TECHNISCHE UNIVERSITÄT MÜNCHEN
Lehrstuhl für Steuerungs- und Regelungstechnik

Dynamic Assist Functions in Haptic Telepresence

Ulrich Unterhinninghofen

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:	Univ.-Prof. Dr.-Ing. Eckehard Steinbach
Prüfer der Dissertation:	
1.	Univ.-Prof. Dr.-Ing./Univ. Tokio Martin Buss
2.	Univ.-Prof. Dr.-Ing. Michael Zäh

Die Dissertation wurde am 27.10.2009 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 22.03.2010 angenommen.

Foreword

The ideas and results presented in this thesis are the outcome of my research at the Institute of Automatic Control Engineering of the Technische Universität München conducted during the past four years under the supervision of Prof. Martin Buss. It has been an exciting journey from the first vague idea to a theoretically well-founded and fully working system.

I want to express my gratitude to my supervisor and thesis advisor Prof. Martin Buss. I highly appreciate the good balance between the guidance and the freedom he gave me to pursue my research. At his institute, he provided an extraordinarily open working environment, which made working and doing research a pleasure.

The team of the institute was a constant source of help and inspiration for my work. For the technical realization of the project, I could rely on the excellent staff in the mechanical and electrical workshop as well as an always helpful and patient system administrator. Furthermore, this work strongly benefitted from the inputs of colleagues at the Universität der Bundeswehr München, namely Franziska Freyberger and Verena Nitsch.

Many students have helped to make this work a success. I want to thank Guy Berg, Alexandre Jung, Thomas Schauß, Carolina Weber, and Tianguang Zhang for their strong commitment and their contributions to this thesis.

I would like to express my deepest gratitude to my parents Jürgen and Hanni and my elder brother Roland for their invaluable support and encouragement all the way along.

This work has been supported by the German Research Foundation (DFG) within the collaborative research centre SFB453 “High-Fidelity Telepresence and Teleaction”.

München, 2010 Ulrich Unterhinninghofen

Contents

Notations

Conventions

Scalars, Vectors, and Matrices

Scalars are denoted by upper and lower case letters in italic type. *Vectors* are denoted by lower case letters in boldface and are assumed to be column vectors such that $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T$. *Matrices* are denoted by upper case letters in boldface, where X_{ij} represents the element of $\boldsymbol{X}$ at row i and column j.

x	scalar value
$\boldsymbol{x}$	vector
$\boldsymbol{X}$	matrix

Subscripts and Superscripts

x_i	i-th component of vector $\boldsymbol{x}$
$\boldsymbol{x}_i$	i-th vector out of a sequence of vectors $\{\boldsymbol{x}_n\}$
${}^A\boldsymbol{x}$	position $\boldsymbol{x}$ expressed in coordinate frame Σ_A
$\hat{x}$	estimated value of x
$\hat{x}^{(j)}$	j-th estimate of x in an iteration
$\boldsymbol{X}^{-1}$	inverse of matrix $\boldsymbol{X}$
$\boldsymbol{X}^T$	transpose of matrix $\boldsymbol{X}$

Symbols

General

$F, \boldsymbol{F}$	force
$G(\cdot)$	cost function
$h(\cdot)$	model function
$\boldsymbol{I}$	identity matrix
$\boldsymbol{J}$	Jacobian matrix
$x, \boldsymbol{x}$	position
$\|\cdot\|$	Euclidean norm
$\langle\cdot,\cdot\rangle$	Scalar product

Intention Estimation

i	sample index
k	estimation step
n_k	number of data points available at estimation step k
$\boldsymbol{p}$	coefficient vector of polynomial in τ
$\boldsymbol{q}$	coefficient vector of polynomial in t
$\boldsymbol{q}_s, \boldsymbol{q}_f$	starting orientation, final orientation expressed as quaternion
t_s, t_f	starting time, final time
$\boldsymbol{x}_s, \boldsymbol{x}_f$	starting position, final position
T	duration of movement $T = t_f - t_s$
α	step size
τ	relative time $\tau = t - t_f$
$\boldsymbol{\xi}$	state vector (position, velocity, acceleration)

Scene Recognition

$\boldsymbol{e}$	image-space collinearity error
$\boldsymbol{\varepsilon}$	object-space collinearity error
$\boldsymbol{E}$	projection error matrix
n	number of observed points
n_p	number of object points
n_c	number of cameras
$\boldsymbol{p}$	point in object coordinates
$\boldsymbol{q}$	point in world coordinates
$\boldsymbol{r}$	point in camera coordinates
$\boldsymbol{R}$	rotation matrix
$\boldsymbol{t}$	translation vector
$\boldsymbol{v}$	image point coordinates
$\boldsymbol{V}$	line-of-sight projection matrix

Assist Functions

ΔF	guidance force generated by the assist system
$\boldsymbol{x}, \boldsymbol{x}_f$	current position, final position
$\Delta\boldsymbol{x}$	relative position to target
$\Delta\boldsymbol{x}^{\perp}$	lateral component of relative position to target
$\Delta\boldsymbol{x}^{\parallel}$	longitudinal component of relative position to target
$\Delta\boldsymbol{x}_O$	relative position to target on operator side
$\Delta\boldsymbol{x}_T$	relative position to target on teleoperator side
$\Delta\boldsymbol{x}_r$	relative position to target according to the reference trajectory

Acronyms

ANOVA	Analysis of Variance
AOP	Absolute Orientation Problem
BF	Bidirectional Force
BS	Bidirectional Scaling
BTL	Bradley-Terry-Luce Model/Scaling
CC	Communication Channel
CF	Constant Force
CS	Constant Scaling
DOF	Degree of Freedom
GLOH	Gradient Location and Orientation Histogram
HMD	Head Mounted Display
HMM	Hidden Markov Models
HO	Human Operator
HSI	Human System Interface
IPC	Intrinsically Passive Control
JND	Just Noticeable Difference
MANOVA	Multivariate Analysis of Variance
OCE	Object-Space Collinearity Error
PC	Position Correction
PF	Position Dependent Force
PS	Position Dependent Scaling
RE	Remote Environment
RHBP	Rate-Hardness-Based Prediction
SBP	Stiffness-Based Prediction
SIFT	Scale-Invariant Feature Transforms
SNR	Signal-to-Noise Ratio
SURF	Speeded Up Robust Features
SVD	Singular Value Decomposition
TF	Time Dependent Force
TOP	Teleoperator
UF	Unidirectional Force
US	Unidirectional Scaling

Abstract

In this thesis, a human-machine cooperation scheme is presented which increases task performance and feeling of presence in haptic telepresence at the same time. The goal is to assist the operator in telepresent task execution by means of an intelligent teleoperator. Thereby, performance and transparency losses shall be reduced, which are typically induced by unavoidable technical imperfections of the telepresence system.

The approach is based on the insight that current robots can perform simple manipulation tasks more precisely in an autonomous way than a human. This lays the foundation for an effective collaboration between human and robot. While the human operator has the ability to execute tasks in a responsible, flexible, and creative manner, the robot can tirelessly perform precise manipulations. The presented solution does not require the human operator to explicitly trade the control between the robot and himself. Instead, a natural collaboration between human and robot takes place.

Three prerequisites must be fulfilled to allow for effective assistance: Firstly, the manipulation task which the operator intends to perform must be estimated by the teleoperator. Secondly, the teleoperator must have a precise plan to implement the manipulation task. And finally, there must be a method to smoothly combine the motions of the operator with the motions planned by the teleoperator. All these requirements are addressed in the thesis: The intended manipulation task is estimated based on a human operator model. This model interacts with a camera-based identification of the target environment which is used for task planning. Finally, the motions of human operator and teleoperator are fused by a passivity maintaining controller.

Zusammenfassung

Die in dieser Arbeit vorgestellten Verfahren ermöglichen eine Verbesserung der Leistungsfähigkeit und Wirklichkeitsnähe von haptischen Telepräsenzsystemen durch den Einsatz eines Mensch-Maschine-Kooperationsschemas. Das Ziel besteht darin, den Operator durch lokale Intelligenz des Teleoperators in der Durchführung von telepräsenten Arbeiten zu unterstützen. Dadurch sollen die Verluste bzgl. Performanz und Transparenz, die wegen technischer Unzulänglichkeiten des Telepräsenzsystems unvermeidlich sind, ausgeglichen werden.

Der Arbeit liegt die Erkenntnis zugrunde, dass heutige Roboter einfache Manipulationsaufgaben autonom präziser durchführen können als ein Mensch. Dadurch ist die Grundlage für eine effektive Kooperation zwischen Mensch und Roboter gegeben, bei der die menschlichen Fähigkeiten zur verantwortungsvollen, flexiblen und kreativen Aufgabenplanung mit der Fähigkeit des Roboters, präzise Manipulationen ermüdungsfrei durchzuführen, kombiniert werden. Bei der vorgestellten Lösung ist es nicht erforderlich, dass der Mensch diese Aufgabenteilung explizit steuert, sondern es findet eine natürliche Kooperation zwischen Mensch und Roboter statt.

Zur wirksamen Assistenz müssen drei Voraussetzungen erfüllt werden: 1. Die Manipulationsaufgabe, die der Operator durchzuführen beabsichtigt, muss vom Teleoperator geschätzt werden. 2. Der Teleoperator muss einen präzisen Plan zur Durchführung seiner Aufgabe haben. 3. Die Bewegungen des Operators müssen mit den vom Teleoperator geplanten Bewegungen in Einklang gebracht werden können. Diese drei Problemstellungen werden in der vorliegenden Arbeit eingehend betrachtet. Zur Bestimmung der Manipulationsabsicht wird ein modellbasierter Schätzer verwendet. Dieser steht in Wechselwirkung zu einer bildbasierten Erkennung der Zielumgebung, die zur Aufgabenplanung verwendet wird. Schließlich werden die Bewegungen von einem Passivität gewährleistenden Regler fusioniert.

1 Introduction

Despite all progresses in autonomous robotics, still a lot of work which is dangerous, tedious, or has to be performed under awkward working conditions has to be carried out by humans. As long as autonomous robots are not yet sophisticated enough to replace human workers in these domains, *telepresence systems* play an important role to make this work less dangerous and less tedious. Telepresence rests upon assigning a proxy to the human, referred to as the *teleoperator (TOP)*, which performs the work on behalf of the human, referred to as the *human operator (HO)*. In this way, the human is not exposed directly to the dangers of the working site any more, but the teleoperator is instead. Also, works requiring high forces over extended periods of time can be facilitated by scaling forces between human and robot. Furthermore, work can be performed in areas which are inaccessible to the human such as underwater or space missions. Finally, it is also cost-efficient if a human does not need to travel to a certain working site, but a present teleoperator can work on his behalf.

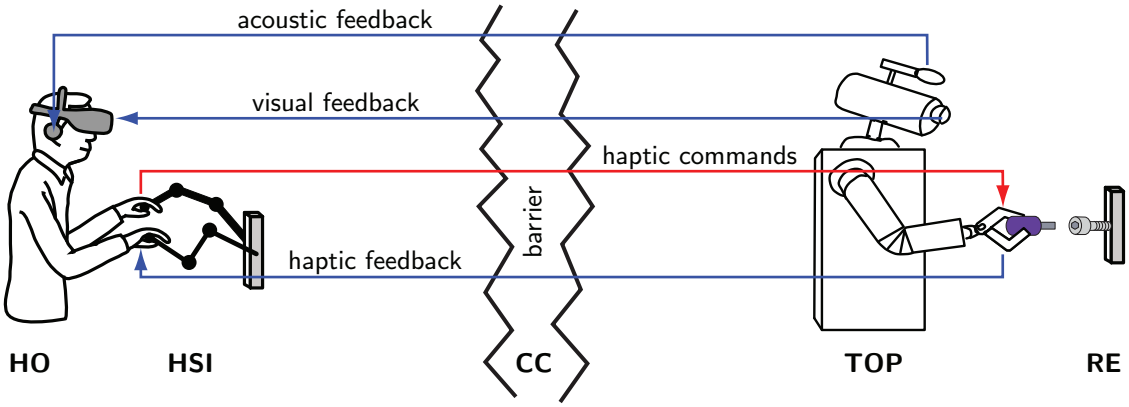

Figure 1.1: A typical multi-modal telepresence system. The human operator (HO) interacts with the human system interface (HSI), which captures motions and conveys visual, auditory, and haptic feedback. The motions are replicated by the teleoperator (TOP), which acts in the remote environment (RE) in place of the HO and acquires the multi-modal feedback signals. Command and feedback signals are exchanged through the communication channel (CC), which bridges the barrier separating HO and RE.

A telepresence system as depicted in Fig. 1.1 consists of three major components: the human system interface, the communication channel, and the teleoperator.

- The *human system interface (HSI)* is the device via which the human operator (HO) interacts with the telepresence system and thus indirectly with the remote environment (RE). It can include a number of different information channels addressing the

various senses of the human being. Typically, visual feedback from the remote environment is presented through a screen or a *head mounted display (HMD)*, acoustic feedback is replayed by use of headphones, and finally physical interaction is enabled by using haptic displays which allow interchanging energy between human operator and remote environment.

- The *communication channel (CC)* is responsible for transferring the digitized information from human operator to teleoperator and back. The information transfer has a certain time delay, which can be constant or time varying depending on the communication architecture. Also, in some architectures, partial loss of information can occur in the communication channel.
- The *teleoperator (TOP)* acts as the representative of the human in the *remote environment (RE)*. It replicates all motions the human operator conveys to the human system interface and transmits sensed information back to the human. Thereby, the telepresence loop is closed between HO and RE over the three components HSI, CC, and TOP.

The ideal telepresence system would convey visual feedback covering the complete field of view at the resolution of the human eyes, give perfect audio feedback, and exactly replicate forces and motions between human operator and remote environment. All information would be instantaneously transferred between the two sites. Ultimately, the human operator would not be able to distinguish between direct and telepresent interaction with the environment. However, none of these criteria can be practically satisfied. Limits are imposed by physical, technical, and – maybe most importantly – economic constraints. Therefore, the advantages of a telepresence system come at the cost of a reduced dexterity and a distorted perception of the remote environment. Depending on the degree of degradation, this can compromise the quality of telepresent work.

In order to overcome the drawbacks of telepresence, the signals from operator to teleoperator and vice versa can be deliberately altered and augmented by a computer. Several such assistance concepts have been proposed, which are supposed to compensate the quality reductions imposed by the system. These architectures vary in the type of modality they address as well as in the level of abstraction on which the assist system intervenes. These design choices also determine which types of reductions (delays, reduced vision, limited *degrees of freedom (DOF)*, etc.) the assist system can compensate best. Also the type of interaction between assist system and human operator differs from very explicit, dialog based concepts to more implicit concepts. Many systems enter a middle-ground between telepresence and telerobotics, i.e. the feeling of being present in the remote environment is often traded for higher task performance.

In Fig. 1.2, the additional elements of an assisted telepresence system as compared to the previously illustrated unassisted telepresence system (cf. Fig. 1.1) are shown. The changes to the interchanged command and feedback signals can be introduced both, on operator and on teleoperator side. If the communication channel between both sides does not impose information loss or time delay, this is an arbitrary choice. In those cases, however, where this condition is not met, the assist functions should be implemented at the side, where the necessary information are more accurate and less delayed. Typically, changes to the command signals are imposed at teleoperator side, where additional information may be

available through additional sensors, such as an eye-in-hand camera. Analogously, changes to the feedback signals, also referred to as augmentation, are more commonly imposed at operator side. In both cases, semi-autonomous control loops are closed at operator and teleoperator side, respectively.

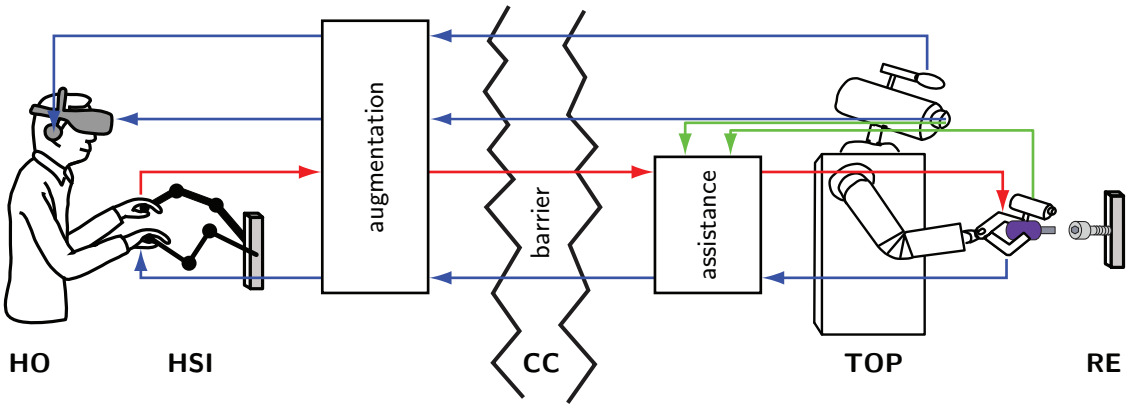

Figure 1.2: Telepresence system with additional components to provide assistance. The complete assist system comprises the augmentation and the assist block. The former is placed on operator side and alters the feedback signals in order to increase the feeling of presence. The latter is placed on teleoperator side and alters the command signals, and optionally the feedback signals, in order to increase the task performance. The assistance block uses additional sensor data, e.g. images from an eye-in-hand camera.

It is desirable to find an assistance concept in which the degree of immersion into the remote environment and the performance of telepresent task execution are optimized at the same time. This requires a thorough understanding of the factors which influence the human's feeling of presence. Furthermore, a good coordination between machine intervention and human planning is required in order to exploit both, the efficiency of human creativity and the precision of a robot. The right concept determines very much whether the human-robot interaction is cooperative or destructive.

1.1 State of the Art

The roots of telepresence systems can be traced back to a purely mechanical telepresence system built by Goertz in the 1940s to handle nuclear material. Although the system does not include any electric or electronic components, the above listed components can be identified with some abstractions: The human system interface consists of a mechanical handle, the communication channel is matched by precise mechanical leverages, and the teleoperator is a mechanical gripper. The visual channel is simply provided by a thick quartz window.

Obviously, the distance which can be bridged by mechanical telepresence systems is limited to a few meters. Consequently, the next generation of telepresence systems uses electric servo motors to mediate positions and forces between human system interface and tele-

operator [90]. The fundamental advantages of electrically controlled telepresence systems triggered a long and intense research leading to fully digital systems with communication over the internet.

1.1.1 Application Areas

The increased flexibility offered by electrical telepresence systems has also increased the range of applications. While telepresence etymologically refers to mediating presence to a distant location, the applicability of the telepresence concept goes far beyond. The barrier which hinders the human operator from performing a task directly is not restricted to distance, but also comprises danger, scale, endurance, accessibility, and others. The most prominent applications associated with these different types of barriers are presented in the following:

- Bridging the *distance* to a remote location can be beneficial for several reasons. For remote locations on earth, it can save time and costs for travel if a person is needed at some distant location. In space missions, the costs, time, risks, and discomfort of travelling to the destination are even much higher, which again makes telepresence an attractive choice.

  Space missions constitute the application in which the spatial distance between operator and teleoperator is most conspicuous. Distances range from approx. 100 km in orbital missions to over 50 million km for missions on the planet Mars. The biggest challenge posed by large distances is the associated time delay. Even for relatively short distances, complex signal relaying and processing leads to large delays. In the ROTEX experiments, a robot was operated from a ground station with a varying round-trip delay of 5-7 s [79]. In order to make telepresence possible despite this delay, shared local autonomy and predictive visual displays were used. In the succeeding ROKVISS experiment, a direct dedicated link reduced the round-trip delay to less than 20 ms [79].

  Counterintuitively, intercontinental teleoperation on earth typically spans larger distances than operation in low earth orbit. The most used communication medium for intercontinental teleoperation is the internet. The transmission over the internet has a time-varying delay and may lead to partial information loss [42]. In [27], a teleoperation experiment between Munich (Germany) and Tokyo (Japan) is presented. At a distance of approx. 5800 km and a roundtrip delay of approx. 280 ms, a bilateral telepresence architecture can be successfully applied.

- In many areas where humans are exposed to *dangerous* situations, telepresence can produce relief by substituting the human by a robot operating on his behalf. Obvious applications can be found in the military domain with remotely controlled reconnaissance and weapon systems. Furthermore, space missions, deep-sea operations, and disaster recovery tasks involve a high-risk for the human operators.

  In [83], telepresence has been successfully applied to deep-sea scenarios. The teleoperator can operate in depths of up to 7000 m and communicates over an acoustic modem. Demining of contaminated areas is an extremely dangerous procedure which

still causes a lot of injuries and fatalities. A telepresence has been tested as demining tool in [55]. This system has been developed further to a versatile system for disaster recovery in [21].

Avoiding dangerous operations is also the driving factor in using telepresence technology for the maintenance of live power lines [9]. This application has been commercialized already by Kyushu Electric Power Co. Another commercial example is the *telerob Explosive Ordnance Disposal and observation robot (tEODor)* by telerob Gesellschaft für Fernhantierungstechnik mbH [2].

- Many objects in the environment are either too small or too large and heavy to be amenable to direct manipulation by a human. Nevertheless, it would be most convenient if these objects could be handled in the same way as objects which can be physically handled by a human. Telepresence can provide the necessary scaling to allow manipulating very small or very large objects in the way humans are used to when treating medium-sized objects.

  Changing the *scale* of operations can be performed in two directions: In micro-assembly, the sizes of the manipulated structures are too small to be directly handled by a human or even to be seen with the unaided eye. Appropriate telepresence system can scale the dimensions of these structures to a level which is comfortable for the human operator [107, 108].

  In contrast, some disaster recovery operations may require to lift objects which are far too big and heavy to be handled by a human helper. Again, an appropriate scaling of positions and forces can make these tasks manageable in a telepresence setup. Examples are the Japanese T-52 Enryu and the US American *Articulated Remote Manipulator System (ARMS)*. As both projects are classified, no detailed information is publicly available.

- A particular type of barrier is present in medical applications. The *accessibility* to the operation area, e.g. the heart or other internal organs, is obstructed by the surrounding tissue and the skin. At this point, telepresence can play an important role. By introducing small robot manipulators through a trocar into the human body, the operation can be performed with much less strain to the patient than in classic open-sky procedures.

  In [74], a prototypical telesurgery system is presented. The teleoperator side consists of a special scalpel and a stereo laparoscope, which are inserted into the body through two separate trocars. The operator side is composed of a haptic input device and a stereoscopic display. The miniature force/torque-sensor in the tip of the scalpel allows force-feedback during the procedure. According to an experimental study, this significantly reduces the risk of injuries.

  Telesurgery is one of the commercially most promising applications of telepresence. Although high demands on safety, high development costs, and liability issues retard the commercialization of telesurgery, there is one very successful system on the market. The da Vinci System by Intuitive Surgical Inc. can be clinically used for a number of cardiac procedures. It comprises two robotic arms and a camera on teleoperator side and two haptic interfaces and a stereoscopic screen on operator side.

While the haptic interface is able to deliver force-feedback, currently no appropriate sensorized surgical tools exist on the market [1].

Of course, many applications are faced with more than one barrier. E.g. space missions take place at a distant location and pose a significant danger. Telesurgery must provide access to the operation area and sometimes needs to scale motions and forces to facilitate the procedure.

1.1.2 Integrated Telepresence

Research on fundamental aspects of telepresence is complemented by research on integrated telepresence systems using state-of-the-art components. In this research, the focus lies on multi-modal systems with no limitations regarding degrees of freedom and workspace. A prototypical example is presented in [21] and partly shown in Fig. 1.3.

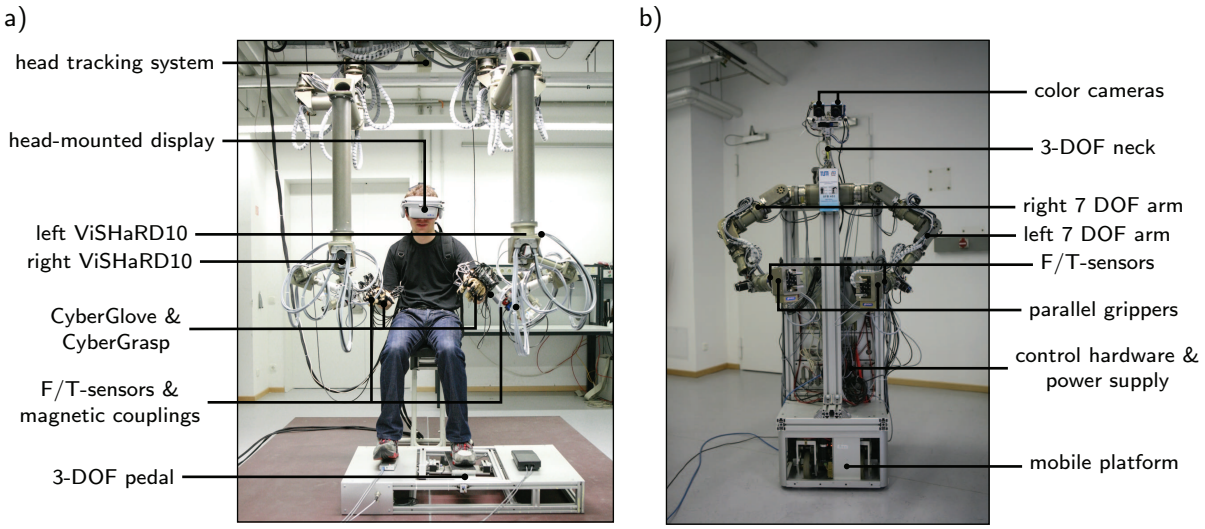

Figure 1.3: An integrated multi-modal telepresence system. The complete system features bimanual, mobile manipulation in six degrees of freedom.

The system consists of a mobile teleoperator and a mobile haptic interface to enable intuitive manipulations in arbitrarily large environments. The teleoperator comprises two anthropomorphically designed arms with grippers, a stereo camera head with microphones, and an omnidirectional mobile base. The human-system interface on operator side equivalently contains two haptic displays, a head-mounted display with headphones, and another omnidirectional mobile base. Additionally, a large-scale tracking system tracks the poses of body and head of the operator.

By appropriately connecting operator and teleoperator side, the complete system enables the operator to freely walk around in the remote environment and to manipulate objects. During the process, he always receives a realistic haptic, visual and auditory feedback.

1.1.3 Assist Systems

Assist systems for telepresence applications have been developed since a long time, and they come in a large variety of different flavors. Typically, they borrow insights from a broad range of different areas, such as telepresence concepts, control theory, stability analysis, image processing, as well as human physiology and psychophysics. The importance of the individual areas heavily depends on the thematic priority of the assist system.

