scenarios, i.e., when additional hand contacts are required. In the following, only the sequence of stages for the generation of standard walking patterns is described.

Stage 1: Foot Step Locations, Foot Trajectories, and Load Factors In the first stage, the footstep locations are determined. Based on a user-commanded target location and a map of the environment — typically acquired by a vision system — footstep locations may be autonomously planned [112]. In the context of this thesis, however, only fixed sequences of footsteps with predefined timing and step lengths[4] are used, see Figure 3.3. Corresponding collision-free foot trajectories are planned via a heuristic using fifth-order polynomials for parametrization [266]. Toe joint rotation is used to increase the margins to the kinematic limits for large steps. In addition, the load factors $\boldsymbol{\gamma}_{\mathrm{id}} = [\gamma_{\mathrm{RF}}, \gamma_{\mathrm{LF}}, \gamma_{\mathrm{RH}}, \gamma_{\mathrm{LH}}]^T$ are planned according to the planned contact state. The factors indicate the fraction of the maximum contact wrench capability of an end effector, which may be used by contact force control to generate the desired total CoM wrenches and stabilize the robot. Typical trajectories for the foot and the corresponding load factor are depicted in Figure 3.3.

Stage 2: ZMP Trajectory The WPG plans a piecewise linear ZMP trajectory with predefined margin to the edges of the BoS, see Figure 3.4. This solution minimizes the change of the ZMP over time and thus makes the resulting contact forces easier to track by contact force control [37]. The ZMP trajectory is traversed with constant velocity, except for the first and last step [266]. Note that this trajectory is just one possible solution, which works well with the proposed control schemes and leads to smooth walking. The ZMP-based planning generally allows arbitrary walking gaits as it is just related to feasibility of the motion, see Section 3.1.1.

[3] LOLA's latest WPG has been developed by Philipp Seiwald.

[4] The step length is the distance between the left and right foot when taking a step. In contrast, the stride length is the distance a foot travels before touching the ground again.

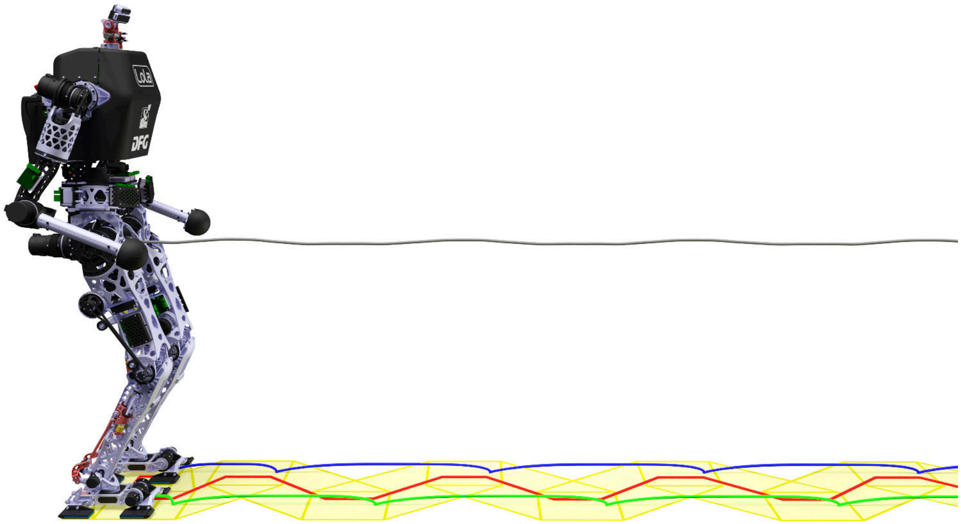


Figure 3.4: Typical ZMP trajectory (red) with the BoS (yellow), foot trajectories (green, blue), and resulting CoM trajectory (grey) for Lola.

Stage 3: CoM Trajectory As proposed in [33], the WPG uses a three-mass model — one mass for the torso, and one for each leg — as reduced dynamic model for the biped. Compared to the commonly-used linear inverted pendulum model, this additionally takes the dynamics of swing foot accelerations into account. Another advantage is that the vertical CoM trajectory can be defined arbitrarily. Based on the model, the linear- and angular momentum $\dot{\boldsymbol{p}}, \dot{\boldsymbol{L}}^{\text{CoM}}$ at the CoM may be written as

$$\dot{\boldsymbol{p}} = \boldsymbol{F} + m\boldsymbol{g} \tag{3.13}$$

$$\dot{\boldsymbol{L}}^{\text{CoM}} = \dot{\boldsymbol{L}}^{\text{CoM}}_{\text{RF}} + \dot{\boldsymbol{L}}^{\text{CoM}}_{\text{LF}} + \dot{\boldsymbol{L}}^{\text{CoM}}_{\text{Torso}} = \boldsymbol{T} \tag{3.14}$$

$$\Lambda^{CoM}_{\text{id}} = \begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{T} \end{bmatrix} \tag{3.15}$$

with the mass of the robot m, and the contact forces $\boldsymbol{F}$ and torques $\boldsymbol{T}$ on the CoM. The angular momentum for the reduced dynamics model can be split into parts for the torso and the foot masses. The first two components of $\boldsymbol{T}$ are already known from the planned ZMP trajectory. Furthermore, the vertical components of $\dot{\boldsymbol{p}}$ and $\dot{\boldsymbol{L}}^{\text{CoM}}_{\text{Torso}}$ are specified by the desired vertical CoM trajectory and the desired torso rotation around the vertical axis, respectively. The previously planned foot trajectories completely define $\dot{\boldsymbol{L}}^{\text{CoM}}_{\text{RF}}$ and $\dot{\boldsymbol{L}}^{\text{CoM}}_{\text{LF}}$. This leaves the horizontal motion of the torso mass (and therefore the CoM), and the vertical component of $\boldsymbol{T}$ as unknowns to be determined. Using all known entities, the equations of motions become two linear ordinary differential equations, which form a boundary value problem together with boundary conditions for the start and end of the robot's motion. The problem is overdetermined and its solution is approximated by a collocation method with quintic splines. More details can be found in [266].

The solution for a CoM trajectory is run iteratively to consider kinematic limits properly. Based on a simplified kinematic model, the desired vertical torso trajectory is changed in each iteration when kinematic limits are violated. This method is generally not active in the experiments conducted as part of this thesis unless noted differently.

Multi-Contact For improved robustness, the WPG module can use additional hand contacts in challenging scenarios. This involves selecting a suitable support point, planning feasible

Parameter	Value	Note
Step Time	0.8 s	Time between two double-support phases
Step Length	0.4 m	Distance between two steps (half the stride length)
Step Height	7 cm	
Mean Torso Velocity	0.5 m/s	

Table 3.3: The default walking pattern gait parameters used for experiments in this thesis.

hand trajectories, and determining the timing for a transition to contact control and vice versa. Moreover, the planner may decide to explicitly consider a certain contact force at the hands when calculating the CoM trajectory, which results in leaning against the support. As an ultimate goal, the planner should use vision-based data to generate an environment model for an autonomous generation of trajectories. In its current preliminary state, the WPG uses predefined hand positions and timings without vision-based data in the loop.

Interface to SIK At every control-module time step Δt_{cont}, the WPG evaluates the planned trajectories for the current time and sends the result to SIK via shared memory. The data contains the complete task-space vector $\boldsymbol{X}$, the ideal contact wrench $\boldsymbol{\Lambda}_{\text{id}}^{\text{CoM}}$ from Equation (3.15), the load factors γ_{id}, task-space selection factors ξ, and surface normals $\boldsymbol{N}$ for the end effectors:

$$\text{WPG}_{\text{data}} = \{\boldsymbol{X} \quad \boldsymbol{V} \quad {}_{W}\boldsymbol{\Lambda}_{\text{id}}^{\text{CoM}} \quad \gamma_{\text{id}} \quad \xi \quad \boldsymbol{N}\}. \tag{3.16}$$

The contact normals for the feet always point along the ${}_{E}z$ axis; for the hands, the normals are based on the environment model.

Default Gait Parameters The default gait parameters used for all experiments in this thesis — unless noted differently — are described in Table 3.3.

3.6 Sensor Data Preprocessing

In the following, essential parts of the sensor data preprocessing module (10 in Figure 3.1) are described. Further parts of this module are described in Chapter 5 along with the data-consuming modules.

3.6.1 Floating-Base Inclination

The floating-base inclination is defined by the rotations in the sagittal and frontal plane between the ideal world FoR W and the inertial frame I, see Section 3.4. In general, this transformation can be obtained when the planned (ideal) and the actual orientation of any robot body is known, see Section 3.1. For LOLA, the IMU is located at the torso, and the floating-base inclination is consequently calculated from this body's orientation.

The IMU provides roll-pitch-yaw data for the transformation from the IMU-fixed frame U to the inertial frame I and the corresponding angular velocity ${}_{U}\boldsymbol{\omega}_{\text{IMU}} = {}_{U}\boldsymbol{\omega}_{I,U_{\text{m}}}$ (index m for measured). The roll and pitch angles are automatically initialized via the gravity field when the sensor is switched on. The yaw angle, however, is set to zero in an unknown initial pose of the robot and it is not required for the stabilization of the robot's floating base.

Consequently, the yaw measurements are not used for the calculation of the floating-base inclination, assuming the robot always follows the desired (planned) yaw orientation.

Based on the measured roll and pitch angles, setting yaw to zero, the rotation vector ${}_{I'}\boldsymbol{\vartheta}_{\mathrm{IMU}} = {}_{I'}\boldsymbol{\vartheta}_{I',U_{\mathrm{m}}}$ is calculated. To retrieve a full transformation to the inertial FoR, the desired yaw transformation ${}_{I}\boldsymbol{s}_{I'}$ is multiplied to the left:

$$ {}_{I}\boldsymbol{s}_{U_{\mathrm{m}}} = {}_{I}\boldsymbol{s}_{I'} \otimes \mathrm{quat}({}_{I'}\boldsymbol{\vartheta}_{\mathrm{IMU}}). \tag{3.17} $$

The pure yaw transformation ${}_{I}\boldsymbol{s}_{I'}$ is calculated from the angle between the vectors $\boldsymbol{e}_x$ and $\mathrm{diag}([1,\ 1,\ 0])\,{}_{W}\boldsymbol{A}_{U_{\mathrm{id}}}\boldsymbol{e}_x$ using planned entities only. The floating-base inclination is then calculated from the actual orientation ${}_{I}\boldsymbol{s}_{U_{\mathrm{m}}}$ and the ideal (planned) orientation ${}_{W}\boldsymbol{s}_{U_{\mathrm{id}}}$. The angular velocity of the floating base results from the actual and the planned entities ${}_{U}\boldsymbol{\omega}_{\mathrm{IMU}}$, and ${}_{W}\boldsymbol{\omega}_{W,U_{\mathrm{id}}}$, respectively:

$$ {}_{U}\boldsymbol{\vartheta}_{\mathrm{FB}} = \mathrm{wrap}\left[\mathrm{rotVec}\left({}_{W}\boldsymbol{s}_{U_{\mathrm{id}}}{}^{*} \otimes {}_{I}\boldsymbol{s}_{U_{\mathrm{m}}}\right)\right] \tag{3.18} $$

$$ {}_{U}\boldsymbol{\omega}_{\mathrm{FB}} = {}_{U}\boldsymbol{\omega}_{\mathrm{IMU}} - \mathrm{rotMat}({}_{W}\boldsymbol{s}_{U_{\mathrm{id}}})^{T}\ {}_{W}\boldsymbol{\omega}_{W,U_{\mathrm{id}}}. \tag{3.19} $$

The wrap() operator shifts the rotation vector to the smallest representation to get the shortest angle between the two orientations.

3.6.2 Contact Transition Detection

The ideal timing of contacts with the environment is determined by the planned load factors γ_{id}. Disturbances of the robot's state and undetected terrain properties, however, change the actual contact timing. The purpose of this preprocessing step is the detection of contact transitions on the end effectors using FT sensor data and discrete contact state data $\boldsymbol{\chi}_m$ from contact switches (see Section 2.1.1) or a tactile sensor (see Section 2.4). The sole use of FT sensor data may lead to false positives when the sensors experience high accelerations (finding a force threshold is difficult).

Actual Contact Factors The actual contact factor $\kappa_e \in [0, 1]$ for an end effector e describes the actual contact state (1.0 closed, 0.0 open). The input data for the generation of these factors is discrete, i.e., non-smooth, but κ_e is used to activate or deactivate the output of velocity-level controllers continuously. The factors must be C^1 smooth to achieve finite desired accelerations from the control laws' output. Thus, the discrete states are rate limited with δ_κ and low-pass filtered, see Algorithm 2.

Early Contacts An early contact occurs when the contact on an end effector closes earlier than expected from the ideal plan. The algorithm to detect an early contact event on an arbitrary end effector is described in Algorithm 3. A combination of FT data and measured contact states from contact switches or a tactile foot sole is used. By using a force threshold $F_{ec,th,e}$ for the activation, minor contact-timing imperfections are tolerated; without this threshold, almost every step would be classified as being early, e.g., due to slight floating-base inclinations. The detection logic can equally be used for hand contacts with the environment, using a different parameterization for the contact force thresholds. Theoretically, the algorithm works with data acquired solely from the FT sensors. However, the combination of data from the contact switches and the FT sensors makes the algorithm robust to wrong sensor readings from either sensor type. The early contact state ends when the desired load on the end effector becomes non-zero, and the contact force controller is completely activated.

```
input         : Contact force norm $F_e = ||\boldsymbol{\lambda}_{e,\mathrm{m}}||$, measured binary contact state $\boldsymbol{\chi}_{e,\mathrm{m}}$
                for end effector $e$, sample time $\Delta t_{\mathrm{cont}}$
output        : Actual contact factor $\kappa_e$
initial state: linFactor $\leftarrow 0$;
contactClosed $\leftarrow ||\boldsymbol{\chi}_{e,\mathrm{m}}|| > 0 \;||\; F_e > F_{c,th,e}$;
if contactClosed then
    changeRate $\leftarrow \delta_\kappa$;
else
    changeRate $\leftarrow -\delta_\kappa$;
end
linFactor $\leftarrow$ linFactor $+$ changeRate $\Delta t_{\mathrm{cont}}$ ;
linFactor $\leftarrow \max(\min(1.0,$ linFactor$), 0.0)$;
$\kappa_e \leftarrow \mathrm{lpf1}(T = 0.003\,\mathrm{s},$ linFactor$)$;
```

Algorithm 2: The algorithm to generate a C^1 smooth actual contact factor κ_e from raw sensor data $\boldsymbol{\chi}_{e,\mathrm{m}}$ for end effector e. The measured input contact state is acquired from discrete contact switches or a tactile skin. A typical parametrization is $F_{c,th,f} = 200\,\mathrm{N}$, $F_{c,th,h} = 10\,\mathrm{N}$, $\delta_\kappa = \frac{1}{0.02\,\mathrm{s}}$. For the implementation of lpf1 refer to Appendix A.3.

```
input         : Contact force norm $F_e = ||\boldsymbol{\lambda}_{e,\mathrm{m}}||$, contactClosed from Algorithm 2, ideal
                load $\gamma_e$, force control activation factor from the last time step
                $\beta_e[n-1]$, see Algorithm 5, sample time $\Delta t_{\mathrm{cont}}$
output        : Early contact state $\epsilon_e$
initial state: plannedAirTime $\leftarrow 0$;
if $!\epsilon_e$ then
    if contactClosed $\&\&\ \gamma_e == 0\ \&\&\ F_e > F_{ec,th,e}\ \&\&$ plannedAirTime $> \Delta t_{\epsilon,dead}$ then
        $\epsilon_e \leftarrow 1$;
    end
else if $\gamma_e > 0\ \&\&\ \beta_e[n-1] > 0.99$ then
    $\epsilon_e \leftarrow 0$;
end
if $\gamma_e > 0$ then
    plannedAirTime $\leftarrow 0$;
end
plannedAirTime $\leftarrow$ plannedAirTime $+ \Delta t_{\mathrm{cont}}$;
```

Algorithm 3: The algorithm to detect an early contact for an end effector. A typical parametrization is $F_{ec,th,f} = 100\,\mathrm{N}$, $F_{ec,th,h} = 100\,\mathrm{N}$, $\Delta t_{\epsilon,\mathrm{dead}} = 0.3\,\mathrm{s}$.

Late Contacts A late contact of an end effector happens when the contact of an end effector closes significantly later than ideally planned via γ_e. The algorithm to detect a late contact is depicted in Algorithm 4.

```
input        : contactClosed from Algorithm 2, ideal load $\gamma_e$, sample time $\Delta t_{\text{cont}}$
output       : Late contact state $\iota_e$
initial state: contactDelay $\leftarrow$ 0;
if $!\iota_e$ then
    if !contactClosed && $\gamma_e > 0$ then
        contactDelay $\leftarrow$ contactDelay + $\Delta t_{\text{cont}}$;
    else
        contactDelay $\leftarrow$ 0.0
    end
    if contactDelay > $\Delta t_{\iota,dead}$ then
        $\iota_e \leftarrow 1$
    end
else if contactClosed || $\gamma_e == 0$ then
    $\iota_e \leftarrow 0$;
    contactDelay $\leftarrow$ 0.0
end
```

Algorithm 4: The algorithm to detect a late contact for an end effector. A typical parametrization for the feet is $\Delta t_{\iota,\text{dead}} = 0.12\,\text{s}$.

3.6.3 Force Control Activation

An end effector may either be position-controlled (adhering to the planned reference trajectories), or force-controlled (when in contact with the environment). The nature of biped walking requires to repeatedly switch between these control modes. A force-controlled end effector corresponds to a force control activation factor $\beta = 1$, while position-control mode means $\beta = 0$. This control mode blending factor β_e for an end effector e is generated from the algorithm depicted in Algorithm 5.

Generally speaking, the force control mode is turned on when there is a planned load on the respective end effector. In the case of early contact, the end effector is immediately switched to force control. In the case of late contact, it seems rational to keep the opposite foot (the swing foot) in force-controlled mode until the late contact situation is over (the planned stance foot made contact). Because the swing foot can not break contact with the environment as long as the planned stance foot is in the air, this would otherwise lead to contact with the environment in position-controlled mode, reducing the ability to stabilize the robot. The control mode factors are changed linearly over time with the specified rate limits and processed by a subsequent low-pass filter to ensure C^1 smoothness. This is required to have C^2 smoothness of the control approaches using the activation factors in their control laws, see Chapter 5.

```
input         : Early contact states $\boldsymbol{\epsilon}$, late contact states $\boldsymbol{\iota}$, ideal loads $\boldsymbol{\gamma}$, sample time
                $\Delta t_{\text{cont}}$
output        : Force control activation factors $\boldsymbol{\beta}$
initial state: lateContactActivation $\leftarrow 0$;

for all end effectors e do
    if $\gamma_e > 0$ then
        changeRate $\leftarrow \delta_{\beta,e}$;
    else
        changeRate $\leftarrow -\delta_{\beta,e}$;
    end
    /* Immediate increase on early contact                              */
    if $\epsilon_e > 0$ then
        linBeta$_e \leftarrow 1.0$;
        changeRate $\leftarrow 0$;
    end
    linBeta$_e \leftarrow$ linBeta$_e$ + changeRate $\Delta t_{\text{cont}}$ ;
    linBeta$_e \leftarrow \max(\min(1.0,$ linBeta$_e), 0.0)$;
end

/* Keep swing foot force-controlled when stance foot contact is
   late                                                                 */
if $\iota_{RF}$ || $\iota_{LF}$ then
    lateContactActivation $\leftarrow 1.0$;
    changeRate $\leftarrow 0.0$;
else
    changeRate $\leftarrow -\delta_{\beta,lc}$
end
lateContactActivation $\leftarrow$ lateContactActivation + changeRate $\Delta t$;
lateContactActivation $\leftarrow \max($lateContactActivation$, 0.0)$;
with swing foot as f
    linBeta$_f \leftarrow \max($linBeta$_f$, lateContactActivation$)$;
end

for all end effectors e do
    $\beta_e \leftarrow \mathrm{lpf1}(T = 0.003\,\mathrm{s},$ linBeta$_e)$;
end
```

Algorithm 5: The algorithm to generate C^1 smooth force control activation factors. A typical parametrization is $\delta_{\beta,f} = \frac{1}{0.03\,\mathrm{s}}$, $\delta_{\beta,h} = \frac{1}{0.06\,\mathrm{s}}$, $\delta_{\beta,lc} = \frac{1}{0.1\,\mathrm{s}}$.

3.7 Reactive Trajectory Adaptation

Parts of this section have previously been published in [292].

This section describes the trajectory adaptation module (2 in Figure 3.1), which modifies the planned reference trajectories based on discrete events and desired quantities. The module does not close (continuous) feedback loops using sensor data, but includes strategic feedforward approaches.

3.7.1 Early Contact Reflex

In order to achieve precise tracking of the end-effector trajectories — this is especially important for stepping over obstacles [114] or avoiding arm collisions with the environment — high-gain position control loops are required. On LOLA, the distributed joint controllers ensure the positional tracking of the desired joint trajectories. However, when the swing foot hits an undetected obstacle, the nonzero impact velocity causes high contact forces and disturbs the robot's state. In the following, an approach to mitigate the influence of such early contacts is presented, which preserves the quality of the end effectors' positional tracking. The strategy does not require re-planning of the CoM, or end-effector trajectories; instead, it reactively modifies the currently planned trajectories to minimize the impact on the system.

Feed-forward Control In the following, the control approach is described for early contacts on the feet. The methodology is equally applied to the robot's hands.

In an early-contact situation, the swing foot still has a (planned) non-zero velocity, when it hits the ground. The general idea is to reduce these planned velocities $\boldsymbol{v}_{e,\mathrm{id}}$ as fast as possible using event-triggered feed-forward control. The following velocity is added to the ideal trajectories of an early contact end effector:

$$ {}_W\Delta\boldsymbol{v}_{e,ec} = \mathrm{lpf2}(T = 0.003\,\mathrm{s}, d = 1.0, -\epsilon_{e,a}\, {}_W\boldsymbol{v}_{e,\mathrm{id}}). \tag{3.20} $$

The second-order low-pass filter is used to ensure C^2 smooth trajectories and limit the bandwidth of the feedforward controller to match limitations imposed by the robot's hardware. It uses critical damping to reduce overshoot of the foot (away from the obstacle). For the filter implementation, refer to Appendix A.3. The activation factor $\epsilon_{e,a}$ is set to 1.0 in the case of an early-contact, and is ramped-down linearly when the early contact situation is over to slowly return to the original foot velocities:

$$ \epsilon_{e,a} = \max\left(\epsilon_e,\ 1.0 - \sum_{n_{\mathrm{lastEC},e}}^{n} \delta_{\epsilon,a}\,\Delta t_{\mathrm{cont}}\right). \tag{3.21} $$

The variable $n_{\mathrm{lastEC},e}$ denotes the last time step $\epsilon_e > 0$ held. The maximum rate of change is typically parametrized with $\delta_{\epsilon,a} = \frac{1}{0.1\,\mathrm{s}}$. Ideally, this method reaches ${}_W\Delta\boldsymbol{v}_{e,ec} = -{}_W\boldsymbol{v}_{e,\mathrm{id}}$ shortly after the beginning of the early-contact situation, which means full compensation of the planned ideal foot velocities. The time-constant T of the low-pass filter determines the build-up time of the feed-forward approach (99 % in $5T$) and can be used to tune the speed of adaptation. The activation factor, planned, and adapted vertical velocity on an impacting

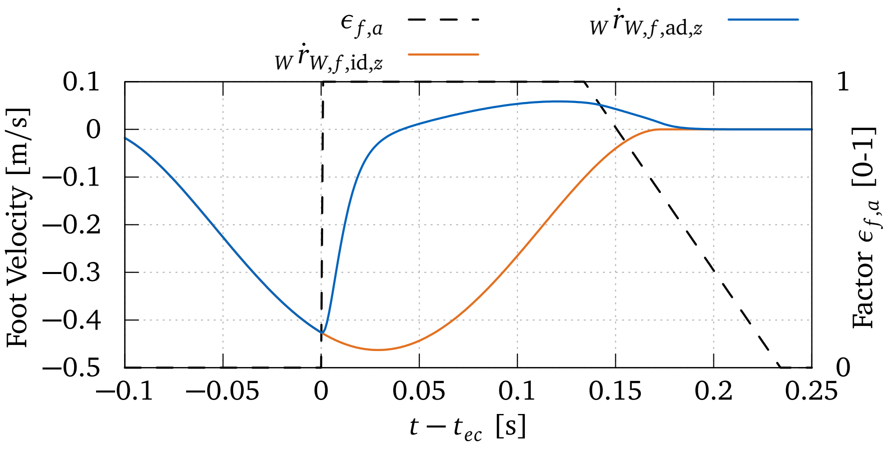


Figure 3.5: Activation factor (dashed black), planned vertical foot velocity (orange), and resulting adapted vertical foot velocity (blue) in an exemplary early-contact scenario at time t_{ec}.

foot are visualized in Figure 3.5. The ideal task-space trajectories are modified with the integrated velocity, which yields the adapted task-space trajectories:

$$ {}_W\Delta \boldsymbol{v}'_{e,ec}[n] = \text{posCtl}({}_W\Delta\boldsymbol{v}_{e,ec}, (1-\beta_e)(1-\epsilon_e)\,{}_W\Delta\boldsymbol{x}'_{e,ec}[n-1]) \tag{3.22} $$

$$ {}_W\Delta\boldsymbol{x}'_{e,ec} = \text{int}({}_W\Delta\boldsymbol{v}'_{e,ec}) \tag{3.23} $$

$$ \left(\boldsymbol{X}'_{\text{ad}}, \boldsymbol{V}'_{\text{ad}}\right) = {}_W\,\text{modifyAll}_e\left(\boldsymbol{X}_{\text{id}}, \boldsymbol{V}_{\text{id}}, {}_W\Delta\boldsymbol{x}'_{e,ec}, {}_W\Delta\boldsymbol{v}'_{e,ec}\right). \tag{3.24} $$

The position controller (see Appendix A.5), which drives the modifications back to zero, is active when the end effector control mode is position control and there is no early contact situation. Note that the integration of rotational velocities in Equation (3.23) is only approximative, see Appendix A.2.

Contact Transition The feed-forward control described above operates in the position domain to reduce the velocities of the end effectors as fast as possible. In parallel, feedback control of the contact forces is activated on the impacting end effector, see Section 3.6.3. The planned load on the end effector, however, is still zero. In order to keep the (early) contact at the end effector, and to avoid subsequent late contact situations, the planned load factor is increased to a low value γ_{ec}. Algorithm 6 describes the adaptation strategy. Note that the sum of the load factors of both feet is always 1, while the hand load factors are not constrained. This is expected by the wrench distribution heuristics, see Section 4.1. Figure 3.6 visualizes the resulting ideal load γ_{id} and adapted load γ_{ad} on the impacting foot.

Validation In Sygulla and Rixen [292], an earlier version of the early contact reflex has been validated and compared with LOLA's original approach [35] on LOLA 2020 (fig. 2.1c). Compared to the strategy described above, [292] uses a different task-space definition and leaves tangential impact velocities uncompensated. Unfortunately, due to software and hardware incompatibilities, a fair comparison of the strategy described above with the method in [35] is no longer possible on recent LOLA hardware.

Nevertheless, the current strategy is evaluated on the newest hardware concerning floating-base inclination and vertical contact forces on the impacting foot. The results of all three variants are depicted in Figures 3.7 and 3.8. For best comparability with the older results, the vertical CoM acceleration method described in Section 4.5 is activated with reduced gain $G_{CoM} = 0.26$.

```
input         : Early contact states $\boldsymbol{\epsilon}$, ideal loads $\boldsymbol{\gamma}_{\mathrm{id}}$
output        : Adapted loads $\boldsymbol{\gamma}_{\mathrm{ad}}$
initial state: $\Delta\gamma_e \leftarrow 0$ for all $e$;

$\boldsymbol{\gamma}_{\mathrm{ad}} \leftarrow \boldsymbol{\gamma}_{\mathrm{id}}$;
for all end effectors $e$ do
    if $\epsilon_e > 0$ then
        $\Delta\gamma_e \leftarrow \Delta\gamma_e + \delta_{\gamma,ec}\,\gamma_{ec}\,\Delta t$;
        $\Delta\gamma_e \leftarrow \min(\Delta\gamma_e, \gamma_{ec})$;
    else if $\gamma_{id,e} > \gamma_{ec}$ then
        $\Delta\gamma_e \leftarrow 0$;
    end
    if $\Delta\gamma_e > 0$ then
        $\gamma_{\mathrm{ad},e} = \max(\gamma_{\mathrm{ad},e}, \Delta\gamma_e)$;
        if $e$ is foot then
            // The load of the opposite foot is reduced
            $\gamma_{\mathrm{ad},\breve{e}} \leftarrow 1.0 - \gamma_{\mathrm{ad},e}$;
        end
    end
end
```

Algorithm 6: The algorithm to adapt the load on an end effector for an early contact situation. A typical parametrization is $\gamma_{ec} = 0.05$, $\delta_{\gamma,ec} = \frac{1}{0.05\,\mathrm{s}}$.

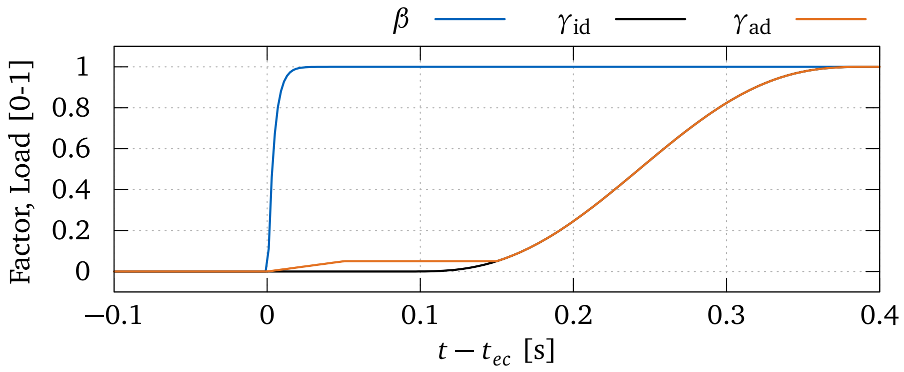


Figure 3.6: Force control activation factor β (blue), and the planned γ_{id} (black) and adapted γ_{ad} (orange) load for the impacting foot of an early contact at time t_{ec}.

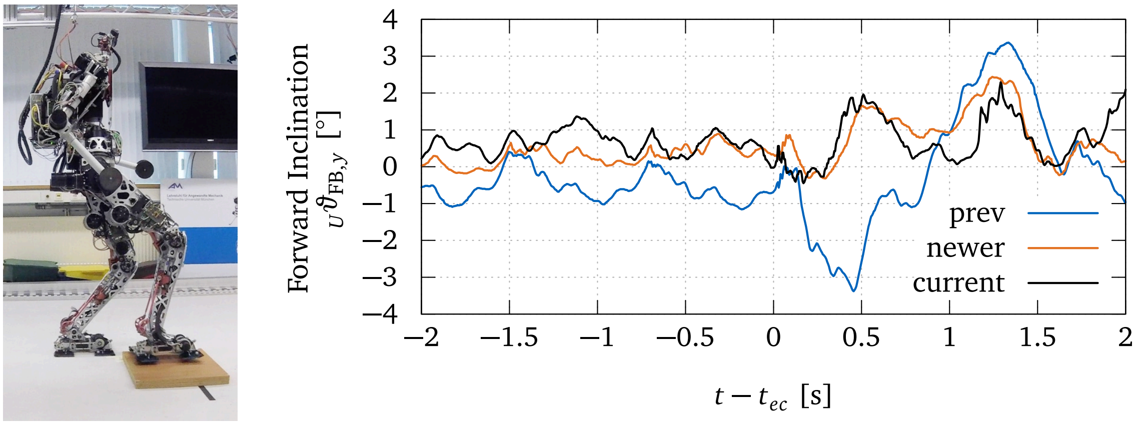


Figure 3.7: Experimental early-contact scenario with an undetected, 4-cm-high wooden board on LOLA 2020 (fig. 2.1c) [292]. The graph on the right shows the sagittal floating base inclination in an early-contact situation for the previous (*prev* [35]), newer method (*newer* [292]), and strategy described above (*current*). Note that the results for the current method are evaluated on LOLA 2021 (fig. 2.3), while the older approaches were tested on LOLA 2020 (fig. 2.1c).

In the experimental test scenario, LOLA walks in the direction of an unknown, 4 cm high undetected obstacle with the gait parameters described in Table 3.3. The earlier implementation of the method has been compared to the previous method of Buschmann, Ewald, Ulbrich, and Büschges [35] for an early-contact situation on LOLA 2020 (fig. 2.1c) [292]. The newer early-contact method provides a reduced impact-induced *change* in sagittal torso inclination from 3.1 ° to 1.2 °, see Figure 3.7. The change in frontal torso inclination is similarly reduced from 3.7 ° to 1.6 °. The impact forces decrease faster, and the "bouncing" of the foot is reduced — i.e., the second force peak is smaller, see Figure 3.8.

The method proposed above is an extension of the work in [292], uses a different task-space representation, and is also able to compensate tangential impact velocities of the feet. The results of an experiment in the same early-contact scenario, see Figures 3.7 and 3.8, proof the effectiveness against early contact situations. However, due to different walking gait parameters and hardware changes, it is impossible to isolate the effects of the tangential velocity compensation. Nevertheless, simulations with a multi-body model of LOLA have shown a benefit of the algorithm's extension.

Related Work An early-contact concept is essential for robots with position-controlled joints and the resulting high, intrinsic stiffness. There are several ways to mitigate the impact effects of the swing foot on uneven terrain, e.g., the modification of the joint controller gains based on the gait cycle timing [101, 218] or the activation of a ground reaction force controller once the swing foot is in contact [201]. Another strategy is implementing reflexive heuristics that change the vertical trajectory of the swing foot when an early contact occurs — i.e., when the vertical contact forces exceed a certain threshold [124, 135]. Other approaches trigger a replanning of the gait-cycle trajectories once an early contact occurs [35] or combine a fast walking-pattern regeneration with heuristics to stop the swing-foot upon early contact [215]. While these global approaches certainly have the ability to consider a reduced dynamics model of the system in the updated trajectories, the latency from an early-contact event to an actual change in trajectories is usually about 20 ms — this may lead to high impact forces for fast walking on uneven ground. Therefore, the proposed method is restricted to maintain the original trajectory plan and to modify the swing-foot trajectory based on the combination of a reflexive heuristic with force control at a latency of Δt_{cont}. Furthermore, findings with the early-contact method [35] previously used on LOLA indicate that maintaining the original

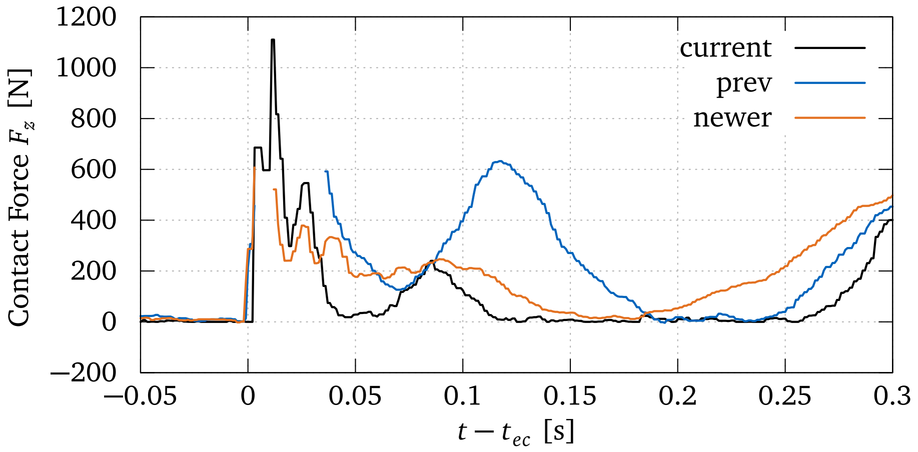


Figure 3.8: Right foot's vertical contact forces in an early-contact situation for the previous (*prev* [35]), newer (*newer* [292]) and current method (*current*). Note that the results for the current method are evaluated on LOLA 2021 (fig. 2.3), while the older approaches were tested on LOLA 2020 (fig. 2.1c). For the time right after the impact, no data is available for *prev* and *newer* as the FT sensor did not provide data at high loads due to a hardware issue by that time.

step timing is beneficial, as this extends the effective double-support phase in the presence of such disturbances. The combination of force control and reflexes has already been described in [101, 156]. In contrast to related work, the approach at hand includes an additional modification of the planned load on the impacting foot. This avoids bouncing effects of the impacting foot just after the early-contact event and allows for higher impact velocities.

3.7.2 Swing Foot Adaptation

When the robot steps on undetected uneven terrain, the early contact reflex stops the foot immediately and switches to contact-controlled mode. Both footholds are now located on different heights, which raises the question of which height the next swing foot should be set in such a case.

Ground Plane Matching Within this thesis, the following strategy is pursued: After an early- oder late contact, the next swing foot is set on exactly the same height as the current stance foot (which initially hit the uneven terrain). To implement this policy, the vertical deviations of both feet to the ideal planned trajectories are calculated from direct kinematics. The required modification of a foot height then results from the negative deviation of the opposite foot:

$$_W\Delta z_{f,\text{plane}} = -\boldsymbol{e}_z{}^T \left({}_W\boldsymbol{r}_{W,\check{f},\text{id}} - {}_W\boldsymbol{r}_{W,\check{f}} \right). \tag{3.25}$$

Note that this compensation is also required for level ground, because small deviations of the foot positions — caused by contact force or balance control, see Chapter 5 — otherwise lead to imperfect contact timing.

Footsole Deformation Compensation At the planning stage, infinite stiffness of the ground contact is assumed when foot trajectories are generated. The foot sole of LOLA consists of a flexible material, though, and the limited bandwidth of joint position control leads to additional compliance. Due to the deformation of the stance foot, the swing foot hits the ground too early, causing a non-smooth gait or disturbed state of the robot. Therefore, a corresponding additional modification

$$_W\Delta z_{\text{deform}} = \frac{m\,g}{c_{\text{foot}}}, \tag{3.26}$$

with the robot mass m, gravity constant g, and the total stiffness of one foot c_{foot}, is calculated. The value of c_{foot} is determined experimentally.

Time-Integration & Discussion Both modifications are applied to the height of the currently position-controlled foot (swing foot), tracked by a position controller:

$$_W\Delta\dot{z}_f[n] = \text{posCtl}(-\frac{_W\Delta z_{cm}[n-1]}{T_{sf,\text{decay}}}, (1.0-\beta_f)(_W\Delta z_f[n-1] - {_W\Delta z_{f,\text{plane}}} - {_W\Delta z_{\text{deform}}})), \tag{3.27}$$

where $_W\Delta z_f[n-1]$ is the integrated modification from the last time step. The common mode of the modification of both feet

$$_W\Delta z_{cm} = \frac{1}{2}(_W\Delta z_{\text{LF}} + {_W\Delta z_{\text{RF}}}) \tag{3.28}$$

decays to zero with time-constant $T_{sf,\text{decay}} = 0.2\,\text{s}$ to avoid building a vertical offset on both feet, for example, when repeated early contacts occur. The calculated velocities are integrated and added to the adapted task-space vector:

$$_W\Delta z_f = \text{int}(_W\Delta\dot{z}_f) \tag{3.29}$$

$$(\boldsymbol{X}_{\text{ad}}, \boldsymbol{V}_{\text{ad}}) = {_W\,\text{modifyAll}_f}\left(\boldsymbol{X}'_{\text{ad}}, \boldsymbol{V}'_{\text{ad}}, \begin{bmatrix}\boldsymbol{e}_z\\ \boldsymbol{0}\end{bmatrix} {_W\Delta z_f}, \begin{bmatrix}\boldsymbol{e}_z\\ \boldsymbol{0}\end{bmatrix} {_W\Delta\dot{z}_f}\right). \tag{3.30}$$

In words, this algorithm adds the vertical deviations of the stance foot inversely to the swing foot's trajectory, and the swing foot height is increased by the estimated deformation of the stance foot.

The presented swing foot adaptation method strongly depends on the chosen task-space definition. In some cases, this logic might not be explicitly necessary to generate the same policy. In general, the topic of swing foot placement in the context of known unknown uncertainty (there is a known uncertainty but not a quantified one) is rarely discussed in the literature. Most concepts avoid to explicitly model uncertainties in the perception of the environment and the consequences caused by undetected obstacles. Wahrmann et al. [315] show that an overestimation of the ground height leads to a smaller disturbance of LOLA than an underestimation, which justifies the chosen policy. However, such analysis is strongly dependent on the type of robot and control mechanisms, making it difficult to find generally valid policies for such cases.

3.8 Feedback Stabilization

The planned, adapted trajectories describe feasible but not necessarily stable motions for the biped robot. In this thesis's scope, the robot is stabilized using three continuous, cascaded feedback control loops. The outer loop modifies the desired total contact wrenches to drive the floating-base inclination $_U\boldsymbol{\vartheta}_{\text{FB}}$ of the robot to zero (see Chapter 4). The inner loops are contact wrench feedback for each end effector (see Chapter 5) and a method to adapt the CoM trajectory to match the current contact wrenches (see Chapter 4). In the following, all relevant control plants for the design of the feedback control loops are introduced.

3.8.1 Control System Plants

Figure 3.9 visualizes the feedback loops as a connection of transfer functions. Several simplifications must be made to describe the control system in this form. First, the system is linearized, i.e., the transfer functions depend on the robot's pose. Second, only the foot contacts are considered in this analysis. Furthermore, symmetry of the right and left foot's dynamics is assumed. All transfer functions describe the system in a settled state, i.e., when contacts are either fully closed or fully open, and the robot is in a standing pose. Although the unknown plants describe the system's dynamics only for a particular state, their structure at different operating points gives hints on the contained uncertainty, which is used to design robust feedback laws in the following chapters. The plants contain valuable information on the dynamics of actuators and sensors and the structural dynamics of the robot's bodies. Because the structural eigenfrequencies of lightweight robots can easily — even when high stiffness is a primary design goal — be in the operating range of feedback controllers (1–20 Hz) [15, 16], the plant data must either be used to drive mechanical design decisions [267] or for the design of robust control loops. The mechanical design of LOLA's torso has already been improved to shift the first eigenfrequencies to higher values, see Section 2.1.1. In the following chapters, the identified plant data is used to further increase the bandwidth of control loops under structural eigenfrequencies' influence.

3.8.2 Onboard Plant Estimation

The identification of the unknown plants in Figure 3.9 is made on LOLA using only onboard sensors and actuators. This approach does not require additional hardware for the identification. It automatically includes sensor and actuator characteristics in the identified plants. The identification must, due to the coupling of the Multiple-Input Multiple-Output (MIMO) plants / controllers, be done open-loop, i.e., with deactivated CoM and contact controllers $\boldsymbol{C}_{\text{CoM}}(s) = \boldsymbol{0}$, $\boldsymbol{C}_{\text{contact}}(s) = \boldsymbol{0}$, see Section 4.5 and chapter 5. Otherwise, it can not be guaranteed that only one input of the plants is excited at a time because the controllers themselves may connect different plant inputs via their loops.

All task-space measures are described on velocity-level to avoid errors caused by slight differences in the robot's pose. A periodic excitation is added to the controller output. By observing the respective system output, the plant transfer function can be identified [183]. In the described case, two different identification input locations (see Figure 3.9) are sufficient to identify all unknown plants. Only one component of one excitation input is used at a time. For the identification of $\boldsymbol{P}_{\text{contact}}(s)$, only one foot is excited via $\boldsymbol{u}_1$. $\boldsymbol{P}_{\text{structure}}(s)$ is identified by simultaneous, in-phase excitation of both feet (via $\boldsymbol{u}_1$). The third plant $\boldsymbol{P}_{\text{CoM}}(s)$ is excited via the additional input $\boldsymbol{u}_2$. A superposition of multiple sine waves is used for the excitation of the system. Compared to the commonly used chirp signal, single frequencies are excited for a longer time, which reduces the influence of sensor noise and the transient states of the system (for example, caused by friction in the joint actuators). A further advantage is the precise control of the energy brought in for a certain frequency range. Compared to a chirp signal, the system is not excited at frequencies outside the specified band [183]. This comes at the price of higher measurement effort and lower frequency resolution. The sine waves use Schroeder phases to minimize the crest factor — i.e., overlapping peaks — of the signal [260]. Table 3.4 lists the excited frequency ranges.

System identification is conducted in the robot's standing pose (see Figure 3.2), once in double support and once in single support on the left foot. Based on the periodograms of recorded input and output signals for an unknown plant, a transfer function is estimated for every pose and frequency range using the *Matlab Signal Processing Toolbox* [304].

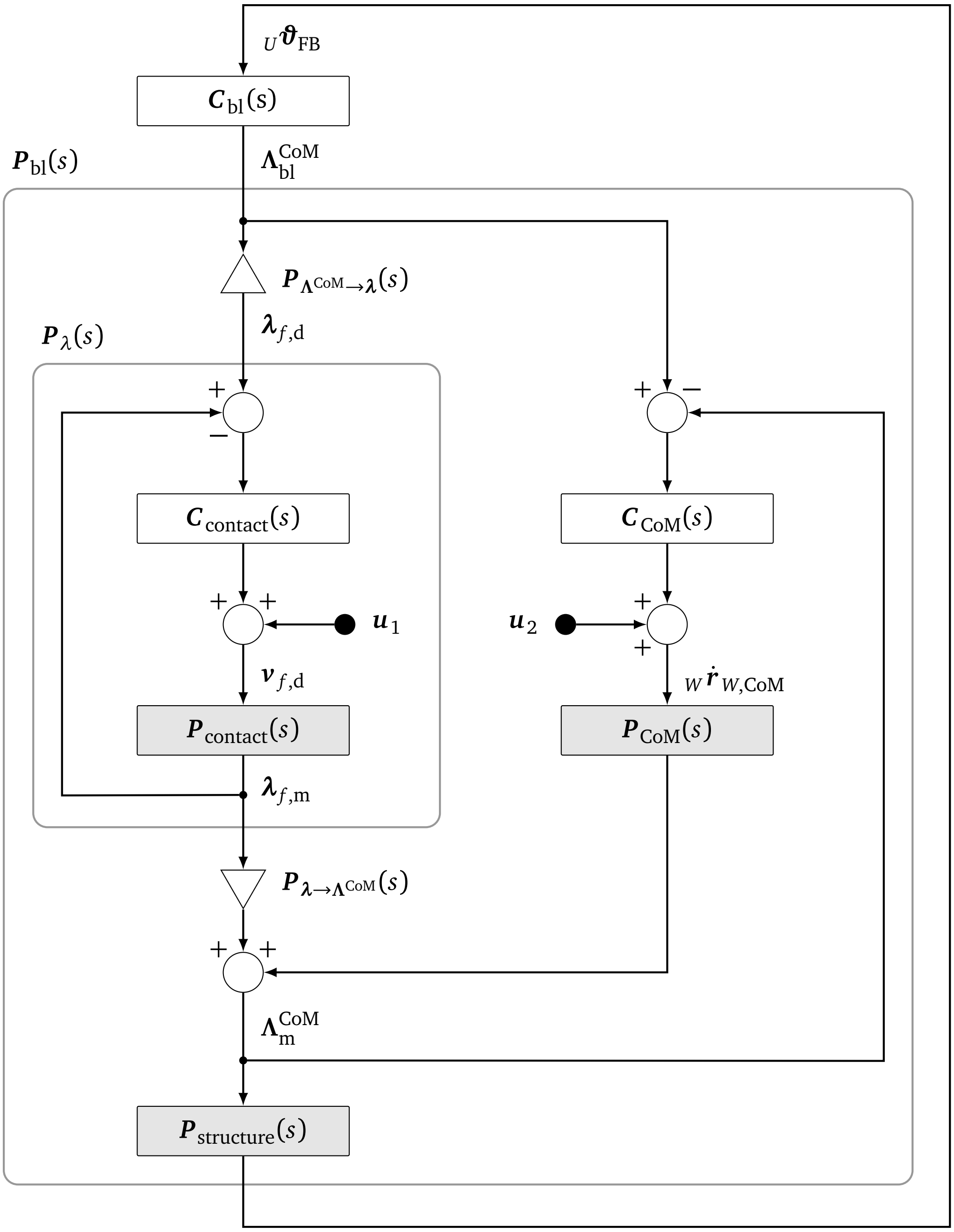

${}_U\boldsymbol{\vartheta}_{\mathrm{FB}}$
$\boldsymbol{C}_{\mathrm{bl}}(s)$
$\boldsymbol{P}_{\mathrm{bl}}(s)$
$\boldsymbol{\Lambda}_{\mathrm{bl}}^{\mathrm{CoM}}$
$\boldsymbol{P}_{\boldsymbol{\Lambda}^{\mathrm{CoM}}\to\boldsymbol{\lambda}}(s)$
$\boldsymbol{\lambda}_{f,\mathrm{d}}$
$\boldsymbol{P}_{\lambda}(s)$
$+$
$-$
$+$
$-$
$\boldsymbol{C}_{\mathrm{contact}}(s)$
$\boldsymbol{C}_{\mathrm{CoM}}(s)$
$+$
$+$
$\boldsymbol{u}_1$
$\boldsymbol{u}_2$
$+$
$+$
$\boldsymbol{v}_{f,\mathrm{d}}$
${}_W\dot{\boldsymbol{r}}_{W,\mathrm{CoM}}$
$\boldsymbol{P}_{\mathrm{contact}}(s)$
$\boldsymbol{P}_{\mathrm{CoM}}(s)$
$\boldsymbol{\lambda}_{f,\mathrm{m}}$
$\boldsymbol{P}_{\boldsymbol{\lambda}\to\boldsymbol{\Lambda}^{\mathrm{CoM}}}(s)$
$+$
$+$
$\boldsymbol{\Lambda}_{\mathrm{m}}^{\mathrm{CoM}}$
$\boldsymbol{P}_{\mathrm{structure}}(s)$

Figure 3.9: Simplified representation of the feedback control loops in the frequency domain. Transfer functions with white background are designed in the following chapters; blocks with gray background depend on the robot's structural and multi-body dynamics, and the contacts with the environment. Triangle blocks describe the wrench distribution and its inverse, see Section 4.3. There are two excitation inputs $\boldsymbol{u}_1$, $\boldsymbol{u}_2$ for the system. $\boldsymbol{C}_{\mathrm{bl}}(s)$ denotes the transfer function of the inclination controller, see Section 4.2. $\boldsymbol{C}_{\mathrm{contact}}(s)$ and $\boldsymbol{P}_{\mathrm{contact}}(s)$ denote the transfer function of a contact force controller and its plant, respectively, see Section 5.2. $\boldsymbol{C}_{\mathrm{CoM}}(s)$ and $\boldsymbol{P}_{\mathrm{CoM}}(s)$ describe the CoM acceleration controller and its respective plant, see Section 4.5. The loop of the inclination controller is closed via the plant $\boldsymbol{P}_{\mathrm{structure}}(s)$ from total wrenches at the CoM to the floating-base inclination.

Frequency Range [Hz]	Nr. of Sine Waves	Frequency Resolution [Hz]
0.5–5	31	0.15
5–10	31	0.17
10–15	31	0.17
15–20	31	0.17
20–25	31	0.17
25–30	31	0.17
30–40	31	0.33
40–50	31	0.33

Table 3.4: The frequency range and number of sine waves for the system identification. The system's response to the excitation in each line is recorded separately.

A *Hann* window is used to reduce leakage effects. The resulting discrete transfer function samples at the excitation frequencies are then merged to an estimated transfer function over the excitation frequency range.

The identification technique has been cross-validated with a modal analysis of LOLA, showing a good match of structural eigenfrequencies and the identified plant's resonances [16]. The concept may be used to compare different hardware revisions of one robot or different robots in terms of their structural eigenfrequencies. Given no changes are necessary on the hardware, the identification can be carried out rather quickly. In contrast to a modal analysis with external excitation and sensors, the system's structural modes can, however, not be identified. Initial experiments with this technique on LOLA were conducted by [184].

3.9 Inverse and Direct Kinematics

The desired task-space trajectories from the feedback module ($\boldsymbol{X}_\mathrm{d}, \boldsymbol{V}_\mathrm{d}$) are fed into a local IK scheme based on Automatic Supervisory Control (ASC) [180]. This IK module (6 in Figure 3.1) calculates the solution of the optimization problem

$$\min \ \frac{1}{2}\dot{\boldsymbol{q}}^T\dot{\boldsymbol{q}} + \alpha_N \left(\frac{\partial H}{\partial \boldsymbol{q}}\right)^T \dot{\boldsymbol{q}} \tag{3.31}$$

$$\text{subject to} \ \ \boldsymbol{v} - \boldsymbol{J}\dot{\boldsymbol{q}} = 0, \tag{3.32}$$

i.e., the joint velocities $\dot{\boldsymbol{q}}$ and the additional objective function H are locally minimized subject to the constraints of the robot's kinematics described by the task-space jacobian $\boldsymbol{J}$ [34]. The problem is solved [206] via the generalized (pseudo) inverse

$$\boldsymbol{J}^\# = \boldsymbol{J}^T\left(\boldsymbol{J}\boldsymbol{J}^T\right)^{-1} \tag{3.33}$$

using the relations

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^\# \boldsymbol{v}_{\mathrm{d},\mathit{eff}} - \alpha_N\left(\boldsymbol{I} - \boldsymbol{J}^\#\boldsymbol{J}\right)\left(\frac{\partial H}{\partial \boldsymbol{q}}\right)^T \tag{3.34}$$

$$\boldsymbol{v}_{d,\mathit{eff}} = \boldsymbol{v}_\mathrm{d} + G_\mathrm{IK}\,\Delta\boldsymbol{x}. \tag{3.35}$$

The desired joint angles $\boldsymbol{q}$ are obtained from $\dot{\boldsymbol{q}}$ via numerical integration. In order to eliminate numerical error $\Delta\boldsymbol{x}$, the desired task-space velocity $\boldsymbol{v}_\mathrm{d}$ is extended by a compensator with gain G_IK.

Direct Kinematics The actual robot state $\boldsymbol{x}$ is obtained via a recursive Direct Kinematics (DK) implementation (for details see [33]), which is based on the output of the IK $\boldsymbol{q}_d^T = \begin{bmatrix} \boldsymbol{q}_{U,\mathrm{d}}^T & \boldsymbol{q}_{\mathrm{J,d}}^T \end{bmatrix}$ — not the measured joint configuration of the robot. This makes the DK data less prone to noise; tracking errors of the joints are not considered, which implicitly assumes perfect tracking. The desired pose for the upper body $\boldsymbol{q}_{U,\mathrm{d}}$ is a by-product of the IK due to all quantities in the task-space being defined relative to W. The DK module (9 in Figure 3.1) provides position and velocity data of the task-space, position and velocity data of all robot bodies, and the Jacobians required for the calculation of IK.

Secondary Objectives The specification of the cost function H allows the exploitation of the robot's redundancy with respect to the chosen task-space definition to meet secondary control objectives. For LOLA, cost terms for a comfort pose, joint limit- and self-collision avoidance, and the minimization of the robot's total vertical angular momentum are considered.

The comfort pose and joint limit avoidance terms are based on the robot's current joint-space configuration, keeping the pose away from joint limits and close to a preferred configuration. Self-collision avoidance is based on distance calculations between predetermined bodies of the robot, mostly the elbows and hips. The vertical angular momentum compensation uses a full dynamics model of the robot to determine the angular momentum caused by the base (feet, hips, torso). Corresponding cost gradients on unconstrained bodies (the arms) are then used to approximately compensate the base-induced angular momentum. For more details on the secondary objective methods, refer to [34, 263, 264].

Task-Space Blending For standard walking, the exact position of the robot's hands is irrelevant, leaving the pose of the arms fully available for the realization of secondary objectives. In the case of a planned contact between the hand and the environment (multi-contact), however, the task-space hand trajectories must be tracked exactly to avoid collisions with the environment and allow precise end-effector force control. The task-space definition needs to be changed online to achieve this behavior with the used velocity-level IK.

One possibility is the use of hierarchical IK algorithms, treating $\boldsymbol{x}_{\mathrm{B}}$ as the highest priority task, and $\boldsymbol{x}_{\mathrm{RH}}$, $\boldsymbol{x}_{\mathrm{LH}}$ as the lower priority tasks. Individual hand position tasks may be blended out using smooth blending factors [151]. The implementation of a corresponding IK scheme, however, does either not provide exact tracking of lower-priority tasks [47] or is computationally expensive and prone to algorithmic singularities (when subtasks become orthogonal) [151, 275].

A different possibility is to change the task-space definition online, sometimes also termed task-space or nullspace switching. Exchanging the pseudoinverses of different task-space definitions, however, leads to discontinuities in the solution [192]. Interpolations in task-space selectively nullify regions of the Jacobians and task-space vectors [192]. Additional coupling matrices need to be computed to ensure smoothness in the joint-space during the continuous blending process.

A more straightforward solution, using interpolation in joint-space, is described by An and Lee [8]. A modified version of this approach is used for the multi-contact IK solution in the SIK module. The continuous task-space selection factors ξ define if a corresponding hand end effector is in the task-space or not, see Section 3.4.2. The factors must be C^1 smooth to achieve C^2 smooth joint trajectories because they are multiplied to the joint velocities, which require C^1 smoothness.

Based on the four different task-space combinations

$$\mathcal{S}_v = \left\{ \boldsymbol{v}_1 = {}_W\boldsymbol{v}_{\mathrm{B}},\ \boldsymbol{v}_2 = \begin{bmatrix} {}_W\boldsymbol{v}_{\mathrm{B}} \\ {}_W\boldsymbol{v}_{\mathrm{RH}} \end{bmatrix},\ \boldsymbol{v}_3 = \begin{bmatrix} {}_W\boldsymbol{v}_{\mathrm{B}} \\ {}_W\boldsymbol{v}_{\mathrm{LH}} \end{bmatrix},\ \boldsymbol{v}_4 = \begin{bmatrix} {}_W\boldsymbol{v}_{\mathrm{B}} \\ {}_W\boldsymbol{v}_{\mathrm{RH}} \\ {}_W\boldsymbol{v}_{\mathrm{LH}} \end{bmatrix} \right\}, \tag{3.36}$$

separate IK solutions are defined:

$$\dot{\boldsymbol{q}}_{n,\mathrm{d}} = \boldsymbol{J}_n^{\#} \boldsymbol{v}_{n,\mathrm{d},eff} - \alpha_N \left(\boldsymbol{I} - \boldsymbol{J}_n^{\#}\boldsymbol{J}_n\right)\left(\frac{\partial H}{\partial \boldsymbol{q}}\right)^T \qquad \boldsymbol{J}_n = \frac{\partial \boldsymbol{v}_n}{\partial \dot{\boldsymbol{q}}} \tag{3.37}$$

$$\boldsymbol{v}_{n,\mathrm{d},eff} = \boldsymbol{v}_{n,\mathrm{d}} + G_{\mathrm{IK}} \Delta \boldsymbol{x}_n \qquad \forall\, n = \{1,2,3,4\}. \tag{3.38}$$

The task-space error for the base vector must be calculated via the separate operator deltaX$_B$ to correctly consider the orientations:

$$\Delta \boldsymbol{x}_1 = \mathrm{deltaX}_B({}_W\boldsymbol{x}_{B,\mathrm{d}},\ {}_W\boldsymbol{x}_{\mathrm{B}}) \qquad \Delta \boldsymbol{x}_2 = \begin{bmatrix} \Delta \boldsymbol{x}_1 \\ {}_W\boldsymbol{x}_{\mathrm{RH,d}} - {}_W\boldsymbol{x}_{\mathrm{RH}} \end{bmatrix} \tag{3.39}$$

$$\Delta \boldsymbol{x}_3 = \begin{bmatrix} \Delta \boldsymbol{x}_1 \\ {}_W\boldsymbol{x}_{\mathrm{LH,d}} - {}_W\boldsymbol{x}_{\mathrm{LH}} \end{bmatrix} \qquad \Delta \boldsymbol{x}_4 = \begin{bmatrix} \Delta \boldsymbol{x}_2 \\ {}_W\boldsymbol{x}_{\mathrm{LH,d}} - {}_W\boldsymbol{x}_{\mathrm{LH}} \end{bmatrix} \tag{3.40}$$

The four IK solutions represent the possible extremes that may occur when the ξ factors are chosen to their maximum or minimum. Solutions in between are calculated via bilinear interpolation in the joint-space:

$$\dot{\boldsymbol{q}}_{\mathrm{d}} = \sum_{n=1}^{4} w_n \dot{\boldsymbol{q}}_{n,\mathrm{d}}, \tag{3.41}$$

with the weighing factors

$$w_1 = (1-\xi_{\mathrm{RH}})(1-\xi_{\mathrm{LH}}) \qquad w_2 = \xi_{\mathrm{RH}}(1-\xi_{\mathrm{LH}}) \tag{3.42}$$

$$w_3 = \xi_{\mathrm{LH}}(1-\xi_{\mathrm{RH}}) \qquad w_4 = \xi_{\mathrm{LH}}\,\xi_{\mathrm{RH}}. \tag{3.43}$$

All four IK solutions must be calculated for this algorithm to work. However, the implementations of the individual IKs are numerically efficient, see Chapter 6. Note that only the desired mechanical joint trajectories $\boldsymbol{q}_{\mathrm{J,d}}, \dot{\boldsymbol{q}}_{\mathrm{J,d}}$ — a subspace of $\dot{\boldsymbol{q}}_{\mathrm{d}}$ — are sent to the HWL, see Equation (3.2).

3.10 Chapter Summary

This chapter introduces the concepts of feasibility and stability in the context of biped robot research. Furthermore, an overview of the proposed control framework of LOLA and previous concepts is given. The chapter describes the chosen frames of reference, the task space, and IK algorithm. The utilized WPG module is explained as far as necessary for the experiments conducted in this thesis.

Further, algorithms for sensor data preprocessing and reactive trajectory adaptations are described. These approaches deal with the contact timing imperfections that occur in the context of uneven and unknown terrain. The reactive concepts enable the robot to deal with unexpected closing of contacts even at high impact speeds. Induced disturbances on the floating base can be mitigated effectively.

Also, applying an onboard system identification concept to acquire the robot's essential control plants is described. This chapter's concepts and definitions are essential prerequisites for developing stabilization methods described in the following chapters.

Chapter 4

Balance Control

Parts of this chapter have previously been published in [292].

This chapter deals with concepts to stabilize the floating-base dynamics of a biped robot in the presence of disturbances or uneven terrain. The methods modify the adapted trajectory $\boldsymbol{X}_{\text{ad}}$ and the total desired contact wrench on the CoM $\boldsymbol{\Lambda}_{\text{d}}^{\text{CoM}}$ (3 in Figure 3.1). Furthermore, a concept for the distribution of the total wrench to multiple end effectors in contact is described in this chapter (4 in Figure 3.1). The resulting desired wrenches $\boldsymbol{\lambda}_{\text{d}}$ for every end effector and the modified position trajectories $\boldsymbol{X}_{\text{bl}}$ describe the interface to the contact force controllers presented in Chapter 5.

The balance controller's primary goal is to stabilize the planned trajectories by counteracting deviations of the CoM location in an inertial FoR. In this work, the measured floating-base inclination between the planner's ideal world W and the actual inertial FoR I is used to balance the robot's state. The corresponding control approach for inclinations in the frontal and sagittal plane is covered in Section 4.2. A further balance strategy involving an additional vertical acceleration of the CoM is treated in Section 4.5. This chapter investigates the relationship between the balance control performance and structural resonances of the robot. Different approaches to mitigate parasitic effects from mechanical vibrations are implemented and compared experimentally. All presented balance concepts do not change the footstep locations or gait pattern and can thus be easily combined with obstacle avoidance algorithms.

4.1 Previous Work on Lola

The original balance controller used for Lola is proposed in [33] and uses a PD-type approach to calculate modifications of the torque acting on the CoM:

$$\Delta T_x^{\text{CoM}} = K_{P,x}\,\Delta\varphi_x + K_{D,x}\,\Delta\dot{\varphi}_x \tag{4.1}$$

$$\Delta T_y^{\text{CoM}} = K_{P,y}\,\Delta\varphi_y + K_{D,y}\,\Delta\dot{\varphi}_y, \tag{4.2}$$

where $\Delta\varphi$ describes the difference between desired and measured inclinations of the torso using a special representation of rotations introduced by Löffler [185]. The special angle definition allows an independent representation of inclinations in the sagittal (φ_x) and frontal plane (φ_y). In addition, vertical deviations of the CoM are regulated via modification of the total vertical contact force

$$\Delta F_z^{\text{CoM}} = K_{P,z}\,\boldsymbol{e}_z{}^T({}_W\boldsymbol{r}_{W,\text{CoM,id}} - {}_W\boldsymbol{r}_{W,\text{CoM}}) + K_{D,z}\,\boldsymbol{e}_z{}^T({}_W\dot{\boldsymbol{r}}_{W,\text{CoM,id}} - {}_W\dot{\boldsymbol{r}}_{W,\text{CoM}}) \tag{4.3}$$

to avoid drifting effects caused by inaccuracies or modeling errors when the feet are force-controlled. Typically, the derivative term was inactive with $K_{D,z} = 0$. The equations in this section originate from [33, p. 95] and are adapted to the notation used in this thesis.

Wrench Distribution The previous wrench distribution for LOLA uses a heuristic based on the planned load factors $\boldsymbol{\gamma}$ for the feet. Buschmann [33] proposes to maximize the use of normal forces due to the foot torques being limited by the size of the feet. The heuristic uses the planned load factors and a modification of the normal forces $\Delta F_{z,\text{stab}}$ to calculate the contact wrenches for both feet:

$$
\begin{aligned}
{}_W\boldsymbol{F}_{\text{RF}} &= \gamma_{\text{RF}}\, {}_W\boldsymbol{F}_{\text{bl}} + \boldsymbol{e}_z \Delta F_{z,\text{stab}} &\quad (4.4)\\
{}_W\boldsymbol{F}_{\text{LF}} &= \gamma_{\text{LF}}\, {}_W\boldsymbol{F}_{\text{bl}} - \boldsymbol{e}_z \Delta F_{z,\text{stab}} &\quad (4.5)\\
{}_W\boldsymbol{T}_{\text{feet}} &= {}_W\boldsymbol{T}_{\text{bl}} - {}_W\boldsymbol{r}_{\text{CoM,RF}} \times {}_W\boldsymbol{F}_{\text{RF}} - {}_W\boldsymbol{r}_{\text{CoM,LF}} \times {}_W\boldsymbol{F}_{\text{LF}} &\quad (4.6)\\
{}_W\boldsymbol{T}_{\text{RF}} &= \gamma_{\text{RF}}\, {}_W\boldsymbol{T}_{\text{feet}} &\quad (4.7)\\
{}_W\boldsymbol{T}_{\text{LF}} &= \gamma_{\text{LF}}\, {}_W\boldsymbol{T}_{\text{feet}}. &\quad (4.8)
\end{aligned}
$$

To calculate the normal force modification, the desired total torque component perpendicular to the connecting line of the feet (TCPs) in the ground plane is considered. $\Delta F_{z,\text{stab}}$ is chosen to realize this torque component (minimizing the foot torques) under the constraint of the unilateral contact and a maximum admissible normal force $F_{z,max} = \gamma_f\, 2mg$ for foot f, where m is the robot's total mass. The approach does not consider hand contacts. For details on the concept refer to [33]. Typical planned load factors are depicted in Figure 3.3. The original method was later extended to restrict the normal force modifications keeping a minimum normal force of 30 N per foot [292]. This extension helps to keep the feet in contact at any time, reducing the probability of late contacts.

Event-Based Walking Control For late-contact situations, the extension of the single support by horizontal acceleration of the CoM has been proposed for LOLA [316]. However, this concept requires an additional modification of the footstep locations to match the modified horizontal CoM trajectory. It is shown in the following that a vertical acceleration of the CoM can effectively be used on late-contact scenarios without having to modify the footstep locations.

4.2 Inclination Control

The stabilization of the rotational floating-base DoFs, i.e., the inclination in the sagittal and the frontal plane, is the primary aim of the controller described in this section. Its linearization corresponds to the transfer function $\boldsymbol{C}_{\text{bl}}(s)$ in Figure 3.9, which defines the outer loop of the feedback control system.

The floating-base rotation vector ${}_U\boldsymbol{\vartheta}_{\text{FB}}$ and its angular velocity ${}_U\boldsymbol{\omega}_{\text{FB}}$ are measured indirectly via the difference between the planned orientation of the torso in the planner's world FoR and the actual rotation vector obtained from the IMU at the torso, see Section 3.6.1. The control scheme is motivated from the corresponding floating-base dynamics in Equation (3.3):

$$
\boldsymbol{M}_{\vartheta_{xy}}\, {}_U\ddot{\boldsymbol{\vartheta}}_{\text{FB},xy} + \boldsymbol{h}_{\vartheta_{xy}} = -\boldsymbol{M}_{\vartheta_{xy},\text{J}}\, \ddot{\boldsymbol{q}}_{\text{J,d}} + {}_U\boldsymbol{T}^{\text{CoM}}_{xy}, \quad (4.9)
$$

where ${}_U\boldsymbol{\vartheta}_{\text{FB},xy} = [1,\, 1,\, 0]^T\, {}_U\boldsymbol{\vartheta}_{\text{FB}} \in \mathbb{R}^2$ denotes the floating-base inclination in the sagittal and frontal plane of the upper-body FoR, ${}_U\boldsymbol{T}^{\text{CoM}}_{xy}$ the corresponding commanded contact torques

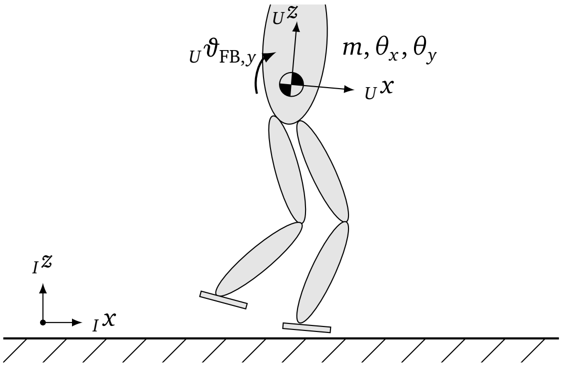


Figure 4.1: Illustration of the simplified floating-base dynamics in the sagittal plane used to derive an inclination controller. A single mass at the location of the CoM with rotational inertia is considered. The y-component of the floating-base inclination in the upper-body FoR is defined by $_U\vartheta_{\mathrm{FB},y}$.

at the CoM, and $\boldsymbol{M}_{\boldsymbol{\vartheta}_{xy}}$, $\boldsymbol{h}_{\boldsymbol{\vartheta}_{xy}}$ describe the system dynamics. Note that $_U\boldsymbol{\vartheta}_{\mathrm{FB},xy}$ is a rotation vector and the x- and y-components of this vector thus equal the projected angles in the frontal and sagittal plane respectively. The total torque on the system can be separated in the ideal (planned) torques $_U\boldsymbol{T}^{\mathrm{CoM}}_{xy,\mathrm{id}}$ and a modification $_U\Delta\boldsymbol{T}^{\mathrm{CoM}}_{xy}$. The same applies to the desired joint accelerations $\ddot{\boldsymbol{q}}_{\mathrm{J,d}}$, which yields:

$$\boldsymbol{M}_{\boldsymbol{\vartheta}_{xy}}\, {}_U\ddot{\boldsymbol{\vartheta}}_{\mathrm{FB},xy} + \boldsymbol{h}_{\boldsymbol{\vartheta}_{xy}} = -\boldsymbol{M}_{\boldsymbol{\vartheta}_{xy},\mathrm{J}}\, \Delta\ddot{\boldsymbol{q}}_{\mathrm{J,d}} - \boldsymbol{M}_{\boldsymbol{\vartheta}_{xy},\mathrm{J}}\, \ddot{\boldsymbol{q}}_{\mathrm{J,d,id}} + {}_U\boldsymbol{T}^{\mathrm{CoM}}_{xy,\mathrm{id}} + {}_U\Delta\boldsymbol{T}^{\mathrm{CoM}}_{xy}. \tag{4.10}$$

Assuming sufficient accuracy of the reduced planning model in the WPG,

$${}_U\boldsymbol{T}^{\mathrm{CoM}}_{xy,\mathrm{id}} - \boldsymbol{M}_{\boldsymbol{\vartheta}_{xy},\mathrm{J}}\, \ddot{\boldsymbol{q}}_{\mathrm{J,d,id}} \approx \boldsymbol{0} \tag{4.11}$$

holds since the planner does not intentionally create a non-zero floating-base inclination[1]. In addition, $\boldsymbol{M}_{\boldsymbol{\vartheta}_{xy},\mathrm{J}}\, \Delta\ddot{\boldsymbol{q}}_{\mathrm{J,d}} \approx \boldsymbol{0}$ is assumed, i.e., the influence of joint accelerations caused by trajectory modifications of the ideal plan are treated as disturbances on the system. If further $_U x$ and $_U y$ describe main axes of inertia, the full floating-base dynamics reduces to separate (nonlinear) inverted pendulum models:

$$\begin{bmatrix} \theta_{xx} & 0 \\ 0 & \theta_{yy} \end{bmatrix}^{\mathrm{CoM}} {}_U\ddot{\boldsymbol{\vartheta}}_{\mathrm{FB},xy} - mgl \begin{bmatrix} \sin({}_U\vartheta_{\mathrm{FB},x}) \\ \sin({}_U\vartheta_{\mathrm{FB},y}) \end{bmatrix} = {}_U\Delta\boldsymbol{T}^{\mathrm{CoM}}_{xy}, \tag{4.12}$$

with the rotational inertia θ_{xx}, θ_{yy}, the robot's mass m, and the shortest distance l between the CoM and the pivot of the inclination (the TCP in the ground plane), see Figure 4.1.

4.2.1 Basic Control Scheme

The control objective is to minimize the floating-base inclination $\boldsymbol{\vartheta}_{\mathrm{FB},xy}$ following a predefined second-order dynamics:

$$\boldsymbol{\zeta}_v \odot \boldsymbol{\zeta}_v \odot {}_U\ddot{\boldsymbol{\vartheta}}_{\mathrm{FB},xy} + 2\boldsymbol{d}_v \odot \boldsymbol{\zeta}_v \odot {}_U\dot{\boldsymbol{\vartheta}}_{\mathrm{FB},xy} + {}_U\boldsymbol{\vartheta}_{\mathrm{FB},xy} = \boldsymbol{0}, \tag{4.13}$$

[1] The floating-base inclination is defined as the difference between the planner's ideal world and the actual inertial FoR. Thus, differences are caused by errors in the planner's model and external disturbances.

with time constant and damping vectors $\boldsymbol{\zeta}_v$, $\boldsymbol{d}_v$. By solving Equation (4.13) for ${}_U\ddot{\boldsymbol{\vartheta}}_{\mathrm{FB},xy}$ and inserting into Equation (4.12), a PD-type controller with nonlinear compensation of gravity

$$ {}_U\Delta\boldsymbol{T}^{\mathrm{CoM}}_{xy} = \begin{bmatrix}\theta_{xx}\\ \theta_{yy}\end{bmatrix}^{\mathrm{CoM}} \odot\left(-\hat{\zeta}_v\odot\hat{\zeta}_v\odot{}_U\boldsymbol{\vartheta}_{\mathrm{FB},xy} - 2\boldsymbol{d}_v\odot\hat{\zeta}_v\odot{}_U\dot{\boldsymbol{\vartheta}}_{\mathrm{FB},xy}\right) - mgl\begin{bmatrix}\sin({}_U\vartheta_{\mathrm{FB},x})\\ \sin({}_U\vartheta_{\mathrm{FB},y})\end{bmatrix} \tag{4.14} $$

is obtained. Introducing the gain vectors

$$ \boldsymbol{G}_v = \begin{bmatrix}\theta_{xx}\\ \theta_{yy}\end{bmatrix}^{\mathrm{CoM}} \odot\hat{\zeta}_v\odot\hat{\zeta}_v \qquad\qquad \boldsymbol{G}_\omega = 2\begin{bmatrix}\theta_{xx}\\ \theta_{yy}\end{bmatrix}^{\mathrm{CoM}} \odot\boldsymbol{d}_v\odot\hat{\zeta}_v, \tag{4.15} $$

and using the relation[2] ${}_U\dot{\boldsymbol{\vartheta}}_{\mathrm{FB},xy} \approx {}_U\boldsymbol{\omega}_{\mathrm{FB},xy}$ simplifies the control law to

$$ {}_U\Delta\boldsymbol{T}^{\mathrm{CoM}}_{xy} = -\boldsymbol{G}_v\odot{}_U\boldsymbol{\vartheta}_{\mathrm{FB},xy} - \boldsymbol{G}_\omega\odot{}_U\boldsymbol{\omega}_{\mathrm{FB},xy} - mgl\begin{bmatrix}\sin({}_U\vartheta_{\mathrm{FB},x})\\ \sin({}_U\vartheta_{\mathrm{FB},y})\end{bmatrix}. \tag{4.16} $$

For the distance l, the planned (ideal) vertical height of the CoM relative to the ground (TCPs) is used. The inclination controller directly modifies the ideal total CoM wrench to

$$ {}_W\boldsymbol{\Lambda}^{\mathrm{CoM}}_{\mathrm{bl}} = {}_W\boldsymbol{\Lambda}^{\mathrm{CoM}}_{\mathrm{id}} + \begin{bmatrix}\boldsymbol{0}\\ {}_W\boldsymbol{A}_U\begin{pmatrix}{}_U\Delta\boldsymbol{T}^{\mathrm{CoM}}_{xy}\\ 0\end{pmatrix}\end{bmatrix} = \begin{bmatrix}{}_W\boldsymbol{F}^{\mathrm{CoM}}_{\mathrm{id}}\\ \underbrace{{}_W\boldsymbol{T}^{\mathrm{CoM}}_{\mathrm{id}} + {}_W\boldsymbol{A}_U\begin{pmatrix}{}_U\Delta\boldsymbol{T}^{\mathrm{CoM}}_{xy}\\ 0\end{pmatrix}}_{{}_W\boldsymbol{T}^{\mathrm{CoM}}_{\mathrm{bl}}}\end{bmatrix}, \tag{4.17} $$

with ${}_W\boldsymbol{A}_U$ calculated from the desired orientation of the upper body FoR. The approach is a PD-type control scheme with additional gravity compensation, which is commonly used in related work, see [104, 226] for example. Compared to the previous implementation on LOLA, the nonlinear compensation of gravitational torques is added.

4.2.2 The Effect of Structural Resonances on Inclination Control

The plant of the inclination controller depends on the wrench distribution (Section 4.3), the contact force controller (Chapter 5), the actual contact, and the mechanical structure of the robot between the feet and the mounting of the IMU at the torso, see Figures 3.1 and 3.9. The control cascade requires the inclination controller to have significantly slower open-loop dynamics than the inner contact force control loop to preserve stability. Due to uncertainty in the properties of the contact, the gains $\boldsymbol{G}_v$ and $\boldsymbol{G}_\omega$ must be chosen carefully.

However, experience with LOLA has shown that the mechanical robot structure's flexibility imposes a far more significant performance limitation. The discrete nature of footsteps excites the system's mechanical eigenfrequencies, causing vibrations of the IMU mounting. Primarily via the sensed angular velocities, ${}_U\boldsymbol{\omega}_{\mathrm{FB},xy}$, the vibrations are fed into the control loop and strongly limit the maximum value for $\boldsymbol{G}_\omega$. The relation between the mechanical eigenfrequencies of the structure and resonances observed in IMU measurements was confirmed with an experimental modal analysis of LOLA [16]. A redesign of LOLA's torso intended to increase the structural eigenfrequencies with a stiffer design to reduce their impacts on the control loops [267]. However, it is generally difficult to significantly improve the eigenfrequencies as the reachable specific modulus is limited; only an increase of the eigenfrequencies by ≈ 2 Hz has been achieved [17].

[2] For this 2D case, the assumption holds for small angles $\tan({}_U\boldsymbol{\vartheta}_{\mathrm{FB},xy}) \approx {}_U\boldsymbol{\vartheta}_{\mathrm{FB},xy}$, see [273].

Frequency [Hz]	Sharpness Q	Damping d
9.3	1.17	0.1353
17.5	2.2	0.0227
30.0	3.77	0.04194

Table 4.1: Notch filter parameters for the inclination controller.

To analyze the effects of the structure on the inclination control loop, the estimated transfer function $\boldsymbol{P}_{\text{structure}}(s)$ from desired torques at the CoM to the floating-base inclination is used, see Section 3.8.1. The combined open-loop transfer function for inclination control Equation (4.16) and the estimated plant is depicted in Figure 4.2. For the raw plant transfer function estimates, see Appendix C.

The floating-base modes of the system are located at low frequencies $< 3\,\text{Hz}$ and need to be stabilized by the inclination controller. Especially in the x direction, the phase drops significantly at 3 Hz. Thus, the resonances above 9 Hz require relatively low inclination controller gains to preserve closed-loop stability. However, this reduces the crossover frequency and disturbance rejection for the critical range around the floating-base eigenfrequencies. In the following, several methods for the suppression of structural resonances are described and compared.

4.2.3 Filter-Based Resonance Rejection

One approach to deal with the structural resonances is to shape the open-loop transfer function using frequency-domain filters. Two approaches with different filter types are considered in the following. The first approach uses the low pass filter $\text{lpf2}(T = \frac{1}{2\pi\, 7.0\,\text{Hz}}, d = 0.65)$ (Appendix A.3) to attenuate all significant resonances above 7 Hz. The design effort for this filter is low. However, it creates an additional phase shift in the low-frequency ranges, limiting the crossover frequency.

The second approach is based on three notch filters (Appendix A.3) for the highest eigenfrequencies and a second-order low pass $\text{lpf2}(T = \frac{1}{2\pi\, 26.0\,\text{Hz}}, d = 0.65)$ with 26 Hz cutoff frequency. In contrast to the low pass, the notches affect the open-loop transfer function only in the resonances' vicinity. However, the robot's structural resonances likely depend on its pose, which changes during walking. The antiresonances of the notches must, therefore, be chosen wide enough to tolerate these uncertainties. Unfortunately, plant identification in different poses is difficult, and the choice of parameters is thus solely based on experience. An advantage is that the eigenfrequencies are linearly dependent on the square root of the stiffness per mass, which causes low sensitivity to mass (inertia) changes at higher frequencies. Also, the small differences between eigenfrequencies of the old and revamped torso design [17, 267], and similar results for robot manipulators [15] indicate low sensitivity of the eigenfrequencies to changes of the robot's pose. However, the eigenfrequencies might — depending on the pose — be either visible in the x- or y direction of the inclination control plant. Thus, the same filters are used for both directions. The parameters for the notch filters are listed in Table 4.1.

Both approaches' filter transfer functions are depicted in Figure 4.3. The floating-base angles ${}_{U}\boldsymbol{\vartheta}_{\text{FB},xy}$ and angular velocities ${}_{U}\boldsymbol{\omega}_{\text{FB},xy}$ are calculated from IMU data, which uses a fiber-optic gyroscope and physically measures velocities. Due to internal filter mechanisms of the IMU and the low amplitude of vibrations, resonances are only visible in the angular velocity signal and not the integrated angles themselves. Thus, the filters are applied to the angular velocity signals only.

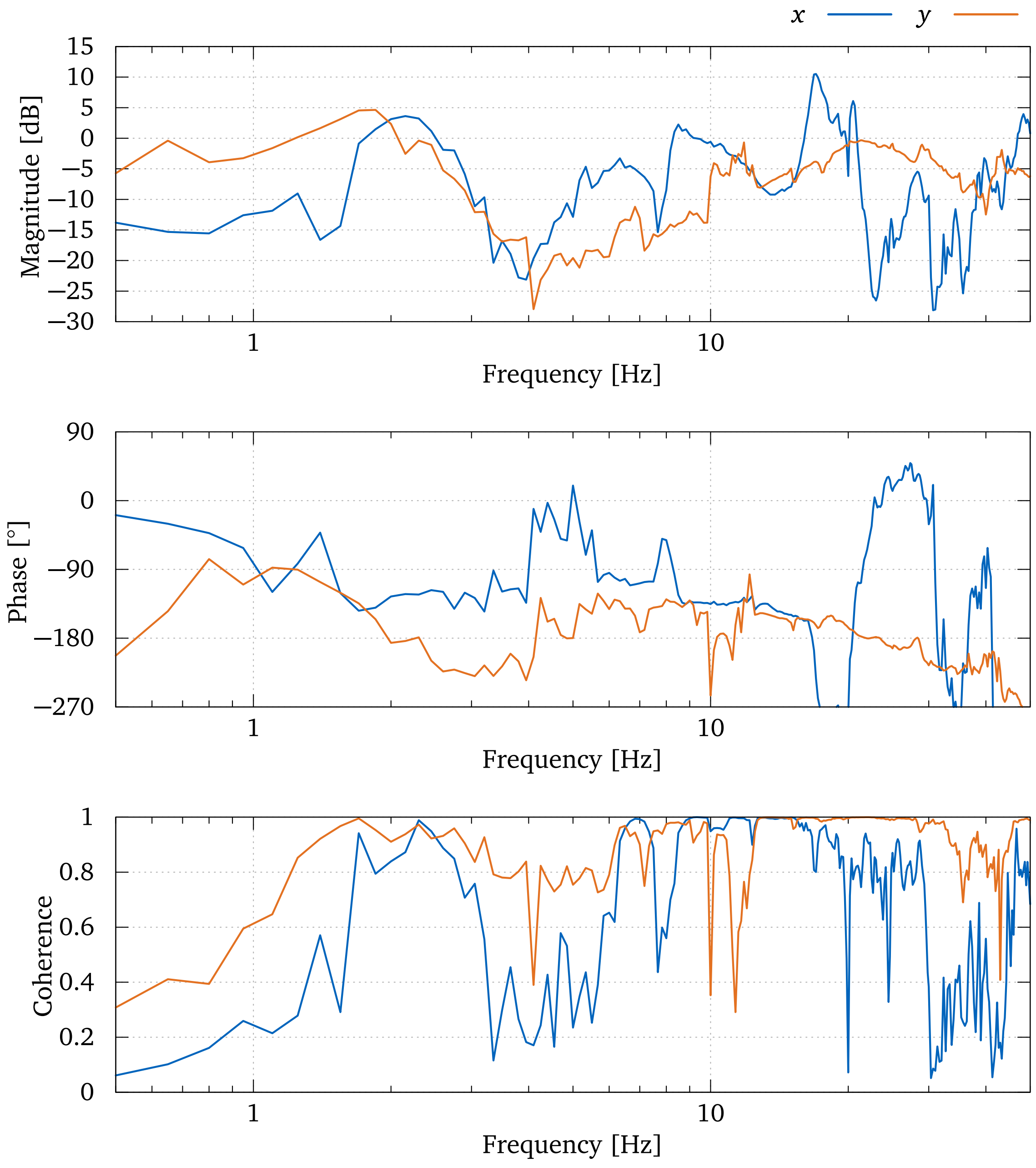


Figure 4.2: Estimated transfer function for the open loop of the inclination controller, consisting of the plant $\boldsymbol{P}_{\text{structure}}(s)$ and the controller Equation (4.16) for $\boldsymbol{G}_v = 700\,{}^{\mathrm{Nm}}\!/\!_{\mathrm{rad}} \begin{bmatrix}1 & 1\end{bmatrix}^T$, $\boldsymbol{G}_\omega = 175\,{}^{\mathrm{Nms}}\!/\!_{\mathrm{rad}} \begin{bmatrix}1 & 1\end{bmatrix}^T$. Both rotational directions around the x- and y-axis are depicted. The closed-loop dynamics of the contact controller is considered ideal (neglected).

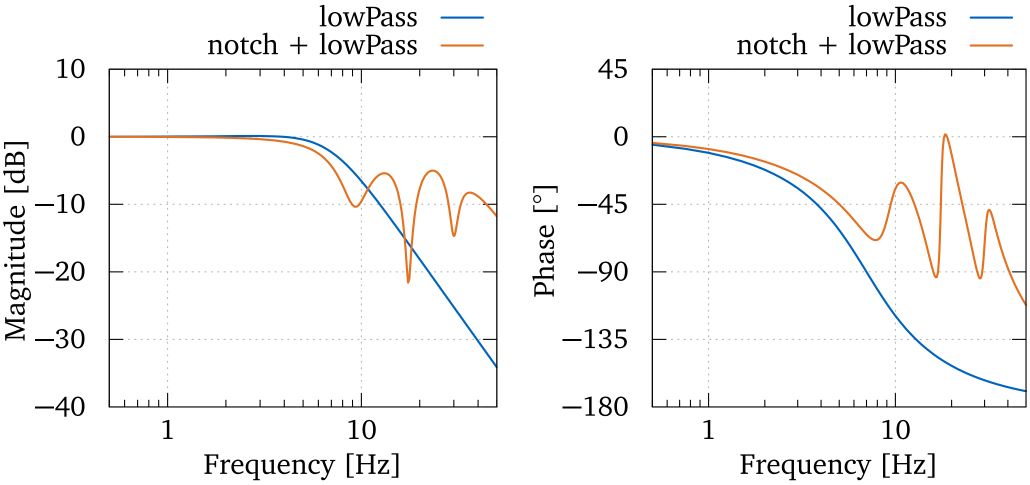


Figure 4.3: Transfer function of the low and notch filter approaches to suppress structural resonances in the inclination control loop. The filters are applied on the floating-base angular velocities of both directions.

4.2.4 Time-Domain Passivity Control

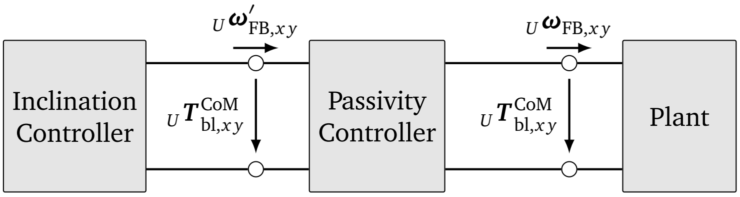
A system is passive if and only if the energy flowing into the system is at all times greater than the energy flowing out of the system, i.e., the system does not generate energy. In the following, mechanical one-port systems with input $\boldsymbol{u}$ and output $\boldsymbol{y}$ are considered, where $\boldsymbol{u}^T\boldsymbol{y} > 0$ indicates energy is flowing into the system. This leads to the following condition for passivity:

$$\int_0^t \boldsymbol{u}(\tau)^T\boldsymbol{y}(\tau)\,\mathrm{d}\tau + E_0 \geq 0, \quad \forall t \geq 0, \tag{4.18}$$

where $\boldsymbol{u}$, $\boldsymbol{y}$ is a force/velocity port pair, and E_0 the initially stored energy. Passivity is a sufficient condition for stability of both linear and nonlinear systems. For physical systems, a collocated input–output pair satisfying Equation (4.18) must exist due to energy conservation.

The concept of passivity is transferred to the inclination controller in the presence of unmodeled structural dynamics. The power port between controller and plant is defined by the desired torque at the CoM ${}_U\boldsymbol{T}^{\mathrm{CoM}}_{\mathrm{bl},xy}$ and the corresponding velocity of the floating-base ${}_U\boldsymbol{\omega}_{\mathrm{FB},xy}$. The control law Equation (4.16) may be interpreted as a spring–damper system, and is, therefore, always passive with respect to this power port. However, the controlled robot is not necessarily passive because the power port is not a collocated input–output pair: the desired torques from the controller output do not act at the IMU but the feet. This means the structural resonances may “generate” energy, which leads to the observed instability of the inclination controller.

Hannaford and Ryu [98] proposed the concept of a time-domain Passivity Controller (PC), which observes a system’s energy ports and adds additional damping on the ports when energy is generated. The concept wraps the original system with a new power port, for which the system is passive, see Figure 4.4. The approach is, for example, used in teleoperation or haptic interaction tasks where unknown and varying time-delays occur, because these are challenging to model. The method is purely based on energy considerations and does not require any information on the plant’s inner structure. The inclination control plant follows a force-to-velocity causality, and thus a parallel PC is used [98].



Inclination Controller
Passivity Controller
Plant
$_U\boldsymbol{\omega}'_{\mathrm{FB},xy}$
$_U\boldsymbol{\omega}_{\mathrm{FB},xy}$
$_U\boldsymbol{T}^{\mathrm{CoM}}_{\mathrm{bl},xy}$
$_U\boldsymbol{T}^{\mathrm{CoM}}_{\mathrm{bl},xy}$

Figure 4.4: Total network of the control loop. The time-domain passivity controller exposes the new passive port $({}_U\boldsymbol{T}^{\mathrm{CoM}}_{\mathrm{bl},xy}, {}_U\boldsymbol{\omega}'_{\mathrm{FB},xy})$ to the inclination controller by adding damping.

The power port of the system is observed with a passivity observer, which updates the stored energy estimate E_{ob} in every time-step n. The inclination controller is fed with the updated angular velocity

$$ {}_U\boldsymbol{\omega}'_{\mathrm{FB},xy} = {}_U\boldsymbol{\omega}_{\mathrm{FB},xy} + \alpha\, {}_U\boldsymbol{T}^{\mathrm{CoM}}_{\mathrm{bl},xy}, \tag{4.19} $$

with the time-varying damping factor $\alpha > 0$. The required damping is calculated from the observed energy of the system:

$$ \alpha = \begin{cases} \min\left(\frac{-E_{\mathrm{ob}}}{||\boldsymbol{T}^{\mathrm{CoM}}_{\mathrm{bl},xy}||^2}, \alpha_{\max} \right) & \text{for} \quad E_{\mathrm{ob}} < 0 \\ 0 & \text{for} \quad E_{\mathrm{ob}} \geq 0 \end{cases} \tag{4.20} $$

with a maximum damping factor $\alpha_{\max} > 0$. The energy estimate for the system is given by

$$ E_{\mathrm{ob}}[n] = \min(E_{\mathrm{ob}}[n-1] + \Delta t_{\mathrm{cont}}\, \dot{E}[n], E_{\max}) \tag{4.21} $$

with the power flow

$$ \begin{aligned} \dot{E}[n] = & \underbrace{{}_U\boldsymbol{T}^{\mathrm{CoM}}_{\mathrm{bl},xy}[n]^T\, {}_U\boldsymbol{\omega}_{\mathrm{FB},xy}[n]}_{\dot{E}_1} + \underbrace{(\boldsymbol{G}_\omega \odot {}_U\boldsymbol{\omega}'_{\mathrm{FB},xy}[n-1])^T\, {}_U\boldsymbol{\omega}'_{\mathrm{FB},xy}[n-1]}_{\dot{E}_2} \\ & + \underbrace{\alpha[n-1]\, ||\boldsymbol{T}^{\mathrm{CoM}}_{\mathrm{bl},xy}[n]||^2}_{\dot{E}_3}. \end{aligned} \tag{4.22} $$

$\dot{E}_1$ describes the power flow at the plant's port. $\dot{E}_2$ considers the known power dissipation of the inclination controller, which depends on the velocity in the last step and the gain for angular velocities. Without this term, power dissipated in the inclination controller and flowing out of the plant would lead to an increase in damping. $\dot{E}_3$ considers the dissipation generated by the PC in the last time step. The maximum energy the system may generate over a contiguous set of time-steps before the PC adds damping to the port is defined by $E_{\max}$.

4.2.5 Experimental Comparison

Experiments on LOLA are conducted to analyze and compare the performance of the two filter-based and the passivity-based approach. In the test setup, the robot is commanded to step in place with inclination control gains $\boldsymbol{G}_v = 700\,\mathrm{Nm/rad}\begin{bmatrix}1 & 1\end{bmatrix}^T$, $\boldsymbol{G}_\omega = 175\,\mathrm{Nms/rad}\begin{bmatrix}1 & 1\end{bmatrix}^T$, which lead to significant amplification of the structural resonances in the control loop. The reference controller and the PC use an additional prefilter lpf2$(T = \frac{1}{2\pi\, 26.0\,\mathrm{Hz}}, d = 0.65)$ with 26 Hz cutoff frequency to attenuate high-frequency noise. The PC is parametrized with $E_{\max} = 0$, and $\alpha_{\max} = 0.003\,\mathrm{rad/Nms}$.

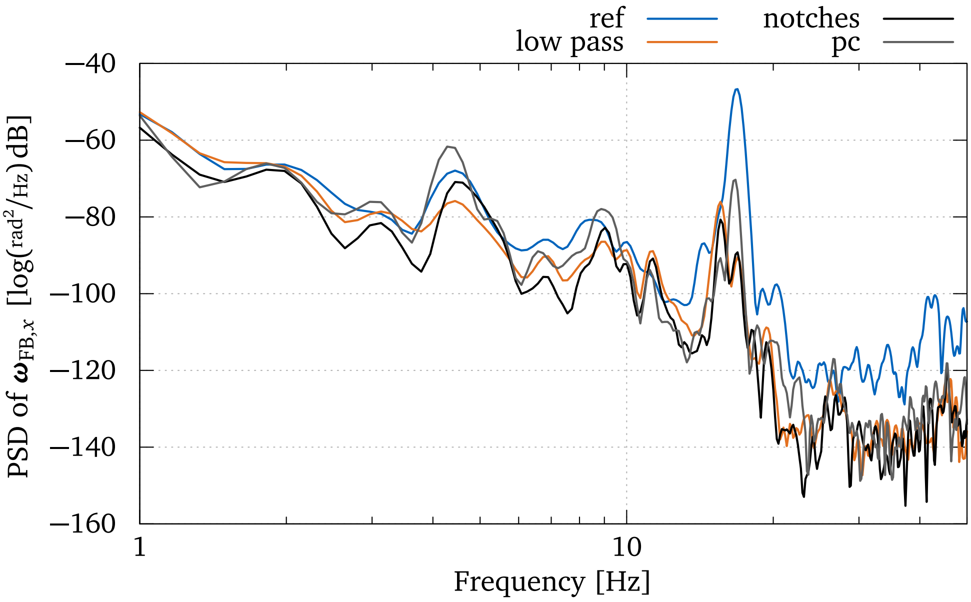


Figure 4.5: Power Spectral Density (PSD) of the floating-base angular velocity signal for different vibration damping strategies in a walking experiment with Lola.

In the standard pose, the structural resonances are dominant in the rotations around the x-axis, see Figure 4.2. Thus, only this direction is considered in the following. Figure 4.5 shows the PSD of the measured floating-base angular velocity around the x-axis for all three controllers and the reference. Especially the high peak of vibrations at 17 Hz in the reference is effectively reduced by all three approaches. The low-pass filter and the notch filters yield comparable results over the whole frequency range. The results for the PC are close; however, it does not reach the same vibration suppression level with the used parametrization. Significant differences between the filter-based approaches and the passivity-based approach are only visible in the floating-base inclination, see Figure 4.6. With the PC but otherwise identical parameters of the inclination controller, significantly higher oscillations of the robot's floating-base occur. In fact, the disturbance rejection gets even worse for higher values of $\alpha_{\max}$. A variation of this parameter has shown that the PC is not Pareto optimal with respect to floating-base inclination and vibration suppression.

4.2.6 Discussion in the Context of Related Work

The Relation to DCM Tracking Control A common approach to balance biped robots is the DCM, see also Section 3.1.2. The concept can be used to generate a reference trajectory for the CoM. Furthermore, DCM tracking control provides a control scheme to track this trajectory based on sensor data. For the tracking control, the CoM reference trajectory from any planner may be used.

In the following, the extension of the DCM to 3D is considered [67]. For the approach, point feet are assumed, and the robot is approximated with a single point mass. Due to the point-mass model, a zero rate of angular momentum is assumed.

The DCM approach uses the fact that it is not required to consider the second-order dynamics of the CoM to track its position in space. Instead, the DCM ζ with time constant b is introduced:

$$\zeta = \boldsymbol{r}_{W,CoM} + b\dot{\boldsymbol{r}}_{W,CoM}. \tag{4.23}$$

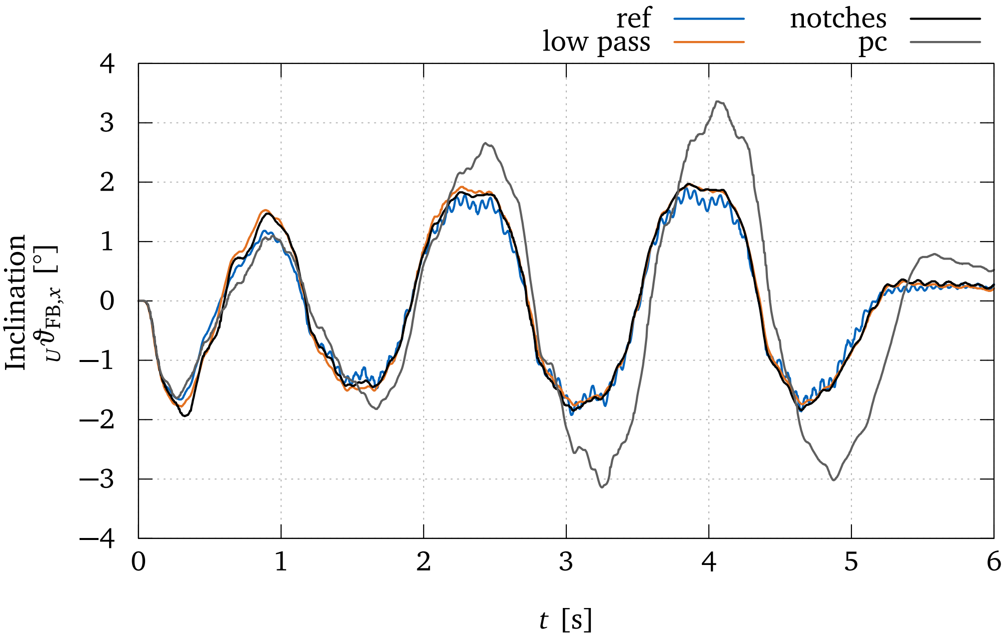


Figure 4.6: Floating-base inclination of LOLA in the frontal plane for walking on the spot with different vibration damping controllers. Interestingly, good vibration damping with the PC leads to worse stabilization of the floating base's unstable dynamics. In contrast, the filter-based methods produce similar results.

With this substitution, the tracking task simplifies to the first-order dynamics of ζ as it is known that $\boldsymbol{r}_{W,CoM}$ follows ζ with stable dynamics. The variable b is a design parameter related to the (average) height of the CoM above ground. Based on these considerations, the DCM tracking controller

$$\boldsymbol{F}_{CoM} = \frac{m}{b^2}(\boldsymbol{r}_{W,CoM} + b\dot{\zeta}_{\mathrm{d}} - \zeta + k\,b(\zeta_{\mathrm{d}} - \zeta)) \tag{4.24}$$

calculates the force on the CoM that tracks a reference DCM trajectory ζ_{d} using the robot's mass m and a control gain k [67].

Re-substituting Equation (4.23) into Equation (4.24) yields

$$\boldsymbol{F}_{CoM} = \underbrace{m\ddot{\boldsymbol{r}}_{W,CoM}}_{\boldsymbol{F}_{CoM,\mathrm{id}}} + \frac{m}{b}\left[k(\boldsymbol{r}_{W,CoM,\mathrm{d}} - \boldsymbol{r}_{W,CoM}) + (1+kb)(\dot{\boldsymbol{r}}_{W,CoM,\mathrm{d}} - \dot{\boldsymbol{r}}_{W,CoM})\right], \tag{4.25}$$

which is basically a PD-type controller on the CoM position and velocity, with feedforward term $\boldsymbol{F}_{CoM,\mathrm{id}}$ for the planned contact forces. In contrast, the balance controller described in Section 4.2 tracks the floating-base inclination of the robot, i.e., an angular difference to the planned reference. Naturally, the CoM position error and the floating-base inclination are related, although not necessarily linearly. A further difference is the zero rate of angular momentum assumption from the DCM tracking controller — the desired ground reaction forces always point to the CoM. In contrast, the concept from Section 4.2 does only use torques at the CoM to balance the robot and track the reference trajectory. It assumes that rotational disturbances rather than translational ones cause deviations of the CoM trajectory.

In [43], for example, the DCM tracking controller is used to modify a planned ZMP location. In general, this leads to additional torque around the CoM, i.e., it violates the zero rate of angular momentum assumption of the DCM formulation. The approach is a viable solution from a control perspective, though, and even closer to the implementation from Section 4.2. In this case, the only difference is the approximation of a rotational disturbance via the error in the CoM location.

Structural Resonance Rejection The greatest advantage of the PC approach — not requiring any knowledge on the plant — is also its greatest disadvantage. Operating in the time-domain, the PC damps all frequencies equally, including those belonging to the floating base's unstable modes. A closer look at Equations (4.16) and (4.19) shows that the PC effectively reduces the absolute value of the angular velocity fed into the inclination controller when it adds damping. Experiments have shown that a reduction of the damping gain ($\boldsymbol{G}_\omega = 114\,\mathrm{Nms/rad}\begin{bmatrix}1 & 1\end{bmatrix}^T$) for the reference (just prefilter) leads to a similar behavior of the system concerning the inclinations. Nevertheless, the PC provides better damping of the structural resonances than just reducing the gains of the reference controller. The concept of the time-domain PC originates in teleoperation, where *time-varying* plants with deadtimes are common. Of course, the approach does work for other plants and can passivate a system without further knowledge of the inner workings. This simplicity, however, comes at the price of a reduced tracking performance of the closed loop. Typical robot plants do not contain significant deadtimes[3] and are not *per se* time-varying but depend on the robot's configuration. The results show that even for the moving structural eigenfrequencies, loop shaping via notch filters provides a better overall performance.

There is no related work on analyzing time-domain passivity control for vibration suppression on biped robots to the author's best knowledge. The technique has, however, been applied for different purposes in the field. Kim, Lee, Ryu, and Kim [160] use a PC for a landing force control of a biped robot. In [105], the approach is used for a compliant balance controller of a biped to make the system robust to soft and moving ground. In contrast to the acceleration-level (torque-based) contact control in [105], the force control approaches presented in Chapter 5 provide robust stability via velocity-level feedback; an additional passivation is not required.

Both the low-pass and the notch-based variants seem adequate solutions to deal with the structural resonances. Because the notches lead to less phase shift at low frequencies and the variation of the eigenfrequencies seems unproblematic, they are the preferred approach used in this thesis. Notch filters are a standard way in the literature to suppress mechanical vibrations [243]. In contrast to phase-based vibration canceling techniques, the notch filters can be applied to non-collocated control loops (as is the case here). For all subsequent experiments in this thesis, the new default values $\boldsymbol{G}_v = 725\,\mathrm{Nm/rad}\begin{bmatrix}1 & 1\end{bmatrix}^T$, $\boldsymbol{G}_\omega = 250\,\mathrm{Nms/rad}\begin{bmatrix}1 & 1\end{bmatrix}^T$ are used for the parametrization of the inclination controller.

In [13], the filter-based methods are compared to a model-based state observer with an optimal controller. The work assumes the structural dynamics are known beforehand and evaluates the approaches on a simulation model that considers some flexibility in LOLA's structures. Interestingly, the differences in the achieved performance are minor, while the design- and tuning effort for the model-based approaches are significantly higher.

There are only a few publications on vibration suppression on humanoid robots. In [138], a model of structural resonances is identified from estimated transfer functions and used to design a vibration-damping controller based on observer design. However, this requires identifying state-space (or other whitebox) plant models from the transfer function data, which requires high effort, especially for MIMO systems. In contrast, the notch filters can be designed right from the raw transfer function data. Another example of vibration damping is described by [157] and uses an additional accelerometer to damp vibrations. In [329], a feedforward control method is used to suppress vibrations of flexible robot parts.

Although structural vibrations seem to limit robots' performance in many cases, only a few publications explicitly deal with this problem. This may also be because the effects of

[3] The sample frequency is usually a lot faster than the dynamics of the robot. The dynamics are dominated by the inertia of the joint drive modules (high gear ratio) and the robot's links. Friction effects can effectively be mitigated by appropriate design of the joint-level control loops.

the structure on these complex systems are not fully understood. For example, the oscillation problems reported in [66] may be caused by the robot's structure, and it might be necessary to reveal the propagation paths of the vibrations to facilitate effective countermeasures. When structural resonances are not explicitly considered in the control loop design, they are often suppressed by choosing suboptimal parameters for observers or control loop gains, making the closed loops slower than necessary. Also, to the author's best knowledge, the only publication describing an experimental modal analysis of a humanoid deals with Lola [16, 17]. These tests are essential to prove that the structure causes the resonances identified in the control loops.

4.3 Optimal Contact Wrench Distribution

The desired contact wrench $\Lambda_{\text{bl}}^{\text{CoM}}$ from the inclination controller is realized by contact force controllers, one for each end effector, see Chapter 5. When multiple end effectors (feet, hands) are in contact with the environment, the distribution of wrenches to the individual contact controllers becomes redundant. In previous work on Lola, the wrench distribution to the feet in the double support phase is achieved via a heuristic, see Section 4.1. However, hand contacts are not considered by this method. In the following, an optimization-based approach is presented, which enables multi-contact with the hands and feet, and considers (different) friction at the end effectors. To make the algorithm computationally efficient and compatible with hard real-time requirements, the problem is formulated as a convex QP. The wrench distribution module is part of the inclination control loop and its linearization is depicted with triangle blocks in Figure 3.9.

4.3.1 Prerequisites

The constraint of limited friction sets a relation between the tangential forces on an end effector ${}_E\boldsymbol{F}_{e,t}$, the friction coefficient μ_e, and force ${}_E\boldsymbol{F}_{e,n}$ normal to the contact surface:

$$||{}_E\boldsymbol{F}_{e,t}|| \leq \mu_e ||{}_E\boldsymbol{F}_{e,n}||. \tag{4.26}$$

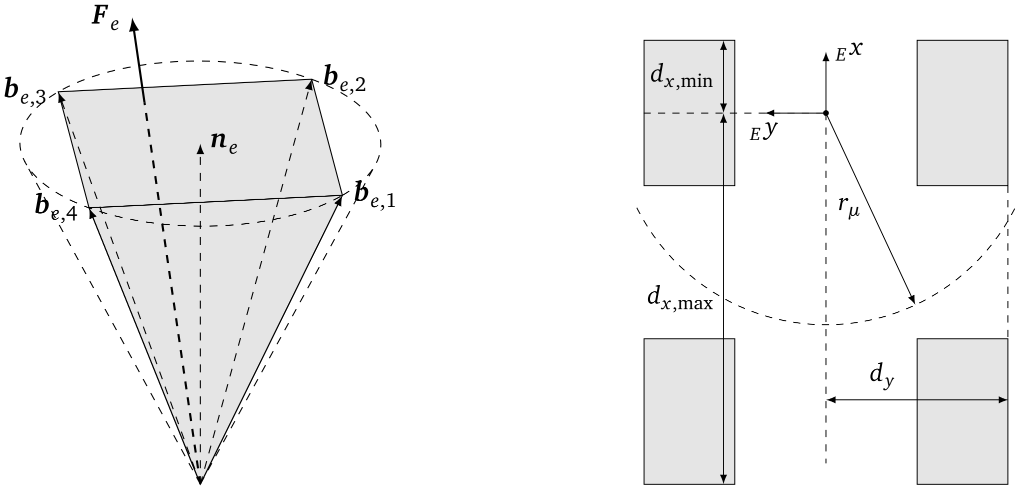
Unfortunately, Equation (4.26) is a nonlinear relation and must be linearized first to obtain a convex problem. The friction cone is commonly approximated via regular pyramids [119, 166, 169]. Based on the contact surface normal $\boldsymbol{n}_e$ and the friction coefficient μ_e, a set of p unit basis vectors $\boldsymbol{b}_{e,i} \in \mathbb{R}^3$, $i = 1, \ldots, p$ is calculated for each end effector. The surface normals $\boldsymbol{N}$ from the WPG are used to generate the basis vectors. These vectors geometrically approximate the friction cone, i.e., the set of possible force directions compatible with the friction constraint, see Figure 4.7a. The contact force vector is then given by

$${}_E\boldsymbol{F}_e = \sum_{i=1}^{p} {}_E\boldsymbol{b}_{e,i}\,\varphi_{e,i}, \tag{4.27}$$

where $\varphi_{e,i} > 0$ denote the coordinates in the new basis. The relation can be written in matrix-vector form, which yields the more compact formulation

$${}_E\boldsymbol{F}_e = {}_E\boldsymbol{B}_e\,\boldsymbol{\phi}_e, \tag{4.28}$$

with ${}_E\boldsymbol{B}_e \in \mathbb{R}^{3\times p}$, $\boldsymbol{\phi}_e \in \mathbb{R}^p$. The matrix ${}_E\boldsymbol{B}_e$ is calculated from the contact normal and friction coefficient of an end effector and is defined in the corresponding end-effector FoR.



(a) Friction cone and linearized representation with $p = 4$ basis vectors of a regular pyramid.

(b) Foot geometry distances and friction radius used for the foot torque contraint, exemplarily shown for LOLA's feet.

Figure 4.7: Friction cone model and foot geometry variables used in the unilateral contact constraints.

In order to be able to limit the contact forces in the QP, a relation for the normal contact forces as a function of the $\varphi_{e,i}$ coordinates is required. The normal force component for an end effector is

$$F_{e,n} = {}_E\boldsymbol{n}_e{}^T\,{}_E\boldsymbol{F}_e = {}_E\boldsymbol{n}_e{}^T \sum_{i=1}^{p} {}_E\boldsymbol{b}_{e,i}\,\varphi_{e,i} = \sum_{i=1}^{p} \underbrace{{}_E\boldsymbol{n}_e{}^T\,{}_E\boldsymbol{b}_{e,i}}_{\cos(\operatorname{atan}(\mu_e))}\,\varphi_{e,i} = \frac{1}{\sqrt{1+\mu_e^2}} \sum_{i=1}^{p} \varphi_{e,i}. \tag{4.29}$$

Due to the hands' spherical geometry, only point contacts with the environment are assumed, and no additional torques are considered at the hand end effectors. The position of a hand's contact point relative to the world FoR is denoted ${}_W\boldsymbol{r}_{W,h,c}$ and results from the contact surface normal, the geometry of the hand, and the location of the hand's TCP.

Additional contact torques can be realized with the feet, which must be considered in the optimization problem. The unilateral contact poses constraints on the maximum achievable contact torques, depending on the current normal (${}_E z$ direction) contact force. The torques $\boldsymbol{e}_x{}^T\,{}_E\boldsymbol{T}_f$, $\boldsymbol{e}_y{}^T\,{}_E\boldsymbol{T}_f$ in the ground plane additionally depend on the foot's length and width. The torque $\boldsymbol{e}_z{}^T\,{}_E\boldsymbol{T}_f$ around the foot plane normal is further limited by the friction coefficient and the effective friction radius

$$r_\mu = \frac{1}{A} \iint\limits_S r_i(x,y)\,\mathrm{d}S_i \tag{4.30}$$

$$|\boldsymbol{e}_z{}^T\,{}_E\boldsymbol{T}_f| \leq r_\mu\,\mu_f\,{}_E\boldsymbol{e}_z{}^T\,{}_E\boldsymbol{F}_f, \tag{4.31}$$

where A is the area of the foot and $r_i(x,y)$ denotes the radius to a surface element i relative to the origin in the foot plane. Note that this limitation is a common approximation — the yaw torque limit generally depends on the CoP location and the tangential forces [44]. However, the effect of this simplification is assumed to be low, because the yaw torques are not actively controlled and already defined by the planned WPG motion. Further, the friction coefficient or radius may simply be reduced for a stricter limit. Figure 4.7b illustrates the geometry parameters and the friction radius of LOLA's foot. The planned contact surface, i.e., either full contact of the whole foot or the toe segment is used for the calculation of the geometry parameters. The latter case is only relevant for walking on LOLA's toes, e.g. for stair climbing.

The unilateral contact constraint on a foot torque ${}_E\boldsymbol{T}_f$ is then described with

$$
{}_E\boldsymbol{D}_{f,\min}\,{}_E\boldsymbol{F}_f \leq {}_E\boldsymbol{T}_f \leq {}_E\boldsymbol{D}_{f,\max}\,{}_E\boldsymbol{F}_f, \tag{4.32}
$$

where the matrices

$$
{}_E\boldsymbol{D}_{f,\min} = \begin{bmatrix} -d_y \\ -d_{x,\min} \\ -r_\mu\,\mu_f \end{bmatrix} \boldsymbol{e}_z{}^T \qquad\qquad {}_E\boldsymbol{D}_{f,\max} = \begin{bmatrix} d_y \\ d_{x,\max} \\ r_\mu\,\mu_f \end{bmatrix} \boldsymbol{e}_z{}^T \tag{4.33}
$$

contain the relevant geometry and friction parameters. This representation is relatively compact in comparison to the use of multiple point contacts per foot. However, it is limited to rectangularly shaped feet.

4.3.2 Problem Formulation

The state vector of the optimization problem is given by

$$
\boldsymbol{Z}^T = \begin{bmatrix} \boldsymbol{\phi}_{\mathrm{RF}}^T & \boldsymbol{\phi}_{\mathrm{LF}}^T & \boldsymbol{\phi}_{\mathrm{RH}}^T & \boldsymbol{\phi}_{\mathrm{LH}}^T & {}_E\boldsymbol{T}_{\mathrm{RF}}^T & {}_E\boldsymbol{T}_{\mathrm{LF}}^T \end{bmatrix}. \tag{4.34}
$$

The static QP which finds an optimal $\boldsymbol{Z}_*$ is defined to:

$$
\begin{aligned}
\min_{\boldsymbol{Z}} \quad & \sum_{f=\{\mathrm{RF,LF}\}} \left(\boldsymbol{\phi}_f^T\,{}_E\boldsymbol{B}_f^T\,\boldsymbol{H}_F\,{}_E\boldsymbol{B}_f\,\boldsymbol{\phi}_f + {}_E\boldsymbol{T}_f^T\,\boldsymbol{H}_T\,{}_E\boldsymbol{T}_f\right) && (4.35)\\
& + \sum_{h=\{\mathrm{RH,LH}\}} \left(\boldsymbol{\phi}_h^T\,{}_E\boldsymbol{B}_h^T\,\boldsymbol{H}_H\,{}_E\boldsymbol{B}_h\,\boldsymbol{\phi}_h\right) \quad \text{(regularization)} && (4.36)\\
& + \boldsymbol{F}_{\mathrm{res}}^T\,\boldsymbol{H}_{F,\mathrm{res}}\,\boldsymbol{F}_{\mathrm{res}} + \boldsymbol{T}_{\mathrm{res}}^T\,\boldsymbol{H}_{T,\mathrm{res}}\,\boldsymbol{T}_{\mathrm{res}} \quad \text{(constraint residuals)} && (4.37)\\
\text{with} & \\
\boldsymbol{F}_{\mathrm{res}} = & {}_W\boldsymbol{F}_{\mathrm{bl}} - \sum_{e=\{\mathrm{RF,LF,RH,LH}\}} \left({}_W\boldsymbol{A}_E\,{}_E\boldsymbol{B}_e\,\boldsymbol{\phi}_e\right) \quad \text{(desired force)} && (4.38)\\
\boldsymbol{T}_{\mathrm{res}} = & {}_W\boldsymbol{T}_{\mathrm{bl}}^{\mathrm{CoM}} - \sum_{f=\{\mathrm{RF,LF}\}} \left({}_W\boldsymbol{A}_E\,{}_E\boldsymbol{T}_f + {}_W\tilde{\boldsymbol{r}}_{W,f}\,{}_W\boldsymbol{A}_E\,{}_E\boldsymbol{B}_f\,\boldsymbol{\phi}_f\right) && (4.39)\\
& - \sum_{h=\{\mathrm{RH,LH}\}} \left({}_W\tilde{\boldsymbol{r}}_{W,h,c}\,{}_W\boldsymbol{A}_E\,{}_E\boldsymbol{B}_h\,\boldsymbol{\phi}_h\right) \quad \text{(desired torque)} &&\\
\text{subject to} & \\
\varphi_{e,i} \geq & 0 \quad \forall e = \{\mathrm{RF,LF,RH,LH}\},\ i = 1,..,p \quad \text{(friction cone)} && (4.40)\\
\gamma_{\mathrm{ad},e}\,F_{e,n,\min} \leq & F_{e,n} \leq F_{e,n,\max}\,\gamma_{\mathrm{ad},e} \quad \forall e = \{\mathrm{RF,LF,RH,LH}\} \quad \text{(force limits)} && (4.41)\\
{}_E\boldsymbol{D}_{f,\min}\,{}_E\boldsymbol{B}_f\,\boldsymbol{\phi}_f \leq & {}_E\boldsymbol{T}_f \leq {}_E\boldsymbol{D}_{f,\max}\,{}_E\boldsymbol{B}_f\,\boldsymbol{\phi}_f \quad \forall f = \{\mathrm{RF,LF}\} \quad \text{(torque limits)} && (4.42)
\end{aligned}
$$

The cost Term (4.35) describes quadratic costs in the contact forces and torques of the feet, which may be weighted via the diagonal matrices $\boldsymbol{H}_F$, $\boldsymbol{H}_T$. Similarly, Term (4.36) describes costs for the hand contact forces with diagonal weighing matrix $\boldsymbol{H}_H$. Further, a deviation from the desired forces and torques is penalized via the diagonal matrices $\boldsymbol{H}_{F,\mathrm{res}}$, and $\boldsymbol{H}_{T,\mathrm{res}}$ respectively. Typically, the entries of these matrices are set to high values to ensure the residual forces and torques only become nonzero when no other solution to the wrench distribution problem exists.

Constraint Equation (4.40) enforces the unilateral contact with the environment. Equation (4.41) limits the normal contact forces to an admissible range $[F_{e,n,\min}, F_{e,n,\max}]$ and ensures end effectors are not considered when the respective load $\gamma_{\mathrm{ad},e}$ is zero (no planned contact). This allows a smooth transition between single- and double support phase, for example. Equation (4.42) limits the contact torques at the feet, refer to Equations (4.32) and (4.33).

The main objective of the wrench distribution problem is encoded in Equations (4.38) and (4.39). The sum of all contact forces — transformed to the world FoR via ${}_{W}\boldsymbol{A}_{E}$ — should equal the desired total forces from the inclination controller, see Equation (4.38). Analogously, the total torque at the CoM should match the desired torque, see Equation (4.39). The residuals of these soft constraints are minimized with high cost values to ensure these constraints are seldomly violated.

The optimization problem is solved once every time step without reuse of data from the previous run. Due to the convexity, a proper choice of cost values leads to a smooth optimal state $\boldsymbol{Z}_{*}$ over time given all input variables to the problem are smooth. Due to the inequality constraints, only C^0 smoothness may be reached. The desired contact wrenches for the end effectors result from the optimal state vector:

$$\boldsymbol{\lambda}_{f,\mathrm{d}} = \begin{bmatrix} \boldsymbol{\phi}_{f*}^{T}\,{}_{E}\boldsymbol{B}_{f}^{T} & {}_{E}\boldsymbol{T}_{f*} \end{bmatrix}^{T} \qquad \boldsymbol{\lambda}_{h,\mathrm{d}} = \begin{bmatrix} \boldsymbol{\phi}_{h*}^{T}\,{}_{E}\boldsymbol{B}_{h}^{T} & \boldsymbol{0} \end{bmatrix}^{T}. \tag{4.43}$$

4.3.3 Implementation

The linearized friction cone is parametrized with $p = 4$ for application of the wrench distribution on LOLA. The friction radius is approximated with the mean value of the radii to the four pads' center. Given that there is significant uncertainty in the friction coefficient, this simplification seems valid with a view to the overall accuracy. From experiments with the pad material, the static friction coefficient is known for different ground materials. The maximum of these coefficients $\mu_f = 0.8$ is chosen for the feet. In contrast, the material of the hands does not provide much grip with $\mu_h = 0.05$. The cost factors' parametrization is reduced to three scalars for the foot forces, torques, and hand forces. Furthermore, the hand force and torque costs are scaled relatively to foot force costs based on typical ranges of the three different quantities:

$$\boldsymbol{H}_F = h_f\,\boldsymbol{I} \qquad \boldsymbol{H}_T = h_t\,\frac{F_{f,n,\mathrm{typ}}^2}{T_{f,\mathrm{typ}}^2}\mathrm{m}^2\,\boldsymbol{I} \qquad \boldsymbol{H}_H = h_h\,\frac{F_{f,n,\mathrm{typ}}^2}{F_{h,n,\mathrm{typ}}^2}\,\boldsymbol{I}, \tag{4.44}$$

$$\text{with} \tag{4.45}$$

$$F_{f,n,\mathrm{typ}} = 600\,\mathrm{N} \qquad T_{f,\mathrm{typ}} = 100\,\mathrm{Nm} \qquad F_{h,n,\mathrm{typ}} = 200\,\mathrm{N}. \tag{4.46}$$

The residual wrench costs are set to $\boldsymbol{H}_{F,\mathrm{res}} = 5000\,\boldsymbol{I}$, and $\boldsymbol{H}_{T,\mathrm{res}} = 50000\,\boldsymbol{I}$. The torques contain the output of the inclination controller and should, therefore, be realized by all means with corresponding high costs. Errors in the desired forces have less impact on the stabilization — in fact, the horizontal components are equally encoded in the desired task-space trajectories. Also, the lower costs enable the algorithm to generate the desired torques at the CoM via a modulation of the (vertical) contact forces when no other option exists. The realization of these forces via acceleration of masses is described in Section 4.5.

The minimum and maximum normal forces are set as follows:

$$F_{f,n,\mathrm{min}} = 30\,\mathrm{N} \qquad F_{f,n,\mathrm{max}} = 1200\,\mathrm{N} \tag{4.47}$$

$$F_{h,n,\mathrm{min}} = 10\,\mathrm{N} \qquad F_{h,n,\mathrm{max}} = 400\,\mathrm{N}. \tag{4.48}$$

The nonzero minimum forces help to avoid altogether losing contact with the environment. Otherwise, the end effectors may drift away from the contact and lead to a late-contact

Method	$\omega_{\text{FB},z,\text{RMS}}$ [rad/s]	${}_{W}e^{\text{CoM}}_{T_x,\text{RMS}}$ [Nm]	${}_{W}e^{\text{CoM}}_{T_y,\text{RMS}}$ [Nm]
Heuristic	0.1405	**31.5**	25.7
QP, $h_f = 1$, $h_t = 1$	0.1416	32.9	24.0
QP, $h_f = 10$, $h_t = 1$	**0.1322**	32.4	**23.2**
QP, $h_f = 20$, $h_t = 1$	0.1502	34.7	30.3
QP, $h_f = 1$, $h_t = 10$	0.1515	32.5	26.6

Table 4.2: Experimentally identified metrics for different wrench distribution methods and parametrizations. Bold values represent the optimum of a column.

situation when contact forces need to be applied (again). The QP is solved with a custom solver generated via *CVXGEN*, see [194]. The web-based tool uses C-code generation to create tailored solvers for convex problems. The iterations per cycle are limited to 40. The tolerance for solver-related residuals on constraints and the duality gap are set to 1×10^{-4}. The problem can easily be solved under the real-time constraints of the control loop, see Section 6.3.

4.3.4 Experimental Comparison

The optimization-based wrench distribution is experimentally compared to the previously used heuristics for LOLA, see Section 4.1. An early contact scenario, see Figure 3.7, is used for the comparison of the foot-wrench distribution in the double-support phase. The vertical CoM acceleration method described in Section 4.5 is activated with reduced gain $G_{CoM} = 0.26$. The costs for hand contact forces are consistently set to $h_h = 10$. A validation experiment of the optimal wrench distribution in a multi-contact scenario is described in Section 5.5.

Several measures are used for the comparison. The Root Mean Square (RMS) value of the floating-base angular velocity around the vertical axis $\omega_{\text{FB},z,\text{RMS}}$ is used to quantify slipping effects on the ground. Slipping usually occurs when the swing foot is accelerated forwards, and leads to a rotation around the vertical axis. In addition, the RMS errors between the desired and actual torques at the CoM ${}_{W}e^{\text{CoM}}_{T_x,\text{RMS}}$, ${}_{W}e^{\text{CoM}}_{T_y,\text{RMS}}$ indicate how well a wrench distribution solution can be tracked by the contact force controllers. Table 4.2 contains the results for the heuristic and different parametrizations of the QP.

The results show that neither the heuristic nor the QP-based wrench distribution represent global optimums concerning the chosen metrics. The most significant difference between heuristic and single parametrizations of the QP approach is shown for the tracking of the torques around the y-axis with a maximum improvement by 9.7 %. The QP-based approach has advantages concerning slipping for some parametrizations. It is notable that the heuristic already performs quite well given no information on the ground friction is used. The heuristic has another advantage concerning the smoothness of the resulting contact wrenches. The target wrenches generated from the QP are subject to more constraints, causing kinks in the target wrenches when a boundary is hit. An example of the desired contact wrenches resulting from both concepts is depicted in Figure 4.8.

Both methods can be computed in real time. However, the QP-based approach is ≈ 45× slower than the heuristics, see Table 6.1. Nevertheless, the QP-based solution with parametrization $h_f = 10$, $h_t = 1$, $h_h = 10$ is adopted for all subsequent experiments presented in this work. A systematic advantage is the adaptability via cost function tuning. Moreover, only the QP-based solution makes multi-contact balancing possible, see Section 5.5.

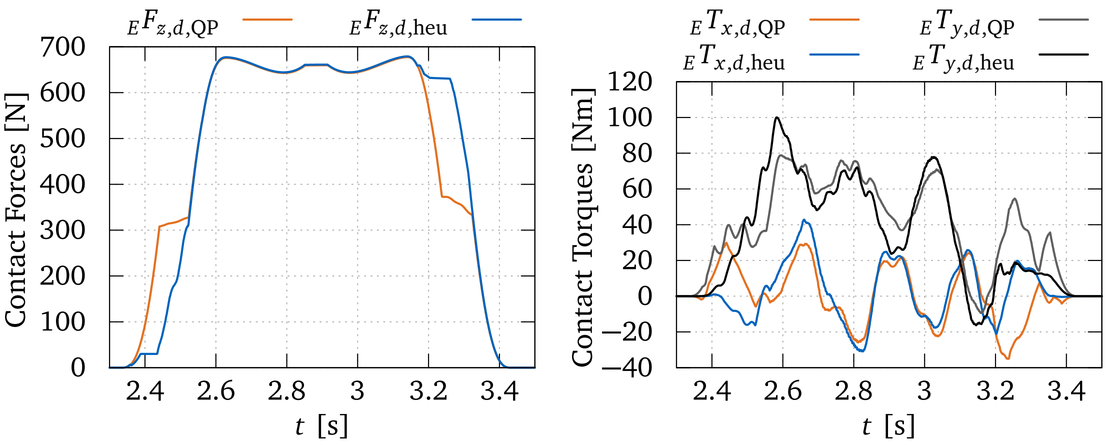


Figure 4.8: The right foot's desired contact wrenches for heuristic (heu) and QP-based optimization method (QP) with $h_f = 10$, $h_t = 1$, $h_h = 10$.

4.3.5 Discussion in the Context of Related Work

There are several examples in the literature using QPs for the distribution of contact wrenches on humanoid robots, e.g. [43, 106, 166, 195]. A comparable formulation with linearized friction cones is described in [106]. The problem additionally considers torque limitations of the joints and does not describe limitations of the yaw torque (this is, however, mentioned in a later publication reusing the method [195]). Instead of regularizing the wrench costs of hands or feet, the wrenches are weighted depending on their origin (planner, interaction task, balance controller). On LOLA, the total desired wrench is a sum of planned wrenches and the wrenches from the inclination controller. Because the inclination controller also mitigates model errors of the planner, no separate weighting is used.

In [43], another related QP formulation is introduced. The friction cone is linearized similarly and considers the correct analytical relation for the yaw torque friction in contrast to the approximation used above. The QP tracks the desired wrench via a soft constraint in the cost function, minimizes the ankle joint torques, and regularizes the normal foot forces to a predefined load, similar to the load factors used in this work. The regularization of normal forces is required to achieve continuous contact forces when contacts open or close. A potential problem with this soft constraint is the parametrization of its weight. When chosen too low, a significant contact force may be assigned to end effectors not in contact. When chosen too high, the load factors (relative force between feet) are tracked ideally, and the algorithm may not use variations of the load to realize torques. In contrast, force and torque limitations are encoded as inequality constraints for the presented approach.

The wrench distribution problem is often solved as part of a bigger QP, which additionally considers the centroidal dynamics [166, 169] or even solves the control problem at task-level [52]. In contrast, the approaches presented in this work stick to the principle of single responsibility of a module. While this may cause suboptimal solutions, the modules can be exchanged — and more importantly — different implementations can be compared. Furthermore, using the planned contact state (load factors) for the distribution is essential for the hierarchical approach's performance. Otherwise, the contact force controller will not make an effort to close an unexpectedly open contact because the wrench distribution commands zero force. Still, using the actual load (measured) for the end effectors may have advantages in late contact scenarios, where the current method still assigns contact wrenches to the corresponding end effector. As a possible enhancement for future research, for example, two problems could be solved: one with the desired loads to calculate wrenches for end effectors currently not in contact; another one using actual loads to optimize the use of currently contacting end effectors. Seemingly, the fundamental problem is to make the resultant contact forces smooth when contacts open or close.

The implementations for wrench distributions in literature are quite similar, with only subtle differences in the chosen cost function or constraints. There is, however, little information on the sensitivity of these approaches to the cost parameters or their performance compared to straightforward heuristics. Notably, the results are not clearly in favor of the optimization problem. The relatively simple heuristic already achieves quite good performance, for some parameter choices even better than the optimization-based solution. Nevertheless, the generalization of the heuristic to multiple contacts in 3D (not a single plane), i.e., hand contacts, is not straightforward. One general disadvantage of the QP implementation seems a lacking constraint on the contact wrench change rate, which should match the maximum dynamics of the subsequent contact force controllers. Because the total performance tracking error depends not only on the physical correctness of the desired wrenches but also on its compatibility with the subsequent force control approaches, tuning the QP parameters can be tedious (has to be done on real hardware). As a possible extension in the future, subsequent control layers could communicate the current cost for realizing the contact wrenches to the upstream modules.

4.4 Vertical Center of Mass Tracking

The reference trajectory for the CoM of the robot in the world FoR is generated by the WPG, see Section 3.5. As part of the task-space description, it is considered in the IK and realized via the joint controllers without drift. Consequently, an additional tracking controller for the CoM position is generally not required.

However, the vertical position of the feet in the world FoR is modified by several control modules, for example the early contact reflex (Section 3.7.1), or the contact force controllers (Chapter 5). Because the CoM height is described relative to the same FoR, the relative height of the CoM above the (stance) foot changes. As a further example, consider the robot stepping on an undetected large plate from ground level. This triggers the early contact reflex, causing all subsequent footsteps to be placed on the new ground height. However, the CoM height remains on the same level, effectively being reduced relative to the new ground height. Therefore, the vertical CoM height relative to the current stance foot must be tracked explicitly. Additionally, the tracking controller is a prerequisite for the method explained in the following section.

The control scheme is based on a virtual spring-damper pair between the desired and the actual CoM height relative to the ground. This concept is passive, provides C^2 smooth trajectories, and is in the literature also used on robots with torque-controlled joints where positional tracking of the CoM is required, see for example [106]. The actual height of the ground and its velocity are determined from the load-weighted location of both feet:

$$z_{\mathrm{ground}} = \boldsymbol{e}_z{}^T \left(\gamma_{\mathrm{ad,RF}}\, {}_W\boldsymbol{r}_{W,\mathrm{RF}} + \gamma_{\mathrm{ad,LF}}\, {}_W\boldsymbol{r}_{W,\mathrm{LF}}\right), \tag{4.49}$$

$$\dot{z}_{\mathrm{ground}} = \boldsymbol{e}_z{}^T \left(\gamma_{\mathrm{ad,RF}}\, {}_W\dot{\boldsymbol{r}}_{W,\mathrm{RF}} + \gamma_{\mathrm{ad,LF}}\, {}_W\dot{\boldsymbol{r}}_{W,\mathrm{LF}}\right). \tag{4.50}$$

In the double-support phase, this is the average height of both feet. In the single-support phase, this equals the height of the stance foot. Analogously, the same can be formulated for the desired, planned foot locations:

$$z_{\mathrm{ground,id}} = \boldsymbol{e}_z{}^T \left(\gamma_{\mathrm{ad,RF}}\, {}_W\boldsymbol{r}_{W,\mathrm{RF,id}} + \gamma_{\mathrm{ad,LF}}\, {}_W\boldsymbol{r}_{W,\mathrm{LF,id}}\right), \tag{4.51}$$

$$\dot{z}_{\mathrm{ground,id}} = \boldsymbol{e}_z{}^T \left(\gamma_{\mathrm{ad,RF}}\, {}_W\dot{\boldsymbol{r}}_{W,\mathrm{RF,id}} + \gamma_{\mathrm{ad,LF}}\, {}_W\dot{\boldsymbol{r}}_{W,\mathrm{LF,id}}\right). \tag{4.52}$$

The error between planned CoM height above planned ground and actual CoM height above actual ground is then given by

$$e_{\text{CoM},z} = \left(\boldsymbol{e}_z{}^T \, {}_W\boldsymbol{r}_{W,\text{CoM,id}} - z_{\text{ground,id}} \right) - \left(\boldsymbol{e}_z{}^T \, {}_W\boldsymbol{r}_{W,\text{CoM}} - z_{\text{ground}} \right). \tag{4.53}$$

Using this difference, its velocity, and the robot's mass m, the accelerations on the CoM generated by the tracking controller result to

$$\Delta\ddot{z}_{\text{CoM,track}} = \frac{1}{m} \left[K \, e_{\text{CoM},z} + D \, \dot{e}_{\text{CoM},z} \right], \tag{4.54}$$

with the virtual stiffness K and virtual damping D. Together with an additional acceleration input $\Delta\ddot{z}_{\text{CoM,ctl}}$ (described in Section 4.5), the tracking controller output is integrated and added to the task-space trajectories:

$$\Delta\ddot{z}_{\text{CoM}} = \Delta\ddot{z}_{\text{CoM,track}} + \Delta\ddot{z}_{\text{CoM,ctl}} \tag{4.55}$$

$$\Delta\dot{z}_{\text{CoM}} = \text{int}(\Delta\ddot{z}_{\text{CoM}}) \qquad \Delta z_{\text{CoM}} = \text{int}(\Delta\dot{z}_{\text{CoM}}) \tag{4.56}$$

$$\boldsymbol{X}_{\text{bl}}{}' = \boldsymbol{X}_{\text{ad}} + \boldsymbol{S}_{\text{CoM},z} \, \Delta z_{\text{CoM}} \qquad \boldsymbol{V}'_{\text{bl}} = \boldsymbol{V}_{\text{ad}} + \boldsymbol{S}_{\text{CoM},z} \, \Delta\dot{z}_{\text{CoM}}. \tag{4.57}$$

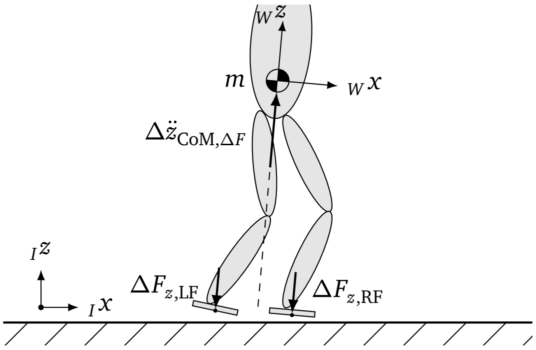
The binary selection vector $\boldsymbol{S}_{\text{CoM},z}$ is defined such that

$$\boldsymbol{S}^T_{\text{CoM},z} \boldsymbol{X} = \boldsymbol{e}_z{}^T \, {}_W\boldsymbol{r}_{W,\text{CoM}}. \tag{4.58}$$

The choice of the parameters K, D is a tradeoff between precise tracking of the reference on the one hand and smooth resulting contact force profiles, i.e., low accelerations, on the other hand. Precise tracking is required to avoid violations of the kinematic limits. However, abrupt changes in the CoM height might disturb the robot's state and may cause the robot to fall. The tracking controller is primarily active when the robot steps on an undetected obstacle, which requires to increase the CoM height relative to the new ground height. The fact that the robot does already step on the obstacle means the tracking of the CoM height is a long-term kinematic problem. For example, a consecutive increase in ground height might then lead to violations of the kinematic limits. In contrast, the robot's state may already be disturbed when stepping on an undetected new ground. Thus, smooth force profiles are chosen over precise tracking of the ideal CoM height. In addition, a method presented in the following section uses deliberate deviations from the planned trajectory to improve the robot's balancing. Hence, the coefficients for the virtual spring–damper system are chosen to relatively low values $K = 4500\,\text{N/m}$, $D = 800\,\text{Ns/m}$, which correspond to an eigenfrequency of $\approx 1.3\,\text{Hz}$ and a damping ratio of ≈ 0.73. The damping rate must be high enough to avoid unwanted oscillations of the CoM for example on soft ground. As an alternative, a low damping can be combined with a time-domain PC approach, see [105]. The chosen parameter set keeps the desired CoM height in the long-term, and allows deviations for a short time.