In the following subsections, a classification of assist systems will be developed and relevant previous work will be assigned into this classification. The classification scheme describes both, the *control paradigm* of the underlying telepresence system and the *assistance paradigm*, resting upon this telepresence system. While the control paradigm mainly describes the degree of autonomy and level of abstraction of the system, the assistance paradigm describes how the assistance helps to improve the system performance.

Control Paradigms

It is the nature of a telepresence system that a human operator controls the remotely located robot, the teleoperator. Consequently, commands are sent from the operator to the teleoperator and status feedback is returned in the opposite direction. There are numerous ways in which these commands can be given, mainly differing in the level of abstraction and the degree of robot intelligence required.

There is a continuum of control paradigms ranging from pure bilateral control on one end to fully autonomous operation on the other end [90]. In *bilateral control*, signals are directly interchanged between human and environment through human-system-interface and teleoperator. If a computer is inserted in this information flow, an aided control scheme results, which is typically termed *shared control*. If the teleoperator closes some local control-loops by autonomously reacting to events in the environment, a *supervisory control* scheme is in place, where the human operator gives more general directives which are implemented by the teleoperator. In an even higher degree of autonomy, the teleoperator acts in fully *autonomous control*, once it has received its task assignment, and the human operator can only monitor the execution.

The different control paradigms can be ordered according to many other properties, which are, however, closely associated with the above described continuum. The abstraction level of the commands increases when moving from bilateral control to autonomous operation: While force and velocity signals are usually exchanged in bilateral control, some task primitives in supervisory control, a complete task is assigned in autonomous mode. Accordingly, the update rate in the human-robot control loop decreases with higher autonomy of the robot. Bilateral control is operated at around 1 kHz; in autonomous mode, the execution of a single task may take hours or even much longer. Also, the acceptable time delay is closely related, being roughly proportional to the inverse of the update rate. Lastly, the control paradigms differ in the ratio between the number of robots and the number of humans involved. As the name suggests, this ratio is typically one for bilateral control. The more autonomy the robots have, the less attention of a human operator or supervisor is required, which allows assigning a higher number of robots to a single person.

	Bilateral Control	Shared Control	Supervisory Control	Autonomous Control
Degree of Autonomy	low			high
Abstraction Level	low			high
Update Rate	1 kHz			$\ll$1 Hz
Max. Time Delay	1 ms			$\gg$1 s
Robot:Human Ratio	one			many

Table 1.1: Properties of different control paradigms. Telepresence architectures span a broad spectrum of control paradigms ranging from bilateral control to autonomous control. Depending on the control paradigm the fundamental properties of the telepresence system change.

As illustrated qualitatively in Table 1.1, these properties are typically strongly correlated. Therefore, telepresence systems can be classified according to the above mentioned continuum from pure bilateral control to full autonomous operation.

Assistance Paradigms

Classical, unassisted telepresence systems strive for an exact replication of command and feedback signals on remote side and operator side, respectively. In contrast, assist systems deliberately change these signals in a purposeful manner in order to improve the performance of the telepresence system. As assisted telepresence systems often supply additional information to the operator, they are also referred to as *Augmented Telepresence Systems*.

Assistance paradigms can be classified in different ways. Firstly, the type of interaction between operator and assist system is a major factor. Secondly, the modalities which are involved in the assist systems form an important property. Furthermore, assist systems can be differentiated according to the addressed abstraction level, which is strongly dependent on the control paradigm in use (see above). Finally, they can be grouped by the application or the deficiency they are designed to cure.

The way in which the human operator and the assist system interact can vary from very explicit communication to purely implicit interaction. In the former case, the human operator must instruct the assist system by specific commands via specific input devices to undertake a certain task, and the assist system will give explicit feedback on the status. In the latter case, the assist system must infer from the current task and context, whether assistance is desired and which type of assistance should be employed. While explicit control of the assist system offers richer possibilities and is less prone to errors, implicit control requires less or no training effort and does not charge the operator with the additional load of controlling the assist system.

The three principal modalities, which are transported by typical telepresence systems, namely vision, acoustics, and haptics, are also exploited by assist systems. Typical examples are visual markers, paths, and warnings, as well as acoustic indicators or warnings, and

finally haptic guides such as *virtual fixtures*. For any assistance concept, two modalities are involved: the *augmented modality* and the *augmenting modality*, cf. Table 1.2. When both modalities are the same, the assistance concept is called *intramodal*, otherwise it is called *crossmodal*. Scaling is a simple intramodal assistance concept, which can be applied to all modalities. E.g., if the image is magnified in a micro assembly task, the visual modality is enhanced by visual means. In contrast, visual indicators of excess forces, e.g. by coloring the tool tip, are crossmodal assistance, because an originally haptic property is conveyed to the human through the visual channel. As the technical means to (re-)produce visual and acoustic cues are more advanced than those for haptic cues, crossmodal assist systems usually support the haptic modality by either the visual or the acoustic modality. The most widespread type of intramodal haptic assistance are virtual guides or virtual forbidden regions, where the human operator is assisted in following predefined paths or in not entering predefined regions.

		augmenting modality		
		visual	**acoustic**	**haptic**
augmented modality	**visual**	*scaling (magnification)* Augmented Reality [11] Enhancing Live Video Streams with Virtual Reality [78]		
	acoustic		*scaling (amplification)* Spatial Audio Reproduction [52]	
	haptic	*sensory substitution* Sensory Substitution and Information Redundancy [81]	*sensory substitution* Metaphors [7] Sensory Substitution and Information Redundancy [81]	*scaling (force/velocity)* *virtual guides/fixtures* Virtual Fixtures [4]

Table 1.2: Characteristics and examples of crossmodal and intramodal assistance paradigms. Assistance paradigms are printed in italics; examples are printed in regular font.

As mentioned above, the assist systems are typically strongly dependent on the control paradigm. In a bilateral telepresence system with interchange of position and force data, an assist system will normally change these signals to improve system behavior. In a more abstract system, on the other hand, an assist system will rather propose the most likely task primitives, check the validity of operator commands, etc.

The most prominent application domain of assistance concepts for telepresence is telesurgery. Telesurgery is characterized by highly trained personnel, intensive preparations of medical procedures, and extremely low fault tolerance, which makes it a good candidate for explicit assist systems. Other application examples include grasping and positioning tasks, which occur in many areas. These tasks are hard to execute precisely without assistance, because they require delicate feedback.

Previous Work

Based on the previously presented classification schemes, some major work in the field of assisted telepresence, which is closely related to the work in this thesis, will be classified and discussed in the following.

As mentioned above, the prevailing implementations of intramodal haptic assist functions are based on virtual fixtures. These virtual fixtures constrain the motions of the operator in a purposeful manner. While some constraints can be implemented mechanically, they are usually implemented virtually in the control of the haptic interface [3, 5, 7, 98]. Due to this implementation, which offers a much greater flexibility than the mechanical implementation, virtual fixtures are also termed *active constraints*. Two major types of virtual fixtures have gained a lot of attention in the research: *guidance virtual fixtures* and *forbidden regions virtual fixtures*. Guidance virtual fixtures are designed to force the operator to stay on a predefined path [6] such that he can only control the velocity and the force in tangential direction. In contrast, forbidden regions virtual fixtures have the purpose to protect certain vulnerable areas by forcing the operator to keep away from these zones [4]. Both types of virtual fixtures are prototypical examples of the shared control paradigm, because they keep the operator in charge of the operation while constraining his actions in order to increase accuracy and safety.

The concept of virtual fixtures has been extended from purely kinematic constraints to reshaping the dynamics of the remote tool. This technique, called *virtual tool dynamics*, is very useful for increasing or reducing the inertia and damping of the actual tool [22, 48]. In collaborative telemanipulation tasks, virtual tool dynamics can also be used to define the dynamics of the interaction between the collaborating partners.

For any virtual fixture to work correctly, the guidance paths or forbidden regions must be priorly defined. This has two possible implications: If the virtual fixture is manually defined, the path can be travelled autonomously, and the human operator is only responsible for intervening when unexpected events occur. In contrast, if manual task execution is desired, virtual fixtures must be planned and updated automatically based on the intentions of the operator.

Manual planning of virtual fixtures is most commonly done in high-risk applications such as surgical procedures and space missions. These plans are afterwards executed autonomously, which results in a structure following the supervisory control paradigm, because the robot also performs higher level control tasks while the human only monitors the execution. An example of such a supervisory control scheme is presented in [37].

However, in many applications, it is not possible to define a fixed set of virtual fixtures before the actual teleoperation procedure. For these cases, dynamic virtual fixtures are employed, which change in accordance with the current intention of the operator. The focus of this research lies on the intention recognition. A common tool for distinguishing intentions from a discrete set of alternatives are *hidden markov models (HMM)*, which can be trained to serve as classifiers [62, 63, 106]. If additional parameters describing the current intention must be estimated, specific models of the particular task must be taken into account [23].

Assist functions for telepresence systems have been studied for a variety of application domains. The majority of research on assisted telepresence is directed at medical applications, namely at facilitating surgical procedures. This research spans the full range of control paradigms [24], ranging from bilateral control to autonomous control. In this research field, many assist systems can be implemented either as telepresence systems or as *cooperative manipulators* [4]. For cooperative manipulators, the human operator and the robot physically interact with the same tool. In contrast, in a telepresence system, the operator physically interacts with the haptic interface, and the intervention of the assist system is added to the command and force signals of teleoperator and haptic interface, which leads to a similar behavior, although operator and teleoperator are spatially separated.

Other applications of assisted telepresence systems are nuclear waste handling [75] and space maintenance missions [79]. There are also many general assistance concepts, which are not designed for a specific application. Just as the fundamental telepresence control concepts, they can be adapted to a wide range of application domains and are then evaluated for these domains, e.g. to operations in space and to surgical procedures [80].

The assistance concepts in literature also differ significantly in the scope of their implementation. While many concepts have only been implemented to perform very specific studies, there also exist some fully integrated and versatile systems. A system for grasp assistance, which is based on the shared control paradigm, is presented in [36]. A supervisory control scheme for general manipulation tasks can be found in [54].

The above mentioned assist functions are all employed in the assistance block in Fig. 1.2. Thus, they aim to make the motions more precise. Other concepts have been introduced which are employed in the augmentation block and which are designed to make the feedback more realistic. The most effective methods are based on using local impedance models or on matching high-frequency force components.

In the presence of delays in the communication channel, the forces which occur in reaction to movements in the remote environment are reflected back to the operator with a delay. Besides the stability problems, this has negative effects on the feeling of presence. These negative effects can be reduced if a linearized model of the remote environment is estimated and used locally on operator side to predict the reaction forces [38].

More recently, it has been shown that humans can detect high-frequency force components and use them to judge mechanical properties of the material in contact [60]. Although the human is not able to react to these signals at an equally high rate, these components are, nevertheless, important to provide a realistic sensation. This insight has been exploited for telepresence systems by recording high-frequency force components on the remote side and replaying them on the operator side [56, 57].

1.2 Problem Statement

Concluding from the above reviewed state of the art, telepresence systems can be employed in a wide range of applications, which pose their individual challenges. Accordingly, a large

variety of telepresence architectures exists, which can be tuned to different application scenarios regarding force scaling, velocity scaling, time delays, fidelity, stability, etc. However, most of these architectures are designed from a purely control theoretic point of view, which prevents them from taking knowledge about the characteristics of the human operator, the remote environment, and the current task into account. This knowledge can be exploited to cope with missing, delayed, or disturbed data.

In order to benefit from the above mentioned additional knowledge about the human operator, the remote environment, and the current task, assist functions have been added to telepresence systems. A large part of the telepresence assist systems introduce some level of explicit control. Thus, the operator is charged with the additional load of controlling the assistant itself in addition to the control of the actual manipulation task. Even those variants, which adapt to the current task without operator intervention, show a very obtrusive behavior, which corrupts the feeling of presence. In summary, existing assist systems require dedicated training efforts, which compromises the promise of telepresence to easily transport the abilities of a human to a remote site.

In this thesis, an integrated telepresence system shall be developed which does not reduce the operator's feeling of being present at the remote site and which increases the performance of task execution. Typically, increasing the task performance requires to change the command signals of the teleoperator and entrust the teleoperator itself with some partly autonomous execution of the task. Normally this discrepancy between operator signals and teleoperator actions can lead to a reduction of the perceived realism. However, this rationale is only valid under the assumption of an ideal telepresence system. In a real telepresence system with disturbed and delayed signals, the teleoperator actions may be erroneous, i.e. different from the user inputs, in the first place. In these cases, altering the actions back to the originally intended actions will actually increase the perceived realism. The challenge is, therefore, to recognize the intent of the human operator and understand how he would implement the intent. In a second step, deviations from this intent must be detected and it must be decided between unwanted deviations and voluntary deviations which indicate a changed intention.

1.3 Proposed Method

In this thesis, an assisted telepresence system is developed which increases task performance and feeling of presence at the same time by using local intelligence of the teleoperator. In contrast to existing systems, the assist system operates completely hidden from the human operator, i.e. its presence is only indirectly perceivable through the increased degree of task performance and feeling of presence.

For this endeavor, three major problems must be solved (cf. Fig. 1.4):

- In order to provide assistance, the intelligent telepresence system must have some knowledge of the remote scene. From this knowledge, possible intentions of the human operator can be derived. In addition, the scene recognition can provide an exact reference path to the intentional goal, once it has been selected. The scene recognition is based on the head cameras of the teleoperators and additional sensors,

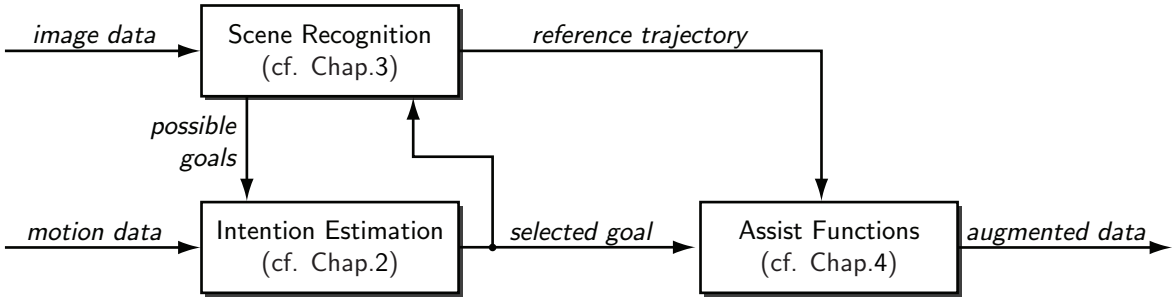

Figure 1.4: Interconnection of the fundamental blocks, which constitute the proposed assisted telepresence system

e.g. an eye-in-hand camera. Therefore, the teleoperator disposes of data, which is more accurate than the data provided to the operator, because it is not disturbed or delayed by the communication channel.

- As the system shall efficiently assist the operator without explicitly prompting for the type of assistance to provide, the teleoperator must silently infer the intention of the human operator from his actions and motions. This intention estimation process is based on possible goals, which have been priorly extracted by the scene recognition process. The intention must be correctly estimated as early as possible to allow effective assistance. This requires an online intention estimation algorithm with fast convergence.
- When the intention of the operator is provided by the intention estimation unit and a reference plan how to graciously implement this intention is available from the scene recognition unit, the assist functions must correct or augment position and force data in such a way that the task performance increases. At the same time, the assist functions must be unobtrusive and preserve a high feeling of presence. Furthermore, the stability of the telepresence system must be guaranteed.

1.4 Contribution and Outline

The proposed method gives rise to a number of scientific questions related to the fields of human-machine interaction, computer vision, telepresence, and psychophysics. These will be considered in detail in the following chapters. Each chapter starts with an introductory overview of pre-existing knowledge in the literature. Based on this foundation, the theory of the new method is described, followed by a performance evaluation. Finally, the results and the generalizable applications of the method are discussed in each chapter.

Intention Estimation

The particular purpose of the intention estimation within the assist system requires the results of the estimation to be available in real-time and as early as possible. This rules out the majority of existing approaches for classification of motion patterns. The new method, which is proposed in Chapter 2, is based on a dynamic model of human point-to-point movements. By fitting the model parameters to the observed motion data, the future course of the motion can be extrapolated. The results can be made more robust by introducing additional hypotheses about the target points. Two model fitting methods are compared w.r.t. their performance, real-time capability, and computational requirements. Numerical considerations regarding the estimation process are given in Appendix A.

Scene Recognition

As the pose of the target in the remote environment must be exactly known in order to enable effective assistance, it must be visually measured when the target object is approached. The concept presented in Chapter 3 uses a multi-camera setup to localize artificial markers on the target object and subsequently derive the pose of the object. The pose estimation algorithm, which calculates the 3D pose of an object based on its 2D projections on camera images, is specifically designed to handle non-overlapping point sets from multiple cameras. It is thus superior to existing techniques based on stereo-matching and is also shown to outperform the single-camera version of the Orthogonal Iteration Algorithm.

Assistance and Augmentation

Assistance and augmentation is achieved by altering the command and feedback signals in an intelligent manner. Existing assist concepts typically concentrate on reducing task execution time and failure rate by using some sort of virtual fixtures. In Chapter 4, a novel method is described, which exploits the data from the intention estimation and scene recognition. The core idea consists in amplifying those operator commands which are adequate for reaching the target and attenuating those which are not. This technique keeps the human in charge of the operation and increases the task performance at the same time.

As the intervention of the assist functions changes the closed-loop behavior of the telepresent control loop, the system can be possibly destabilized by the assist functions. Thus, an appropriate telepresence architecture must be selected, which allows integrating assist functions while still guaranteeing stability. In Appendix B, a review of existing architectures is given, and an architecture based on the port-Hamiltonian approach is proposed.

Experimental Evaluation

The overall performance of human-machine collaborative systems depends on both, the human part and the machine part. Thus, an appropriate evaluation must include human

factors, which can be ensured by a psychophysically well founded experimental design. The relevant psychological background is described in Chapter 5. Subsequently, two experimental studies on the influence of the assist functions on performance and feeling of presence are presented. It is shown that some assist functions can significantly increase performance and feeling of presence at the same time. The study also provides some guidelines on the proper parametrization of the assist functions. The technical specifications of the employed hardware devices are listed in Appendix C.

Assisted telepresence systems are an emerging field with many open directions. This work holds several extension points which are worth to be considered in future research. These are presented in Chapter 6.

2 Intention Estimation

The overall aim of this thesis stipulates that the human operator neither explicitly instructs nor explicitly perceives the assist system. Assist functions are not activated in response to direct instructions, but in response to inferred intentions. Therefore, the intention recognition process becomes an indispensable requirement of the assist system, and its reliability largely affects the reliability of the whole assist system.

In behavioral science, *intention* is defined as "a plan of action the organism chooses and commits itself to in pursuit of a goal" [20]. Thus, it does not only specify the *goal*, which the human strives to achieve, but it also defines the *plan of action*, which is chosen to achieve this goal. The pursuit of an intention, the *intentional action*, is based on three fundamental components: the action goal, the means to purposefully change the environment, and the means to compare the environment with the action goal. These components have their counterparts in a typical controller with reference generator, actuator, and controller [97].

Nature has equipped all higher animals with the ability to infer the intentions of others based on motion cues. This ability significantly contributes to the chances of an individual to survive and to reproduce [12]. Among all species, humans have the most highly developed mechanisms to recognize the intentions of other humans, to harmonize their intentions, and to create common action plans. This ability, referred to as *shared intentionality*, is crucial for the social behavior of humans. For intention recognition, humans mainly rely on visually perceived motions of the observed subject. From motion cues alone, they can deduce properties such as sex, emotional state, action plans, etc. [17].

The most common tool for computer based intention recognition are *Hidden Markov Models (HMM)* [63, 96, 106]. HMM are typically used to assign observed motion patterns to previously learned intention classes. While HMM show an impressive learning and classification performance, they normally provide a classification result only after the intentional goal has already been accomplished. This is not satisfactory for assistance applications, which require the intention to be known while the action plan is still being executed [96]. Furthermore, HMM are better suited for classifying intentions into specific categories than for extracting parameters such as execution speed, trajectories, etc. from the observed actions.

In this thesis, the focus lies on peg-in-hole tasks, which are part of most manipulation tasks and are thus frequently used in telepresence. The application example, which is used throughout this thesis, is inspired by [77] and consists in inserting a hexagon screwdriver into a hexagon socket. As there is only one class of possible intentions considered, the intention classification can be omitted. The intention recognition process is, therefore, restricted to determining the goal, i.e. the target of the peg-in-hole task, and the action plan, i.e. the trajectory to the target. This *intention estimation* problem is illustrated in Fig. 2.1. When the screwdriver has travelled from point 1 to point 2, the most likely target

(A, B, or C) must be selected, and the future trajectory must be predicted based on the trajectory travelled up to this point. In a more general setting, a classifier must be used in the first step to determine the class of intention, which is currently being pursued, and in the second step, an estimator, which is specifically designed for this class of intention, can be applied to extract parameters.

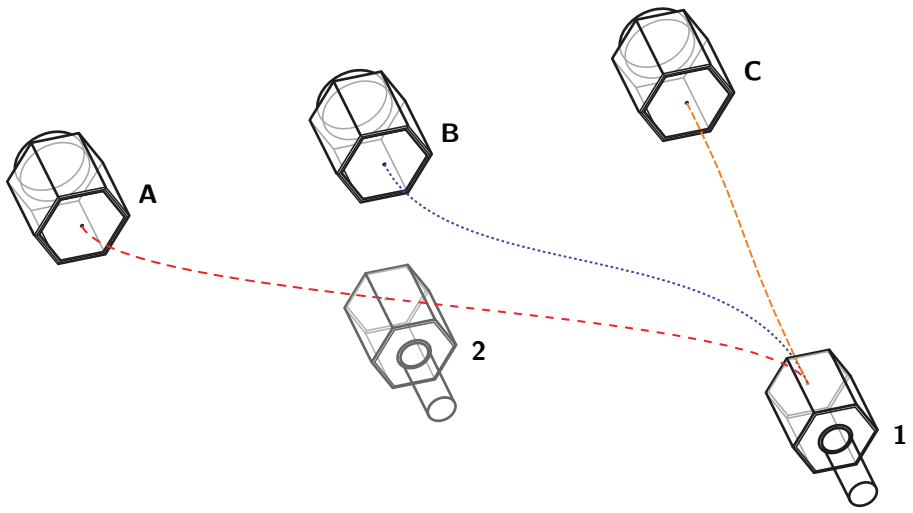

Figure 2.1: Intention estimation for a peg-in-hole task. When the operator has moved the screwdriver from point 1 to point 2, it can be deduced, whether his most likely target is socket A, B, or C. This is done by comparing the travelled trajectory to the ideal trajectories from the starting point to each of the target points.

While humans can effortlessly extract the motion cues which are required for intention estimation from visual information, computer vision which is able to extract these motion cues is still very complex and requires a lot of computational power. As the motions of the human operator must be acquired in a haptic telepresence system anyway, it is far easier to feed the intention estimation process directly with haptic data instead of visual data.

As described above, peg-in-hole tasks are an important part of most manipulation tasks. The peg-in-hole task itself can be decomposed into three phases: approach, alignment, and insertion. In the approach phase, the tool is moved toward the workpiece such that velocity and acceleration are approximately zero when the target area is reached. In the alignment phase, the tool is fine positioned and oriented in preparation for the actual insertion. Finally, in the insertion phase the tool is slid into the workpiece. In a typical execution, these phases overlap such that the tool alignment starts during the final approach phase and the insertion phase begins while the alignment is not fully completed. Therefore, it is reasonable to distinguish between freespace and contact phase, which are clearly separated.

The process of intention estimation for peg-in-hole tasks can be mathematically formulated as model fitting problem. Given a mathematical model of human arm movements, the sampled positions of a partial trajectory, and a number of possible target points of the peg-in-hole task, find the model parameters such that the calculated trajectory best matches the sampled trajectory and ends in one of the possible target points. These model parameters directly yield the extrapolated trajectory to the target point as well as the remaining

time to the target. Obviously, the quality of the estimates will be the better the more information is available. Therefore, the intention estimation process must be repeatedly applied to the sampled trajectory data, and thus it must be real-time capable.

The remainder of this chapter is structured as follows: in Sec. 2.1 a model of human motion planning for reaching movements is presented, which forms the basis of the intention estimation process. Sec. 2.2 is devoted to a trajectory generator based on that human motion model. Two different methods for online intention estimation are presented in Sec. 2.3. A discussion of the presented methods is given in Sec. 2.4, respectively.

2.1 Human Motion Planning

In general, human motion planning is concerned with the underlying principles of trajectory and torque generation in human movements. The most active areas are motion planning for humanoid walking, i.e. movement of legs, and for manipulation, i.e. movement of arms. Here, only arm movements are of interest, and the following considerations are restricted to a very specific class of movements, namely point-to-point movements.

There is a large variety of models for human arm movements [44, 51]. However, almost all of them rely on optimizing some criterion, and the differences between them are produced by different optimality criteria. An important class of models optimizes the mean square jerk of a trajectory, where jerk denotes the third-order time derivative of the position. This criterion can be either applied to coordinates in Cartesian space [31] or to coordinates in joint space [92]. Both versions have in common that they have an analytical solution and that the calculated trajectories are invariant to external dynamics. More complex models take the dynamics of the human arm into account, e.g. the class of minimum torque change model [72, 100]. These models are characterized by high prediction accuracy even with external interaction forces. However, they need human arm models which are non-linear and human dependent. Many other optimization criteria have been proposed such as minimum-power, minimum-acceleration, etc.

2.1.1 Modeling Point-to-Point Movements

The selected model shall be used in two ways: On the one hand, it must be able to generate a humanlike trajectory from a given start pose and end pose. On the other hand, it must be possible to determine the most likely end-pose given a part of the trajectory. Below, the mathematical foundations and the properties of the model are described.

Translational Movements

As mentioned above, there are infinitely many trajectories to implement a motion from one point $\boldsymbol{x}_s$ to another point $\boldsymbol{x}_f$ in space. On the one hand, the spatial course connecting these two points can have an arbitrary shape. On the other hand, the time course of the motion on this curve can have different characteristics. The chosen optimization

criterion determines the unique trajectory which best matches this criterion. From the many optimization criteria which are reviewed above, the minimum hand-jerk criterion [31] is selected, which constitutes a good compromise between accuracy and mathematical simplicity.

For a given trajectory $\boldsymbol{x}(t)$, $t_s \leq t \leq t_f$ lasting from starting time t_s to final time t_f, the minimum-jerk criterion can be expressed by integrating the square of the magnitude of jerk over the whole duration of the operation:

$$G = \frac{1}{2} \int_{t_s}^{t_f} \left(\frac{d^3\boldsymbol{x}}{dt^3} \right)^T \left(\frac{d^3\boldsymbol{x}}{dt^3} \right) dt \tag{2.1}$$

The criterion in (2.1) can be solved for given start time t_s, final time t_f, and boundary conditions for $\boldsymbol{x}_s = \boldsymbol{x}(t_s)$ and $\boldsymbol{x}_f = \boldsymbol{x}(t_f)$. The solution to this optimization problem is presented in [31] by use of dynamic optimization theory. It yields a fifth-order polynomial. For a point-to-point movement, where velocities and accelerations at start and final point are zero, the polynomial takes the following form:

$$\boldsymbol{x} = \boldsymbol{x}_f + (\boldsymbol{x}_f - \boldsymbol{x}_s) \left(6\frac{\tau^5}{T^5} + 15\frac{\tau^4}{T^4} + 10\frac{\tau^3}{T^3} \right) \tag{2.2}$$

where $\tau = t - t_f$ and $T = t_f - t_s$.

Some characteristics of the minimum-jerk-based optimization are depicted in Fig. 2.2. It produces straight lines for point-to-point movements, which is a good approximation to human arm movements, although the latter show a slightly curved path. The velocity profile is bell-shaped with a maximum in the middle between start point and end point. The acceleration profile is symmetric. Position, velocity, and acceleration do not show any discontinuities.

It is shown in [82] that the match between human motions and minimum jerk trajectories is superior to those created by optimization criteria which use higher time derivatives of the position, e.g. minimum snap, minimum crackle, etc. The minimum-jerk criterion is validated only for planar motions and motions which do not exhibit high velocities or accelerations. As telepresent manipulations are typically performed at low speeds, this does not pose a significant constraint.

Rotational Movements

Mathematically, the above described model for translational movements can be transferred to rotational movements by using quaternion interpolation. Although this approach is not validated to correctly reproduce the human behavior, it is used in this thesis for trajectory generation because it produces smooth rotational trajectories. In the practical evaluation, no evidence is found that these motion patterns are disturbing to the human operator.

According to Euler's theorem on finite rotation, an arbitrary orientation in space can be transferred to any other orientation in space by a single rotation along a certain axis over

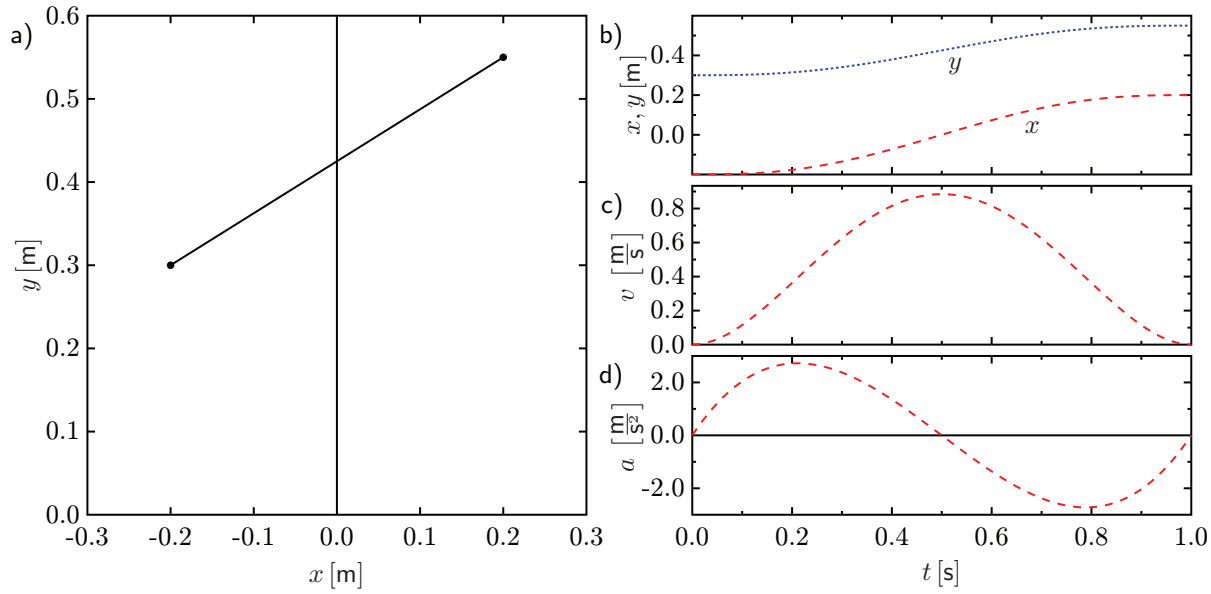

Figure 2.2: Properties of minimum-jerk trajectories. The spatial and the temporal course of a point-to-point movement with minimum-jerk characteristics are shown. a) The spatial course of the trajectory in the x-y plane is a line connecting starting point and endpoint. b) The temporal course of the x and y coordinates smoothly blends from starting point to endpoint with a point symmetric profile. c) The bell-shaped profile of velocity v directed from starting point to endpoint. d) The point symmetric profile of acceleration a directed from start point to endpoint.

a certain angle. When moving from a start orientation $\boldsymbol{q}_s$ to a target orientation $\boldsymbol{q}_f$, it is thus desired that the rotation axis remains constant. The complexity of this interpolation depends on the selected orientation representation. Quaternions provide a simple way of interpolating between two orientations by means of the slerp function [91]:

$$\boldsymbol{q} = \text{slerp}\left(\boldsymbol{q}_s, \boldsymbol{q}_f; 6\frac{\tau^5}{T^5} + 15\frac{\tau^4}{T^4} + 10\frac{\tau^3}{T^3} + 1\right). \tag{2.3}$$

The slerp functions is defined as:

$$\text{slerp}(\boldsymbol{q}_0, \boldsymbol{q}_1; r) = \frac{\sin(1-r)\theta}{\sin\theta}\boldsymbol{q}_0 + \frac{\sin r\theta}{\sin\theta}\boldsymbol{q}_1, \tag{2.4}$$

where θ is the angle spanned by the two quaternions $\boldsymbol{q}_0$ and $\boldsymbol{q}_1$ on a four dimensional unit sphere:

$$\cos\theta = \frac{\langle \boldsymbol{q}_0, \boldsymbol{q}_1 \rangle}{\|\boldsymbol{q}_s\| \cdot \|\boldsymbol{q}_1\|}. \tag{2.5}$$

2.2 Trajectory Generation

The assist system must continuously plan a reference trajectory to the intended target of the human motion. Deviations of the actual trajectory, which is commanded by the human

operator, from the reference trajectory are smoothly corrected to guarantee a successful execution of the intended action. As stated above, it is important that the reference trajectory complies with the characteristics of human motions. Otherwise, the human operator would perceive the corrections as unintuitive and disturbing. Therefore, a trajectory generator based on the previously presented principles of human motion planning is designed. In the following, the trajectory generator is described for a single scalar component. This is possible, because the Cartesian components are decoupled in the minimum-jerk criterion.

At any sampled time t_k, the trajectory generator is provided with the current position x_k, velocity $\dot{x}_k$, and acceleration $\ddot{x}_k$ as well as the desired target point x_f and target time t_f, where $t_f > t_k$. Based on this information, the trajectory generator must provide a trajectory which fulfills the boundary conditions and has humanlike motion characteristics. Although the target point x_f and target time t_f are continuously recalculated, the trajectory generator must ensure that position, velocity, and acceleration do not exhibit any discontinuities.

The full state of the trajectory generator is described by the state vector $\boldsymbol{\xi}$:

$$\boldsymbol{\xi} = \begin{pmatrix} x \\ \dot{x} \\ \ddot{x} \end{pmatrix}. \tag{2.6}$$

With this definitions, the continuity condition, and the desired target conditions can be easily expressed as:

$$\boldsymbol{\xi}(t_k) = \boldsymbol{\xi}_k \qquad \boldsymbol{\xi}(t_f) = \boldsymbol{\xi}_f. \tag{2.7}$$

These six equations fully define the fifth-order polynomial which is used to model the humanlike trajectory.

For some computations, it is convenient to define the relative time to target τ such that $\tau = t - t_f$. Obviously, τ is always negative while travelling the trajectory ($t < t_f$), and zero, when the target point is reached ($t = t_f$). Furthermore, it is important to note that the relative time τ is not forcibly monotonic because adjustments of the final time t_f during the execution of the trajectory can make the time τ go backward, i.e. $\tau_{k_1} > \tau_{k_2}$ for $k_1 < k_2$.

As the humanlike trajectories are expressed as fifth-order polynomial, the vector $\boldsymbol{\tau}$ is introduced, which contains the powers of τ. The vector $\boldsymbol{\tau}$ and its derivatives $\dot{\boldsymbol{\tau}}$ and $\ddot{\boldsymbol{\tau}}$ take the following form:

$$\boldsymbol{\tau} = \begin{pmatrix} 1 \\ \tau \\ \tau^2 \\ \tau^3 \\ \tau^4 \\ \tau^5 \end{pmatrix} \qquad \dot{\boldsymbol{\tau}} = \begin{pmatrix} 0 \\ 1 \\ 2\tau \\ 3\tau^2 \\ 4\tau^3 \\ 5\tau^4 \end{pmatrix} \qquad \ddot{\boldsymbol{\tau}} = \begin{pmatrix} 0 \\ 0 \\ 2 \\ 6\tau \\ 12\tau^2 \\ 20\tau^3 \end{pmatrix}. \tag{2.8}$$

With the definition of $\boldsymbol{\tau}$ and the six dimensional coefficient vector $\boldsymbol{p}$, the trajectory can be easily described as:

$$x(\tau) = \boldsymbol{\tau}^T \boldsymbol{p}. \tag{2.9}$$

The corresponding trajectory of the full state is:

$$\boldsymbol{\xi}(\tau) = \begin{pmatrix} \boldsymbol{\tau}^T \boldsymbol{p} \\ \dot{\boldsymbol{\tau}}^T \boldsymbol{p} \\ \ddot{\boldsymbol{\tau}}^T \boldsymbol{p} \end{pmatrix}. \tag{2.10}$$

The parameter vector $\boldsymbol{p}$ to generate a trajectory which fulfills (2.7) is obtained by solving:

$$\begin{pmatrix} \boldsymbol{\tau}_k^T \\ \dot{\boldsymbol{\tau}}_k^T \\ \ddot{\boldsymbol{\tau}}_k^T \\ \boldsymbol{\tau}_f^T \\ \dot{\boldsymbol{\tau}}_f^T \\ \ddot{\boldsymbol{\tau}}_f^T \end{pmatrix} \boldsymbol{p} = \begin{pmatrix} 1 & \tau_k & \tau_k^2 & \tau_k^3 & \tau_k^4 & \tau_k^5 \\ 0 & 1 & 2\tau_k & 3\tau_k^2 & 4\tau_k^3 & 5\tau_k^4 \\ 0 & 0 & 2 & 6\tau_k & 12\tau_k^2 & 20\tau_k^3 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \end{pmatrix} \boldsymbol{p} = \begin{pmatrix} x_k \\ \dot{x}_k \\ \ddot{x}_k \\ x_f \\ \dot{x}_f \\ \ddot{x}_f \end{pmatrix}. \tag{2.11}$$

For constant $\boldsymbol{\xi}_f$ and constant t_f, the coefficient vector $\boldsymbol{p}$ also remains constant. In a typical application scenario, however, the target state $\boldsymbol{\xi}_f$ is continuously updated in order to take the increasingly accurate estimate of the target position from the scene recognition system into account. Likewise, the target time t_f is continuously updated to reflect the increasingly accurate estimate of the target time from the intention estimation process.

Therefore, the coefficient vector must also be continuously updated in order to incorporate the current $\boldsymbol{\xi}_f$ and t_f. This is done by recalculating (2.11) at every time step k and

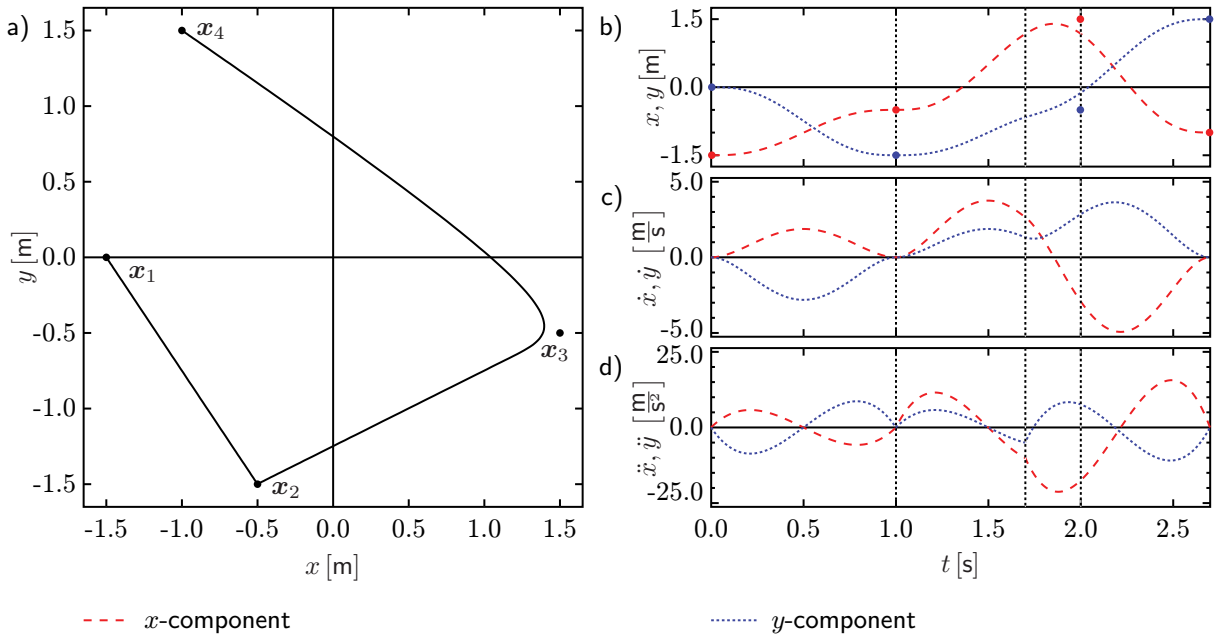

Figure 2.3: Trajectory generation with four target points. The trajectory generator is consecutively commanded to plan trajectories to the points $\boldsymbol{x}_1 = (-1.5\text{ m}, 0\text{ m})^T$, $\boldsymbol{x}_2 = (-0.5\text{ m}, -1.5\text{ m})^T$, $\boldsymbol{x}_3 = (1.5\text{ m}, 0.5\text{ m})^T$, and $\boldsymbol{x}_4 = (-1\text{ m}, 1.5\text{ m})^T$. Each segment is supposed to take 1 s. While travelling from point $\boldsymbol{x}_2$ to point $\boldsymbol{x}_3$, the execution is interrupted after 0.7 s. a) x-y plot of the trajectory. b) x- and y-coordinates as a function of time. c) Velocity profiles. d) Acceleration profiles.

propagating the internal state $\boldsymbol{\xi}_k$ via:

$$\boldsymbol{\xi}_{k+1} = \begin{pmatrix} \boldsymbol{\tau}_{k+1}^T \boldsymbol{p} \\ \dot{\boldsymbol{\tau}}_{k+1}^T \boldsymbol{p} \\ \ddot{\boldsymbol{\tau}}_{k+1}^T \boldsymbol{p} \end{pmatrix}. \tag{2.12}$$

The output of the trajectory generator is illustrated in Fig. 2.3 for a two-dimensional movement. The trajectory generator is commanded to travel from the initial position $\boldsymbol{x}_1 = (-1.5\text{ m}, 0\text{ m})^T$ to the position $\boldsymbol{x}_2 = (-0.5\text{ m}, -1.5\text{ m})^T$ at $t_f = 1$ s. Next, it is commanded to travel to position $\boldsymbol{x}_3 = (1.5\text{ m}, 0.5\text{ m})^T$ at $t_f = 2$ s. However, at $t = 1.7$ s the target position is altered to $\boldsymbol{x}_4 = (-1\text{ m}, 1.5\text{ m})^T$ at $t_f = 2.7$ s. As shown in Fig. 2.3 b), the produced position, velocity, and acceleration trajectories are continuous.

The trajectory generator can be used for equally sampled time steps as well as not equally sampled time steps. This can relieve the real-time conditions of the target system.

2.3 Online Intention Estimation

In Sec. 2.1, the general form of the mathematical model for human point-to-point movements is stated to be a fifth-order polynomial. With given coefficients, the trajectory, i.e. the spatial as well as the temporal course of the movement, is exactly determined. The intention estimation process gives rise to the inverse problem: Given a part of a human point-to-point trajectory, find the coefficients of the model, which best match the observed trajectory. These coefficients yield the intended target of the human motion.

Recall that the estimated intentions are to be used to activate and control the appropriate assist function. Consequently, the estimate should be available as early as possible in order to allow early assistance. In any case, it must be available before the intended action is completed. This translates to some requirements on the estimation algorithms such as fast convergence, real-time capability, which implies an upper bound computation time, and possibly an incremental update law.

Pilot studies suggest that the translational part of the peg-in-hole approach movement is far better suited for intention estimation than the rotational part. Fig. 2.1 also illustrates that different goals may require the same alignment procedure, which can thus not be used to differentiate between these goals. Therefore, the intention estimation process only relies on the translational components. In order to simplify the mathematical notation, only scalar positions x will be considered in the following. However, the method can be directly applied to vectorial positions $\boldsymbol{x}$.

Mathematically, the intention estimation process can be described as curve fitting problem. The physical trajectory travelled by the hand of the operator, which is assumed to be identical to the unaltered physical trajectory travelled by the tool of the teleoperator, is described as continuous function of the time:

$$x = f(t) \tag{2.13}$$

The physical trajectory is sampled by the telepresence system at instants t_i. In the following, the system is assumed to be uniformly sampled, although this is not a strict requirement. Therefore, the observed trajectory takes the form:

$$x_i = f(t_i), \quad t_s \leq t_i < t_f \tag{2.14}$$

The general model function, which is used to approximate the data, is a fifth-order polynomial with coefficient vector[1] $\boldsymbol{q}$:

$$\hat{x}_i(\boldsymbol{q}) = h(t_i, \boldsymbol{q}) \tag{2.15}$$

The input to the intention estimation algorithm is the observed part of a point-to-point trajectory. It is given as a sequence of n time stamps t_i and corresponding positions x_i, $i = 1 \ldots n$, where $t_s \leq t_1$ and $t_n < t_f$. From this data, the intention estimation algorithm is supposed to find the parameters $\boldsymbol{q}$ of the model function $h(t, \boldsymbol{q})$, which best approximate the observed data. The best approximate is defined in a least squares sense by the cost function $G(\boldsymbol{q})$:

$$G(\boldsymbol{q}) = \frac{1}{2} \sum_{i=1}^{n} \|\hat{x}_i(\boldsymbol{q}) - x_i\|^2 = \frac{1}{2} \sum_{i=1}^{n} \|h(t_i, \boldsymbol{q}) - x_i\|^2 \tag{2.16}$$

2.3.1 Comparison of Different Estimation Methods

Due to the high variability of human arm movements and the limited accuracy of the employed model, an estimation solely based on minimizing the cost function (2.16) yields inaccurate results. Prior knowledge, which is introduced in the form of additional constraints in the optimization procedure, can significantly improve the accuracy. In the following, different ways of incorporating prior knowledge are compared.

The effects of constraints in the optimization procedure are illustrated in Fig. 2.4, which shows the longitudinal component of a point-to-point movement. In each subfigure, the recorded exemplary course of a human point-to-point movement and the extrapolated trajectories for three different estimation methods are plotted. The Figs. 2.4a) through 2.4d) differ in the percentage of data points which are used to estimate the parameters of the trajectory model. In Fig. 2.4a), the estimate is produced after 0.25 s, in Fig. 2.4b) after 0.5 s, in Fig. 2.4c) after 1.0 s, and in Fig. 2.4d) after 2.0 s, where the total duration of the movement from start point to end point is 2.3 s. All trajectories are normalized to start at position $x = 0$. The time axis is the relative time $\tau = t - t_f$ such that the movement ends at $\tau = \tau_f = 0$.

The following three estimation methods are considered:

- Estimation without additional constraints: The trajectories are obtained by linear regression based on (2.16). As shown by the narrow dashed lines in Fig. 2.4, the extrapolated trajectories yield a bad fit and do not cross the target point, even when

[1] Note that the coefficient vector $\boldsymbol{q}$ parameterizes a polynomial in the absolute time t, whereas the coefficient vector $\boldsymbol{p}$, defined in Sec. 2.2, parameterizes a polynomial in the relative time τ

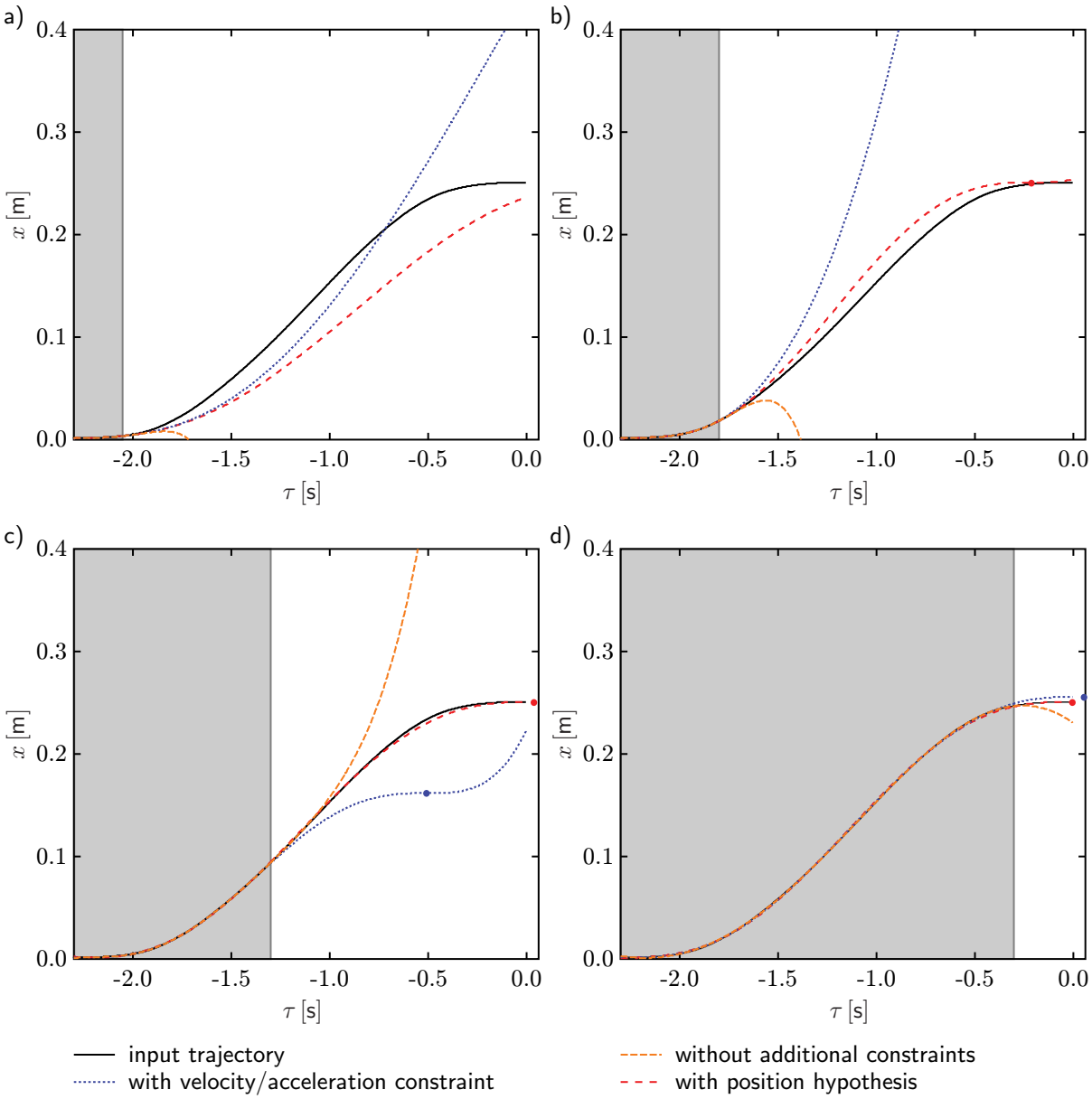

Figure 2.4: Comparison of intention estimation methods. The figure illustrates the performance of the estimation method w.r.t. their ability to extrapolate the trajectory based on different amounts of observed data, which increases from Fig. a) to Fig. d) as indicated by the shaded area. In Fig. a), the first 0.25 s of the trajectory data is given, in Fig. b) 0.5 s, in Fig. c) 1.0 s, and in Fig. d) 2.0 s of the trajectory is already observed. The total duration of the point-to-point trajectory is 2.3 s.

a large portion of the trajectory data is available for the estimation. At the final point of the movement, velocity and acceleration are expected to be zero, i.e. $\dot{x}(t_f) = 0$ and $\ddot{x}(t_f) = 0$. As the estimated trajectories do not exhibit points where velocity and acceleration are zero at the same time, a unique target position and target time cannot be derived from the trajectory at all.

- By adding appropriate constraints, the trajectory can be forced to zero velocity and zero acceleration at some future time $t_f > t_n$, i.e. $\dot{\boldsymbol{x}}(t_f) = 0$, $\ddot{\boldsymbol{x}}(t_f) = 0$. The dotted lines in Fig. 2.4 show that this enhances the prediction accuracy. The estimates of target position and target time converge to the correct values when enough data is available. However, estimates at reasonable accuracy are only available shortly before the end of the trajectory is reached. Furthermore, the estimation process uses a combined line-search and Levenberg-Marquardt method, which does not meet the real-time requirements.

- If not only target velocity $\dot{x}(t_f) = 0$ and target acceleration $\ddot{x}(t_f) = 0$ are known, but also the target position $x(t_f) = x_f$ is known, the only unknown variable is the target time t_f. Fig. 2.4, sparse dashed lines, shows that this method provides reasonably exact predictions of t_f even when only a small portion of the trajectory data is known and becomes very accurate when the trajectory is close to the target point. As the target position, velocity, and acceleration are introduced as constraints, the extrapolated trajectory always crosses that target point with zero velocity and acceleration.