4.5 Vertical Center of Mass Acceleration

It is known that an acceleration of the robot's masses directly causes contact forces, see Equation (3.3). A simplified way to create such reaction forces is to accelerate all robot bodies in the same direction, i.e., accelerating the CoM. This reduces the robot's dynamics to a single point mass and requires only little effort to model and plan feasible acceleration trajectories in contrast to dealing with individual joint accelerations. One possibility is the horizontal acceleration of the CoM. Unfortunately, a modification of the horizontal location of



$_Wz$
m
$_Wx$
$\Delta\ddot{z}_{\mathrm{CoM},\Delta F}$
$_Iz$
$_Ix$
$\Delta F_{z,\mathrm{LF}}$
$\Delta F_{z,\mathrm{RF}}$

Figure 4.9: Illustration of the used dynamics model with the vertical acceleration of the CoM $\Delta\ddot{z}_{CoM,\Delta F}$ and the vertical force tracking errors of the left and right foot $\Delta F_{z,\mathrm{LF}}$, $\Delta F_{z,\mathrm{RF}}$. Adapted from [295] ©2017 IEEE.

the CoM may also require a modified BoS, i.e., a change of the footstep locations. Thus, such approaches generally need to be combined with global footstep modification methods [114, 316]. In contrast, vertical accelerations of the CoM are unproblematic when the footstep locations are kept and are also not limited by the friction of the contact.

The method presented in the following shows that the modification of the vertical contact forces on the robot is a powerful way to improve the robot's balancing. Because the approach modifies the contact forces with the environment, it may also be interpreted as an extension of a force controller. Because the modification acts on the total force of all end effectors, however, it is covered in this chapter. The general idea behind the approach has first been published in Sygulla et al. [295] and has been extended in Sygulla and Rixen [292]. For this thesis, the method is ported to the new task-space definition. Furthermore, the influence of the structural dynamics is considered, which significantly increases the approach's performance on LOLA. The com acceleration module is part of the inclination control loop, parallel to contact force controllers and the wrench distribution. Its linearization is described via $\boldsymbol{C}_{\mathrm{CoM}}(s)$ in Figure 3.9.

4.5.1 Control Approach

The centroidal dynamics of the robot connects the contact wrenches on the feet with the acceleration of the CoM. For translational accelerations, the dynamics can be reduced to a single point mass, see Figure 4.9. For the calculation of CoM accelerations, only differences of the actual (measured) contact forces $\boldsymbol{F}_{f,\mathrm{m}}$ to the desired forces $\boldsymbol{F}_{f,\mathrm{d}}$ are considered. Furthermore, the forces are expressed in the world FoR:

$$\Delta F_{z,f} = \boldsymbol{e}_z{}^T\, {}_W\boldsymbol{A}_E\, ({}_E\boldsymbol{F}_{f,\mathrm{d}} - {}_E\boldsymbol{F}_{f,\mathrm{m}}). \tag{4.59}$$

Errors in the contact forces occur when the contact force controllers (see Chapter 5) are unable to track their setpoints. For example, this may occur when the end effectors' planned contact state does not match the actual one or if the ground has unexpected mechanical properties. From the contact force errors, a desired acceleration of the CoM is then calculated:

$$\Delta\ddot{z}_{\mathrm{CoM},\Delta F} = \frac{G_{\mathrm{CoM}}}{m}(\Delta F_{z,\mathrm{RF}}\,\kappa_{\mathrm{RF}} + \Delta F_{z,\mathrm{LF}}\,\kappa_{\mathrm{LF}}), \tag{4.60}$$

with additional scaling gain $G_{\mathrm{CoM}} \in]0,1]$. An acceleration of the CoM only has an effect on the contact forces of the feet in contact. Consequently, the force errors are scaled with the actual contact factors $\kappa_f \in [0,1]$. For the CoM controller, only foot contacts are considered.

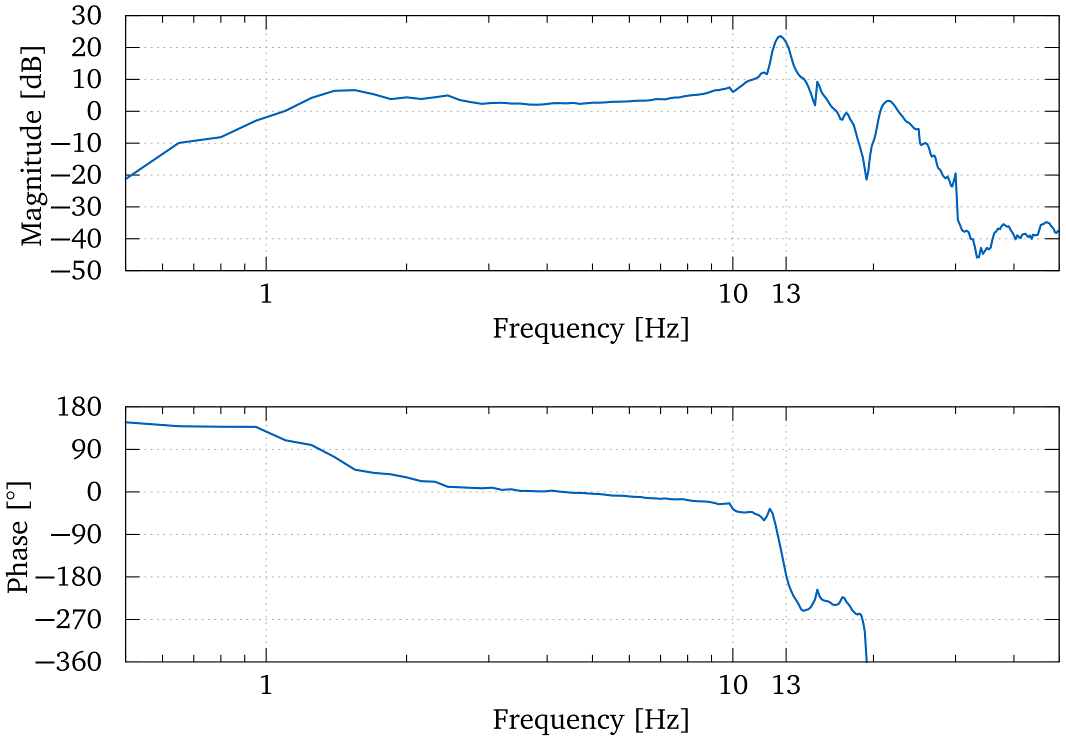


Figure 4.10: Estimated transfer function for the open loop of the vertical acceleration controller, consisting of estimated plant $P_{\mathrm{CoM},3}$, controller Equation (4.60), and CoM tracker Equation (4.54) for $G_{\mathrm{CoM}} = 1$. The original estimated plant is depicted in Figure C.4.

Frequency [Hz]	Sharpness Q	Damping d
12.7	1.6	0.017618
15.17	1.1914	0.13271
21.67	2.72	0.05806

Table 4.3: Notch filter parameters for the vertical CoM acceleration controller. The locations of the notches match the resonance frequencies depicted in Figure 4.10. For the notch implementation refer to Appendix A.3.

As an example, consider a late contact on the right foot. Because the corresponding κ_{RF} is zero, in this case, only the force errors on the left foot are considered. Because the left foot must additionally provide the forces planned for the right foot, $\Delta F_{z,\mathrm{LF}} < 0$ and the CoM is accelerated downwards.

Experiments with LOLA show that the control approach becomes unstable for gain values $G_{\mathrm{CoM}} > 0.3$. This has already been observed in Sygulla and Rixen [292], and the cause has been suspected in structural resonances of the robot. By analyzing the corresponding estimated plant $\boldsymbol{P}_{\mathrm{CoM}}$, see Section 3.8.2, this assumption can be confirmed.

The open-loop transfer function for the vertical acceleration controller is depicted in Figure 4.10. The critical phase is reached at ≈ 13 Hz and the three distinct resonances above this frequency make the control loop unstable for $G_{\mathrm{CoM}} = 1$. In accordance with the strategy for the inclination controller, a cascade of notch filters is used on the controller output $\Delta\ddot{z}_{\mathrm{CoM},\Delta F}$ to shape the open-loop transfer function and suppress structural resonances, see Table 4.3. The output of the notch() operators is then used for the additional acceleration input $\Delta\ddot{z}_{\mathrm{CoM,ctl}}$ in Equation (4.55).

4.5.2 Late-Contact Foot Acceleration

The control approach only accelerates the CoM, without changing the foot positions in the world FoR. Consider the case of a foot being in a late contact, i.e., contact is expected but not present. In this case, it is advantageous to accelerate the CoM and the foot not in contact simultaneously. This strategy reduces the time required to close the late contact and is also used in the previous implementation [292]. Due to a changed task-space definition, where the foot positions are no longer described relative to the CoM, this strategy must be implemented explicitly compared to the implicit realization in [292]. Based on the total commanded CoM velocity $\Delta\dot{z}_{CoM}$, see Equation (4.56), a vertical velocity for each foot is calculated:

$$ {}_W\Delta\dot{z}_{f,\mathrm{acc}}[n] = \mathrm{hybridCtl}(\beta_f, (1-\kappa_f)\min(\Delta\dot{z}_{\mathrm{CoM}}, 0), {}_W\Delta z_{f,\mathrm{acc}}[n-1]). \tag{4.61} $$

The controller uses the hybrid control scheme described in Appendix A.5. Only velocities directed to the ground are applied to the swing foot — otherwise the foot may lift off unexpectedly and never make contact when the CoM is accelerated upwards. The modification is active when there is no actual contact $(1-\kappa_f) > 0$ and the foot is force controlled ($\beta_f > 0$). It is driven back to zero when the foot becomes position controlled with $\beta_f = 0$. The resulting velocity is integrated and added to the trajectories of the feet:

$$ {}_W\Delta z_{f,\mathrm{acc}} = \mathrm{int}({}_W\Delta\dot{z}_{f,\mathrm{acc}}) \tag{4.62} $$

$$ (\boldsymbol{X}_{\mathrm{bl}}, \boldsymbol{V}_{\mathrm{bl}}) = {}_W\,\mathrm{modifyAll}_f\left(\boldsymbol{X}'_{\mathrm{bl}}, \boldsymbol{V}'_{\mathrm{bl}}, \begin{bmatrix}\boldsymbol{e}_z\\ \boldsymbol{0}\end{bmatrix} {}_W\Delta z_{f,\mathrm{acc}}, \begin{bmatrix}\boldsymbol{e}_z\\ \boldsymbol{0}\end{bmatrix} {}_W\Delta\dot{z}_{f,\mathrm{acc}}\right). \tag{4.63} $$

This extension to the pure CoM acceleration makes the proposed method an effective way to deal with undetected changes in the ground level.

4.5.3 Experimental Validation

The effect of the vertical CoM acceleration controller on the performance of LOLA in an early contact scenario, see Figure 3.7, is evaluated in the following. The robot walks at a speed of 0.5 m/s on an undetected board having a height of 4 cm. A comparison between the reference (no vertical CoM acceleration) and the controller with different gains G_{CoM} is conducted. Oscillations in the vertical contact forces are observed during single support. These oscillations increase for higher gains, see Figure 4.11. It is in particular interesting that the vibrations are not dominant in the desired accelerations of the CoM controller, i.e., they are already attenuated by the loop-shaping filters. Also, an additional notch filter directly at the frequency of the vibration did not change the outcome. Furthermore, the vibrations are not reduced for lower gains of the inclination controller (see Section 4.2) or of the contact force controllers (see Chapter 5). Further, the vibrations are effectively damped by the contact force controllers, and reduced force control gains lead to higher amplitudes.

The acceleration controller with its double integrator introduces a negative phase shift and may theoretically cause closed-loop instability. However, the fact that the vibrations also exist in the reference and damping does not change significantly with higher gains indicates a different reason. A higher gain causes higher vertical accelerations and a higher excitation of mechanical eigenfrequencies of the system. Thus, the contact force variations may simply be caused by a strong excitation of upper-body eigenfrequencies. One should note that the mechanical vibrations are barely visible on the real robot, i.e, only have a small (position) amplitude.

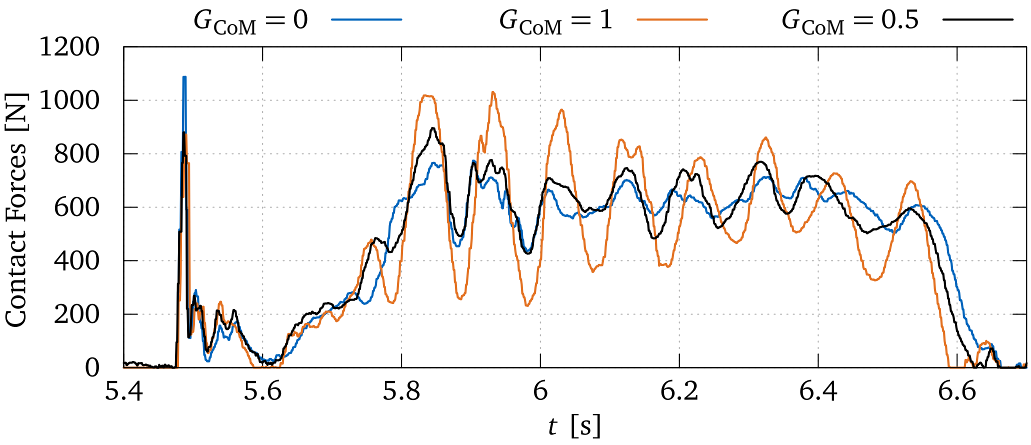


Figure 4.11: Vertical contact force on the right foot after an early contact and for different CoM acceleration gains G_{CoM}.

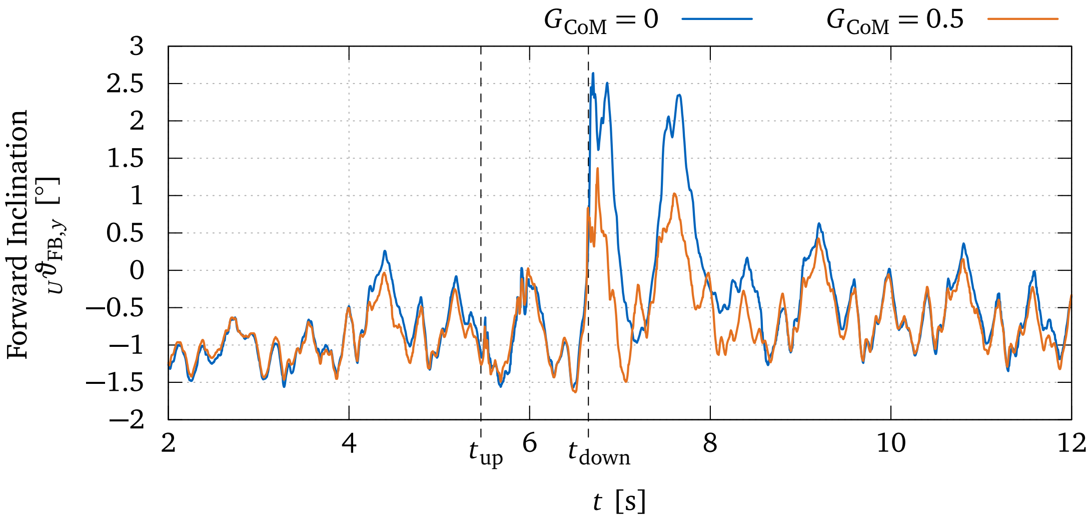


Figure 4.12: Forward floating-base inclination of LOLA walking over an unexpected obstacle without and with vertical CoM acceleration.

Fortunately, the relation between the gain G_{CoM} and the performance of the whole system in terms of the floating-base inclination is nonlinear. For low gains already high performance improvements are achieved; for higher gains, the relative increase is less. Thus, a reduced gain $G_{\text{CoM}} = 0.5$ already provides good performance and acceptable oscillations in the vertical forces, see the floating-base inclinations for stepping on and off an unexpected obstacle in Figure 4.12. This parametrization is used for all experiments in this thesis unless noted differently. The data shows that the concept is most beneficial when the robot walks down an unexpected negative ground height change. The forward inclination is reduced by approximately 30 % compared to the reference controller. Further, using the vertical CoM acceleration, the disturbance on the robot is reduced faster, taking only two steps for the inclination to return to the common pattern during straight walking. Snapshots of the motion showing the resulting CoM trajectory are depicted in Figure 4.13. On flat ground, the method slightly improves the peak-to-peak value for the floating-base inclinations. Further experiments with higher obstacles are described in Section 7.2. In these scenarios, the reference controller is no longer able to overcome the obstacles.

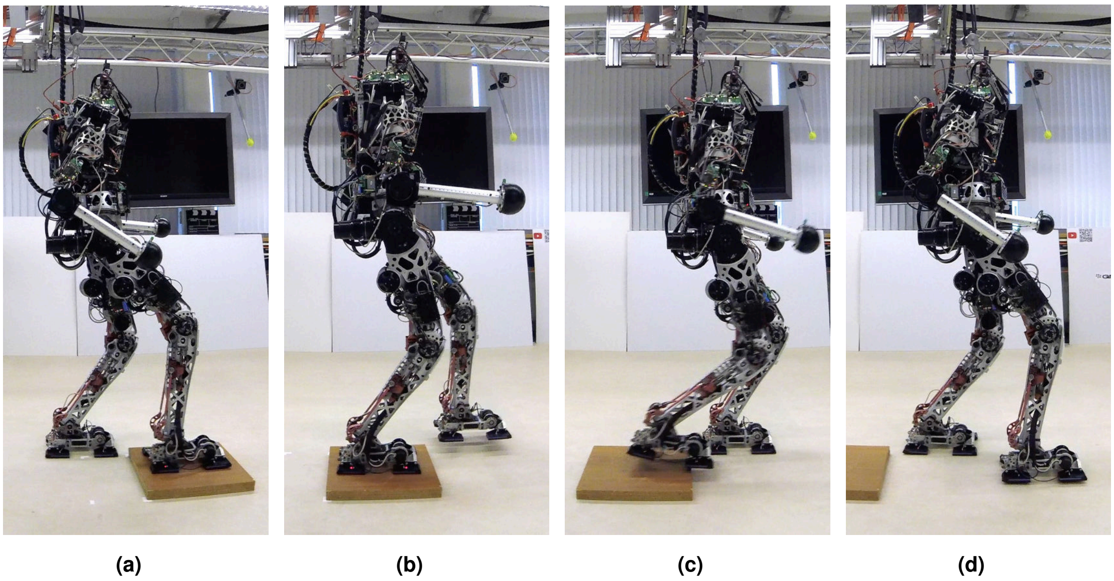


Figure 4.13: Motion snapshots from initial early contact (a) to full compensation of the disturbance (d). The CoM height is increased to a maximum in (b) to match the new ground height. In (c) the CoM is accelerated downwards to track new ground and reduce the impact forces.

4.5.4 Discussion in the Context of Related Work

The proposed method improves the robustness of the biped to undetected terrain using two different mechanisms. The first generates contact forces via the vertical acceleration of the CoM. The second reduces the duration of a late contact by tying the vertical foot movements to the CoM accelerations of the first mechanism.

CoM Admittance Control The vertical CoM acceleration controller Equation (4.60) must be analyzed in combination with the CoM tracking controller Equation (4.54), which limits the deviations from the reference height and velocity. If one assumes for the moment that the feet are exactly at their planned locations, an error in the CoM height can only be induced by the acceleration controller itself, i.e., $e_{\text{CoM},z} = -\Delta z_{\text{CoM}}$, see Equations (4.53) to (4.56). This simplification is merely done for the purpose of making the following lines easier to digest for the reader. The statements written in the following do hold independently of this assumption. From Equations (4.54), (4.55) and (4.60) then follows

$$m\,\Delta\ddot{z}_{\text{CoM}} + D\,\Delta\dot{z}_{\text{CoM}} + K\,\Delta z_{\text{CoM}} = G_{\text{CoM}}\,(\Delta F_{z,\text{RF}}\,\kappa_{\text{RF}} + \Delta F_{z,\text{LF}}\,\kappa_{\text{LF}}). \tag{4.64}$$

Equation (4.64) describes an admittance controller changing the vertical CoM position based on an error in the contact forces. Because the CoM is tracked relatively to the ground (the average foot height), this may equally be interpreted as admittance on the foot position relative to the base (CoM).

Admittance control — also known as position-based impedance control — is often utilized to mitigate the landing impacts of the swing foot. One approach for robots with position-controlled joints is to reduce the swing leg's joint controller gains just before anticipated contact [101, 218]. Alternatively, admittance control can be implemented by modifying the desired trajectories based on FT sensor information [181, 232, 233, 257, 277]. The force reference for this controller usually depends on a predefined [232, 233, 277] or previously recorded force profile [257]. Later work extended the concept to using model-based force references — i.e., the output of a ZMP controller [201]. In general, admittance control leads to a tradeoff between impact mitigation during the landing phase and the control of the desired forces once the foot is in safe contact with the ground. In the first case, the stiffness should be relatively low, while in the second case, the stiffness needs to be high to track the desired force profiles. In related work, this is often resolved by varying the impedance parameters based on the gait cycle phase [181, 232, 257, 277]. Another approach is to switch between different control concepts, e.g., admittance and position-control [103], or admittance and computed torque control [233].

A different approach for force control on biped robots with position-controlled joints is the direct force-control scheme [33, 86, 295], which does not impose a particular admittance on the system but tracks a given force directly. Based on these concepts, the contact forces at the end effectors of LOLA are controlled to track the reference force trajectories from the wrench distribution. For the vertical contact forces, the control efforts from the contact force controllers and the admittance controller on the CoM in Equation (4.64) are superimposed. The proposed combination of these approaches has several advantages. The contact controllers precisely track the desired force trajectories primarily using relative motions between the end effectors. Both the contact controllers and the CoM admittance controller react to vertical disturbances on the feet. The combination makes time-varying parameters for the admittance controller unnecessary. Furthermore, the approaches complement each other: In the double-support phase, the contact controllers ensure precise vertical force tracking; in the single-support phase, the contact forces can only be influenced via acceleration of the CoM.

Balance Control via Vertical CoM Height Modification The method of CoM acceleration is motivated by the generation of contact forces on the feet to reduce the error to a reference force. This vertical reference force on the CoM is generated in the planner from a predefined CoM height trajectory, see Figure 3.4, and a reduced dynamic model of the biped. It is notable that the method also modifies the vertical CoM trajectory to compensate for inaccuracies in these dynamic models, which slightly improves the balancing on flat ground.

The CoM height of bipeds typically follows a predefined trajectory and is not modified — even in the case of disturbances. Koolen, Posa, and Tedrake [165] show, however, that a biped can be balanced solely by a modification of the CoM height trajectory. The considered controller is based on the orbital energy of a 2D Variable-Height Inverted Pendulum (VHIP) in simulation.

In contrast, the prevailing idea in the proposed method is to improve the tracking of vertical forces by an acceleration of masses. The method presented in this thesis has first been introduced in [295] and has been extended in [292]. When the CoP reaches the borders of the BoS, the modulation of the vertical contact forces is the best way (not limited by friction) to induce stabilization torques on the floating base and can only be achieved by a vertical acceleration of masses (with consequently changed vertical CoM trajectory). One may argue the vertical reference force is predefined and not modified when a disturbance acts on LOLA. However, the wrench distribution method may choose to violate the predefined forces on the CoM in favor of stabilizing torques when the end effector wrench reserves are depleted, see Section 4.3. Furthermore, impacts on the robot's base (especially after a late contact) directly cause vertical CoM accelerations. The results indicate that the foot does not solely land earlier in a scenario with negative ground height change, but that the resulting disturbance of the robot's state is reduced faster to nominal.

Later, Van Hofslot, Griffin, Bertrand, and Pratt [312], described a concept of balancing via vertical CoM accelerations applied to the biped *Valkyrie*. When the CoP reaches the BoS — i.e., when the feet can no longer apply torques on the ground — a bang-bang controller applies a vertical acceleration profile to the CoM. A disturbance rejection scenario in still-stand shows an advantage of the method compared to a reference implementation. Recently, another approach on a biped with position-controlled joints is described in [42], which develops an extended DCM controller for the VHIP model to consider CoM height variations in the desired contact forces. The linear best-effort DCM controller results from a QP, which is solved at each time step to consider the time-varying properties of the VHIP. Like the work at hand, vertical admittance control on the normalized leg stiffness is applied to track the total contact forces on the position-controlled biped with a low gain. It is reported that higher gains lead to vibrations of the system. The contact force controllers only act on the foot forces' difference and are, therefore, not superimposed to the admittance controller.

Conceptionally, control frameworks for robots with torque-controlled joints do not need to generate accelerations from contact force errors explicitly. The problem changes from a force-control to a position-control task of the CoM. Popular control frameworks usually deploy high-gain PD control to track a CoM reference trajectory [166, 175], which does not allow modifications to a predefined reference trajectory. In other cases, low impedance gains are only used for balancing in stillstand and not for walking [106, 127]. Only recently, interesting work replaced the hard constraint on the height of the CoM with a soft constraint of the knee joint angle [92], allowing variations of the CoM trajectory (vertical CoM accelerations).

Late Contact Strategy By moving a foot in a late contact with the same vertical velocity as the CoM, the contact controllers are supported in their effort to close these contacts as fast as possible. This significantly reduces the disturbance on the floating base in such late contact scenarios. It enables the robot to overcome unexpected height changes of several centimeters, see also Section 7.2.

In [316], a late-contact strategy using horizontal accelerations of the CoM is proposed. It is based on the observation that the planned CoP is outside of the BoS in a late-contact scenario. The CoM is accelerated forwards, extending the single-support phase to reestablish feasibility of the CoM trajectory concerning the unilateral contact. Due to the horizontal acceleration, the robot's floating-base is not disturbed by the missing contact force. However, the effect can not be held up infinitely long, and the method primarily buys time to let the contact controllers establish ground contact. The modification of the horizontal CoM location generally requires changing the next footstep location to avoid falling forwards when the foot finally makes contact with the ground. This assumes that a change of the footstep location is possible regarding collisions with the environment. In contrast, the vertical adaptation of the CoM proposed above — which is also limited to short periods due to kinematic limits — reduces the time until full contact is made and does not require additional stabilization techniques or an adaptation of footstep positions. Both methods may be combined to reduce the floating-base disturbance further while contact with the ground is established.

Late contacts are also investigated in [201]. The approach applies a constant vertical velocity to a foot with late contact to reduce the duration until full contact is made. In contrast, the presented approach is driven by differences in the actual contact forces.

Biologically Inspired Walking The concept is somehow related to biologically inspired semi-passive walking machines, which use a limited number of actuators and compliant elements in the legs to achieve energy-efficient walking. Typically, a combination of a point mass with a spring or spring-damper pair is used to model the system's dynamics. These machines achieve impressing robustness to ground height changes [125, 229], are, however, restricted to a subspace of possible motions compared to a full actuation of the joints. Findings in [21] indicate that a set of task-level priorities for a damped spring-mass system with an actuator in series is sufficient to describe the ground reaction forces of robust and energy-aware walking. This outcome is obtained by analyzing birds' behavior when walking over an expected (seen) obstacle. The investigations consider point feet with the spring-damper pair being aligned with the leg. Although in this thesis, only the vertical force components are considered for the CoM acceleration, several similarities exist. The admittance on the CoM describes a virtual spring-damper pair between the CoM and the feet. Only forces deviating from the reference, however, act on this admittance. It seems possible to realize similar behavior by selecting a particular force reference related to the series actuator work in the biological model. The CoM strategy for overcoming an unexpected obstacle, which follows from the CoM tracking controller, seems related to the "Energy exchange strategy" described in [21, Fig. 1., p.3787]. The CoM height of LOLA is increased to keep the distance to the ground constant. However, the control scheme operates only in the vertical direction, without actually exchanging kinetic energy to potential energy — the horizontal velocity of the CoM is kept constant, and the actuators must spend energy to increase the CoM height. The relation to the semi-passive biological model also further motivates the additional late contact strategy for the swing foot. When the feet are considered spring-damper pairs in the vertical direction, the swing foot is naturally tied to the movements of the CoM.

Compared to the virtual spring-damper pair, mechanical compliances have the advantage of zero delays in response to impacts and may store energy. However, the floating-base dynamics are relatively slow, making it possible to use virtual components for rapid prototyping of biological actuation principles. The experiments with LOLA show that the control schemes are fast enough to handle ground height changes of several centimeters.

Limitations of Vibration Suppression With the approaches described in this chapter, structural resonances can effectively be decoupled from feedback loops. This means that the control scheme's stability is not affected by the mechanical structure, and the crossover frequency may be improved. However, the mechanical structure and the damping of the eigenfrequencies are not changed by these concepts. Although the amplitudes may be low, structural vibrations can still significantly affect the ground reaction forces, especially when the corresponding eigenmodes are only slightly damped. An effective countermeasure is damping control; see the contact force controllers in Chapter 5. However, the corresponding control loops on bipeds are not collocated with respect to the ground reaction forces, the ground properties are subject to uncertainty, and the actuator dynamics are limited. Thus, only a limited amount of damping can be injected into the system. These observations support the design goal of sufficient damping in the mechanical structure and the controlled joints.

4.6 Chapter Summary

This chapter describes methods related to balancing the floating-base of a biped robot using IMU data. It is shown that the mechanical structure of the robot has a significant influence on corresponding control loops. Several approaches for the suppression of structural resonances in these feedback loops are evaluated and discussed. The application of these methods enables a higher crossover frequency of floating-base balancing controllers, and therefore a more effective way of disturbance rejection.

The chapter further deals with techniques to distribute an overall desired contact wrench to several end effectors of the robot. An optimization-based approach is compared with a heuristic in experiments with LOLA. The results indicate only minor differences in the two approaches for normal walking. In addition, a concept to use vertical accelerations of the CoM for the stabilization of the floating-base is proposed and experimentally evaluated. On the one hand, the method uses an acceleration of the robot's bodies to reduce errors in the vertical contact force. On the other hand, it enables to effectively overcome late contact scenarios, i.e., an undetected decrease in ground height.

The methods described in this chapter generate modified task-space trajectories and desired contact wrenches for the end effectors. How the force trajectories are realized using contact wrench control is discussed in following chapter.

Chapter 5

Contact Force Control

Parts of this chapter have previously been published in [292, 295].

The desired contact wrenches for the end effectors, which the balance controller calculates (see Chapter 4), are regulated via individual contact force controllers (5 in Figure 3.1). This chapter introduces several control approaches designed to achieve robust, high-bandwidth control of contact wrenches. A general design guideline is that the mechanical properties of the contact are assumed unknown and variable. Thus, the control schemes must deal with completely different mechanical properties while preserving stability and, ideally, the closed-loop crossover frequency. A lower bound of the contact control bandwidth is critical since the cascaded higher-level balance control loop may become unstable otherwise.

This chapter presents control strategies for the ground reaction forces, which are either robust to the contact's uncertainty or adapt to it. Experimental data and theoretical considerations prove the robustness of the approaches. Additional sensor data from contact sensors — or more generally a tactile foot sole, see Section 2.4 — is utilized to improve the performance when only parts of the foot are in contact with the environment. For this, a formulation of an explicit contact model, which directly parametrizes the contact surface, is described first. Moreover, the chapter deals with adaptive control schemes to handle different mechanical ground properties. Finally, a contact force control scheme for the robot's hands is presented to enable multi-contact locomotion. The contact force control module operates as an inner loop of the feedback loops. Its linearized transfer function is denoted with $\boldsymbol{C}_{\text{contact}}(s)$ in Figure 3.9.

5.1 Previous Work on Lola

Previous work on a hybrid position/force controller for Lola is described in [38]. The approach uses an explicit contact model, which is defined as gradient of forces/torques with respect to motions in the configuration space:

$$\dot{\boldsymbol{\lambda}}_f = \frac{\partial \boldsymbol{\lambda}_f}{\partial \boldsymbol{q}} \dot{\boldsymbol{q}}. \tag{5.1}$$

This requires the calculation of Jacobians for the contact points and an additional IK within the hybrid position/force controller to solve for task-space modifications. Although being an elegant formulation in many ways, the effects of changing contact surfaces are hard to formalize for a gradient in the configuration space $\boldsymbol{q}$.

5.2 Ground Reaction Force Control

The control scheme basis for ground reaction force control at the feet consists of a contact model and a regulator based on hybrid position/force control. Each foot has a separate independent control instance. The components of this control approach are described in the following.

5.2.1 Explicit Contact Model in Task-Space

The proposed explicit contact model describes the relation between ground reaction forces and a foot's corresponding movements in task space. It is parameterized with geometry information on the contact surface and the mean contact stiffness of the ground. This explicit way to formulate the contact model enables the use of sensory information on the current contact state to improve the tracking in partial contact situations, see Section 5.3. The proposed model is intended for flat robot feet with or without toe elements. The following work has — with a minor difference in formulation — previously been published in [295]. In its original form, the model was used in an inertial frame. This was mainly due to restrictions of the task-space definition by that time. The foot-attached frame formulation shown here is certainly more suitable to curve-walking and other scenarios with arbitrary foot orientation (ramps for example).

Choosing the force-controlled space The contact between a robot's foot and the ground is unilateral, i.e., the robot can not pull on the ground, and three components of the contact wrench (${}_ET_x$, ${}_ET_y$, ${}_EF_z$) depend on friction only. These tangential friction-dependent components, whose feasibility is greatly influenced by the ground's unknown mechanical properties, are not considered in the force control scheme. Looking at the way the WPG generates the desired contact wrench and the motion of the center of mass, see Section 3.5, reveals that these tangential contact forces automatically result from the planned movement of the feet, or more precisely, the acceleration of the CoM. There is no need for an additional feedback loop on these forces as long as there is sufficient friction; furthermore, strategies that reduce the robot's angular momentum around its vertical axis can help avoid slipping effects [264].

The rest of the degrees of freedom of the foot, namely the torques ${}_ET_x$, ${}_ET_y$, and the normal contact force ${}_EF_z$ are susceptible to changes of the environment, the timing of the motion, or a disturbed state of the robot. Thus, these directions are force-controlled and considered in the explicit contact model; all other directions remain position-controlled. The separation of force-controlled and position-controlled directions is done in the foot-attached end-effector frame E.

The infinitesimal contact element The contact with the environment is approximated by a linear elastic stiffness acting on each infinitesimal element of the contact area. This equals an infinite number of decoupled linear springs, which act on the robot's contacting foot, Figure 5.1. The stiffness $c_z > 0\,\mathrm{N/m^3}$ of these elements can, for instance, be derived from the elasticity of the contact material E and the effective thickness d of the ground with $c_z = E/d$. Alternatively, it may be determined experimentally by measuring the vertical stiffness of a ground contact k_z with a contact area A: $c_z = k_z/A$. It is assumed that all contact elements, and the TCP lie in the same plane of the foot. Within that plane, the position of each contact element i is described via ${}_E\boldsymbol{r}_i = [{}_Ex_i\,, {}_Ey_i\,, 0]^T$ with ${}_Ex_i, {}_Ey_i$ defined in the foot-attached end-effector frame E. Considering such a contact element and an infinitesimal small pitch angle

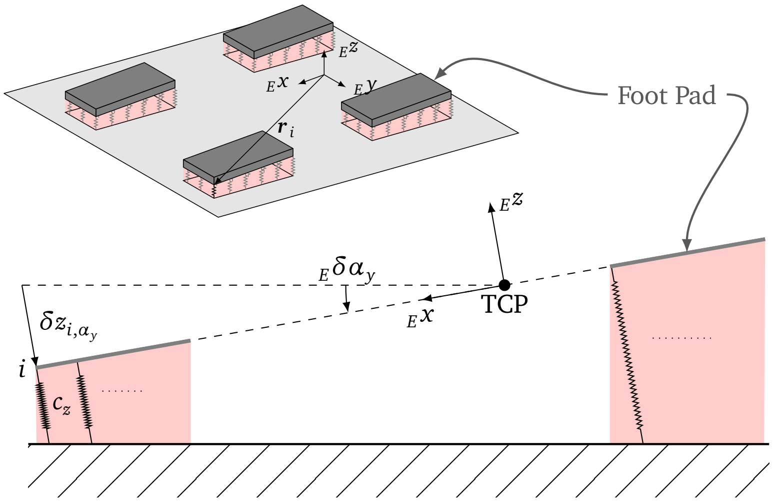


Figure 5.1: LOLA's foot with the four contact pads and the proposed contact element model in top and side view. This visualization applies to LOLA's foot design from 2013, see Figure 2.1b. The foot is rotated with the pitch angle $_E\delta\alpha_y$. Adapted from [295] ©2017 IEEE.

of the foot $_E\delta\alpha_y$ around the $_Ey$ axis — see Figure 5.1 — the pitch-induced deformation of the contact element results to

$$\delta z_{i,\alpha_y} = {_Ex_i}\, {_E\delta\alpha_y}. \tag{5.2}$$

With these differential rotations being infinitesimal small around the coordinate system's axes, the total deformation of a contact element — depending on the roll angle $_E\delta\alpha_x$, pitch angle $_E\delta\alpha_y$, and a vertical movement of the foot $_E\delta z_{\text{TCP}}$ — is then a superposition of the deformations caused by the individual components:

$$\delta z_i = {_Ex_i}\, {_E\delta\alpha_y} - {_Ey_i}\, {_E\delta\alpha_x} - {_E\delta z_{\text{TCP}}}, \tag{5.3}$$

which yields the following differential vertical force (compressive forces are positive) for contact element i:

$$\delta f_{z,i} = c_z \left[\, {_Ex_i}\, {_E\delta\alpha_y} - {_Ey_i}\, {_E\delta\alpha_x} - {_E\delta z_{\text{TCP}}} \right]. \tag{5.4}$$

Parameterization for arbitrary contact surfaces For an arbitrarily shaped planar contact surface S, the resulting differential torques and forces on the foot — caused by the differential movements $_E\delta\alpha_x$, $_E\delta\alpha_y$, and $_E\delta z_{\text{TCP}}$ — are calculated by integration over every (active) contact surface element $\mathrm{d}S_i = \mathrm{d}_Ex_i\, \mathrm{d}_Ey_i$. This yields equations for the differential torques $_E\delta T_x$, $_E\delta T_y$, and the differential normal force $_E\delta F_z$:

$$_E\delta T_x = \iint_S \delta f_{z,i}\, {_Ey_i}\, \mathrm{d}S_i \tag{5.5}$$

$$_E\delta T_y = -\iint_S \delta f_{z,i}\, {_Ex_i}\, \mathrm{d}S_i \tag{5.6}$$

$$_E\delta F_z = \iint_S \delta f_{z,i}\, \mathrm{d}S_i. \tag{5.7}$$

Inserting Equation (5.4) into Equation (5.5) and simplifying yields:

$$ {}_E\delta T_x = c_z \Big[{}_E\delta\alpha_y \underbrace{\iint_S {}_Ex_i\, {}_Ey_i \,\mathrm{d}S_i}_{I_{xy}} - {}_E\delta\alpha_x \underbrace{\iint_S {}_Ey_i^2 \,\mathrm{d}S_i}_{I_{xx}} - {}_E\delta z_{\mathrm{TCP}} \underbrace{\iint_S {}_Ey_i \,\mathrm{d}S_i}_{A\,r_{c,y}} \Big]. \tag{5.8} $$

The first two surface integrals can be interpreted as the second moments of area $I_{xy}, I_{xx} > 0$ of the contact area $A > 0$. The third integral contains the position of the center of the contact area $r_{c,y}$ in ${}_Ey$-direction with respect to the TCP. Equation (5.6) analogously becomes

$$ {}_E\delta T_y = c_z \Big[- {}_E\delta\alpha_y \underbrace{\iint_S {}_Ex_i^2 \,\mathrm{d}S_i}_{I_{yy}} + {}_E\delta\alpha_x \underbrace{\iint_S {}_Ey_i\, {}_Ex_i \,\mathrm{d}S_i}_{I_{xy}} + {}_E\delta z_{\mathrm{TCP}} \underbrace{\iint_S {}_Ex_i \,\mathrm{d}S_i}_{A\,r_{c,x}} \Big], \tag{5.9} $$

with the second moment of area $I_{yy} > 0$ and the position of the contact surface centroid $r_{c,x}$ in ${}_Ex$-direction. Using the same relations for the remaining term ${}_E\delta F_z$ leads to the following linear relation between infinitesimal deformations and respective wrenches:

$$ {}_E\delta\boldsymbol{\lambda}_{f,u} = \begin{pmatrix} {}_E\delta T_x \\ {}_E\delta T_y \\ {}_E\delta F_z \end{pmatrix} = c_z \underbrace{\begin{pmatrix} -I_{xx} & I_{xy} & -A\,r_{c,y} \\ I_{xy} & -I_{yy} & A\,r_{c,x} \\ -A\,r_{c,y} & A\,r_{c,x} & -A \end{pmatrix}}_{\boldsymbol{U}} \begin{pmatrix} {}_E\delta\alpha_x \\ {}_E\delta\alpha_y \\ {}_E\delta z_{\mathrm{TCP}} \end{pmatrix} = \boldsymbol{U}\, {}_E\delta\boldsymbol{x}_{f,u}. \tag{5.10} $$

The complete information on the foot's geometry, material stiffness, the effective contact area (closed contacts), and the coupling between torques and forces is contained in the symmetric matrix $\boldsymbol{U}$. It describes the gradient of foot torques/forces with respect to corresponding movements in a 3D subspace. Using a binary selection matrix $\boldsymbol{S}_f$, the relation can be written in the full 6D force/motion space of one foot, connecting differential wrenches ${}_E\delta\boldsymbol{\lambda}_f$ with differential motions ${}_E\delta\boldsymbol{x}_f$:

$$ {}_E\delta\boldsymbol{\lambda}_f = \begin{pmatrix} {}_E\delta F_x \\ {}_E\delta F_y \\ {}_E\delta F_z \\ {}_E\delta T_x \\ {}_E\delta T_y \\ {}_E\delta T_z \end{pmatrix} = \boldsymbol{S}_f^T\,\boldsymbol{U}\,\boldsymbol{S}_f \begin{pmatrix} {}_E\delta x_{\mathrm{TCP}} \\ {}_E\delta y_{\mathrm{TCP}} \\ {}_E\delta z_{\mathrm{TCP}} \\ {}_E\delta\alpha_x \\ {}_E\delta\alpha_y \\ {}_E\delta\alpha_z \end{pmatrix} = \boldsymbol{S}_f^T\,\boldsymbol{U}\,\boldsymbol{S}_f\, {}_E\delta\boldsymbol{x}_f, \tag{5.11} $$

with

$$ \boldsymbol{S}_f = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}. \tag{5.12} $$

Lemma 5.2.1. *The real-valued contact model matrix $\boldsymbol{U}$ defined in Equation* (5.10) *is negative definite for $c_z > 0$, $I_{xx} > 0$, $I_{yy} > 0$, $A > 0$.*

Proof. $\boldsymbol{U}$ is negative definite, if and only if it has only negative eigenvalues. Further, it is real symmetric and can be decomposed to

$$ \boldsymbol{U} = \boldsymbol{Q}\,\boldsymbol{M}\,\boldsymbol{Q}^T, \tag{5.13} $$

with the orthogonal matrix $\boldsymbol{Q}$. $\boldsymbol{M}$ is in diagonal form and contains the eigenvalues of $\boldsymbol{U}$ on its diagonal:

$$\boldsymbol{M} = c_z \begin{pmatrix} -I'_{xx} & 0 & 0 \\ 0 & -I'_{yy} & 0 \\ 0 & 0 & -A \end{pmatrix}. \tag{5.14}$$

Geometrically, this form is obtained by rotating the coordinate axes into the second moment of the contact area's principal axes and shifting the coordinates to the surface's centroid. From the assumptions directly follows that $\boldsymbol{U}$ is negative definite. □

The inverse contact model is given by

$$_E\delta\boldsymbol{x}_{f,u} = \boldsymbol{U}^{-1}\,{}_E\delta\boldsymbol{\lambda}_{f,u} \tag{5.15}$$

$$_E\delta\boldsymbol{x}_{f} = \boldsymbol{S}_f^T\,\boldsymbol{U}^{-1}\,\boldsymbol{S}_f\,{}_E\delta\boldsymbol{\lambda}_{f} \tag{5.16}$$

$$\boldsymbol{U}^{-1} = \frac{1}{\det(\boldsymbol{U})}\begin{pmatrix} I_{yy} - Ar_{c,x}^2 & I_{xy} - Ar_{c,x}r_{c,y} & I_{xy}r_{c,x} - I_{yy}r_{c,y} \\ I_{xy} - Ar_{c,x}r_{c,y} & I_{xx} - Ar_{c,y}^2 & I_{xx}r_{c,x} - I_{xy}r_{c,y} \\ I_{xy}r_{c,x} - I_{yy}r_{c,y} & I_{xx}r_{c,x} - I_{xy}r_{c,y} & -\frac{1}{A}(I_{xy}^2 - I_{xx}I_{yy}) \end{pmatrix} \tag{5.17}$$

$$\det(\boldsymbol{U}) = c_z(I_{xy}^2 - 2AI_{xy}r_{c,x}r_{c,y} + AI_{xx}r_{c,x}^2 + AI_{yy}r_{c,y}^2 - I_{xx}I_{yy}).$$

The contact surface parameters used for LOLA are depicted in Table 5.1. The geometry parameters are calculated for a full foot contact (all four pads). The stiffness is parametrized to $c_z = 1.9 \times 10^7\,\mathrm{N/m^3}$ and matches the properties of LOLA's compliant foot sole. This assumption of infinitely stiff ground is used for all experiments conducted as part of this thesis, as the mechanical properties of the environment are assumed unknown.

Limitations of the model The proposed contact model has a set of limitations, which are shortly addressed in this paragraph. First of all, the model is linear, which seems like a severe restriction for real-world applications. In the context of its use as a gradient for (adaptive) feedback control loops, however, nonlinear effects play a secondary role. The deformations are typically small, and the control law treats deviations as disturbance, which effectively leads to a linear approximation of the real contact. Besides, nonlinear models generally take more parameters, which need to be estimated for an unknown environment.

The contact model does consider the geometry of the contact surface. It does, however, not describe any contact dynamics caused by involved masses. The dynamics are separately considered because it is not only influenced by contact parameters but also the pose of the robot, see Section 5.2.2.

A further limitation of the model is the neglect of damping. There are several reasons for this design choice: (1) The actual damping is hard to estimate. (2) A wrong estimation for the damping may even be worse than not considering damping at all. (3) As the proposed force control acts on velocity-level, disturbances caused by damping effects are already effectively mitigated. (4) Damping information is only relevant in combination with the contact dynamics, which depend on the robot's configuration, see Section 5.2.2.

	Reference	$I_{xx}\ [m^4]$	$I_{yy}\ [m^4]$	$I_{xy}\ [m^4]$	$A\ [m^2]$	$r_{c,x}\ [m]$	$r_{c,y}\ [m]$
One Pad	centroid	8.854×10^{-7}	2.559×10^{-6}	0.0	4.25×10^{-3}	0.0	0.0
Full Foot	TCP	1.264×10^{-4}	3.206×10^{-4}	0.0	0.017	-0.096	0.0
Toe Only	TCP	6.318×10^{-5}	5.118×10^{-6}	0.0	8.5×10^{-3}	0.0	0.0

Table 5.1: Contact model parameters for a single pad, full foot contact, and toe-only contact of LOLA's feet.

Finally, the coupling of the feet in the double-support phase is neglected, i.e., influences caused by the other foot's contact control are not explicitly modeled. Simulations with LOLA have shown that a coupled contact model is less effective because this leaves less DoFs for the contact controllers.

5.2.2 Influences from Unilateral Contact Dynamics

Unlike the static case, a foot or hand contact's dynamic properties are only defined in connection with the floating-base dynamics and the structural dynamics between actuators and the contact. This is due to the contact's unilateralism, which requires the simultaneous consideration of the floating-base movement, the inertia of the connected structure, and the mechanical contact properties. The resulting dynamics depend on the configuration of the robot, can include nonlinear effects, e.g., caused by friction, and are greatly influenced by the previously unknown contact parameters. Instead of using a comprehensive model of the system — with partially unknown parameters — the experimentally identified contact dynamics (see Section 3.8.2) are analyzed concerning their uncertainty in different scenarios. The examined plant $\boldsymbol{P}_{\text{contact}}(s)$ also contains the actuator dynamics.

Figure 5.2 depicts experimentally identified transfer functions from vertical velocities of one foot to the resulting vertical forces on the same foot for different configurations and contact properties. From all directions of the full MIMO plant $\boldsymbol{P}_{\text{contact}}(s)$, this is the one with the highest uncertainty. Further plant directions are depicted in Appendix C. Several direct observations can be made from the transfer functions:

- especially the double-support plants show multiple resonance/anti-resonance pairs at low frequencies (1–10 Hz)
- all resonances/anti-resonances are shifted to lower frequencies on soft ground, i.e., there are no fixed eigenfrequencies observable both on soft- and hard ground
- the static gain decreases for softer ground
- there are significant differences of the plant for single and double support

Furthermore, the static gains

$$\boldsymbol{P}_u = \lim_{s \to 0} \boldsymbol{S}_f \, \boldsymbol{P}_{\text{contact}}(s) \, \boldsymbol{S}_f^T \, \boldsymbol{U}_{\text{full}}^{-1} \tag{5.18}$$

calculated with the assumed inverse contact model $\boldsymbol{U}_{\text{full}}^{-1}$ for full foot contact should theoretically equal identity. However, $\boldsymbol{P}_u$ calculated from the experimental plant estimates is not even diagonal. This can be explained with a caused floating-base inclination when the feet are excited at low frequencies. The inclination also changes the feet's actual contact surface and thus the contact model matrix $\boldsymbol{U}$. Consequently, there is an uncertainty in the parameters of $\boldsymbol{U}$ even on level ground, where full surface contact is possible. The following additional hints are known from the structure of the contact problem:

- the contact is defined by a series connection of LOLA's compliant foot sole and the ground; the total stiffness may, therefore, only be less than the known foot sole stiffness
- the combined damping of foot sole and ground has an upper bound defined by the foot sole; thus, the low damping of LOLA's foot sole reduces the uncertainty of the total contact damping (ground damping must be lower to have an effect)
- for lower stiffness, the damping ratio increases for fixed damping property of a material; low-stiffness materials often have high damping.

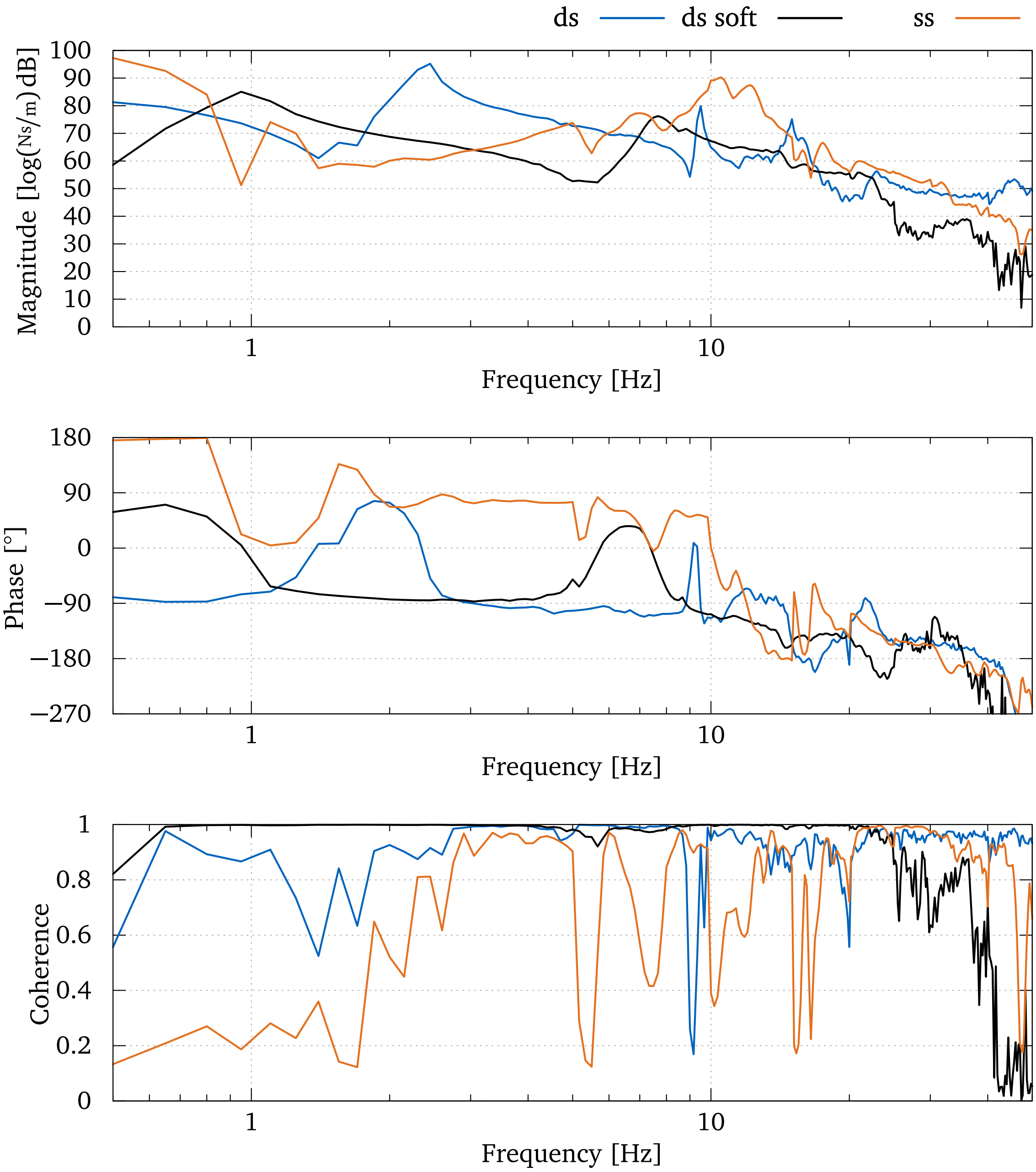


Figure 5.2: Estimated measured transfer functions from vertical foot velocity to vertical contact forces $-P_{\text{contact},3,3}(s)$ for double support (ds), double support on soft ground (ds soft), and single support (ss). In addition, a magnitude-squared coherence estimate is shown. The foot velocities and contact torques have inverted directions (see Section 3.4.2). Thus, the transfer function to negative contact forces is shown to avoid unintuitive phase readings. A positive phase does not violate physical consistency due to the velocity-level plant input.

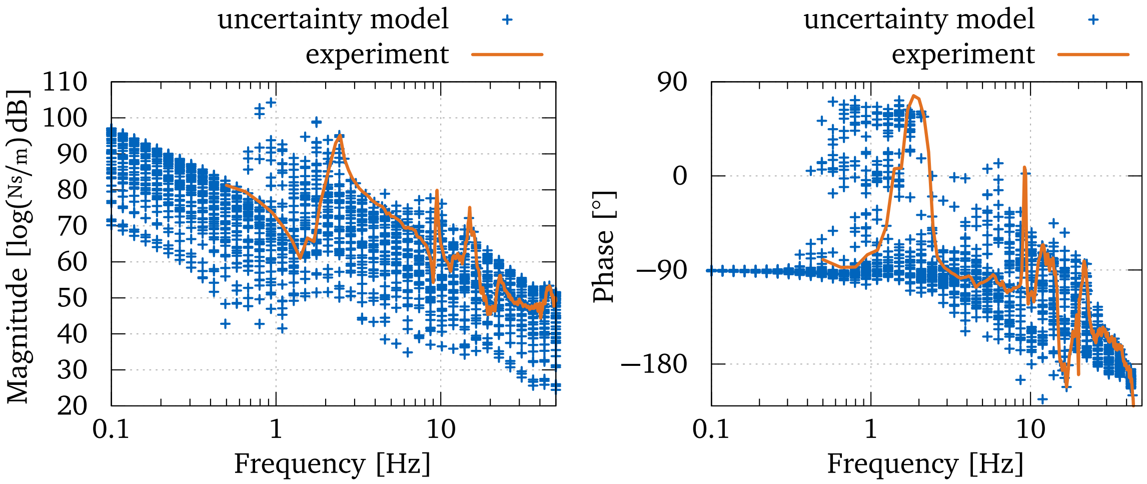


Figure 5.3: Modeled uncertainty for a variation of the double-support transfer function $-P_{\text{contact},3,3}(s)$. The stiffness is reduced up to a factor of 25, the damping up to a factor of 2. The plot displays a subset of all possible uncertainty transfer function points to visualize the estimated plant uncertainty over frequency.

Combining the observations from the experiments with known clues, the uncertainty in the contact dynamics can be modeled for a wide range of real-world contacts. The expected uncertainty for contacts ranging to 1/25 of the foot-sole stiffness and 1/2 of its damping is shown in Figure 5.3. This uncertainty model is generated from a white-box system description of the estimated transfer function in double support. The parameters of this system are randomly varied according to the known constraints, which yields the set of possible transfer functions, i.e., the uncertainty model.

The uncertainty model already shows several beneficial plant properties for a robust control design. First, the infinity norm of the uncertain plant is bounded, and the supremum is located at low frequencies, where sufficient phase-reserve is available. Second, all anti-resonances and resonances are only shifted without swapping their place. Thus, the location of the critical phase is also only shifted.

A theoretical stability analysis conducted in Section 5.2.4 supports the observations of these experimentally identified and stability-related properties.

5.2.3 Robust Hybrid Force/Motion Control Approach

Following the deployed IK approach, the proposed force control scheme modifies the *velocity* of a foot to achieve the desired contact wrench ${}_E\boldsymbol{\lambda}_{f,\text{d}}$. Generally, higher-order derivatives (acceleration-level, for example) are possible, yet this reduces the reachable bandwidth in trade for smoother trajectories. Every foot has a dedicated controller, which acts independently of the other foot's controller. Only a subspace of the full foot task-space vector — see the contact model in Section 5.2.1 — is used for force control; the rest is motion-controlled, which makes it a hybrid force/motion control scheme [54]. Furthermore, the control mode of a foot changes from hybrid force/motion control to pure motion control at the beginning of the swing phase. This switching of control modes is based on the control mode blending factors $\boldsymbol{\beta}$. The basic control design has previously been published in [295].

Starting from a first-order error dynamics

$$ {}_E\dot{\boldsymbol{e}}_f + G_\lambda\, {}_E\boldsymbol{e}_f = 0 \qquad \text{with} \qquad {}_E\boldsymbol{e}_f = \boldsymbol{S}_f\, {}_E\boldsymbol{\lambda}_{f,\text{d}} - \boldsymbol{S}_f\, {}_E\boldsymbol{\lambda}_{f,\text{m}} \tag{5.19} $$

in the subspace of the contact model Equation (5.10) and the scalar gain $G_\lambda \in \mathbb{R}_{>0}$, a relation for the desired change of contact wrenches ${}_E\delta\dot{\boldsymbol{\lambda}}_{f,u} \in \mathbb{R}^3$ results:

$$ {}_E\delta\dot{\boldsymbol{\lambda}}_{f,u} = \boldsymbol{S}_f\, {}_E\dot{\boldsymbol{\lambda}}_{f,\mathrm{d}} + G_\lambda(\boldsymbol{S}_f\, {}_E\boldsymbol{\lambda}_{f,\mathrm{d}} - \boldsymbol{S}_f\, {}_E\boldsymbol{\lambda}_{f,\mathrm{m}}). \tag{5.20} $$

The wrench distribution, see Section 4.3, does not provide the derivative of the desired wrench ${}_E\dot{\boldsymbol{\lambda}}_{f,\mathrm{d}}$. Thus, a discrete derivative implementation is used for the feedforward part

$$ {}_E\delta\dot{\boldsymbol{\lambda}}_{f,u} = G_d\ \mathrm{ddt}(T_l,\, \boldsymbol{S}_f\, {}_E\boldsymbol{\lambda}_{f,\mathrm{d}}) + G_\lambda(\boldsymbol{S}_f\, {}_E\boldsymbol{\lambda}_{f,\mathrm{d}} - \boldsymbol{S}_f\, {}_E\boldsymbol{\lambda}_{f,\mathrm{m}}), \tag{5.21} $$

with the filter time-constant $T_l > 0$ and additional gain $0 < G_d < 1$. This feedforward gain is required to limit the amplification of noise inserted from the balance controller.

From the definition of the contact model in Section 5.2.1 follows a relation between changes in the contact wrenches and corresponding motion gradients of the foot. Hence, the control law for the ground reaction forces follows from the combination of Equation (5.15) and Equation (5.21):

$$ {}_E\delta\boldsymbol{v}_{f,u} = \boldsymbol{U}^{-1}\left(G_d\ \mathrm{ddt}(T_l,\, \boldsymbol{S}_f\, {}_E\boldsymbol{\lambda}_{f,\mathrm{d}}) + G_\lambda\,(\boldsymbol{S}_f\, {}_E\boldsymbol{\lambda}_{f,\mathrm{d}} - \boldsymbol{S}_f\, {}_E\boldsymbol{\lambda}_{f,\mathrm{m}})\right). \tag{5.22} $$

For now the contact model matrix $\boldsymbol{U}^{-1}$ is constant and parametrized with the geometry of the total foot surface. Furthermore, it is assumed that $\boldsymbol{U}^{-1}$ contains all (static) coupling effects between the individual wrench directions.

5.2.4 Robust Stability Analysis

The relatively simple approach described above provides robust stability of the contact force control loop, which is proven in the following. In this section, the work in [295] is extended via a robust stability analysis. Only the force-controlled subspace of the full MIMO plant $\boldsymbol{P}_{\mathrm{contact}}(s)$ is considered in the following with

$$ \boldsymbol{P}_{\mathrm{contact},u}(s) = \boldsymbol{S}_f\, \boldsymbol{P}_{\mathrm{contact}}(s)\, \boldsymbol{S}_f^T, \tag{5.23} $$

see the definition of the contact model in Equation (5.10). The other directions are not affected by the force controller, but result from the planned position trajectories of the feet, see also Section 5.2.1. The plant of the force controller — refer to Figure 3.9 — can be separated into an integrator, the actuator dynamics, the mechanical system, and the constant contact model matrix $\bar{\boldsymbol{U}}$:

$$ \boldsymbol{P}_{\mathrm{contact},u}(s) = \bar{\boldsymbol{U}}\, \boldsymbol{P}_{\mathrm{mech}}(s)\, \boldsymbol{P}_{\mathrm{act}}(s)\, \frac{1}{s}. \tag{5.24} $$

It is assumed that the joint actuator control is asymptotically stable; the mechanical system is a real passive system.

Lemma 5.2.2 (see [190, p.228])**.** *Let the transfer function $\boldsymbol{P}(s)$ with static part*

$$ \boldsymbol{P}_s = \lim_{s\to 0} \boldsymbol{P}(s), \tag{5.25} $$

describe an asymptotically stable time-invariant MIMO plant with equal number of inputs and outputs k. Further let $\mathcal{P}$ be a closed negative feedback loop consisting of $\boldsymbol{P}(s)$ and an integral controller

$$ \boldsymbol{C}(s) = a\, \boldsymbol{K}\, \frac{1}{s} \tag{5.26} $$

with scalar gain a, matrix gain $\boldsymbol{K} \in \mathbb{R}^{k \times k}$, and the complex frequency parameter s. Then, the closed loop system

$$\boldsymbol{P}_{\mathcal{P}}(s) = \boldsymbol{P}(s)\,\boldsymbol{C}(s)\,(\boldsymbol{I} + \boldsymbol{P}(s)\,\boldsymbol{C}(s))^{-1} \tag{5.27}$$

is asymptotically stable for an interval $0 < a \leq a^+$, if and only if

$$\operatorname{Re}\{\lambda_i\{\boldsymbol{P}_s\,\boldsymbol{K}\}\} > 0, \quad i = 1, 2, ..., k, \tag{5.28}$$

where λ_i are the eigenvalues of $\boldsymbol{P}_s\,\boldsymbol{K}$.

Proof. The lemma is proven in [190, p.228]. □

Theorem 5.2.3. *Let*

$$\boldsymbol{P}(s) = \bar{\boldsymbol{U}}\,\boldsymbol{P}_d(s)\,\frac{1}{s} \tag{5.29}$$

$$\lim_{s \to 0} \boldsymbol{P}_d(s) = \boldsymbol{I} \tag{5.30}$$

$$||\bar{\boldsymbol{U}}|| \leq ||\bar{\boldsymbol{U}}^+||, \tag{5.31}$$

be an uncertain time-invariant MIMO plant with equal number of inputs and outputs k, asymptotically stable dynamics $\boldsymbol{P}_d(s)$, bounded uncertain coupling matrix $\bar{\boldsymbol{U}} \in \mathbb{R}^{k \times k}$, and the complex frequency parameter s. Further let $\mathcal{P}$ be a closed negative feedback loop consisting of $\boldsymbol{P}(s)$ and the controller

$$\boldsymbol{C}(s) = \boldsymbol{U}^{-1}\,G, \tag{5.32}$$

with gain $G \in \mathbb{R}_{>0}$ and the matrix $\boldsymbol{U}^{-1} \in \mathbb{R}^{k \times k}$. Then there exists a gain value G which makes the closed loop system $\mathcal{P}$ of the uncertain plant asymptotically stable, if and only if

$$\operatorname{Re}\{\lambda_i\{\bar{\boldsymbol{U}}\,\boldsymbol{U}^{-1}\}\} > 0, \quad i = 1, 2, ..., k. \tag{5.33}$$

This includes the special case

$$\bar{\boldsymbol{U}}\,\boldsymbol{U}^{-1} = p\,\boldsymbol{I}, \tag{5.34}$$

for some $0 < p < p^+$.

Proof. By rearranging, the plant and control law can be rewritten to

$$\boldsymbol{P}(s)' = \frac{\bar{\boldsymbol{U}}}{||\bar{\boldsymbol{U}}||}\,\boldsymbol{P}_d(s) \tag{5.35}$$

$$\boldsymbol{C}(s)' = ||\bar{\boldsymbol{U}}||\,\boldsymbol{U}^{-1}\,G\,\frac{1}{s}, \tag{5.36}$$

without changing the open loop system $\boldsymbol{P}(s)\boldsymbol{C}(s) = \boldsymbol{P}(s)'\boldsymbol{C}(s)'$. From Lemma 5.2.2 immediately follows asymptotic stability of the closed loop system for some $||\bar{\boldsymbol{U}}|| \leq ||\bar{\boldsymbol{U}}^+||$, if and only if Equation (5.33) holds. □

Corollary 5.2.3.1. *Theorem 5.2.3 shows that there exists a gain G_λ for the control law in Equation* (5.22)*, which makes the closed-loop with plant Equation* (5.24) *asymptotically stable for any bounded contact $||\bar{\boldsymbol{U}}|| \leq ||\bar{\boldsymbol{U}}^+||$, if the assumed contact geometry matches the real one, i.e., $\bar{\boldsymbol{U}}\,\boldsymbol{U}^{-1} = p\,\boldsymbol{I}$ for some $0 < p < p^+$.*

This means that only an upper bound on the contact stiffness is required for robust stability in the case of a known contact surface. The following considerations show that contact surface equality is actually not required in general.

Lemma 5.2.4. *The product $\boldsymbol{AB}$ of two symmetric positive definite matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ has only real positive eigenvalues.*

Proof. $\boldsymbol{B}$ has a positive definite square root $\boldsymbol{B}^{1/2}$ with

$$\boldsymbol{B}^{1/2}\boldsymbol{B}^{1/2} = \boldsymbol{B}, \tag{5.37}$$

and the matrix $\boldsymbol{B}^{1/2}\boldsymbol{A}\boldsymbol{B}^{1/2}$ is consequently also positive definite. Due to similarity transformation $\boldsymbol{AB}$ has the same real positive eigenvalues as the positive definite matrix $\boldsymbol{B}^{1/2}\boldsymbol{A}\boldsymbol{B}^{1/2}$ [18, p.256]:

$$\boldsymbol{AB} = \boldsymbol{A}\boldsymbol{B}^{1/2}\boldsymbol{B}^{1/2} = \boldsymbol{B}^{-1/2}(\boldsymbol{B}^{1/2}\boldsymbol{A}\boldsymbol{B}^{1/2})\boldsymbol{B}^{1/2}. \tag{5.38}$$

□

Theorem 5.2.5. *There exists a gain $G_\lambda > 0$ for the control law in Equation* (5.22)*, which makes the closed-loop with uncertain plant Equation* (5.24) *asymptotically stable for all bounded contacts $||\bar{\boldsymbol{U}}|| \leq ||\bar{\boldsymbol{U}}^+||$, i.e., bounded stiffness and surface area, if $\bar{\boldsymbol{U}}$, $\boldsymbol{U}^{-1}$ match the structure defined by Equations* (5.10) *and* (5.17) *with arbitrary parameters.*

Proof. From Theorem 5.2.3 follows that the control loop is asymptotically stable, if and only if

$$\mathrm{Re}\{\lambda_i\{\bar{\boldsymbol{U}}\boldsymbol{U}^{-1}\}\} > 0, \quad i = 1,2,3. \tag{5.39}$$

The assumed inverse contact model $\boldsymbol{U}^{-1}$ uses a contact model matrix $\boldsymbol{U}$ with different parameters than $\bar{\boldsymbol{U}}$. From Lemma 5.2.1 follows that $-\bar{\boldsymbol{U}}$ and $-\boldsymbol{U}$ are symmetric positive definite matrices. Furthermore, the inverse of $-\boldsymbol{U}$ is again symmetric positive definite. From Lemma 5.2.4 follows $(-\bar{\boldsymbol{U}})(-\boldsymbol{U})^{-1} = \bar{\boldsymbol{U}}\boldsymbol{U}^{-1}$ has only real, positive eigenvalues. □

Unfortunately, Theorem 5.2.5 can not be used to make a statement on the (asymptotic) stability when $\dot{\bar{\boldsymbol{U}}} \neq 0$, i.e., when the contact surface or stiffness changes. Nevertheless, $\dot{\bar{\boldsymbol{U}}} \neq 0$ is limited to the quite short contact transition phase and the changing contact model can therefore be considered a disturbance of the system. Furthermore, the theorems assume asymptotic stability of the joint controllers and mechanical system.