  As the target position x_f is part of the estimation, it cannot be known beforehand. However, hypotheses about possible target points can be set up. In order to plan a reference trajectory to the target, the assist system must be able to identify and localize possible target points in the environment anyway. These target points can be directly used as hypotheses in the estimation process. Therefore, the hypothesis based estimation approach does not need to be able to estimate the correct target point from the observed trajectory, but it must only be able to reliably differentiate between several hypotheses.

  The hypothesis based trajectories in Fig. 2.4 are also produced with the combined line-search and Levenberg-Marquardt method, which is not real-time capable. However, the optimization process is much simpler than in the previous case, because only one scalar parameter (the target time t_f) must be estimated instead of four parameters (the target time t_f and the three-valued vector $\boldsymbol{x}_f$). Therefore, gradient-based optimization techniques can be used to estimate t_f in real-time.

In the following subsections, two estimation methods are presented, both of which rely on hypotheses of the target point x_f. At each estimation step k, the following data is available to the estimation procedure:

k	: current estimation step
n_k	: number of data points available at estimation step k
t_i, x_i	: sample times and sampled positions, where $i = 1 \dots n_k$, $x_i = f(t_i)$
x_f	: hypothesized final position of the trajectory
$\hat{t}_f^{(k)}$	: k-th estimate of the target time t_f

The result of the estimation step is the updated estimate of the target time $\hat{t}_f^{(k+1)}$.

At each sample step, the number of observed data points increases. The estimation steps k do not need to be synchronized with the data sample steps. When more than one estimation step is performed per added data point, k grows faster than the number of

collected data points n_k. Likewise, when less than one estimation step per added data point is performed, k grows slower than n_k.

Both presented methods cannot be proven to converge. However, empirical analyses show that convergence is reached for all practical cases. The lack of a proof does not render these methods inapplicable. If no trajectory fit can be found, the performance only degrades to the performance of the unassisted telepresence system, but it does not compromise the stability or safety of the system.

2.3.2 Trajectory Estimation with Known Start Point

In order to simplify the estimation process, it is assumed that starting point x_s and endpoint x_f are known, as well as the start time t_s, where $x(t_s) = x_s$. The only unknown parameter is the target time t_f. Recalling (2.2), the model takes the following form:

$$h(t, t_f) = x_f + (x_f - x_s)\left(6\frac{\tau^5}{T^5} + 15\frac{\tau^4}{T^4} + 10\frac{\tau^3}{T^3}\right), \tag{2.17}$$

where $\tau = t - t_f$ and $T = t_f - t_s$.

Obviously, the model fulfills the requirements at starting point and endpoint $h(t_s, t_f) = x_s$ and $h(t_f, t_f) = x_f$. At both points, it yields zero velocity and zero acceleration $\dot{h}(t_s) = \dot{h}(t_f) = 0$ and $\ddot{h}(t_s) = \ddot{h}(t_f) = 0$. The model generates estimates of the point coordinates $\hat{x}_i$ based on the final time t_f:

$$\hat{x}_i(t_f) = h(t_i, t_f). \tag{2.18}$$

The respective cost function, which determines the optimal value t_f, is given as (cf. (2.16)):

$$G(t_f) = \frac{1}{2}\sum_{i=1}^{n_k} \|\hat{x}_i(t_f) - x_i\|^2, \tag{2.19}$$

such that the estimated final time $\hat{t}_f$ can be expressed as:

$$\hat{t}_f = \arg\min_{t_f} G(t_f). \tag{2.20}$$

Minimizing (2.19) constitutes a one-dimensional non-linear least-squares problem. This can be solved by the *Gauss-Newton* method [16], which is the most commonly used method for minimizing a cost function G which is the sum of squared functions. In each iteration step, the Gauss-Newton algorithm descends toward the minimum in the opposite direction of the gradient ∇G.

For the cost function (2.19), the iteration takes the following form:

$$\hat{t}_f^{(k+1)} = \hat{t}_f^{(k)} - \frac{\sum_{i=1}^{n_k}\left(\frac{\partial}{\partial t_f}\hat{x}_i(\hat{t}_f^{(k)})\right)\left(\hat{x}_i(\hat{t}_f^{(k)}) - x_i\right)}{\sum_{i=1}^{n_k}\left(\frac{\partial}{\partial t_f}\hat{x}_i(\hat{t}_f^{(k)})\right)^2} \tag{2.21}$$

While the trajectory is travelled, the number of observed data points continuously increases. As the estimated parameter $\hat{t}_f$ varies slowly with additional data points, one iteration step per added data point is sufficient, i.e. $k = n_k$.

The computational costs of performing one iteration step according to (2.21) increase linearly with the number of data points. In order to keep the computational costs constant, an *incremental gradient method* can be employed [16], which performs one iteration step per added data point with the modified iteration rule:

$$\hat{t}_f^{(k+1)} = \hat{t}_f^{(k)} - \alpha_k \left(\frac{\partial}{\partial t_f} \hat{x}_i(\hat{t}_f^{(k)}) \right) \left(\hat{x}_i(\hat{t}_f^{(k)}) - x_i \right) \tag{2.22}$$

The factor α_k must be a strictly positive number and determines the step size of the iteration step. The choice of α_k is crucial for the convergence of the incremental gradient method. Empirically, the following rule provides a good compromise between convergence performance and stability:

$$\alpha_k = \left(\frac{\hat{t}_f^{(k)} - t_{n_k} + \lambda}{x_f - x_s} \right)^2, \quad \lambda = 0.1 \text{ s} \tag{2.23}$$

The performance of the hypothesis based intention estimation algorithms can be assessed by two main criteria. The first criterion determines how well the algorithm can differentiate between different hypotheses. The second criterion determines how fast the algorithm converges to the correct final time. In Fig. 2.5, exemplary results which illustrate the performance of the standard Gauss-Newton method and the incremental gradient method are presented. For both methods, the estimated relative time $\hat{\tau}$ is plotted as a function of the true relative time τ. Furthermore, the estimated trajectory and the residuals between true and estimated trajectory are shown as functions of time. The input trajectory is tested against three hypotheses: the correct one ($d = 0.00\,\text{m}$), a hypothesis with a target deviation of $d = -0.01\,\text{m}$, and a hypothesis with a target deviation of $d = 0.10\,\text{m}$.

For the correct hypothesis, the standard Gauss-Newton method shows very accurate estimates of the target time already 1.5 s before the target is reached. The target times of the wrong hypotheses are underestimated or overestimated depending on the sign of the deviation. The correct estimate of the target time yields a good match between the input and the estimated trajectory and small residuals. After 0.8 s before the target is reached, no confusion of the hypotheses occurs.

The incremental gradient method shows a similar performance. The accuracy of the time estimate is slightly lower, reaching good values not until 0.5 s before the target is reached. As the incremental gradient method does not iterate over the full trajectory data in each update step, the estimated trajectory follows the input trajectory more closely. The residuals also allow a clear selection of the correct hypothesis.

2.3.3 Trajectory Estimation with Unknown Start Point

In many natural motion sequences, it is difficult to exactly extract the start of a point-to-point movement. In some cases, such a start point might even not exist when the point-to-

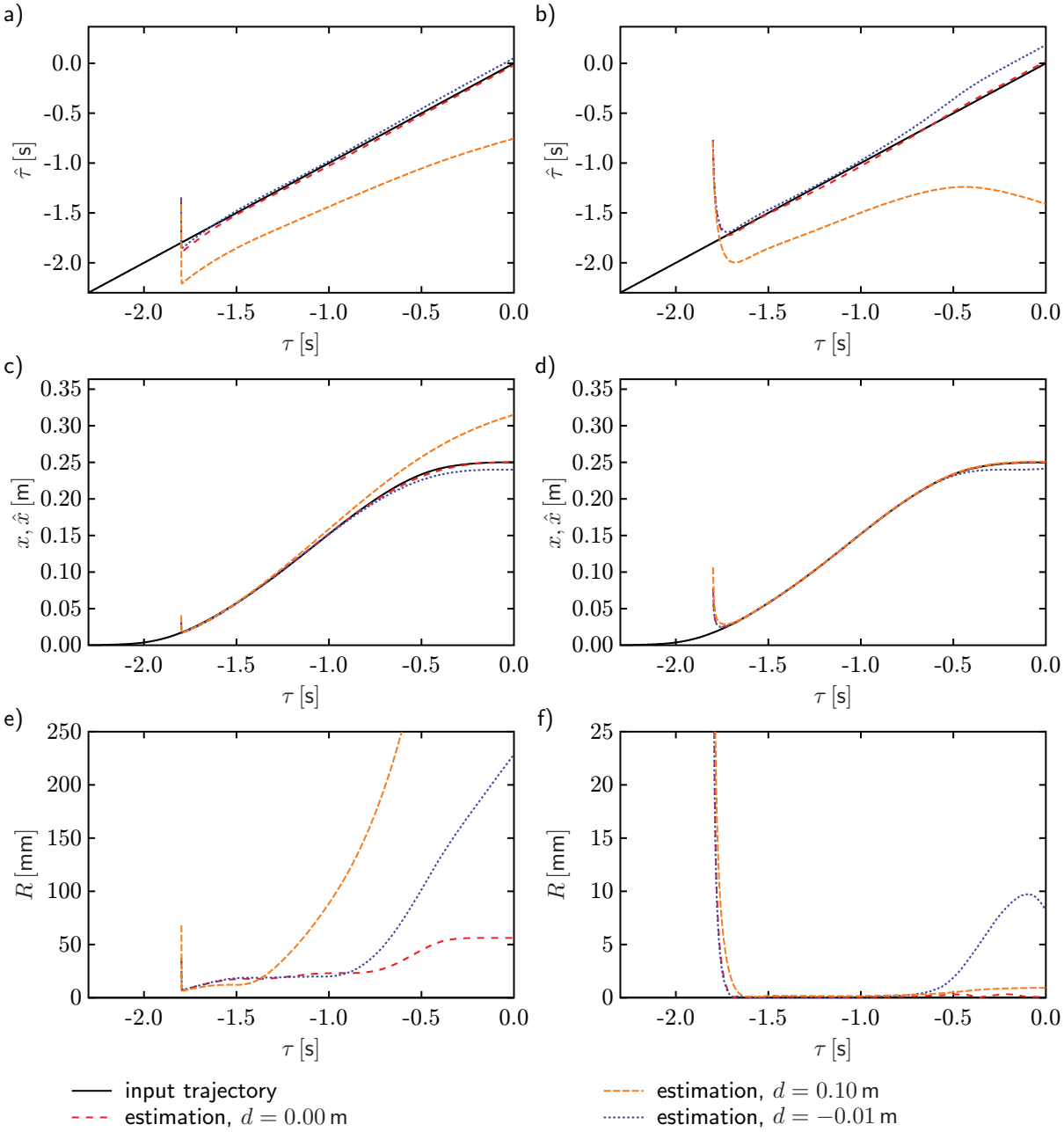

Figure 2.5: Properties of trajectory estimation with known start point. On the left hand side, in Figs. a), c), and e), results of the Gauss-Newton method are shown. On the right hand side, in Figs. b), d), and f), results of the incremental gradient method are shown. The top row, i.e. Figs. a) and b), illustrates the course of the estimated relative target time $\hat{\tau}$ as a function of the real relative target time τ. In the middle row, i.e. Figs. c) and d), the observed trajectory x_i is compared to the estimated trajectories $\hat{x}_i$. In the bottom row, i.e. Figs. e) and f), the residuals of (2.19) are plotted as a function of time.

point movement emerges from another movement, e.g. avoiding an obstacle. Therefore, an intention estimation algorithm which does not depend on such a start point is desirable.

Recalling definition (2.8), the mathematical model can be expressed as

$$h(t, t_f, \boldsymbol{p}) = \boldsymbol{\tau}^T \boldsymbol{p}, \tag{2.24}$$

where t is the independent variable, and t_f and $\boldsymbol{p}$ are model parameters.

The boundary conditions are given by

$$h(t_f, t_f, \boldsymbol{p}) = x_f \quad \dot{h}(t_f, t_f, \boldsymbol{p}) = \dot{x}_f \quad \ddot{h}(t_f, t_f, \boldsymbol{p}) = \ddot{x}_f, \tag{2.25}$$

where final velocity $\dot{x}_f$ and final acceleration $\ddot{x}_f$ are typically zero.

For given parameters t_f and $\boldsymbol{p}$ the position estimates $\hat{x}_i$ can be expressed analogously to (2.18):

$$\hat{x}_i(t_f, \boldsymbol{p}) = h(t_i, t_f, \boldsymbol{p}) \tag{2.26}$$

When the final time t_f is given, the optimal coefficient vector $\boldsymbol{p}$ can be determined from the trajectory data x_i and the end-state $\boldsymbol{\xi}_f$ by linear regression. Therefore, the coefficient vector $\boldsymbol{p}$ can be expressed as a function of t_f: $\boldsymbol{p} = \boldsymbol{p}(t_f)$. As $\boldsymbol{p}$ depends non-linearly on t_f, t_f is estimated by a Gauss-Newton optimization method, which minimizes the cost function of the following form:

$$G(t_f) = \frac{1}{2} \sum_{i=1}^{n_k} \left\| \hat{x}_i(t_f, \boldsymbol{p}(t_f)) - x_i \right\|^2 \tag{2.27}$$

Starting from an initial estimate $\hat{t}_f^{(k)}$, the estimation process involves the following operations

1. The coefficient vector $\boldsymbol{p}^{(k)}$ is calculated at the current estimate of the final time $\hat{t}_f^{(k)}$. As (2.27) is linear in $\boldsymbol{p}$, the solution can be found analytically by linear regression. The three components p_0, p_1, p_2 can be directly obtained by substituting (2.25) into (2.24), because the coefficient vector $\boldsymbol{p}$ describes a polynomial in the relative time τ and $\tau_f = 0$ by definition.

2. The n_k-dimensional vector of residual function $\boldsymbol{g}(t_f)$ is constructed, where the component functions are $g_i(t_f) = \hat{x}_i(t_f, \boldsymbol{p}(t_f)) - x_i$ for $i = 1 \dots n_k$.

3. The Jacobian is calculated as $\boldsymbol{J} = \frac{\partial}{\partial t_f} \boldsymbol{g}(\hat{t}_f^{(k)})$.

4. The next estimate $\hat{t}_f^{(k+1)}$ is obtained by the update equation:

$$\hat{t}_f^{(k+1)} = \hat{t}_f^{(k)} - (\boldsymbol{J}^T \boldsymbol{J})^{-1} \boldsymbol{J}^T \boldsymbol{g}(\hat{t}_f^{(k)}) \tag{2.28}$$

The above described calculations are computationally expensive and prone to numerical instabilities. An incremental method for reducing the computations required for solving the linear regression problem is presented in Appendix A. In this appendix, also some comments on how to avoid the numerical problems are given.

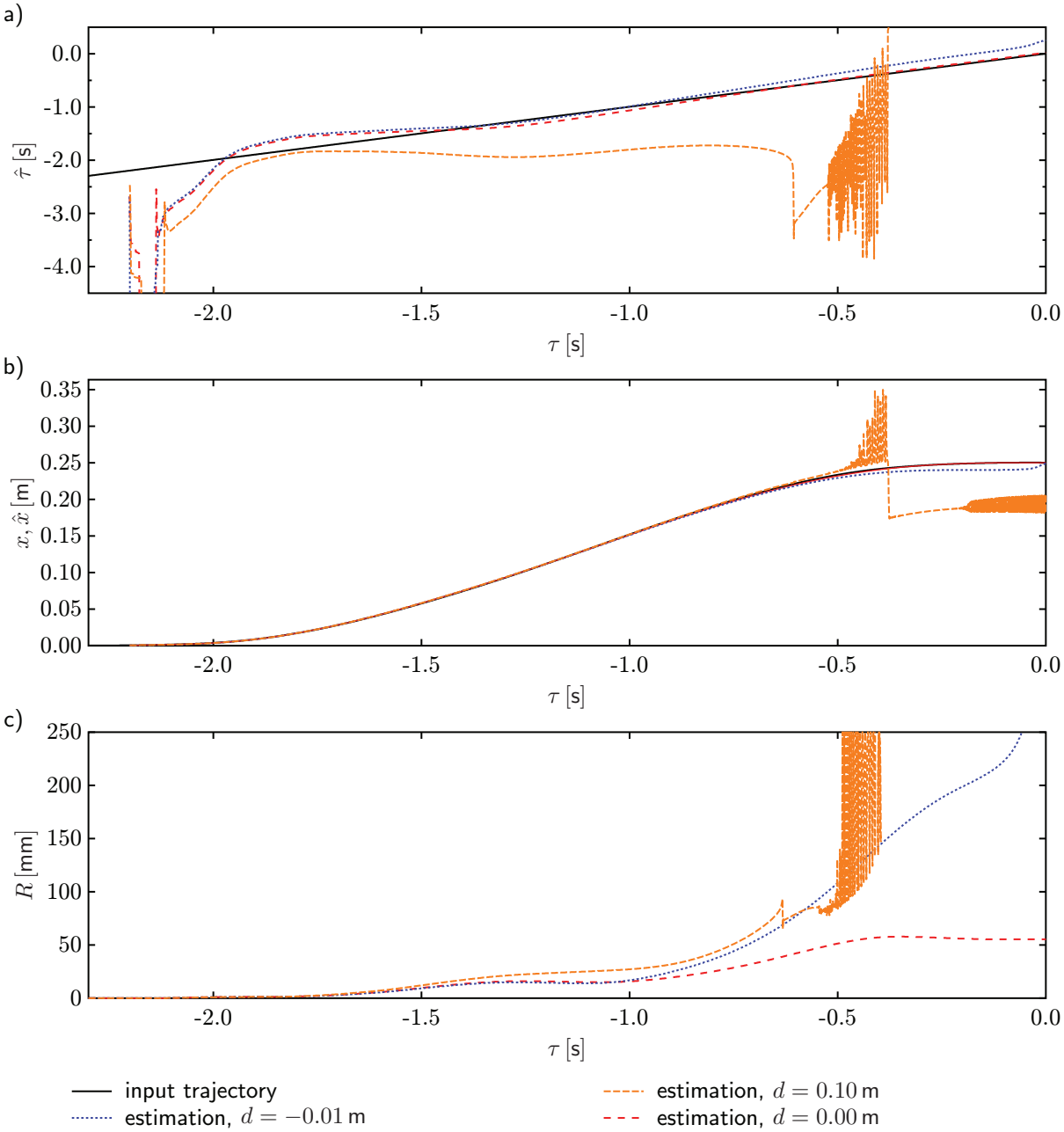

Figure 2.6: Properties of trajectory estimation with unknown start point. The figures show estimation results of the Gauss-Newton optimization. Fig. a) illustrates how the estimates of the relative time $\hat{\tau}_f$ develop over time. In Fig. b) the observed trajectory x_i is compared to the estimated trajectories $\hat{x}_i$. The residuals of the estimation process are depicted in Fig. c)

Fig. 2.6 shows the performance of the intention estimation with unknown start point in a similar way to Fig. 2.5. The estimated relative time $\hat{\tau}$ is plotted over the true relative time τ, and the estimated trajectory as well as residuals are shown as functions of time. Again, the input trajectory is tested against the correct hypothesis and two false hypotheses.

For the correct hypothesis, good estimates of target time and future trajectory are available approx. 0.8 s before the target is reached. A robust selection of the correct goal is reached even 1.0 s before reaching the target. For a deviation of $d = 0.10$ m, the estimator produces oscillatory results. However, the oscillations do not reduce the robustness of the intention estimation.

2.4 Discussion

The design of an assisted telepresence system requires a thorough understanding of human motion planning and human haptic perception in order to achieve a true collaboration between human and robot. On the one hand, this understanding can be used to infer the intentions of the human operator from his actions. On the other hand, it enables the robot to mimic the human behavior and, thus, to expose a more natural collaboration.

While a large body of literature exists which models human behavior for certain tasks, little effort has been taken to exploit this knowledge for real-time intention estimation. In this chapter, a novel integrated approach for online estimation and generation of humanlike motion trajectories was developed. Its high estimation accuracy allows estimating the target and reshaping the trajectory toward the target in real time.

The approach was applied to and validated for point-to-point movements, which are modeled by linear differential equations. The model assumes that point-to-point movements in free space are planned in the Cartesian space by optimizing a performance criterion. It was experimentally confirmed that humans conduct movements such that the mean squared jerk over time is minimized. This insight yields the mathematical description of humanlike motion trajectories. In contrast to commonly used Hidden Markov Models, the dynamic model does not require any training.

The trajectory generator and estimator presented in this chapter are designed to closely interact with each other: The trajectory generation scheme can accommodate changes of the desired motion target as well as motion duration while executing the trajectory. The changes are integrated without sacrificing the continuity of position, velocity, and acceleration signals. The complementary trajectory estimation scheme determines the expected motion target and motion duration from parts of a trajectory. The estimator works online and continuously updates the estimates when provided with new data from the progressing trajectory. Thus both parts, trajectory generator and trajectory estimator, can be easily combined to extrapolate trajectory data in a humanlike fashion.

Although only the class of point-to-point movements was considered in this thesis, the concept of estimating and reshaping human actions by a complementary estimator and generator is very general. Along with new knowledge provided by neuroscience, the class of actions which can be estimated will be broadened. The presented intention estimation

concept can even be applied to multi-user scenarios by using models of human-human collaboration.

3 Scene Recognition

In order to fulfill its purpose, the assist system needs some knowledge about the remote environment. As the assist concept does not include a preplanning phase, the remote environment must be analyzed online by the assist system. For the considered screwdriving task, the required knowledge consists in the locations of possible targets. These must be known for two reasons: On the one hand, they serve as hypotheses to the intention estimation process described in Chapter 2, on the other hand the pose of the selected target is used to plan a reference trajectory.

The problem of estimating the target poses is twofold. In the first step, possible targets must be identified in the images of one or more cameras, and the image coordinates of characteristic points of the target must be measured. In the second step, these 2D image coordinates must be translated to 3D coordinates, which describe the relative pose between target and teleoperator.

The hardware which serves as basis for the scene recognition is described in Sec. 3.1. A review of feature extraction algorithms and a toolkit for the localization of artificial features are presented in Sec. 3.2. Sec. 3.3 is devoted to the actual pose estimation process, which translates 2D coordinates to 3D coordinates. An evaluation of the complete pose estimation system can be found in Sec. 3.4. The chapter concludes with a discussion of the results.

3.1 Hardware Setup

The teleoperator which is considered throughout this thesis is anthropomorphically designed. It has about human height and is equipped with two human scale robotic arms. Furthermore, it is equipped with a stereoscopic camera-head, which can be arbitrarily rotated in three degrees of freedom. In order to enable assistance, the anthropomorphic setup is supplemented with two orthogonally mounted eye-in-hand cameras, which provide a close-up view of the manipulation area, see Fig. 3.1.

For maximum resolution, the camera head can be equipped with two cameras which provide grayscale images at a resolution of 1024×768 pixels at a frame rate of 30 Hz. Having a diagonal viewing angle of 58°, this leads to a pixel size of approx. 0.61 mm at a typical working distance of 0.6 m. As the human operator can be expected to keep the manipulation target in the viewing area of both cameras, coarse pose estimates of possible targets can be extracted from these images.

The eye-in-hand cameras are mounted in such a way that they always capture the area in front of the tool. They have a resolution of 640×480 pixels at a frame rate of 30 Hz. The diagonal viewing angle of 66° yields a pixel size of approx. 0.11 mm at the point where the optical axes of the cameras and the tool axis intersect. This resolution is high enough to

a) b)

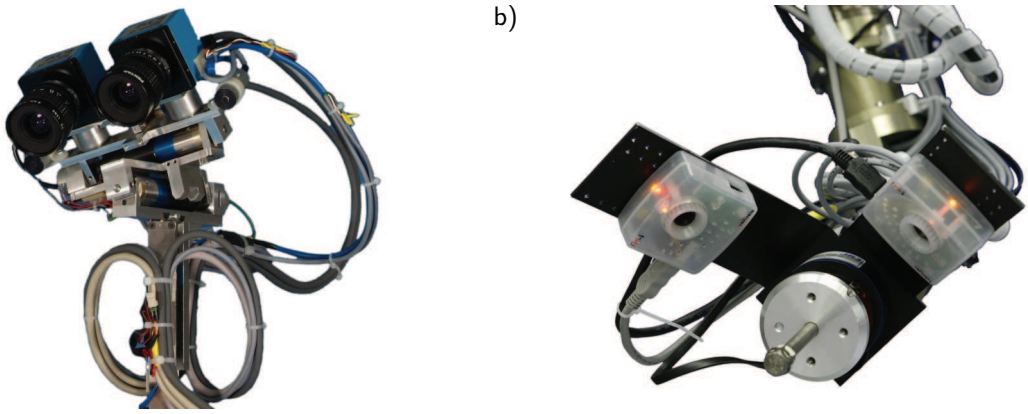

Figure 3.1: Hardware setup of camera system. In Fig. a), the camera head comprising a stereo camera setup mounted on a 3 DOF neck is shown. In Fig. b), the eye-in-hand cameras mounted at the end-effector of the robot arm are shown. They are tilted by $45°$ such that their optical axes point at the tip of the tool. For all cameras, the x-axis is pointing in direction of the optical axis, the y-axis is pointing to the left, and the z-axis is pointing up.

enable exact positioning of the tool relative to the workpiece, e.g. the screwdriver relative to the screwhead.

The setup has the advantage that coarse measurements for the early stage of intention estimation can be performed on basis of the head cameras. The eye-in-hand cameras are used in the close range of the target. As the pixel size decreases with decreasing distance, the measurement increases while the target is being approached. Thus, the system reaches its maximum accuracy especially in the critical phases.

3.2 Feature Extraction

In order to detect a target in the remote environment and to determine its pose relative to the camera, it must first be recognized and localized in the camera image. From the position, orientation, size, and shape of the projected target in the camera image, the absolute pose of the target in the remote environment can be subsequently calculated.

The process of extracting relevant data w.r.t. to possible targets from the camera image is called *feature extraction*. More formally, feature extraction describes the dimensionality reduction from a high dimensional input space to a feature space of much lower dimension. A data point in the feature space is called *feature vector*. A well-defined feature vector is supposed to carry all relevant data from the input space while omitting redundant or unnecessary information. Obviously, the selection of an appropriate feature space is application dependent.

3.2.1 Comparison of Features

Features used in object tracking and pose estimation applications are typically local invariant features. These denote characteristic points in the image, which should be invariant to changes in illumination, perspective, etc. Typically, local features coincide with image points which exhibit significant changes in intensity, color, or texture compared to their neighborhood. Good features show a high degree of repeatability, accuracy, and efficiency [69]. In other words, in two images of the same object, the same features should be detected at the same object locations with little computational effort. For pose estimation, it is furthermore important that the features are distinct, i.e. that different features in the same image can be reliably discriminated.

A very common type of features are extracted by *corner detectors*. As the name suggests, they detect points at the intersection of two edges. The detected corners in the 2D camera image do not forcibly stem from corners of the corresponding 3D object, because they can also occur at occlusion boundaries, shadow lines etc. The most prominent examples of corner detectors include the Harris Detector [41], variants such as the Harris-Laplace and Harris-Affine Detector, as well as the SUSAN Detector. A complementary feature detection approach is given by the *blob detectors* [64]. These detect points or regions in the image which differ in intensity from their neighborhood. The most widely used blob detectors are the Hessian Detector and its advancements, the Hessian-Laplace and Hessian-Affine Detectors.

A more specific class of features are *Scale-Invariant Feature Transforms (or SIFT)* [66]. Similar to blobs, SIFT features are based on local extrema in the image space, so called keypoints. However, these extrema are required to be invariant to scaling. Furthermore, all keypoints with low contrast and keypoints along edges are discarded, because their position is unstable. Finally, a keypoint descriptor, the feature vector, is constructed based on the image gradient around the keypoint. SIFT features are one of the most powerful feature classes currently known.

More recently, some advances of SIFT features have been presented with *Speeded Up Robust Features (SURF)* [13] and *Gradient Location and Orientation Histogram (GLOH)* [70]. They show similar performance as SIFT features and are computationally more efficient.

3.2.2 Artificial Features

The process of feature extraction can be made simpler, more robust, and computationally less expensive by placing special markers on the objects of interest. These markers are designed to produce a distinctive pattern in the image, which makes it easier to detect and localize them.

Different marker types have been proposed, where round and square shapes dominate. Most markers can encode a unique identification code, which allows differentiating between different markers on the same object and makes pose estimation possible. Such markers are called *Coded Fiducials* and are very popular in augmented reality applications. Square

shaped fiducials provide the advantage that the four corner points are sufficient for pose estimation such that a single detected fiducial is sufficient.

ARToolKit

In the presented thesis, the *ARToolKit*, developed by Mark Billinghurst und Hirokazu Kato at the Human Interface Technology Lab of the University of Washington, is used. The ARToolKit is based on square fiducials for feature detection.

The ARToolKit uses classic camera calibration techniques. The intrinsic parameters of a camera are described by a perspective projection matrix K, which relates 3D camera coordinates to 2D image coordinates:

$$h \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = K\boldsymbol{x}_c = \begin{pmatrix} s\,f_x & 0 & x_c & 0 \\ 0 & s\,f_y & y_c & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix}. \tag{3.1}$$

The observed image coordinates (x_i, y_i) are distorted by lens effects. They are transformed to ideal screen coordinates (u, v) by the following equations:

$$d^2 = (x_i - x_0)^2 + (y_i - y_0)^2 \quad p = (1 - fd^2) \tag{3.2}$$

$$u = p(x_i - x_0) + x_0 \tag{3.3}$$

$$v = p(y_i - y_0) + y_0 \tag{3.4}$$

The image processing flow, performed by ARToolKit, is illustrated in Fig. 3.2.

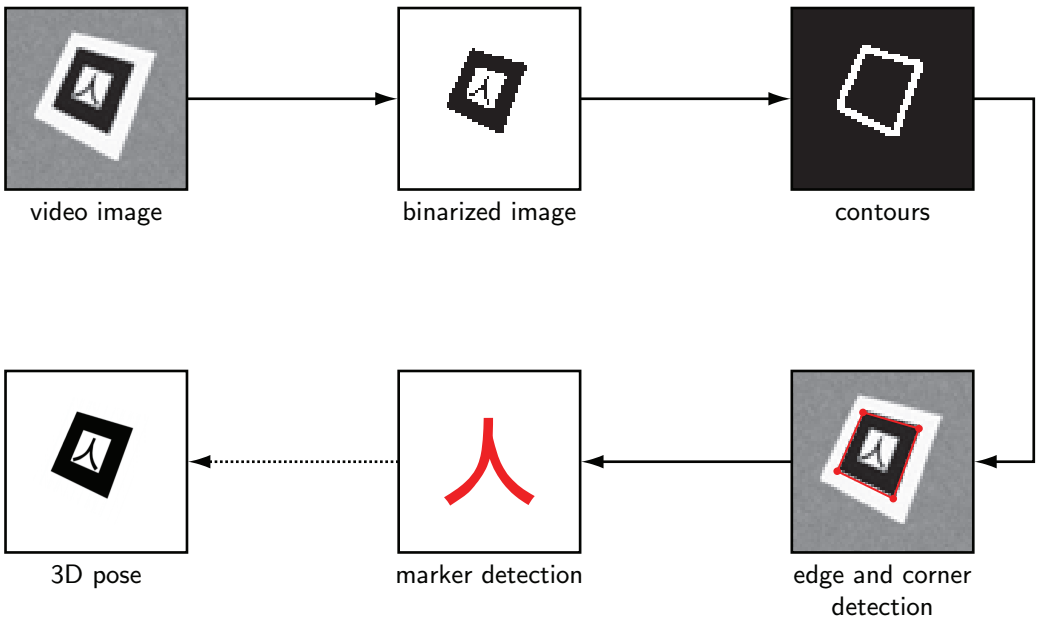

Figure 3.2: Functional Diagram of ARToolKit

1. The grayscale image is transformed to a black-and-white image by using a dynamic binarization threshold.
2. The black frames on white background are searched for and their contours are extracted
3. From the contour image, the edges and corner of the square frame are determined
4. The center area of the marker is rectified according to the detected edges and compared to reference patterns from a marker database
5. Finally, the 3D pose of the identified marker can be computed

While the ARToolKit can compute the 3D pose of a single marker, it cannot fuse marker points from several markers nor make use of multiple cameras. Therefore, only the 2D image coordinates produced by ARToolKit are used in this thesis as inputs to the Orthogonal Iteration Algorithm.

Performance evaluation

The performance of the whole pose estimation system is largely determined by the accuracy with which the marker points can be localized in the camera image. This localization accuracy is experimentally assessed, and the results are illustrated in Fig. 3.3.

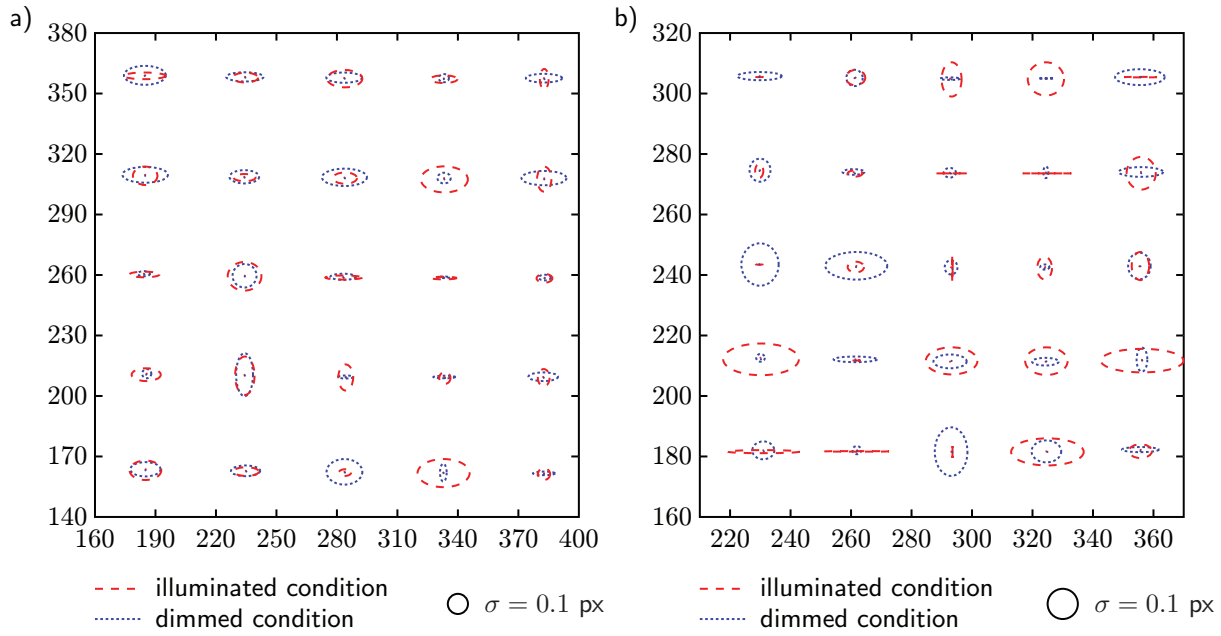

Figure 3.3: Localization accuracy of ARToolKit. The diagrams show the standard deviation of the pixel coordinates obtained from 2×15 snapshots per location, where one series is recorded with illuminated marker and the other series is recorded with dimmed marker. In Fig. a) and Fig. b) differ in the distance between camera and marker, which is larger for Fig. b). Note the different scaling of the axes.

The camera is oriented such that its viewing direction is perpendicular to the flat marker, and the image axes are aligned with the axes of the marker. In this orientation the camera is positioned at 25 locations in a 5×5 grid. At each location, 5 consecutive measurements of the marker position are performed. The measurement series is repeated three times yielding a total of 15 measurements per location. In Fig. 3.3, the mean and the standard deviation of these 15 measurements are plotted. The whole procedure is repeated four times, where two distances between camera and marker as well as two illumination conditions are covered. The results show that systematic errors are cancelled out by the camera calibration and that the imposed illumination changes have no significant effect on the localization accuracy. As expected, the pixel accuracy is independent of the distance between camera and marker, which translates to a metric lateral accuracy which is inversely proportional to the distance.

3.3 Pose Estimation

The classic pose estimation problem consists in finding the relative pose between a camera and an object of known geometry based on the camera images of the object. In mathematical terms: Given n 3D points $\boldsymbol{p}_i$ in an object fixed coordinate system and n corresponding 2D image points $\boldsymbol{v}_i$, find the rotation matrix $\boldsymbol{R}$ and the translation vector $\boldsymbol{t}$ which relate object fixed coordinates to camera fixed coordinates.

3.3.1 State of the Art

As this problem plays an important role in computer vision and robotics, a large body of related research exists.

Analytical solutions to the pose estimation problem exist only for a small number of points. For three points, the solution can be efficiently calculated, but in general four possible solutions exist [25, 40]. By adding a forth point, the ambiguity can be resolved for non-degenerate cases [26, 30, 35, 45]. If the image points can be determined with little noise, the analytical methods provide accurate pose estimates. However, for noisy measurements or ill-posed configurations, a higher number of points is needed to obtain accurate results.

For more than four points, the problem becomes a non-linear least squares optimization problem. The solutions mostly rely on the Gauss-Newton [65] or Levenberg-Marquardt [50] method. Therefore, they are typically slow and cannot be proven to converge. More advanced solutions take the orthogonality properties of the rotation matrix into account [40].

A very powerful pose estimation technique is introduced with the *orthogonal iteration algorithm* [67]. The algorithm is proven to be globally convergent and is empirically tested to efficiently converge to the optimal solution. As the algorithm is designed to deal with a high number of feature point, it is robust to image noise.

With dropping costs of image acquisition hardware, solutions which use more than one camera are gaining more and more attention. The classic approach of pose estimation

with multiple cameras in a fixed configuration consists in obtaining a set of 3D-points from the images by triangulation and subsequently reconstructing the relative pose between the camera frame and the object. While this approach can be solved analytically and in a computationally efficient way, it can only use points which are seen in at least two camera images. Only a few methods consider the problem of finding the pose directly from multiple camera images [33, 84]. However, none of these methods has a comparable performance to the orthogonal iteration algorithm.

3.3.2 Orthogonal Iteration Algorithm

The orthogonal iteration algorithm makes the following assumptions. There are $n \geq 3$ points $\boldsymbol{p}_i = (x_i, y_i, z_i)^T$, $i = 1 \dots n$ defined in an object fixed coordinate system. The corresponding coordinates $\boldsymbol{q}_i = (x'_i, y'_i, z'_i)^T$ in a camera fixed coordinate system system are obtained by a rigid transformation[1]:

$$\boldsymbol{q}_i = \boldsymbol{R}\boldsymbol{p}_i + \boldsymbol{t}, \tag{3.5}$$

where $\boldsymbol{R}$ is a 3×3 rotation matrix, and $\boldsymbol{t}$ is a translation vector.

Under the assumption of an ideal pinhole camera, the corresponding image coordinates $\boldsymbol{v}_i = (u_i, v_i, 1)^T$ are:

$$u_i = \frac{x'_i}{z'_i} \qquad v_i = \frac{y'_i}{z'_i} \tag{3.6}$$

The quantization effects of the camera and uncompensated lens distortions lead to errors between these idealized image points and the observed image points $\hat{\boldsymbol{v}}_i$. The difference between ideal and observed image point coordinates is called *image-space collinearity error*:

$$\boldsymbol{\varepsilon}_i = \boldsymbol{v}_i - \hat{\boldsymbol{v}}_i. \tag{3.7}$$

An alternative way of expressing the error between ideal and observed image point is given by the *object-space collinearity error*. Each image point $\boldsymbol{v}_i$ defines a line-of-sight projection matrix $\boldsymbol{V}_i$, which yields the projection of an arbitrary point on the ray which is cast by the respective image point. For the observed image point $\hat{\boldsymbol{v}}_i$, the observed line-of-sight projection matrix is expressed by:

$$\hat{\boldsymbol{V}}_i = \frac{\hat{\boldsymbol{v}}_i \hat{\boldsymbol{v}}_i^T}{\hat{\boldsymbol{v}}_i^T \hat{\boldsymbol{v}}_i}. \tag{3.8}$$

Based on the projection matrix $\hat{\boldsymbol{V}}_i$, the object-space collinearity error can be calculated for a given $\boldsymbol{R}$ and $\boldsymbol{t}$:

$$\boldsymbol{e}_i = (\boldsymbol{I} - \hat{\boldsymbol{V}}_i)(\boldsymbol{R}\boldsymbol{p}_i + \boldsymbol{t}) \tag{3.9}$$

The two ways of expressing collinearity errors as well as the camera-fixed and the object-fixed coordinate system are illustrated in Fig. 3.4.

[1] In computer vision and computer photogrammetry, the coordinate systems are usually oriented such that the x-axis is pointing to the left, the y-axis is pointing up, and the z-axis is pointing in direction of the optical axis. For reasons of consistency, this notation is also used in this section (cf. Fig. 3.4). However, in the evaluation, robot coordinates are used as defined in Sec. 3.1.

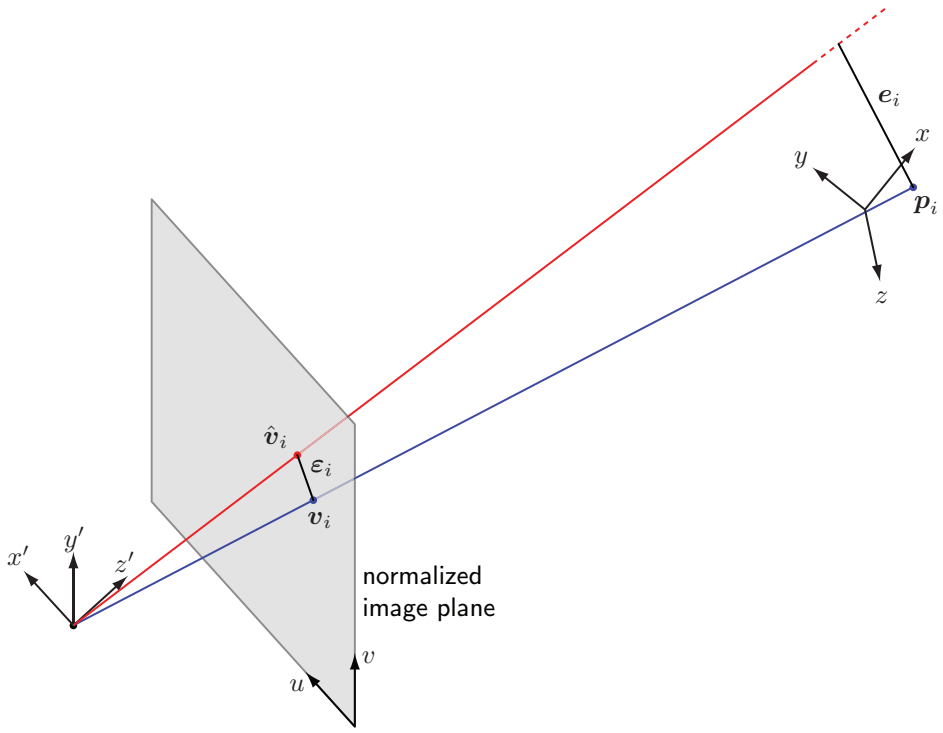

Figure 3.4: Definition of the camera coordinate system, the object coordinate system, and the normalized image plane. The representative 3D point $\boldsymbol{p}_i$ is projected onto the image plane as 2D point $\boldsymbol{v}_i$. Due to quantization effects and unmodeled distortions, the observed image point $\hat{\boldsymbol{v}}_i$ may differ. The corresponding image-space collinearity error is denoted $\boldsymbol{\varepsilon}_i$, the object-space collinearity error is denoted $\boldsymbol{e}_i$ [67].

Absolute Orientation Problem

As a precursor to the solution of the pose estimation problem, the solution to a simplified problem, the *absolute orientation problem*, is described: Given n 3D points $\boldsymbol{p}_i$, $i = 1 \dots n$ in object coordinates and n corresponding 3D points $\boldsymbol{q}_i$ in camera coordinates, find the optimal transformation $\boldsymbol{R}$ and $\boldsymbol{t}$, which minimizes the errors of (3.5) in a least-squares sense:

$$G_{AOP}(\boldsymbol{R}, \boldsymbol{t}) = \sum_{i=1}^{n} \|\boldsymbol{R}\boldsymbol{p}_i + \boldsymbol{t} - \boldsymbol{q}_i\|^2 \tag{3.10}$$

There exist various approaches to the absolute orientation problem [46, 47]. Here, a *singular value decomposition (SVD)* is employed to solve the problem [10] because of its simplicity and computational efficiency. The solution is based on the insight that the rotation matrix $\boldsymbol{R}$ can be determined solely based on the positions $\boldsymbol{p}_i$ and $\boldsymbol{q}_i$ relative to their respective centroids $\bar{\boldsymbol{p}}$ and $\bar{\boldsymbol{q}}$. The centroids are calculated by

$$\bar{\boldsymbol{p}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{p}_i, \quad \bar{\boldsymbol{q}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{q}_i, \tag{3.11}$$

and the corresponding relative positions $\boldsymbol{p}'_i$ and $\boldsymbol{q}'_i$ are

$$\boldsymbol{p}'_i = \boldsymbol{p}_i - \bar{\boldsymbol{p}}, \quad \boldsymbol{q}'_i = \boldsymbol{q}_i - \bar{\boldsymbol{q}}. \tag{3.12}$$

With the definition of the matrix $\boldsymbol{M}$ as

$$\boldsymbol{M} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{q}'_i \boldsymbol{p}'^T_i, \tag{3.13}$$

the optimal rotation matrix $\boldsymbol{R}^*$ can be expressed by

$$\boldsymbol{R}^* = \arg\max_{\boldsymbol{R}} \operatorname{tr}(\boldsymbol{R}^T \boldsymbol{M}) \tag{3.14}$$

Finally, the value of $\boldsymbol{R}^*$ is obtained by singular value decomposition:

$$\boldsymbol{R}^* = \boldsymbol{W}\boldsymbol{U}^T, \tag{3.15}$$

where $\boldsymbol{U}^T\boldsymbol{M}\boldsymbol{W} = \boldsymbol{\Sigma}$ form an SVD. When the point set consists of coplanar points, the resulting matrix $\boldsymbol{R}^*$ can be a reflection matrix instead of a rotation matrix. As the corners of a single fiducial always produce a coplanar point set, these cases must be detected and corrected [10]. Presuming that the singular values in $\boldsymbol{\Sigma}$ are ordered in descending order, the modified version of (3.15) is calculated:

$$\boldsymbol{R}^{**} = \boldsymbol{W} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \boldsymbol{U}^T. \tag{3.16}$$

If $\boldsymbol{R}^*$ is a rotation matrix, i.e. $\det(\boldsymbol{R}^*) = 1$, than $\boldsymbol{R}^{**}$ is a reflection matrix, i.e. $\det(\boldsymbol{R}^{**}) = -1$, and vice versa. Therefore, the one matrix which is a rotation matrix is chosen:

$$\hat{\boldsymbol{R}} = \begin{cases} \boldsymbol{R}^*, & \text{if } \det(\boldsymbol{R}^*) = 1 \\ \boldsymbol{R}^{**}, & \text{if } \det(\boldsymbol{R}^*) = -1 \end{cases} \tag{3.17}$$

The estimated rotation matrix $\hat{\boldsymbol{R}}$ directly yields the estimated translation vector $\hat{\boldsymbol{t}}$:

$$\hat{\boldsymbol{t}} = \bar{\boldsymbol{q}} - \hat{\boldsymbol{R}}\bar{\boldsymbol{p}}, \tag{3.18}$$

such that the absolute orientation problem is completely solved.

Iteration Procedure

In contrast to the absolute orientation problem, the 3D-points $\boldsymbol{q}_i$ in camera coordinates are not given in the classic pose estimation problem. Instead, only their observed 2D projections $\hat{\boldsymbol{v}}_i$ or, equivalently, their line-of-sight projection matrices $\hat{\boldsymbol{V}}_i$ are available.

Thus, in the pose estimation problem, the minimization is not performed on the Euclidean distances between 3D points (3.10), but on the object-space collinearity errors $\|\boldsymbol{e}_i\|$ as defined in (3.9):

$$G_{OCE}(\boldsymbol{R},\boldsymbol{t}) = \sum_{i=1}^{n} \|\boldsymbol{e}_i\|^2 = \sum_{i=1}^{n} \|(\boldsymbol{I} - \hat{\boldsymbol{V}}_i)(\boldsymbol{R}\boldsymbol{p}_i + \boldsymbol{t})\|^2 \tag{3.19}$$

This optimization problem cannot be solved analytically, but there exists a globally convergent iterative solution [67]. Starting with an initial estimate of the rotation matrix $\hat{\boldsymbol{R}}^{(k)}$, the corresponding estimate of the translation vector $\hat{\boldsymbol{t}}^{(k)}$ can be analytically calculated from (3.19), because the cost function is quadratic in $\boldsymbol{t}$:

$$\hat{\boldsymbol{t}}^{(k)} = \boldsymbol{t}(\hat{\boldsymbol{R}}^{(k)}) = \frac{1}{n}\left(\boldsymbol{I} - \frac{1}{n}\sum_{i=1}^{n}\hat{\boldsymbol{V}}_i\right)^{-1}\sum_{i=1}^{n}(\hat{\boldsymbol{V}}_i - \boldsymbol{I})\hat{\boldsymbol{R}}^{(k)}\boldsymbol{p}_i \tag{3.20}$$

By using the estimates of the rigid transformation between object and camera coordinate systems $\hat{\boldsymbol{R}}^{(k)}$ and $\hat{\boldsymbol{t}}^{(k)}$, the known points $\boldsymbol{p}_i$ in object-fixed coordinates can be transformed to the camera coordinate system. Subsequently, they are projected onto the corresponding lines-of-sight by means of the line-of-sight projection matrix $\hat{\boldsymbol{V}}_i$:

$$\hat{\boldsymbol{q}}_i^{(k)} = \hat{\boldsymbol{V}}_i(\hat{\boldsymbol{R}}^{(k)}\boldsymbol{p}_i + \hat{\boldsymbol{t}}^{(k)}) \tag{3.21}$$

These estimates $\hat{\boldsymbol{q}}_i^{(k)}$ are inserted in (3.10) to yield a new estimate of the rotation matrix $\hat{\boldsymbol{R}}^{(k+1)}$

Summarizing, each iteration k of the orthogonal iteration algorithm proceeds in three steps, where an initial guess of the rotation matrix $\hat{\boldsymbol{R}}^{(k)}$ is assumed:

1. Calculate a new approximation $\hat{\boldsymbol{t}}^{(k)}$ by minimizing the mean squared object-space collinearity error (cf. (3.20)).
2. Calculate the projected points $\hat{\boldsymbol{q}}_i^{(k)}$ based on the rigid transformation $\hat{\boldsymbol{R}}^{(k)}$, $\hat{\boldsymbol{t}}^{(k)}$ and the observed line-of-sight projection matrices $\hat{\boldsymbol{V}}_i$ (cf. (3.21)).
3. Calculate a new approximation $\hat{\boldsymbol{R}}^{(k+1)}$ by solving the absolute orientation problem defined by $\boldsymbol{p}_i$ and $\hat{\boldsymbol{q}}_i^{(k)}$ (cf. (3.10)).

3.3.3 Extended Orthogonal Iteration Algorithm

Depending on the orientation between target object and camera, the pose estimates may become inaccurate due to quantization noise. Thus, it is favorable to use more than one camera, which yields images of the target object from different perspectives. The differences in the viewing perspective become the larger the closer the object is approached with the cameras. Thus, the accuracy reaches its maximum, when it is most needed.

As stated above, there are two possible approaches for pose estimation in multi-camera systems. The prevailing approach uses some kind of stereo-matching to obtain 3D coordinates of the markers, which than constitute an absolute orientation problem. However, this approach requires the marker points to be visible in two or more images to be used. In contrast, the here presented extension to the orthogonal iteration algorithm can deal with fully disjunct point sets.