Indeed, the described control scheme has been used extensively for Lola [292, 295] and has shown its robustness to uncertain terrain, including grass, asphalt, and cobblestone, see https://youtu.be/cNkQT2SUegE/and https://youtu.be/pmtKv8VEItY/.

5.2.5 Extension via Lag Reduction

In this section, an extension of [295] leading to increased performance is described. The crossover frequency of the closed loop is limited by the value of G_λ, which in turn is limited by the intended robustness to the plant's magnitude uncertainties (the gain scales the open-loop transfer function). The performance of the control loop can only be increased (at similar robustness) by increasing the open-loop phase near the crossover frequency.

Lead Compensator A standard way to introduce positive phase shift is the lead compensator, see Figure 5.4. The positive phase-shift has its maximum between the location of the pole/zero pair of the compensator. In addition, the lead compensator amplifies higher frequencies to compensate magnitude roll-off. Per definition, a lead compensator introduces

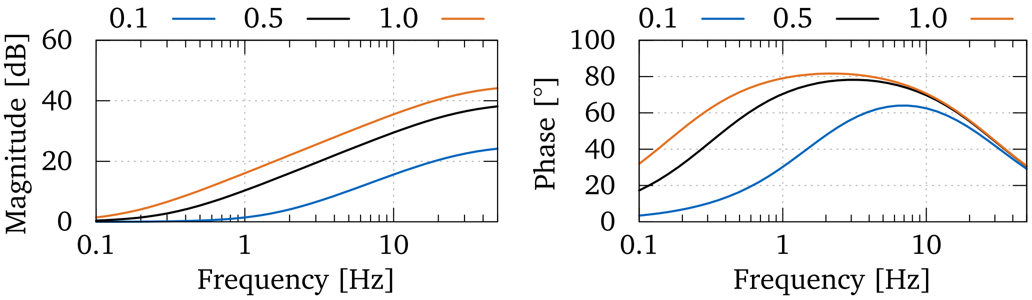


Figure 5.4: Bode diagram of a lead compensator $\frac{gs+1}{T_l s+1}$ for $T_l = \frac{1}{2\pi 30\,\mathrm{Hz}}$ and different gains g.

phase lead (positive) and reduces the lag (negative phase) of a system. With a first-order lead compensator, the force control law (Equation (5.22)) in Laplace domain changes to

$$\boldsymbol{C}(s) = \frac{s\boldsymbol{X}_{f,u}(s)}{\boldsymbol{E}_f(s)} = \boldsymbol{U}^{-1}\,\mathrm{diag}\left(\frac{G_{\dot\lambda}\,s + G_\lambda}{T_l\,s+1}\right), \tag{5.40}$$

with additional derivative gain $G_{\dot\lambda} \in \mathbb{R}_{>0}$, and filter time constant $T_l > 0$. In theory, higher order lead compensators can be used. This, however, also puts higher demands on the signal-to-noise ratio of the sensor signals. The implementation in discrete time domain with the already described feedforward approach yields

$$\begin{aligned} {}_E\delta\dot{\boldsymbol{\lambda}}_{f,u} &= G_d\ \mathrm{ddt}(T_l,\, \boldsymbol{S}_f\,{}_E\boldsymbol{\lambda}_{f,\mathrm{d}}) + G_{\dot\lambda}\,\mathrm{ddt}(T_l, \boldsymbol{S}_f\,{}_E\boldsymbol{\lambda}_{f,\mathrm{d}} - \boldsymbol{S}_f\,{}_E\boldsymbol{\lambda}_{f,\mathrm{m}}) \\ &\quad + G_\lambda\,\mathrm{lpf1}(T_l, \boldsymbol{S}_f\,{}_E\boldsymbol{\lambda}_{f,\mathrm{d}} - \boldsymbol{S}_f\,{}_E\boldsymbol{\lambda}_{f,\mathrm{m}}), & (5.41) \\ {}_E\delta\boldsymbol{v}_{f,u} &= \boldsymbol{U}^{-1}\,{}_E\delta\dot{\boldsymbol{\lambda}}_{f,u}. & (5.42) \end{aligned}$$

The value of the lead-compensator time constant T_l also depends on the noise and quantization of the measured contact wrenches. Ideally, T_l is placed to get the maximum phase shift in the critical frequency range 7–15 Hz. However, this requires very low values of T_l with accompanying high sensor noise amplification. Experiments with different values for the lead-compensator gain are described in Section 5.2.7.

Robust Stability Discussion The lead-compensator does not contain poles near the imaginary axis nor poles with positive real part. With higher gains, the corresponding poles of the closed loop move to the left. This means Theorem 5.2.5 on the asymptotical stability of the closed loop applies unchanged to the modified control law Equation (5.41), if and only if the gain $G_{\dot\lambda} < l^+$ is bounded with some $l^+ > 0$ [190].

5.2.6 Time-Integration and Tuning

The control scheme modifies the trajectories calculated from the balance control module described in Chapter 4. For each foot f, a foot velocity modification is calculated based on a hybrid force/motion control scheme (see Appendix A.5) and the control effort Equation (5.42):

$${}_E\Delta\boldsymbol{v}_{f,\lambda x}[n] = \mathrm{hybridCtl}\Big(\beta_f,\, \boldsymbol{S}_f^T\,{}_E\delta\boldsymbol{v}_{f,u},\, {}_E\Delta\boldsymbol{x}_{f,\lambda x}[n-1]\Big), \tag{5.43}$$

with the control mode blending factor β_f, the contact model selection matrix $\boldsymbol{S}_f$ (see Equation (5.12)), and the integrated modifications from the last time step ${}_E\Delta\boldsymbol{x}_{f,\lambda x}[n-1]$. The quantities are described in the end-effector FoR E to match the definition of the contact model. The time-integration of the foot's velocity modification is done via

$$ {}_{E}\Delta\boldsymbol{x}_{f,\lambda x} = \begin{bmatrix} \mathrm{intA}\left(\left[\boldsymbol{I}_3\ \boldsymbol{0}_{3\times3}\right]{}_{E}\Delta\boldsymbol{v}_{f,\lambda x},\ {}_{W}\boldsymbol{A}_{E}\right) \\ \mathrm{int}\left(\left[\boldsymbol{0}_{3\times3}\ \boldsymbol{I}_3\right]{}_{E}\Delta\boldsymbol{v}_{f,\lambda x}\right) \end{bmatrix}. \tag{5.44} $$

The translational modification velocities are transformed to the W FoR before integration to keep the modification constant in the world FoR when the orientation of the foot changes. For the rotations, the modifications are integrated in the foot-attached FoR. Note that the integration of rotational velocities is only approximative, see Appendix A.2. Finally, the foot modifications ${}_{E}\Delta\boldsymbol{x}_{f,\lambda x}$ for both feet are applied to the input trajectories of the contact control module to get

$$ \left(\boldsymbol{X}_{\mathrm{d}}', \boldsymbol{V}_{\mathrm{d}}'\right) = {}_{E}\,\mathrm{modifyAll}_{f}\left(\boldsymbol{X}_{\mathrm{bl}}, \boldsymbol{V}_{\mathrm{bl}}, {}_{E}\Delta\boldsymbol{x}_{f,\lambda x}, {}_{E}\Delta\boldsymbol{v}_{f,\lambda x}\right). \tag{5.45} $$

Common-Mode Rejection Equations (5.43) to (5.45) are generally sufficient to modify the task-space trajectories based on the contact force control efforts. However, a characteristic caused by the unilateral contact is the indifference of the contact wrenches to common vertical modifications of both feet in the same direction. In stillstand, the force controllers may, for instance, set a constant vertical velocity on both feet based on model or sensor inaccuracies, which causes a drift of the robot's pose, but does not influence the contact forces.

Although this drift is compensated by CoM tracking and inclination control (see Chapter 4), the time-integration of CoM tracking and contact force controllers uses different integrators and only the sum of the modifications is zero. Due to the large values, inaccuracies may occur in the final trajectories. Consequently, the contact control approach should not accumulate vertical modifications of the feet that point in the same direction. To avoid this effect, the common-mode translational modification for foot f is calculated:

$$ {}_{E}\Delta z_{\mathrm{cm},f,\lambda x} = \frac{{\boldsymbol{e}_z}^T}{1+\beta_{\check{f}}}\left(\left[\boldsymbol{I}_3 \quad \boldsymbol{0}_{3\times3}\right]{}_{E}\Delta\boldsymbol{x}_{f,\lambda x} + \beta_{\check{f}}\left[{}_{E}\boldsymbol{A}_{\check{E}} \quad \boldsymbol{0}_{3\times3}\right]{}_{\check{E}}\Delta\boldsymbol{x}_{\check{f},\lambda x}\right). \tag{5.46} $$

This quantity is the β-weighted sum[1] of the vertical modifications of both feet, represented in the FoR of foot f. It describes the share of both feet's vertical modifications with the same sign. Consequently, the control approach should minimize this common-mode modification. To achieve this, Equation (5.43) is modified to yield

$$ {}_{E}\Delta\boldsymbol{v}_{f,\lambda x}[n] = \mathrm{hybridCtl}\left(\beta_f,\, \boldsymbol{S}_f^T\, {}_{E}\delta\boldsymbol{v}_{f,u} - \frac{{}_{E}\Delta z_{\mathrm{cm},f,\lambda x}[n-1]}{T_{\mathrm{decay}}}\begin{pmatrix}\boldsymbol{e}_z \\ \boldsymbol{0}_3\end{pmatrix},\, {}_{E}\Delta\boldsymbol{x}_{f,\lambda x}[n-1]\right), \tag{5.47} $$

with the common-mode decay time constant $T_{\mathrm{decay}} > 0$.

Tuning The parameters of the control approaches described above must be tuned experimentally on the real robot. At first, $G_{\dot{\lambda}} = 0$ is used and the gain G_{λ} is slowly increased until a desired stability margin or response in the time domain is achieved. It is important to test the gains during walking and with different step parameters to evaluate the performance in single- and double support and different configurations. Moreover, tuning must be done on stiff ground to reach guaranteed stability of the gains for all ground stiffnesses. When a suitable G_{λ} is found, $G_{\dot{\lambda}}$ is increased until either the performance of the controller decreases, or the noise propagated to the actuators leaves acceptable ranges. The noise amplification can additionally be adjusted via the filter time constant T_l. The decay time constant T_{decay} should be set to relatively high values to avoid performance degradation and is chosen to $T_{\mathrm{decay}} = 1.0\,\mathrm{s}$ in this thesis.

[1] The weighing is imperfect as the expression is not weighted with β_f to avoid division by zero, assuming $\beta_f = 1$. In practice, this does not make a difference as common-mode rejection is only relevant when $\beta_{\mathrm{LF}} = \beta_{\mathrm{RF}} = 1$.

$G_{\dot{\lambda}}$	${}_{W}e^{\text{CoM}}_{T_x,\text{RMS}}$ [Nm]	${}_{W}e^{\text{CoM}}_{T_y,\text{RMS}}$ [Nm]
0	29.7	12.9
0.1	28.7	10.6
0.15	29.6	10.7
0.2	29.4	**10.2**
0.3	30.0	10.6
0.4	**28.6**	10.3

Table 5.2: The RMS contact torque tracking errors for walking on the spot with $G_\lambda = 24\,s^{-1}$ and different lead compensator gains. Bold values represent the optimum of a column.

5.2.7 Experimental Validation

The contact force control scheme is experimentally evaluated on LOLA, performing measurements while the robot is stepping in place — standard forward walking imposes higher disturbances on the robot and may cause timing imperfections on the contacts. The more periodic and reproducible stamping motion is chosen because only the contact controller's performance shall be evaluated. Vertical accelerations of the CoM are deactivated with $G_{\text{CoM}} = 0$ to isolate effects of the contact controllers. The gain G_λ is determined as indicated in the previous section. For the feedforward term, a relatively low value $G_d = 0.05$ is required on LOLA due to the numerical differentiation of wrenches calculated from IMU measurements. Higher values propagate noise and vibrations to the force controllers and do not provide substantially higher performance. The filter time-constant for lead compensator and feed forward term is chosen to $T_l = \frac{1}{2\pi 20.0\,\text{Hz}}$.

The total CoM wrench RMS tracking errors for different lead compensator gains are depicted in Table 5.2. Small values for $G_{\dot{\lambda}}$ already reduce the tracking errors of torques around the x- and y-axis by 3.4 % and 17.8 % respectively. Higher values do not further improve the tracking errors or even lead to worse results. Further, low gain values have the advantage of reduced noise amplification. Consequently, $G_\lambda = 24\,\text{s}^{-1}$, $G_{\dot{\lambda}} = 0.1$ is selected for the foot contact controllers. These values are consistently used in all experiments described in this thesis. The relatively high torque tracking error around the x-axis compared to the error around the y-axis can be explained with the robot's configuration. Timing imperfections (late contacts) primarily cause torque errors around the x-axis because the feet are aligned along the y-axis when stepping in place. Exemplary desired and actual contact wrenches for a foot during the stamping motion are visualized in Figure 5.5. It is noticeable that the vertical forces contain (damped) oscillations, evident in the single-support phase. These vibrations are discussed in detail in Section 4.5.

The validity of the proposed contact model has been experimentally evaluated on LOLA 2020 (fig. 2.1c) via comparison with the previous approach presented in [38]. For a default walking gait with an unexpected obstacle of 4 cm height, the RMS tracking errors of the total CoM torque were compared. For similar target error dynamics, the method with the proposed contact model has reached a 3 % and 9 % improvement of torque tracking for x-, and y-axis, respectively. The experiment uses a different task-space definition and slightly different implementation of the force control scheme than the method described above. For more information, refer to [295]. Unfortunately, a comparison on up-to-date hardware is no longer possible due to irreversible soft- and hardware changes. In all described experiments in the context of this thesis, the proposed force control scheme is active. Further experiments with this method are described in several other parts of this work, notably in Chapter 7.

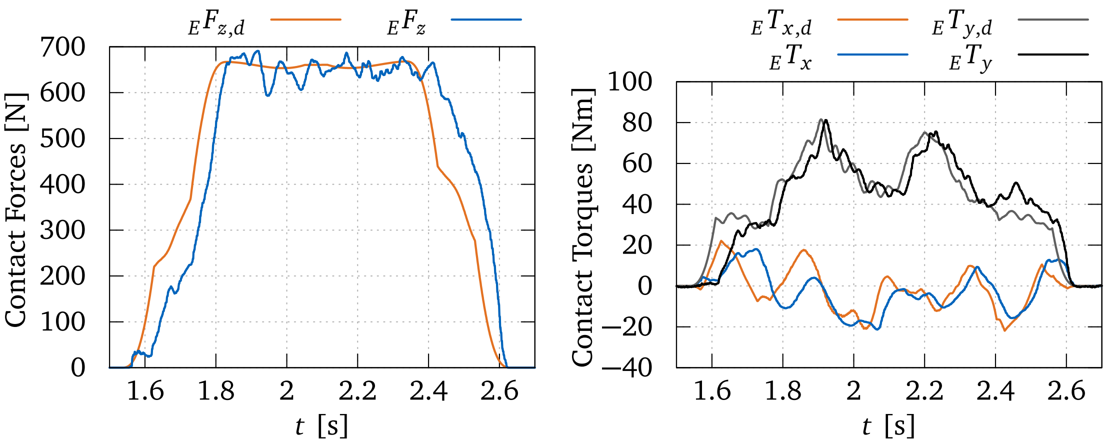


Figure 5.5: Desired and actual contact wrench on the right foot for stamping on stiff ground.

5.2.8 Discussion in the Context of Related Work

Fujimoto and Kawamura [86] initially proposed using a hybrid position/force control scheme for the foot torques to track a ZMP reference. Hybrid position/force control was originally formulated for torque-controlled manipulators [54]. The application to position-controlled robots led to the development of hybrid position/force control with inner position-loop [274]. This control scheme is deployed in a large number of position-controlled humanoid robots. [135] uses a 2-DoF force controller for the ground reaction torques. The controller maps errors in the contact torques to the joints' velocities and is specified as feedback filters in the frequency domain. A similar approach with additional position constraint on the modifications is described in [137]. Nishiwaki and Kagami [211] use acceleration-level feedback to attain desired ground reaction forces. A quite similar approach with additional position constraint is described by [201]. Acceleration-level feedback typically produces smoother output trajectories, which are easier to follow by the joint controllers. However, that typically comes at the price of reduced closed-loop dynamics.

The velocity-level force control methods described in the literature are conceptionally similar to the robust control scheme described in Equation (5.21). However, none of the approaches described above uses an explicit contact model. Instead, the controllers' gains are tuned accordingly, and geometric coupling between individual directions is ignored. Furthermore, explicit contact models provide the possibility to consider the exact contact geometry. Compared to the previous force control strategy [38] for Lola, the contact surface with the ground is parametrized explicitly, which enables the adaptation to partial contact control (see Section 5.3) and an analysis of stability.

The proposed contact model was first published in 2017 [295]. In parallel, a similar control approach was proposed in [228], using a velocity-level force controller with contact model matrix to approximate a static nonlinear footsole model. Still, this contact model is static and not explicitly parametrized with the contact geometry.

Although the basic velocity-level force control concepts for bipeds have often been used in many publications, the robustness has, to the author's best knowledge, not been investigated concerning contact stiffness- and contact surface uncertainties. There is, however, literature on robust control design for torque-controlled quadrupeds [79]. Also, the work at hand investigates the effect of an additional lead compensator on the robustness and performance of contact force control for bipeds. Of course, the presented methods only apply to biped robots with position-controlled joints. A theoretical comparison of differences and similarities to robots with torque-controller joints is given in Section 7.5.

The robustness of the presented force controller to contacts with arbitrary mechanical properties is remarkable and has been proven through experiments and theoretical considerations. It is commonly known that such control schemes with force/position causality[2] may suffer from instability problems for very stiff contacts. However, the results show that if the total stiffness to the environment is limited, for example, by having a compliant element in series, and the controller is tuned for the maximum contact stiffness, the control loop is robust to all possible contacts. This also applies to robot manipulators, see [294].

Interestingly, some desired properties on the compliant contact material can be inferred from the theoretical and experimental analysis. First of all, the material should provide sufficiently high internal damping, which is usually the case for softer materials. Regarding the stiffness, the following trade-offs have to be kept in mind: On the one hand, a low stiffness has an advantage regarding impacts with the environment. On the other hand, a low stiffness puts higher demands on sensors and actuators (required higher gains) for a predefined closed-loop crossover frequency. Due to nonlinear (sensor) effects, the stiffness optimization will most likely require comparisons of the closed-loop performance. Note that the gains must be re-tuned for different materials.

A disadvantage of the control method's robustness is a reduced force tracking performance for soft grounds of unknown stiffness. One possibility to reduce this tradeoff is presented in Section 5.4.

5.3 Sensor-Based Contact Model Updating

Usually, full foot contact with the environment is assumed for force-control approaches on humanoid robots. However, in scenarios with partial contact between the robot's feet and the ground, the actual contact dynamics can differ. Based on the contact model and control scheme introduced in Section 5.2, a new approach to consider the shape and location of the actual contact surface in the ground reaction force-control scheme is proposed. The general approach has previously been published in [292]. It is migrated to the improved force control concept with lead compensator and re-validated on current hardware of LOLA. Moreover, the viability of the concept in combination with a tactile foot sole is validated in simulation.

5.3.1 Control Concept

In Section 5.2, the inverse contact model matrix $\boldsymbol{U}^{-1}$ is considered constant, which enables robust stability for varying contact surfaces. However, when the actual contact surface is smaller than the foot, the performance of the controller degrades due to the lower wrench gradient w.r.t. foot movements. Thus, the contact matrix must be time-varying with a distinct representation $\boldsymbol{U}_f^{-1}(t)$ for each foot f to mitigate these limitations. Equation (5.15) then changes to

$$ {}_E\delta\boldsymbol{x}_{f,u} = \boldsymbol{U}_f^{-1}(t)\,{}_E\delta\boldsymbol{\lambda}_{f,u}. \tag{5.48}$$

Differentiating this relation with respect to time yields

$$ {}_E\delta\boldsymbol{v}_{f,u} = \dot{\boldsymbol{U}}_f^{-1}(t)\,{}_E\delta\boldsymbol{\lambda}_{f,u} + \boldsymbol{U}_f^{-1}(t)\,{}_E\delta\dot{\boldsymbol{\lambda}}_{f,u}. \tag{5.49}$$

[2] Often also called admittance control, although strictly speaking no admittance model is used for the control but direct force control with arbitrary reference trajectory.

The first term of Equation (5.49) leads to practical problems for the controller design. Depending on the contact transition (opening, closing), $\dot{\boldsymbol{U}}_f^{-1}$ may either be positive or negative definite. Furthermore, there are no known limits for the matrix norm $||\dot{\boldsymbol{U}}_f^{-1}||$ (contacts may open and close very fast), which may cause unbounded positive feedback and potential instability of the system. Besides, the derivative of the contact surface can not be directly measured. Consequently, only feedback via the (time-dependent) inverse contact matrix itself is applied:

$$ {}_E\delta\boldsymbol{v}_{f,u} = \boldsymbol{U}_f^{-1}(t)\,{}_E\delta\dot{\boldsymbol{\lambda}}_{f,u}. \tag{5.50}$$

The only difference to the control laws described in Section 5.2.3 is the time dependency of $\boldsymbol{U}_f^{-1}(t)$. In the following sections, techniques for the computation of the time-varying contact model from sensor data are proposed.

In the case of a quasi-static plant and controller with $\bar{\boldsymbol{U}}_f$, $\boldsymbol{U}_f^{-1}$ piecewise constant, some statements on the stability of the concept can be made. From Theorem 5.2.5 follows the existence of a stable gain for each $\boldsymbol{U}_f^{-1}(t)$, but the gain then also depends on the time. For all combinations of actual and assumed contact geometry, the circular gain of the overall system must be below an arbitrary threshold for the controller to be stable with a constant set of gains. As a result, $\boldsymbol{U}_f^{-1}(t)$ must additionally be bounded with some $||\boldsymbol{U}_f^{-1}(t)|| < ||\boldsymbol{U}_f^{-1+}||$. This means that the assumed contact area must have a lower positive bound for robust stability when the gain is tuned for the lowest contact area. However, gain tuning is usually done for full surface contact. To achieve robust stability for this parametrization, the more restrictive condition $||\bar{\boldsymbol{U}}_f(t)\boldsymbol{U}_f^{-1}(t)|| \leq 1$ must hold for $\boldsymbol{U}_f^{-1}(t)$. In other words, this means the actual contact surface must be less or equal to the assumed contact surface in the inverse contact model.

In the transient case with $\dot{\bar{\boldsymbol{U}}}_f \neq 0$ and $\dot{\boldsymbol{U}}_f^{-1} \neq 0$, statements about the system's stability are difficult, as uncertain plant and controller then describe time-varying MIMO systems. However, the transitions can be considered as disturbances on the system due to their short duration.

5.3.2 Contact Surface Estimation

For this section, the foot contact geometry is measured based on the state of the discrete contact switches located in each foot pad of Lola, see Section 2.1.1. Vast parts of this section have previously been published in [292].

Once a switch is closed, full contact of the corresponding pad with the ground is assumed. When no switch is closed, full contact at all pads of the foot is assumed (the contact model must be parametrized with some $A > 0$). The switches are far from being ideal contact indicators; they do not trigger when the foot is inclined relative to the contact surface, and there is a certain threshold of $\approx 30\,\mathrm{N}$ for activation. Therefore, the raw sensor signal per foot is enhanced with information from the FT sensors. Figure 5.6 shows the whole pipeline from the raw sensor signals to the final contact geometry matrix $\boldsymbol{U}_f^{-1}(t)$. It is separated into several modules, which are described next. Each foot has a separate pipeline with an independent estimation.

Contact State Enhancement This module contains several steps to improve the quality of the raw sensor contact state. For every foot, an enhanced contact state is calculated to get a more realistic representation of the actual contact geometry.

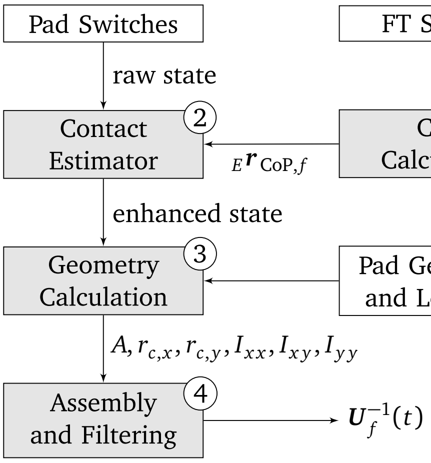
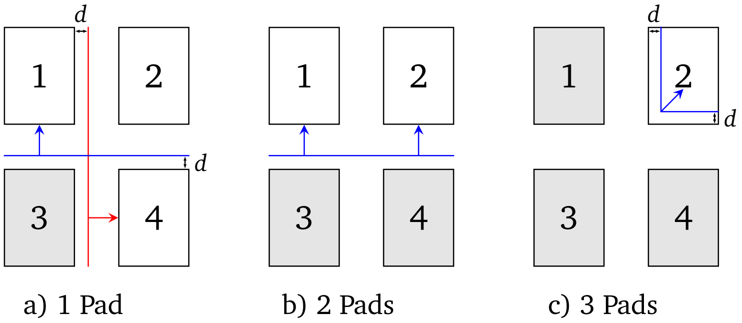

Pad Switches
FT Sensor
raw state
${}_E\boldsymbol{\lambda}_{f,m}$
Contact Estimator
2
CoP Calculator
1
${}_E\boldsymbol{r}_{\mathrm{CoP},f}$
enhanced state
Geometry Calculation
3
Pad Geometry and Location
$A, r_{c,x}, r_{c,y}, I_{xx}, I_{xy}, I_{yy}$
Assembly and Filtering
4
$\boldsymbol{U}_f^{-1}(t)$

Figure 5.6: The contact geometry estimation pipeline based on discrete contact switches [292].



a) 1 Pad b) 2 Pads c) 3 Pads

Figure 5.7: Top-down view of the foot with its four pads [292]. Grey pads have already been detected as closed contacts in these three exemplary states. The boundaries (red, blue) for the CoP-based enhancement of the contact state are defined by the pad geometry and a margin d. If the location of the CoP crosses a boundary in the direction of the corresponding arrow, the pads pointed to are treated as closed contact in the enhanced sensor state.

First, a contact closing delay is applied to the sensor data, i.e., a contact state must be active for a certain duration $\Delta t_{d,c} = 0.02\,\mathrm{s}$ until it is captured in the enhanced state. This applies only to the initial foot contact with the environment. This logic's main purpose is to prevent erroneously detected partial contact situations in the first milliseconds after contact due to delayed activation of the contact switches.

Second, contacts may only close based on sensor data but not open again. This assumption is necessary, as the contact sensors' discrete nature otherwise leads to instabilities caused by the permanent transition between closed and open contact states. Therefore, if a pad contact was closed in the past, it remains closed in the enhanced state. The enhanced state is reset to zero — i.e., no contact — in the corresponding foot's swing phase.

Finally, the calculated position of the CoP is fused with sensor data from the contact switches to detect contact-closing transitions faster and more reliably. Based on the full 6-axis data from the FT sensor, the location ${}_E\boldsymbol{r}_{\mathrm{CoP},f}$ of the CoP in the foot plane is calculated. Note that this is the CoP per foot, not the overall CoP of both feet. Sensor noise and acceleration of the foot mass can falsify the calculated ${}_E\boldsymbol{r}_{\mathrm{CoP},f}$. Therefore the module is only active when the foot load is higher than 30 % of the robot's weight.

The algorithm works with geometric margins for the current CoP and therefore requires at least one contact sensor of the foot to have detected a contact first. The general principle is simple and visualized in Figure 5.7: Once the CoP is outside the geometric region of closed-contact pads, some of the other pads must also be in contact. The algorithm handles all permutations of the three scenario classes shown there. For classes a) and b), only one dimension of ${}_E\boldsymbol{r}_{\text{CoP}}$ is evaluated for the decision, which is a simplification for class b) scenarios (two contacts are considered closed by looking only at a single CoP coordinate). Furthermore, type c) estimation is also simplified, as it does not consider the more complicated actual BoS of the three closed-contact pads. The simplifications make the algorithm robust against false data, as it tends to overestimate the contact area, which in turn leads to less aggressive contact force control. Currently, two diagonally opposite closed contacts are not considered, i.e., the state is not enhanced based on the CoP location in this case.

Geometry Calculation & Filtering The enhanced contact state from the *Contact Estimator* is used in conjunction with the known geometry and location of the footpads to calculate the contact model's geometry parameters. These consist of the position of the contact area centroid (${}_E r_{c,x}$, ${}_E r_{c,y}$), the contact area A, as well as the corresponding second moments of area I_{xx}, I_{yy}, I_{xy} relative to the location of the foot's TCP, see Section 5.2.1. When a pad contact is closed, the whole area of the pad is considered the contact area.

The contact geometry parameters are then used to actually compute the raw inverse contact model matrix $\boldsymbol{U}_{f,r}^{-1}(t)$ for each foot f. As the geometry information is discontinuous due to the discrete nature of the contact switches, the raw matrix is then filtered with a second-order low-pass filter:

$$\boldsymbol{U}_f^{-1}(t) = \text{lpf2}(T = 0.005\,\text{s}, d = 1.0, \boldsymbol{U}_{f,r}^{-1}(t)). \tag{5.51}$$

The filter is applied element-wise, i.e., every matrix element is filtered independently. This step represents the end of the contact estimation process, and the resulting contact model is used for the control approach described above.

5.3.3 Limitations & Implementation

It becomes clear from the contact estimation pipeline that $\boldsymbol{U}_f^{-1}(t)$ is bounded — the smallest assumed contact surface is one footpad. However, the delay in the contact switches may cause the assumed contact surface to be less than the actual contact surface and the condition $||\bar{\boldsymbol{U}}_f(t)\boldsymbol{U}_f^{-1}(t)|| \leq 1$ is violated. This can result in stability issues due to increased total open-loop gain when the gains are tuned for the nominal (full) contact surface.

This trade-off between high performance in the nominal and partial contact cases on the one side, and stability on the other side, is solved with different gains for partial and full contact. Equation (5.50) then changes to

$${}_E\delta\boldsymbol{v}_{f,u} = \left(\boldsymbol{U}_{\text{full}}^{-1} + G_p\left[\boldsymbol{U}_f^{-1}(t) - \boldsymbol{U}_{\text{full}}^{-1}\right]\right){}_E\delta\dot{\boldsymbol{\lambda}}_{f,u}, \tag{5.52}$$

with the static full contact model, the additional gain $0 < G_p < 1$ and ${}_E\delta\dot{\boldsymbol{\lambda}}_{f,u}$ from Equation (5.41). The approach scales the difference of the inverse contact model to the full contact with an additional gain to make the control loop robust to delayed contact surface information. The output of Equation (5.52) is inserted into Equation (5.47) to get the complete contact control law.

5.3.4 Experimental Validation

The method is evaluated in several scenarios where the contact surface is only a part of the full foot surface. This partial contact is not considered in the planned trajectories for the

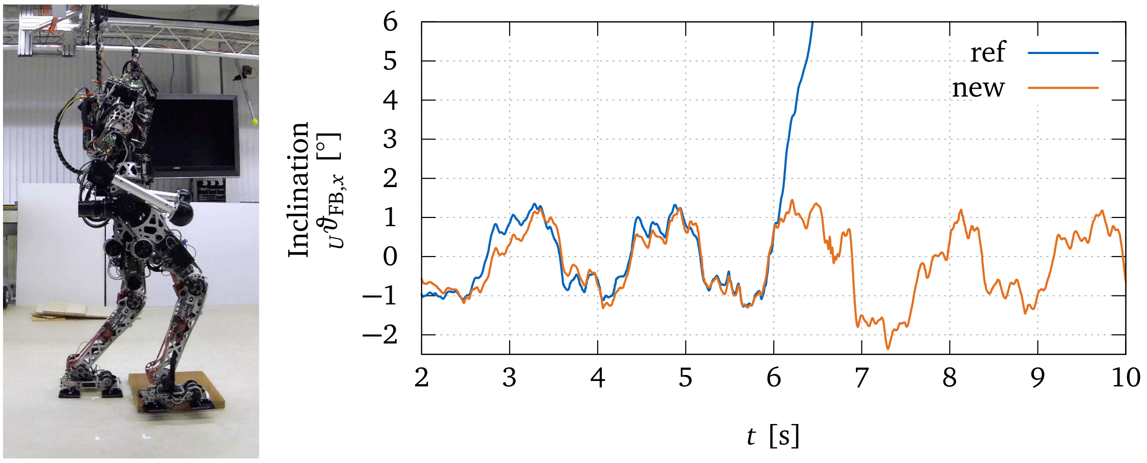


Figure 5.8: Experimental partial foothold scenario with an unexpected, 4-cm-high wooden board. The graph on the right shows the frontal (left/right) floating-base inclination for walking over this obstacle at a speed of 0.5 m/s for the reference force controller (ref) and the proposed controller with changing contact model (new). In the reference, the robot inclines further and falls.

robot, i.e., it is unexpected. In one case, LOLA walks in the direction of an unexpected 4 cm-high obstacle at a walking speed of 0.5 m/s. The obstacle's location is chosen to provide only a partial foothold at the inner side of the right foot, see Figure 5.8. This setup is especially tricky to overcome because it induces a disturbance directing away from the next stance. Even stabilization schemes via step modification may fail to compensate such a disturbance because this requires crossing the feet, which may be kinematically infeasible. The methods are evaluated with activated vertical CoM acceleration using the parameters[3] $G_{\mathrm{CoM}} = 0.4$, $D = 600\,\mathrm{Ns/m}$.

The dynamic contact model gain is set to a relatively low value $G_p = 0.05$. Although higher values do not immediately cause instability problems, the contact model's abrupt changes — caused by the contact switches' discrete nature — lead to high accelerations of the feet. Experiments with LOLA show this reduced gain is already very effective in partial foothold scenarios. With the reference controller (constant $\boldsymbol{U}_f^{-1}$), LOLA fails to overcome the partial foothold; with the proposed time-varying contact model $\boldsymbol{U}_f^{-1}(t)$, the disturbance is effectively mitigated, see Figure 5.8. Once contact with the ground is made, the *Contact Estimator* generates an estimate of the actual surface contact from raw contact switch data, see Figure 5.9. At the time $t = 5.48\,\mathrm{s}$, the partial foot contact (the left pads only) with the obstacle is established. The raw sensor values are particularly noisy, but the proposed enhanced contact state utilizing FT sensor data provides a robust estimation. At the time $t = 5.90\,\mathrm{s}$, the right side of the foot makes contact with the ground, closing full contact of the foot. It is noteworthy that the contact switches do not detect this contact surface change due to the foot's inclination relative to the ground. However, the enhanced state reflects this state change, as the location of the measured CoP leaves the permitted boundaries. The time-varying contact model leads to a higher effective feedback gain of the force controller during the partial contact. The resulting faster reaction reduces the time required to return to a full foot contact and results in lower torque tracking errors, see Figure 5.10.

Various experiments of the approach in heel-only and other side-only contacts indicate similar behavior for all partial footholds detectable by the contact switches. More experiments related to partial footholds are described in Section 7.1. A comparison with the

[3] The default parameters given in Sections 4.4 and 4.5 were optimized later and provide even better performance.

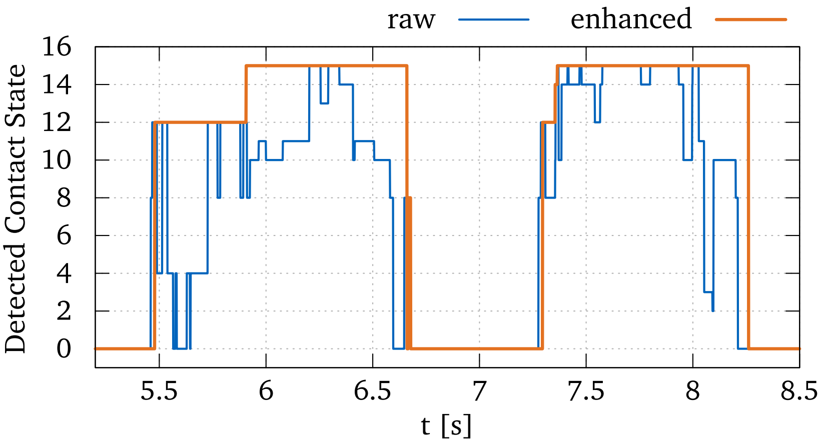


Figure 5.9: The raw and enhanced contact state for the right foot in the partial foothold scenario with time-varying contact model. The value is bit-encoded with 15 being all pad switches closed, 0 being no contact, and 12 indicating contact only on the left pads, see Section 2.1.1.

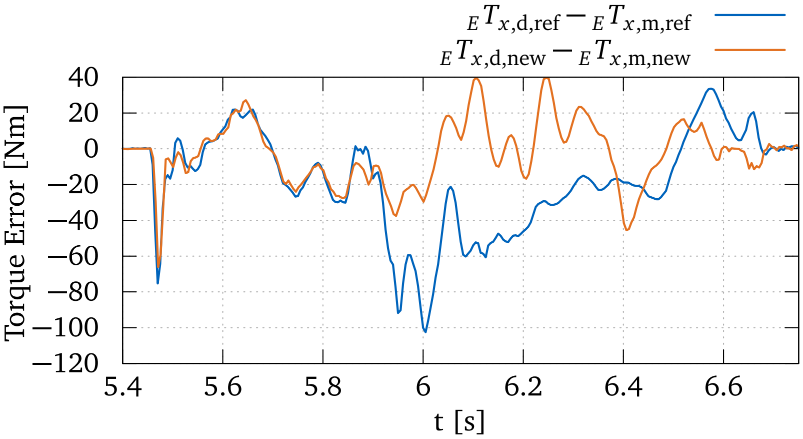


Figure 5.10: Torque tracking errors around the ${}_E x$-axis of the right foot for a partial foothold. The reference (ref) uses a static contact model, the new approach a time-varying contact model (new).

reference for walking on flat ground does not show significant differences. The IMU data indicates a small reduction in the peak-to-peak inclination angles. With time-varying contact model, the total CoM wrench tracking errors ${}_{W}e^{\mathrm{CoM}}_{T_x,\mathrm{RMS}}$, ${}_{W}e^{\mathrm{CoM}}_{T_y,\mathrm{RMS}}$ change by −2.3 % and +1.9 % respectively. The changes are not significant with improvements for the x-axis torques and almost equal deteriorations for the y-axis.

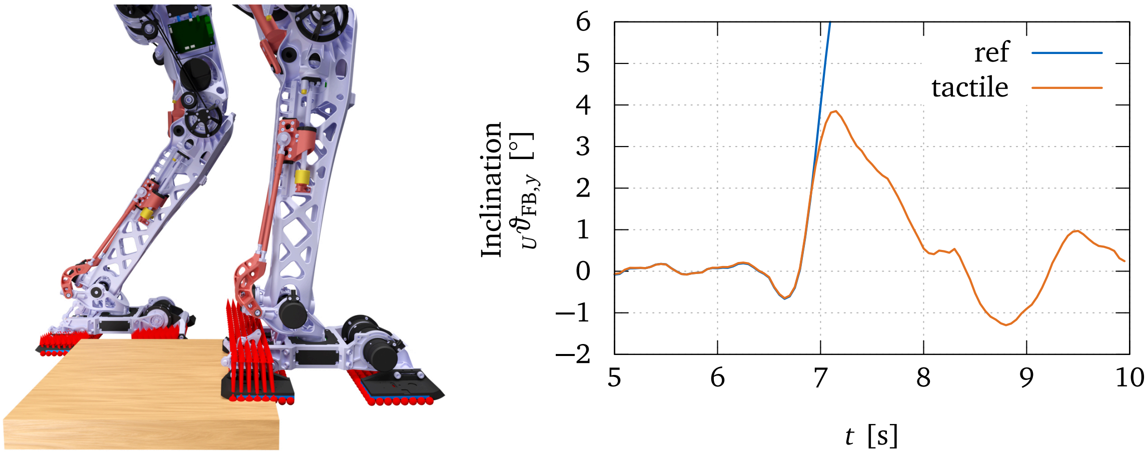
5.3.5 Discussion in the Context of Related Work

There has only been little research on the use of terrain information directly in the ground reaction force control to the author's knowledge. One interesting approach uses feet with four-point contacts and force sensors in every corner to get information on the contact geometry [222]. The work uses a set of foot-motion primitives — entitled environmental modes — to control the contact forces that affect the ZMP. Furthermore, a strategy for a two-point contact along the feet's diagonal axis is proposed to avoid instability if the desired ZMP is on that axis. Other work proposes two different force sensors at the center and front position of the foot. Based on the two force sensor readings, a heuristic detects if uneven terrain is present and lowers the stiffness of a virtual impedance of the ankle joint [331]. The parameters used for the virtual impedance are predefined and switch in a digital way when uneven terrain is detected. In contrast to the work presented in this thesis, both approaches have no direct coupling between the contact geometry information and a force control's contact model.

Another approach estimates the rough terrain profile based on direct kinematics and the location of the center of pressure at initial touchdown [330]. The disturbed robot is driven back to the reference trajectories using kinematic and IMU data. Also, in other work, the actual CoP is used to determine simple information on the contact state and heuristically adapt the motion of the robot [217]. However, it is not possible to directly measure the contact area from the instantaneous CoP information. Nevertheless, it is possible to use an active exploration mechanism to determine the current contact area from foot movements. This approach makes the robot ATLAS walk on partial footholds, and even line contacts [324]. Naturally, the exploration of the allowable region for the CoP takes time in contrast to an instantaneous contact surface measurement. Furthermore, the contribution focuses on the online generation of specific trajectories for such narrow support regions. In contrast, the work at hand focuses on realizing already planned contact wrenches when an unplanned partial foothold occurs.

There is related work on tactile foot soles for biped robots [134, 161, 162, 298], see also Section 2.4.4. However, contact surface information is, to the author's best knowledge, not used for contact force control. [252] describes the use of the measured foot contact area to determine if a contact is safe. In the case of partial contact, the footstep location is replanned. In contrast to the work at hand, the motions are executed quasi-statically, and no feedback loop is closed over the contact surface information.

The experimental results indicate that the surface-aware force controller's primary benefit is a significantly lower duration until full foot contact is established. Just as for all contact force control schemes presented in this chapter, the motion of the CoM is not changed, which requires the realization of the contact wrenches intended for full foot contact. Consequently, the approach fails when a full surface contact can not be achieved, e.g., in the case of a very high obstacle. The method can be combined with step-modification stabilization schemes to mitigate disturbances in these cases. For uneven terrain with heights in the robot's step height range, the proposed method provides a significant improvement for unexpected partial contacts. This is despite the coarse estimation of the contact surface via four discrete contact



(a) **(b)**

Figure 5.11: (a) Partial-contact simulation scenario with small contact surface and the virtual tactile foot. Every red sphere corresponds to a taxel. The arrows indicate the force on the taxels. (b) Sagittal floating-base inclination from a simulation experiment with static contact model (ref) and with contact-surface updating from tactile sensor data (tactile). In the reference, the robot falls in this scenario.

switches. Although the contact switch data is fused with FT sensor data, the contact surface estimation is delayed due to the contact switches' activation thresholds.

5.3.6 Contact Surface Estimation using a Tactile Foot Sole

A higher resolution and lower delay for the contact estimations can be achieved using a tactile foot sole, see Section 2.4. Compared to the discrete contact switches, the tactile sensor triggers already at low contact pressures, see Section 2.3.3, reducing the time required to detect the surface. A simulation model of the tactile foot sole covering nonlinearity, hysteresis, drift, and taxel-crosstalk is developed in [250]. Based on this model, simulative experiments with a virtual tactile foot, see Figure 5.11, are conducted for LOLA.

A binary contact state is extracted via thresholding the quotient of the current resistance to each taxel's unloaded resistance. This provides binary contact data for every cm^2 of the foot sole, not only for complete footpad areas. From the binary state matrix, the contact model parameters are calculated. Simulation results considering the full multi-body dynamics of LOLA support the sensor's suitability for contact surface detection. The robot overcomes the obstacle when the contact model is updated based on tactile sensor data; in the reference, the simulated LOLA falls, see Figure 5.11. However, a direct comparison with the method using the discrete sensors is not conducted due to a lacking realistic simulation model for the discrete pad switches. The author expects a performance gain for contact surfaces smaller than a single footpad. Nevertheless, the opening and closing of contacts occurs very fast, and the resulting high changes in the contact model put high torque requirements on the robot's hardware. A limitation of the contact model change rate seems necessary even for estimations from a tactile sensor sole.

A different approach to improving the detection of contact surfaces is based on additional distance sensors or cameras at the feet. Based on such sensor data, the ground topology can be detected before the contact is made, which enables in-air adaptation of the swing foot trajectory to avoid early- and late contacts. Initial experiments with foot distance sensors on LOLA in this direction are promising [258]. An additional advantage of distance sensors is that the contact surface may be estimated beforehand without the tactile sensor's delay.

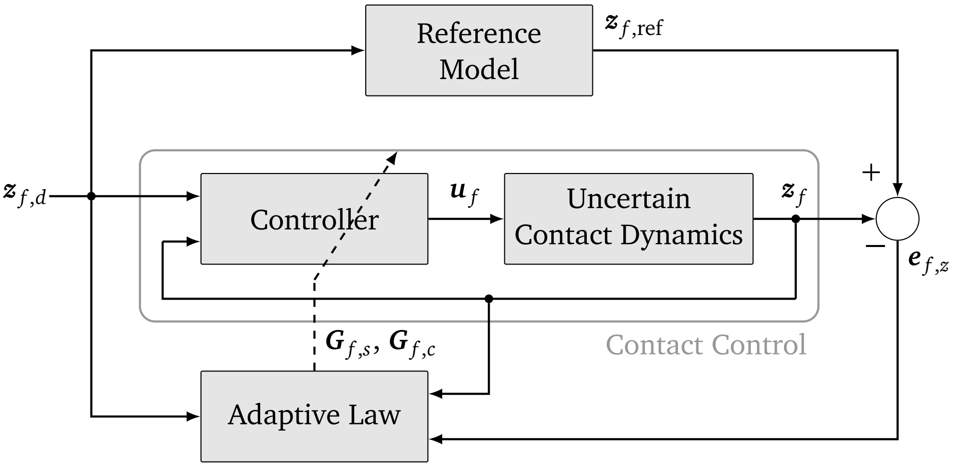


Figure 5.12: Exemplary MRAC control scheme with reference model for the closed-loop contact control of foot f. An error between the reference state $z_{f,\text{ref}}$ and the actual state z_f causes an adaptation law to modify the controller gains $G_{f,s}$, $G_{f,c}$.

5.4 Adaptive Contact Force Control

The control approach described in Section 5.3 uses the sensed contact surface to avoid degradation of the controller performance for partial contacts with the environment. However, the dynamics of the controller are not adapted to changing ground stiffness. Because the plant of a soft ground has a lower static gain (see Section 5.2.1), the nominal controller gains lead to degraded controller performance, i.e., lower crossover frequency of the closed loop. A straightforward countermeasure is the direct estimation of the mechanical contact properties with consideration in the inverse contact model Equation (5.15). However, this requires very accurate information on the positions and orientations of the feet in the inertial FoR. This is typically difficult due to elasticities in the mechanical structures and joints, and limited information on the floating-base DoFs [239].

Instead, a different approach is taken: changes of the ground stiffness are detected indirectly by comparing the contact wrench dynamics to a reference model of the nominal (stiff) closed-loop dynamics. A direct Model-Reference Adaptive Control (MRAC) scheme is deployed on each foot to adapt the control law gains when the actual contact wrench dynamics diverge from the reference model, see Figure 5.12. The design is based on the lag-compensated contact controller described in Equation (5.41) and combined with the sensor-based model updating described in Section 5.3. The approach mainly aims at improving contact control — and therefore robustness of the robot — on soft ground.

5.4.1 Reference Model

The reference model defines the nominal (expected) dynamics of the closed-loop contact force controller. It calculates a reference state $z_{f,\text{ref}}$ from the desired input state $z_{f,\text{d}}$. To find a suitable reference model, the state vector z_f must be defined first. The MRAC scheme is based on time-varying full state feedback of z_f, and the structure of this adaptive controller should match the well-tested approach used in the preceding sections, i.e., Equation (5.41).

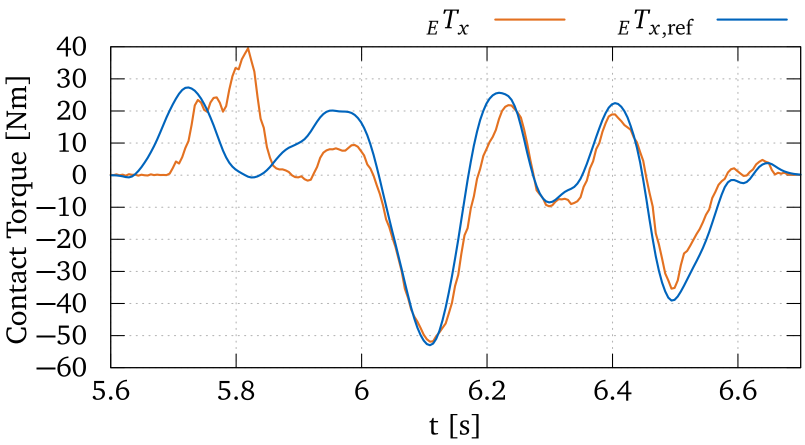


Figure 5.13: Actual contact torque around the x-axis and corresponding torque calculated from the reference model for one step on hard ground. Note that the models do only match some time after initial contact due to timing errors and a changing contact surface.

Thus, the following definitions for the actual and desired state are used:

$$\boldsymbol{z}_f = \begin{bmatrix} {}_E\boldsymbol{\lambda}_{f,u} \\ {}_E\dot{\boldsymbol{\lambda}}_{f,u} \end{bmatrix} = \begin{bmatrix} \mathrm{lpf1}(T_l,\, \boldsymbol{S}_f\, {}_E\boldsymbol{\lambda}_{f,\mathrm{m}}) \\ \mathrm{ddt}(T_l,\, \boldsymbol{S}_f\, {}_E\boldsymbol{\lambda}_{f,\mathrm{m}}) \end{bmatrix} \in \mathbb{R}^6 \qquad \boldsymbol{z}_{f,\mathrm{d}} = \begin{bmatrix} {}_E\boldsymbol{\lambda}_{f,u,\mathrm{d}} \\ {}_E\dot{\boldsymbol{\lambda}}_{f,u,\mathrm{d}} \end{bmatrix} = \begin{bmatrix} \mathrm{lpf1}(T_l,\, \boldsymbol{S}_f\, {}_E\boldsymbol{\lambda}_{f,\mathrm{d}}) \\ \mathrm{ddt}(T_l,\, \boldsymbol{S}_f\, {}_E\boldsymbol{\lambda}_{f,\mathrm{d}}) \end{bmatrix} \in \mathbb{R}^6, \tag{5.53}$$

with time-constant T_l from Section 5.2.7. Note that in theory, any choice of the state vector is valid. However, the full state vector must be observable from the only measurable variable ${}_E\boldsymbol{\lambda}_{f,\mathrm{m}}$. In this case, simple filters are used to generate the filtered contact wrench and its derivative.

The following decoupled second-order system is chosen as reference model for the nominal closed-loop dynamics of foot f:

$$\boldsymbol{\zeta}_n \odot \boldsymbol{\zeta}_n \odot {}_E\ddot{\boldsymbol{\lambda}}_{f,u,\mathrm{ref}} + 2\boldsymbol{d}_n \odot \boldsymbol{\zeta}_n \odot {}_E\dot{\boldsymbol{\lambda}}_{f,u,\mathrm{ref}} + {}_E\boldsymbol{\lambda}_{f,u,\mathrm{ref}} = {}_E\boldsymbol{\lambda}_{f,u,\mathrm{d}}, \tag{5.54}$$

where $\boldsymbol{\zeta}_n \in \mathbb{R}^3$ and $\boldsymbol{d}_n \in \mathbb{R}^3$ are vectors of time constants and damping factors for each direction of ${}_E\boldsymbol{\lambda}_{f,u,\mathrm{ref}} \in \mathbb{R}^3$ respectively. The parameters of the reference model are obtained from experiments with the nominal controller on stiff ground and with full foot surface contact. A state-space representation of the reference model is given by

$$\dot{\boldsymbol{z}}_{f,\mathrm{ref}} = \underbrace{\begin{bmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 \\ -\operatorname{diag}(\hat{\boldsymbol{\zeta}}_n \odot \hat{\boldsymbol{\zeta}}_n) & -2\operatorname{diag}(\boldsymbol{d}_n \odot \hat{\boldsymbol{\zeta}}_n) \end{bmatrix}}_{\boldsymbol{A}_{\mathrm{ref}} \in \mathbb{R}^{6\times6}} \boldsymbol{z}_{f,\mathrm{ref}} + \underbrace{\begin{bmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \operatorname{diag}(\hat{\boldsymbol{\zeta}}_n \odot \hat{\boldsymbol{\zeta}}_n) & \boldsymbol{0}_{3\times3} \end{bmatrix}}_{\boldsymbol{B}_{\mathrm{ref}} \in \mathbb{R}^{6\times6}} \boldsymbol{z}_{f,\mathrm{d}}, \tag{5.55}$$

with $\boldsymbol{z}^T_{f,\mathrm{ref}} = \left[{}_E\boldsymbol{\lambda}^T_{f,u,\mathrm{ref}} \quad {}_E\dot{\boldsymbol{\lambda}}^T_{f,u,\mathrm{ref}}\right]$, and $\hat{(\cdot)}$ the element-wise (Hadamard) inverse. The reference model is asymptotically stable. To parametrize the reference model, desired and actual contact wrench measurements are acquired for LOLA walking on flat ground. Time-constants and damping values are then chosen for a best fit between the reference model output and the actual contact wrench, resulting in the parametrization $\boldsymbol{\zeta}_n = \left[0.01327\,\mathrm{s}, 0.01327\,\mathrm{s}, 0.01769\,\mathrm{s}\right]^T$, $\boldsymbol{d}_n = \left[0.95, 0.9, 0.4\right]^T$. The actual torque around the x-axis of the right foot and the corresponding torque from the reference model are depicted in Figure 5.13 for normal walking on hard ground.

5.4.2 Uncertain Plant Description

The design of the adaptive control law requires a state space description of the uncertain actuator and contact dynamics plant for a foot. For a robot with position-controlled joints, the actual foot state $\bar{\boldsymbol{z}}_f^T = \begin{bmatrix} \boldsymbol{x}_f^T & \boldsymbol{v}_f^T \end{bmatrix} \in \mathbb{R}^{1\times 12}$, i.e., position and orientation of the foot and their derivatives, follows the desired trajectory $\bar{\boldsymbol{u}}_f = \boldsymbol{v}_{f,\mathrm{d}} \in \mathbb{R}^6$ with some dynamics:

$$\dot{\bar{\boldsymbol{z}}}_f = \bar{\boldsymbol{A}}\,\bar{\boldsymbol{z}}_f + \bar{\boldsymbol{B}}\,\bar{\boldsymbol{u}}_f. \tag{5.56}$$

From the actual contact model $\bar{\boldsymbol{U}}$ (see Section 5.2.4) follows the relation to the force-controlled contact wrench:

$$_E\boldsymbol{\lambda}_{f,u} = \begin{bmatrix} \bar{\boldsymbol{U}}\boldsymbol{S}_f & \boldsymbol{0}_{3\times 6} \end{bmatrix} \bar{\boldsymbol{z}}_f. \tag{5.57}$$

The dynamics matrix $\bar{\boldsymbol{A}} \in \mathbb{R}^{12\times 12}$ is considered unknown. A decoupled actuator dynamics[4] is assumed with input matrix structure:

$$\bar{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{0}_{6\times 6} \\ \boldsymbol{I}_6 \end{bmatrix}. \tag{5.58}$$

Note that the system can have a maximum relative degree of two, which reduces the approximation of the actuator dynamics to a single pole. This restriction comes from the definition of the system state. Higher-order models for the plant require a larger state vector with resulting high effort for the design of a state observer.

The state vector of the uncertain plant description must match the definition used for the reference model above in order to use it for an adaptive controller. In the following, the system is transformed to a suitable formulation. Note that the input and output of the state space model must not match the physical ports of the system. Using the transform

$$\boldsymbol{z}_f = \underbrace{\begin{bmatrix} \bar{\boldsymbol{U}}\boldsymbol{S}_f & \boldsymbol{0}_{3\times 6} \\ \boldsymbol{0}_{3\times 6} & \bar{\boldsymbol{U}}\boldsymbol{S}_f \end{bmatrix}}_{\boldsymbol{L}} \bar{\boldsymbol{z}}_f, \tag{5.59}$$

the plant can be written with the state $\boldsymbol{z}_f$. Furthermore, the control input of the plant is shifted to match the output of the robust force control Equation (5.41) with inverse contact model $\boldsymbol{U}^{-1}$:

$$\dot{\boldsymbol{z}}_f = \underbrace{\boldsymbol{L}\bar{\boldsymbol{A}}\boldsymbol{L}^{-1}}_{A\in\mathbb{R}^{6\times 6}} \boldsymbol{z}_f + \boldsymbol{L}\bar{\boldsymbol{B}}\boldsymbol{S}_f^T \boldsymbol{U}^{-1}\,{}_E\delta\dot{\boldsymbol{\lambda}}_{f,u}. \tag{5.60}$$

It is further known that the robust force-control feedback is linear in the gains G_λ, $G_{\dot{\lambda}}$. Consequently, the system's input is further shifted to

$$\dot{\boldsymbol{z}}_f = \boldsymbol{A}\,\boldsymbol{z}_f + \boldsymbol{B}\ \mathrm{diag}(\Xi)\,\boldsymbol{u}_f \tag{5.61}$$

$$_E\boldsymbol{\lambda}_{f,u} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_{3\times 3} \end{bmatrix} \boldsymbol{z}_f, \tag{5.62}$$

where the input matrix $\boldsymbol{B} \in \mathbb{R}^{6\times 6}$ now contains the actual contact $\bar{\boldsymbol{U}}$, the inverse contact model $\boldsymbol{U}^{-1}$, and the nominal gains G_λ, $G_{\dot{\lambda}}$:

$$\boldsymbol{B} = \boldsymbol{L}\bar{\boldsymbol{B}}\boldsymbol{S}_f^T\boldsymbol{U}^{-1} \begin{bmatrix} \boldsymbol{I}_3 G_\lambda & \boldsymbol{I}_3 G_{\dot{\lambda}} \end{bmatrix} \tag{5.63}$$

$$= \begin{pmatrix} \boldsymbol{0}_{3\times 6} \\ \bar{\boldsymbol{U}}\boldsymbol{U}^{-1} \begin{bmatrix} \boldsymbol{I}_3 G_\lambda & \boldsymbol{I}_3 G_{\dot{\lambda}} \end{bmatrix} \end{pmatrix}. \tag{5.64}$$

[4] Due to the high gear ratios of the robot, the mass matrix of the dynamic system is diagonally dominant. In combination with the decentralized joint controllers, the closed-loop actuator dynamics are largely decoupled.

This selection of the system input has two advantages: First, it allows the straightforward integration of sensor-based contact surface data, see below. Second, the nominal state feedback matrices become identity matrices.

Moreover, an unknown positive definite diagonal uncertainty $\mathrm{diag}(\boldsymbol{\Xi}) \in \mathbb{R}^{6\times 6}$ is introduced to the left of $\boldsymbol{u}_f^T = \begin{bmatrix} \boldsymbol{u}_{f,\lambda}^T & \dot{\boldsymbol{u}}_{f,\lambda}^T \end{bmatrix} \in \mathbb{R}^{1\times 6}$. It is directly related to an uncertainty in the diagonal entries of $\bar{\boldsymbol{U}}\boldsymbol{U}^{-1}$ — an unmatched ground stiffness, for example. The plant uncertainty may, however, also cover differences of the plant between single and double support.

5.4.3 Adaptive Control Law

In the following, an adaptive control law based on a direct MRAC scheme is derived for the uncertain contact dynamics. In contrast to indirect MRAC, where the uncertain plant is estimated first, directly the gains of the feedback controller are adapted. The control scheme provides asymptotic tracking of the reference model, i.e., the closed-loop dynamics are adapted via the gains to match the reference-model dynamics. Comprehensive literature on the subject can, for example, be found in [209]. Let the adaptive controller with time-varying gains be defined as

$$\boldsymbol{u}_f = \boldsymbol{G}_{f,c}(t)\,\boldsymbol{z}_f + \boldsymbol{G}_{f,s}(t)\,\boldsymbol{z}_{f,\mathrm{d}}. \tag{5.65}$$

Provided there exist gains $\boldsymbol{G}_{f,c*} \in \mathbb{R}^{6\times 6}$, $\boldsymbol{G}_{f,s*} \in \mathbb{R}^{6\times 6}$ for the unknown $\boldsymbol{A}$, and uncertainty $\mathrm{diag}(\boldsymbol{\Xi})$ such that

$$\boldsymbol{A} + \boldsymbol{B}\ \mathrm{diag}(\boldsymbol{\Xi})\,\boldsymbol{G}_{f,c*} = \boldsymbol{A}_{\mathrm{ref}} \tag{5.66}$$

$$\boldsymbol{B}\ \mathrm{diag}(\boldsymbol{\Xi})\,\boldsymbol{G}_{f,s*} = \boldsymbol{B}_{\mathrm{ref}}, \tag{5.67}$$

then an adaptive control law to reach this equality with the reference model is given by

$$\dot{\boldsymbol{G}}_{f,c}^T = \mathrm{diag}(\boldsymbol{\Gamma}_c)\,\boldsymbol{z}_f\,\boldsymbol{e}_{f,z}^T\,\boldsymbol{Y}\boldsymbol{B} \tag{5.68}$$

$$\dot{\boldsymbol{G}}_{f,s}^T = \mathrm{diag}(\boldsymbol{\Gamma}_s)\,\boldsymbol{z}_{f,\mathrm{d}}\,\boldsymbol{e}_{f,z}^T\,\boldsymbol{Y}\boldsymbol{B} \tag{5.69}$$

$$\boldsymbol{e}_{f,z} = \boldsymbol{z}_{f,\mathrm{ref}} - \boldsymbol{z}_f, \tag{5.70}$$

where $\boldsymbol{\Gamma}_c$, $\boldsymbol{\Gamma}_s$ parametrize the adaptation rates for each direction of $\boldsymbol{z}_f$, $\boldsymbol{z}_{f,\mathrm{d}}$, and $\boldsymbol{Y}$ solves the Lyapunov equation

$$\boldsymbol{Y}\boldsymbol{A}_{\mathrm{ref}} + \boldsymbol{A}_{\mathrm{ref}}^T\boldsymbol{Y} = -\boldsymbol{Q} \tag{5.71}$$

for the reference-model state matrix $\boldsymbol{A}_{\mathrm{ref}}$, and $\boldsymbol{Q}$ some positive definite matrix [209]. The adaptation law requires the knowledge of the input matrix structure $\boldsymbol{B}$, which, however, depends on the unknown actual contact model matrix $\bar{\boldsymbol{U}}$. Thus, a matching contact surface with $\bar{\boldsymbol{U}}\boldsymbol{U}^{-1} = p\boldsymbol{I}$ for some $p > 0$ is assumed and

$$\boldsymbol{B} \approx \begin{pmatrix} \boldsymbol{0}_{3\times 6} \\ \begin{bmatrix} \boldsymbol{I}_3 G_\lambda & \boldsymbol{I}_3 G_{\dot{\lambda}} \end{bmatrix} \end{pmatrix} \tag{5.72}$$

is used for the adaptation laws in Equations (5.68) and (5.69). The default (nominal) gains are defined by

$$\boldsymbol{G}_{f,c,t_0} = -\boldsymbol{I}_6 \tag{5.73}$$

$$\boldsymbol{G}_{f,s,t_0} = \boldsymbol{I}_6. \tag{5.74}$$

When a foot f is switched to position control ($\beta_f = 0$), its gains $\boldsymbol{G}_{f,c}$, $\boldsymbol{G}_{f,s}$ are reset to the default values. This is motivated by the assumption that the ground at each footstep location has different mechanical properties. In addition, the adaptation of the gains is halted, when the normal force on the foot is below a threshold $F_{a,th} = 40\,\mathrm{N}$. This avoids adaptation when a foot is in the air or is near to break contact with the environment, because the reference model does not consider opening contacts.

The resulting modification of the foot velocities with feedforward term from Equation (5.41) and constant inverse contact model $\boldsymbol{U}^{-1}$ results to

$$ {}_E\delta\boldsymbol{v}_{f,u} = \boldsymbol{U}^{-1}\left(G_d\,\mathrm{ddt}(T_l, \boldsymbol{S}_f\,{}_E\boldsymbol{\lambda}_{f,\mathrm{d}}) + \left[\boldsymbol{I}_3\,G_\lambda \quad \boldsymbol{I}_3\,G_{\dot{\lambda}}\right]\boldsymbol{u}_f\right), \tag{5.75} $$

which defines the contact force controller together with Equations (5.47) and (5.65).

Stability Discussion In the nominal case, the control law equals Equation (5.41) and the corresponding statements on the control loop stability apply, see Section 5.2.4. The tracking error between the reference state and the actual state $\boldsymbol{e}_{f,z}$ is asymptotically stable, if Equations (5.66) and (5.67) hold and the actual contact surface matches the expected with $\bar{\boldsymbol{U}}\boldsymbol{U}^{-1} = p\,\boldsymbol{I}$ for some $p > 0$. However, it is known that adaptive controllers may lead to instability in the case of unmodeled dynamics of the plant [209]. Because the contact dynamics plant is generally uncertain, additional measures for robust stability are taken and described in the following.

Robust Adaptive Control The total stiffness of the ground and foot sole of the robot can only decrease but never increase beyond the stiffness of the foot sole. In addition, there are limits to the ground softness encountered in real-world scenarios. Consequently, the adapted gains are limited element-wise

$$ -G_a \leq G_{f,c,ij} \leq -I_{ij} \tag{5.76} $$
$$ I_{ij} \leq G_{f,s,ij} \leq G_a, \tag{5.77} $$

with some scalar gain limit $G_a > 1$. For maximum robustness, the maximum gains are chosen to yield marginal stability reserves for the contact force controller on stiff ground. From Theorem 5.2.5 then follows stability for all ground stiffnesses and the allowed range of gains under the specified conditions.

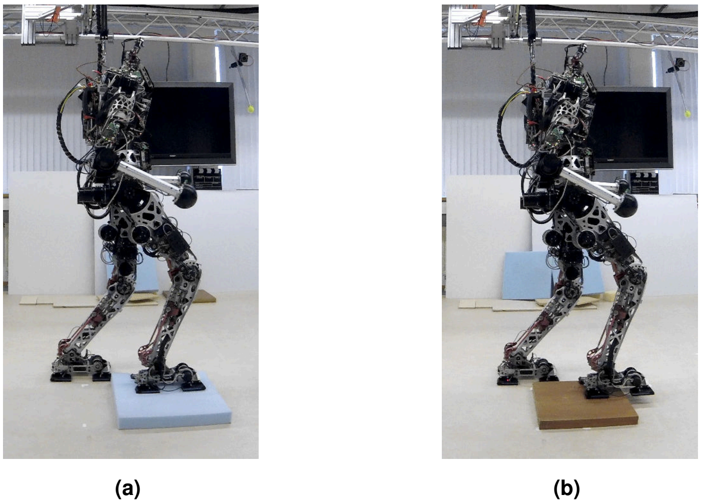
In addition, the σ modification, a well-known method to improve the robustness of adaptive controllers, is applied to the control scheme [130]. This basically introduces damping parametrized via $\sigma > 0$ to the adaptive laws, which change to

$$ \dot{\boldsymbol{G}}^T_{f,c} = \mathrm{diag}(\boldsymbol{\Gamma}_c)\,\boldsymbol{z}_f\,\boldsymbol{e}^T_{f,z}\,\boldsymbol{Y}\boldsymbol{B} - \sigma\,\boldsymbol{G}^T_{f,c} \tag{5.78} $$
$$ \dot{\boldsymbol{G}}^T_{f,s} = \mathrm{diag}(\boldsymbol{\Gamma}_s)\,\boldsymbol{z}_{f,\mathrm{d}}\,\boldsymbol{e}^T_{f,z}\,\boldsymbol{Y}\boldsymbol{B} - \sigma\,\boldsymbol{G}^T_{f,s}. \tag{5.79} $$

In combination with the gain limits, the approach adds a higher “cost” to high gains and slowly returns to the minimal (default gains) when the deviation from the reference model is small. However, the tracking of the reference model is no longer asymptotic. Due to the reset of the gains for every physical step, this plays a minor role compared to the higher robustness of the system.

5.4.4 Adaptive Control with Sensor-Based Contact Model Updating

The adaptive controller is able to compensate an uncertainty of the contact plant to some degree, whether caused by a different ground stiffness, contact surface, or robot pose. Nevertheless, the sensor-based information on the contact surface is available anyway and provides



(a) (b)

Figure 5.14: Evaluation experiments for the adaptive force controller. (a) with a soft foam board and (b) with a wooden board and a heel-only partial contact. The robot walks at the same speed of 0.5 m/s. The density of the foam is 40 kg/m³, the compression hardness is specified to 40 % compression at 6 kPa. The uncompressed height of the foam is 5 cm. In single support the foam is compressed to ≈ 1.5 cm height. Due to the softness of the material, the foot pads *and* the FT sensor body in the middle of the foot make contact, i.e., the contact surface is greater than just the four foot pads.

an instantaneous estimation. The combination of the approach described in Section 5.3 and the adaptive control law is straightforward. In fact, $\bar{\boldsymbol{U}}\boldsymbol{U}^{-1} = \boldsymbol{I}$ is assumed for the design of the adaptive laws and can only be achieved for a partial contact, if sensor-based contact data is used.

The approach described in the following is based on the estimation using the discrete contact switches of LOLA, see Section 5.3.2. Because of the coarse spatial resolution and delay, the feedback path with time-varying contact model is again limited with the additional gain G_p. Equation (5.75) with time-varying contact model $\boldsymbol{U}_f^{-1}(t)$ becomes

$$ {}_E\delta\boldsymbol{v}_{f,u} = \left(\boldsymbol{U}_{\text{full}}^{-1} + G_p\left[\boldsymbol{U}_f^{-1}(t) - \boldsymbol{U}_{\text{full}}^{-1}\right]\right)\left(G_d\,\text{ddt}(T_l, \boldsymbol{S}_f\,{}_E\boldsymbol{\lambda}_{f,\text{d}}) + \left[\boldsymbol{I}_3\,G_\lambda \quad \boldsymbol{I}_3\,G_{\dot{\lambda}}\right]\boldsymbol{u}_f\right) \tag{5.80} $$

and is plugged into Equation (5.47) as usual. This control scheme contains all other approaches described in this chapter, i.e., only the parametrization of this control structure must be changed to reach the exact behavior of all other described methods. The concept deals with the plant uncertainties using different strategies. On the one hand, plant uncertainties are tolerated with decreased quality of control, on the other hand some uncertainties are compensated by adaptation without or with lower performance loss.

5.4.5 Experimental Validation

The adaptive force control scheme is evaluated in different experiments containing soft ground and partial footholds, see Figure 5.14. The adaptive law is parametrized with $\boldsymbol{\Gamma}_c = 7 \times 10^{-4}\begin{bmatrix}1 & 1 & 0 & 0 & 0 & 0\end{bmatrix}^T$, $\boldsymbol{\Gamma}_s = 9 \times 10^{-4}\begin{bmatrix}1 & 1 & 0 & 0 & 0 & 0\end{bmatrix}^T$, $\sigma = 0.15$, and $G_a = 1.4$. These values are found from various experiments with LOLA to have minimal adaptation on flat terrain while still providing sufficiently fast adaptation for soft ground. The adaptation rates corresponding to the lead-compensator gains are set to zero to avoid stronger amplification of noise. Further, the vertical force direction gains are not adapted because vertical force control has only limited leverage in the single-support phase. A higher gain does not change the dynamics significantly, leading to a fast adaptation to the maximums. High gains in the vertical direction result in significant differences in the vertical foot trajectory, i.e, the CoM height in the single-support phase. In contrast to the method described in Section 4.5, these modifications are generated on velocity-level (not acceleration-level), i.e., without direct leverage on the vertical contact forces and are thus undesired.

Experiments on soft ground further revealed the force threshold for the early contact reflex (see Section 3.6.2) is too low for soft ground. The stance foot sinks into the soft material, which means the swing foot must also compress the ground to some extend. When the force threshold is too low, a false early contact is triggered on contact buildup with the soft material. The resulting stop of the foot motion may lead to the robot tipping over to the front. Thus, it is increased to $F_{ec,th,f} = 200\,\mathrm{N}$ for the experiments in the following. The setup is also tested on stiff ground, showing no significant differences to the original lower value. The default gait parameters from Table 3.3 are used.

Static Contact Model For the evaluation on soft terrain, see Figure 5.14a, the contact model $\boldsymbol{U}^{-1}$ is static and not updated (but the adaptation is active). The RMS values of the torque tracking errors ${}_{W}e^{\mathrm{CoM}}_{T_x,\mathrm{RMS}}$, ${}_{W}e^{\mathrm{CoM}}_{T_y,\mathrm{RMS}}$, and the floating-base inclination ${}_{U}\vartheta_{\mathrm{FB},x,\mathrm{RMS}}$, ${}_{U}\vartheta_{\mathrm{FB},y,\mathrm{RMS}}$ quantify the overall system performance, see Table 5.3. On stiff terrain, the results obtained from walking experiments are typically deterministic, i.e, repeatable. However, on the soft terrain, already slight differences in the initial position of the robot or the obstacle lead to measurable differences in the results. Thus, several runs are conducted for every method; mean and maximum values are given for each set of experiments.

The results are clearly in favor of the adaptive control law when compared to the base force controller ($\boldsymbol{\Gamma}_c = \boldsymbol{\Gamma}_s = \boldsymbol{0}$). Figure 5.15 shows the gains are primarily adapted when the foot is on soft ground — up to the maximum absolute value G_a. The adapted force-control gains and resulting higher crossover frequency can also be observed in the floating-base inclination, see Figure 5.16. With the adaptive control law, the peak-to-peak inclinations around the x-axis are reduced by ≈ 34 % relative to the reference; the peak-to-peak inclinations around the y-axis are reduced by ≈ 25 %.

Changing Contact Model The adaptive control law is further evaluated in the context of partial footholds, see Figure 5.14b. In this case, the actual foot surface differs from the expected one, causing different dynamics than specified by the reference model. Thus, the adaptive control law is expected to have advantages in these scenarios — even without sensor-based contact model updating.

The adaptive controller with static, and dynamic contact model is compared to the non-adaptive variants of the base controller with static (Section 5.2.3) and dynamic (Section 5.3) contact model. The RMS values of the torque tracking errors and the floating-base inclination are depicted in Table 5.4. Figure 5.17 shows the robot's floating-base inclination for the conducted experiments over time.

	${}_{W}e^{\mathrm{CoM}}_{T_x,\mathrm{RMS}}$ [Nm]		${}_{W}e^{\mathrm{CoM}}_{T_y,\mathrm{RMS}}$ [Nm]		${}_{U}\vartheta_{\mathrm{FB},x,\mathrm{RMS}}$ [10^{-3} rad]		${}_{U}\vartheta_{\mathrm{FB},y,\mathrm{RMS}}$ [10^{-3} rad]	
Method	mean	max	mean	max	mean	max	mean	max
reference	43.1	43.6	40.0	41.6	25.0	25.6	22.1	23.0
adaptive	39.5	40.2	38.1	41.0	21.0	21.9	18.9	21.0

Table 5.3: Comparison of tracking error results for adaptive and reference controller in the soft-board scenario. The mean and maximum values over a set of four experiments for each method are shown.

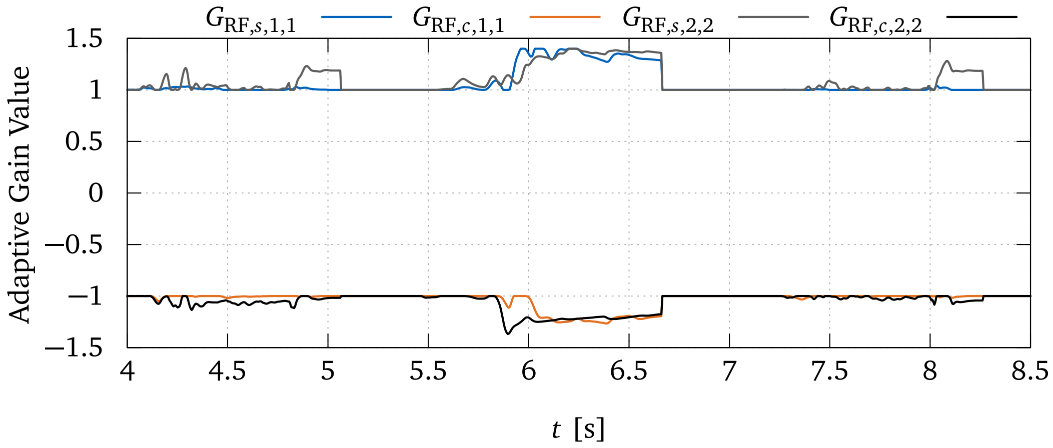


Figure 5.15: Adapted diagonal force-control gains when walking over a soft foam board with the right foot.

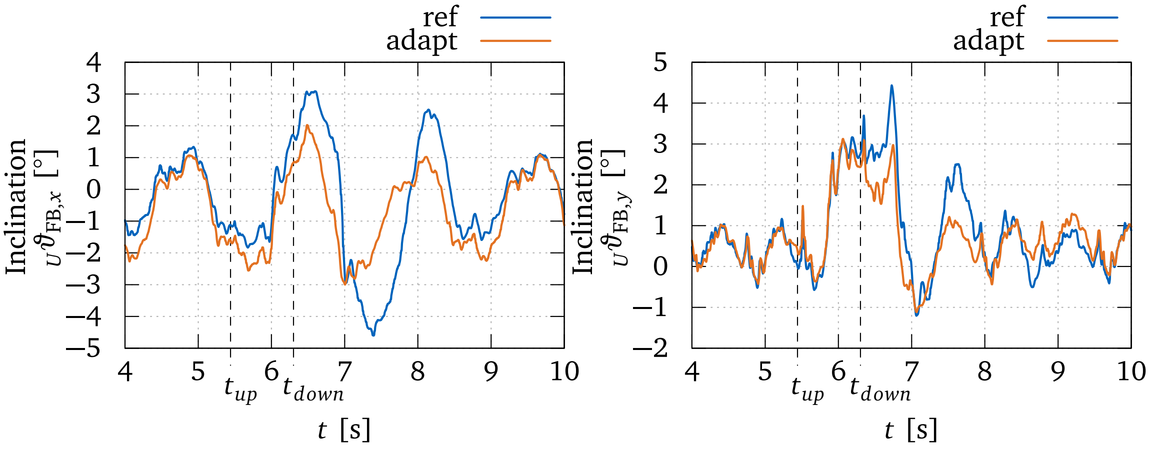


Figure 5.16: Floating-base inclination for walking over a soft foam board with the reference force controller and adaptive control law.

Method	${}_{W}e^{\text{CoM}}_{T_x,RMS}$ [Nm]	${}_{W}e^{\text{CoM}}_{T_y,\text{RMS}}$ [Nm]	${}_{U}\vartheta_{\text{FB},x,\text{RMS}}$ [10^{-3} rad]	${}_{U}\vartheta_{\text{FB},y,\text{RMS}}$ [10^{-3} rad]
reference	35.6	45.4	16.7	25.0
adapt	37.6	41.8	19.1	22.3
senmodel	**35.4**	**38.9**	**15.3**	**16.9**
senmodel+adapt	36.6	40.5	17.7	17.8

Table 5.4: Comparison of tracking error results in a heel-only partial contact scenario for the base controller without (*ref*) and with sensor-based contact model (*senmodel*), and the adaptive controller without (*adapt*) and with sensor-based contact model (*adapt+senmodel*). Bold values represent the optimum of a column.

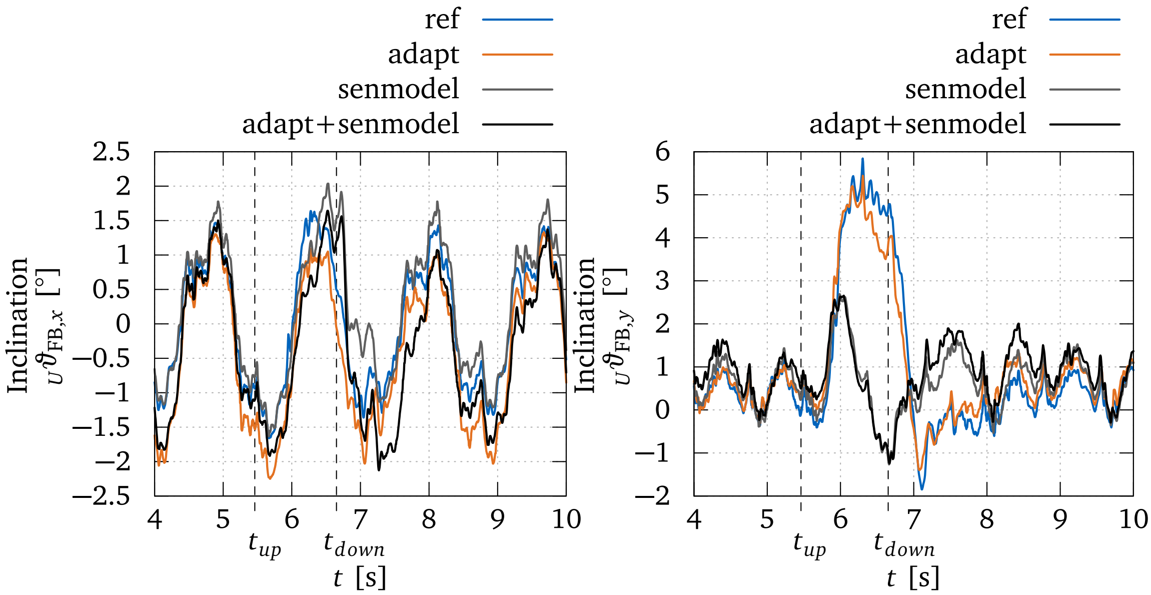


Figure 5.17: Floating-base inclination in a heel-only partial contact scenario for the base controller without (*ref*) and with sensor-based contact model (*senmodel*), and the adaptive controller without (*adapt*) and with sensor-based contact model (*adapt+senmodel*).

Compared to the reference with the static contact model, the adaptive control law shows a tracking error improvement for the y-axis but a degradation for the x-axis. However, looking at the floating-base inclination around the x-axis over time reveals an offset of the inclination with similar peak-to-peak values. Nevertheless, the reference controller with the dynamic contact model clearly outperforms the adaptive controller with the static contact model. Moreover, it is striking that the combination of the adaptive control law and the dynamic contact model approach is worse than the reference controller with the dynamic contact model. The time-data shows an offset for the x-axis inclinations, explaining the high RMS error values for this axis. However, the peak-to-peak inclinations in both directions are also slightly worse when combining the two methods.

5.4.6 Discussion in the Context of Related Work

One primary goal of the proposed adaptive control law is the traversal of unknown terrain with arbitrary mechanical properties at constant tracking quality for desired contact wrenches. The concept is presented for bipeds with position-controlled joints, where the necessary outer contact control loop is robust, but degrades in performance for softer ground. Due to the collocation of sensor and actor, joint torque control loops do not suffer from this problem. The higher-level balance and pose tracking controllers depend on the changing contact plant for robots with torque-controlled joints and the concept is assumed to pose advantages there, too.

There is only little research on biped walking over soft ground. In [104], dynamic walking over soft ground with a static-gain force controller is shown. It is unclear if the gains were tuned for rigid or soft ground. Quasi-static walking over soft terrain with torque-controlled joints is shown in [122, 195]. In [106], balancing on soft ground is shown for a robot standing on the spot. None of these approaches uses an adaptive control scheme for the contact controller.

In [90], an MPC scheme for a quadruped robot is extended with frequency-shaped cost functions to be able to tolerate soft ground, i.e., a violation of the rigid contact constraint. Further research on soft-terrain quadruped walking is presented in [75]. To the author's best knowledge, this is the only work actually closing a feedback loop over estimated contact parameters. The compliance of the ground is explicitly estimated via the state of the quadruped and the measured contact forces. This estimate is then fed into a contact-consistent whole-body controller to generate adapted trajectories. Further work describes the explicit detection of different terrain types via visual [84], haptic [20, 28, 121, 319], or multi-sensor [152] information. The resulting terrain estimate is in some cases used to adapt high-level gait parameters — e.g. step height, step width, timing — either based on heuristics [28, 84], or pre-computed cost maps [319]. In some cases, the different terrain classes needed to be known beforehand [20, 84, 121, 152, 319].

In contrast to related work, the approach proposed in this thesis uses a reference model instead of an explicit estimation of the mechanical contact properties. This has the advantage that it is not necessary to estimate the biped's floating-base state. Furthermore, the method does not change high-level gait parameters because experimental studies have shown that humans do not significantly modify their gait parameters when walking over soft ground [104]. The presented approach has two main advantages to the state of the art: First, the contact parameters' implicit estimation only requires the measurement of the contact forces and a reference model. Second, it can be combined with sensor-based contact surface data to use available parameters and estimate unknown ones.

On soft ground, the adaptive controller has a clear advantage compared to a static-gain approach. The results further indicate the adaptation of the gains is primarily triggered on soft ground, while a stiff ground only leads to a minimal deviation from the nominal values. While the original approach proposed in Section 5.2.3 is robust to the uncertainty of the mechanical ground parameters, the adaptive control approach adapts itself to the uncertainty with consequently higher performance of the closed-loop. The gains are only changed locally without threatening the stability margins of the other foot's control loop — which may stand on stiff ground. Moreover, the results show the proposed use of a reference model triggers adaptation for an unexpected ground stiffness *and* an unexpected contact surface.

However, the adaptive controller is worse than the reference controller when the contact-model is updated based on sensor data. The underlying cause can not be clearly concluded from the results. A plausible hypothesis is a mismatch between the reference model and the actual contact dynamics for partial contact surfaces. The adaptive control law does track the reference model dynamics by all means, even when the actual dynamics are advantageous. The reference controller with the sensor-based contact model may cause an error dynamics faster than specified by the reference model, especially with the coarse and delayed detection of the contact surface. This deviation also leads to an adaptation of the feedback gains to reach the slower reference dynamics. However, this hypothesis can not be clearly confirmed from the experiments. It is generally challenging to derive cause/effect relations due to the components' high coupling (controller, adaptive law, reference model, contact surface). Further tests with different reference models are required to prove the hypothesis.

Although the adaptive control approach leads to slightly lower performance on stiff ground — especially with partial contact surfaces — it improves the system's total robustness when arbitrary stiff and soft obstacles are encountered. The general idea combines the robustness to uncertainty (basic force control approach) with the adaptation to uncertainty, which to the author's best knowledge, has not been used for legged robot stabilization so far. Further experiments showing this control scheme's robustness are presented in Section 7.3.

The gains are currently reset after each step, which is a relatively conservative but robust approach. It might be possible to use a forgetting factor and prime the gains for the next

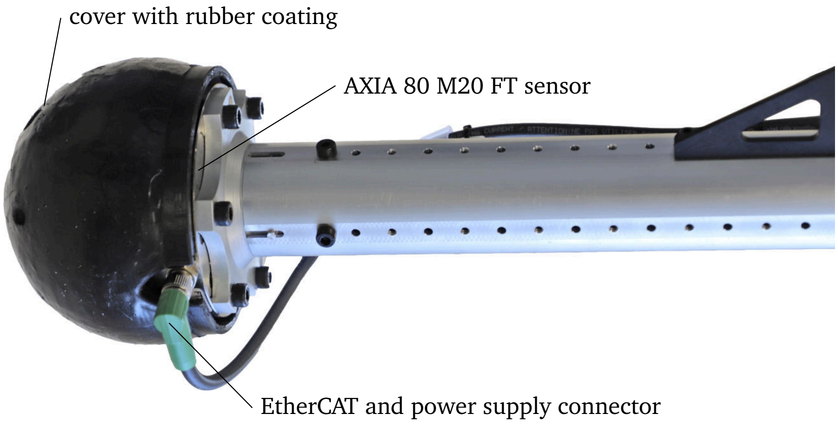

cover with rubber coating
AXIA 80 M20 FT sensor
EtherCAT and power supply connector

Figure 5.18: LOLA's hand with 6-axis FT sensor and spherical end effector. The cover is 3D-printed from PLA.

step with values between the default and the adapted gains of the current step. Moreover, the vertical force control gains are not adapted due to lacking correlation between the gains and the actual dynamics for the single-support phase. However, an adaptation of these gains purely in the double-support phase seems a reasonable extension of the algorithm.

5.5 Multi-Contact Force Control

In the context of this thesis, multi-contact control refers to using not only the feet, but also the hands of the robot for desired contacts with the environment. Additional contacts provide more potential to stabilize the robot's pose in case of a disturbance or kinematically/dynamically challenging movements. Each hand of LOLA is equipped with an additional FT sensor to enable control of the contact wrenches, see Figure 5.18. Just equally to the feet, the desired wrenches for the hand contacts are calculated in the wrench distribution QP, see Section 4.3. Note that the decision when to use a hand contact and where to place the end effector is determined by the WPG. For the experiments described in this thesis, the locations of the hands are predefined. Moreover, the hands use the same early-contact reflex and force-control activation logic than the feet, see Chapter 3.

5.5.1 Control Approach

In order to have sufficient kinematic reserves [267], only the positions of the hands are in the task space, while the orientation remains unspecified, see Section 3.4. The TCP of a hand is located at the geometric center of the spherical end effectors. Even for multiple contacts on one hand, the resulting forces can always be modeled as the result of a single point contact. Consequently, the contact mechanics is modeled with a 3D linear spring with constant stiffness k_h and the force control scheme is based on the approach described in Section 5.2.3:

$$
\begin{aligned}
{}_E\delta\dot{\boldsymbol{\lambda}}_{h,u} = {} & G_d\ \mathrm{ddt}(T_l,\, \boldsymbol{S}_h\, {}_E\boldsymbol{\lambda}_{h,\mathrm{d}}) + G_{\dot{\lambda},h}\,\mathrm{ddt}(T_l, \boldsymbol{S}_h\, {}_E\boldsymbol{\lambda}_{h,\mathrm{d}} - \boldsymbol{S}_h\, {}_E\boldsymbol{\lambda}_{h,\mathrm{m}}) \\
& + G_{\lambda,h}\,\mathrm{lpf1}(T_l, \boldsymbol{S}_h\, {}_E\boldsymbol{\lambda}_{h,\mathrm{d}} - \boldsymbol{S}_h\, {}_E\boldsymbol{\lambda}_{h,\mathrm{m}}),
\end{aligned}
\tag{5.81}
$$

with the measured ${}_E\boldsymbol{\lambda}_{h,\mathrm{m}}$ and desired ${}_E\boldsymbol{\lambda}_{h,\mathrm{d}}$ contact wrench for the hand. The additional gains $G_{\dot{\lambda},h}$, $G_{\lambda,h}$ are defined for the hand contact controllers. All other constants are shared with

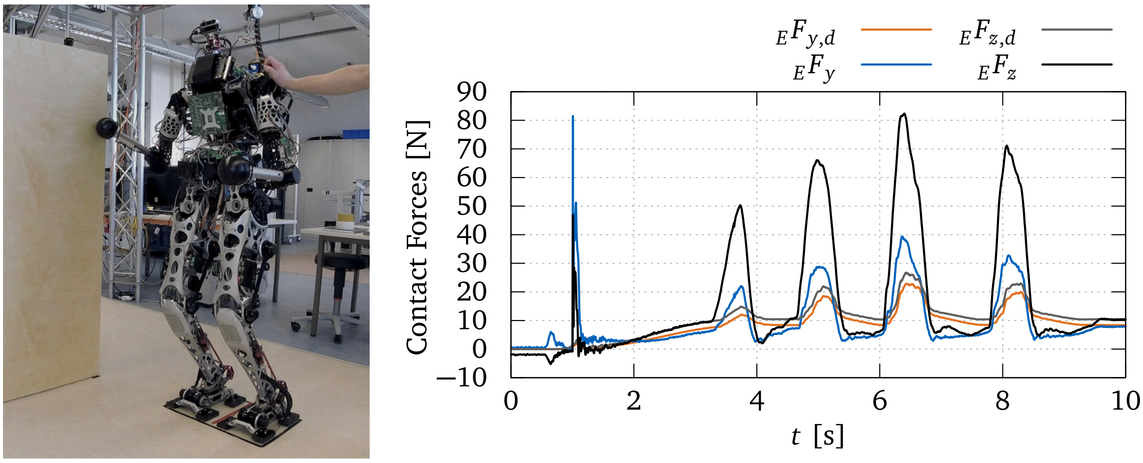


Figure 5.19: The multi-contact validation scenario (left), and the desired and actual contact forces in y and z direction. For clarity of the plot, the x-axis forces are not shown (these components point in direction of the arm with no significant forces).

the foot force control implementations. The velocity modification for the contact control task results to

$$ {}_E\delta \boldsymbol{v}_{h,u} = -\frac{1}{k_h}\, {}_E\delta \dot{\boldsymbol{\lambda}}_{h,u}. \tag{5.82}$$

Similarly to the feet, a hybrid control scheme (see Appendix A.5) is deployed to bring the modifications back to zero when the contact opens. This results in the following total velocity modification for a hand h:

$$ {}_E\Delta \boldsymbol{v}_{h,\lambda x}[n] = \mathrm{hybridCtl}\big(\beta_h,\, {}_E\delta \boldsymbol{v}_{h,u},\, {}_E\Delta \boldsymbol{x}_{h,\lambda x}[n-1]\big), \tag{5.83}$$

with the control mode blending factor β_h and the integrated modifications from the last time step ${}_E\Delta \boldsymbol{x}_{h,\lambda x}[n-1]$. Integration of the combined velocity is done via

$$ {}_E\Delta \boldsymbol{x}_{h,\lambda x} = \mathrm{intA}({}_E\Delta \boldsymbol{v}_{h,\lambda x}, {}_W\boldsymbol{A}_E). \tag{5.84}$$

Finally, the modifications on both hands are applied to the task-space trajectories calculated from the foot force controllers (see Section 5.2.6) to retrieve the final task-space trajectories for the IK:

$$ (\boldsymbol{X}_\mathrm{d}, \boldsymbol{V}_\mathrm{d}) = {}_E\,\mathrm{modifyAll}_h\big(\boldsymbol{X}'_\mathrm{d}, \boldsymbol{V}'_\mathrm{d}, {}_E\Delta \boldsymbol{x}_{h,\lambda x}, {}_E\Delta \boldsymbol{v}_{h,\lambda x}\big). \tag{5.85}$$

5.5.2 Experimental Validation

Multi-contact force control is validated for an additional contact of the right hand while the robot stands in place, see Figure 5.19. The controller gains are tuned with the same methodology used for the feet, resulting in $G_{\lambda,h} = 5\,\mathrm{s}^{-1}$, $G_{\dot{\lambda},h} = 0.02$. The location and timing of the hand contact is predefined in the WPG. Once the hand is in contact, the robot is disturbed by pushing on the left shoulder in the wall's direction. The desired and actual contact forces on the right hand are depicted in Figure 5.19. At time $t \approx 1\,\mathrm{s}$ the hand impacts with the wooden wall. After contact is made, the robot is pushed four times. The data shows the desired values are reached in the steady-state case.

The closed loop's crossover frequency is too low for the dynamics of the external pushes, resulting in significant errors in the force. Still, the actual contact forces are higher than the desired contact forces, i.e., an increase of the force-control gain would lead to a reduced counterforce to the disturbance. Because of the finite gain of the inclination controller (see Section 4.2), the robot basically behaves like a spring-damper system and allows a certain inclination. Note that this experiment also shows the consistency of the wrench distribution QP (see Section 4.3) for hand contacts. Further experiments utilizing both hands are described in Section 7.4.

5.5.3 Discussion in the Context of Related Work

Multi-contact locomotion is currently an active field of research in the robotics community. Yet, the term "multi-contact" describes quite different matters, ranging from multiple contacts on one foot to full-body contacts. In this thesis's context, multi-contact control defines the use of the feet and hands of LOLA to stabilize the robot in challenging environments. In related work, the multi-contact problem, i.e., the calculation of feasible contact points and a contact-consistent motion for the CoM and the end effectors, is often solved by pure planning approaches. On robots with position-controlled joints, this results in no force feedback at all by using models for the dynamics, and the contacts [45, 167]. However, these approaches are naturally sensitive to errors in the utilized models and often require planning times between minutes and hours.

To overcome these problems, outer force-control loops are added to the control schemes. In [116], damping control with additional decay of the modification over time is added to control the hand contact forces. The controller presented above uses the same general principle of force-feedback on velocity-level. Differences are in the additional switching between position- and force-controlled hands, making a general decay term unnecessary. The approach is commonly used and also applied in more recent work [254]. For robots with position-controlled joints, this control scheme is generally considered state of the art. The approach in this thesis is combined with the early-contact reflex to handle impacts (see Section 3.7.1), and the extension to an adaptive controller for unexpected mechanical properties is straightforward, see Section 5.4.

The general approach is also used on robots with torque-controlled joints when closing the torque-control loop on joint-level is undesired due to high sensor noise [52]. In this case, the data from additional FT sensors is fed back on velocity-level, and the ultimate desired configuration is realized via inverse dynamics. The approach is validated using a simulation only and can easily be applied to robots with position-controlled joints, practically not using the advantages of torque-control on joint-level. In contrast, the control framework described in [106, 196, 321] for multi-contact of a torque-controlled robot facilitates high-bandwidth torque control at joint-level and shows impressive compliance of the robot in these scenarios.

The experiment also validates the wrench distribution QP for use in multi-contact scenarios, showing the concept's general applicability. The desired contact forces assigned to the hand are relatively low even for large external disturbances of the robot. Thus, putting lower costs on the hand contact forces in the QP might improve the overall system performance in multi-contact scenarios even further.

The experimental data indicate that a low bandwidth of the outer contact-force control loops is not directly a disadvantage for robots with position-controlled joints. In this case, the robot's arm behaves stiffer than specified by the balance controller and the wrench distribution QP, ultimately leading to a more aggressive rejection of an external disturbance.

The hand's orientation is currently not considered in the force control approach, and the TCP of the hands is located in the geometric center of the spherical end effectors. This means that a hand's angular velocity—the orientations are not in the task space—may cause a non-zero contact point velocity that acts as disturbance on the force controller. As a future extension, it might be possible to estimate the contact point location from the measured contact torques and the geometrical parameters of the sphere to compensate for these effects.

The hand controllers show the same robust stability to the environment's unknown properties as the ground reaction force controllers. Also, impacts on the environment are handled by the same early contact algorithm. Although the contact material of the hands is relatively stiff, impact forces can be reduced effectively. Softer hand materials could further improve the robustness to impacts. The same principles as for the feet apply to the design of this compliant material, see Section 5.2.8.

5.6 Chapter Summary

This chapter proposes methods for the contact wrench control at the end effectors of the biped. For the ground reaction forces, an explicit contact model formulation is introduced, enabling the description of contact surface and geometry. The robustness of the control schemes to the mechanical ground parameters' uncertainty is shown using experimental and theoretical considerations. Furthermore, an approach for partial footholds is proposed, which uses contact sensors at the feet to parametrize the explicit contact model online. Moreover, the chapter presents an adaptive control scheme to deal with varying mechanical ground stiffness, i.e., adapting to the uncertainty instead of just tolerating it. The presented approaches are key enablers for walking over uneven and unstructured terrain with arbitrary contact surface and stiffness.

Based on the knowledge gained from the foot force controller design, control schemes for the robot's hands are designed to extend the stabilization to multiple hand and foot contacts. All methods are validated experimentally on LOLA.

Chapter 6

Software Architecture and System Testing

This chapter describes the control software architecture and ecosystem utilized for the development and testing of algorithms for LOLA. For software development, special emphasis is put on hard real-time constraints and full knowledge of the software's behavior. Thus, all methods running online and onboard of the robot are implemented completely in C++. Robot software frameworks like ROS [244] or YARP [197] are not used due to the missing support for hard real-time constraints, see also Section 2.2.5. On the one hand, this design choice increases the implementation effort for new methods; on the other hand, it enables very efficient implementations with low control latency and accurate runtime measurements.

Besides the custom control software, a specialized software ecosystem is used to develop software for LOLA. This includes a full multi-body dynamics simulation with a tailored contact model, data logging and visualization tools, and testing tools for the system to detect gaps between simulation and experiments at an early stage. The advantage of this ecosystem lies in its explicit specialization to LOLA. Especially the simulation environment lacks generalizability to other robots. However, this also makes the simulation particularly efficient and accurate because no compromises need to be made. Most of the ecosystem tools are not constrained by real-time requirements and are thus implemented in Python to reduce the development effort.

Because most of the code for LOLA is custom and the codebase is large, reusing software parts is especially important. To support code reuse across multiple robotic projects at TUM and worldwide, a dedicated C++ software library (*broccoli*) has been extracted from the refactored code of the LOLA project. Both *broccoli* and the LOLA codebase use the linear algebra C++ template library *Eigen* [95].

Safety is an important design goal in the context of powerful and expensive machines like LOLA. However, too strict safety considerations, e.g., strict velocity and acceleration limits on the joints, can greatly influence the control methods' performance. Because LOLA is a research platform, the system's safety is primarily achieved by extensive testing (in simulation) and not by limitations on the performance.

The three control system modules WPG, SIK, HWL, and the software utilities of the development ecosystem are visualized in Figure 6.1. In the following, previous work on LOLA's software architecture is described briefly. Furthermore, the current architecture of those software modules primarily designed by this author (SIK and HWL) is presented. Please note that many other improvements, tools, and particularly the WPG software module have been developed by Philipp Seiwald and are — although not documented here — equally important to the biped system's operation. Finally, a learning-based system testing approach is proposed and evaluated in the context of the software development for LOLA.

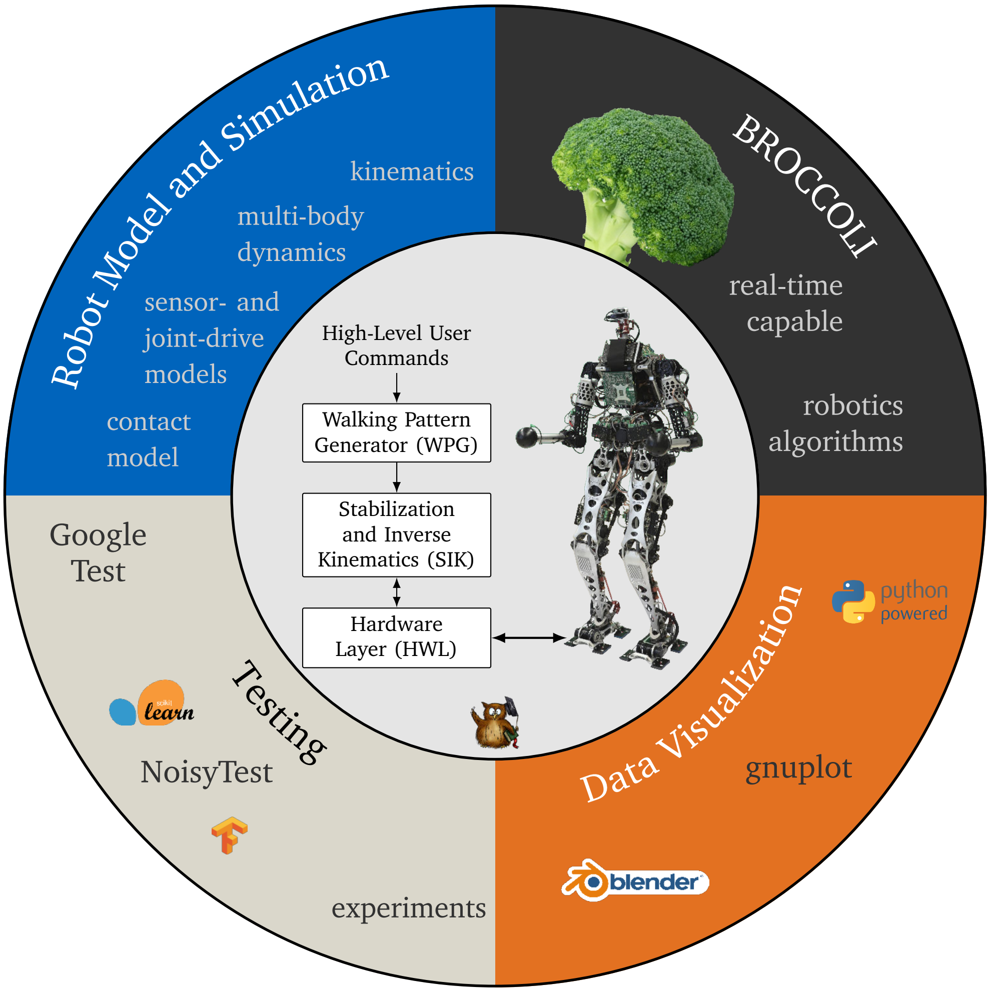


Figure 6.1: The LOLA software ecosystem. At the core of the LOLA codebase are the three main control modules WPG, SIK, and HWL, which run in independent processes on the real-time operating system *QNX* Neutrino. The separation of the code in different programs reduces the risk of a fault: as long as the HWL application runs, the system can transition to a safe state should WPG or SIK crash. The main control modules utilize a robot model, which is simultaneously used to simulate the controlled system on a Linux- or macOS-based development host. Simulation results or experimental data may be visualized and analyzed using a custom *blender*(https://blender.org) plugin and the gnuplot(http://gnuplot.info) graphing utility. Testing is done with unit and system tests via *Google Test*(https://github.com/google/googletest), a proposed machine-learning system test tool *Noisytest*, and real-world experiments. The open-source library *broccoli* [265] originates from the LOLA codebase and provides general robotics algorithms.

6.1 Previous Work on LOLA

The first software system for the control of LOLA was developed by Buschmann [33] in the years 2004 to 2010. Later, new functionalities have been added by Schwienbacher [263], Favot [80], Wittmann [325], Hildebrandt [111], and Wahrmann Lockhart [318]. With the beginning of the project dedicated to multi-contact locomotion for LOLA, it became clear that the existing software architecture had grown too quickly over time and that necessary changes would have been difficult to implement. Therefore, large parts of the software were redesigned to provide the necessary flexibility for the coming years. The former software architecture and the gained knowledge on good and less good parts of the software served as a basis for this new development. Parts of the old structure still exist and were only adapted to work with the new software system. This includes the original publish–subscribe system for parameters of the methods [33], and the complete robot model considering kinematics, electromechanical dynamics, and contacts [33, 34, 263]. The legacy code uses the custom linear algebra library *matvec* [278], which does not support vectorization features of modern CPUs. The library is successively replaced with *Eigen*. In the following, the architecture of the refactored software is presented. Furthermore, improvements to the remaining legacy code are described briefly.

6.2 The Broccoli Library

The **B**eautiful **Ro**bot **C++** **Co**de **Li**brary (*broccoli*) is an open-source header-only library for common robotics algorithms [265]. This library's development is a joint project and was initiated with the refactoring of several software modules for the LOLA control software. The extraction of refactored LOLA code to this project has several advantages. First, each software component in *broccoli* must have a stable and general public interface, which favors the development of reusable code[1]. Second, the old code was not covered by unit tests, and it was found easier to implement tests for new code in *broccoli* than to add them to legacy code in the LOLA project. Third, moving common utility components from the LOLA codebase to *broccoli* reduces the overall maintenance effort. *broccoli* can be tested and maintained separately from the LOLA codebase (which is then also smaller) and tends to be used by a larger userbase than the LOLA code. The library is already used for another robotics project at TUM, reducing the overall implementation effort.

Furthermore, this author believes that the separation of code responsibilities just by extraction of LOLA code into *broccoli* alone has a positive effect on long-term development effort and code quality. Lastly, one has to add that *broccoli* is certainly not perfect regarding state-of-the-art code quality measures. In the end, the time in research projects is always limited, and unfortunately, the quality of code in a robotics research project is only of minor interest in the research community. The created modules and major contributions to individual software components by this author are described briefly for the *broccoli* version 2.9.0 "belstar" in the following. For an in-detail documentation of features, refer to the *broccoli* API (https://am.pages.gitlab.lrz.de/broccoli/).

control The *control* module contains algorithms related to control theory and signal processing. It defines templates to represent discrete-time signals of different data types, consisting of an actual value and a corresponding sample time. Standard unary and binary mathematical operators and further signal processing steps are defined for the *broccoli* signals, which allows using these objects just as normal variables. However, the implementation of signals uses expression templates — a C++ metaprogramming technique — to define complex signal processing chains. The technique encodes an expression in the data type of an object at compile time. This enables "lazy evaluation" of expressions, which avoids temporary variables and makes some operations more efficient.

As an example, consider the product of two scalar signals. This operation's return value is not another signal containing the result but an object storing the operator (product) and the original operands. This signal expression is only evaluated when required by a signal sink or enforced by the user. The approach allows a user to define arbitrarily complex expressions in a human-readable way without drawbacks on efficiency. At compile time, the expression is basically condensed to a single statement. Expression templates are commonly used in

[1] In research projects, time is often a limiting factor. The development of reusable code requires higher effort, which often results in specialized pseudo modular structures. By requiring a well-defined public interface, at least some minimum reusability level is enforced.

modern linear algebra libraries, such as *Eigen* [95]. *broccoli* signal expressions are compatible with *Eigen*. The following submodules exist:

- **operators:** Contains operators for signal expressions, i.e, standard unary and binary operators, integrators, saturators, linear faders, and more.
- **sources:** Contains signal sources such as deterministic gaussian noise and the multisine implementation used for the onboard plant estimation, see Section 3.8.2.
- **feedback:** Contains essential feedback control implementations.
- **filters:** Contains several digital filter implementations (Finite Impulse Response and Infinite Impulse Response types), including those described for the use in the proposed control schemes, see Appendix A.3.
- **lti:** Contains discrete-time state-space representations for linear time-invariant systems. Additionally, converters from time-continuous to time-discrete representations are implemented in this submodule.
- **kinematics:** Contains efficient inverse kinematics implementations and related helpers.

core The core module contains miscellaneous utility components for mathematical operations, high-precision timing, and more. This author contributed the following two components and several minor extensions to this module:

- **platform:** Contains a wrapper for platform-specific functions used in other parts of *broccoli*, which are related to threading and accurate timing.
- **float:** Contains functions to compare floating-point numbers for approximative equality or inequality based on their units in the last place.

geometry/rotations This component implements functionality related to rotations and in particular the operators defined in Appendix A.2. It also contains additional software components from other authors of *broccoli*.

hwl The hardware layer module provides a framework for real-time communication between a control computer and distributed sensors or actuators of a robot. It implements the real-time middleware described in Section 2.2 with interfaces and wrappers for the Bus Variable concept. It further includes device abstractions for standard devices in robotics. The module is still under development; it strives to provide a complete set of building blocks for the easy integration of robot devices via different Fieldbus technologies with emphasis on real-time requirements and flexibility.

ode/integration This module provides a set of interface declarations to integrate system dynamics containing both time-continuous and time-discrete parts. Simulation models for robots must contain both the time-continuous electro-mechanical dynamics and the time-discrete data processing occurring in sensors, control loops, and actuators at different sample times. With the defined abstract interfaces, a simulated robotic system's components can be structured as a tree. Each leaf node represents a single, solvable entity with time-continuous and/or time-discrete part; each branch node is a set of other nodes that are solved in a predefined order. The tree structure makes it possible to define the order of time-discrete processes and group entities to a bigger system using a branch node. An exemplary tree

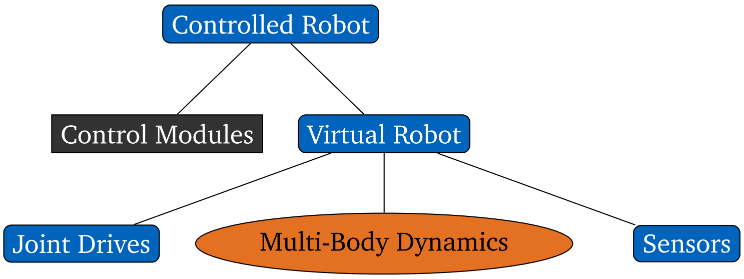


Figure 6.2: Exemplary tree structure for a solvable system (ode/integration) with time-continuous (ellipsoid), time-discrete (rectangular), and hybrid (rounded-edge rectangular) solvable entities.

structure for the simulation of a robot is depicted in Figure 6.2.

Although the current implementation only supports sequential (single-threaded) integration of entities, the concept can easily be extended to support parallel simulation of a node's child entities by replacing the implementation for this branch node. This may, for example, be used to simulate the dynamics of the joint drive current loops in parallel. Because the mechanical dynamics are significantly slower, an independent (parallel) simulation for each joint may be feasible without a big accuracy deficit.

The integration solver automatically finds the largest block in continuous time between two discrete events from the tree data. This makes the actual integration loop smaller (it does not have to care about discrete events), opening up potential for automatic and manual code optimizations, for example, loop unrolling, vectorization or optimizations targeting branch prediction. A basic forward integration scheme is included with the interface declarations.

6.3 Stabilization and Inverse Kinematics Application

A brief overview of the SIK software architecture is visualized in Figure 6.3. SIK contains a unidirectional real-time shared memory interface for the planner data and a bidirectional shared memory interface for the exchange of sensor and joint target data with the HWL. Also, the parameters for the methods, for example, gains, are received from the operator computer via a legacy (not real-time capable) publish–subscribe system during initialization.

The SIK module contains a finite state machine primarily for the abstraction of a fault state and the robot's initialization. When the operator gives the command to switch the robot's state from idle to operational, an integrated pose planner creates simple trajectories from the joints' current positions to the desired pose calculated via IK (the standing pose).

The control approaches are part of the SIK Module component. While the SIK Application is executed only on the robot's computer, the SIK Module is also instantiated in the simulation environment to control a virtual LOLA. The different subcomponents contain the methods described in various sections of this thesis, see Figure 6.3 and are executed in the depicted order from top to bottom. For the main components, the relative runtime is given in percent.

It is particularly noticeable that the computation of DK and IK together accounts for 68 % of the total SIK Module runtime. The direct kinematics module still contains a large amount of legacy code tied to LOLA's kinematic model, which is not optimized for modern CPU vectorization commands. Regarding the IK algorithm, 31 % of the runtime is used for the nullspace gradient calculation, i.e., mainly the collision checking. Furthermore, four different IK solutions need to be solved due to the bilinear interpolation between different task-space configurations, see Section 3.9. Currently, the four IK solvers run sequentially; a parallelization of these operations is straightforward, though.

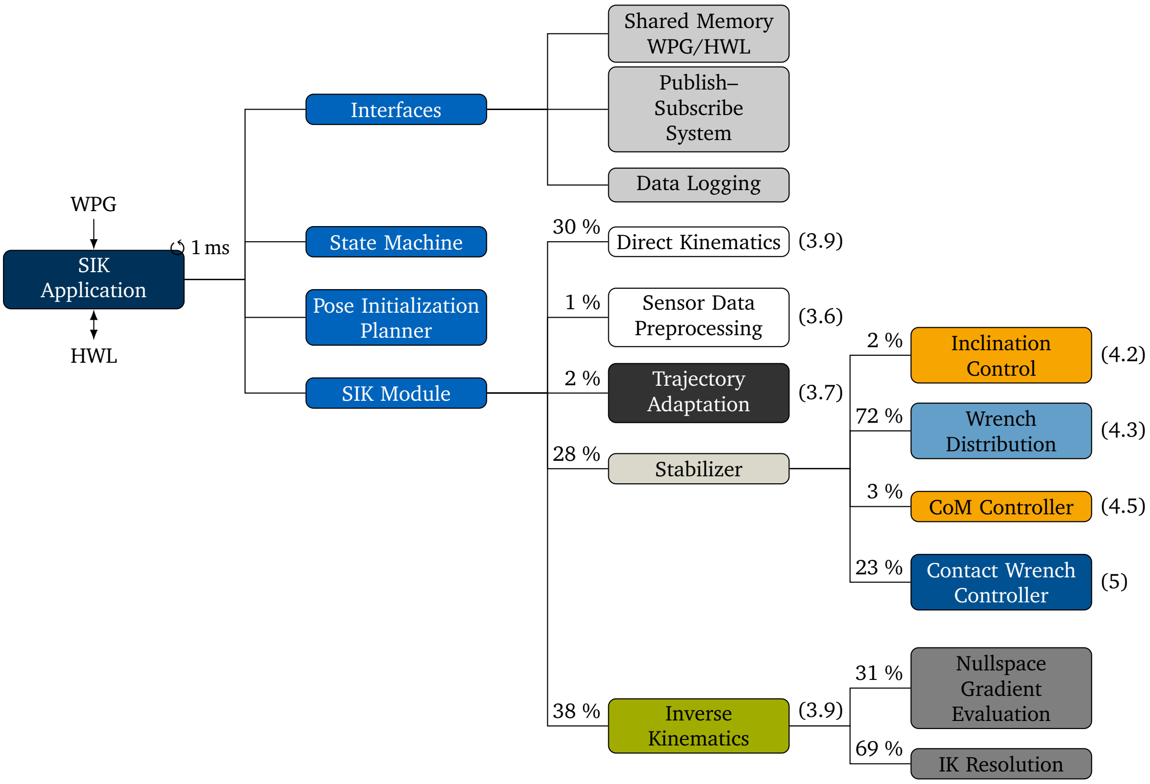


Figure 6.3: Simplified overview of SIK's software architecture showing major components. The mean relative runtimes of the modules in percent of the parent's runtime are given where available. The runtimes are measured with the *broccoli* high-precision timer for a multi-contact experiment. The numbers in brackets indicate the chapter or section describing the corresponding methods.

Module and Variant	Mean [μs]	Standard Deviation [μs]	Max @99.99 % Confidence [μs]
Wrench Distribution Heuristics	1.3	0.2	2.1
Wrench Distribution Optimization	57.3	7.7	87.3
Contact Control Non-Adaptive	5.1	0.8	8.2
Contact Control Adaptive	14.1	2.3	23.2
SIK Module	322.0	17.7	391.0

Table 6.1: Measured runtimes of several modules gathered from multiple walking and multi-contact experiments with LOLA. A data point is acquired for every control loop cycle (@1 kHz); the duration of the measurements ranges from 12 s to 19 s. The worst values from all experiments are used and several different walking sequences are used to compare the runtimes of specific methods. The clock resolution for the measurement is 1 ns; the delay caused by the time measurement itself is $\approx 0.18\,\mu s$ — it is identified from two consecutive measurements in a test application.

Further runtime measurements are summarized in Table 6.1. The QP-based wrench distribution method is significantly slower than the simple heuristics. Also, the activation of the adaptive control law significantly increases the runtime for the contact force controllers. Nevertheless, the total execution time for the SIK Module is theoretically low enough for a 2 kHz operation of SIK. However, the FT sensor data from the feet is currently only available at 1 kHz and the network-attached storage does not support data-logging rates > 1 kHz. A direct EC interface for the FT sensors and local storage on the onboard computing unit of LOLA may be used in the future to mitigate these bottlenecks.

Compared to the previous implementation, the refactored control modules and structures offer higher flexibility concerning future software changes or the integration of completely different control architectures. The refactoring further made major required changes possible in the first place, e.g., multi-contact support or the new task-space representation with rotation vectors. Special attention is devoted to the descriptive naming of components, methods, and objects to make the code easier to understand. The single-responsibility principle is applied wherever possible. The “standardization” of software modules by extraction to the *broccoli* library reduces the code base’s complexity and maintenance effort.

6.4 Further Software Enhancements

The codebase has been improved in various other parts. Enhancements on the toolchain are the transition from *QNX* Neutrino version 6 to version 7 with 64bit and C++14 support. Moreover, the build system configuration and folder structure have been reworked, and a Continuous Integration (CI) pipeline with automatic tests, test coverage, and code-style checking has been added. It should still be emphasized that even for research projects with small teams (2 people), it makes sense to set up an extensive CI pipeline. Productivity increases in the short term and the long term as the code becomes more readable and tested. The alternative often is that researchers do not test their code and implement a rough proof of concept. While this strategy can be very time-effective, it comes with high long-term costs.

Simulation Environment The high-level interface of the simulation environment has been extensively refactored. This involved removing a legacy solver structure and the replacement with the *broccoli/ode/integration* design. Following the new tree definition for solvable objects, the first virtual devices (simulation models for joint controllers) have been extracted from a monolithic robot simulation class. However, further refactorings are required to retrieve fully modular components of the simulation with loose coupling between the joint controllers, sensors, electrical- and mechanical systems. Individual improvements and refactorings are listed in the following.

- **Object Creation:** A further effort has gone into a refactored creation interface for simulation components. This enhancement enabled the definition of configurable, automated system tests of the virtual LOLA. Conditions for failure or success may be defined by attaching diagnostic listener objects to a virtual LOLA instance. Heuristics on the floating-base inclination are used, for example, to check for failure. Similarly, modifier objects may be used to alter the simulation's behavior, e.g., apply a virtual external force on the torso during the simulation.

- **Contact Model:** The contact with the environment is described via a set of uncoupled Kelvin-Voigt elements [33]. The solver and creation code for these contact elements has been refactored to simplify the parametric generation of contact surfaces and allow the integration of contact points at the robot's hands (multi-contact).

- **Joint Controller Model:** The virtual joint controllers' implementation (models of the joint drives) contains several new components to improve simulated joint tracking accuracy. The additional components are documented in the *Elmo Gold* drive user manual [64] and include an anti-windup and saturation of the Pulse-Width Modulation (PWM) voltage signal. Furthermore, the slope of a current command is additionally limited according to the real drive's implementation.

- **Sensor Models:** The tactile sensor model from Rainer [250] has been added. Moreover, several improvements of the IMU and FT sensor models, for example related to the sensor noise, lead to a more accurate simulation of these devices.

Utilities Regarding utilities, an important enhancement is an *Eigen/matvec* interface. Because parts of the codebase still use the legacy linear algebra library *matvec*, a transparent interface between the two libraries has been implemented. This enables a step-by-step refactoring or replacement of legacy code components. However, the approach in many cases requires making copies of the data due to the missing 16-byte alignment for *matvec* variables. Thus, it is still advisable to minimize the number of conversions by porting the legacy code completely to *Eigen*.

6.5 Learning the Noise of Failure: NoisyTest

The content in this section is published as preprint [293].

Experimental testing of new control algorithms for robots is often time-consuming and sometimes a safety issue. However, it is common to reduce the time spent on experiments and run simulated system tests of the robot with virtual hardware and environment. The robot can be tested safely by simulating its movement using the full multi-body dynamics, an environment model, and the actuator/sensor characteristics. Failure evaluation can either be done by manual inspection of the simulation results or as part of automated system tests in combination with distinct conditions for failure or success. Based on the used simulation models' granularity, the results can be pretty close to real-world observations. Nevertheless, there will undoubtedly be unmodeled effects, which require final experimental tests.

The experience gained from experiments with LOLA indicates that simulation data often contains clues on the robot's code flaws. Still, these minor issues often remain undetected in an automated or manual inspection of the data because the robot generally achieves the high-level goal, i.e., it does not fall or incline abnormally. The standard heuristics used for automated testing on LOLA are not sensitive enough to detect these soft failures. However, one would immediately hear something is wrong from the robot's noise in a corresponding real-world experiment. Possible error causes are oscillating feedback loops, hard impacts between feet and the ground, or discontinuities in the desired trajectories.

Therefore, the question with which this section is concerned is: Is it possible to reliably extract the hidden information on soft failures from simulation data to use it for automated system tests?

6.5.1 Simulation Environment and Dataset

The proposed failure detection method is applied to data from LOLA's simulation environment. All data shown in the following is simulated for the 24 DoF version LOLA 2020 (fig. 2.1c). An application to the current 26 DoF simulation is straightforward yet requires the regeneration of training and validation data.

The utilized full simulation model of LOLA considers rigid multi-body dynamics, contacts with the environment, and joint drive models — consisting of a decentralized joint controller, electrical system, motor, and gearbox [33, 80]. Furthermore, arbitrary walking sequences and obstacles can be defined. The obstacles are, in this case, unknown to the control software of the humanoid. The relevant outputs from the simulation are the desired and actual joint trajectories $\boldsymbol{q}_{\mathrm{J,d}}, \dot{\boldsymbol{q}}_{\mathrm{J,d}} \in \mathbb{R}^{24}$, $\boldsymbol{q}_{\mathrm{J}}, \dot{\boldsymbol{q}}_{\mathrm{J}} \in \mathbb{R}^{24}$, and the floating-base trajectories $\boldsymbol{q}_{\mathrm{FB}}, \dot{\boldsymbol{q}}_{\mathrm{FB}} \in \mathbb{R}^{6}$. The simulation output is already validated using several heuristics to check, for example, that the robot did not fall.

6.5.2 Noise-Related Feature-Extraction

The simulation results for a biped like LOLA are high-dimensional time-series data. The processing of such data for machine learning requires high computational effort both for training and evaluation. To make the approach computationally efficient, the data is compressed in the following using engineering domain knowledge. As motivated above, the real robot's noise changes significantly when one of the soft failures occurs. Therefore, a scalar signal related to the robot's radiated noise is generated from the raw simulation data. Notably, the signal must not have a direct physical relationship with the real noise; it must only contain the same features as the robot's sound at points in time where failures occur.

To derive this coarse approximation of the noise radiated by the virtual robot, let's consider a single robot link i with rotational DoF q_i. The radiated sound power of this body depends on the average velocity of the body's surface $\bar{\dot{\boldsymbol{r}}}$ [55]:

$$P_i(t) = \rho\, c\, \sigma_i A_i\, \bar{\dot{\boldsymbol{r}}}^2(t), \tag{6.1}$$

with the density of the air ρ, the speed of sound c, the surface area A_i and the radiation efficiency $\sigma_i \in]0,1[$. In particular, the sound power depends on the squared average velocity of the oscillating surface. The simulation environment assumes rigid bodies, which means the surface velocity only depends on the joint velocity $\dot{q}_i$ and the velocity of the parent links. Because only the relative joint velocity adds unique features to the noise estimate, a body's radiated sound power feature signal uses the relation

$$\hat{P}_i(t) = \dot{q}_i^2(t). \tag{6.2}$$

For the total noise power feature signal over all n joints, the sum of the individual bodys is used

$$\bar{P}(t) = ||\dot{\boldsymbol{q}}||^2(t), \tag{6.3}$$

with $\dot{\boldsymbol{q}} \in \mathbb{R}^n$. This already provides a feature signal related to the robot joints' radiated sound power caused by the actual movement. The scaling between the noise coming from the individual joints is lost with this approximation. However, although the relative weighing of the noise sources is lost, the signal still contains features for events local in time. Besides the data from the actual robot configuration, the sound power related to the desired movement is added to get

$$\hat{P}(t) = ||\dot{\boldsymbol{q}}||^2(t) + ||\dot{\boldsymbol{q}}_{\mathrm{d}}||^2(t), \tag{6.4}$$

with the desired joint velocities $\dot{\boldsymbol{q}}_{\mathrm{d}} \in \mathbb{R}^m$. This ensures that features only visible in the desired movement — $\boldsymbol{q}$ will in general not follow the desired $\boldsymbol{q}_{\mathrm{d}}$ exactly — are also part of the noise feature signal. By assuming radial and even distribution of this sound power, the observed pressure at a large distance d to the bodies is [55]:

$$p(t) = \sqrt{\hat{P}(t)\frac{\rho c}{4\pi d^2}}. \tag{6.5}$$

Again, only the basic physical relation is used by removing all constants from the equations to obtain the final noise feature signal:

$$\hat{p}(t) = \sqrt{||\dot{\boldsymbol{q}}||^2 + ||\dot{\boldsymbol{q}}_{\mathrm{d}}||^2}. \tag{6.6}$$

In the case of LOLA 2020 (fig. 2.1c), $\boldsymbol{q} = [\boldsymbol{q}_{\mathrm{J}}^T, \boldsymbol{q}_{\mathrm{FB}}^T]^T \in \mathbb{R}^n$, $\boldsymbol{q}_{\mathrm{d}} = \boldsymbol{q}_{\mathrm{J,d}} \in \mathbb{R}^m$ holds for the noise feature signal, with $n = 30$, $m = 24$.

6.5.3 Failure Symptoms and Scenarios

Several test scenarios are considered for the generation of training and validation data with the robot simulation. In all those scenarios, the robot does not fall; all these tests are considered successful by classical system tests. Still, the following symptoms may indicate a failure in these simulations:

- *Hard impacts with the environment*: Faulty trajectories of the feet may lead to impacts on the ground - even for walking on level ground. Also, inclinations of the robot's floating base can lead to unexpected early contacts (impacts) even when the inclination is low enough to continue walking.

- *Unusually high joint accelerations*: Badly parametrized control systems without limitation of the actuating variable may lead to velocity jumps or otherwise high accelerations.

- *Oscillations*: Unstable control loops may lead to oscillations of the actuated bodies. The amplitude of these vibrations is usually small and does not cause the robot to fall — still, it is an undesired behavior.

Nr	Scenario Description	Intentional Defect	Parameters	Contained Labels
1	Walking straight	-	-	*OK*
2	Curve walking	-	-	*OK*
3–4	Walking sideways, forward, backward, on the spot	-	Variation of step sequence and step duration	*OK*
5–9	Walking over an undetected obstacle	Unexpected positive ground height change (obstacle)	Variation over different initial contact points with the obstacle (full foot, heel only, toe only, side only) resulting in different impact times in the simulation data	*OK*, *Impact*, *HighAcc*
10	Walking straight	Decalibrated right ankle joint; the right foot is inclined on touchdown	0.05 rad calibration error	*OK*, *Impact*
11–19	Walking straight	Multiple sine-wave oscillations superposed to the vertical task-space position of the feet	Multi-sine with $1\,{}^{\text{sine}}/_{\text{Hz}}$, amplitude: 2 mm; several experiments with different frequency ranges: {20–30, 30–40, 40–50, 50–60, 60–100, 100–150, 150–250, 250–350, 900–1000} Hz	*Oscillations*
20–21	Walking straight	Unexpected positive ground change (obstacle) only on left foot; leads to repeated early and late contact	Height deviation: 2 cm; variation of step height and step duration	*Impact*
22	Walking down a platform	Unexpected negative ground height change; the last step on the platform is a partial contact; causes the control scheme to execute dynamic foot motions with high accelerations.	6 cm platform height	*HighAcc*, *Oscillations*
23–24	Walking straight	Faulty FT sensor signals with 400N force or 30Nm torque signal jumps	-	*HighAcc*, *Oscillations*

Table 6.2: The simulation scenarios and defects used for the generation of training and validation data. Intentional defects are changes to the simulation environment that influence the robot's performance and are not known to the control software. In the parameter column, variations of a scenario and the corresponding parameter ranges are described. Every variation has an individual scenario number. The last column indicates failure symptoms (manually) annotated for a certain time-region of the data.

Based on the failure symptoms, four different labels for a block of time-domain noise feature data are defined: $cls = \{$*OK, Impact, HighAcc, Oscillations*$\}$. The noise feature data is generated from simulation runs for the scenarios defined in Table 6.2.

The scenarios were designed to generate noise data for the failure symptoms in multiple ways while still relating to real-world problems. The data is manually annotated in the time-domain, i.e., every time block of simulation data is linked to one label. The start and the end time of a block is set manually based on the data. The total set of 83 time-data blocks corresponds to a total of 45.7 s annotated noise signal training data, and 27.6 s annotated noise signal validation data — both at a sample rate of $f_s = 10\,\text{kHz}$.

Note that an observed symptom can indicate a failure in one scenario but can be completely valid in another. For example, in scenarios 5–9, 20–21, and 22, the impact or high accelerations are expected, and the control algorithms react accordingly. Nevertheless, the testing tool is trained on these features because an impact that does not happen exactly when an unknown obstacle is hit, may indicate a (software) failure. The interpretation of *NoisyTest*'s results depending on the context of the test scenario is considered the task of a higher-level system test software.

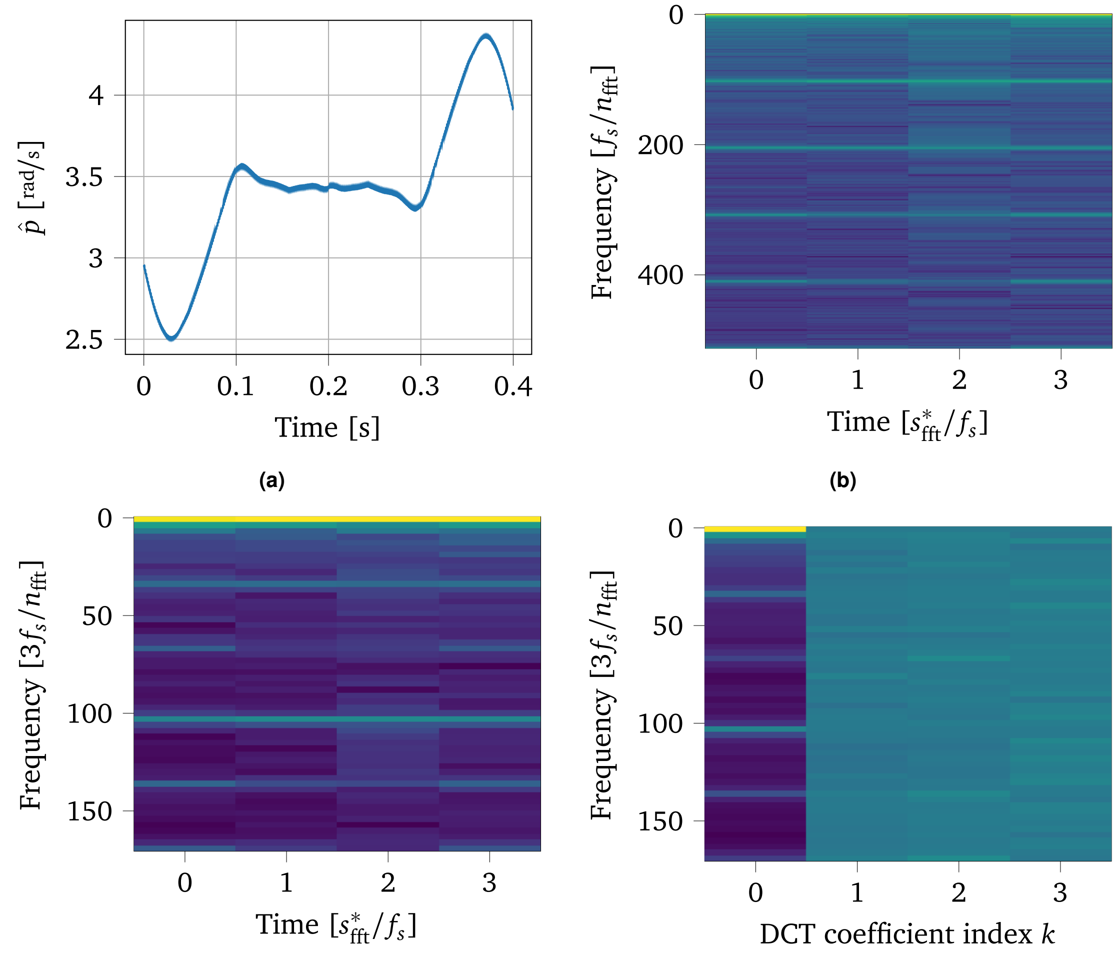
6.5.4 Signal Preprocessing

The preprocessing step transforms the noise feature signal in the time-domain to a meaningful, dense representation with preferably high relation to the failure classes. The structure of the preprocessing pipeline is motivated by techniques used in speech recognition systems. In the following, all steps of the pipeline are described. Several parameters for the preprocessing pipeline and classifier are introduced. Variables marked with a superscript $\star$ are subject to a later hyper-parameter search, see Section 6.5.6; the values of all other parameters are directly motivated within this section.

Framing The incoming labeled noise signal blocks all have a different duration/number of samples. By framing the data, all blocks are transformed to the same number of samples n_f. Failure patterns in the noise signal are expected to fit in a time frame of $\Delta t_F = 0.4\,\text{s}$; consequently, the frame size is set to $n_f = 4000$ samples. Given a data block with arbitrary size n_c, the following cases exist:

(1) When $n_c \geq n_f$, several overlapping frames of size n_f are extracted using a stride length $s = 1000$ samples for a sliding window. This increases the total number of training frames and pertubates the time-domain data by time shifts. All extracted frames share the same label data of the original data block. Only windows of size n_f are extracted, i.e., the shifted window must completely fit into the larger data block. The parameters n_f and s are set based on expert knowledge of possible data patterns and a coarse parameter search. They directly influence the size and composition of the data set and are considered fixed in the following.

(2) When $n_c < n_f$, the block is padded with $n_f - n_c$ samples at its end by reflecting the samples at the edge value both in the direction of the time and amplitude axis, see Figure 6.4a. This way, the padded frame is continuous at the edge points and resembles the original data in a repeated manner. Padding is only required during training to avoid mixing data with different failures when these failures occur in a short period. For validation data, always a full frame is used without padding and the whole block is labeled with the



(a)

(b)

(c)

(d)

Figure 6.4: Exemplary visualization of the 4 preprocessing steps. a) shows framed time data. The data is padded at its actual end ($t = 0.2\,\mathrm{s}$). b) and c) show the original and compressed spectrograms, respectively. This visualization uses $s^{\star}_{\mathrm{fft}} = 834$. The colors encode low (blue) to high (yellow) values of the absolute log-scaled magnitude [log(rad/s)]. d) shows the final DCT coefficients (before flattening). The colors encode low (blue) to high (yellow) values of the coefficients [log(rad/s)].

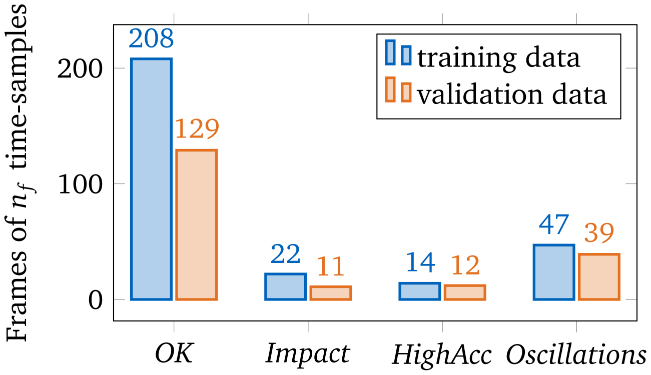


Figure 6.5: The used data sets and their failure symptom distribution. Note that the events of *Impact* and *HighAcc* symptoms are usually short, thus yielding a lower number of frames.

corresponding annotation data. This is required because test input data is always processed in full frames (with unknown failure location).

This preprocessing step's output are $N_T = 291$ training data frames of n_f time samples each, and $N_V = 191$ validation data frames. The failure symptom distribution of the data set is shown in Figure 6.5.

Spectrogram Generation In the second step, the time series data is de-correlated by transformation to the frequency domain. This step improves the performance of downstream machine learning approaches, which perform best for uncorrelated input data. To preserve information on the noise signal changes over time, a short-time Fourier transformation is performed on every frame. The performed Fast Fourier Transform (FFT) uses $n_{\text{fft}} = 1024$ points with Hann window and a stride length $s^{\star}_{\text{fft}}$, resulting in n_t different transforms. The FFT length is chosen based on the minimum resolvable frequency $f_{\text{min}} = {}^{f_s}/_{n_{\text{fft}}} \approx 9.8\,\text{Hz}$ — a value where typically the lowest mechanical eigenfrequencies of robots reside [16]. The amplitudes resulting from the FFTs are used for the spectrogram, see Figure 6.4b; phase data is not used. The output of this preprocessing step is a $[N_T \times n_t \times 513]$ training data tensor and a $[N_V \times n_t \times 513]$ validation data tensor.

Frequency Domain Compression The resolution of the fourier transform in frequency domain is ${}^{f_s}/_{n_{\text{fft}}} \approx 9.8\,\text{Hz}$, which leads to ${}^{n_{\text{fft}}}/_{2} + 1 = 513$ frequency bins for the frequency range $[0, {}^{f_s}/_{2}]$. Several tests of the whole classification architecture have shown this high resolution to be unnecessary for the classification task. Thus, the frequency spectrum is compressed with a factor of 3 by binning the frequency samples into ${}^{(n_{\text{fft}}/2+1)}/_{3} = 171$ linear-spaced bins. The magnitude for three original frequency bins is accumulated (summed up) into one resulting frequency bin. This step has shown to significantly improve the accuracy on the validation data set, as it assumably reduces overfitting the training data. The amplitudes of the spectrogram are log-scaled after compression, see Figure 6.4c. The output of this preprocessing step is a $[N_T \times n_t \times 171]$ training data tensor and a $[N_V \times n_t \times 171]$ validation data tensor.