With the introduction of multiple cameras, multiple coordinate systems come into play (Fig. 3.5). The transformation $\boldsymbol{R}$ and $\boldsymbol{t}$ now describes the relation between 3D points in object coordinates $\boldsymbol{p}_i$ and world coordinates $\boldsymbol{q}_i$. Additionally, a rigid transformation for each camera j is given by $\boldsymbol{R}_j$ and $\boldsymbol{t}_j$ such that the object coordinates $\boldsymbol{p}_i$ map to camera coordinates $\boldsymbol{r}_{i,j} = (x''_{i,j}, y''_{i,j}, z''_{i,j})^T$ according to the the following transformation chain:

$$\boldsymbol{p}_i \xrightarrow{\boldsymbol{R},\boldsymbol{t}} \boldsymbol{q}_i \xrightarrow{\boldsymbol{R}_j,\boldsymbol{t}_j} \boldsymbol{r}_{i,j}. \tag{3.22}$$

For minimizing the object-space collinearity, the mathematical description will be used:

$$\boldsymbol{r}_{i,j} = \boldsymbol{R}_j\boldsymbol{q}_i + \boldsymbol{t}_j = \boldsymbol{R}_j\boldsymbol{R}\boldsymbol{p}_i + \boldsymbol{R}_j\boldsymbol{t} + \boldsymbol{t}_j. \tag{3.23}$$

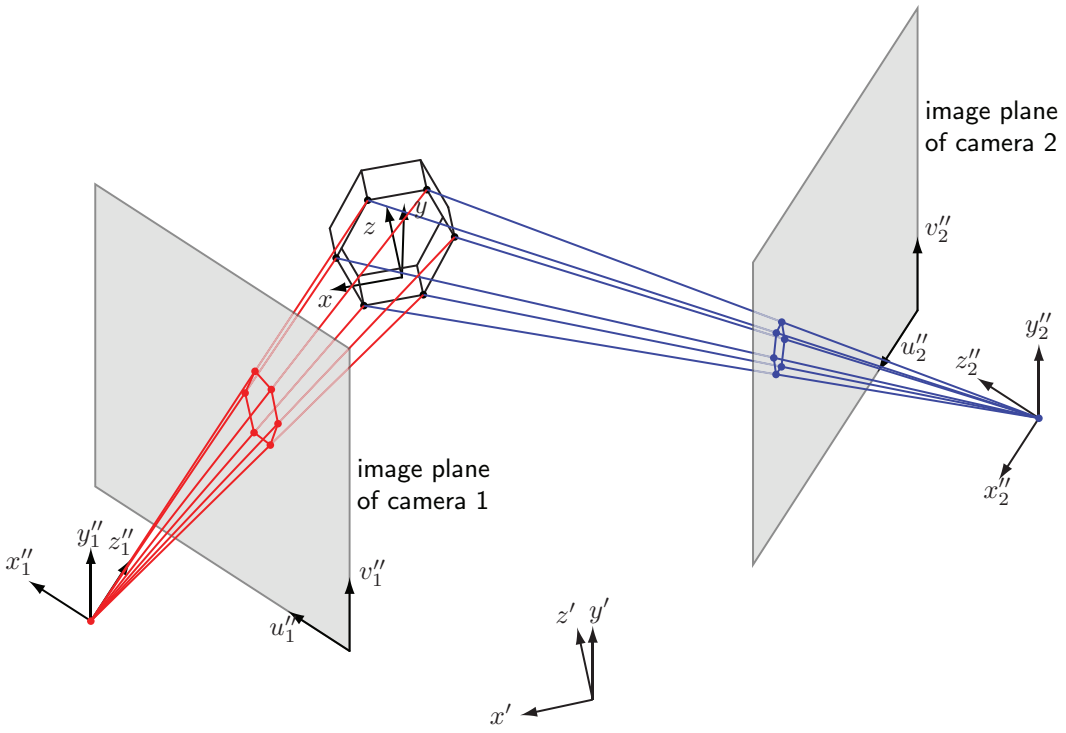

Figure 3.5: World coordinate system and local coordinate systems of two cameras and the observed object. The feature points are projected onto both image planes yielding the projected point sets $\boldsymbol{v}_{i,1}$ and $\boldsymbol{v}_{i,2}$.

With these definitions, the pose estimation problem can be reformulated for a number of object points n_p and a number of cameras n_c. The total number of all observed object points is $n \leq n_p n_c$. The k-th observed object point is described by the index $i_k = 1 \ldots n_p$, which denotes object point coordinates $\boldsymbol{p}_{i_k}$, and the index $j_k = 1 \ldots n_c$, which specifies the camera and thus determines the camera pose $\boldsymbol{R}_{j_k}$, $\boldsymbol{t}_{j_k}$, where $k = 1 \ldots n$.

In order to simplify the notation, the observed projection error matrix $\hat{\boldsymbol{E}}_k$ is defined as

$$\hat{\boldsymbol{E}}_k = \boldsymbol{I} - \hat{\boldsymbol{V}}_{i_k,j_k}. \tag{3.24}$$

The object-space collinearity error $\boldsymbol{e}_k$ of the k-th observed object point, i.e. of object point i_k observed by camera j_k is expressed by

$$\boldsymbol{e}_k = \boldsymbol{e}_{i_k,j_k} = \hat{\boldsymbol{E}}_k \left(\boldsymbol{R}_{j_k} \boldsymbol{R} \boldsymbol{p}_{i_k} + \boldsymbol{R}_{j_k} \boldsymbol{t} + \boldsymbol{t}_{j_k} \right). \tag{3.25}$$

The new cost function is an extension of (3.19) such that the object-space collinearity errors of all observed object points $k = 1 \ldots n$ are minimized in a least-squares sense:

$$G'_{OCE}(\boldsymbol{R}, \boldsymbol{t}) = \sum_{k=1}^{n} \|\boldsymbol{e}_k\|^2 = \sum_{k=1}^{n} \| \hat{\boldsymbol{E}}_k \left(\boldsymbol{R}_{j_k} \boldsymbol{R} \boldsymbol{p}_{i_k} + \boldsymbol{R}_{j_k} \boldsymbol{t} + \boldsymbol{t}_{j_k} \right) \|^2. \tag{3.26}$$

The optimal translation $\hat{\boldsymbol{t}}$ can be expressed as a function of the estimated rotation matrix $\hat{\boldsymbol{R}}$:

$$\hat{\boldsymbol{t}}(\hat{\boldsymbol{R}}) = \arg\min_{\boldsymbol{t}} G'_{OCE}(\hat{\boldsymbol{R}}, \boldsymbol{t}) \tag{3.27}$$

Just as its counterpart for the single-camera case (3.19), the full optimization problem in (3.19) is solved iteratively. Starting from an initial estimate of the rotation matrix $\hat{\boldsymbol{R}}^{(l)}$, the translation vector $\hat{\boldsymbol{t}}^{(l)}$ can be analytically calculated. The linear least-squares problem in (3.27), can be most conveniently solved by applying the pseudoinverse of $\boldsymbol{C}$.

With the definitions

$$\boldsymbol{C} = \begin{pmatrix} -\boldsymbol{E}_1 \boldsymbol{R}_{j_1} \\ -\boldsymbol{E}_2 \boldsymbol{R}_{j_2} \\ \vdots \\ -\boldsymbol{E}_n \boldsymbol{R}_{j_n} \end{pmatrix} \quad \boldsymbol{s}(\boldsymbol{R}) = \begin{pmatrix} \boldsymbol{E}_1 \left(\boldsymbol{R}_{j_1} \boldsymbol{R} \boldsymbol{p}_{i_1} + \boldsymbol{t}_{j_1} \right) \\ \boldsymbol{E}_2 \left(\boldsymbol{R}_{j_2} \boldsymbol{R} \boldsymbol{p}_{i_2} + \boldsymbol{t}_{j_2} \right) \\ \vdots \\ \boldsymbol{E}_n \left(\boldsymbol{R}_{j_n} \boldsymbol{R} \boldsymbol{p}_{i_n} + \boldsymbol{t}_{j_n} \right) \end{pmatrix}, \tag{3.28}$$

the solution for $\boldsymbol{t}$ can be directly obtained:

$$\hat{\boldsymbol{t}}^{(l)} = \boldsymbol{t}(\hat{\boldsymbol{R}}^{(l)}) = (\boldsymbol{C}^T \boldsymbol{C})^{-1} \boldsymbol{C}^T \boldsymbol{s}(\hat{\boldsymbol{R}}^{(l)}). \tag{3.29}$$

As the dependence on the current estimate $\hat{\boldsymbol{R}}^{(l)}$ is fully encoded in $\boldsymbol{s}$, the matrix $\boldsymbol{C}$ and its pseudoinverse do not need to be recomputed in every iteration, which makes the method computationally efficient.

Finally, the iteration loop is closed by calculating a new estimate $\hat{\boldsymbol{R}}^{(l+1)}$ by solving the extended absolute orientation problem:

$$G'_{AOP}(\boldsymbol{R}, \boldsymbol{t}) = \sum_{k=1}^{n} \| \boldsymbol{R}_{j_k} \boldsymbol{R} \boldsymbol{p}_{i_k} + \boldsymbol{R}_{j_k} \boldsymbol{t} + \boldsymbol{t}_{j_k} - \boldsymbol{r}_{i_k,j_k} \|^2 \tag{3.30}$$

3.3.4 Simulation Results

In order to document the theoretical estimation accuracy of the extended orthogonal iteration algorithm in the presence of Gaussian pixel noise, simulation results are presented. The simulations are performed and compared for setups with one camera and with two cameras, respectively.

For the simulation, two ideal pin-hole cameras are assumed. Perturbation of the observed image points is produced by adding a Gaussian noise with a signal-to-noise ratio (SNR) in the range of 20 dB to 50 dB. The cameras are arranged in a symmetric configuration such that their coordinate systems are rotated by $-45°$ and $45°$, respectively, around the y-axis w.r.t. the world coordinate system. The optical axes of both cameras intersect in the origin of the world coordinate system, and the distance between the centers of projection and the origin are set to 0.1 m. A coplanar set of object points is used in the simulation. This complies with the typical feature sets of a screwhead as well as with the corners of a flat artificial marker. In the single camera simulations, twelve equally spaced points on a circle of 0.01 m diameter form the object point sets. In the multi camera simulations, only six equally spaced points on a circle of the same size are used. Thus, the total number of observed points is kept constant. The circle lies in the x-y-plane of the object and its center point coincides with the origin of the world coordinate system.

In Fig. 3.6, the simulation results are illustrated. As a correct estimate of the rotation yields an optimal estimate of the translation in a least-squares sense, only the rotational error is shown. This is obtained by comparing the true rotation $\boldsymbol{R}$ and the estimated rotation matrix $\hat{\boldsymbol{R}}$:

$$\phi = \operatorname{acos}\frac{\operatorname{tr}(\boldsymbol{R}^T\hat{\boldsymbol{R}}) - 1}{2} \tag{3.31}$$

The plotted errors are averaged results from 1000 trials. For each trial, 200 iterations of the orthogonal iteration algorithm are performed. Empirically, this number of iterations is enough to ensure a convergence to a precision of 10^{-5}. One trial takes about 4 ms to compute on an AMD Athlon 3700+ processor running an optimized C++ implementation of the algorithm.

In one series, the object is rotated around its x-axis, which is identical with the x-axis of the world-coordinate system. Due to the symmetric configuration of the two cameras, also the two camera images are symmetric in the unperturbed case. For the angles $\alpha = 90°$ and $\alpha = 270°$, all object points lie in the plane spanned by the optical axes of the two cameras. Thus, the projected points lie on a line in the image plane. If the object is observed by only one camera, this configuration leads to the highest estimation errors as shown in Fig. 3.6 a). In the multi-camera setup, however, the estimation errors are even lower in these configurations, see Fig. 3.6 b). This indicates that the extended orthogonal iteration algorithm indirectly obtains depth information from point pairs which in turn lead to a higher estimation accuracy. Generally, the estimation errors of the multi-camera setup are approximately one order of magnitude lower than those of the single-camera setup.

In a complementary series, the object is rotated around its y-axis, which is identical with the y-axis of the world-coordinate system. During this rotation, the projection of the point set cannot degenerate to a line in both cameras at the same time. Thus, the results

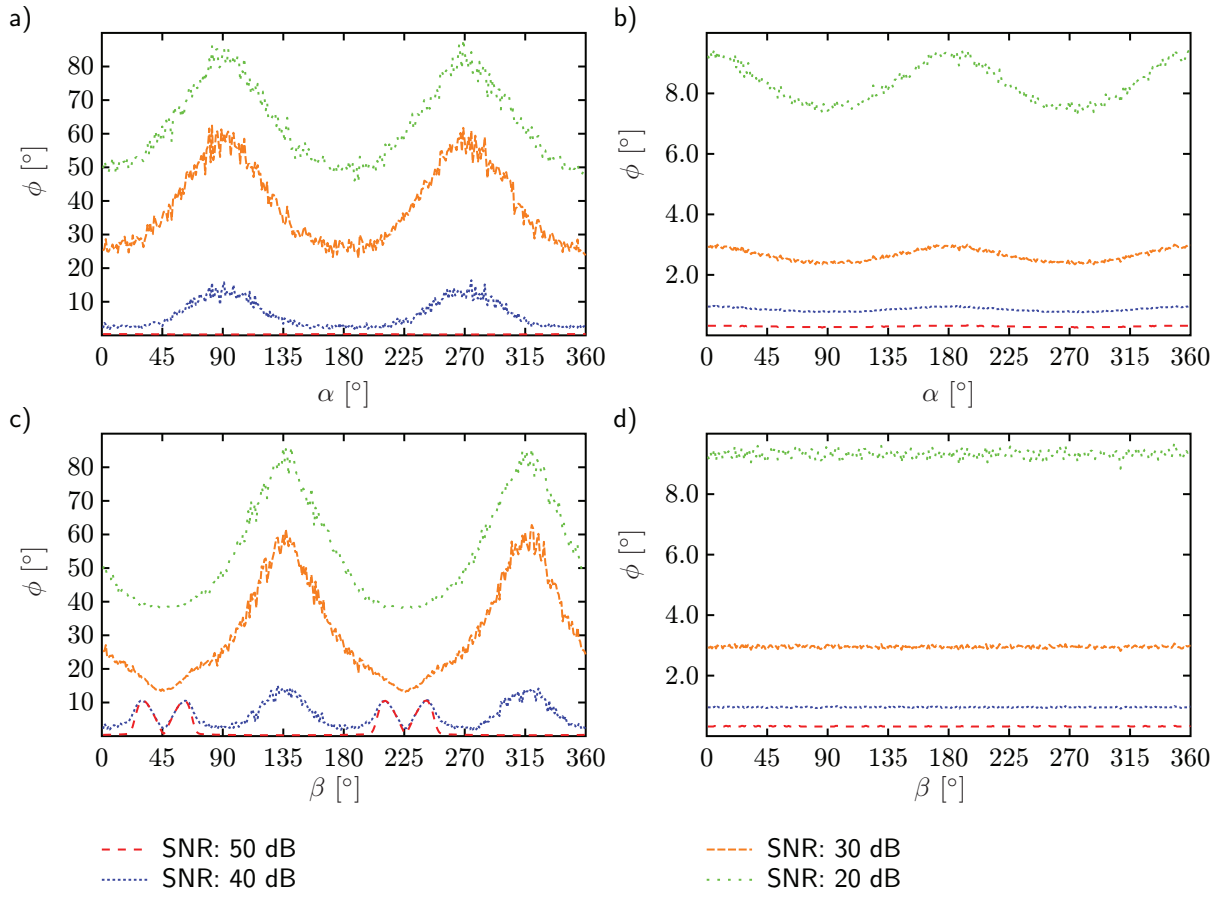

Figure 3.6: Comparison of the estimation accuracy for pose estimation with a single camera vs. multiple cameras. The rotational estimation error ϕ is shown for different amounts of pixel noise. On the left hand side, in Figs. a) and c), results for a single camera setup are presented. On the right hand side, in Figs. b) and d), corresponding results for a setup with two cameras are shown. The plots in the top row, i.e. Figs. a) and b), are created by rotating the object around the x-axis by α. The plots in the bottom row, i.e. Figs. c) and d), are created by rotating the object around the y-axis by β. Note the different scaling of the ϕ axis.

can be expected to be comparable to the rotation around the x-axis for the single-camera setup. For the multiple-camera setup, however, the estimation accuracy is expected to remain constant. These expectations are confirmed by Figs. 3.6c) and d). Due to the fixed rotation of the camera itself, the maximum errors are reached at angles of $\beta = 135°$ and $\beta = 315°$. Again, the multi-camera setup is approximately one order of magnitude more accurate than the single-camera setup.

3.4 Experimental Evaluation

In Sec. 3.2 and Sec. 3.3, the accuracy of the marker localization in the camera image and the simulated accuracy of the extended orthogonal iteration were assessed. In this section,

the position measurement accuracy of the complete camera-based pose estimation system illustrated in Fig. 3.1 is evaluated.

In order to measure the accuracy of the camera system, the end-effector with the cameras is sequentially positioned at 125 positions in an orthogonal $5 \times 5 \times 5$ grid in 3D space. The distance of the grid points is 5 mm in each direction. Thus, the covered volume is 20 mm $\times$ 20 mm $\times$ 20 mm. The size of the artifical marker is 20 mm $\times$ 20 mm. The whole sequence is repeated three times leading to a total number of 375 measurement points. The error $\boldsymbol{e}$ is calculated as the difference between the position which is commanded to the robot and the position which is estimated by the camera system.

The results are illustrated in Fig. 3.7, where the accuracy of a setup with two cameras, Fig. a), is compared to a setup with one camera, Fig. b). In both figures, the norm of the error $\|\boldsymbol{e}\|$ is shown as a function of the x-, y-, and z-coordinate, respectively.

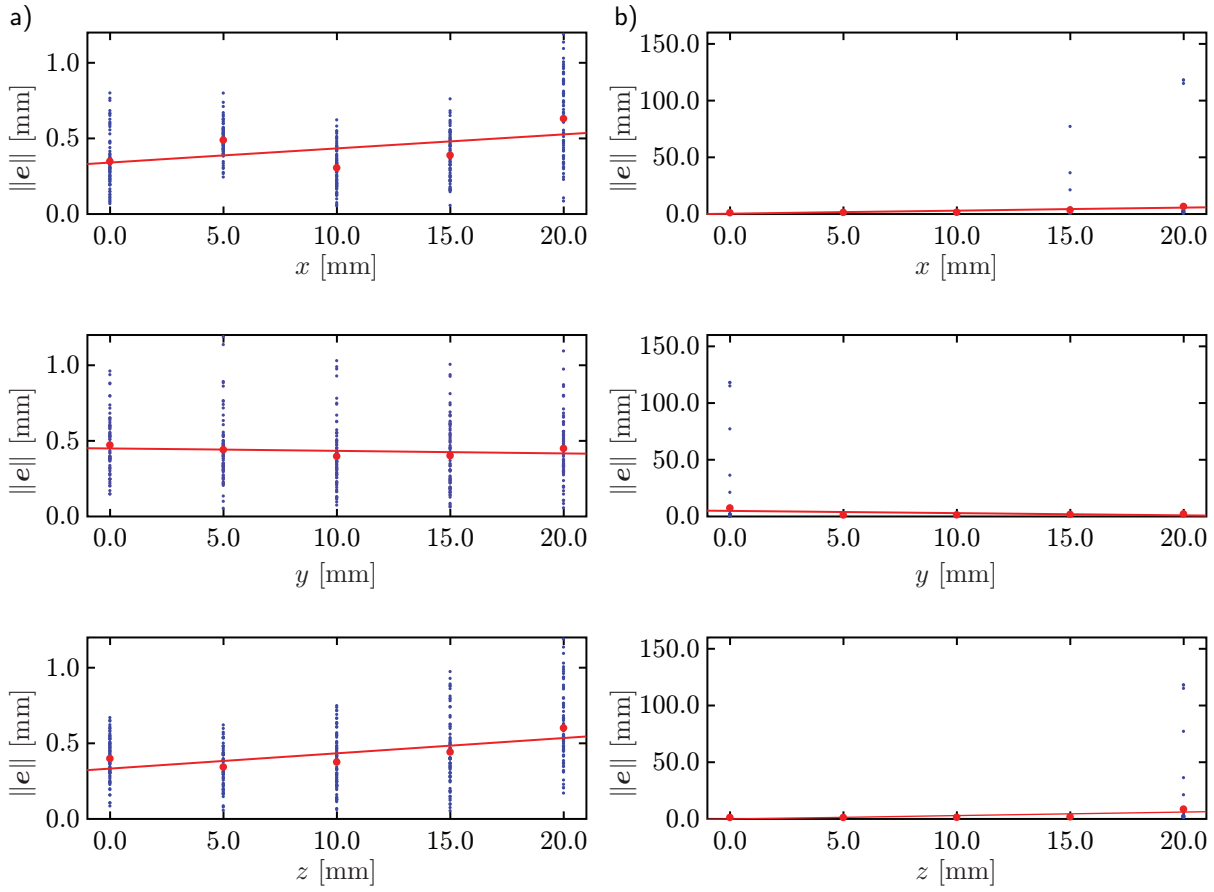

Figure 3.7: Comparison of estimation accuracy with one and two cameras. In Fig. a), the results of a setup with two cameras are illustrated; in Fig. b), the results of a setup with one camera are illustrated. The norm of the Cartesian error $\|\boldsymbol{e}\|$ is plotted as a function of the x-, y-, and z-coordinate, respectively. For each position, the mean error is shown. A line is fitted to the mean errors in order to detect dependencies with the coordinate values.

The most obvious observation from the figures is that the one-camera setup produces some outliers, whereas the two-camera setup does not. Furthermore, the accuracy of the one-camera setup is lower than the accuracy of the two-camera setup by a factor of approx. 6.

For the two-camera setup, the accuracy is independent of the position in y-direction, while it decreases with increasing x- and z-coordinate. As the camera system is tilted by 45° around the y-axis, the distance between cameras and marker increases linearly with the x- and z-coordinate, and the decreasing accuracy can be attributed to the increasing distance to the marker. The maximum error of the two-camera system is 1.2 mm, and the standard deviation is below 0.5 mm. The obtained accuracy fully satisfies the requirements of the screwdriving task. For other tasks which require an even higher accuracy, the position measurement accuracy of the camera system can be increased by using higher-quality cameras with higher-quality lenses. Furthermore, the calibration of intrinsic as well as extrinsic camera parameters is subject to improvements.

3.5 Discussion

In order to support the human operator in achieving the intended goal, the assisted telepresence system must possess knowledge about the location of possible manipulation targets in the remote environment. This knowledge is exploited twice. On the one hand, possible targets are used as hypotheses in the intention estimation process. On the other hand, the pose of the selected target is used as destination in the trajectory generation process. The extraction of possible targets can be performed by using visual data from the remote scene. This chapter extends the state of the art in two ways. Firstly, a hardware setup and software toolchain is proposed which enables precise visual scene recognition by the teleoperator. Secondly, a new pose estimation algorithm for multi-camera setups is presented which significantly outperforms existing solutions.

By means of the stereoscopic camera head and the stereoscopic eye-in-hand camera, the system is able to coarsely identify targets in an early stage and to precisely determine the pose of a specific target when it is closely approached. Using artificial markers, which are extracted from the scene by the ARToolKit library, poses can be measured with an accuracy of better than 0.5 mm.

The main contribution of the chapter lies in the development and assessment of a new method for estimating the 3D pose of a known object based on its 2D projection to the image planes of multiple cameras. The algorithm emerges as an extension to the orthogonal iteration algorithm, which solves the pose estimation problem for a single camera. Simulation results show that the presented extended orthogonal iteration algorithm delivers accurate results even for ill-posed point sets, e.g. coplanar points.

Finally, a practical evaluation confirms the accuracy of the pose estimation results which are produced by integrating the presented subsystems. The scene recognition system can provide coarse estimate of the target poses at an early stage. When the target is approached, these estimates are continuously refined based on the image data provided by the eye-in-hand cameras.

Besides its usage in an assisted telepresence system, the presented setup and pose estimation algorithm can be applied to a wide range of problems from classical computer photogrammetry to autonomous robotics. Due to its robustness, it relaxes the requirements on the cameras and lenses and can thus help to reduce costs.

4 Assist Functions

In the previous chapters, the foundations for providing assistance to the operator in an intuitive manner are laid: The current intention of the human operator is classified and expressed in form of a predicted trajectory, and a precise reference trajectory to the target which the operator is actually aiming to reach is calculated. Based on this knowledge about the current task, the assist functions are supposed to augment the interchanged position and force signals such that the task performance and the feeling of presence are improved. The stability of the telepresence system must be maintained by the assist functions.

While the concept of intuitive assistance is very general, the actual design and implementation of the assist functions is task dependent. In accordance with the previous chapters, the screwdriving task serves as application example. The assistance which is most needed for this task is twofold: firstly, the approach and alignment phase must be optimized in such a way that the screwdriver is exactly positioned in front of the screwhead; secondly, the contact between screwdriver and screwhead must be established without bouncing and excessive forces. Generalizing, the screwdriving task is regarded as a peg-in-hole task, and the approach movement is regarded as an unconstrained point-to-point movement.

For point-to-point movements, which are part of the peg-in-hole task, the movement can be decomposed into a longitudinal and a lateral movement as depicted in Fig. 4.1. The longitudinal component is oriented tangentially to the trajectory; the lateral component is oriented perpendicularly to the trajectory. As unconstrained point-to-point movements are implemented by humans as straight lines, only the case of straight reference paths, illustrated in Fig. 4.1 a), is relevant for the following considerations.

The model of human point-to-point movements, given in Chapter 2, describes the spatial as well as the temporal course of the movement, i.e. it describes the curve on which the human travels, and it describes the progression on this curve as a function of time. When the human behaves exactly in accordance with the model, no deviation occurs, neither on the spatial nor on the temporal course. Accordingly, two types of deviations can be distinguished: The human does not travel on the modeled curve, or he does not follow the modeled speed profile, where both types of deviations can occur at the same time. It is impossible to uniquely determine the type of deviation from the observed data. However, it is reasonable to assign longitudinal differences between the observed and the modeled trajectory to temporal deviations and lateral differences to spatial deviations.

The correction of spatial and temporal deviations by appropriate assist functions can be considered separately. Following the above definitions, a spatial assist function applies changes only to lateral positions and forces, whereas a temporal assist function applies changes only to longitudinal positions and forces. In the following sections, both types of assist functions, spatial and temporal, are discussed.