Resolving Time-Locality The spectrogram contains information on the evolution of the frequency spectrum over time. When using this data as input for a classical machine learning approach, time locality must be considered. The input data should be independent of the relative time in the spectrogram where a failure event occurs. Otherwise, the flattened spectrogram (feature or input vector) may be completely different for two identical failure events, making it more difficult to learn a common model. Therefore, a 1D type-II Discrete

Cosine Transform (DCT) [5] in the orthogonal form is applied on every frequency row of the spectrogram:

$$\begin{aligned}
\mathrm{dct}\colon \mathbb{R}^{n_t} &\to \mathbb{R}^{n_t} \\
x_n &\mapsto X_n, \text{ with} \\
X_0 &= \frac{1}{\sqrt{n_t}} \sum_{n=0}^{n_t-1} x_n, \\
X_k &= \sqrt{\frac{2}{n_t}} \sum_{n=0}^{n_t-1} x_n \cos\left[\frac{\pi}{2}\left(n+\frac{1}{2}\right)k\right], \\
k &= 1, \ldots, n_t - 1.
\end{aligned} \tag{6.7}$$

This way, time-locality is resolved and the temporal evolution is stored in form of DCT coefficients for every frequency bin, see Figure 6.4d. More information on DCT transforms can be found in [219]. The final preprocessing step flattens the output to a $[N_T \times 171\, n_t]$ training data tensor and a $[N_V \times 171\, n_t]$ validation data tensor.

6.5.5 Classification via Support Vector Machine

This section describes the classification approach used to automatically learn the relations between the preprocessed noise signal and the annotated failure labels. The classification uses a classical machine learning tool, the Support Vector Machine (SVM). SVMs have the advantage to be effective in high dimensional input spaces, even when the number of features in input data is higher than the size of the dataset — which is the case here. Furthermore, they are resource efficient and provide adequate training times for small datasets [259, 281].

Two-Class SVM Formulation *NoisyTest* uses the *libsvm* [46] implementation of the C Support Vector Classification (C-SVC) approach [25, 53]. Given a data set of n-dimensional vectors $\boldsymbol{x}_i \in \mathbb{R}^n, i = 1, \ldots, m$, and corresponding binary label data $\boldsymbol{y} \in \mathbb{R}^m, y_i \in \{1, -1\}$, C-SVC looks for the weights $\boldsymbol{w} \in \mathbb{R}^l$ and bias b by solving the following primal optimization problem [46]:

$$\min_{\boldsymbol{w}, b, \xi} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{m} \xi_i \tag{6.8}$$

$$\text{s.t.} \quad y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i) + b) \geq 1 - \xi_i, \quad i = 1, \ldots, m \tag{6.9}$$

$$\xi_i \geq 0, \quad i = 1, \ldots, m, \tag{6.10}$$

where $\phi(\boldsymbol{x}_i) : \mathbb{R}^n \to \mathbb{R}^l$ maps into the higher-dimensional space $l > n$, and $C > 0$ is a regularization parameter with the constraint error variables ξ_i. The value of C is inversely proportional to the l_1 regularization of the weights $\boldsymbol{w}$, which helps circumvent overfitting the training data. Regularization of $\boldsymbol{w}$ tries to keep the weights minimal, i.e., use a minimal set of weights to represent a decision boundary.

For a high dimensional $\boldsymbol{w}$, it is computationally expensive or even unfeasible to solve the problem in its primal form. Instead, the corresponding dual formulation is used [46, 53]:

$$\max_{\boldsymbol{\alpha}} \quad -\frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{Q}\boldsymbol{\alpha} + \sum_{i=1}^{m} \alpha_i \tag{6.11}$$

$$\text{s.t.} \quad \boldsymbol{y}^T \boldsymbol{\alpha} = 0, \tag{6.12}$$

$$0 \le \alpha_i \le C, \;\; i = 1, \ldots, m, \tag{6.13}$$

$$\text{where} \quad Q_{ij} = y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) = y_i y_j \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j). \tag{6.14}$$

An explicit calculation of the high-dimensional vector $\phi(\boldsymbol{x}_i)$ is no longer necessary with this formulation. Instead, the scalar kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is directly calculated from its arguments without explicit mapping to the l-dimensional space. Furthermore, at the optimal solution, only a few of the Lagrangian multipliers α_i are non-zero — namely those which correspond to the *support vectors* [53]. This set of input data vectors $\boldsymbol{x}_i$ defines the decision boundaries in the input data hyperplanes.

Common choices for the kernel function are the linear kernel $K_{\text{linear}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^T \boldsymbol{x}_j$, and the Radial Basis Function (RBF) kernel $K_{\text{rbf}} = \exp\left(-\gamma^\star ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2\right)$. From the solution of the dual problem and the primal-dual relationship, the decision function for classification of an input vector $\boldsymbol{x}$ results [46]:

$$\hat{y} = \operatorname{sgn}\left(\boldsymbol{w}^T \phi(\boldsymbol{x}) + b\right) = \operatorname{sgn}\left(\sum_{i=1}^{m} y_i \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b\right). \tag{6.15}$$

Multi-Class SVM Formulation To extend the binary SVM classification to multiple classes in the data, the “one-vs-rest” approach is deployed [259]. For each class *cls* in the dataset, a separate binary C-SVC is used to learn the appearance of the class in the data. The C-SVCs share the same parameters and kernel, except for the regularization parameters C_{cls}, which are set inversely proportional to the respective class frequency ${}^{N_T}/_{N_{T,cls}}$ of the training data set: $C_{cls} = {}^1/_4 {}^{N_T}/_{N_{T,cls}} \hat{C}^\star$. The global regularization parameter $\hat{C}^\star$ is subject to the hyperparameter search described in Section 6.5.6.

6.5.6 Results

NoisyTest is implemented in *python3*. It uses *scikit-learn*’s [235] wrapper to *libsvm* [46] for the support vector machine, and *TensorFlow* [1] for data preprocessing. The *NoisyTest* code [290] and used data set [289] are available online.

Hyperparameter Search The preprocessing pipeline and the SVM implementation have a set of hyperparameters, which need to be tuned to the data set and application for maximum performance. These parameters are the FFT window stride length $s^\star_{\text{fft}}$, the regularization parameter $\hat{C}^\star$, and the kernel parameter $\gamma^\star$ for radial basis function kernel. The best set of parameters maximizes all classes’ accuracy (subset accuracy) on the validation data set. It is found with a grid search by training the SVM for every parameter combination and evaluating the accuracy on the validation data set. For more details, refer to the implementation. The resulting optimal hyperparameters are shown in Table 6.3. The sensitivity of the hyperparameters for the RBF kernel is shown in Figure 6.6.

Accuracy Evaluation The accuracy of *NoisyTest* is evaluated on the validation data set. Due to the relatively low sensitivity of the hyperparameters (see Figure 6.6) and the small data

Parameter	Linear kernel	RBF kernel
FFT stride length $s^{\star}_{\text{fft}}$	401	834
Inv. regularization $\hat{C}^{\star}$	1.0	1.1
Kernel parameter $\gamma^{\star}$	-	5.7×10^{-4}
n_t	8	4
SVM input vector size	1368	684

Table 6.3: Optimal hyperparameter values and terms resulting from this selection.

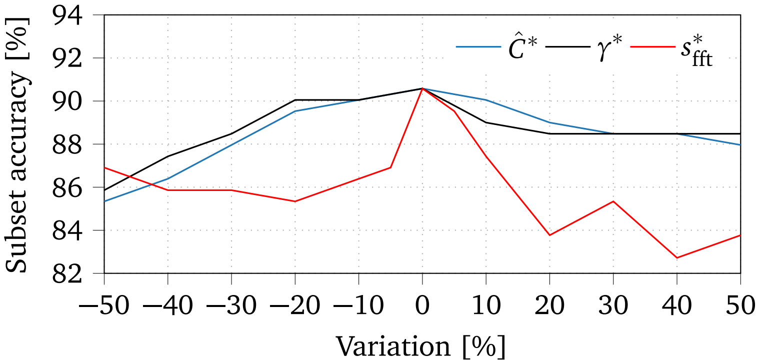


Figure 6.6: Accuracy for a variation of one parameter from its optimal value for the RBF kernel.

set, no extra test data set is used. The results for the linear and radial basis function kernel are summarized in Section 6.5.6. Both approaches perform well in terms of the overall detection of failures (*OK* vs. rest). The RBF kernel performs best in terms of subset accuracy. However, it shows a non-zero false-negative rate for the *OK* class, i.e., falsely detected failures. In the following, only the results of the RBF kernel variant are discussed.

The shown accuracy data and additional evaluations with reduced validation data set show that the undetected failures are caused by the *Impact* symptom in scenario 21. Furthermore, the *HighAcc* events in scenario 24 are falsely detected as *Impact* and *Oscillations* symptoms. However, the torque signal jumps induced in this scenario are hard to classify. They are closely related to the effects of an impact and certainly also lead to subsequent oscillations. In reality, it is likely that the classes only occur together or in close succession. Nevertheless, identifying the possible failure symptom (instead of just a single-class evaluation) may provide the user with additional hints on the failure cause.

	OK	*Impact*	*HighAcc*	*Oscillations*
False negative errors **linear** kernel	0.0 %	1.0 %	5.8 %	4.7 %
False negative errors **RBF** kernel	0.5 %	2.6 %	6.3 %	0.0 %
False positive errors **linear** kernel	5.8 %	5.8 %	0.0 %	0.0 %
False positive errors **RBF** kernel	2.1 %	2.6 %	0.0 %	4.7 %
Failure detection rate **linear** kernel	94.2 %			
Failure detection rate **RBF** kernel	97.9 %			
Subset accuracy **linear** kernel	88.5 %			
Subset accuracy **RBF** kernel	90.6 %			

Table 6.4: Achieved accuracies on the validation data set.

	OK	*Impact*	*HighAcc*	*Oscillations*
False negative errors	1.6 %	4.2 %	6.3 %	7.3 %
False positive errors	9.4 %	7.9 %	0.0 %	2.1 %
Failure detection rate			90.6 %	
Subset accuracy			80.6 %	

Table 6.5: Achieved RBF-kernel SVM accuracies on the validation data set with reduced noise feature signal. Only the knee and ankle joints of both feet are used.

An additional test is conducted to evaluate if the SVM learns the concrete features of a failure independently of the number of involved joints. For this check, a reduced noise signal for LOLA was generated using only the knee and ankle joints of both feet. The reduced number of involved joints results in a completely different time-domain signal. Evaluating the trained SVM with RBF kernel on a separated validation set with the reduced noise estimate still gives reasonable detection performance, see Table 6.5.

Runtimes The hyperparameter search takes several hours on a standard notebook computer. However, one training and validation run on the presented dataset takes only ≈ 8 s. Checking 10 s of simulated noise data takes ≈ 4 s, which makes the algorithm generally real-time capable. Furthermore, most of the runtime required for checking is related to loading the required libraries and not the actual evaluation of the SVM.

6.5.7 Discussion in the Context of Related Work

Typical fault detection and fault diagnosis approaches for robots focus on detecting failures during their use in a production environment — caused by faulty sensors or actuators [153]. Common techniques for manipulators use the residual of a model-based observer on the system's dynamics to detect and isolate actuator failures [60, 62, 313]. These methods target the detection of failures on a real robotic system. The detected failures are typically unrecoverable, like a collision or actuator malfunction. For testing control methods in a simulation environment, these observer-based approaches are unnecessary because the state of the robot and the environment are known anyway. In contrast, the application scope and the approach for tests with *NoisyTest* are different. By training a model on the robot's expected standard behavior and on several symptoms that might indicate soft failures (performance degradation), a statement on the quality of the robot's performance is possible. This allows identifying possible performance restrictions at an early stage in the development process. For LOLA, the normal closing of foot contacts with the environment is, for example, considered normal by the tool. However, an abnormally strong impact in one of the steps (which may indicate a software flaw) is immediately recognized as outside normal parameters. Although these detections may in some cases be possible with simple heuristics, the automatic learning of the default behavior, which can also change over time, makes manual tuning of failure thresholds superfluous.

Vibration signals are commonly analyzed with machine learning approaches for fault diagnosis of rotating machinery [182]. Typically, measured acceleration signals are transformed into the frequency domain, and the levels of individual subbands are then fed into neural networks or support vector machines [82, 193]. Alternative approaches use wavelet transform coefficients [96] or a large set of features from classical signal analysis [255] as input data. In contrast, the work presented here uses a simulated signal related to the virtual system's noise — not a vibration (acceleration) signal of a real system. Moreover, the preprocessing pipeline uses a short-time Fourier transformation of the estimated noise signal and compresses the resulting spectrum instead of using individual frequency subbands.

Note that the signal that is motivated by the noise of the robot's bodies has no direct relationship with the system's real-world noise. This is actually not necessary for the successful application of *NoisyTest* in simulation — the signal is just used to carry features that are usually also part of the real system's acoustic noise. Because the signal is scalar and the results show that also a reduced noise feature signal (from a subset of joints) still yields high failure detection accuracy, it is assumed that a single trained model may even work for different robots. To prove this, further simulation tests for different robots are required.

The use of a *NoisyTest* model trained in simulation for experimental data for LOLA turned out to be difficult. For a standard walking sequence, several false positives are reported. However, this is somewhat expected as experimental data contains more noise, and the model was not trained on this data. Nevertheless, the use of *NoisyTest* on experiments is technically possible. Still, it is unclear if a sufficiently high accuracy can be reached on real hardware data.

While the proposed tool works to detect simple soft failures, it is still a work in progress with many possible extensions. First, it might be possible to run *NoisyTest* on several noise estimates generated from intermediate signals in the control algorithm's data flow to localize the faulty code. Second, it is interesting to test its performance on noise estimates generated from experimental data. A possible use case is a fully automatic system test of the actual robot system with a noise-based evaluation of its performance. Furthermore, the used classification technique is computationally efficient, presumably allowing online real-time diagnostics of the robot.

6.6 Chapter Summary

This chapter describes the implementation of the methods presented in Chapters 3 to 5 and gives details on the runtime of these approaches in the real-time environment. Moreover, the ecosystem for the development of software for LOLA is described, and improvements on this ecosystem contributed by this author are presented. An important part of the ecosystem is the open-source robotics library *broccoli*, a by-product of code refactoring in the LOLA project.

In addition to the software ecosystem, the gap between simulation and experiment in the context of system testing is discussed. A machine learning tool for additional virtual system testing which uses a noise-related signal is proposed. The goal is the earlier detection of soft failures caused by the control software.

Chapter 7

Performance Evaluation

In the preceding chapters, individual methods were validated in repeatable and relatively simple terrain scenarios. In contrast, the experiments presented in this chapter try to evaluate the whole system's performance in application-relevant complex scenarios. Primarily, the robustness of LOLA to all kinds of undetected terrain (changes) is evaluated. A discussion of existing limitations follows the presentation of several scenarios and the corresponding experimental results. Videos of the experiments are available online and are referred to in the corresponding sections. Note that no camera-based data nor prior information on the environment is used for all experiments shown in this thesis. The algorithms always assume the ground is flat and stiff before contact is made. The default walking gait with a velocity of 0.5 m/s is used for all experiments unless noted differently.

All methods are parameterized as described in the corresponding method sections. The only exception is the early-contact detection force threshold (see Section 3.6.2), which is retrospectively changed to the higher value $F_{ec,th,f} = 350\,\mathrm{N}$. This improves the robot's stability on soft ground while having no significant effect on the stiff-ground performance.

7.1 Inclined Uneven Terrain

https://youtu.be/jtG7kswtEo0

Three different test scenarios define the benchmark for inclined unexpected terrain, see Figure 7.1. In the first two scenarios, the step height is parametrized to 7 cm; in the last scenario, it is set to 10 cm. The ground reaction force controllers use the sensor-based contact model. However, the adaptive control law is turned off due to the slightly negative effect on the system's performance in combination with partial contacts, see Section 5.4.6. For the first scenario (Figure 7.1a), six different runs with different orientations of the board and starting positions were conducted. In four of these six experiments, the robot successfully traversed the obstacles. In the two failed experiments, an obstacle is hit exactly with the middle of a foot, i.e., none of the four footpads makes contact. Thus, the partial contact surface remains undetected (no contact switch triggers), resulting in a slower reaction of the force controllers. In this case, the early-contact reflex further relies on FT sensor data only, which delays the algorithm's activation. Over the set of successful runs, the maximum floating-base inclinations around x- and y-axis are 5.1 ° and 5.7 °, respectively. For the second scenario (Figure 7.1b), the maximum floating-base inclinations of the robot around x- and y-axis are 4.8 ° and 4.6 °, respectively. The initial drop from the platform makes no significant difference concerning the floating-base inclinations. For the third scenario, again six different runs of the experiment were conducted. Again, four of the runs were successful. In each of the

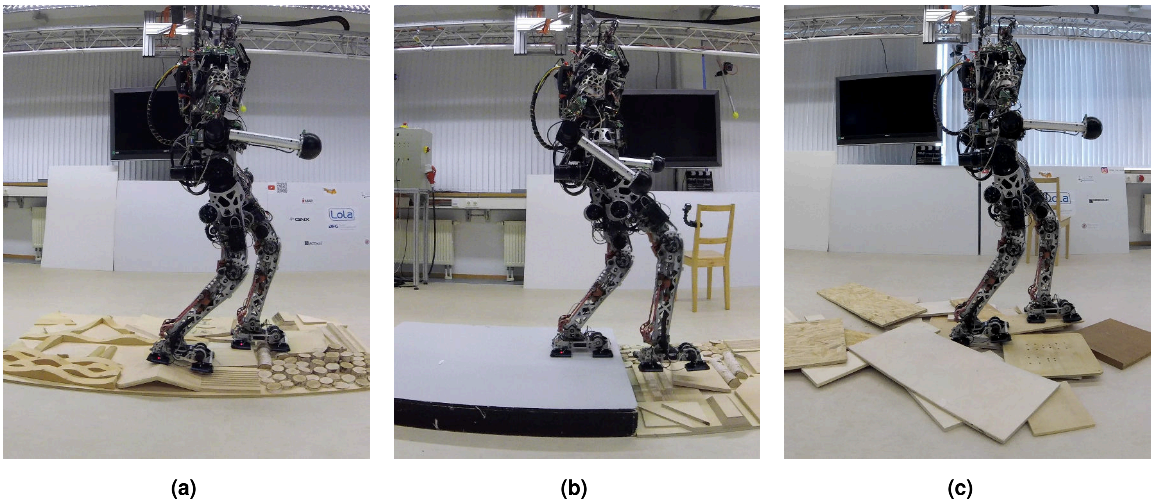


Figure 7.1: The test scenarios for uneven terrain performance evaluation: (a) a wooden plate with obstacles of maximum up to 6 cm in height, (b) a combination of a 12 cm-high platform with the wooden plate, and (c) loose wooden boards with a total height of up to 9 cm.

two failure cases, the swing foot got stuck below one of the boards, leaving no chance for the balance controllers to mitigate the disturbance. Over the set of successful runs, the maximum floating-base inclinations of the robot around x- and y-axis are 6.4 ° and 4.8 °, respectively.

7.2 Ground Height Changes

https://youtu.be/DzuFXPc2U5c

The robustness of Lola to ground height changes is already evaluated in Section 4.5 in the context of vertical CoM accelerations. In this section, further experiments are documented and described briefly. In all scenarios, a step height of 7 cm is used. In the experiments, the basic control scheme for the ground reaction forces is generally used, i.e., with a static contact model and without adaptive control law. The extensions are not strictly necessary in these scenarios. However, for the leg-in-hole experiment, the sensor-based contact model is activated, which slightly improves the performance due to the partial contact with the ground. Due to the relatively simple structure, the experiments are deterministic and repeatable. The first scenario (Figure 7.2a) contains a positive and negative ground height change (6.7 cm). The maximum contact forces are 1292 N and the maximum floating-base inclinations of the robot around x- and y-axis are 2.6 ° and 3.2 °, respectively. The maximum inclination occurs while walking down from the board, i.e., in the late-contact situation and not the early-contact situation.

In the second scenario (Figure 7.2b), the order of ground height changes (6 cm) is changed to negative first and positive second (leg-in-hole). The maximum contact forces are again 1292 N and the maximum floating-base inclinations of the robot around x- and y-axis are 2.7 ° and 4.5 °, respectively. Again, the maximum inclinations occur at the late contact when the right foot touches the ground. However, the maximum contact forces are reached when accelerating the robot upwards and not at the immediate impact (600 N). Although the ground height difference is higher in the first scenario, the maximum forward inclination is larger for the leg-in-hole experiment. However, the CoM is still moving upwards to the

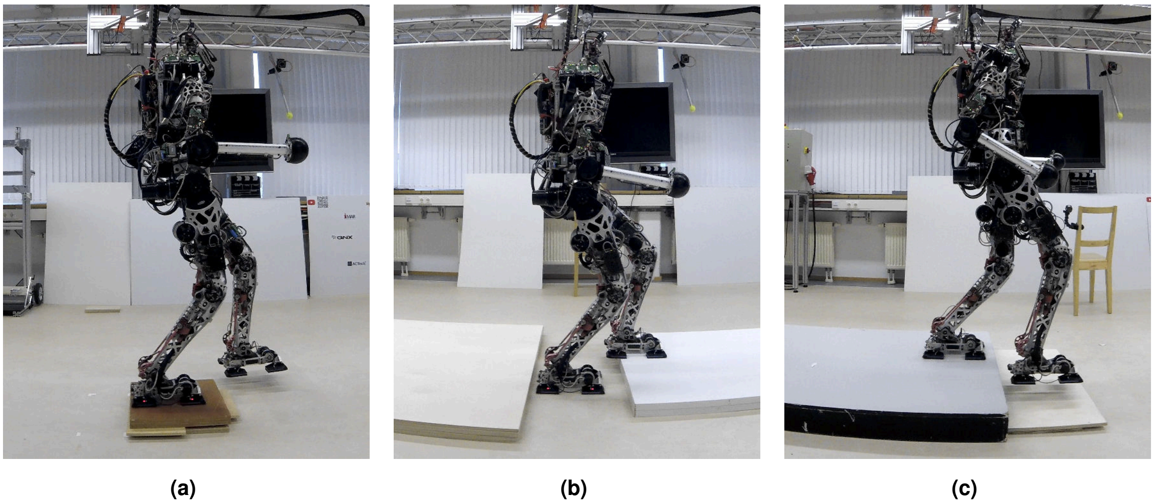


Figure 7.2: The test scenarios for performance evaluation with ground height changes: (a) a wooden board with height of 6.7 cm, (b) a leg-in-hole experiment with 6 cm height change, and (c) a drop of 9 cm from a platform.

nominal height when the late contact occurs in the first scenario, which reduces the foot's height above ground.

In the third scenario (Figure 7.2c), a higher negative ground height change (9 cm) is evaluated. The maximum contact forces are 1267 N and the maximum floating-base inclinations of the robot around x- and y-axis are 2.9 ° and 6.1 °, respectively. The disturbance on the floating-base of the robot is already significantly increased compared to the other experiments. Nevertheless, the robot robustly overcomes the ground-height change at this relatively high walking speed. Greater height differences were not tested. On the one hand, these generally seem possible; on the other hand, the platform's height may approach the robot's kinematic limits. This can lead to the swing foot being dragged over the platform after the robot stepped down, effectively generating a tilting torque in forward direction.

7.3 Hard and Soft Ground

https://youtu.be/HH2a1mNmCkY
https://youtu.be/Vzqm94bxjKI

The additional experiments described in this section show that (a) the robot is able to overcome soft ground even when multiple steps are soft, and (b) that the adaptive contact controller allows to traverse Parcours with soft and hard obstacles, see Figure 7.3. This shows that the controllers' parameters are not specifically tuned for soft or hard ground and that the control scheme is robust to different mechanical ground properties. For the soft foam, the step height is increased to 8 cm, and the timing of the horizontal foot motion is changed slightly[1] to avoid dragging the foam sheets with the swing foot. For experiments containing random obstacles, the standard walking gait with 7 cm step height is sufficient. The full set of methods is active regarding contact force control, i.e., sensor-based contact model and adaptive control law. In the first scenario (Figure 7.3a), four foam sheets are placed at the

[1] The horizontal foot motion at lift-off is delayed by 0.05 s (lag) to produce only vertical foot motions when the foam is still compressed.

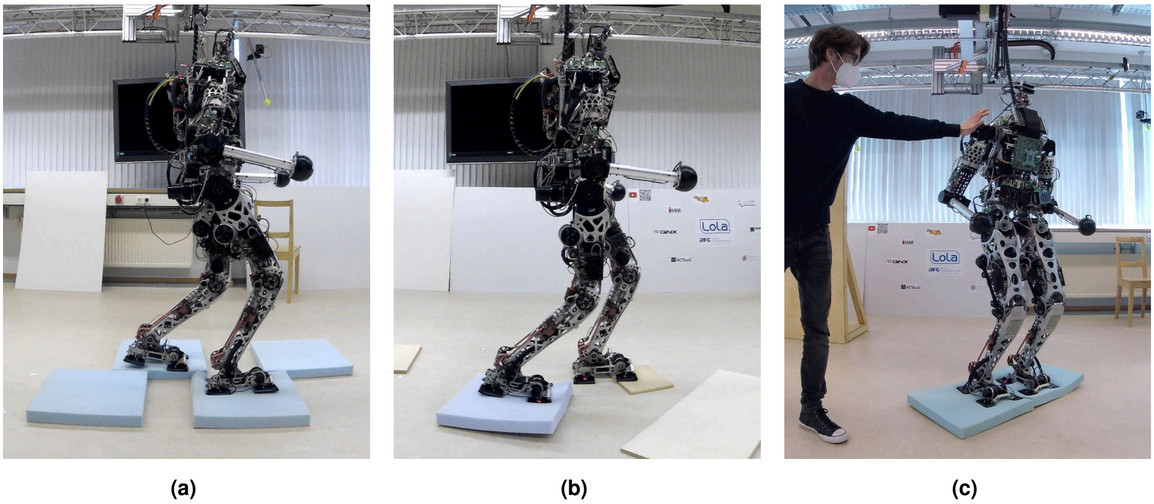


Figure 7.3: The test scenarios for performance evaluation on soft and hard ground: (a) soft foam sheets with an uncompressed height of 5 cm, (b) randomly arranged soft and hard obstacles, and (c) disturbance rejection on soft and hard ground.

footstep locations of the robot. The density of the foam is $40\,\mathrm{kg}/\mathrm{m}^3$, the compression hardness is specified to 40 % compression at 6 kPa with a uncompressed height of 5 cm. For single support the foam is compressed to ≈ 1.5 cm height. Due to the softness of the material, the foot pads *and* the FT sensor body in the middle of the foot make contact, i.e., the actual contact surface is greater than the surface of all pads. The maximum floating-base inclinations of the robot around x- and y-axis are 4.0 ° and 2.6 °, respectively.

In the second scenario (Figure 7.3b), soft and hard obstacles are randomly placed in front of the robot. Six different runs with different locations of the obstacles are conducted. In four of these experiments, the robot successfully walks over the obstacles. In two cases, the swing foot gets stuck under an obstacle, and the robot falls. Over the set of successful runs, the maximum floating-base inclinations of the robot around x- and y-axis are 5.1 ° and 3.3 °, respectively.

The third test scenario involves external pushes on the robot while standing and stamping on soft or hard ground. The video footage particularly shows the modification of the vertical CoM height for large inclinations. Furthermore, the stabilization scheme's effectiveness on soft ground is further validated with the disturbed stamping motions on the foam sheets.

7.4 Multi-Contact Balancing

https://youtu.be/gUNZ0AmLiWU

Balancing the robot in difficult terrain using additional hand contacts is evaluated in several scenarios, see Figure 7.4. The decision to use hands, the desired hand positions, and the timing when to make contact with the hands are predefined in the planner. In the future, this information may originate from a camera-based system that estimates the difficulty of the terrain and sends corresponding data to the WPG.

In the first scenario (Figure 7.4a), the robot walks over the wooden terrain plate support by both hands. The second footstep on the plate makes contact only with the middle of the foot and not the foot pads. On the one hand, this prevents the contact controllers from a

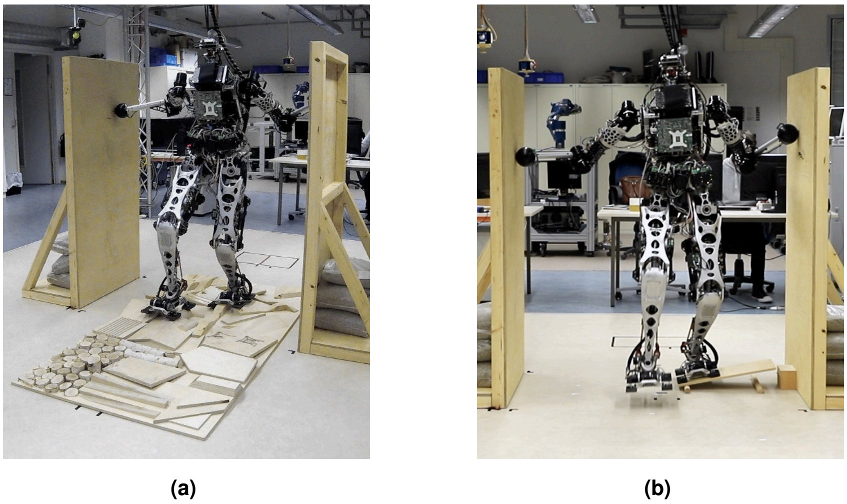


Figure 7.4: The test scenarios for multi-contact performance evaluation: (a) uneven terrain with additional hand support, and (b) rolling obstacle with additional hand support.

correct estimation of the contact surface; on the other hand, the maximum foot torques are further limited by the smaller surface. However, the hands are in contact during this step, keeping the robot balanced. Total contact forces (vector norm) up to 440 N are measured on the right hand, and up to 300 N on the left hand. Both hands simultaneously push on the walls; this results from the solution of the wrench distribution QP. Due to the higher normal forces on the wall, tangential contact forces can be supported to reduce the tilt of the biped in the sagittal plane. The maximum floating-base inclinations of the robot around x- and y-axis are 2.6 ° and 5.2 °, respectively.

In the second scenario (Figure 7.4b), LOLA steps onto a rolling wooden board, only keeping balance due to the additional hand support. The total contact force (vector norm) is up to 90 N per hand. The maximum floating-base inclinations of the robot around x- and y-axis are 5.2 ° and 6.5 °, respectively. The high inclination around the y-axis occurs when stepping down from the obstacle. During the multi-contact balancing, mainly an inclination around the x-axis is measured. This also explains the lower contact forces of the hands, because no forces tangential to the wall must be supported.

7.5 Discussion

The ultimate goal for biped robots is locomotion in unstructured, human-centered environments. This involves high robustness to uneven terrain and unexpected disturbances. To be competitive, the walking speed must furthermore be sufficiently high. This section discusses the experimental outcomes of this thesis in the context of these high-level goals. Furthermore, differences to robots with torque-controlled joints are discussed. For a broad overview of state of the art, refer to Section 1.1.

Uneven Terrain and Ground Height Changes The walking experiments with LOLA on uneven terrain demonstrate the effectiveness of the balancing and stabilization methods proposed in this thesis. Because no terrain data from cameras or prior information is used, the robot is technically blind. Therefore, one can argue that the system could tackle even higher obstacles and completely unstructured terrain when a vision system is used in the loop to adjust footstep locations and timing. The system is mainly limited by the coarse resolution of the contact surface detection (via the four discrete footpads), kinematic limitations of the robot, and the foot's shape.

The footpads offer a relatively large number of interfering edges, which can get stuck on the terrain. Also, it is not guaranteed that only the pads themselves make contact with the ground on uneven terrain. Smaller obstacles can also make contact with the housing of the force sensor between the pads. These contacts are currently not detected, and there is also no damping of the impact by the soft sole of the pads. Moreover, due to the relatively low height and large contact area, the entire foot can easily get under moving obstacles and drag them along.

In related work, unexpected uneven terrain is usually traversed quasi-statically at low walking speeds, i.e., 1.6 to 45 times slower than shown in this work, see Section 1.1. Furthermore, the uneven terrain only in one case has a similar complexity [166], i.e., the uneven terrain scenarios mostly consist of rather small ($< 4\,\mathrm{cm}$) ground height changes with rather large and flat contact surfaces. Besides the speed advantage, successful walking over moving terrain (loose wooden plates) with a biped robot has not been reported so far in the literature to the author's best knowledge. In contrast to all other robot's performance worldwide, the hydraulically actuated *Atlas* from Boston Dynamics achieves high walking speeds across uneven terrain. Nevertheless, only video footage is available for these experiments without further insight into the applied methods, used assumptions and sensor systems, or prior knowledge of the environment.

The author argues that the unexpected ground-height changes (9 cm for sure) LOLA can overcome are also challenging for humans when the change is unexpected (unseen). Furthermore, it is realistic to have a camera-based perception of the environment with errors less than 9 cm. Thus, the robot's performance in the context of these disturbances seems fully sufficient for generally robust and fast biped walking over arbitrary uneven terrain. There are only a few biped robots that can overcome unexpected height changes of this magnitude at a walking speed of $\geq 0.5\,\mathrm{m/s}$, see Sections 1.1 and 4.5.4. Notable are the planar semi-passive walkers [125, 229], which, however, have a reduced versatility of possible motions. Based on the available literature, LOLA is the first humanoid robot with fully actuated joints to have shown robustness to unexpected ground-height changes of this magnitude to the author's best knowledge.

Walking over Soft Ground The presented approaches enable LOLA to walk over hard and soft ground with a single parametrization for the controllers. Using these algorithms, the robot can overcome soft terrain significantly faster than state-of-the-art systems. In fact, experiments in related work only reach a 2.3–10 times slower walking speed over soft ground, see Section 1.1. It is further unclear if a single configuration of the robot's algorithms does work for hard and soft ground.

Limitations of LOLA's ability to traverse soft terrain are again imposed by the foot's design itself. Experiments have shown that for soft materials with a sink-in depth $\geq 10\,\mathrm{cm}$, the foot's sharp edges get easily stuck in the material and make the robot fall. Also, the results indicate kinematic limits of the robot play a more important role than the stabilization control. Fortunately, there are very few real-world examples of soft ground with a high sink-in depth that humans (or robots) have to cross.

Multi-Contact Balancing The force control concepts deployed on LOLA's feet can easily be adapted for additional hand contact force control. Combined with an extended wrench distribution method, this enables LOLA to dynamic multi-contact locomotion on uneven terrain. Conceptionally, the extension of the robot stabilization to multiple contacts is straightforward with only slight modifications. However, fast dynamic multi-contact locomotion is achieved with these approaches, which is in great contrast to the slow and quasi-static motions described in literature [45, 167, 254, 321]. Even the fastest example in literature uses a contact transition time of approximately 2 s [196] compared to 0.8 s for the multi-contact experiments on LOLA. Furthermore, the used WPG[2] and all SIK modules run in real-time, while state-of-the-art multi-contact planners take several seconds to generate corresponding movements.

Position-Controlled Joints vs. Torque-Controlled Joints The algorithms proposed in this thesis are designed for robots with position-controlled joints. In the last years, joint torque control has gained increasing attention in the robotics community. The technology is motivated by safety aspects — due to the torque sensor in each joint, collisions with the environment (or humans) can be detected, and the robot can react in a compliant way. However, the safety of robots does not necessarily require torque-control at the joint level [24]. Stiff objects surround humans, and it seems unnecessary that a robot gives way compliantly when not moving before the collision. When a robot is in motion, it must be guaranteed that the kinetic energy is limited or reduced before impact when a collision is inevitable. Provided an imminent collision can be detected — for example, from drive currents or external sensors — only the gear ratio, the mass of the robot's bodies, and the joint drives' power determine the impact force. Conceptionally, robots with torque-controlled joints can reach a higher performance for interaction tasks than robots with position-controlled joints, which require an additional outer force-control loop.

In the context of biped walking, both positions and forces are important. On the one hand, footstep positioning must be accurate because this defines the BoS and greatly influences the dynamics of the system. On the other hand, contact wrench control must mitigate impact effects from timing imperfections and is important for the system's dynamics when an end effector is in contact. It seems to be due to these requirements in both domains (positions and forces) that robots with torque-controlled joints so far have not shown significant benefits compared to the less complex position-controlled ones. Apart from the many different control frameworks for both approaches, the two necessary control loops for positions and forces basically swap places in a cascaded structure. This means that robots with torque-controlled joints have a slower position-control loop than force-control loop to preserve stability; consequently, for robots with position-controlled joints, the position-control loop is faster than the force-control loop. However, the differences are fluid, and the quality of gears, sensors, and mechanical structure introduce biases in one or the other direction.

Position-control on joint-level can be designed as collocated control when using encoder feedback on the motor-side, allowing high closed-loop bandwidth. Of course, this leads to imperfect tracking of positions on the joint side. However, the deviations are usually neglectable when the gearbox is reasonably stiff, and feedforward control approaches can be used to improve joint-side accuracy. Force control loops must operate as an outer loop, which requires external force sensors. Also, the mechanical structures between the FT sensor and the joint motors are part of the force control loop's plant and can limit the crossover frequency.

Torque-control on joint-level is usually done with a joint-side sensor to be able to detect external forces (safety). However, this means the inner force control loop is not collocated, and its crossover frequency depends on the gearbox's quality. Moreover, special actuation designs for biped robots try to shift the heavy motors upwards using belt and roller screw mechanisms. Due to the friction in these mechanisms, errors between the actual contact forces and the forces estimated from the joint torque-sensors exist. In some cases, these

[2] The achievements of multi-contact planning are not part of this thesis' contributions. For details, refer to the work of Philipp Seiwald.

effects are reduced by additional FT sensors at the end effectors, making the overall system even more complex. An advantage of joint torque control is the ability to measure contact forces on the humanoid's whole body, although it is not always possible to separate the individual contributions of multiple contact points. An outer position-control loop is required to track positional trajectories for CoM and the end effectors and avoid joint limits. In contrast to robots with position-controlled joints, the DK is always calculated from actual encoder values and not the desired configuration. This puts higher demands on the quality of the sensors and state-estimation techniques to retrieve noise-free pose estimates.

As shown in this thesis, the contact force control of robots with position-controlled joints is robust to the ground's mechanical uncertainties. These control schemes become unstable when the ground is stiffer than the nominal parametrization, which is not relevant in reality due to the foot sole's finite stiffness. In contrast, torque-control loops directly at joint-level can become unstable on soft or moving ground without further measures or reduced gains [75, 106]. The observations described for soft ground in Section 5.4 are inverted in this setup: The outer position-control loop's performance degrades on soft ground, leading to higher deviations from the reference pose than on hard ground. Depending on the parametrization, the changes of the pose may be drastic. It may be possible to transfer the MRAC approach to the outer position controller in these control schemes to get a uniform behavior for different grounds.

Each approach tends to favor either position- or force control due to the inner and outer loop location. However, both are required for biped walking, making it hard to tell if one method is generally better suited than the other. Both approaches strongly benefit from back-drivable joints (low friction) just like the drives on LOLA. Both approaches can be affected by structural resonances. Thus, the related methods described in this thesis can be transferred to robots with torque-controlled joints.

The author believes that the impact of drive components and mechanical design on performance is underestimated. A robot with high-performance position-controlled joints may ultimately have a higher force-control bandwidth than a robot with inferior torque-controlled joints. A fair comparison of the two control methods is only possible when done on hardware best suited for both approaches. To the author's best knowledge, such a comparison has not been conducted yet.

7.6 Chapter Summary

This chapter contains experiments showing the achieved overall performance of LOLA on uneven terrain, soft ground, and multi-contact locomotion. In contrast to the relatively simple validation experiments for the individual control methods, complex scenarios are considered and tested on LOLA. The biped's newly gained abilities are compared to state of the art described in Chapter 1. Disadvantages of the current design and control schemes and differences to robots with torque-controlled joints are discussed. Based on the considered experiments, conclusions on the research topic and outlooks on future work are given in Chapter 8.

Chapter 8

Closure

This chapter summarizes the work presented in this thesis. Furthermore, conclusions on the gained knowledge are given and recommendations for future work are described briefly.

8.1 Contributions of this Thesis

This thesis's contributions are three-folded: First, hardware-near software and a tactile sensor/foot sole design for improved perception of the ground surface are presented. Second, control schemes for balancing biped robots on uneven terrain using foot and hand contacts are presented and experimentally evaluated. Third, software modules and tools to stabilize bipeds, generally control robots, and test robotic systems with a machine-learning approach are proposed. The contributions to these three parts are listed in the following.

Hardware-Related

- A high-performance hardware-layer implementation with particular emphasis on hard real-time is presented. A new concept for the abstraction of bus communication is proposed and published open-source.
- Several designs for a general-purpose low-cost tactile sensor are proposed and compared experimentally. The accuracy of these flexible and easy-to-replicate designs is analyzed.
- A design outlook for integrating tactile sensing into the foot sole of a humanoid robot is given. The design targets the instantaneous measurement of contact surfaces in partial contact scenarios.

Control Schemes The proposed control schemes rely on simple dynamical and geometric (contact) models. Some of the required plants are identified via onboard system identification. Others require basic data, for example, the humanoid's mass and foot design. All control policies are experimentally evaluated on LOLA.

- An improved inverse-kinematics scheme for task-space switching is presented.
- Reactive control strategies for early- and late contacts are proposed for the compensation of contact timing deviations. These strategies find application for foot and hand contact control.
- Methods to compensate parasitic effects caused by structural dynamics are compared for inclination control schemes.

- A QP formulation for the wrench distribution to multiple contact locations is compared with an existing heuristic.
- A control concept to overcome vertical ground-height changes is proposed. The method induces vertical CoM-height changes from the ground-reaction forces.
- A contact force control scheme with an explicit contact model is proposed. The control scheme's robustness to unknown ground properties is analyzed theoretically and with support from experimentally identified control plants.
- The use of sensor-based foothold surface data in the contact model of the ground-reaction force controllers is proposed. The approach improves the performance in partial foothold scenarios. It is shown that a tactile foot sole can increase the potential of this method.
- An extension of the ground-reaction force controller with adaptive control law is proposed. The algorithm reacts to previously unknown mechanical ground properties, for example, softer ground.
- The developed techniques for ground-reaction force control are ported to hand contact controllers, enabling multi-contact balancing and locomotion with dynamic contact transitions.

Software Architecture and Ecosystem

- The software architecture for the stabilization and control of LOLA is presented as part of a greater software ecosystem. Further enhancements to the simulation environment and development toolchain are presented. Special emphasis is put on hard real-time capability.
- Several control-related software modules used for LOLA are published open-source as part of the introduced broccoli library. The software is free to use and finds applications in other TUM robotics projects.
- A novel machine-learning-based system testing tool is presented. The approach uses noise-related features for the detection of failures from simulation data.

The developed control algorithms significantly improve the robustness of LOLA to undetected uneven and soft terrain. Note that no vision-based data or prior knowledge is used in the experiments, i.e., the robot is technically blind. In the experiments, LOLA overcomes inclined wooden terrain, loose wooden boards, ground-height changes of up to 9 cm, and a mixture of soft and hard obstacles at a walking speed of 0.5 m/s. The terrain's difficulty and the walking speed are significantly higher than reported in state of the art. With additional hand-contacts, LOLA can further overcome footsteps that may roll away using fast contact transitions. Related experiments have not been reported in the literature to the author's best knowledge.

8.2 Conclusions

The hypothesis stated in Chapter 1 is supported by the experimental results: Although the presented control schemes do not use the full dynamics models, the robot performs exceptionally well on uncertain terrain. Instead of using a complex model that needs to be accurate, the approaches deal with uncertainty by taking only a few things as well-known and adapting to deviations when appropriate. Of course, full model-based control of bipeds can further improve the performance. However, dealing with uncertainty in the model and the terrain typically becomes more complex for such approaches. In the following, further conclusions drawn from the experimental results are presented.

Hardware Performance Looking at state of the art on biped robots (see Section 1.1), one may notice that all high-performance robots are owned and developed by companies. This author believes that this is mainly due to insufficient hardware (integration) on research platforms. It is particularly striking that publications on hardware or hardware-related integration are not considered valuable to the community. Still, control methods based on full multi-body dynamics are often validated by slow quasi-static motions of a biped, which do not require considering dynamics at all.

The results shown in this thesis and the experience gained from the control schemes' design indicate the high importance of low-level control and integration. Furthermore, control schemes should live as close to the hardware as possible, adding complexity only where necessary. Indeed, the control schemes must be based on sound theoretical approaches — good integration alone is just as unhelpful. Still, the validation and fair comparison of methods require high-performance underlying control layers and hardware.

Low-Cost Tactile Sensors Experimental results with the proposed flexible, low-cost tactile sensor show the suitability of such designs to detect contact (locations) or contact surfaces, see Section 2.3. However, the attained pressure readings are not particularly accurate. A remaining problem for a high density of taxels is the connection of sensor wires to a suitable readout electronics. Especially for low taxel density, the proposed design can be replicated quickly and with low effort.

Foot Design The primary reason for the failure of experiments on uneven terrain with LOLA is related to the foot design, see Section 7.5. First, the separation in four separate pads is not ideal for small obstacles or inclined terrain. Second, the flat foot pads can get caught under protruding edges or loose objects. These results indicate the necessity of a foot redesign. The presented tactile foot design is a basic prototype for a new development.

Wrench Distribution A comparison of a heuristic with a QP-based solution shows surprisingly small differences for walking on flat ground, see Section 4.3.5. Other parts of the general control approach (contact force controllers, balance controllers) seem to have a higher impact on the overall robot performance. Moreover, tuning of the cost function parameters can be tedious due to the missing physical relation and the dependency on underlying control loops in the hierarchy. Nevertheless, the wrench distribution approach is essential for multi-contact locomotion to generate contact-consistent desired wrenches for all end effectors.

Vertical CoM Acceleration Experiments with unexpected ground-height changes show significant improvements in robustness when the CoM is additionally accelerated vertically based on the ground-reaction force tracking error, see Section 4.5.4. On the one hand, the acceleration of masses dynamically generates contact forces when required. On the other hand, the actual height of the CoM is adapted automatically. In contrast to horizontal CoM accelerations, a modification of the vertical contact forces and the CoM height are independently from the BoS and the horizontal dynamics.

The control scheme resembles a virtual spring-damper pair and has relations to biomechanical models. It may further be used to study energy-exchange mechanisms of humans[1] when walking over unexpected obstacles.

Contact Force Control: Robustness to Uncertainty This thesis's theoretical considerations and experiments prove the proposed contact force control scheme's high robustness to uncertain ground properties, see Section 5.2.4. The key to these methods' robust stability lies in a bounded stiffness of the control plant, which consists of the robot's structure, the foot sole, and the actual ground. Especially the inherent compliance of the foot sole enables robust stability of the control loops on soft and hard ground, loose and moving wooden parts. Thus, the design of the compliant contact elements is essential for the system's performance.

Contact Force Control: Adaptation to Uncertainty On soft terrain, or for partial footholds, the robust contact force controllers' stability margins increase further at the cost of the closed-loop crossover frequency.

This thesis's results show that the measured contact surface can be considered in an explicit contact model to reduce these effects for partial footholds, see Section 5.3. Already coarse surface measurements lead to better force tracking on uneven terrain with initial partial contacts of the feet. Simulations have shown an even higher performance when high-resolution tactile foot soles are used.

Experiments with an adaptive control approach show similar achievements concerning the uncertainty of unexpectedly soft ground, see Section 5.4. This approach's critical part is a reference model, which must encode the expected force-tracking behavior on hard ground. A deviation of the closed-loop from the reference model's dynamics leads to an adaptation of the control gains and avoids reducing the crossover frequency on soft ground. The method enables LOLA to walk on mixed soft and hard obstacles at walking speeds significantly higher than reported in the literature.

While the general robustness tolerates uncertainty in the ground properties, these additional approaches can compensate the uncertainty to some extent to improve the tradeoff between stability and performance.

The Stiffness of Contact Elements The compliance of the elements making contact with the environment (foot soles, hand surfaces) is important for the contact force controllers' robustness. Moreover, an optimum for these elements' stiffness exists, which depends on the actuator's limitation and the quality of the measured contact forces, see Section 5.2.8. A careful choice of these materials reduces force peaks during impacts while maintaining a high crossover frequency of the force-control loops.

Dynamic Contact Transitions The proposed contact-force control concepts are robust to dynamic and unexpected contact transitions. Experiments on uneven terrain and in multi-contact scenarios show the effectiveness of the event-based continuous switching between position- and force control. Furthermore, contact-timing imperfections are compensated by the implemented early- and late-contact reflexes.

Structural Resonances This work deals with resonances in the control loop plants caused by the flexibility of the robot's structure or joints. It is shown that filter-based loop shaping

[1] When walking over an unexpected obstacle, one strategy of humans is to trade in walking speed for CoM height to keep the CoM height over ground constant.

effectively reduces structural dynamics effects and increases the closed-loop crossover frequency, see Section 4.2.6. Interestingly, this approach is sufficiently robust to the variation of eigenfrequencies for different configurations of the robot. In contrast, time-domain approaches can not specifically target the structural resonance frequencies causing a general reduction of the control bandwidth.

Nevertheless, only the amplification of measured structural resonances via the control loops is avoided with these approaches. The results show that the corresponding bodies still vibrate, affecting the total system's performance, see Section 4.5.4. The amount of damping inserted via the joint-position-, force-, and balance control loops, is limited and the oscillations must be observable at the sensor locations. This leads to the conclusion that the robot's structure itself should be significantly damped for best performance. The results indicate that this may be even more important than a high stiffness.

Position-Control on Joint Level From a comparison of state of the art and the results shown in this thesis, one can conclude that currently, bipeds with torque-controlled joints do not show a systematic advantage to robots with position-controlled joints. Concerning balancing (in multi-contact scenarios), a stiff behavior of the robot's joints can be an advantage when dedicated methods for possible impacts are in place, see Section 7.4. From the discussion in Section 7.5 follows a strong dependency on specific hardware features. Therefore, a general statement is difficult, especially since biped walking requires precise control of both position and torques.

Software Quality and Ecosystem Two things can be concluded from work presented in this thesis concerning software development, see Chapter 6. First, software quality is essential for long-term research projects, and higher initial efforts usually pay off quickly. Second, common robotics frameworks like ROS are not suited for all robotics applications, in particular, due to missing hard real-time capability. The experiences with LOLA show that heterogeneous ecosystems with loosely-coupled modules are a lot more flexible than complete solutions, which probably will not fit all problems. In this context, small and independent research libraries (like *broccoli*) can facilitate higher flexibility in the development while still being able to share code because these can easily be used in combination with other frameworks.

Noise or Vibrations are a viable source for failure detection The early detection of software errors is essential for the effective development of new approaches for biped robots. Experiments are usually time-consuming and can put the expensive hardware at risk. The results in Section 6.5 show that a signal related to the virtual noise of the robot contains clues on failure symptoms. Due to this effective compression of dimensionality, applied machine-learning techniques are computationally efficient. An advantage to heuristics is a low effort required to add new software failures or train new expected behavior.

8.3 Recommendations for Future Work

Minor recommendations for improvements are given in the sections of the respective methods. In the following, future research topics on a broader scope are recommended. This includes both topics more specifically targeted on LOLA and topics generally considered relevant in biped robotics.

Next-Generation Robot Structures The structural parts for biped robots must withstand high forces during walking or when hitting an obstacle unexpectedly. At the same time, lightweight designs are required to enable fast movements and higher walking speeds. A certain compliance of the robot's links is not critical as long as the structure is sufficiently damped. Future work in this direction could focus on how to specifically introduce damping into the structural parts and joints. Possible candidates are 3D-printing or compound materials. The research could be inspired from the human physiology where the load-bearing bones are surrounded by soft tissue.

Full Centroidal Dynamics It seems particularly interesting to compare the hierarchical control approach for LOLA, which already has some notion on the (vertical) centroidal dynamics, with whole-body control frameworks like [166]. This could help to extract other simple control policies that arise from the consideration of centroidal dynamics. This author expects that a similarly constrained whole-body control approach leads to comparable results. The application on LOLA requires additional work due to the missing possibilities for joint-level torque control [94].

Global Stabilization Schemes The presented stabilization schemes are local in time and do not change footstep timing or locations. The methods benefit from the relatively large foot dimensions for LOLA. Consequently, the next step is implementing a global, predictive step recovery method on top of these control schemes. However, for accurate predictions of the biped's dynamics and resulting next footstep locations, large feet are disadvantageous. With large feet, an error in the robot's estimated state leads to a larger difference of the initial contact point of a foot, i.e., the robot ideally should be equipped with point-feet. A possible solution to this tradeoff between contact-force tracking performance and precise prediction of the dynamics may be a new foot design. It is further recommended to consider the conclusions on known foot-design issues described above.

Extended Ground-Profile Sensors State-of-the-art vision sensors usually have a pretty narrow field of view and limited resolution in the centimeter range. In particular, a head camera can not acquire data on the terrain directly in front of the robot. Additional sensors at the feet or legs of a biped could be used to detect the ground surface relevant for the next footstep. These concepts may be especially beneficial to predictive stabilization concepts, which need prior knowledge on the terrain below the next footstep to predict the dynamics accurately.

Hybrid Joint Control The control of biped robots requires accurate tracking of the joint positions and the contact forces. A possible direction for future work is integrating hybrid control loops at joint-level or in the fast 4 kHz loop of the communication bus. Currently, the switching between force-controlled and position-controlled directions is done in task-space. Shifting this hybrid scheme closer to the hardware can potentially increase performance. Although additional joint-torque sensors are beneficial, a hybrid implementation on LOLA can also use motor current measurements on back-drivable joints (for example, the knee) to achieve higher-bandwidth torque control. A possible strategy could be to switch the knees to current control before touchdown and switch back to position-control when the swing phase begins. A particular challenge is the consistency of the desired position- and torque trajectories[2] and the smooth switching between different control schemes.

[2] The control objectives of the individual joint controllers, which can be in position- or force control mode, must not conflict with each other or the global force-/motion targets.

Falling Strategies All control strategies for LOLA and most other bipeds are designed to avoid falling by all means. However, also humans do fall from time to time or on extremely complex terrain. Advances in this field are closely related to safety concepts. Are there alternatives to shutting off the power supply when things go wrong? Which strategies can be used to reduce impact and return to a safe standing pose afterwards? The relevance of falling strategies is not only justified by an increased autonomy in operation but also in the possibility to automated test and recovery. The later is a beneficial prerequisite for the fully autonomous acquisition of data sets in the context of data-driven optimization/machine-learning.

Automated Test Quality Evaluation A possible extension of the work started with *NoisyTest* may be the fully automatic evaluation of the quality of a hardware experiment. Instead of learning to detect specific failures in the data, the quality of an experiment may be linked with its measurement data. Based on training data, the system over time could automatically judge on the performance of an experiment. The automatic evaluation is a key part for the deployment of automatic, iterative optimizations run on real hardware (see below).

Solving Parameter Complexity Numerous parameters define the walking gait of LOLA, i.e., the step timing, the exact shape of the foot trajectories, the ZMP trajectory, and more. Furthermore, the stabilization module contains additional parameters and gains, and some parameter sets work better for specific scenarios than others.

An exciting direction for future work is how to determine these parameters and adapt them to the current situation of the biped. One possibility is a heuristic that may, for example, use the robot's state and vision-based semantic information of the scene to select or activate different gait modes and control schemes. This may be implemented in the form of a supervising software module that coordinates all planning and control efforts and considers correlations between the modules' methods.

As a further extension, the optimal parameters and their relation to the robot's state may be identified using unsupervised learning on the hardware. This approach requires the automated evaluation of an experiment's quality described above and is further supported by falling strategies. From many automated tests on different terrain, the software learns reasonable parameter choices, which makes parameter tuning superfluous. As an alternative to a learned reward function (see above), reference models may be used to define the biped robot system's desired behavior. Given that the reference-model state gradients depending on the parameters are known (these can be evaluated in simulation), optimal parameters may be calculated online when the actual state differs from the reference model.

Appendix A

Notation and Fundamentals

A.1 Frames of Reference

A vector quantity, like the position of a point p, $\boldsymbol{r}_p$, may be written in different FoRs. The corresponding FoR is denoted by the left hand index of the quantity. For example, ${}_j\boldsymbol{r}_p$ describes the vector $\boldsymbol{r}_p$ measured in the j frame of reference. ${}_i\boldsymbol{r}_{k,p} = {}_i\boldsymbol{r}_p - {}_i\boldsymbol{r}_k$ denotes the vector from point k to point p written in frame i.

A.2 Rotations in 3D

This section summarizes the basics on rotations in 3D and quaternions. The summary does not claim to be complete. Relationships and operators are explained only as far as necessary; some definitions are only given implicitly. The following content is based on the corresponding literature [51, 56, 97, 132, 200, 251].

Rotation Matrices A rotation matrix ${}_j\boldsymbol{A}_i \in \mathbb{R}^3$ defines a frame transformation from i to j using the notation:

$$
\begin{aligned}
{}_j\boldsymbol{r}_p &= {}_j\boldsymbol{A}_i\,{}_i\boldsymbol{r}_p && \text{(A.1)}\\
{}_k\boldsymbol{A}_i &= {}_k\boldsymbol{A}_j\,{}_j\boldsymbol{A}_i. && \text{(A.2)}
\end{aligned}
$$

By developing ${}_j\dot{\boldsymbol{A}}_i$ via finite differences

$$
{}_j\dot{\boldsymbol{A}}_i = \lim_{\Delta t \to 0} \frac{{}_j\boldsymbol{A}_i(t+\Delta t) - {}_j\boldsymbol{A}_i(t)}{\Delta t} = \lim_{\Delta t \to 0} \frac{{}_j\boldsymbol{A}_i(t)\,{}_i\Delta\boldsymbol{A}_i - {}_j\boldsymbol{A}_i(t)}{\Delta t}, \tag{A.3}
$$

using an infinitesimal small rotation ${}_i\Delta\boldsymbol{A}_i$, the relation to the instantaneous angular velocity between the frames i and j ${}_i\boldsymbol{\omega}_{j,i}$ is found. The rotation matrix ${}_i\Delta\boldsymbol{A}_i$ is constructed from the superposition of infinitesimal small *frame rotations* around the unit vectors of frame i using $\cos(x) \approx 1$, $\sin(x) \approx x$, and neglecting higher-order terms:

$$
\boldsymbol{R} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -{}_i\Delta\alpha \\ 0 & {}_i\Delta\alpha & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & {}_i\Delta\beta \\ 0 & 1 & 0 \\ -{}_i\Delta\beta & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -{}_i\Delta\gamma & 0 \\ {}_i\Delta\gamma & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{A.4}
$$

$$
{}_i\Delta\boldsymbol{A}_i = \begin{pmatrix} 1 & -{}_i\Delta\gamma & {}_i\Delta\beta \\ {}_i\Delta\gamma & 1 & -{}_i\Delta\alpha \\ -{}_i\Delta\beta & {}_i\Delta\alpha & 1 \end{pmatrix} \approx \boldsymbol{R}. \tag{A.5}
$$

Inserting Equation (A.5) into Equation (A.3) then yields

$$ {}_j\dot{\boldsymbol{A}}_i = {}_j\boldsymbol{A}_i\, {}_i\tilde{\boldsymbol{\omega}}_{j,i}, \tag{A.6}$$

with the tilde operator being defined to

$$\tilde{\boldsymbol{\omega}} = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} \forall \boldsymbol{\omega} \in \mathbb{R}^3. \tag{A.7}$$

Quaternions There are several notations for quaternions. To clarify the use of quaternions in this thesis, the notation and some basic relations are introduced in the following. The original Hamiltonian (right-handed) definition [97] is used in this thesis and defines a quaternion to

$$S = s_w + s_x i + s_y j + s_z k \in \mathbb{H}, \text{ with} \tag{A.8}$$

$$i^2 = j^2 = k^2 = ijk = -1, \tag{A.9}$$

and its vectorial form

$$\boldsymbol{s} = \begin{bmatrix} s_w \\ \boldsymbol{s}_v \end{bmatrix} = \begin{bmatrix} s_w \\ s_x \\ s_y \\ s_z \end{bmatrix} \in \mathbb{R}^4. \tag{A.10}$$

The Hamiltonian product of two quaternions is denoted with ⊗:

$$\boldsymbol{s} \otimes \boldsymbol{t} = \begin{bmatrix} s_w t_w - \boldsymbol{s}_v^T \boldsymbol{t}_v \\ s_w \boldsymbol{t}_v + t_w \boldsymbol{s}_v + \boldsymbol{s}_v \times \boldsymbol{t}_v \end{bmatrix} \neq \boldsymbol{t} \otimes \boldsymbol{s}. \tag{A.11}$$

The definition of an inverse quaternion is given by

$$\boldsymbol{s} \otimes \boldsymbol{s}^{-1} = \boldsymbol{s}^{-1} \otimes \boldsymbol{s} = \boldsymbol{s}_1 = \begin{bmatrix} 1 \\ \boldsymbol{0} \end{bmatrix}, \text{ with} \tag{A.12}$$

$$\boldsymbol{s}^{-1} = \frac{\boldsymbol{s}^*}{||\boldsymbol{s}||^2} = \frac{\boldsymbol{s}^*}{s_w^2 + ||\boldsymbol{s}_v||^2}, \tag{A.13}$$

and the conjugate of a quaternion

$$\boldsymbol{s}^* = \begin{bmatrix} s_w \\ -\boldsymbol{s}_v \end{bmatrix}. \tag{A.14}$$

Describing Rotations with Quaternions To describe rotations in $\mathbb{R}^3$ with the help of unit quaternions ($||\boldsymbol{s}|| = 1$) on the unit 3-sphere $\mathbb{S}^3$, a notation similar to the one for rotation matrices is used:

$$\begin{bmatrix} 0 \\ {}_j\boldsymbol{r}_p \end{bmatrix} = {}_j\boldsymbol{s}_i \otimes \begin{bmatrix} 0 \\ {}_i\boldsymbol{r}_p \end{bmatrix} \otimes {}_j\boldsymbol{s}_i^* \tag{A.15}$$

$${}_k\boldsymbol{s}_i = {}_k\boldsymbol{s}_j \otimes {}_j\boldsymbol{s}_i \in \mathbb{S}^3. \tag{A.16}$$

The quaternion ${}_j\boldsymbol{s}_i$ encodes the information of a frame transformation from i to j and is a unit quaternion with $||{}_j\boldsymbol{s}_i|| = 1$. From considerations on Equation (A.15) and Equation (A.11) with vectors describing small rotations follows that a small rotation of the *frame i* with angles ${}_i\Delta\alpha$, ${}_i\Delta\beta$, ${}_i\Delta\gamma$ is described by

$$ {}_i\Delta\boldsymbol{s}_i = \begin{bmatrix} 1 & \frac{1}{2}{}_i\Delta\alpha & \frac{1}{2}{}_i\Delta\beta & \frac{1}{2}{}_i\Delta\gamma \end{bmatrix}^T . \tag{A.17} $$

Similar to the rotation matrices, and using Equation (A.17), the quaternion time derivative is developed using finite differences:

$$ \begin{aligned} {}_j\dot{\boldsymbol{s}}_i &= \lim_{\Delta t \to 0} \frac{{}_j\boldsymbol{s}_i(t+\Delta t) - {}_j\boldsymbol{s}_i(t)}{\Delta t} = \lim_{\Delta t \to 0} \frac{{}_j\boldsymbol{s}_i(t) \otimes {}_i\Delta\boldsymbol{s}_i - {}_j\boldsymbol{s}_i(t)}{\Delta t} \\ &= \lim_{\Delta t \to 0} \frac{{}_j\boldsymbol{s}_i(t) \otimes \left(\begin{bmatrix} 1 & \frac{1}{2}{}_i\Delta\alpha & \frac{1}{2}{}_i\Delta\beta & \frac{1}{2}{}_i\Delta\gamma \end{bmatrix}^T - \boldsymbol{s}_1 \right)}{\Delta t} \\ &= \lim_{\Delta t \to 0} \frac{{}_j\boldsymbol{s}_i(t) \otimes \begin{bmatrix} 0 & \frac{1}{2}{}_i\Delta\alpha & \frac{1}{2}{}_i\Delta\beta & \frac{1}{2}{}_i\Delta\gamma \end{bmatrix}^T}{\Delta t} \\ &= {}_j\boldsymbol{s}_i \otimes \begin{bmatrix} 0 \\ \frac{1}{2}{}_i\boldsymbol{\omega}_{j,i} \end{bmatrix} . \end{aligned} \tag{A.18} $$

Using the quaternion transform Equation (A.15), Equation (A.18) can be rewritten with the angular velocity in the j frame:

$$ {}_j\dot{\boldsymbol{s}}_i = \begin{bmatrix} 0 \\ \frac{1}{2}{}_j\boldsymbol{\omega}_{j,i} \end{bmatrix} \otimes {}_j\boldsymbol{s}_i . \tag{A.19} $$

Relation between Quaternions and Rotation Matrices The transformations in Equations (A.1) and (A.15) show the equivalence of rotation matrices and quaternions for the description of rotations. Both representations may be converted to each other. The following operator is defined to describe the conversion of a rotation-describing unit quaternion to a corresponding transformation matrix:

$$ \begin{aligned} \mathrm{rotMat}\colon \mathbb{S}^3 &\to \mathbb{R}^{3\times 3} \\ {}_j\boldsymbol{s}_i &\mapsto {}_j\boldsymbol{A}_i . \end{aligned} \tag{A.20} $$

While the composition of rotations is usually done via quaternions, the actual transformation of entities (e.g. a position vector) is done via the corresponding rotation matrix. This is due to the fact that the multiplication of rotation matrices is less efficient than the multiplication of quaternions, but transformation matrices are efficient when performing the actual transformation.

Rotation Vectors A rotation vector $\boldsymbol{\vartheta} \in \mathbb{R}$ is an axis-angle representation of rotations in 3D, where the angle is encoded in the length of the vector. From a rotation axis ${}_j\boldsymbol{u}$ and angle $\theta_{j,i}$ for a frame rotation from i to j, the corresponding rotation vector is given by ${}_j\boldsymbol{\vartheta}_{j,i} = \theta_{j,i}\,{}_j\boldsymbol{u}$. From the solutions of the differential equations (A.18), (A.19) for constant angular velocity

$$ {}_j\boldsymbol{s}_i(t) = {}_j\boldsymbol{s}_i(0) \otimes \exp\left(\begin{bmatrix} 0 \\ \frac{1}{2}{}_i\boldsymbol{\omega}_{j,i}\, t \end{bmatrix} \right) = \exp\left(\begin{bmatrix} 0 \\ \frac{1}{2}{}_j\boldsymbol{\omega}_{j,i}\, t \end{bmatrix} \right) \otimes {}_j\boldsymbol{s}_i(0) \tag{A.21} $$

also follows a direct relation between $\frac{1}{2}{}_j\boldsymbol{\omega}_{j,i}\, t = \frac{1}{2}{}_j\boldsymbol{\vartheta}_{j,i}$ and a corresponding unit quaternion for ${}_j\boldsymbol{\omega}_{j,i} = \text{const.}$:

$$ {}_j\boldsymbol{s}_i = \exp\left(\begin{bmatrix} 0 \\ \frac{1}{2}{}_j\boldsymbol{\vartheta}_{j,i} \end{bmatrix} \right) \qquad \text{and} \qquad \begin{bmatrix} 0 \\ \frac{1}{2}{}_j\boldsymbol{\vartheta}_{j,i} \end{bmatrix} = \ln({}_j\boldsymbol{s}_i), \tag{A.22} $$

where the exponential and natural logarithm are defined by their Taylor series. The relation between a rotation vector (angle-axis representation) and quaternion parameters can also be derived via the Euler-Rodrigues formula [200, pp.42].

The following operators are additionally defined to describe the direct mapping between a rotation vector in $\mathbb{R}^3$ and a unit quaternion in $\mathbb{S}^3$:

$$\begin{aligned} \text{quat}\colon \mathbb{R}^3 &\to \mathbb{S}^3 & \qquad \text{rotVec}\colon \mathbb{S}^3 &\to \mathbb{R}^3 \\ \boldsymbol{\vartheta} &\mapsto \exp\left(\begin{bmatrix} 0 \\ \frac{1}{2}\boldsymbol{\vartheta} \end{bmatrix}\right) = \text{quat}(\boldsymbol{\vartheta}) = \boldsymbol{s} & \qquad \boldsymbol{s} &\mapsto \text{quat}^{-1}(\boldsymbol{s}) = \boldsymbol{\vartheta}. \end{aligned} \tag{A.23}$$

Rotation vectors are a minimal but complete description of rotations in 3D. The additional operator wrap() takes a rotation vector $\boldsymbol{\vartheta}$ and returns an equivalent representation $\boldsymbol{\vartheta}'$, where the norm $||\boldsymbol{\vartheta}'||$ is wrapped to $||\boldsymbol{\vartheta}'|| \in [-\pi, \pi)$. When the rotation vector describes a difference between two orientations, this ensures always the smallest angle is returned.

From instantaneous angular velocities to orientation modifications For the control of the foot orientations, it is necessary to synthesize a rotation from the integration of instantaneous angular velocities. In this thesis, the following approximative relation is used to change the orientation of Lola's feet. For a modification velocity ${}_i\Delta\boldsymbol{\omega}(t)$ in the i frame, the approximative rotation vector is obtained via integration:

$${}_i\boldsymbol{\vartheta}_{i,i'} \approx \int_0^t {}_i\Delta\boldsymbol{\omega}(\tau)\,\mathrm{d}\tau. \tag{A.24}$$

Note that the instantaneous angular velocity ${}_i\Delta\boldsymbol{\omega}(\tau)$ is always interpreted in the i frame, not the i' frame. This approximation assumes the angle between ${}_i\Delta\boldsymbol{\omega}(\tau)$ and ${}_i\boldsymbol{\vartheta}_{i,i'}$ is small, which is true when the axis of the angular velocity only changes slowly. The exact analytical relation between ${}_i\dot{\boldsymbol{\vartheta}}_{i,i'}$ and ${}_i\Delta\boldsymbol{\omega}(\tau)$ is described in [273] and is singular in certain configurations.

An alternative solution that uses a piecewise-constant angular velocity in a time-discrete, iterative scheme is based on Equation (A.21):

$${}_i\boldsymbol{s}_{i,i'}[n+1] = \text{quat}({}_i\Delta\boldsymbol{\omega}[n]\,\Delta t) \otimes {}_i\boldsymbol{s}_{i,i'}[n]. \tag{A.25}$$

This algorithm calculates a modification quaternion from a small rotation vector and the quaternion of the last time-step n. Simulations with Lola have shown neglectable differences between the two solutions from Equations (A.24) and (A.25) even when walking on uneven terrain with relatively large foot inclinations. A further alternative is the numerical integration of the quaternion parameters using Equation (A.19). However, this requires the normalization of resulting quaternion parameters after each integration step. Due to simplicity and the reuse of code for the integrators, the approximative integration Equation (A.24) is used.

The orientation modification may be applied relative to a time-dependent frame i, see Appendix A.4 for an example. Let ${}_j\boldsymbol{s}_i(t)$ be a unit quaternion describing the orientation of frame i relative to j. The orientation modification described in the i frame may then also be combined to a total orientation:

$${}_j\boldsymbol{s}_{i'}(t) = {}_j\boldsymbol{s}_i(t) \otimes \text{quat}({}_i\boldsymbol{\vartheta}_{i,i'}). \tag{A.26}$$

Note that the order of rotations must be respected, i.e., for modifications in the j frame, the order of multiplication flips.

A.3 Digital Filter Implementations

In the following, the implementations of several digital filters are described. These filters follow from their continuous-time counterpart using the bilinear (tustin) transformation method. The filter frequencies are not adapted (prewarped), i.e., the characteristic frequencies start to shift when they come close to the nyquist frequency $\frac{1}{2\Delta t_{\text{cont}}}$.

First-Order Low-Pass Filter The lpf1$(T, \boldsymbol{u})$ with the time constant T, and input vector $\boldsymbol{u}$ is an element-wise digital implementation of the continuous-time transfer function

$$P_1(s) = \frac{1}{Ts+1} \tag{A.27}$$

with the time-discrete representation

$$P_1(z) = \frac{\Delta t_{\text{cont}}(1+z^{-1})}{(\Delta t_{\text{cont}}+2T)+(\Delta t_{\text{cont}}-2T)z^{-1}}. \tag{A.28}$$

Second-Order Low-Pass Filter The lpf2$(T, d, \boldsymbol{u})$ with the time constant T, the damping factor d, and input vector $\boldsymbol{u}$ is an element-wise digital implementation of the continuous-time transfer function

$$P_2(s) = \frac{1}{T^2s^2+2dTs+1}, \tag{A.29}$$

with the time-discrete representation

$$P_2(z) = \frac{\Delta t_{\text{cont}}^2(1+2z^{-1}+z^{-2})}{4T^2+4dT\Delta t_{\text{cont}}+\Delta t_{\text{cont}}^2+(2\Delta t_{\text{cont}}^2-8T^2)z^{-1}+(4T^2-4dT\Delta t_{\text{cont}}+\Delta t_{\text{cont}}^2)z^{-2}}. \tag{A.30}$$

Discrete-Time Derivative Filter The ddt$(T, \boldsymbol{u})$ filter with the time constant T, and input vector $\boldsymbol{u}$ is an element-wise numeric derivative of the first-order time-discrete low-pass filter described above with the time-discrete representation

$$P_{\text{dt}}(z) = \frac{(1-z^{-2})}{(\Delta t_{\text{cont}}+2T)+(\Delta t_{\text{cont}}-2T)z^{-1}}. \tag{A.31}$$

Discrete-Time Integrator The int$(\boldsymbol{u})$ integrator with the input vector $\boldsymbol{u}$ is an element-wise discrete-time forward Euler integrator defined by the transfer function

$$P_{\text{int}}(z) = \frac{\Delta t_{\text{cont}}}{1-z^{-1}}. \tag{A.32}$$

Notch Filter The notch$(T, Q, d, \boldsymbol{u})$ with the time constant T, the sharpness factor Q, the damping factor d, and input vector $\boldsymbol{u}$ is an element-wise digital implementation of the continuous-time transfer function

$$P_n(s) = \frac{T^2s^2+2dTs+1}{T^2s^2+\frac{T}{Q}s+1}, \tag{A.33}$$

with the time-discrete representation

$$P_n(z) = Q\frac{(4T^2+4dT\Delta t_{\text{cont}}+\Delta t_{\text{cont}}^2)+(2\Delta t_{\text{cont}}^2-8T^2)z^{-1}+(4T^2-4dT\Delta t_{\text{cont}}+\Delta t_{\text{cont}}^2)z^{-2}}{4QT^2+2T\Delta t_{\text{cont}}+Q\Delta t_{\text{cont}}^2+(2Q\Delta t_{\text{cont}}^2-8QT^2)z^{-1}+(4QT^2-2T\Delta t_{\text{cont}}+Q\Delta t_{\text{cont}}^2)z^{-2}}. \tag{A.34}$$

A.4 Task-Space Operators

The following operators are required for the modification and comparison of task-space vectors. They are used in several parts of this thesis; see also Section 3.4.3. The modification of a task-space vector is not straightforward due to the included rotation vectors. The considerations from Appendix A.2 on the modification of orientations are used in the following.

For the right foot, the operator for a modification of the foot trajectory in W is given by

$$ {}_{W}\mathrm{modify}_{\mathrm{RF}}\colon \mathbb{R}^{28} \times \mathbb{R}^{28} \times \mathbb{R}^{6} \times \mathbb{R}^{6} \longrightarrow \mathbb{R}^{28} \times \mathbb{R}^{28} $$

$$ (\boldsymbol{X}, \boldsymbol{V}, {}_{W}\Delta\boldsymbol{x}_{RF}, {}_{W}\Delta\boldsymbol{v}_{RF}) \mapsto \left(\begin{bmatrix} {}_{W}\boldsymbol{r}_{W,\mathrm{CoM}} \\ {}_{W}\boldsymbol{\vartheta}_{W,U} \\ {}_{W}\boldsymbol{r}_{W,\mathrm{RF}} + {}_{W}\Delta\boldsymbol{r}_{W,\mathrm{RF}} \\ \mathrm{rotVec}\big(\mathrm{quat}({}_{W}\Delta\boldsymbol{\vartheta}_{W,\mathrm{RF}}) \otimes \mathrm{quat}({}_{W}\boldsymbol{\vartheta}_{W,\mathrm{RF}})\big) \\ \varphi_{\mathrm{RT}} \\ {}_{W}\boldsymbol{r}_{W,\mathrm{LF}} \\ {}_{W}\boldsymbol{\vartheta}_{W,\mathrm{LF}} \\ \varphi_{\mathrm{LT}} \\ \varphi_{\mathrm{P}} \\ \varphi_{\mathrm{T}} \\ {}_{W}\boldsymbol{x}_{\mathrm{RH}} \\ {}_{W}\boldsymbol{x}_{\mathrm{LH}} \end{bmatrix}, \begin{bmatrix} {}_{W}\dot{\boldsymbol{r}}_{W,\mathrm{CoM}} \\ {}_{W}\boldsymbol{\omega}_{W,U} \\ {}_{W}\dot{\boldsymbol{r}}_{W,\mathrm{RF}} + {}_{W}\Delta\dot{\boldsymbol{r}}_{W,\mathrm{RF}} \\ {}_{W}\boldsymbol{\omega}_{W,\mathrm{RF}} + {}_{W}\Delta\boldsymbol{\omega}_{W,\mathrm{RF}} \\ \dot{\varphi}_{\mathrm{RT}} \\ {}_{W}\dot{\boldsymbol{r}}_{W,\mathrm{LF}} \\ {}_{W}\boldsymbol{\omega}_{W,\mathrm{LF}} \\ \dot{\varphi}_{\mathrm{LT}} \\ \dot{\varphi}_{\mathrm{P}} \\ \dot{\varphi}_{\mathrm{T}} \\ {}_{W}\boldsymbol{v}_{\mathrm{RH}} \\ {}_{W}\boldsymbol{v}_{\mathrm{LH}} \end{bmatrix} \right). \tag{A.35} $$

Correspondingly, the operator for modifying the right foot trajectory in the end-effector FoR is given by

$$ {}_{\mathrm{RF}}\mathrm{modify}_{\mathrm{RF}}\colon \mathbb{R}^{28} \times \mathbb{R}^{28} \times \mathbb{R}^{6} \times \mathbb{R}^{6} \longrightarrow \mathbb{R}^{28} \times \mathbb{R}^{28} $$

$$ (\boldsymbol{X}, \boldsymbol{V}, {}_{\mathrm{RF}}\Delta\boldsymbol{x}_{RF}, {}_{\mathrm{RF}}\Delta\boldsymbol{v}_{RF}) \mapsto \left(\begin{bmatrix} {}_{W}\boldsymbol{r}_{W,\mathrm{CoM}} \\ {}_{W}\boldsymbol{\vartheta}_{W,U} \\ {}_{W}\boldsymbol{r}_{W,\mathrm{RF}} + \mathrm{rotMat}(\mathrm{quat}({}_{W}\boldsymbol{\vartheta}_{W,\mathrm{RF}}))\,{}_{\mathrm{RF}}\Delta\boldsymbol{r}_{W,\mathrm{RF}} \\ \mathrm{rotVec}\big(\mathrm{quat}({}_{W}\boldsymbol{\vartheta}_{W,\mathrm{RF}}) \otimes \mathrm{quat}({}_{\mathrm{RF}}\Delta\boldsymbol{\vartheta}_{W,\mathrm{RF}})\big) \\ \varphi_{\mathrm{RT}} \\ {}_{W}\boldsymbol{r}_{W,\mathrm{LF}} \\ {}_{W}\boldsymbol{\vartheta}_{W,\mathrm{LF}} \\ \varphi_{\mathrm{LT}} \\ \varphi_{\mathrm{P}} \\ \varphi_{\mathrm{T}} \\ {}_{W}\boldsymbol{x}_{\mathrm{RH}} \\ {}_{W}\boldsymbol{x}_{\mathrm{LH}} \end{bmatrix}, \begin{bmatrix} {}_{W}\dot{\boldsymbol{r}}_{W,\mathrm{CoM}} \\ {}_{W}\boldsymbol{\omega}_{W,U} \\ {}_{W}\dot{\boldsymbol{r}}_{W,\mathrm{RF}} + \mathrm{rotMat}(\mathrm{quat}({}_{W}\boldsymbol{\vartheta}_{W,\mathrm{RF}}))\,{}_{\mathrm{RF}}\Delta\,\mathring{\boldsymbol{r}}_{W,\mathrm{RF}} \\ {}_{W}\boldsymbol{\omega}_{W,\mathrm{RF}} + \mathrm{rotMat}(\mathrm{quat}({}_{W}\boldsymbol{\vartheta}_{W,\mathrm{RF}}))\,{}_{\mathrm{RF}}\Delta\boldsymbol{\omega}_{W,\mathrm{RF}} \\ \dot{\varphi}_{\mathrm{RT}} \\ {}_{W}\dot{\boldsymbol{r}}_{W,\mathrm{LF}} \\ {}_{W}\boldsymbol{\omega}_{W,\mathrm{LF}} \\ \dot{\varphi}_{\mathrm{LT}} \\ \dot{\varphi}_{\mathrm{P}} \\ \dot{\varphi}_{\mathrm{T}} \\ {}_{W}\boldsymbol{v}_{\mathrm{RH}} \\ {}_{W}\boldsymbol{v}_{\mathrm{LH}} \end{bmatrix} \right). \tag{A.36} $$

The operators for the left foot are defined accordingly. The operator for modifying a the hand trajectory, exemplarily for the right hand, is given by

$$ {}_{\mathrm{RH}}\mathrm{modify}_{\mathrm{RH}}\colon \mathbb{R}^{28} \times \mathbb{R}^{28} \times \mathbb{R}^{3} \times \mathbb{R}^{3} \longrightarrow \mathbb{R}^{28} \times \mathbb{R}^{28} $$

$$ (\boldsymbol{X}, \boldsymbol{V}, {}_{\mathrm{RH}}\Delta\boldsymbol{x}_{RH}, {}_{\mathrm{RF}}\Delta\boldsymbol{v}_{RH}) \mapsto \left(\begin{bmatrix} {}_{W}\boldsymbol{r}_{W,\mathrm{CoM}} \\ {}_{W}\boldsymbol{\vartheta}_{W,U} \\ {}_{W}\boldsymbol{r}_{W,\mathrm{RF}} \\ {}_{W}\boldsymbol{\vartheta}_{W,\mathrm{RF}} \\ \varphi_{\mathrm{RT}} \\ {}_{W}\boldsymbol{r}_{W,\mathrm{LF}} \\ {}_{W}\boldsymbol{\vartheta}_{W,\mathrm{LF}} \\ \varphi_{\mathrm{LT}} \\ \varphi_{\mathrm{P}} \\ \varphi_{\mathrm{T}} \\ {}_{W}\boldsymbol{x}_{\mathrm{RH}} + {}_{W}\boldsymbol{A}_{\mathrm{RH}}\,{}_{\mathrm{RH}}\Delta\boldsymbol{x}_{RH} \\ {}_{W}\boldsymbol{x}_{\mathrm{LH}} \end{bmatrix}, \begin{bmatrix} {}_{W}\dot{\boldsymbol{r}}_{W,\mathrm{CoM}} \\ {}_{W}\boldsymbol{\omega}_{W,U} \\ {}_{W}\dot{\boldsymbol{r}}_{W,\mathrm{RF}} \\ {}_{W}\boldsymbol{\omega}_{W,\mathrm{RF}} \\ \dot{\varphi}_{\mathrm{RT}} \\ {}_{W}\dot{\boldsymbol{r}}_{W,\mathrm{LF}} \\ {}_{W}\boldsymbol{\omega}_{W,\mathrm{LF}} \\ \dot{\varphi}_{\mathrm{LT}} \\ \dot{\varphi}_{\mathrm{P}} \\ \dot{\varphi}_{\mathrm{T}} \\ {}_{W}\boldsymbol{v}_{\mathrm{RH}} + {}_{W}\boldsymbol{A}_{\mathrm{RH}}\,{}_{\mathrm{RH}}\Delta\boldsymbol{v}_{RH} \\ {}_{W}\boldsymbol{v}_{\mathrm{LH}} \end{bmatrix} \right). \tag{A.37} $$

IK algorithms require the calculation of a difference between two task-space vectors. For the hand task-space vectors this is the standard subtraction. For the base vector the difference is defined via

$$
\begin{aligned}
\mathrm{deltaX}_B \colon \mathbb{R}^{22} \times \mathbb{R}^{22} &\to \mathbb{R}^{22} \\
({}_W\boldsymbol{x}_{B,\mathrm{d}}, {}_W\boldsymbol{x}_B) &\mapsto \begin{bmatrix}
{}_W\boldsymbol{r}_{W,\mathrm{CoM,d}} - {}_W\boldsymbol{r}_{W,\mathrm{CoM}} \\
\mathrm{wrap}(\mathrm{rotVec}(\mathrm{quat}({}_W\boldsymbol{\vartheta}_{W,U,\mathrm{d}}) \otimes \mathrm{quat}({}_W\boldsymbol{\vartheta}_{W,U})^*)) \\
{}_W\boldsymbol{r}_{W,\mathrm{RF,d}} - {}_W\boldsymbol{r}_{W,\mathrm{RF}} \\
\mathrm{wrap}(\mathrm{rotVec}(\mathrm{quat}({}_W\boldsymbol{\vartheta}_{W,\mathrm{RF,d}}) \otimes \mathrm{quat}({}_W\boldsymbol{\vartheta}_{W,\mathrm{RF}})^*)) \\
\varphi_{\mathrm{RT,d}} - \varphi_{\mathrm{RT}} \\
{}_W\boldsymbol{r}_{W,\mathrm{LF,d}} - {}_W\boldsymbol{r}_{W,\mathrm{LF}} \\
\mathrm{wrap}(\mathrm{rotVec}(\mathrm{quat}({}_W\boldsymbol{\vartheta}_{W,\mathrm{LF,d}}) \otimes \mathrm{quat}({}_W\boldsymbol{\vartheta}_{W,\mathrm{LF}})^*)) \\
\varphi_{\mathrm{LT,d}} - \varphi_{\mathrm{LT}} \\
\varphi_{\mathrm{P,d}} - \varphi_{\mathrm{P}} \\
\varphi_{\mathrm{T,d}} - \varphi_{\mathrm{T}}
\end{bmatrix} .
\end{aligned}
\tag{A.38}
$$

A.5 Control Schemes

The following control schemes are repeatedly used in this thesis and are intended for velocity-level operation, i.e., their output and the control input $\boldsymbol{u}$ are velocities. The following functions are intended to make the control scheme equations in this thesis easier to read.

Both control schemes use a position control gain $G_x = 25\,\mathrm{s}^{-1}$. Due to the first-order structure of this controller, $\frac{5}{G_x} = 0.2\,\mathrm{s}$ is the time until 99 % of a static modification are compensated. The value is chosen such that modifications on the feet can safely be driven back to zero until a swing foot reaches its maximum height, see Figure 3.3.

Position Control The following scheme

$$
\begin{aligned}
\mathrm{posCtl} \colon \mathbb{R}^n \times \mathbb{R}^n &\to \mathbb{R}^n \\
(\boldsymbol{u}, \boldsymbol{x}) &\mapsto \boldsymbol{u} - G_x\,\boldsymbol{x}
\end{aligned}
\tag{A.39}
$$

returns the input velocity $\boldsymbol{u}$ with superposed position controller to drive a supplied modification $\boldsymbol{x}$ back to zero.

Hybrid Control Based on a blending factor $\beta \in [0,1]$, and a control input velocity $\boldsymbol{u}$, the control scheme

$$
\begin{aligned}
\mathrm{hybridCtl} \colon \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n &\to \mathbb{R}^n \\
(\beta, \boldsymbol{u}, \boldsymbol{x}) &\mapsto \beta\,\boldsymbol{u} - (1-\beta)\,G_x\,\boldsymbol{x}
\end{aligned}
\tag{A.40}
$$

either returns $\boldsymbol{u}$ $(\beta = 1)$ or returns a velocity to drive $\boldsymbol{x}$ back to zero $(\beta = 0)$. The scheme is used for time-depending hybrid control of the end effectors (force/position).

Appendix B

Joint Control Settings & Bus Topology

This appendix briefly summarizes the technical details of the joint actuators, the settings used in the decentralized control modules, and the topology of the EC bus. In Table B.1, the inverter settings used on the joint drives are shown. These settings are based on the manufacturer-specified maximums for windings and gearbox, see Table B.2. The motors are, in some cases, operated out of their specifications, due to better thermal properties of the drive modules compared to the manufacturer's reference design. The values were found based on experience and the measurement of the winding temperature, see [188].

The feedback encoder settings for the joint drive modules are summarized in Table B.3. Furthermore, the following settings are applied and saved on each joint controller:

- Single feedback, motor side, no gearbox (between encoder and motor), rotary load.
- Incremental encoder on socket 1: Quadrature encoder port B, no hall sensors.
- Absolute encoder on socket 2: EnDat 2.1 on port A with error bit mask.
- Socket 2 is auxiliary position: CA[79]=2
- General purpose settings according to Table B.3.

In Table B.4, the physical and logical connections between all EC slaves are described.

Joints	Motor Kit Type	Inverter Type	Continuous Current [A]	Peak Current [A]	Max. Motor Speed [RPM]	Pole Pairs
Pelvis Adduction Pelvis Rotation	PB K064050-8Y	G-WHI15/100EE	4.67	12.0$^{\triangle}$	8500	4
Hip Flexion	PB K375100-4D	G-GUT35/100EE	24.0	62.0	4800	6
Hip Adduction Hip Rotation	PB K064075-7Y	G-WHI15/100EE	5.35	16.48	8000	4
Knee Flexion	PB K375050-5Y	G-GUT35/100EE	13.5	32.0	5600	6
Ankle Flexion Ankle Adduction	PB K064050-8Y	G-WHI15/100EE	4.67	14.1	8500	4
Toe Flexion	PB K044025-JY	G-WHI5/100EE	1.06	4.2	8500	3
Arm Flexion	PB K064050-9D	G-WHI15/100EE	6.46	16.5$^{\triangle}$	8000	4
Arm Rotation	PB K064025-GD	G-WHI10/100EE	3.2	10.21	8000	4
Arm Adduction Elbow Flexion	PB K064025-GD	G-WHI15/100EE	3.2	10.21	8000	4
Head Pan	WI MSSI 022H-017C	G-WHI5/100EE	2.3	9.0	25500	3
Head Tilt	HD RSF-5A	G-WHI5/100EE	0.5	1.08	10000	4

Table B.1: Inverter settings for LOLA's joint motors. The settings are primarily chosen based on thermal considerations of the motor windings. In some cases, the overload torque of the gearbox (△) is the limiting factor.

Joints	Drive Type	Torque Motor Constant [Nm/Arms]	Nom. Winding Current [Arms]		Gearbox	Gear Ratio	Gearbox Torque [Nm]		
			Cont.	Peak			avg.	peak	overload
Pelvis Adduction Pelvis Rotation	D PB K064050-8Y	0.133	4.44	14.02	HFUC-20	100	49	82	147
Hip Flexion	A PB K375100-4D	0.119	19.66	62.18	CSG-32	50	140	281	497
Hip Adduction Hip Rotation	B & C PB K064075-7Y	0.159	5.22	16.48	HFUC-25	100	108	157	284
Knee Flexion	PB K375050-5Y	0.129	10.16	32.09	RGTF-12x5	≤ 72	-	-	-
Ankle Flexion Ankle Adduction	PB K064050-8Y	0.133	4.44	14.02	RGTF-8x5	≤ 70	-	-	-
Toe Flexion	G PB K044025-JY	0.115	0.96	3.13	CPL-14	100	11	28	54
Arm Flexion	D PB K064050-9D	0.096	6.15	19.27	HFUC-20	100	49	82	147
Arm Rotation	E PB K064025-GD	0.097	3.20	10.21	HFUC-20	100	49	82	147
Arm Adduction Elbow Flexion	F PB K064025-GD	0.097	3.20	10.21	HFUC-17	100	39	54	110
Head Pan	WI MSSI022H-017C	0.009	2.3	9.0	CSF-11	100	8.9	11	25
Head Tilt	HD RSF-5A	0.013	0.5	1.08	RSF-5A	100	0.65	1.4	-

Table B.2: Manufacturer-provided motor and gearbox maximums for LOLA's joint drives, derived from [188].

Joints	Inc. Encoder $^{\text{ticks}}/_{\text{revolution}}$	Inc. Encoder Direction (right side / left side)	Abs. Encoder Resolution [bits]	Abs. Encoder Direction (right side / left side)	Limit Switch Operation
Pelvis Adduction Pelvis Rotation Hip Flexion Hip Rotation Arm Flexion Arm Rotation	11520	Inverted	17	Inverted	GP5 Low =Inhibit
Arm Adduction	11520	Non-Inverted / Inverted	17	Non-Inverted / Inverted	GP5 Low =Inhibit
Hip Adduction	11520	Non-Inverted / Inverted	17	Non-Inverted / Inverted	GP5 Low =Inhibit
Knee Flexion	11520	Non-Inverted	17	Inverted	GP5 Low =Inhibit
Ankle Flexion (outer motor)	11520	Inverted	16	Non-Inverted	-
Ankle Adduction (inner motor)	11520	Inverted	16	Inverted / Non-Inverted	-
Toe Flexion	4000	Inverted	16	Non-Inverted	GP5 Low =Inhibit
Elbow Flexion	11520	Non-Inverted	17	Non-Inverted	GP5 Low =Inhibit
Head Pan	2000	Non-Inverted	-	-	Homing GP5 High =FLS GP6 High =RLS
Head Tilt	2000	Non-Inverted	-	-	Homing GP5 High =RLS GP6 High =FLS

Table B.3: Incremental (Inc.) and absolute (Abs.) encoder data for LOLA's joint drives.

Device / Joint Drive	Station Address	Physical Parent Address	Physical Child Addresses
CANGateway	1006	Master	1036

EtherCAT Coupler	1036	1006	1013(Out)
			1037(Ebus)

EL3602 Analog Input	1037	1036	1038

Junction	1038	1037	1017(X1)
			1018(X2)

Arm Flexion Right	1018	1038	1019
Arm Adduction Right	1019	1018	1039
Arm Rotation Right	1039	1019	1022
Elbow Flexion Right	1022	1039	1041
Hand FTS Right	1041	1022	End
Arm Flexion Left	1017	1038	1016
Arm Adduction Left	1016	1017	1040
Arm Rotation Left	1040	1016	1015
Head Tilt	1015	1040	1014
Head Pan	1014	1015	1023
Elbow Flexion Left	1023	1014	1042
Hand FTS Left	1042	1023	End
Hip Flexion Right	1013	1036	1007
Hip Adduction Right	1007	1013	1008
Hip Rotation Right	1008	1007	1009
Knee Flexion Right	1009	1008	1010
Ankle Adduction Right	1010	1009	1011
Ankle Flexion Right	1011	1010	1012
Toe Flexion Right	1012	1011	1020
Pelvis Adduction	1020	1012	1021
Pelvis Rotation	1021	1020	1024
Hip Flexion Left	1024	1021	1025
Hip Rotation Left	1025	1024	1026
Hip Adduction Left	1026	1025	1027
Knee Flexion Left	1027	1026	1028
Ankle Flexion Left	1028	1027	1029
Ankle Adduction Left	1029	1028	1030
Toe Flexion Left	1030	1029	End

Table B.4: EtherCAT bus topology and station addresses.

Appendix C

Identified Plants

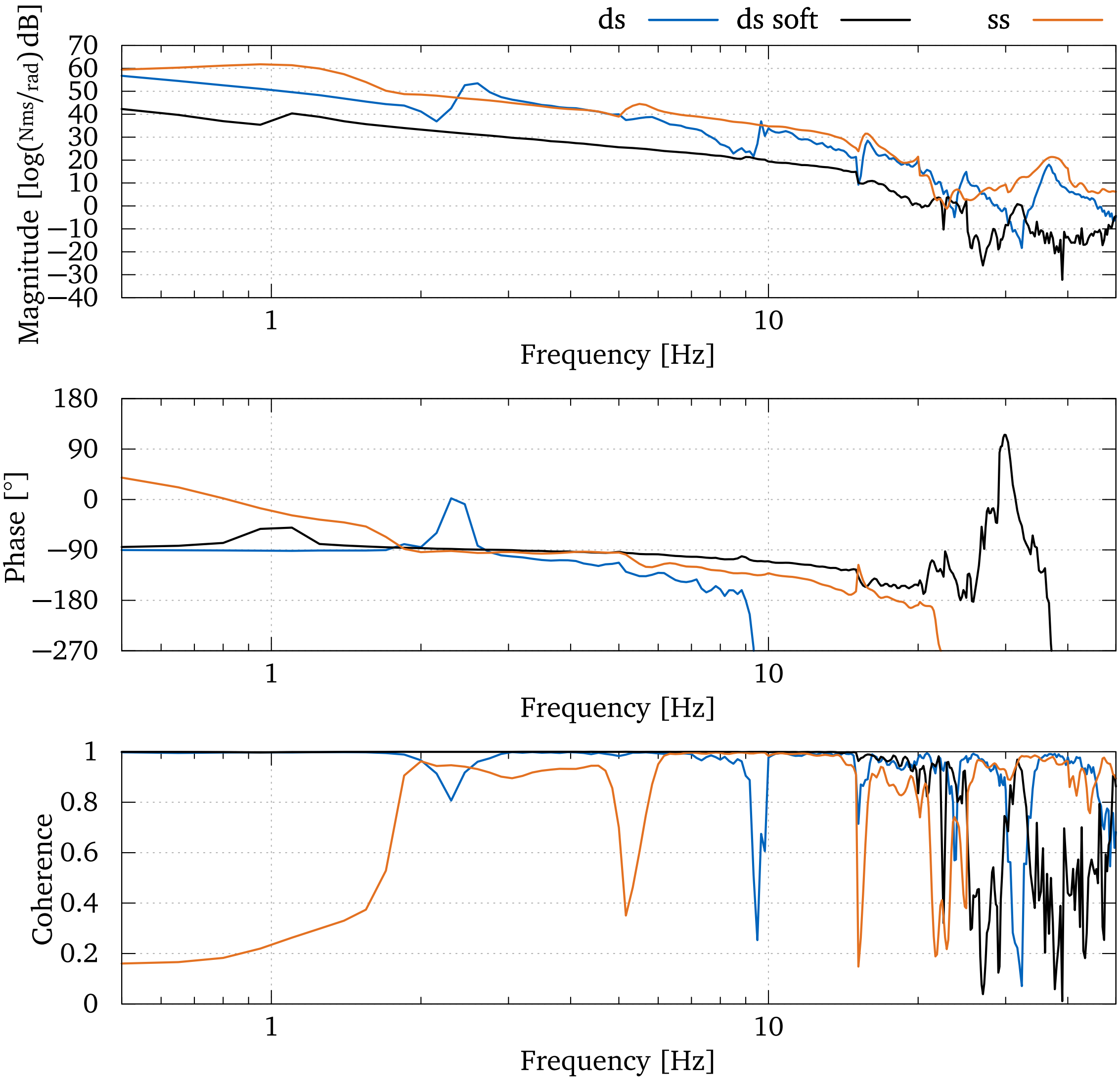


Figure C.1: Estimated measured transfer functions from rotational foot velocity (x-axis) to contact torques (T_x) $-P_{\text{contact},4,4}(s)$ for double support (ds), double support on soft ground (ds soft), and single support (ss). In addition, a magnitude-squared coherence estimate is shown. The foot velocities and contact torques have inverted directions (see Section 3.4.2). Thus, the transfer function to negative contact torques is shown to avoid unintuitive phase readings. A positive phase does not violate physical consistency due to the velocity-level plant input.

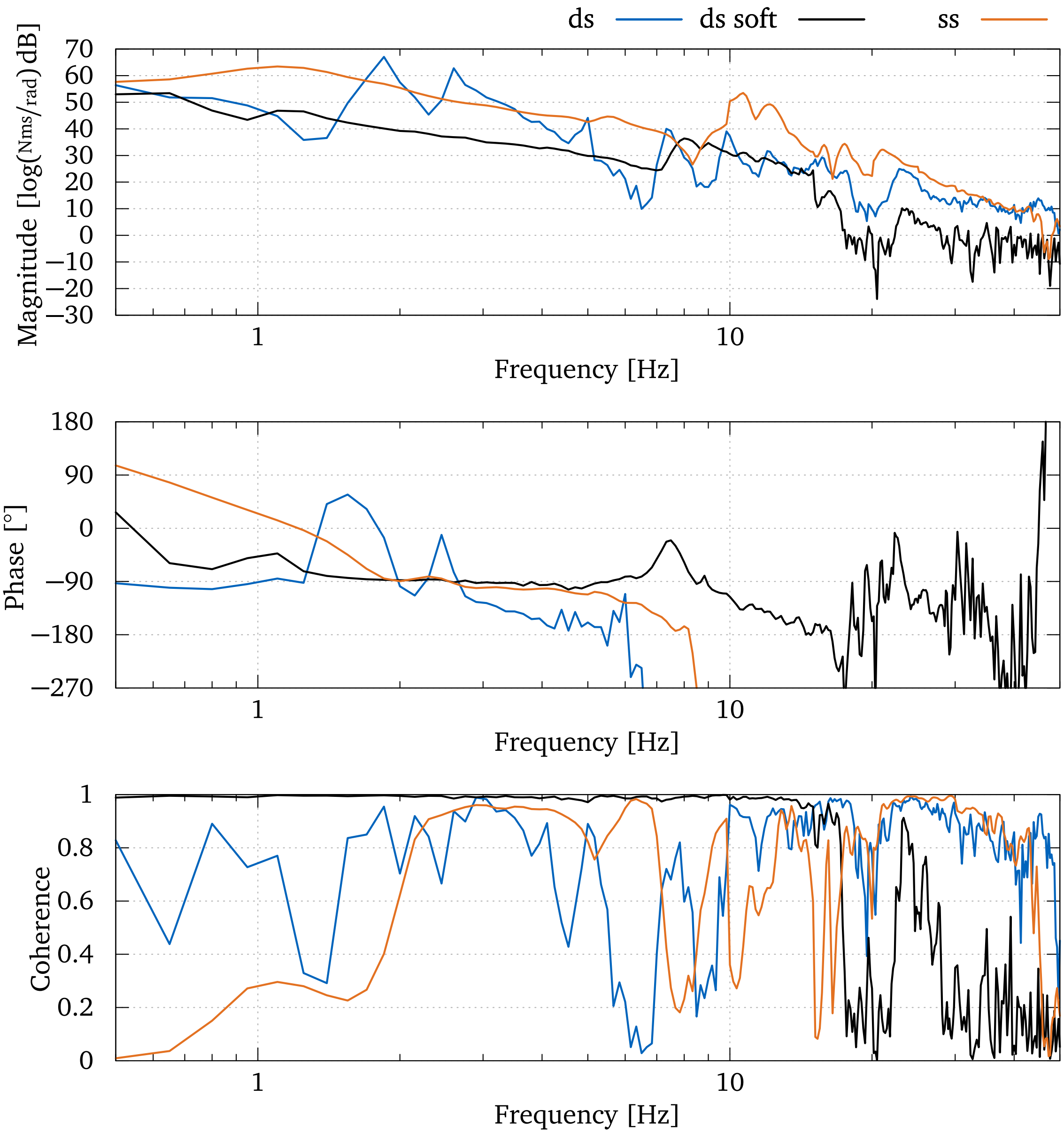


Figure C.2: Estimated measured transfer functions from rotational foot velocity (y-axis) to contact torques (T_y) $-P_{\text{contact},5,5}(s)$ for double support (ds), double support on soft ground (ds soft), and single support (ss). In addition, a magnitude-squared coherence estimate is shown. The foot velocities and contact torques have inverted directions (see Section 3.4.2). Thus, the transfer function to negative contact torques is shown to avoid unintuitive phase readings. A positive phase does not violate physical consistency due to the velocity-level plant input.

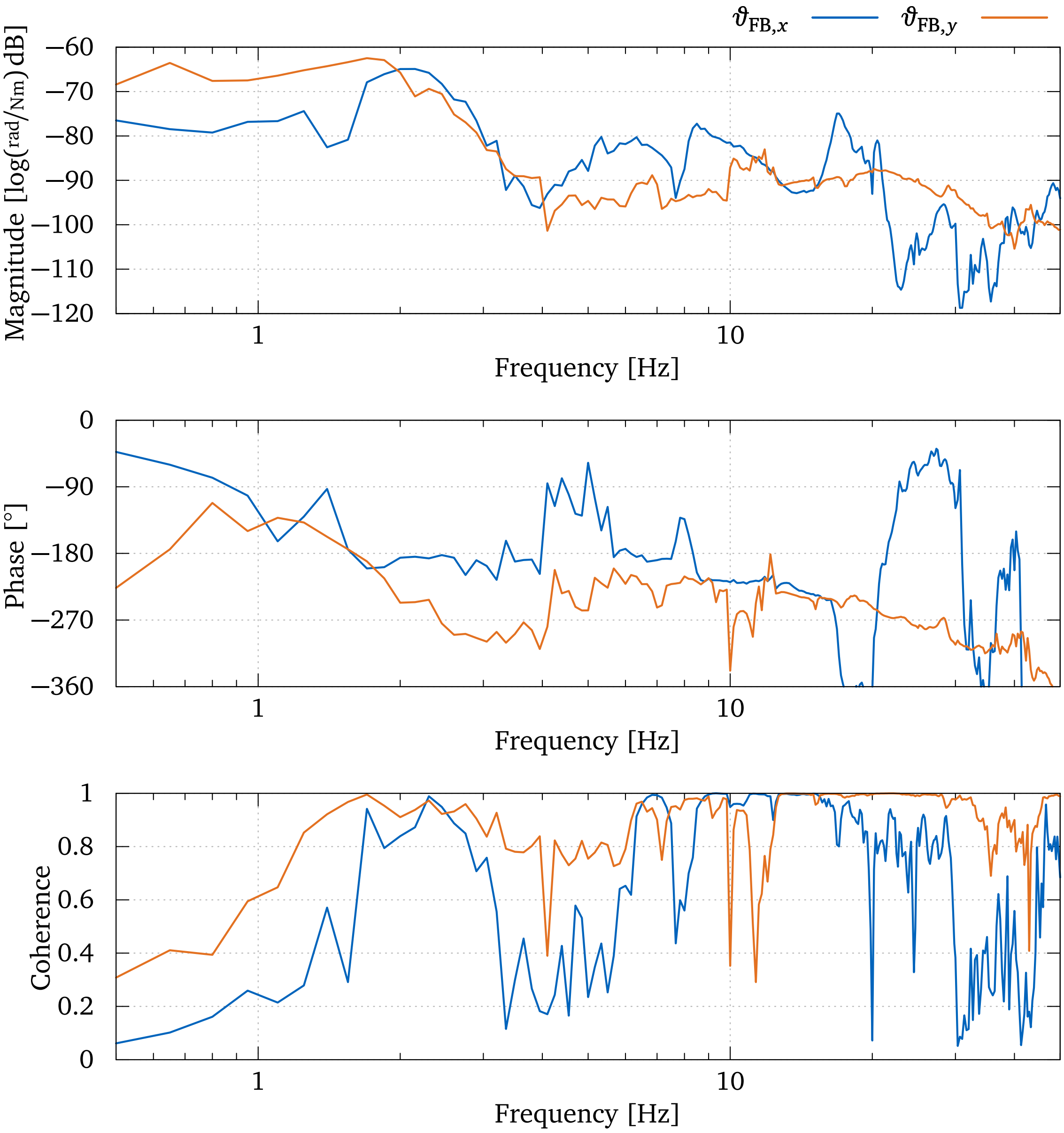


Figure C.3: Estimated transfer function $\boldsymbol{P}_{\text{structure},xy}(s)$ from torques at the CoM to the floating-base inclination ${}_{U}\boldsymbol{\vartheta}_{\text{FB},xy}$ in frontal and sagittal plane, see also Section 3.8.1. The excitation is done solely via vertical contact force variations on both feet, which yields the best coherence for both directions. For the estimation, the transfer from torques to angular velocities is used due to higher sensitivity. The depicted transfer function, however, is already corrected to the correct output (angles).

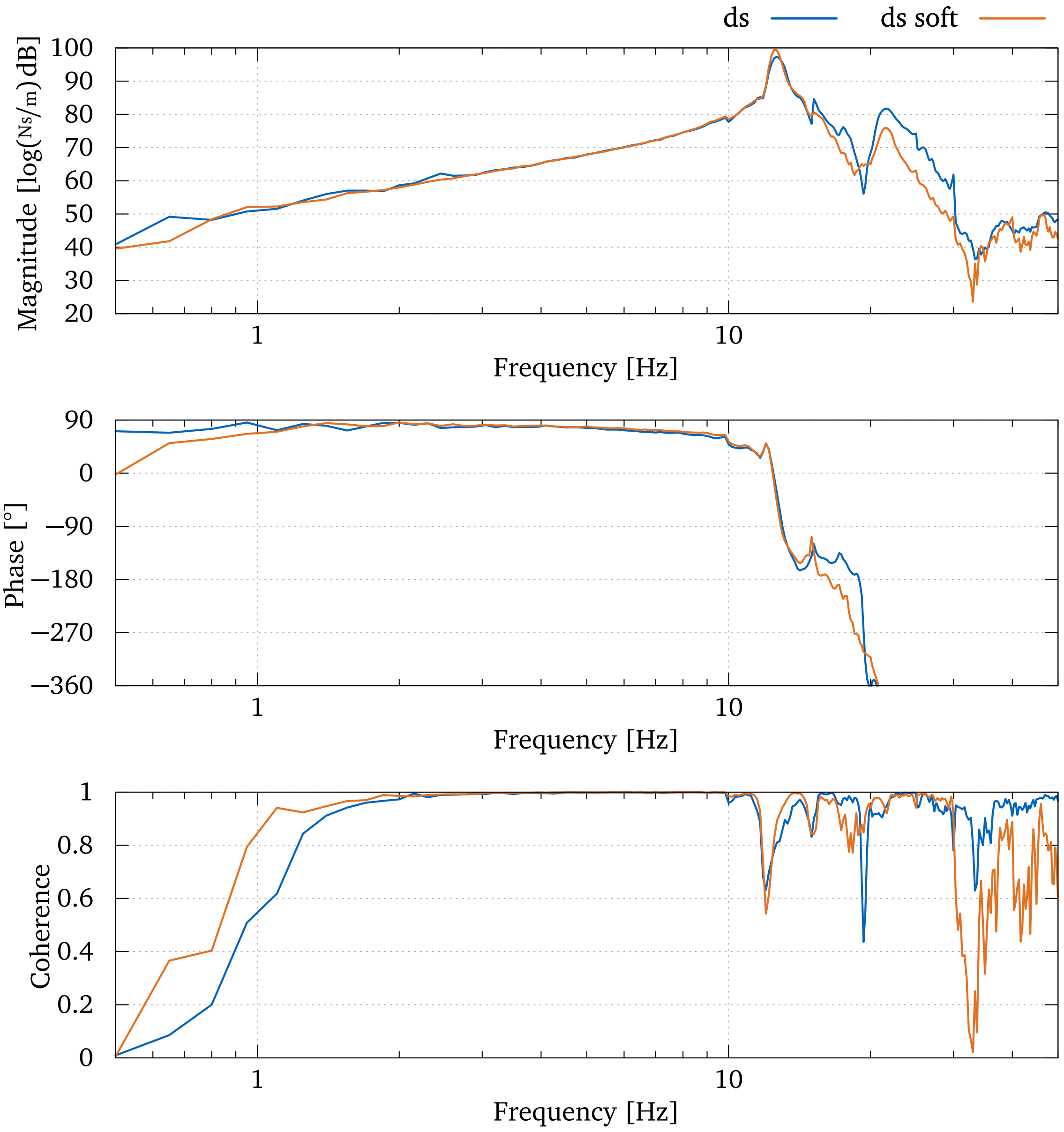


Figure C.4: Estimated transfer function $P_{\text{CoM},3}(s)$ from vertical velocities of the CoM to the total vertical contact forces, see also Section 3.8.1. The estimation is done in double-support (ds), once for a stiff and once for a soft ground. From the matching resonances of both estimates one can draw the conclusion that the contact stiffness has only a damping effect and that the resonances originate from the structure of the robot.

Appendix D

Manufacturing Steps Tactile Sensor

This appendix describes the manufacturing process for the tactile sensor (prototype *C*) described in Section 2.3. The following pictures show the assembly of a toe sensor patch for LOLA. The manufacturing process is mostly independent of the geometry of the sensor patch.

Materials and Tools All required materials and tools are summarized in Table D.1.

Material / Tool	Requirements	Article name
Fabric	Thin but tear-resistant / tough cotton fabric	–
Stainless steel thread	The stainless steel threads must be as thin as possible — otherwise the sewing process may fail or the sensor has a low sensitivity. In this work, a 2-ply stainless steel thread was used.	Adafruit stainless thin conductive thread
Standard thread	This non-conductive thread must not be too thin to get mechanical properties near those of the stainless steel thread. A synthetic material with relatively high stiffness works best.	–
Piezoresistive material	–	Velostat
Sewing machine	Use a robust sewing machine — older machines made of metal, or professional sewing machines perform way better than today's consumer grade machines.	–
Sewing machine needle	Use a thicker needle to get better results with the steel threads, e.g., 1mm thickness.	PN-100/16
Other tools	Fixing pins, scissors, colored pencils, tape	–

Table D.1: Required materials and tools for manufacturing a tactile sensor patch.

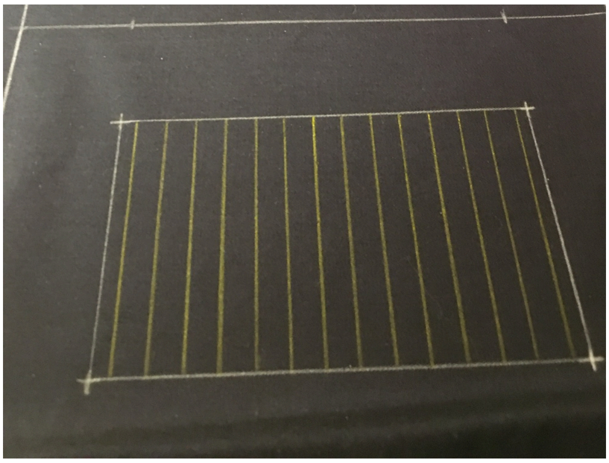
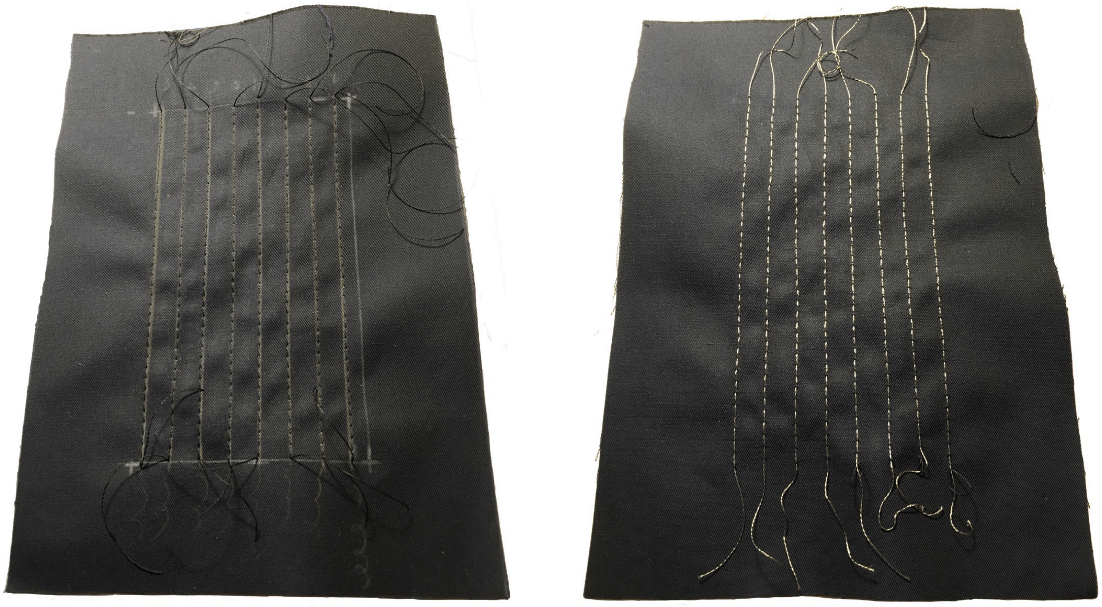


Figure D.1: Sketch of the conductive thread positions (yellow) and the overall size of the senor patch on one of the two fabrics.

Step 1 — Preparing the cotton fabric Start by sketching the sensor patch's desired size and the position of the conductive threads on the cotton fabric (both halves), see Figure D.1. Be aware that the three sensor layers need to be sewed together on the sensor patch's outer rim. This fixing seam should not be too close (> 5 mm distance) to the outer taxels. Otherwise, rim taxels might become pre-tensioned and may behave differently. Cut out the two fabric patches and make sure they are the same size and can be aligned.

Step 2 — Sewing the metal threads into the fabric Use the steel thread as the shuttle (lower) thread in the machine and the non-conductive thread as the upper one. This reduces the wear of the steel material during the sewing process. Stitch lengths of 3 mm or higher worked well. Make sure to leave enough of the steel threads at the patch's edges to connect them to the electronics board later. The cotton fabric does deform a little bit due to the metal threads' tension — this is normal for this sensor design. If the fabric crumples too much or the non-conductive thread ruptures, the two threads' tension is likely too different, and one should select other materials. The final result of one fabric half is shown in Figure D.2.



(a) Upper side

(b) Bottom side

Figure D.2: One fabric half with sewed-in steel threads.

Step 3 — Assembly of the two fabric patches Start by taping the metal threads to the cotton to avoid sewing them into the layered assembly. Use fixing pins to align the two fabric patches on one side, see Figure D.3. Then cut out an appropriate size of the piezoresistive foil — it will be sewed together with the other two layers and should be larger than the actual sensor size. Once the foil is aligned with the planned (marked) positions of the fixing seam on the outer edge, see Figure D.4, tape it down on one side of the cotton. Make sure to remove the tape after the sewing is finished. Fold the three layers together. Again, check the layers' alignment and add more fixing pins to prevent the layers from moving during the sewing process.

Step 4 — Sewing all layers together Use a standard cotton thread to sew all three layers together on the sensor patch's outer rim. Remove all tape and prepare the stainless steel wires for connection.

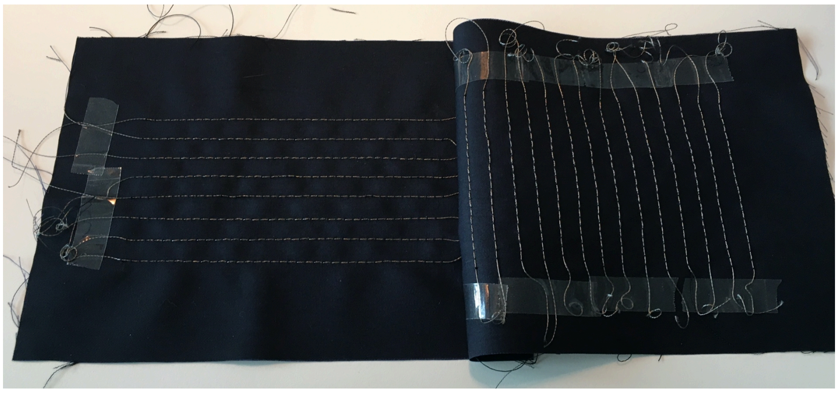


Figure D.3: Aligned fabric halves without piezoresistive layer. Fixing pins are used on the right edge of the two layers to keep the alignment.

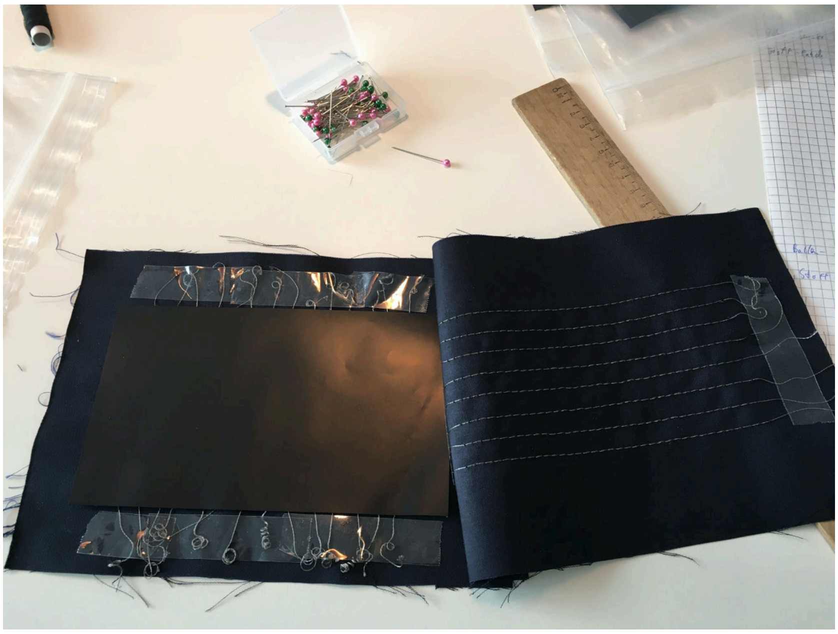


Figure D.4: Aligned fabric halves with piezoresistive layer.

Bibliography

[1] Abadi, M. et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. URL: https://www.tensorflow.org/ (visited on 04/10/2021).

[2] Abi-Farraj, F., Henze, B., Ott, C., Giordano, P. R., and Roa, M. A. "Torque-Based Balancing for a Humanoid Robot Performing High-Force Interaction Tasks". In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 2023–2030. DOI: 10.1109/LRA.2019.2898041.

[3] *Acontis Technologies EC-Master*. URL: https://www.acontis.com/en/ecmaster.html (visited on 03/29/2021).

[4] Agility Robotics. *Digit & Cassie*. URL: https://www.agilityrobotics.com/robots (visited on 04/26/2021).

[5] Ahmed, N., Natarajan, T., and Rao, K. R. "Discrete Cosine Transform". In: *IEEE Transactions on Computers* C-23.1 (1974), pp. 90–93. DOI: 10.1109/T-C.1974.223784.

[6] Akachi, K. et al. "Development of humanoid robot HRP-3P". In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2005, pp. 50–55. DOI: 10.1109/ICHR.2005.1573544.

[7] Albu-Schäffer, A. et al. "The DLR lightweight robot: design and control concepts for robots in human environments". In: *Industrial Robot: An International Journal* 34 (2007), pp. 376–385. DOI: 10.1108/01439910710774386.

[8] An, S. I. and Lee, D. "Prioritized Inverse Kinematics with Multiple Task Definitions". In: *Proceedings IEEE International Conference on Robotics and Automation*. 2015, pp. 1423–1430. DOI: 10.1109/ICRA.2015.7139376.

[9] Ando, N., Suehiro, T., Kitagaki, K., Kotoku, T., and Yoon, W.-k. "RT-Middleware : Distributed Component Middleware for RT (Robot Technology)". In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2015, pp. 3933–3938. DOI: 10.1109/IROS.2005.1545521.

[10] ANYbotics. *ANYmal C*. 2019. URL: https://www.anybotics.com/anymal-legged-robot/ (visited on 04/06/2021).

[11] Audren, H. and Kheddar, A. "3-D Robust Stability Polyhedron in Multicontact". In: *IEEE Transactions on Robotics* 34.2 (2018), pp. 388–403. DOI: 10.1109/TRO.2017.2786683.

[12] Bae, H., Lee, I., Jung, T., and Oh, J. H. "Walking-wheeling dual mode strategy for humanoid robot, DRC-HUBO+". In: *Proceedings IEEE International Conference on Intelligent Robots and Systems*. 2016, pp. 1342–1348. DOI: 10.1109/IROS.2016.7759221.

[13] Baumann, O. "Design of a Controller to Stabilize a Biped Robot under Consideration of Structural Dynamics". Master's Thesis. Technical University of Munich, 2020. URL: https://mediatum.ub.tum.de/1609231.

[14] Bergner, F., Mittendorfer, P., Dean-Leon, E., and Cheng, G. “Event-based signaling for reducing required data rates and processing power in a large-scale artificial robotic skin”. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2015, pp. 2124–2129. DOI: 10.1109/IROS.2015.7353660.

[15] Berninger, T. F. C., Fuderer, S., and Rixen, D. J. “Modal Analysis of a 7 DoF Sweet Pepper Harvesting Robot”. In: *Proceedings Topics in Modal Analysis & Testing, Volume 8*. Springer International Publishing, 2020, pp. 163–170. DOI: 10.1007/978-3-030-12684-1_16.

[16] Berninger, T. F. C., Sygulla, F., Fuderer, S., and Rixen, D. J. “Experimental Analysis of Structural Vibration Problems of a Biped Walking Robot”. In: *Proceedings IEEE International Conference on Robotics and Automation (ICRA)*. 2020, pp. 8726–8731. DOI: 10.1109/ICRA40945.2020.9197282.

[17] Berninger, T. F. C., Seiwald, P., Sygulla, F., and Rixen, D. J. “Evaluating the Mechanical Redesign of a Biped Walking Robot Using Experimental Modal Analysis”. In: *Proceedings Topics in Modal Analysis & Testing, Society for Experimental Mechanics Series*. Ed. by Dilworth B.J., M. M. Vol. 8. 2021. DOI: 10.1007/978-3-030-75996-4_6.

[18] Bhatia, R. *Matrix Analysis*. Springer Science+Business Media New York, 1997. ISBN: 978-1-4612-0653-8. DOI: 10.1007/978-1-4612-0653-8.

[19] Bhattacharjee, T., Jain, A., Vaish, S., Killpack, M. D., and Kemp, C. C. “Tactile Sensing over Articulated Joints with Stretchable Sensors”. In: *Proceedings IEEE World Haptics Conference*. 2013, pp. 103–108. DOI: 10.1109/WHC.2013.6548392.

[20] Bhattacharya, S. et al. “Surface-Property Recognition with Force Sensors for Stable Walking of Humanoid Robot”. In: *IEEE Access* 7 (2019), pp. 146443–146456. DOI: 10.1109/ACCESS.2019.2945983.

[21] Birn-Jeffery, A. V. et al. “Don’t break a leg: Running birds from quail to ostrich prioritise leg safety and economy on uneven terrain”. In: *Journal of Experimental Biology* 217.21 (2014), pp. 3786–3796. DOI: 10.1242/jeb.102640.

[22] Bjelonic, M., Sankar, P. K., Bellicoso, C. D., Vallery, H., and Hutter, M. “Rolling in the Deep - Hybrid Locomotion for Wheeled-Legged Robots Using Online Trajectory Optimization”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 3626–3633. DOI: 10.1109/LRA.2020.2979661.

[23] Bledt, G. and Kim, S. “Extracting Legged Locomotion Heuristics with Regularized Predictive Control”. In: *Proceedings IEEE International Conference on Robotics and Automation*. 2020, pp. 406–412. DOI: 10.1109/ICRA40945.2020.9197488.

[24] Blue Danube Robotics GmbH. *AirSkin*. URL: https://www.airskin.io (visited on 03/24/2021).

[25] Boser, B. E., Guyon, I. M., and Vapnik, V. N. “A Training algorithm for optimal margin classifiers”. In: *Proceedings Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*. Association for Computing Machinery, 1992, pp. 144–152. ISBN: 0-89791-497-X. DOI: 10.1145/130385.130401.

[26] Boston Dynamics. *Atlas, The Next Generation*. URL: https://youtu.be/rVlhMGQgDkY (visited on 03/29/2021).

[27] Boston Dynamics. *Spot*. URL: https://www.bostondynamics.com/spot (visited on 03/29/2021).

[28] Bosworth, W., Whitney, J., Kim, S., and Hogan, N. “Robot locomotion on hard and soft ground: Measuring stability and ground properties in-situ”. In: *Proceedings IEEE International Conference on Robotics and Automation*. Vol. 2016-June. 2016, pp. 3582–3589. DOI: 10.1109/ICRA.2016.7487541.

[29] Brossette, S., Vaillant, J., Keith, F., Escande, A., and Kheddar, A. “Point-cloud multi-contact planning for humanoids: Preliminary results”. In: *Proceedings IEEE Conference on Robotics, Automation and Mechatronics*. 2013, pp. 19–24. DOI: 10.1109/RAM.2013.6758553.

[30] Brossette, S. et al. “Integration of non-inclusive contacts in posture generation”. In: *Proceedings IEEE International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 933–938. DOI: 10.1109/IROS.2014.6942671.

[31] Büscher, G., Meier, M., Walck, G., Haschke, R., and Ritter, H. J. “Augmenting Curved Robot Surfaces with Soft Tactile Skin”. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2015, pp. 1514–1519. DOI: 10.1109/IROS.2015.7353568.

[32] Büscher, G. H., Kõiva, R., Schürmann, C., Haschke, R., and Ritter, H. J. “Flexible and stretchable fabric-based tactile sensor”. In: *Robotics and Autonomous Systems* 63.P3 (2015), pp. 244–252. DOI: 10.1016/j.robot.2014.09.007.

[33] Buschmann, T. “Simulation and Control of Biped Walking Robots”. Dissertation. Technical University of Munich, 2010. ISBN: 978-3-86853-804-5. URL: http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20101201-997204-1-6.

[34] Buschmann, T. “Dynamics and Control of Redundant Robots”. Habilitation Thesis. Technical University of Munich, 2014. URL: http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20150605-1254716-1-0.

[35] Buschmann, T., Ewald, A., Ulbrich, H., and Büschges, A. “Event-Based Walking Control - From Neurobiology to Biped Robots”. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2012. DOI: 10.1109/IROS.2012.6385783.

[36] Buschmann, T., Favot, V., Lohmeier, S., Schwienbacher, M., and Ulbrich, H. “Experiments in Fast Biped Walking”. In: *Proceedings IEEE International Conference on Mechatronics*. IEEE, 2011, pp. 863–868. DOI: 10.1109/ICMECH.2011.5971235.

[37] Buschmann, T., Lohmeier, S., Bachmayer, M., Ulbrich, H., and Pfeiffer, F. “A Collocation Method for Real-Time Walking Pattern Generation”. In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2007, pp. 1–6. DOI: 10.1109/ICHR.2007.4813841.

[38] Buschmann, T., Lohmeier, S., and Ulbrich, H. “Biped Walking Control Based on Hybrid Position/Force Control”. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2009, pp. 3019–3024. DOI: 10.1109/IROS.2009.5354643.

[39] Buschmann, T. et al. “Walking in Unknown Environments — a Step Towards More Autonomy”. In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2010, pp. 237–244. DOI: 10.1109/ICHR.2010.5686338.

[40] CAN in Automation (CiA). *CANopen*. URL: https://www.can-cia.org/can-knowledge/canopen/canopen/ (visited on 04/10/2021).

[41] Cannata, G., Maggiali, M., Metta, G., and Sandini, G. “An embedded artificial skin for humanoid robots”. In: *Proceedings IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*. 2008, pp. 434–438. DOI: 10.1109/MFI.2008.4648033.

[42] Caron, S. “Biped Stabilization by Linear Feedback of the Variable-Height Inverted Pendulum Model”. In: *Proceedings IEEE International Conference on Robotics and Automation*. 2020, pp. 9782–9788. DOI: 10.1109/ICRA40945.2020.9196715.

[43] Caron, S., Kheddar, A., and Tempier, O. “Stair climbing stabilization of the HRP-4 humanoid robot using whole-body admittance control”. In: *Proceedings IEEE International Conference on Robotics and Automation*. 2019, pp. 277–283. DOI: 10.1109/ICRA.2019.8794348.

[44] Caron, S., Pham, Q. C., and Nakamura, Y. “Stability of surface contacts for humanoid robots: Closed-form formulae of the Contact Wrench Cone for rectangular support areas”. In: *Proceedings IEEE International Conference on Robotics and Automation*. 2015, pp. 5107–5112. DOI: 10.1109/ICRA.2015.7139910.

[45] Carpentier, J. and Mansard, N. “Multicontact Locomotion of Legged Robots”. In: *IEEE Transactions on Robotics* 34.6 (2018), pp. 1441–1460. DOI: 10.1109/TRO.2018.2862902.

[46] Chang, C. C. and Lin, C. J. “LIBSVM: A Library for support vector machines”. In: *ACM Transactions on Intelligent Systems and Technology* 2.3 (2011), pp. 1–27. DOI: 10.1145/1961189.1961199.

[47] Chiaverini, S. “Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators”. In: *IEEE Transactions on Robotics and Automation* 13.3 (1997), pp. 398–410. DOI: 10.1109/70.585902.

[48] Cho, B. K., Park, S. S., and Oh, J. H. “Controllers for Running in the Humanoid Robot, HUBO”. In: *Proceedings IEEE/RAS International Conference on Humanoid Robots*. 2009, pp. 385–390. DOI: 10.1109/ICHR.2009.5379574.

[49] Cho, C. and Ryuh, Y. “Fabrication of flexible tactile force sensor using conductive ink and silicon elastomer”. In: *Sensors and Actuators A: Physical* 237 (2016), pp. 72–80. DOI: 10.1016/j.sna.2015.10.051.

[50] Choi, Y., Kim, D., Oh, Y., and You, B. J. “Posture/Walking Control for Humanoid Robot Based on Kinematic Resolution of CoM Jacobian With Embedded Motion”. In: *IEEE Transactions on Robotics* 23.6 (2007), pp. 1285–1293. DOI: 10.1109/TRO.2007.904907.

[51] Chou, J. C. K. “Quaternion Kinematic and Dynamic Differential Equations”. In: *IEEE Transactions on Robotics and Automation* 8.1 (1992), pp. 53–64. DOI: 10.1109/70.127239.

[52] Cisneros, R., Benallegue, M., Morisawa, M., and Kanehiro, F. “QP-based task-space hybrid / parallel control for multi-contact motion in a torque-controlled humanoid robot”. In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2019, pp. 663–670. DOI: 10.1109/Humanoids43949.2019.9035038.

[53] Cortes, C. and Vapnik, V. “Support-vector networks”. In: *Machine Learning* 20.3 (1995), pp. 273–297. DOI: 10.1007/BF00994018.

[54] Craig, J. J. and Raibert, M. H. “A systematic method of hybrid position/force control of a manipulator”. In: *Proceedings IEEE Computer Society’s Third International Computer Software and Applications Conference*. 1979, pp. 446–451. DOI: 10.1109/CMPSAC.1979.762539.

[55] Cremer, L., Heckl, M., and Petersson, B. *Structure-Borne Sound*. Springer-Verlag Berlin Heidelberg, 2005. ISBN: 978-3-540-22696-3.

[56] Dam, E. B., Koch, M., and Lillholm, M. *Quaternions, Interpolation and Animation*. Tech. rep. DIKU-TR-98/5. University of Copenhagen, Department of Computer Science, 1998.

[57] Dan Stiehl, W. and Breazeal, C. "A sensitive skin for robotic companions featuring temperature, force, and electric field sensors". In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2006, pp. 1952–1959. DOI: 10.1109/IROS.2006.282324.

[58] DARPA. *DARPA Robotics Challenge*. URL: https://archive.darpa.mil/roboticschallenge/ (visited on 04/06/2021).

[59] Day, N., Penaloza, J., Santos, V. J., and Killpack, M. D. "Scalable fabric tactile sensor arrays for soft bodies". In: *Journal of Micromechanics and Microengineering* 28.6 (2018), pp. 1–10. DOI: 10.1088/1361-6439/aab221.

[60] De Luca, A. and Mattone, R. "An identification scheme for robot actuator faults". In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2005, pp. 2613–2617. DOI: 10.1109/IROS.2005.1545370.

[61] Di Carlo, J., Wensing, P. M., Katz, B., Bledt, G., and Kim, S. "Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control". In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2018, pp. 7440–7447. DOI: 10.1109/IROS.2018.8594448.

[62] Dixon, W. E., Walker, I. D., Dawson, D. M., and Hartranft, J. P. "Fault detection for robot manipulators with parametric uncertainty: A prediction-error-based approach". In: *IEEE Transactions on Robotics and Automation* 16.6 (2000), pp. 689–699. DOI: 10.1109/70.897780.

[63] Duchaine, V., Lauzier, N., Baril, M., Lacasse, M.-A., and Gosselin, C. "A Flexible Robot Skin for Safe Physical Human Robot Interaction". In: *Proceedings IEEE International Conference on Robotics and Automation*. 2009, pp. 3676–3681. DOI: 10.1109/ROBOT.2009.5152595.

[64] *Elmo Servo Drives Gold Line*. URL: https://www.elmomc.com/products/industrial-environment/servo-drive-gold-family/ (visited on 03/29/2021).

[65] *Elmo Servo Drives SimpleIQ*. URL: https://www.elmomc.com/products/industrial-environment/servo-drive-simpiq-family/ (visited on 03/29/2021).

[66] Englsberger, J., Mesesan, G., Werner, A., and Ott, C. "Torque-Based Dynamic Walking - A Long Way from Simulation to Experiment". In: *Proceedings IEEE International Conference on Robotics and Automation*. 2018, pp. 440–447. DOI: 10.1109/ICRA.2018.8462862.

[67] Englsberger, J., Ott, C., and Albu-Schäffer, A. "Three-dimensional bipedal walking control using divergent component ofmotion". In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2013, pp. 2600–2607. DOI: 10.1109/IROS.2013.6696723.

[68] Englsberger, J., Ott, C., and Albu-Schäffer, A. "Three-Dimensional Bipedal Walking Control Based on Divergent Component of Motion". In: *IEEE Transactions on Robotics* 31.2 (2015), pp. 355–368. DOI: 10.1109/TRO.2015.2405592.

[69] Englsberger, J., Ott, C., Roa, M. A., Albu-Schäffer, A., and Hirzinger, G. “Bipedal walking control based on capture point dynamics”. In: *Proceedings IEEE International Conference on Intelligent Robots and Systems*. 2011, pp. 4420–4427. DOI: 10.1109/IROS.2011.6048045.

[70] Englsberger, J. et al. “Overview of the torque-controlled humanoid robot TORO”. In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2014, pp. 916–923. DOI: 10.1109/HUMANOIDS.2014.7041473.

[71] Escande, A., Kheddar, A., and Miossec, S. “Planning support contact-points for humanoid robots and experiments on HRP-2”. In: *Proceedings IEEE International Conference on Intelligent Robots and Systems*. 2006, pp. 2974–2979. DOI: 10.1109/IROS.2006.282154.

[72] *esd electronics CAN-EtherCAT Gateway*. URL: https://esd.eu/en/products/can-ethercat (visited on 04/06/2021).

[73] *EtherCAT Technology Group*. URL: https://www.ethercat.org (visited on 03/29/2021).

[74] Ewald, A., Beutler, F., and Buschmann, T. “Ground Contact Detection with Short-stroke Buttons for Biped Robots”. In: *Pamm* 13.1 (2013), pp. 477–478. DOI: 10.1002/pamm.201310231.

[75] Fahmi, S. et al. “STANCE: Locomotion Adaptation over Soft Terrain”. In: *IEEE Transactions on Robotics* 36.2 (2020), pp. 443–457. DOI: 10.1109/TRO.2019.2954670.

[76] Faller, L.-M., Stetco, C., Mitterer, T., and Zangl, H. “An All-Flexible Sensing Sole For Legged Robots”. In: *Proceedings IEEE FLEPS 2019*. 2019, pp. 1–3. DOI: 10.1109/FLEPS.2019.8792287.

[77] Fallon, M. F. et al. “Continuous Humanoid Locomotion over Uneven Terrain using Stereo Fusion”. In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2015, pp. 881–888. DOI: 10.1109/HUMANOIDS.2015.7363465.

[78] Farshidian, F., Jelavic, E., Satapathy, A., Giftthaler, M., and Buchli, J. “Real-Time motion planning of legged robots: A model predictive control approach”. In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2017, pp. 577–584. DOI: 10.1109/HUMANOIDS.2017.8246930.

[79] Farshidian, F., Jelavic, E., Winkler, A. W., and Buchli, J. “Robust whole-body motion control of legged robots”. In: *Proceedings IEEE International Conference on Intelligent Robots and Systems*. 2017, pp. 4589–4596. DOI: 10.1109/IROS.2017.8206328.

[80] Favot, V. *Hierarchical Joint Control of Humanoid Robots*. Dissertation. Technical University of Munich, 2016. URL: http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20161206-1294180-1-9.

[81] Favot, V., Buschmann, T., Schwienbacher, M., Ewald, A., and Ulbrich, H. “The Sensor-Controller Network of the Humanoid Robot LOLA”. In: *Proceedings IEEE/RAS International Conference on Humanoid Robots*. 2012, pp. 805–810. DOI: 10.1109/HUMANOIDS.2012.6651612.