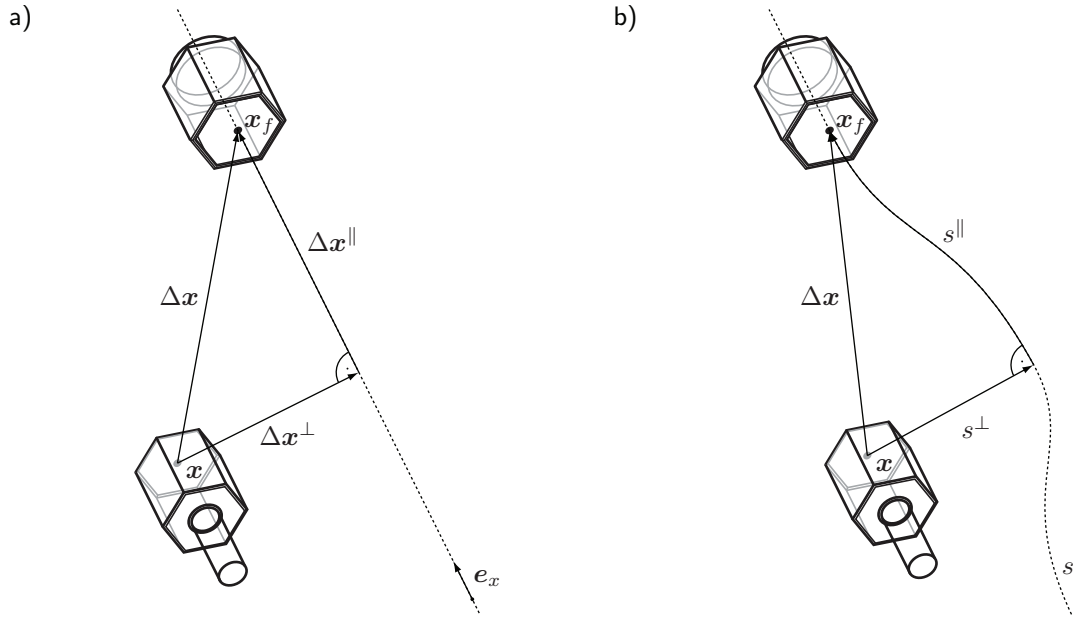

Figure 4.1: Decomposition of trajectory into lateral and longitudinal components. The vector from current point $\boldsymbol{x}$ to final point $\boldsymbol{x}_f$ is decomposed into a component which is tangential to the reference path and a component which is perpendicular to the reference path. In Fig. a), the reference path is a straight line passing through $\boldsymbol{x}_f$ with direction $\boldsymbol{e}_x$. The longitudinal component $\Delta\boldsymbol{x}^{\parallel}$ is parallel to $\boldsymbol{e}_x$, the lateral component $\Delta\boldsymbol{x}^{\perp}$ is perpendicular to $\boldsymbol{e}_x$. The case of a curved reference path s is illustrated in Fig. b). Again, the longitudinal component $s^{\parallel}$ runs along the reference path s, and the lateral component is perpendicular to s.

4.1 Spatial Assist Functions

When the operator leaves the reference path to the estimated target, the assist system can intervene in two ways to guide him back on this path. On the one hand, the strict coupling of the positions of operator and teleoperator can be relaxed such that the teleoperator follows the reference path even when the operator does not. On the other hand, the coupling of the forces of operator and teleoperator can be lifted such that the operator feels a force pushing him toward the reference path, although there are no forces applied to the teleoperator. Both methods can be combined into one system. Thus, it can be differentiated between *position based* assistance, *force based* assistance, and *hybrid* assistance, depending on the signals which are altered by the assist system.

4.1.1 Position-Based Assistance

A *position based assistance* system aims at correcting errors between actual and reference path by assigning a different desired position to the teleoperator than the observed position of the operator. There are many different ways of implementing these corrections, which determine the effects on performance and feeling of presence.

As the employed model of human point-to-point movements yields straight lines connecting starting point and endpoint, the path to the target can be simply decomposed into two vectors (cf. Fig. 4.1). The vector to the target point is defined as:

$$\Delta \boldsymbol{x} = \boldsymbol{x}_f - \boldsymbol{x} \tag{4.1}$$

The longitudinal component $\Delta \boldsymbol{x}^{\parallel}$ and the lateral component $\Delta \boldsymbol{x}^{\perp}$ are defined such that

$$\Delta \boldsymbol{x}^{\parallel} = \boldsymbol{e}_x^T \Delta \boldsymbol{x}\, \boldsymbol{e}_x, \quad \Delta \boldsymbol{x}^{\perp} = \Delta \boldsymbol{x}^{\parallel} - \Delta \boldsymbol{x} \tag{4.2}$$

When no assistance is applied, the positions of operator and teleoperator relative to the target are the same, i.e. $\Delta \boldsymbol{x}_O = \Delta \boldsymbol{x}_T$. In order to guarantee that the target point is reached, the lateral displacement of the teleoperator must be zero when the longitudinal component becomes zero:

$$\|\Delta \boldsymbol{x}_T^{\perp}\| = 0 \text{ if } \|\Delta \boldsymbol{x}_T^{\parallel}\| \leq 0 \tag{4.3}$$

A simple function, which ensures (4.3), maps a cylinder around the reference line onto a cone around the reference line, where the apex of the cone coincides with the target point:

$$\Delta \boldsymbol{x}_T^{\perp} = \frac{\|\Delta \boldsymbol{x}_O^{\parallel}\|}{x_0}\, \Delta \boldsymbol{x}_O^{\perp}. \tag{4.4}$$

The constant x_0 determines the longitudinal distance from the target, at which the cone and the cylinder intersect. For distances larger than x_0, the assistance is switched off. In any case, the longitudinal position is not subjected to the assist function, i.e. $\Delta \boldsymbol{x}_T^{\parallel} = \Delta \boldsymbol{x}_O^{\parallel}$.

The cone mapping ensures that the target point is reached for arbitrarily large deviations. Thus, it significantly improves the task performance. However, a pilot study showed that this position mapping is perceived as disturbing, and thus it compromises the intuitiveness of the assistance.

An alternative mapping, which was shown to be less obtrusive, is based on the lateral velocity instead of the lateral position. The idea of the velocity mapping consists in decreasing lateral motions away from the reference path, while increasing lateral motions toward the reference path. There are a variety of possible implementations of this idea, which can be described by the general form:

$$\Delta \dot{\boldsymbol{x}}_T^{\perp} = f(\Delta \boldsymbol{x}_T, \Delta \dot{\boldsymbol{x}}_O^{\perp})\, \Delta \dot{\boldsymbol{x}}_O^{\perp}. \tag{4.5}$$

The mapping function f determines the scaling between the lateral velocity commanded by the operator $\Delta \dot{\boldsymbol{x}}_O^{\perp}$ and the lateral velocity implemented by the teleoperator $\Delta \dot{\boldsymbol{x}}_T^{\perp}$. The selected scaling depends on the relative position to the target and the direction of the lateral velocity. Again, the longitudinal velocity is not subjected to the assist function such that $\Delta \dot{\boldsymbol{x}}_T^{\parallel} = \Delta \dot{\boldsymbol{x}}_O^{\parallel}$.

Two different scaling methods can be distinguished according to their direction dependence:

- The *unidirectional scaling (US)* damps lateral motions directed away from the reference path, but does not scale motions in the opposite direction[1]:

$$f(\Delta \boldsymbol{x}_T, \Delta \dot{\boldsymbol{x}}_O^{\perp}) = \begin{cases} g(\Delta \boldsymbol{x}_T) & \text{if } \langle \Delta \boldsymbol{x}_T^{\perp}, \Delta \dot{\boldsymbol{x}}_O^{\perp} \rangle > 0 \\ 1 & \text{otherwise} \end{cases} \tag{4.6}$$

- The *bidirectional scaling (BS)* applies scaling in both directions, where motions away from the reference path are damped, and motions toward the reference path are amplified:

$$f(\Delta \boldsymbol{x}_T, \Delta \dot{\boldsymbol{x}}_O^{\perp}) = \begin{cases} g(\Delta \boldsymbol{x}_T) & \text{if } \langle \Delta \boldsymbol{x}_T^{\perp}, \Delta \dot{\boldsymbol{x}}_O^{\perp} \rangle > 0 \\ (g(\Delta \boldsymbol{x}_T))^{-1} & \text{otherwise} \end{cases} \tag{4.7}$$

The function g determines the dependence of the scaling on the distance to the target:

- The *constant scaling (CS)* is independent of the distance to the target point:

$$g(\Delta \boldsymbol{x}_T) = d, \tag{4.8}$$

  where d is a constant scaling factor $d < 1$.

- The *position dependent scaling (PS)* is designed such that the scaling increases with decreasing distance to the target point:

$$g(\Delta \boldsymbol{x}_T) = d \, \frac{\|\Delta \boldsymbol{x}_T^{\parallel}\|}{x_0}, \tag{4.9}$$

  where x_0 specifies the distance to the target at which the assistance sets in, and d is again a constant scaling factor.

According to the pilot study, the velocity mappings effectively reduce the lateral error at the target point. However, in contrast to the position mapping, the velocity mapping cannot guarantee that the lateral deviation becomes exactly zero. Therefore, an additional *Position Correction (PC)* is introduced, which deliberately sets the lateral position error $\Delta \boldsymbol{x}_T^{\perp}$ to zero, just before the target is reached.

In Fig. 4.2, the resulting trajectories for the above described position and velocity scalings are illustrated. It shows the lateral deviation from the straight reference path as a function of the longitudinal position on the reference path. For this figure, the effects of the different assist functions are calculated based on a pre-recorded input trajectory. Thus, reactions of the human operator on the applied changes are not taken into account.

By definition, any position based assist function leads to deviations between operator and teleoperator positions $\Delta \boldsymbol{x}_T$ and $\Delta \boldsymbol{x}_O$. As long as these deviations do not exceed a certain limit and do not change abruptly, they are not perceived by the operator. The velocity mapping, however, can lead to unbounded differences, which is another drawback compared to the position mapping. Therefore, an appropriate velocity mapping, which reduces the position deviation, must be applied, in phases where no assistance is needed.

[1] For clarity of the notation, the scalar product between two vectors is denoted by the angle brackets $\langle \cdot, \cdot \rangle$

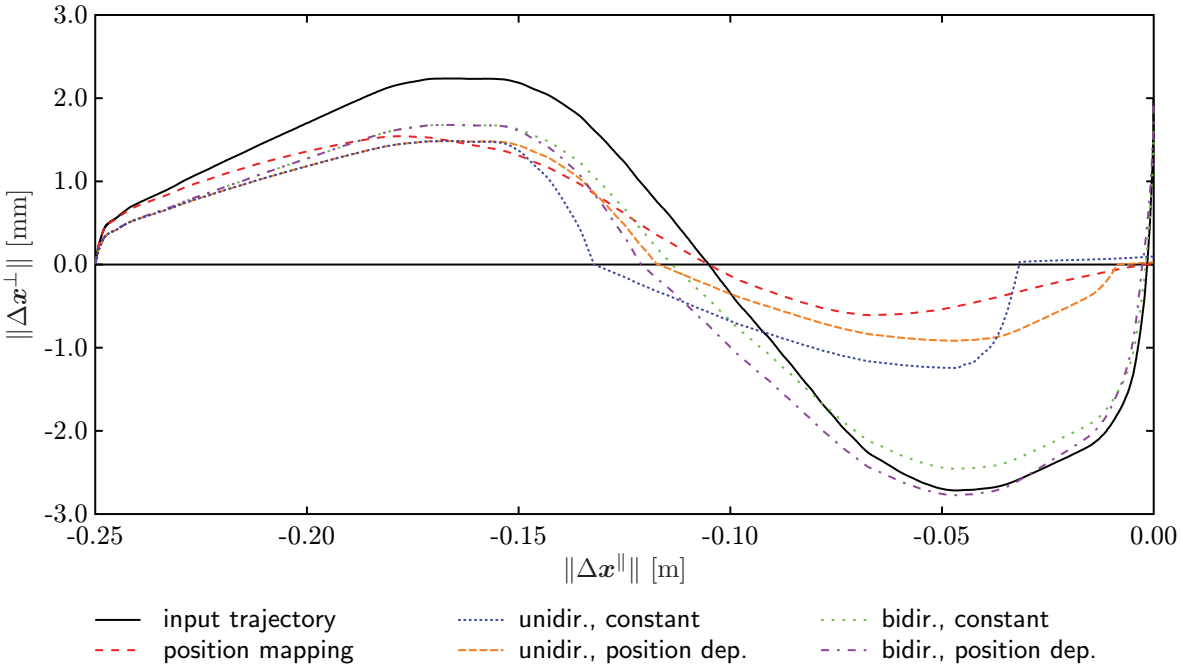

Figure 4.2: Resulting trajectories for different position based assist functions. The lateral deviation of the input signal $\|\Delta \boldsymbol{x}_O^{\perp}\|$ and the corrected signals $\|\Delta \boldsymbol{x}_T^{\perp}\|$ are plotted as functions of the distance to the target $\|\Delta \boldsymbol{x}^{\parallel}\|$. The approach movement starts at $x_0 = 0.25$ m from the target. The effects are calculated in an open-loop fashion, i.e. reactions of the operator on the corrections are not taken into account.

4.1.2 Force-Based Assistance

The *force based assistance* functions constitute the complementary approach to position based assist functions. They generate forces on operator side which are designed to guide the operator to the reference path. The position of operator and teleoperator, by contrast, are kept synchronized. Again, there is a variety of methods to calculate appropriate forces.

The guidance forces are added to the forces reflected from the teleoperator. As the approach phase usually takes place in free space, the measured forces on teleoperator side originate only from the inertia and gravity of the end-effector. The concept, however, is also applicable to scenarios, where damping or stiffness components are present.

$$\boldsymbol{F}_O = \boldsymbol{F}_T + \Delta \boldsymbol{F} \tag{4.10}$$

The guidance force $\Delta \boldsymbol{F}$ depends on the relative position to the target $\Delta \boldsymbol{x} = \Delta \boldsymbol{x}_O = \Delta \boldsymbol{x}_T$ and is always pointing perpendicularly to the reference path, i.e. it is aligned with $\Delta \boldsymbol{x}^{\perp}$:

$$\Delta \boldsymbol{F} = -f(\Delta \boldsymbol{x}, \Delta \dot{\boldsymbol{x}}) \frac{\Delta \boldsymbol{x}^{\perp}}{\|\Delta \boldsymbol{x}^{\perp}\|} \tag{4.11}$$

In contrast to the position based assist functions, here, the function f does not describe a scaling, but it has the dimension of a force.

Analogously to the position based assistance, there are different modes w.r.t. the lateral velocity $\Delta\dot{\boldsymbol{x}}^{\perp}$ of the operator as expressed by different functions f:

- In the *unidirectional force (UF)* mode, forces are only applied, when the operator is moving away from the reference path:

$$f(\Delta\boldsymbol{x}, \Delta\dot{\boldsymbol{x}}) = \begin{cases} g(\Delta\boldsymbol{x}) & \text{if } \langle \Delta\boldsymbol{x}^{\perp}, \Delta\dot{\boldsymbol{x}}^{\perp} \rangle > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.12}$$

- In the *bidirectional force (BF)* mode, forces toward the reference path are always applied, regardless of whether the operator converges to or diverges from the reference path:

$$f(\Delta\boldsymbol{x}, \Delta\dot{\boldsymbol{x}}) = g(\Delta\boldsymbol{x}). \tag{4.13}$$

Three different functions g are implemented, which describe the magnitude of the force as a function of the lateral and longitudinal distance as well as the duration of the error.

- In the simplest case, a *constant force (CF)* F_c is applied in direction toward the reference path, when the orthogonal distance $||\boldsymbol{x}_{\perp}||$ exceeds a threshold r_c:

$$g(\Delta\boldsymbol{x}) = \begin{cases} F_c & \text{if } \|\Delta\boldsymbol{x}^{\perp}\| > r_c \\ 0 & \text{otherwise} \end{cases} \tag{4.14}$$

- In a more sophisticated case, a *position dependent force (PF)* is calculated, which increases as the lateral deviation $\Delta\boldsymbol{x}^{\perp}$ increases or the longitudinal distance $\Delta\boldsymbol{x}^{\parallel}$ to the target decreases:

$$g(\boldsymbol{x}) = \begin{cases} F_c \frac{\|\Delta\boldsymbol{x}^{\perp}\|}{r_c} \left(\frac{x_0 - \|\Delta\boldsymbol{x}^{\parallel}\|}{x_0}\right) & \text{if } \|\Delta\boldsymbol{x}^{\perp}\| > r_c \\ 0 & \text{otherwise} \end{cases}, \tag{4.15}$$

  where x_0 denotes the maximum distance to the target at which the assist function is active.

- In an alternative mode, a *time dependent force (TF)* is applied toward the reference path, which increases linearly with time, when the operator position is outside a cylinder of radius r_c, and is reset to zero, when he reenters this cylinder.

A comparison of the guidance forces generated by the different force based assistance modes is given in Fig. 4.3. The lateral position deviation, which is an input to all presented force-based assist functions, is plotted as a function of the longitudinal distance to the target. The guidance forces generated by the five different force generation schemes are also plotted as functions of the distance to the target. The radius r_c around the reference path, below which no forces are generated, is shaded in gray.

4.2 Temporal Assist Functions

While the previously described spatial assist functions are intended to eliminate lateral position deviations, the *temporal assist functions* are intended to correct deviations from

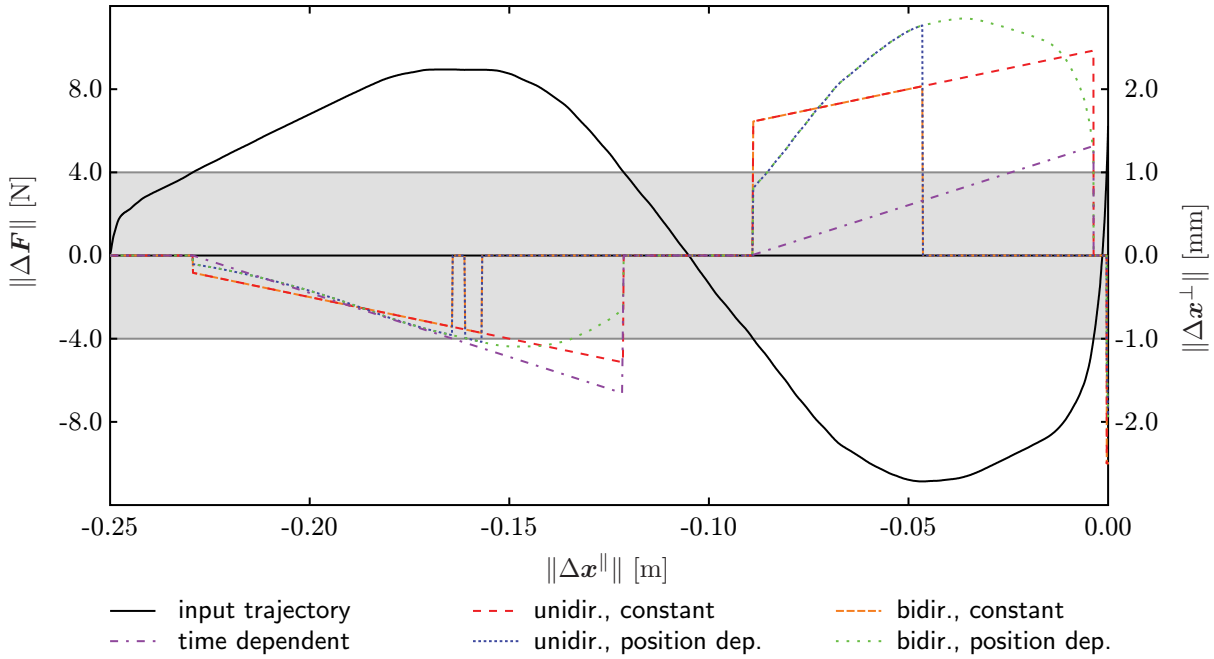

Figure 4.3: Guidance forces for different force based assist functions. The lateral position deviation $\|\Delta \boldsymbol{x}^{\perp}\|$ as well as the generated guidance forces $\|\Delta \boldsymbol{F}\|$ are plotted as functions of the longitudinal distance to the target $\|\Delta \boldsymbol{x}^{\parallel}\|$. The shaded area indicates the cylinder of radius r_c inside of which the guidance forces are deactivated.

the temporal course of the motion. The former ensure that the target point is hit, the latter ensure that it is hit at the correct time and velocity. Again, two types can be distinguished depending on the altered signal, namely, a position-based assistance and a force-based assistance.

4.2.1 Position-Based Assistance

In a practical case of inserting a screwdriver into a screwhead, a perfect alignment of the screwdriver and screwhead cannot be guaranteed. Therefore, the tool will finally slide into the screwhead. In order to avoid excessive forces, when the screwdriver touches the screwhead, the velocity should be small. This is particularly important in a telepresent execution, because the impact forces are usually considerably higher than in direct execution. An additional problem in telepresence is that the operator may overestimate the distance to the target due to impaired vision and approach the target with even higher velocity than normal.

The transition from freespace to stable contact with a stiff object is a challenging task for robots in general and for telepresence systems with time delay in particular. The comparatively high inertia and high structural stiffness of the teleoperator lead to large interaction force peaks with the environment even at low impact velocities. These forces can damage the teleoperator or the environment. Depending on the magnitude of the force peaks, it becomes hard to establish contact at all, because the operator bounces back from

the object in response to the unexpected forces. Unwanted oscillations of the operator and the teleoperator can be the consequence. The design of the telepresence system and the local controller of the teleoperator can reduce these effects, but these measures are limited by physics.

In order to eliminate the excessive interaction forces, the proposed assist function aims at minimizing the kinetic energy and the momentum of the slave at the moment of impact t_f, which implies minimizing the impact velocity $\dot{x}_f$. In order to reduce the velocity at the moment of impact, the trajectory must be reshaped in the temporal domain. To this end, the estimated time of hitting the target $\hat{t}_f$ is used to calculate a reference trajectory which hits the target at the same time but with zero velocity. By smoothly fading from the human commanded trajectory to this reference trajectory, the assistance is expected to not be noticed by the operator.

The human-commanded trajectory $\boldsymbol{x}_O(t)$ and the reference trajectory $\boldsymbol{x}_r(t)$ are merged to provide a smooth transition between both. The result is the trajectory $\boldsymbol{x}_T(t)$, which is actually implemented by the teleoperator:

$$\Delta\boldsymbol{x}_T^{\parallel}(t) = (1-\alpha)\,\Delta\boldsymbol{x}_O^{\parallel}(t) + \alpha\,\Delta\boldsymbol{x}_r^{\parallel}(t). \tag{4.16}$$

The sliding factor $\alpha \in [0;1]$ determines the degree of assistance, which is chosen to depend on the distance to the stiff object and, in an indirect way, on the estimated duration:

$$\alpha = \begin{cases} 0 & \text{if} \quad \|\Delta\boldsymbol{x}_r^{\parallel}(t)\| > x_0 \\ \frac{x_0 - \|\Delta\boldsymbol{x}_r^{\parallel}(t))\|}{x_0} & \text{if} \quad x_0 \geq \|\Delta\boldsymbol{x}_r^{\parallel}(t)\| > 0 \\ 1 & \text{if} \quad 0 \geq \|\Delta\boldsymbol{x}_r^{\parallel}(t)\| \end{cases}, \tag{4.17}$$

where x_0 is the maximum distance between tool and object at which the assist function is activated.

In Fig. 4.4, the operation of the assist function is illustrated. As indicated by the input trajectory $\boldsymbol{x}_O(t)$, the operator overestimates the distance to the target and, thus, touches the target with non-zero velocity. The recalculated reference trajectory $\boldsymbol{x}_r(t)$ touches the target at the same time, but the velocity at this point is zero. Additionally, the merged trajectory $\boldsymbol{x}_T(t)$ is shown in the figure, which smoothly blends from the input to the reference trajectory.

The proposed assist function can serve a secondary purpose. In the presence of a known time delay in the communication channel, the reference trajectory to the target can be used to extrapolate the motion of the teleoperator. Thereby, operator and teleoperator reach the target at the same time, and the time delay in one direction is cancelled out:

$$\boldsymbol{x}_T(t_f) = \boldsymbol{x}_O(t_f) = \boldsymbol{x}_f \tag{4.18}$$

4.2.2 Force-Based Assistance

The previously described assist functions aim at increasing the level of task performance by applying corrections to velocity and force signals. An additional demand to assist systems

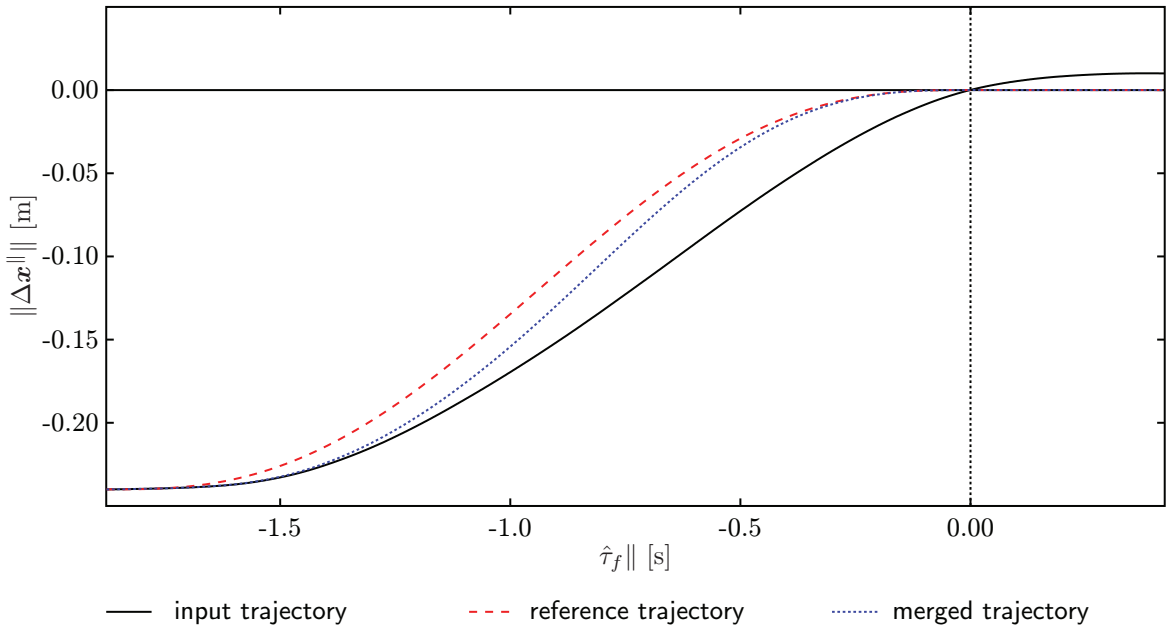

Figure 4.4: Operation of the temporal position based assistance. The input, the reference, and the merged trajectory are plotted as functions of time. The input trajectory $\boldsymbol{x}_O$ is given by the operator. The reference trajectory $\boldsymbol{x}_r$ is calculated by intention estimation and trajectory generator. Finally, the merged trajectory $\boldsymbol{x}_T$ is produced by blending between input and reference trajectory.

consists in increasing the feeling of presence. Thus, the force-based assistance presented in this section is designed to make the force feedback from the remote environment more consistent. The application of force assist functions is only reasonable in contact situations, because the feedback forces are zero in free space. In the exemplary screwdriving operation, the force assist function is effective only after contact between screwdriver and screwhead has been established.