[82] Fernández-Francos, D., Marténez-Rego, D., Fontenla-Romero, O., and Alonso-Betanzos, A. “Automatic bearing fault diagnosis based on one-class m-SVM”. In: *Computers and Industrial Engineering* 64.1 (2013), pp. 357–365. DOI: 10.1016/j.cie.2012.10.013.

[83] Ferrati, M. et al. “The Walk-Man Robot Software Architecture”. In: *Frontiers in Robotics and AI* 3.25 (2016). DOI: 10.3389/frobt.2016.00025.

[84] Filitchkin, P. and Byl, K. “Feature-based terrain classification for LittleDog”. In: *Proceedings IEEE International Conference on Intelligent Robots and Systems*. 2012, pp. 1387–1392. DOI: 10.1109/IROS.2012.6386042.

[85] Freymann, M. von. “Experimentelle Identifikation von Kontakteigenschaften der Fußsohle eines humanoiden Roboters”. Semester Thesis. Technical University of Munich, 2018. URL: https://mediatum.ub.tum.de/1463742.

[86] Fujimoto, Y. and Kawamura, A. “Proposal of Biped Walking Control Based on Robust Hybrid Position/Force control”. In: *Proceedings IEEE International Conference on Robotics and Automation*. 1996, pp. 2724–2730. DOI: 10.1109/ROBOT.1996.506574.

[87] Gong, Y. et al. “Feedback control of a cassie bipedal robot: Walking, standing, and riding a segway”. In: *Proceedings American Control Conference*. American Automatic Control Council, 2019, pp. 4559–4566. DOI: 10.23919/acc.2019.8814833.

[88] Goswami, A. “Foot rotation indicator (FRI) point: a new gait planning tool to evaluate postural stability of biped robots”. In: *Proceedings IEEE International Conference on Robotics and Automation*. Vol. 1. 1999, pp. 47–52. DOI: 10.1109/ROBOT.1999.769929.

[89] Goswami, A. and Kallem, V. “Rate of change of angular momentum and balance maintenance of biped robots”. In: *Proceedings IEEE International Conference on Robotics and Automation*. Vol. 4. 2004, pp. 3785–3790. DOI: 10.1109/ROBOT.2004.1308858.

[90] Grandia, R., Farshidian, F., Dosovitskiy, A., Ranftl, R., and Hutter, M. “Frequency-Aware Model Predictive Control”. In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 1517–1524. DOI: 10.1109/LRA.2019.2895882.

[91] Griffin, R. J., Wiedebach, G., Bertrand, S., Leonessa, A., and Pratt, J. “Walking stabilization using step timing and location adjustment on the humanoid robot, Atlas”. In: *Proceedings IEEE International Conference on Intelligent Robots and Systems*. 2017, pp. 667–673. DOI: 10.1109/IROS.2017.8202223.

[92] Griffin, R. J., Wiedebach, G., Bertrand, S., Leonessa, A., and Pratt, J. “Straight-Leg Walking Through Underconstrained Whole-Body Control”. In: *Proceedings IEEE International Conference on Robotics and Automation*. 2018, pp. 5747–5754. DOI: 10.1109/ICRA.2018.8460751.

[93] Griffin, R. J. et al. “Footstep Planning for Autonomous Walking over Rough Terrain”. In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2019, pp. 9–16. DOI: 10.1109/Humanoids43949.2019.9035046.

[94] Grundner, C. “Whole-Body-Momentum Control: Anwendung auf den humanoiden Roboter LOLA”. Master’s Thesis (unpublished). Technical University of Munich, 2019.

[95] Guennebaud, G., Jacob, B., et al. *Eigen v3*. 2010. URL: http://eigen.tuxfamily.org.

[96] Guo, S., Yang, T., Gao, W., and Zhang, C. “A novel fault diagnosis method for rotating machinery based on a convolutional neural network”. In: *Sensors (Switzerland)* 18.5 (2018), pp. 1–16. DOI: 10.3390/s18051429.

[97] Hamilton, S. W. R. “LXXVIII. On quaternions; or on a new system of imaginaries in Algebra”. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 25.169 (1844), pp. 489–495. DOI: 10.1080/14786444408645047.

[98] Hannaford, B. and Ryu, J.-H. “Time Domain Passivity Control of Haptic Interfaces”. In: *Proceedings IEEE International Conference on Robotics & Automation*. IEEE, 2001, pp. 1863–1869. DOI: 10.1109/ROBOT.2001.932880.

[99] Harada, K., Kajita, S., Kaneko, K., and Hirukawa, H. "An Analytical Method on Real-time Gait Planning for a Humanoid Robot". In: *Proceedings IEEE/RAS International Conference on Humanoid Robots*. 2004, pp. 640–655. DOI: 10.1109/ICHR.2004.1442676.

[100] Harder, M. "Experimentelle Untersuchung der charakteristischen Eigenschaften einer taktilen Roboter- Haut". Bachelor's Thesis (unpublished). Technical University of Munich, 2016.

[101] Hashimoto, K. et al. "Terrain-Adaptive Control to Reduce Landing Impact Force for Human-Carrying Biped Robot". In: *Proceedings IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. 2009, pp. 174–179. DOI: 10.1109/AIM.2009.5230020.

[102] Hashimoto, K., Yoshimura, Y., Kondo, H., Lim, H. O., and Takanishi, A. "Realization of quick turn of biped humanoid robot by using slipping motion with both feet". In: *Proceedings IEEE International Conference on Robotics and Automation*. 2011, pp. 2041–2046. DOI: 10.1109/ICRA.2011.5980498.

[103] Hashimoto, K. et al. "Biped landing pattern modification method with nonlinear compliance control". In: *Proceedings IEEE International Conference on Robotics and Automation*. 2006, pp. 1213–1218. DOI: 10.1109/ROBOT.2006.1641874.

[104] Hashimoto, K. et al. "Realization of biped walking on soft ground with stabilization control based on gait analysis". In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2012, pp. 2064–2069. DOI: 10.1109/IROS.2012.6385684.

[105] Henze, B., Balachandran, R., Roa-Garzon, M. A., Ott, C., and Albu-Schaffer, A. "Passivity analysis and control of humanoid robots on movable ground". In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 3457–3464. DOI: 10.1109/LRA.2018.2853266.

[106] Henze, B., Roa, M. A., and Ott, C. "Passivity-based whole-body balancing for torque-controlled humanoid robots in multi-contact scenarios". In: *International Journal of Robotics Research* 35.12 (2016), pp. 1522–1543. DOI: 10.1177/0278364916653815.

[107] Heo, J.-w., Lee, I.-h., and Oh, J.-h. "Development of Humanoid Robots in HUBO Laboratory , KAIST". In: *Journal of the Robotics Society of Japan* 30.4 (2012), pp. 367–371. DOI: 10.7210/jrsj.30.367.

[108] Herzog, A., Righetti, L., Grimminger, F., Pastor, P., and Schaal, S. "Balancing experiments on a torque-controlled humanoid with hierarchical inverse dynamics". In: *Proceedings IEEE International Conference on Intelligent Robots and Systems*. 2014, pp. 981–988. DOI: 10.1109/IROS.2014.6942678.

[109] Hildebrandt, A.-C. et al. "Real-time predictive kinematic evaluation and optimization for biped robots". In: *Proceedings IEEE International Conference on Intelligent Robots and Systems*. 2016, pp. 5789–5796. DOI: 10.1109/IROS.2016.7759852.

[110] Hildebrandt, A. C. et al. "Torso height optimization for bipedal locomotion". In: *International Journal of Advanced Robotic Systems* 15.5 (2018), pp. 1–11. DOI: 10.1177/1729881418804442.

[111] Hildebrandt, A.-C. "Autonomous Robots in Unknown and Dynamic Scenarios". Dissertation. Technical University of Munich, 2018. URL: http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20181019-1441767-1-4.

[112] Hildebrandt, A.-C. et al. “Real-Time Path Planning in Unknown Environments for Bipedal Robots”. In: *IEEE Robotics and Automation Letters* 2.4 (2017), pp. 1856–1863. DOI: 10.1109/lra.2017.2712650.

[113] Hildebrandt, A.-C. et al. “Kinematic optimization for bipedal robots: a framework for real-time collision avoidance”. In: *Journal of Autonomous Robots* 43.5 (2019), pp. 1187–1205. DOI: 10.1007/s10514-018-9789-3.

[114] Hildebrandt, A.-C. et al. “Versatile and robust bipedal walking in unknown environments: real-time collision avoidance and disturbance rejection”. In: *Journal of Autonomous Robots* 43 (2019), pp. 1957–1976. DOI: 10.1007/s10514-019-09838-3.

[115] Hirai, K., Hirose, M., Haikawa, Y., and Takenaka, T. “The Development of Honda Humanoid Robot”. In: *Proceedings IEEE International Conference on Robotics and Automation*. 1998, pp. 1321–1326. DOI: 10.1109/ROBOT.1998.677288.

[116] Hiraoka, N. et al. “Whole-Body Control of Humanoid Robot in 3D Multi-Contact under Contact Wrench Constraints Including Joint Load Reduction with Self-Collision and Internal Wrench Distribution”. In: *Proceedings IEEE International Conference on Intelligent Robots and Systems*. 2019, pp. 3860–3867. DOI: 10.1109/IROS40897.2019.8967555.

[117] Hirose, M. and Ogawa, K. “Honda humanoid robots development”. In: *Philosophical transactions. Series A, Mathematical, Physical, and Engineering Sciences* 365.1850 (2007), pp. 11–19. DOI: 10.1098/rsta.2006.1917.

[118] Hirschmann, H. “Entwicklung einer taktilen Fußsohle für einen humanoiden Roboter”. Master’s Thesis. Technical University of Munich, 2017. URL: https://mediatum.ub.tum.de/1388135.

[119] Hirukawa, H. et al. “A universal stability criterion of the foot contact of legged robots - adios ZMP”. In: *Proceedings IEEE International Conference on Robotics and Automation*. 2006, pp. 1976–1983. DOI: 10.1109/ROBOT.2006.1641995.

[120] Ho, V. A., Hirai, S., and Naraki, K. “Fabric Interface with Proximity and Tactile Sensation for Human-Robot Interaction”. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2016, pp. 238–245. DOI: 10.1109/IROS.2016.7759061.

[121] Hoepflinger, M. A., Remy, C. D., Hutter, M., Haag, S., and Siegwart, R. “Haptic terrain classification on natural terrains for legged robots”. In: *Proceedings 13th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*. 2010, pp. 785–792. DOI: 10.1142/9789814329927_0097.

[122] Hopkins, M. A., Griffin, R. J., Leonessa, A., Lattimer, B. Y., and Furukawa, T. “Design of a compliant bipedal walking controller for the DARPA Robotics Challenge”. In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2015, pp. 831–837. DOI: 10.1109/HUMANOIDS.2015.7363450.

[123] Hoshi, T. and Shinoda, H. “A Tactile Sensing Element for a Whole Body Robot Skin”. In: *International Symposium on Robotics* (2005), pp. 1–6.

[124] Huang, Q. and Nakamura, Y. “Sensory Reflex Control for Humanoid Walking”. In: *IEEE Transactions on Robotics* 21.5 (2005), pp. 977–984. DOI: 10.1109/TRO.2005.851381.

[125] Hubicki, C. et al. “ATRIAS: Design and validation of a tether-free 3D-capable spring-mass bipedal robot”. In: *International Journal of Robotics Research* 35.12 (2016), pp. 1497–1521. DOI: 10.1177/0278364916648388.

[126] Hutter, M. et al. “ANYmal - A highly mobile and dynamic quadrupedal robot”. In: *Proceedings IEEE International Conference on Intelligent Robots and Systems*. 2016, pp. 38–44. DOI: 10.1109/IROS.2016.7758092.

[127] Hyon, S. H. and Cheng, G. “Passivity-based Full-Body Force Control for Humanoids and Application to Dynamic Balancing and Locomotion”. In: *Proceedings IEEE International Conference on Intelligent Robots and Systems*. 2006, pp. 4915–4922. DOI: 10.1109/IROS.2006.282450.

[128] IEEE Spectrum. *Who Is SCHAFT, the Robot Company Bought by Google and Winner of the DRC?* URL: https://spectrum.ieee.org/automaton/robotics/humanoids/schaft-robot-company-bought-by-google-darpa-robotics-challenge-winner (visited on 03/29/2021).

[129] *iMAR Navigation*. URL: https://www.imar-navigation.de/ (visited on 03/29/2021).

[130] Ioannou, P. A. and Kokotovic, P. V. “Instability analysis and improvement of robustness of adaptive control”. In: *Automatica* 20.5 (1984), pp. 583–594. DOI: 10.1016/0005-1098(84)90009-8.

[131] Ito, Y. et al. “Development and Verification of Life-Size Humanoid with High-Output Actuation System”. In: *Proceedings IEEE International Conference on Robotics and Automation*. 2014, pp. 3433–3438. DOI: 10.1109/ICRA.2014.6907353.

[132] Joan Sola. *Quaternion Kinematics for Error-State KF*. Tech. rep. 2017. URL: https://arxiv.org/abs/1711.02508.

[133] Kagami, S. et al. “Online 3D vision, motion planning and bipedal locomotion control coupling system of humanoid robot: H7”. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2002, pp. 2557–2562. DOI: 10.1109/IRDS.2002.1041655.

[134] Kagami, S. et al. “Humanoid HRP2-DHRC for autonomous and interactive behavior”. In: *Robotics Research*. Ed. by Thrun, S., Brooks, R., and Durrant-Whyte, H. Springer, Berlin, Heidelberg, 2007. DOI: 10.1007/978-3-540-48113-3_10.

[135] Kajita, S., Nagasaki, T., Kaneko, K., Yokoi, K., and Tanie, K. “A Running Controller of Humanoid Biped HRP-2LR”. In: *Proceedings 2005 IEEE International Conference on Robotics and Automation*. 2005, pp. 616–622. DOI: 10.1109/ROBOT.2005.1570186.

[136] Kajita, S. et al. “Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point”. In: *Proceedings IEEE International Conference on Robotics and Automation*. 2003, pp. 1620–1626. DOI: 10.1109/ROBOT.2003.1241826.

[137] Kajita, S. et al. “Biped Walking Stabilization Based on Linear Inverted Pendulum Tracking”. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vol. 2. 2010, pp. 4489–4496. DOI: 10.1109/IROS.2010.5651082.

[138] Kajita, S. et al. “Vertical vibration suppression for a position controlled biped robot”. In: *Proceedings IEEE International Conference on Robotics and Automation*. 2013, pp. 1637–1642. DOI: 10.1109/ICRA.2013.6630789.

[139] Kaminaga, H. et al. “Mechanism and Control of Whole-Body Electro-Hydrostatic Actuator Driven Humanoid Robot Hydra”. In: *Proceedings International Symposium on Experimental Robotics. ISER 2016. Springer Proceedings in Advanced Robotics*. Ed. by Kulić, D., Nakamura, Y., Khatib, O., and Venture, G. Vol. 1. Springer, Cham, 2017, pp. 656–665. DOI: 10.1007/978-3-319-50115-4_57.

[140] Kamioka, T. et al. “Push Recovery Strategy of Dynamic Gait Transition between Walking, Running and Hopping”. In: *International Journal of Humanoid Robotics* 16.3 (2019). DOI: 10.1142/S0219843619400012.

[141] Kaneko, K. et al. “Humanoid robot HRP-2”. In: *Proceedings IEEE International Conference on Robotics and Automation*. 2004, pp. 1083–1090. DOI: 10.1109/ROBOT.2004.1307969.

[142] Kaneko, K., Harada, K., Kanehiro, F., Miyamori, G., and Akachi, K. “Humanoid Robot HRP-3”. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2008, pp. 2471–2478. DOI: 10.1109/IROS.2008.4650604.

[143] Kaneko, K. et al. “Design of prototype humanoid robotics platform for HRP”. In: *Proceedings IEEE International Conference on Intelligent Robots and Systems*. 2002, pp. 2431–2436. DOI: 10.1109/irds.2002.1041632.

[144] Kaneko, K. et al. “Humanoid robot HRP-4 - Humanoid robotics platform with lightweight and slim body”. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2011, pp. 4400–4407. DOI: 10.1109/IROS.2011.6094465.

[145] Kaneko, K. et al. “Humanoid robot HRP-5P: An electrically actuated humanoid robot with high-power and wide-range joints”. In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 1431–1438. DOI: 10.1109/LRA.2019.2896465.

[146] Kanoulas, D. et al. “Vision-Based Foothold Contact Reasoning using Curved Surface Patches”. In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2017, pp. 121–128. DOI: 10.1109/HUMANOIDS.2017.8239546.

[147] Karumanchi, S. et al. “Team RoboSimian: Semi-autonomous Mobile Manipulation at the 2015 DARPA Robotics Challenge Finals”. In: *Journal of Field Robotics* 34.2 (2016), pp. 305–332. DOI: 10.1002/rob.21676.

[148] Kashiri, N. et al. “CENTAURO: A hybrid locomotion and high power resilient manipulation platform”. In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 1595–1602. DOI: 10.1109/LRA.2019.2896758.

[149] Kato, I. “[Development of the Biped Robot WABOT-1]”. In: *[Biomechanisms Japan]* 2 (1973), pp. 173–174. DOI: 10.3951/biomechanisms.2.173.

[150] Katz, B., Carlo, J. D., and Kim, S. “Mini cheetah: A Platform for Pushing the Limits of Dynamic Quadruped Control”. In: *Proceedings IEEE International Conference on Robotics and Automation*. 2019, pp. 6295–6301. DOI: 10.1109/ICRA.2019.8793865.

[151] Keith, F., Wieber, P. B., Mansard, N., and Kheddar, A. “Analysis of the discontinuities in prioritized tasks-space control under discreet task scheduling operations”. In: *Proceedings IEEE International Conference on Intelligent Robots and Systems*. 2011, pp. 3887–3892. DOI: 10.1109/IROS.2011.6048309.

[152] Kertesz, C. “Rigidity-Based Surface Recognition for a Domestic Legged Robot”. In: *IEEE Robotics and Automation Letters* 1.1 (2016), pp. 309–315. DOI: 10.1109/LRA.2016.2519949.

[153] Khalastchi, E. and Kalech, M. “On fault detection and diagnosis in robotic systems”. In: *ACM Computing Surveys* 51.1 (2018), pp. 1–24. DOI: 10.1145/3146389.

[154] Kheddar, A. and Billard, A. “A Tactile Matrix for Whole-Body Humanoid Haptic Sensing and Safe interaction”. In: *Proceedings IEEE International Conference on Robotics and Biomimetics*. 2011, pp. 1433–1438. DOI: 10.1109/ROBIO.2011.6181491.

[155] Kim, G., Vu, C. C., and Kim, J. “Single-layer pressure textile sensors withwoven conductive yarn circuit”. In: *Applied Sciences (Switzerland)* 10.8 (2020), pp. 1–10. DOI: 10.3390/APP10082877.

[156] Kim, J. Y., Park, I. W., and Oh, J. H. "Walking Control Algorithm of Biped Humanoid Robot on Uneven and Inclined Floor". In: *Journal of Intelligent and Robotic Systems* 48.4 (2007), pp. 457–484. DOI: 10.1007/s10846-006-9107-8.

[157] Kim, J. Y., Lee, J., and Ho, J. H. "Experimental realization of dynamic walking for a human-riding biped robot, HUBO FX-1". In: *Advanced Robotics* 21.3-4 (2007), pp. 461–484. DOI: 10.1163/156855307780132063.

[158] Kim, K. et al. "A silicon-based flexible tactile sensor for ubiquitous robot companion applications". In: *Journal of Physics: Conference Series* 34.1 (2006), pp. 399–403. DOI: 10.1088/1742-6596/34/1/065.

[159] Kim, S. et al. "Approach of Team SNU to the DARPA Robotics Challenge finals". In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2015, pp. 777–784. DOI: 10.1109/HUMANOIDS.2015.7363458.

[160] Kim, Y. D., Lee, B. J., Ryu, J. H., and Kim, J. H. "Landing force control for humanoid robot by time-domain passivity approach". In: *IEEE Transactions on Robotics* 23.6 (2007), pp. 1294–1301. DOI: 10.1109/TRO.2007.906250.

[161] Kinoshita, G., Kimura, T., and Shimojo, M. "Dynamic Sensing Experiments of Reaction Force Distributions on the Sole of a Walking Humanoid Robot". In: *IEEE International Conference on Intelligent Robots and Systems* 2.2 (2003), pp. 1413–1418. DOI: 10.1109/iros.2003.1248842.

[162] Kinoshita, G., Oota, C., Osumi, H., and Shimojo, M. "Acquisition of reaction force distributions for a walking humanoid robot". In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* 4.l (2004), pp. 3859–3864. DOI: 10.1109/iros.2004.1390016.

[163] Koenemann, J. et al. "Whole-body model-predictive control applied to the HRP-2 humanoid". In: *Proceedings IEEE International Conference on Intelligent Robots and Systems*. Vol. 2015-Decem. 2015, pp. 3346–3351. DOI: 10.1109/IROS.2015.7353843.

[164] Koiva, R., Zenker, M., Schürmann, C., Haschke, R., and Ritter, H. J. "A highly sensitive 3D-shaped tactile sensor". In: *Proceedings IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. 2013, pp. 1084–1089. DOI: 10.1109/AIM.2013.6584238.

[165] Koolen, T., Posa, M., and Tedrake, R. "Balance control using center of mass height variation: limitations imposed by unilateral contact". In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2016, pp. 8–15. DOI: 10.1109/HUMANOIDS.2016.7803247.

[166] Koolen, T. et al. "Design of a Momentum-Based Control Framework and Application to the Humanoid Robot Atlas". In: *International Journal of Humanoid Robotics* 13.1 (2016). DOI: 10.1142/S0219843616500079.

[167] Kudruss, M. et al. "Optimal control for whole-body motion generation using center-of-mass dynamics for predefined multi-contact configurations". In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2015, pp. 684–689. DOI: 10.1109/HUMANOIDS.2015.7363428.

[168] Kuindersma, S., Permenter, F., and Tedrake, R. "An efficiently solvable quadratic program for stabilizing dynamic locomotion". In: *Proceedings IEEE International Conference on Robotics and Automation*. 2014, pp. 2589–2594. DOI: 10.1109/ICRA.2014.6907230.

[169] Kuindersma, S. et al. “Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot”. In: *Autonomous Robots* 40.3 (2015), pp. 1–27. DOI: 10.1007/s10514-015-9479-3.

[170] KUKA. *The Mobile Robot KMR iiwa*. URL: https://www.kuka.com/en-de/products/mobility/mobile-robots/kmr-iiwa (visited on 03/29/2021).

[171] Kumagai, I., Morisawa, M., Benallegue, M., and Kanehiro, F. “Bipedal Locomotion Planning for a Humanoid Robot Supported by Arm Contacts Based on Geometrical Feasibility”. In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2019, pp. 132–139. DOI: 10.1109/Humanoids43949.2019.9035072.

[172] Künzel, F. “Konstruktion neuartiger Fußgeometrien für einen humanoiden Laufroboter”. Master’s Thesis. Technical University of Munich, 2017. URL: https://mediatum.ub.tum.de/1378389.

[173] Lacasse, M. A., Duchaine, V., and Gosselin, C. “Characterization of the Electrical Resistance of Carbon-Black-Filled Silicone: Application to a Flexible and Stretchable Robot Skin”. In: *Proceedings IEEE International Conference on Robotics and Automation*. 2010, pp. 4842–4848. DOI: 10.1109/ROBOT.2010.5509283.

[174] Lee, H.-K., Chang, S.-I., and Yoon, E. “A Flexible Polymer Tactile Sensor: Fabrication and Modular Expandability for Large Area Deployment”. In: *Journal of Microelectromechanical Systems* 15.6 (2006), pp. 1681–1686. DOI: 10.1109/JMEMS.2006.886021.

[175] Lee, S. H. and Goswami, A. “A momentum-based balance controller for humanoid robots on non-level and non-stationary ground”. In: *Autonomous Robots* 33 (2012), pp. 399–414. DOI: 10.1007/s10514-012-9294-z.

[176] Lee, Y., Hwang, S., and Park, J. “Balancing of humanoid robot using contact force / moment control by task-oriented whole body control framework”. In: *Autonomous Robots* 40.3 (2015), pp. 457–472. DOI: 10.1007/s10514-015-9509-1.

[177] Lee, Y., Lee, H., Hwang, S., and Park, J. “Terrain edge detection for biped walking robots using active sensing with vCoP-position hybrid control”. In: *Robotics and Autonomous Systems* 96.June (2017), pp. 41–57. DOI: 10.1016/j.robot.2017.05.011.

[178] Lengagne, S., Vaillant, J., Yoshida, E., and Kheddar, A. “Generation of whole-body optimal dynamic multi-contact motions”. In: *International Journal of Robotics Research* 32.9-10 (2013), pp. 1104–1119. DOI: 10.1177/0278364913478990.

[179] Li, Z., Vanderborght, B., Tsagarakis, N. G., Colasanto, L., and Caldwell, D. G. “Stabilization for the compliant humanoid robot COMAN exploiting intrinsic and controlled compliance”. In: *Proceedings IEEE International Conference on Robotics and Automation*. 2012, pp. 2000–2006. DOI: 10.1109/ICRA.2012.6224705.

[180] Liégeois, A. “Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 7.12 (1977), pp. 868–871. DOI: 10.1109/TSMC.1977.4309644.

[181] Lim, H. O., Setiawan, S. A., and Takanishi, A. “Balance and Impedance Control for Biped Humanoid Robot Locomotion”. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2001, pp. 494–499. DOI: 10.1109/IROS.2001.973405.

[182] Liu, R., Yang, B., Zio, E., and Chen, X. “Artificial intelligence for fault diagnosis of rotating machinery: A review”. In: *Mechanical Systems and Signal Processing* 108 (2018), pp. 33–47. DOI: 10.1016/j.ymssp.2018.02.016.

[183] Ljung, L. *System Identification: Theory for the User*. Prentice Hall information and system sciences series. Prentice Hall PTR, 1999. ISBN: 978-0-13-656695-3.

[184] Lochner, E. "Closed-Loop Identifikation von Regelstrecken eines humanoiden Laufroboters". Semester Thesis (unpublished). Technical University of Munich, 2018.

[185] Löffler, K. *Dynamik und Regelung einer zweibeinigen Laufmaschine*. Fortschrittberichte VDI, Reihe 8 1094. Düsseldorf: VDI-Verlag, 2006. ISBN: 3-18-509408-5.

[186] Löffler, K., Gienger, M., and Pfeiffer, F. "Model Based Control of a Biped Robot". In: *Proceedings 7th International Workshop on Advanced Motion Control*. 2002, pp. 443–448. DOI: 10.1109/AMC.2002.1026961.

[187] Löffler, K., Gienger, M., Pfeiffer, F., and Ulbrich, H. "Sensors and Control Concept of a Biped Robot". In: *IEEE Transactions on Industrial Electronics* 51.5 (2004), pp. 972–980. DOI: 10.1109/TIE.2004.834948.

[188] Lohmeier, S. "Design and Realization of a Humanoid Robot for Fast and Autonomous Bipedal Locomotion". Dissertation. Technical University of Munich, 2010. ISBN: 978-3-86853-734-5. URL: https://mediatum.ub.tum.de/980754.

[189] Lohmeier, S., Buschmann, T., and Ulbrich, H. "System design and control of anthropomorphic walking Robot LOLA". In: *IEEE/ASME Transactions on Mechatronics* 14.6 (2009), pp. 658–666. DOI: 10.1109/TMECH.2009.2032079.

[190] Lunze, J. *Regelungstechnik 2*. Springer-Verlag Berlin Heidelberg, 2016. ISBN: 978-3-662-52676-7.

[191] Magazino. *The Warehouse Robot Toru*. URL: https://www.magazino.eu/products/toru/?lang=en (visited on 03/29/2021).

[192] Mansard, N., Remazeilles, A., and Chaumette, F. "Continuity of varying-feature-set control laws". In: *IEEE Transactions on Automatic Control* 54.11 (2009), pp. 2493–2505. DOI: 10.1109/TAC.2009.2031202.

[193] Martínez-Rego, D., Fontenla-Romero, O., and Alonso-Betanzos, A. "Power wind mill fault detection via one-class ν-SVM vibration signal analysis". In: *Proceedings of the International Joint Conference on Neural Networks* May 2016 (2011), pp. 511–518. DOI: 10.1109/IJCNN.2011.6033264.

[194] Mattingley, J. and Boyd, S. "CVXGEN: A code generator for embedded convex optimization". In: *Optimization and Engineering* 13.1 (2012), pp. 1–27. DOI: 10.1007/s11081-011-9176-9. URL: www.cvxgen.com.

[195] Mesesan, G., Englsberger, J., Garofalo, G., Ott, C., and Albu-Schaffer, A. "Dynamic Walking on Compliant and Uneven Terrain using DCM and Passivity-based Whole-body Control". In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2019, pp. 25–32. DOI: 10.1109/Humanoids43949.2019.9035053.

[196] Mesesan, G., Englsberger, J., Henze, B., and Ott, C. "Dynamic multi-contact transitions for humanoid robots using Divergent Component of Motion". In: *Proceedings IEEE International Conference on Robotics and Automation*. 2017, pp. 4108–4115. DOI: 10.1109/ICRA.2017.7989473.

[197] Metta, G., Fitzpatrick, P., and Natale, L. "YARP: Yet another robot platform". In: *International Journal of Advanced Robotic Systems* 3.1 (2006), pp. 043–048. DOI: 10.5772/5761.

[198] Metta, G. et al. "The iCub humanoid robot: An open-systems platform for research in cognitive development". In: *Neural Networks* 23.8-9 (2010), pp. 1125–1134. DOI: 10.1016/j.neunet.2010.08.010.

[199] Mittendorfer, P., Dean, E., and Cheng, G. “3D Spatial Self-organization of a Modular Artificial Skin”. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2014, pp. 3969–3974. DOI: 10.1109/IROS.2014.6943120.

[200] Morais, J. P., Georgiev, S., and Sprößig, W. *Real quaternionic calculus handbook*. 2014. ISBN: 978-3-0348-0622-0. DOI: 10.1007/978-3-0348-0622-0.

[201] Morisawa, M., Kanehiro, F., Kaneko, K., Kajita, S., and Yokoi, K. “Reactive Biped Walking Control for a Collision of a Swinging Foot on Uneven Terrain”. In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2011, pp. 768–773. DOI: 10.1109/Humanoids.2011.6100885.

[202] Morisawa, M. et al. “A biped pattern generation allowing immediate modification of foot placement in real-time”. In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2006, pp. 581–586. DOI: 10.1109/ICHR.2006.321332.

[203] Morisawa, M. et al. “Balance control based on Capture Point error compensation for biped walking on uneven terrain”. In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2012, pp. 734–740. DOI: 10.1109/HUMANOIDS.2012.6651601.

[204] Morisawa, M. et al. “Biped locomotion control for uneven terrain with narrow support region”. In: *Proceedings IEEE/SICE International Symposium on System Integration*. 2014, pp. 34–39. DOI: 10.1109/SII.2014.7028007.

[205] Muratore, L. et al. “XBotCore: A real-time cross-robot software platform”. In: *Proceedings IEEE International Conference on Robotic Computing, IRC*. 2017, pp. 77–80. DOI: 10.1109/IRC.2017.45.

[206] Nakamura, Y. *Advanced Robotics: Redundancy and Optimization*. Addison-Wesley, 1991. ISBN: 978-0-201-15198-5.

[207] Nelson, G., Saunders, A., and Playter, R. “The PETMAN and Atlas Robots at Boston Dynamics”. In: *Humanoid Robotics: A Reference*. Springer Nature B.V., 2019, pp. 169–186. DOI: 10.1007/978-94-007-6046-2_15.

[208] Nelson, G. et al. “PETMAN: A Humanoid Robot for Testing Chemical Protective Clothing”. In: *Journal of the Robotics Society of Japan* 30.4 (2012), pp. 372–377. DOI: 10.7210/jrsj.30.372.

[209] Nguyen, N. T. *Model-Reference Adaptive Control*. Springer International Publishing AG, 2018. ISBN: 978-3-319-56393-0. DOI: 10.1007/978-3-319-56393-0.

[210] Nguyen, Q., Powell, M. J., Katz, B., Carlo, J. D., and Kim, S. “Optimized Jumping on the MIT Cheetah 3 robot”. In: *Proceedings IEEE International Conference on Robotics and Automation*. 2019, pp. 7448–7454. DOI: 10.1109/ICRA.2019.8794449.

[211] Nishiwaki, K. and Kagami, S. “Frequent Walking Pattern Generation that Uses Estimated Actual Posture for Robust Walking Control”. In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2009, pp. 535–541. DOI: 10.1109/ICHR.2009.5379519.

[212] Nishiwaki, K. et al. “Design and development of research platform for perception-action integration in humanoid robot: H6”. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2000, pp. 1559–1564. DOI: 10.1109/IROS.2000.895195.

[213] Nishiwaki, K., Chestnutt, J., and Kagami, S. “Autonomous navigation of a humanoid robot over unknown rough terrain using a laser range sensor”. In: *The International Journal of Robotics Research* 31.11 (2012), pp. 1251–1262. DOI: 10.1177/0278364912455720.

[214] Nishiwaki, K. and Kagami, S. “High Frequency Walking Pattern Generation based on Preview Control of ZMP”. In: *Proceedings IEEE/RAS International Conference on Humanoid Robots*. 2006, pp. 2667–2672. DOI: 10.1109/ROBOT.2006.1642104.

[215] Nishiwaki, K. and Kagami, S. “Walking Control on Uneven Terrain with Short Cycle Pattern Generation”. In: *Proceedings IEEE/RAS International Conference on Humanoid Robots*. 2007, pp. 447–453. DOI: 10.1109/ICHR.2007.4813908.

[216] Nishiwaki, K. and Kagami, S. “Strategies for Adjusting the ZMP Reference Trajectory for Maintaining Balance in Humanoid Walking”. In: *Proceedings IEEE International Conference on Robotics and Automation*. 2010, pp. 4230–4236. DOI: 10.1109/ROBOT.2010.5510002.

[217] Nishiwaki, K. and Kagami, S. “Trajectory Design and Control of Edge-landing Walking of a Humanoid for Higher Adaptability to Rough Terrain”. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2012, pp. 3432–3439. DOI: 10.1109/IROS.2012.6386056.

[218] Nishiwaki, K., Kuffner, J., Kagami, S., Inaba, M., and Inoue, H. “The experimental humanoid robot H7: a research platform for autonomous behaviour.” In: *Philosophical Transactions of the Royal Society A* 365.1850 (2007), pp. 79–107. DOI: 10.1098/rsta.2006.1921.

[219] Ochoa-Dominguez, H. and Rao, K. R. *Discrete Cosine Transform*. Second. Milton: Taylor & Francis Group, 2019. ISBN: 978-1-351-39648-6.

[220] Ogura, Y. et al. “Development of a new humanoid robot WABIAN-2”. In: *Proceedings IEEE International Conference on Robotics and Automation*. 2006, pp. 76–81. DOI: 10.1109/ROBOT.2006.1641164.

[221] Ogura, Y. et al. “Human-like walking with knee stretched, heel-contact and toe-off motion by a humanoid robot”. In: *Proceedings IEEE International Conference on Intelligent Robots and Systems*. 2006, pp. 3976–3981. DOI: 10.1109/IROS.2006.281834.

[222] Ohashi, E., Sato, T., and Ohnishi, K. “Motion Control of Environmental Adaptation for Biped Robot”. In: *Proceedings 2008 International Workshop on Advanced Motion Control*. 2008, pp. 306–311. DOI: 10.1109/AMC.2008.4516084.

[223] Ohmura, Y., Kuniyoshi, Y., and Nagakubo, A. “Conformable and Scalable Tactile Sensor Skin for Curved Surfaces”. In: *Proceedings IEEE International Conference on Automation and Robotics*. 2006, pp. 1348–1353. DOI: 10.1109/ROBOT.2006.1641896.

[224] Otani, T. et al. “Joint mechanism that mimics elastic characteristics in human running”. In: *Machines* 4.1 (2016). DOI: 10.3390/machines4010005.

[225] Otani, T. et al. “Upper-body control and mechanism of humanoids to compensate for angular momentum in the yaw direction based on human running”. In: *Applied Sciences (Switzerland)* 8.1 (2018), pp. 1–16. DOI: 10.3390/app8010044.

[226] Ott, C., Roa, M. A., and Hirzinger, G. “Posture and Balance Control for Biped Robots based on Contact Force Optimization”. In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2011, pp. 26–33. DOI: 10.1109/Humanoids.2011.6100882.

[227] Ott, C. et al. “Development of a biped robot with torque controlled joints”. In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2010, pp. 167–173. DOI: 10.1109/ICHR.2010.5686340.

[228] Pajon, A., Caron, S., De Magistri, G., Miossec, S., and Kheddar, A. “Walking on gravel with soft soles using linear inverted pendulum tracking and reaction force distribution”. In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2017, pp. 432–437. DOI: 10.1109/HUMANOIDS.2017.8246909.

[229] Park, H.-W., Ramezani, A., and Grizzle, J. W. “A Finite-state Machine for Accommodating Unexpected Large Ground Height Variations in Bipedal Robot Walking”. In: *IEEE Transactions on Robotics* 29.2 (2013), pp. 331–345. DOI: 10.1109/TRO.2012.2230992.

[230] Park, H. W., Wensing, P. M., and Kim, S. “High-speed bounding with the MIT Cheetah 2: Control design and experiments”. In: *International Journal of Robotics Research* 36.2 (2017), pp. 167–192. DOI: 10.1177/0278364917694244.

[231] Park, I.-W., Kim, J.-Y., Lee, J., and Oh, J.-H. “Mechanical Design of the Humanoid Robot Platform KHR-3 (KAIST Humanoid Robot - 3: HUBO)”. In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2005, pp. 321–326. DOI: 10.1109/ICHR.2005.1573587.

[232] Park, J. H. “Impedance Control for Biped Robot Locomotion”. In: *IEEE Transactions on Robotics and Automation* 17.6 (2001), pp. 870–882. DOI: 10.1109/70.976014.

[233] Park, J. H. and Chung, H. “Hybrid Control for Biped Robots Using Impedance Control and Computed-Torque Control”. In: *Proceedings IEEE International Conference on Robotics and Automation*. 1999, pp. 1365–1370. DOI: 10.1109/ROBOT.1999.772551.

[234] Park, Y.-L., Chen, B., and Wood, R. J. “Design and Fabrication of Soft Artificial Skin Using Embedded Microchannels and Liquid Conductors”. In: *IEEE Sensors Journal* 12.8 (2012), pp. 2711–2718. DOI: 10.1109/JSEN.2012.2200790.

[235] Pedregosa, F. et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830. URL: https://scikit-learn.org.

[236] Pfeiffer, F. “The TUM walking machines”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 365.1850 (2007), pp. 109–131. DOI: 10.1098/rsta.2006.1922.

[237] Pölzleitner, D. “Experimentelle Untersuchung von Halbleiter-Dehnungsmessstreifen für den Einsatz in Kraftsensoren eines humanoiden Roboters”. Bachelor’s Thesis. Technical University of Munich, 2020. URL: https://mediatum.ub.tum.de/1593945.

[238] Popovic, M. B., Goswami, A., and Herr, H. “Ground reference points in legged locomotion: Definitions, biological trajectories and control implications”. In: *International Journal of Robotics Research* 24.12 (2005), pp. 1013–1032. DOI: 10.1177/0278364905058363.

[239] Praegla, P. “Walking on Soft Ground: Online Ground Stiffness Estimation for Humanoid Robots”. Semester Thesis (unpublished). Technical University of Munich, 2019.

[240] Pratt, J. E. and Tedrake, R. “Velocity-based stability margins for fast bipedal walking”. In: *Lecture Notes in Control and Information Sciences* 340 (2006), pp. 299–324. DOI: 10.1007/978-3-540-36119-0_14.

[241] Pratt, J., Carff, J., Drakunov, S., and Goswami, A. “Capture Point: A Step toward Humanoid Push Recovery”. In: *Proceedings IEEE/RAS International Conference on Humanoid Robots*. 2006, pp. 200–207. DOI: 10.1109/ICHR.2006.321385.

[242] Pratt, J. E. et al. “The yobotics-IHMC lower body humanoid robot”. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 410–411. DOI: 10.1109/IROS.2009.5354430.

[243] Preumont, A. *Vibration control of active structures*. 1st ed. Springer Netherlands, 1997. ISBN: 978-94-010-6385-2. DOI: 10.1007/978-94-011-5654-7.

[244] Quigley, M. et al. "ROS: an open-source Robot Operating System". In: *Proceedings ICRA Workshop on Open Source Software*. 2009. URL: http://www.willowgarage.com/papers/ros-open-source-robot-operating-system.

[245] Raibert, M. H., Blankespoor, K., Nelson, G., and Playter, R. "BigDog, the Rough-Terrain Quadruped Robot". In: *Proceedings Proc. of the 17th World Congress The Intl Federation of Automatic Control*. 2008, pp. 10822–10825. DOI: 10.3182/20080706-5-KR-1001.01833.

[246] Raibert, M. H. *Legged Robots that Balance*. Cambridge, Massachusetts: The MIT Press, 1986. ISBN: 0-262-18117-7.

[247] Raibert, M. H. "Trotting, pacing and bounding by a quadruped robot". In: *Journal of Biomechanics* 23.SUPPL. 1 (1990), pp. 79–98. DOI: 10.1016/0021-9290(90)90043-3.

[248] Raibert, M. H., Chepponis, M., and Brown, H. B. "Running on Four Legs As Though They Were One". In: *IEEE Journal on Robotics and Automation* 2.2 (1986), pp. 70–82. DOI: 10.1109/JRA.1986.1087044.

[249] Rainbow Robotics. *Robot Products*. URL: http://www.rainbow-robotics.com/new/index.php (visited on 03/29/2021).

[250] Rainer, J. "Experimentelle Untersuchung und Modellierung des Sensorverhaltens einer taktilen Fußsohle für humanoide Roboter". Bachelor's Thesis. Technical University of Munich, 2020. URL: https://mediatum.ub.tum.de/1591539.

[251] Roberson, R. E. and Schwertassek, R. *Dynamics of Multibody Systems*. 1st ed. Springer-Verlag Berlin Heidelberg, 1988. ISBN: 978-3-642-86466-7. DOI: 10.1007/978-3-642-86464-3.

[252] Rogelio Guadarrama Olvera, J., Leon, E. D., Bergner, F., and Cheng, G. "Plantar Tactile Feedback for Biped Balance and Locomotion on Unknown Terrain". In: *International Journal of Humanoid Robotics* 17.1 (2020), pp. 1–21. DOI: 10.1142/S0219843619500361.

[253] Sakagami, Y. et al. "The intelligent ASIMO: System overview and integration". In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2002, pp. 2478–2483. DOI: 10.1109/IRDS.2002.1041641.

[254] Samadi, S., Caron, S., Tanguy, A., and Kheddar, A. "Balance of Humanoid Robots in a Mix of Fixed and Sliding Multi-Contact Scenarios". In: *Proceedings IEEE International Conference on Robotics and Automation*. 1. 2020, pp. 6590–6596. DOI: 10.1109/ICRA40945.2020.9197253.

[255] Santos, P., Villa, L. F., Reñones, A., Bustillo, A., and Maudes, J. "An SVM-based solution for fault detection in wind turbines". In: *Sensors (Switzerland)* 15.3 (2015), pp. 5627–5648. DOI: 10.3390/s150305627.

[256] Sardain, P. and Bessonnet, G. "Forces Acting on a Biped Robot. Center of Pressure and Zero Moment Point". In: *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 34.5 (2004), pp. 630–637. DOI: 10.1109/TSMCA.2004.832811.

[257] Sato, T., Sakaino, S., and Ohnishi, K. "Switching Impedance Control in Vertical Direction for Biped Robot with Toe and Heel Joints on Uneven Terrain". In: *Proceedings 2010 Industrial Electronics Conference*. 2010. DOI: 10.1109/IECON.2010.5675437.

[258] Sattler, M. “Sensor-Based Adaptation of Foot Trajectories for a Bipedal Robot”. Master’s Thesis (unpublished). Technical University of Munich, 2018.

[259] Schölkopf, B. and Smola, A. J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, Massachusetts: MIT Press, 2002. ISBN: 978-0-262-25693-3.

[260] Schroeder, M. “Synthesis of low-peak-factor signals and binary sequences with low autocorrelation”. In: *IEEE Transactions on Information Theory* 16.1 (1970), pp. 85–89. DOI: 10.1109/TIT.1970.1054411.

[261] Schuetz, C., Pfaff, J., Sygulla, F., Rixen, D., and Ulbrich, H. “Motion Planning for Redundant Manipulators in Uncertain Environments based on Tactile Feedback”. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2015, pp. 6387–6394. DOI: 10.1109/IROS.2015.7354290.

[262] Schürmann, C., Koiva, R., Haschke, R., and Ritter, H. “A modular high-speed tactile sensor for human manipulation research”. In: *Proceedings IEEE World Haptics Conference*. 2011, pp. 339–334. DOI: 10.1109/WHC.2011.5945509.

[263] Schwienbacher, M. “Efficient Algorithms for Biped Robots”. Dissertation. Technical University of Munich, 2013. URL: http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20140623-1175522-0-6.

[264] Schwienbacher, M., Buschmann, T., Lohmeier, S., Favot, V., and Ulbrich, H. “Self-collision avoidance and angular momentum compensation for a biped humanoid robot”. In: *Proceedings IEEE International Conference on Robotics and Automation*. 2011, pp. 581–586. DOI: 10.1109/ICRA.2011.5980350.

[265] Seiwald, P. and Sygulla, F. *broccoli: Beautiful Robot C++ Code Library*. 2019. URL: https://gitlab.lrz.de/AM/broccoli.

[266] Seiwald, P., Sygulla, F., Staufenberg, N. S., and Rixen, D. “Quintic Spline Collocation for Real-Time Biped Walking-Pattern Generation with variable Torso Height”. In: *Proceedings IEEE/RAS International Conference on Humanoid Robots*. 2019, pp. 56–63. DOI: 10.1109/Humanoids43949.2019.9035076.

[267] Seiwald, P. et al. “LOLA v1.1 - An Upgrade in Hardware and Software Design for Dynamic Multi-Contact Locomotion”. In: *Proceedings IEEE/RAS International Conference on Humanoid Robots*. 2021, pp. 9–16. DOI: 10.1109/HUMANOIDS47582.2021.9555790.

[268] Semini, C. et al. “Design of HyQ -A hydraulically and electrically actuated quadruped robot”. In: *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering* 225.6 (2011), pp. 831–849. DOI: 10.1177/0959651811402275.

[269] Seo, S. et al. “Flexible Touch Sensors Made of Two Layers of Printed Conductive Flexible Adhesives”. In: *Sensors (Switzerland)* 16.9 (2016), pp. 1–13. DOI: 10.3390/s16091515.

[270] *Sercos III Fieldbus*. URL: https://www.sercos.org/technology/ (visited on 03/29/2021).

[271] Shimojo, M., Namiki, A., Ishikawa, M., Makino, R., and Mabuchi, K. “A Tactile Sensor Sheet Using Pressure Conductive Rubber With Electrical-Wires Stitched Method”. In: *IEEE Sensors Journal* 4.5 (2004), pp. 589–596. DOI: 10.1109/JSEN.2004.833152.

[272] Shin, H., Ishikawa, T., Kamioka, T., Hosoda, K., and Yoshiike, T. “Mechanistic Properties of Five-bar Parallel Mechanism for Leg Structure Based on Spring Loaded Inverted Pendulum”. In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2019, pp. 320–327. DOI: 10.1109/Humanoids43949.2019.9035071.

[273] Shuster, M. “The kinematic equation for the rotation vector”. In: *IEEE Transactions on Aerospace and Electronic Systems* 29.1 (1993), pp. 263–267. DOI: 10.1109/7.249140.

[274] Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. *Robotics: Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing Series. Springer, 2008. ISBN: 978-1-84996-634-4.

[275] Siciliano, B. and Slotine, J.-J. “A general framework for managing multiple tasks in highly redundant robotic systems”. In: *Proceedings Fifth International Conference on Advanced Robotics 'Robots in Unstructured Environments*. 1991, pp. 1211–1216. DOI: 10.1109/ICAR.1991.240390.

[276] Softbank Robotics. *The Service Robot Pepper*. URL: https://www.softbankrobotics.com/emea/en/pepper (visited on 03/29/2021).

[277] Son, B. G., Kim, J. T., and Park, J. H. “Impedance Control for Biped Robot Walking on Uneven Terrain”. In: *Proceedings IEEE International Conference on Robotics and Biomimetics*. 2009. DOI: 10.1109/ROBIO.2009.5420645.

[278] Sorge, K., Rossmann, T., Weidemann, H.-J., Funk, K., and Buschmann, T. *matvec*. Chair of Applied Mechanics, Technical University of Munich, 2009.

[279] Sreenath, K., Park, H. W., Poulakakis, I., and Grizzle, J. W. “A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on MABEL”. In: *International Journal of Robotics Research* 30.9 (2011), pp. 1170–1193. DOI: 10.1177/0278364910379882.

[280] Stasse, O. et al. “TALOS: A new humanoid research platform targeted for industrial applications”. In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2017, pp. 689–695. DOI: 10.1109/HUMANOIDS.2017.8246947.

[281] Steinwart, I. and Christmann, A. *Support Vector Machines*. New York: Springer, 2008. ISBN: 978-0-387-77241-7.

[282] Stentz, A. et al. “CHIMP, the CMU Highly Intelligent Mobile Platform”. In: *Journal of Field Robotics* 32.2 (2015), pp. 209–228. DOI: 10.1002/rob.21569.

[283] Stephens, B. J. and Atkeson, C. G. “Push Recovery by stepping for humanoid robots with force controlled joints”. In: *Proceedings IEEE/RAS International Conference on Humanoid Robots*. 2010, pp. 52–59. DOI: 10.1109/ICHR.2010.5686288.

[284] Stephens, B. J. and Atkeson, C. G. “Dynamic balance force control for compliant humanoid robots”. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2010, pp. 1248–1255. DOI: 10.1109/IROS.2010.5648837.

[285] Sterr, S. “Entwicklung einer taktilen Fußsohle für humanoide Roboter”. Bachelor’s Thesis. Technical University of Munich, 2020. URL: https://mediatum.ub.tum.de/1388135.

[286] Strohmayr, M. W., Wörn, H., and Hirzinger, G. “The DLR Artificial Skin Step I: Uniting Sensitivity and Collision Tolerance”. In: *Proceedings IEEE International Conference on Robotics and Automation*. 2013, pp. 1012–1018. DOI: 10.1109/ICRA.2013.6630697.

[287] Strohmayr, M., Saal, H., Potdar, A., and Smagt, P. van der. “The DLR Touch Sensor I : A Flexible Tactile Sensor for Robotic Hands based on a Crossed-Wire Approach”. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2010, pp. 897–903. DOI: 10.1109/IROS.2010.5650191.

[288] Suzuki, H. et al. “Torque based stabilization control for torque sensorless humanoid robots”. In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2017, pp. 425–431. DOI: 10.1109/HUMANOIDS.2017.8246908.

[289] Sygulla, F. *NoisyTest Dataset for Lola*. 2020. URL: https://github.com/am-lola/noisytest-data-lola.

[290] Sygulla, F. *NoisyTest: Noise-Based System Test Tool*. 2020. URL: https://github.com/xifle/noisytest.

[291] Sygulla, F., Ellensohn, F., Hildebrandt, A.-C., Wahrmann, D., and Rixen, D. “A Flexible and Low-Cost Tactile Sensor for Robotic Applications”. In: *Proceedings IEEE International Conference on Advanced Intelligent Mechatronics*. 2017, pp. 58–63. DOI: 10.1109/AIM.2017.8013995.

[292] Sygulla, F. and Rixen, D. “A force-control scheme for biped robots to walk over uneven terrain including partial footholds”. In: *International Journal of Advanced Robotic Systems* 17.1 (2020), pp. 1–14. DOI: 10.1177/1729881419897472.

[293] Sygulla, F. and Rixen, D. *Learning the Noise of Failure: Intelligent System Tests for Robots*. 2021. arXiv: 2102.08080 [cs.RO].

[294] Sygulla, F., Schuetz, C., and Rixen, D. “Adaptive motion control in uncertain environments using tactile feedback”. In: *Proceedings IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*. 2016, pp. 1277–1284. DOI: 10.1109/AIM.2016.7576946.

[295] Sygulla, F. et al. “Hybrid Position/Force Control for Biped Robot Stabilization with Integrated Center of Mass Dynamics”. In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2017, pp. 742–748. DOI: 10.1109/HUMANOIDS.2017.8246955.

[296] Sygulla, F. et al. “An EtherCAT-Based Real-Time Control System Architecture for Humanoid Robots”. In: *Proceedings IEEE/RAS International Conference on Automation Science and Engineering*. 2018, pp. 483–490. DOI: 10.1109/COASE.2018.8560532.

[297] Tajima, R., Honda, D., and Suga, K. “Fast Running Experiments Involving a Humanoid Robot”. In: *Proceedings IEEE International Conference on Robotics and Automation*. 2009, pp. 1571–1576. DOI: 10.1109/ROBOT.2009.5152404.

[298] Takahashi, Y., Nishiwaki, K., Kagami, S., Mizoguchi, H., and Inoue, H. “High-speed pressure sensor grid for humanoid robot foot”. In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS* (2005), pp. 1097–1102. DOI: 10.1109/IROS.2005.1545126.

[299] Takenaka, T. “Gait generation system of legged mobile robot”. In: *European Patent* EP1671754A2 (2001).

[300] Takenaka, T., Matsumoto, T., and Yoshiike, T. “Real Time Motion Generation and Control for Biped Robot - 3rd report: Dynamics Error Compensation-”. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2009, pp. 1594–1600. DOI: 10.1109/IROS.2009.5354542.

[301] Takenaka, T., Matsumoto, T., and Yoshiike, T. “Real Time Motion Generation and Control for Biped Robot -1st Report: Walking Gait Pattern Generation-”. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2009, pp. 1084–1091. DOI: 10.1109/IROS.2009.5354662.

[302] Takenaka, T., Matsumoto, T., Yoshiike, T., and Shirokura, S. “Real Time Motion Generation and Control for Biped Robot -2nd Report: Running Gait Pattern Generation-”. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2009, pp. 1084–1091. DOI: 10.1109/IROS.2009.5354662.

[303] Takenaka, T. et al. “Real Time Motion Generation and Control for Biped Robot -4th report: Integrated Balance Control”. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2009, pp. 1601–1608. DOI: 10.1109/IROS.2009.5354522.

[304] The MathWorks. *Matlab tfestimate()*. 2020. URL: https://mathworks.com/help/signal/ref/tfestimate.html.

[305] The Wireshark Community. *Wireshark*. URL: https://wireshark.org.

[306] Tsagarakis, N. G. et al. “WALK-MAN: A High-Performance Humanoid Platform for Realistic Environments”. In: *Journal of Field Robotics* 34.7 (2017), pp. 1225–1259. DOI: 10.1002/rob.21702.

[307] Ulmen, J. and Cutkosky, M. “A robust, Low-Cost and Low-Noise Artificial Skin for Human-Friendly Robots”. In: *Proceedings IEEE International Conference on Robotics and Automation*. 2010, pp. 4836–4841. DOI: 10.1109/ROBOT.2010.5509295.

[308] Um, D., Stankovic, B., Giles, K., Hammond, T., and Lumelsky, V. “A modularized sensitive skin for motion planning in uncertain environments”. In: *Proceedings IEEE International Conference on Robotics and Automation*. Vol. 1. 1998. DOI: 10.1109/ROBOT.1998.676240.

[309] Unitree Robotics. *A1 Quadruped*. 2020. URL: https://www.unitree.com/products/a1 (visited on 04/06/2021).

[310] Urata, J., Nakanishi, Y., Okada, K., and Inaba, M. “Design of High Torque and High Speed Leg Module for High Power Humanoid”. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2010, pp. 4497–4502. DOI: 10.1109/IROS.2010.5649683.

[311] Urata, J. et al. “Online decision of foot placement using singular LQ preview regulation”. In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2011, pp. 13–18. DOI: 10.1109/Humanoids.2011.6100894.

[312] Van Hofslot, B. J., Griffin, R., Bertrand, S., and Pratt, J. “Balancing using vertical center-of-mass motion: A 2-d analysis from model to robot”. In: *IEEE Robotics and Automation Letters* 4.4 (2019), pp. 3247–3254. DOI: 10.1109/LRA.2019.2925303.

[313] Vemuri, A. T., Polycarpou, M. M., and Diakourtis, S. A. “Neural network based fault detection in robotic manipulators”. In: *IEEE Transactions on Robotics and Automation* 14.2 (1998), pp. 342–348. ISSN: 1042296X. DOI: 10.1109/70.681254.

[314] Vukobratović, M. and Borovac, B. “Zero-Moment Point — Thirty Five Years of Its Life”. In: *International Journal of Humanoid Robotics* 01.01 (2004), pp. 157–173. DOI: 10.1142/S0219843604000083.

[315] Wahrmann, D. et al. “Modifying the estimated ground height to mitigate error effects on bipedal robot walking”. In: *Proceedings IEEE International Conference on Advanced Intelligent Mechatronics*. IEEE, 2017, pp. 1471–1476. DOI: 10.1109/AIM.2017.8014226.

[316] Wahrmann, D. et al. “Time-variable, event-based walking control for biped robots”. In: *International Journal of Advanced Robotic Systems* 15.2 (2018). DOI: 10.1177/1729881418768918.

[317] Wahrmann, D. et al. “Vision-Based 3D Modeling of Unknown Dynamic Environments for Real-Time Humanoid Navigation”. In: *International Journal of Humanoid Robotics* 16.1 (2019). DOI: 10.1142/S0219843619500026.

[318] Wahrmann Lockhart, D. “Autonomous Robot Walking in Unknown Scenarios”. Dissertation. Technical University of Munich, 2018. URL: http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20181015-1395259-1-9.

[319] Walas, K., Kanoulas, D., and Kryczka, P. “Terrain classification and locomotion parameters adaptation for humanoid robots using force/torque sensing”. In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2016, pp. 133–140. DOI: 10.1109/HUMANOIDS.2016.7803265.

[320] Weiss, K. and Wörn, H. “Resistive Tactile Sensor Matrices using Inter-Electrode Sampling”. In: *Proceedings IEEE Industrial Electronics Conference*. 2005, pp. 1949–1954. DOI: 10.1109/IECON.2005.1569203.

[321] Werner, A. et al. “Multi-contact planning and control for a torque-controlled humanoid robot”. In: *Proceedings IEEE International Conference on Intelligent Robots and Systems*. 2016, pp. 5708–5715. DOI: 10.1109/IROS.2016.7759840.

[322] Westervelt, E. R., Grizzle, J. W., and Koditschek, D. E. “Hybrid zero dynamics of planar biped walkers”. In: *IEEE Transactions on Automatic Control* 48.1 (2003), pp. 42–56. DOI: 10.1109/TAC.2002.806653.

[323] Wieber, P. B. “Trajectory free linear model predictive control for stable walking in the presence of strong perturbations”. In: *Proceedings Proceedings of the 2006 6th IEEE-RAS International Conference on Humanoid Robots, HUMANOIDS*. 2006, pp. 137–142. DOI: 10.1109/ICHR.2006.321375.

[324] Wiedebach, G. et al. “Walking on Partial Footholds Including Line Contacts with the Humanoid Robot Atlas”. In: *Proceedings 2016 IEEE-RAS International Conference on Humanoid Robots*. 2016, pp. 1312–1319. DOI: 10.1109/HUMANOIDS.2016.7803439.

[325] Wittmann, R. “Robust Walking Robots in Unknown Environments”. Dissertation. Technical University of Munich, 2017. URL: http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20171023-1352929-1-6.

[326] Wittmann, R., Hildebrandt, A.-C., Ewald, A., and Buschmann, T. “An Estimation Model for Footstep Modifications of Biped Robots”. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2014, pp. 2572–2578. DOI: 10.1109/IROS.2014.6942913.

[327] Wittmann, R., Hildebrandt, A.-C., Wahrmann, D., Rixen, D., and Buschmann, T. “Real – Time Nonlinear Model Predictive Footstep Optimization for Biped Robots”. In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2015, pp. 711–717. DOI: 10.1109/HUMANOIDS.2015.7363432.

[328] Wittmann, R. et al. “Model-Based Predictive Bipedal Walking Stabilization”. In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2016, pp. 718–724. DOI: 10.1109/HUMANOIDS.2016.7803353.

[329] Yi, J., Zhu, Q., Xiong, R., and Wu, J. “Vibration suppression based on input shaping for biped walking”. In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2016, pp. 236–241. DOI: 10.1109/HUMANOIDS.2016.7803283.

[330] Yi, J., Zhu, Q., Xiong, R., and Wu, J. “Walking Algorithm of Humanoid Robot on Uneven Terrain with Terrain Estimation”. In: *International Journal of Advanced Robotic Systems* 13.1 (2016). DOI: 10.5772/62245.

[331] Yoo, S. M., Hwang, S. W., Kim, D. H., and Park, J. H. “Biped Robot Walking on Uneven Terrain Using Impedance Control and Terrain Recognition Algorithm”. In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2018, pp. 293–298. DOI: 10.1109/HUMANOIDS.2018.8624996.

[332] Yoshiike, T. et al. “Development of experimental legged robot for inspection and disaster response in plants”. In: *Proceedings IEEE International Conference on Intelligent Robots and Systems*. 2017, pp. 4869–4876. DOI: 10.1109/IROS.2017.8206364.

[333] Zucker, M. et al. “A General-purpose System for Teleoperation of the DRC-HUBO Humanoid Robot”. In: *Journal of Field Robotics* 32.3 (2015), pp. 336–351. DOI: 10.1002/rob.21570.

Co-authored Publications

[16] Berninger, T. F. C., Sygulla, F., Fuderer, S., and Rixen, D. J. “Experimental Analysis of Structural Vibration Problems of a Biped Walking Robot”. In: *Proceedings IEEE International Conference on Robotics and Automation (ICRA)*. 2020, pp. 8726–8731. DOI: 10.1109/ICRA40945.2020.9197282.

[17] Berninger, T. F. C., Seiwald, P., Sygulla, F., and Rixen, D. J. “Evaluating the Mechanical Redesign of a Biped Walking Robot Using Experimental Modal Analysis”. In: *Proceedings Topics in Modal Analysis & Testing, Society for Experimental Mechanics Series*. Ed. by Dilworth B.J., M. M. Vol. 8. 2021. DOI: 10.1007/978-3-030-75996-4_6.

[109] Hildebrandt, A.-C. et al. “Real-time predictive kinematic evaluation and optimization for biped robots”. In: *Proceedings IEEE International Conference on Intelligent Robots and Systems*. 2016, pp. 5789–5796. DOI: 10.1109/IROS.2016.7759852.

[110] Hildebrandt, A. C. et al. “Torso height optimization for bipedal locomotion”. In: *International Journal of Advanced Robotic Systems* 15.5 (2018), pp. 1–11. DOI: 10.1177/1729881418804442.

[112] Hildebrandt, A.-C. et al. “Real-Time Path Planning in Unknown Environments for Bipedal Robots”. In: *IEEE Robotics and Automation Letters* 2.4 (2017), pp. 1856–1863. DOI: 10.1109/lra.2017.2712650.

[113] Hildebrandt, A.-C. et al. “Kinematic optimization for bipedal robots: a framework for real-time collision avoidance”. In: *Journal of Autonomous Robots* 43.5 (2019), pp. 1187–1205. DOI: 10.1007/s10514-018-9789-3.

[114] Hildebrandt, A.-C. et al. “Versatile and robust bipedal walking in unknown environments: real-time collision avoidance and disturbance rejection”. In: *Journal of Autonomous Robots* 43 (2019), pp. 1957–1976. DOI: 10.1007/s10514-019-09838-3.

[261] Schuetz, C., Pfaff, J., Sygulla, F., Rixen, D., and Ulbrich, H. “Motion Planning for Redundant Manipulators in Uncertain Environments based on Tactile Feedback”. In: *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2015, pp. 6387–6394. DOI: 10.1109/IROS.2015.7354290.

[265] Seiwald, P. and Sygulla, F. *broccoli: Beautiful Robot C++ Code Library*. 2019. URL: https://gitlab.lrz.de/AM/broccoli.

[266] Seiwald, P., Sygulla, F., Staufenberg, N. S., and Rixen, D. “Quintic Spline Collocation for Real-Time Biped Walking-Pattern Generation with variable Torso Height”. In: *Proceedings IEEE/RAS International Conference on Humanoid Robots*. 2019, pp. 56–63. DOI: 10.1109/Humanoids43949.2019.9035076.

[267] Seiwald, P. et al. “LOLA v1.1 - An Upgrade in Hardware and Software Design for Dynamic Multi-Contact Locomotion”. In: *Proceedings IEEE/RAS International Conference on Humanoid Robots*. 2021, pp. 9–16. DOI: 10.1109/HUMANOIDS47582.2021.9555790.

[289] Sygulla, F. *NoisyTest Dataset for Lola*. 2020. URL: https://github.com/am-lola/noisytest-data-lola.

[290] Sygulla, F. *NoisyTest: Noise-Based System Test Tool*. 2020. URL: https://github.com/xifle/noisytest.

[291] Sygulla, F., Ellensohn, F., Hildebrandt, A.-C., Wahrmann, D., and Rixen, D. "A Flexible and Low-Cost Tactile Sensor for Robotic Applications". In: *Proceedings IEEE International Conference on Advanced Intelligent Mechatronics*. 2017, pp. 58–63. DOI: 10.1109/AIM.2017.8013995.

[292] Sygulla, F. and Rixen, D. "A force-control scheme for biped robots to walk over uneven terrain including partial footholds". In: *International Journal of Advanced Robotic Systems* 17.1 (2020), pp. 1–14. DOI: 10.1177/1729881419897472.

[293] Sygulla, F. and Rixen, D. *Learning the Noise of Failure: Intelligent System Tests for Robots*. 2021. arXiv: 2102.08080 [cs.RO].

[294] Sygulla, F., Schuetz, C., and Rixen, D. "Adaptive motion control in uncertain environments using tactile feedback". In: *Proceedings IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*. 2016, pp. 1277–1284. DOI: 10.1109/AIM.2016.7576946.

[295] Sygulla, F. et al. "Hybrid Position/Force Control for Biped Robot Stabilization with Integrated Center of Mass Dynamics". In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2017, pp. 742–748. DOI: 10.1109/HUMANOIDS.2017.8246955.

[296] Sygulla, F. et al. "An EtherCAT-Based Real-Time Control System Architecture for Humanoid Robots". In: *Proceedings IEEE/RAS International Conference on Automation Science and Engineering*. 2018, pp. 483–490. DOI: 10.1109/COASE.2018.8560532.

[315] Wahrmann, D. et al. "Modifying the estimated ground height to mitigate error effects on bipedal robot walking". In: *Proceedings IEEE International Conference on Advanced Intelligent Mechatronics*. IEEE, 2017, pp. 1471–1476. DOI: 10.1109/AIM.2017.8014226.

[316] Wahrmann, D. et al. "Time-variable, event-based walking control for biped robots". In: *International Journal of Advanced Robotic Systems* 15.2 (2018). DOI: 10.1177/1729881418768918.

[317] Wahrmann, D. et al. "Vision-Based 3D Modeling of Unknown Dynamic Environments for Real-Time Humanoid Navigation". In: *International Journal of Humanoid Robotics* 16.1 (2019). DOI: 10.1142/S0219843619500026.

[328] Wittmann, R. et al. "Model-Based Predictive Bipedal Walking Stabilization". In: *Proceedings IEEE-RAS International Conference on Humanoid Robots*. 2016, pp. 718–724. DOI: 10.1109/HUMANOIDS.2016.7803353.

Supervised Student Theses

[13] Baumann, O. “Design of a Controller to Stabilize a Biped Robot under Consideration of Structural Dynamics”. Master’s Thesis. Technical University of Munich, 2020. URL: https://mediatum.ub.tum.de/1609231.

[85] Freymann, M. von. “Experimentelle Identifikation von Kontakteigenschaften der Fußsohle eines humanoiden Roboters”. Semester Thesis. Technical University of Munich, 2018. URL: https://mediatum.ub.tum.de/1463742.

[94] Grundner, C. “Whole-Body-Momentum Control: Anwendung auf den humanoiden Roboter LOLA”. Master’s Thesis (unpublished). Technical University of Munich, 2019.

[100] Harder, M. “Experimentelle Untersuchung der charakteristischen Eigenschaften einer taktilen Roboter- Haut”. Bachelor’s Thesis (unpublished). Technical University of Munich, 2016.

[118] Hirschmann, H. “Entwicklung einer taktilen Fußsohle für einen humanoiden Roboter”. Master’s Thesis. Technical University of Munich, 2017. URL: https://mediatum.ub.tum.de/1388135.

[172] Künzel, F. “Konstruktion neuartiger Fußgeometrien für einen humanoiden Laufroboter”. Master’s Thesis. Technical University of Munich, 2017. URL: https://mediatum.ub.tum.de/1378389.

[184] Lochner, E. “Closed-Loop Identifikation von Regelstrecken eines humanoiden Laufroboters”. Semester Thesis (unpublished). Technical University of Munich, 2018.

[237] Pölzleitner, D. “Experimentelle Untersuchung von Halbleiter-Dehnungsmessstreifen für den Einsatz in Kraftsensoren eines humanoiden Roboters”. Bachelor’s Thesis. Technical University of Munich, 2020. URL: https://mediatum.ub.tum.de/1593945.

[239] Praegla, P. “Walking on Soft Ground: Online Ground Stiffness Estimation for Humanoid Robots”. Semester Thesis (unpublished). Technical University of Munich, 2019.

[250] Rainer, J. “Experimentelle Untersuchung und Modellierung des Sensorverhaltens einer taktilen Fußsohle für humanoide Roboter”. Bachelor’s Thesis. Technical University of Munich, 2020. URL: https://mediatum.ub.tum.de/1591539.

[258] Sattler, M. “Sensor-Based Adaptation of Foot Trajectories for a Bipedal Robot”. Master’s Thesis (unpublished). Technical University of Munich, 2018.

[285] Sterr, S. “Entwicklung einer taktilen Fußsohle für humanoide Roboter”. Bachelor’s Thesis. Technical University of Munich, 2020. URL: https://mediatum.ub.tum.de/1388135.