A realistic force feedback is typically described in terms of transparency. According to Lawrence [59], transparency is defined as the ratio of the impedance displayed to the operator Z_t to the environment impedance Z_e:

$$\Gamma = \frac{Z_t}{Z_e}. \tag{4.19}$$

Perfect transparency, i.e. $\Gamma = 1$, implies a direct coupling between the operator and the remote environment. However, technical limitations, such as time delay in the communication channel or sensor deficiencies, degrade the maximum achievable degree of transparency. The idea of force assist functions is to circumvent some of these restrictions by rendering the remote object locally on operator side. Using an online updated model of the remote object, local forces between master and virtual object are computed and replace the measured and delayed transmitted forces from the remote side. With this assistance, the control loop is closed locally during contact, relaxing the requirements on stability. In the enlarged stability region, parameter sets are included, by which an increased feeling of perceived realism is reasonable to expect.

Stiffness-Based Prediction (SBP)

In order to predict the mechanical response of the remote environment on a certain movement, the mechanical properties of the object which is in contact with the teleoperator must be estimated beforehand. For a convincing local rendering of the remote environment, the estimation must be accurate and converge very quickly to the correct values.

There is a large variety of mechanical models ranging from simple time-invariant linear models to complex time-varying non-linear models. Likewise, there is a large amount of different estimation techniques for determining the optimal parameters of these models. In general, the more complex models with a higher number of parameters need more extensive measurement data and longer estimation times. Many estimation techniques furthermore require a well-defined excitation pattern to provide a good generalizability. In a telepresence application, however, the movements which produce the excitation are commanded by the human operator. Thus, the estimation method must be able to cope with arbitrary excitation signals.

The above mentioned requirements favor a very simple model. In a very simplistic approximation, the mechanical behavior of the remote environment can be modeled by a linear spring. This model has only one parameter, which makes the estimation fast and robust. Despite its simplicity, the model is well-suited for the designated application. On the one hand, the velocity and the acceleration of the end-effector are small, when the contact is established, such that damping and inertia properties can be neglected. On the other hand, the adaptation of the model parameter is fast enough to accommodate non-linearities of the compliance property. The mathematical description takes the following form:

$$F_T^{\parallel} = \begin{cases} -k\Delta x_T^{\parallel} & \text{if } \Delta x_T^{\parallel} < 0 \\ 0 & \text{otherwise} \end{cases}, \tag{4.20}$$

where $F_T^{\parallel}$ denotes the force response on teleoperator side, $\Delta x_T^{\parallel}$ denotes the position of the end-effector of the teleoperator, relative to the surface of the object, where $\Delta x_T^{\parallel}$ is negative, when the end-effector advances into the object. The model parameter k represents the mechanical stiffness of the spring. The model is implemented for scalar quantities only, because it is applied only in the longitudinal direction, which has only one degree of freedom.

As mentioned above, many parameter estimation methods for mechanical properties exist. In [105], an overview of estimation methods which are specifically designed for estimating environment parameters in telepresence is given. From the various approaches, the recursive least-squares method is selected because of its fast convergence rate.

Starting from an initial guess of the parameter $\hat{k}^{(i)}$, the estimate is recursively updated according to the following rule:

$$\hat{k}^{(i+1)} = \hat{k}^{(i)} + \left(\frac{\partial \hat{F}_T^{\parallel}}{\partial \hat{k}}\right)^{-1} (F_T^{\parallel} - \hat{F}_T^{\parallel(i)}) \tag{4.21}$$

$$= \hat{k}^{(i)} - \left(\frac{1}{\Delta x_T^{\parallel}}\right) (F_T^{\parallel} - \hat{F}_T^{\parallel(i)}), \tag{4.22}$$

where $F_T^{\parallel}$ denotes the measured interaction force, and $\hat{F}_T^{\parallel(i)}$ denotes its i-th estimate, which is calculated analogously to (4.20):

$$\hat{F}_T^{\parallel} = \begin{cases} -\hat{k}\Delta x_T^{\parallel} & \text{if } \Delta x_T^{\parallel} < 0 \\ 0 & \text{otherwise} \end{cases} . \tag{4.23}$$

The estimation process is started as soon as contact between the end-effector of the teleoperator and the remote environment is established, which can be detected by observing the measured interaction force $F_T^{\parallel}$. The initial value of the estimated parameter k should be as close as possible to the true value. If no prior knowledge about the stiffness of the remote environment exists, a conservative guess, i.e. a high value of k, should be selected to increase the safety of the system.

The estimated stiffness of the remote environment $\hat{k}$ is transmitted to the operator side, where the interaction force $F_O^{\parallel}$ is rendered in a similar way to (4.23):

$$F_O^{\parallel} = \begin{cases} -\hat{k}\Delta x_O^{\parallel} & \text{if } \Delta x_O^{\parallel} < 0 \\ 0 & \text{otherwise,} \end{cases} . \tag{4.24}$$

Obviously, the locally rendered stiffness will never exactly match the stiffness of the remote environment. However, psychophysical findings show that the discrimination threshold of the human is approx. 23 % for mechanical stiffness [49]. As long as the estimation error of $\hat{k}$ is below this *just noticeable difference (JND)*, there is no perceptual difference between direct and telepresent interaction with the environment.

Rate-Hardness-Based Prediction (RHBP)

By reproducing the estimated stiffness of the remote environment on operator side, the experience of contact with an object can be made more realistic. However, sudden changes from freespace motions to contact, i.e. the experience of impact, are not convincingly conveyed. On the one hand, the limited convergence rate of the stiffness estimation smoothes the sudden change in stiffness. On the other hand, more recent psychophysical experiments [60] suggest that humans perceive impacts rather by the rate-hardness than by the mechanical stiffness of the object.

Rate-hardness is defined as the ratio of initial force rate of change to initial penetration velocity:

$$h = \frac{\dot{F}_T^{\parallel}}{\dot{x}_T^{\parallel}} \tag{4.25}$$

The perceptual importance of rate-hardness compared to stiffness increases with increasing stiffness of the touched object. While the stiffness of soft objects can be judged by the ratio of force to penetration depth, stiff objects lead to very small penetration depths, which are hard to discriminate. Thus, higher frequency components of the contact force during impact become more relevant for stiff objects [73].

The concept of rate-hardness-based prediction is comparable to the stiffness-based prediction. The rate-hardness of the object in the remote environment is estimated upon impact, the estimated value is sent to the operator side, and there it is locally rendered by producing the appropriate force rate of change in response to the velocity. The force rate of change is estimated according to the definition

$$\dot{F}_T^{\parallel}(t) = \lim_{\delta \to 0} \frac{F_T^{\parallel}(t+\delta) - F_T^{\parallel}(t)}{\delta} \approx \frac{F_T^{\parallel}(t+aT) - F_T^{\parallel}(t)}{aT}, \tag{4.26}$$

where aT is an integer multiple of the sampling time T. As the rate-hardness changes very quickly, aT should be in the range of 1 ms. The exact value must balance between measurement speed and measurement noise.

The estimate of the rate-hardness $\hat{h}$ can thus be expressed by

$$\hat{h} = \frac{F_T^{\parallel}(t+aT) - F_T^{\parallel}(t)}{x_T^{\parallel}(t+aT) - x_T^{\parallel}(t)}. \tag{4.27}$$

Although this equation bears a close resemblance to the equations for stiffness estimation, the important difference lies in the short estimation duration aT, which emphasizes the higher frequency components.

The complementary part of the rate-hardness estimator is the rate-hardness renderer on operator side. Theoretically, the postulated relationship between force rate of change and initial penetration velocity (4.25) can be fulfilled by rendering a simple spring as described by (4.20) with $k = h$. However, the maximum displayable stiffness of a haptic display is limited by stability considerations, because a spring does not dissipate energy. An alternative way of creating the desired behavior uses a damping term to create the abrupt force changes, which dissipates energy and is thus stable [60].

The respective mathematical model is given by:

$$F_O^{\parallel}(t) = \begin{cases} b\dot{x}_O^{\parallel}(t) + k(x_O^{\parallel}(t) - x_f^{\parallel}) & \text{if } x_O^{\parallel} > x_f^{\parallel} \\ 0 & \text{otherwise.} \end{cases}. \tag{4.28}$$

This model leads to an infinitely high force rate of change, because the force is discontinuous at the transition from freespace to contact. However, a practical implementation, which exhibits a natural low-pass characteristic, has a limited rate of change. Assuming α^{-1} to be the time constant of the first-order low-pass filter, the filtered response of (4.28) becomes:

$$\dot{F}_O^{\parallel}(t) = \begin{cases} -\alpha F_O^{\parallel}(t) + \alpha(b\dot{x}_O^{\parallel}(t) + k(x_O^{\parallel}(t) - x_f^{\parallel})) & \text{if } x_O^{\parallel} > x_f^{\parallel} \\ 0 & \text{otherwise.} \end{cases}. \tag{4.29}$$

As the penetration depth is zero at the moment of impact, the modified equation yields an initial force rate of change of

$$\dot{F}_O^{\parallel}(t_f) = \alpha b \dot{x}_O^{\parallel}(t_f), \tag{4.30}$$

where t_f denotes the moment of impact. Thus, the rate-hardness is $h = \alpha b$ for the model in (4.29).

		domain of assistance	
		spatial	**temporal**
altered signal	**position**	$\Delta \boldsymbol{x}_T^{\perp}$ • correct lateral deviations • guarantee to hit the target • active in approach phase	$\Delta \boldsymbol{x}_T^{\parallel}$ • correct longitudinal deviations • ensure zero impact velocity • active in approach phase
	force	$\Delta \boldsymbol{F}_O^{\perp}$ • correct lateral deviations • active in approach phase	$F_O^{\parallel}$ • render contact forces locally • compensate communication delay • active in contact phase

Table 4.1: Matrix structure of assist functions. The table shows for each assistance mode, which signal component is altered by the assist system and which qualitative results can be achieved.

4.3 Combined Assist Functions

The assist functions, which are presented in the previous sections, can be categorized as spatial and temporal assist functions, or alternatively, as position-based and force-based functions. This categorization gives rise to a representation in a 2×2-matrix. From this structure, it is easy to derive that all four assistance types act on different signal components. Thus, they can be arbitrarily combined to benefit from their different purposes.

Table 4.1 illustrates for each of the four assistance types, which signals are used by the assist system to intervene and which goals are pursued by the intervention. Furthermore, not all assist functions can be employed in all phases of the screwdriving task. While the table states the design goals of the assist functions, it does not include any information about the fitness of the functions for these designed goals. Therefore, the performance of the different assist functions is evaluated in Chapter 5.

4.4 Discussion

The assist functions, presented in this chapter, constitute the core of the entire assist system. Based on the estimated intention of the human operator and the reference trajectory to implement this intention, they compute appropriate augmentation in order to increase task performance and feeling of presence.

The novelty of the presented assist functions consists in their dynamic adaptation to the intention of the operator. In contrast to commonly used virtual guides or forbidden regions, this preserves a high feeling of presence not affected by artifacts of the assist system. In the ideal case, the assist functions are not directly perceived by the operator as such, but only indirectly through increased ease of operation.

The presented assist functions target a single type of task with low complexity. The concept, however, can be extended to more complex tasks, as the understanding of human task planning and the autonomous capabilities of robots increase. The design process for new assist functions can be inspired by the process followed in Chapters 2 to 4: Firstly, find a model which describes the relevant human behavior in the targeted task. The model must be applicable to intention estimation as well as task planning. Secondly, build the scene recognition algorithms which can acquire the necessary data from the environment such that the intended task can be superiorly performed by the teleoperator. Thirdly, define a fusion strategy to merge the signals commanded by the operator with the signals planned by the teleoperator. The fusion strategy must apply the correction in a subtle way, lest the feeling of presence gets disturbed. Finally, a human-centered evaluation should be conducted to validate the designed assist function.

5 Experimental Evaluation

The quality and efficiency of typical control algorithms can be assessed by analytical computations or by simulations. When application dependent quality criteria are of interest, which involve the test under real conditions including noise etc., experimental data may be acquired to assess the performance. In any case, numerical performance indices can be computed from the simulated or recorded data. Although it can be hard to generalize the results in case of badly conditioned problems, the whole evaluation process is well founded on precise measurement data, which entails a high degree of repeatability.

As soon as humans are involved in the overall system, evaluation becomes much harder. The results typically depend heavily on the test persons, their current fitness, and their affinity to and prior experience with the matter of evaluation. This leads to bad generalizability, bad repeatability, and bad validity of the results. Psychophysicists have established methods to cure these problems which are inherent to all experiments, where humans are involved.

Despite all problems, studies with humans are the only way to check, whether the theoretically reasonable approach of the assist system is accepted by the human operator. Due to the bidirectional interaction of human and technical system, subtle changes can build up immense changes in the overall system behavior.

5.1 Psychological Foundations

Psychology is concerned with the scientific study of mental functions and human behavior. Inter alia, it aims at understanding the human in such a way that his behavior and reactions to certain conditions can be predicted. Psychology is a broad discipline covering such diverse subfields as clinical psychology, neuropsychology, media psychology, and organizational psychology. The more relevant subfields for the experimental evaluation of telepresence systems are *experimental psychology* and *psychophysics*.

Experimental psychology provides the methodology to experimentally explore human behavior and its underlying cognitive processes. More specifically, it allows testing certain hypotheses on human behavior. Experimental psychology is strongly governed by approaches from natural sciences.

Psychophysics is concerned with the analysis of human perception in response to physical stimuli. The main research questions consist in measuring and explaining detection and discrimination thresholds for different stimuli. Furthermore, the integration of different stimuli into a single percept is analyzed.

5.1.1 Design of Experiments

In the context of experimental evaluation, the prevailing purpose of experiments is the test of hypotheses. A hypothesis is formulated as alleged influence of one or more *independent variables* on one or more *dependent variables*. By verifying the hypothesis, a causal inference between the considered independent and dependent variables can be proven. Due to the complexity of mental functions and human behavior, the experimental design plays an important role, lest unwanted side effects dominate the results.

A major decision in the design of psychological experiments is the identification of independent and dependent variables. The independent variables are specifically manipulated during the experiments, and the effects on the dependent variables are observed. Using appropriate statistical models, a causal relationship between independent and dependent variables can be either verified or rejected. Analyses involving only one independent variable are called *unifactorial*, those involving more than one independent variable are called *multifactorial*. Analogously, analyses with a single dependent variable are called *univariate*, and those with multiple dependent variables are called *multivariate*. In practical experiments, it is impossible to completely isolate the effects of independent variables on dependent variables. The effects are superposed by the effects of uncontrolled, *extraneous variables*. Typical effects, which deteriorate the validity of an experiment are learning effects or different experience levels. This can be countered by sufficiently large test groups and by randomization of the experimental conditions.

Besides the selection of appropriate experimental variables, the experimental design determines the sample size, the grouping of participants, the combinations and order of test conditions, as well as the way of obtaining the dependent variables. The sample size, given by the number of participants of a study and the number of test conditions per participant, must be balanced between validity of the results and effort of the experiment.

The quality of an experiment is largely determined by the experimental design and is assessed by its *internal* and *external validity*. The internal validity describes the consistency of the obtained results. The external validity describes how well the results can be generalized. In general, internal and external validity are contradicting goals, because a high internal validity requires very narrow test conditions, which in turn deteriorate the external validity.

5.1.2 Statistical Methods

While the design of experiments must ensure that the relevant effects are contained in the recorded data, statistical methods are needed to give a meaningful explanation to the data. This explanation can be in the form of verifying or rejecting hypotheses. Alternatively, the statistical data processing can yield a rating of different test conditions. In any case, the statistical methods are employed to gain reliable results and an assessment of the level of confidence in these results.

From the vast number of statistical methods, only those which will play an important role in the following experimental evaluations are briefly described in the following.

Analysis of Variance (ANOVA)

The *analysis of variance (ANOVA)* is used to ascribe the variations of one dependent variable to the influence of one or more independent variables. More specifically, the core idea consists in partitioning the observed variance of the dependent variable into different explanatory components which are associated with the independent variables and an error component. The error component represents the portion of variance which cannot be explained by the independent variables and is attributed to model errors and the influence of extraneous variables.

The ANOVA is classified according to the number of independent variables and the number of levels in each of these variables. If only one independent variable is considered, the ANOVA is called unifactorial, otherwise multifactorial. An ANOVA with n independent variables, which can take m_i different conditions each, is referred to as $m_1 \times m_2 \times \ldots \times m_n$-ANOVA.

The *multivariate analysis of variance (MANOVA)* is an extension of the ANOVA for the case of multiple dependent variables, where the dependent variables are statistically correlated. It can also be used to detect interactions between independent variables and associations between dependent variables.

Scaling Techniques

In experimental evaluations, *scaling techniques* are employed to quantitatively judge the influence of certain independent variables on certain dependent variables. Although scaling cannot be equated with a metric measurement, it provides an ordinal scale or rank order scale. On this scale, the items are assessed regarding their relative strength. E.g. in a telepresence system, different control parameter settings can be scaled with respect to the feeling of presence they provide to the operator.

Scaling techniques can be classified into *non-comparative scales* and *comparative scales*. When non-comparative scales are applied, each item is judged individually by the participants. In contrast, when comparative scales are applied, participants must judge one item relative to one or more other items.

Non-comparative scales can be further subdivided into *continuous rating scales*, where participants can attribute a continuous grade to a tested item, and *itemized rating scales*, where participants must choose from a pre-defined set of grades. The most popular itemized scales are the *Likert scale*, the 7-point, and the 10-point scale.

There is also a variety of comparative scales. The most commonly used are *paired comparison*, *rank order* and *constant sum*. In a paired comparison experiment, participants are presented with two items and asked which item they value higher w.r.t. a certain property. For the above mentioned example, participants would be asked to judge which control parameter setting provides a better feeling of presence.

Data from comparative scales can be processed to produce a continuous rating scale. There is some evidence that scales from comparative studies are more reliable than those from non-comparative studies [18]. Before the scaling can be performed, the data must be

checked for *consistency* and *concordance*. The consistency describes the degree to which the individual judgments are consistent, i.e. non-contradictory. This is checked by testing the transitivity condition. For any three items S_1, S_2, and S_3, the relation $S_1 \prec S_2 \wedge S_2 \prec S_3 \Rightarrow S_1 \prec S_3$ must hold. The concordance describes the degree to which the judgments of different participants are consistent. The number of inconsistent judgments is compared to the total number of possible inconsistencies. If the ratio becomes too high, the data must be rejected.

A standard method for processing data from paired comparison studies is the *Bradley-Terry-Luce (BTL)* scaling [88]. The BTL scaling assigns scale values to each element from a set of n experimental conditions $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ such that the scale reflects the observed preference judgments. Typically, each of the n experimental conditions is compared to each other, leading to a total of $n(n-1)/2$ paired comparisons. From the preference judgments of all participants, the *dominance matrix* is deduced, which describes for each pair (S_a, S_b) how often item S_a is preferred over item S_b. The BTL model assumes that the probability $p(S_a, S_b)$ with which S_a is preferred over S_b can be solely described by the respective scale values $\phi(S_a)$ and $\phi(S_b)$. The scale values must be chosen in such a way that

$$p(S_a, S_b) = \frac{\phi(S_a)}{\phi(S_a) + \phi(S_b)} \tag{5.1}$$

hold for all $S_a, S_b \in \mathcal{S}$. Scale values are positive real number. Obviously, the scale function can be multiplied by any positive real number to yield an equivalent scale function.

5.2 Definitions

The definition of independent and dependent variables in an experiment requires a definition of the effects which are under investigation. As there is no unique terminology in the telepresence literature, the following definitions detail the meaning of some terms in the following description of the experiments.

- **Task Performance**: The task performance describes the efficiency of executing a task. The efficiency is measured by a comparison of the degree of fulfillment with the expended costs.

  The way in which the degree of fulfillment and the expended costs can be quantitatively measured is task dependent, and accordingly the task performance itself is a task dependent measure. In the context of telepresence, typical measures include position and force tracking accuracy as well as task execution time.

- **Feeling of Presence**: The feeling of presence denotes the degree of similarity in the experience of the real and of the mediated environment. The terms *feeling of presence*, *transparency*, *immersion*, and *perceived realism* are often used synonymously.

  While the similarity between real and mediated environment can be measured in terms of signals, it is unclear how deviations on the signal level translate to deviations in the perception. As the experience and the perception of the task execution in the real or in the mediated environment are not directly accessible, it is not possible to

physically measure the feeling of presence. Therefore, questionnaires are frequently used to assess the feeling of presence as subjectively perceived by the test persons. Alternatively, the feeling of presence can be ranked by pairwise comparing different conditions.

5.3 Spatial Assist Functions

In this section, the evaluation of spatial assist functions is described. As detailed in Chapter 4, the spatial assist functions only change the spatial but not the temporal course of the movement. They aim at eliminating lateral position errors during the approach phase to the target. Consequently, the feed motion is purely controlled by the human operator, and the total duration of the movement as well as the final velocity at the target are not affected by the assist function.

Recalling from Chapter 4, there are two major types of spatial assist functions, namely, the position-based and the force-based assist functions. Both methods can be parameterized to modulate the dependence on the distance to the target and on the derivative of the lateral error. Some additional, method-specific settings can be applied to tune the behavior of the assist functions.

The goal of the evaluation consists in assessing and comparing the position-based and force-based spatial assist functions w.r.t. their influence on task performance and feeling of presence. For both methods, the evaluation shall yield the optimal parameter sets.

5.3.1 Method

Experimental Setup

Following the overall benchmark scenario of the presented thesis, the evaluation is conducted for a screwdriving task. The specific scenario is inspired by a telepresent screwdriving task presented in [77]. The task consists in inserting a hexagon screwdriver into the hexagon socket of a screw and turning it. For the purpose of this study, the experiment is simplified in order to make the psychophysical analysis more meaningful. Instead of using a physical teleoperator, a virtual environment is employed which ensures a better reproducibility and higher safety. Furthermore, the task is restricted to the three translational degrees of freedom. Only the approach phase is analyzed, i.e. no interaction forces occur.

The virtual reality is visually rendered using a head-mounted display (HMD) with SXGA resolution and a frame rate of 30 Hz. The virtual camera pose is determined by tracking the head pose of the test person. An exemplary view of the virtual scene is depicted in Fig. 5.1. Haptic interaction is achieved through the large scale haptic display ViSHaRD10 (also used in [77],[101]). Detailed information on the experimental hardware can be found in Appendix C.

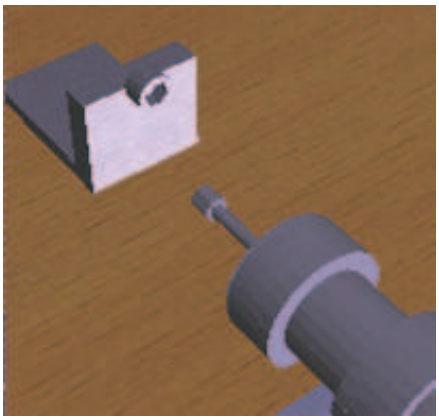

Figure 5.1: Virtual reality scene used as visualization tool in the experiment. The viewing area is adapted to the field of view of the HMD. The pose of the virtual camera in the virtual scene is always synchronized with the tracked head pose of the operator.

Experimental Design

In the experiment, the influence of position-based and force-based spatial assist methods on task performance and feeling of presence is evaluated. The task performance is derived from recorded data; the feeling of presence is subjectively evaluated on a rating scale.

More specifically, the experiment is designed to answer the following questions:

1. How do the parameters within the position-based assist function influence the task performance and the feeling of presence? The parameters are the strength of the assist function, its dependence on the direction, and its dependence on the distance to the target.
2. Does the additional *position correction (PC)* in the final phase of the approach movement influence the feeling of presence?
3. How do the parameters within the force-based assist function influence the task performance and the feeling of presence? The parameters are the strength of the assist function and its dependence on the direction, on the distance to the target, and on the duration of the error.
4. Is there a difference between force-based and position-based assist functions?

Participants

Experiments are conducted with 21 students from the Technische Universität München. 11 participants with an average age of 24 years (4 women, 7 men) are randomly assigned to the position-based assist functions (group A), while 10 participants with an average age of 23 years (5 women, 5 men) are randomly assigned to the force-based assist functions (group B). All test persons are right-handed and have normal or corrected to normal vision.

Independent Variables

1. *Position-based assist function: Influence of scaling factor, direction dependence, and position dependence.* Participants of group A are exposed to $3 \times 2 \times 2 = 12$ different parameterizations of the position-based assist function. Additionally, they have to fulfill the task without any assistance. The factors of interest are:
   - scaling factors varied on 3 levels: $d = 0.75$, $d = 0.50$, $d = 0.10$
   - direction dependence varied on 2 levels: *unidirectional scaling (US)* and *bidirectional scaling (BS)*
   - position dependence varied on 2 levels: *constant scaling (CS)* and *position dependent scaling (PS)*

   All 12 test conditions and the control condition are presented 5 times.
2. *Position-based assist function: Influence of position correction in final phase.* In order to answer the question regarding the influence of correction in the final approach phase, participants of group A also have to test $2 \times 2 = 4$ additional modes with position correction in the final phase:
   - direction dependence varied on 2 levels: *unidirectional scaling (US)* and *bidirectional scaling (BS)*
   - position dependence varied on 2 levels: *constant scaling (CS)* and *position dependent scaling (PS)*

   The scaling factor is kept constant at $d = 0.5$. Each of the 4 additional combinations is also repeated 5 times.
3. *Force-based assist function.* Participants of group B have to fulfill their task under $5 \times 3 = 15$ different parameterizations of the force-based assist function. They also experience the control condition with no assist function activated. The varied factors are:
   - force calculation mode on 5 levels: *time dependent force*, *constant unidirectional force*, *constant bidirectional force*, *position dependent unidirectional force*, *position dependent bidirectional force*
   - scaling factors varied on 3 levels: $d = 0.75$, $d = 0.50$, $d = 0.10$

   Each combination is again presented 5 times.
4. *Force vs. position-based assist function.* In order to compare both assist function types, some data from group A (combinations from 1.) and group B (combinations from 3. without the time dependent force method) entered the analysis.

Dependent Variables

In accordance with the overall evaluation objectives, the following two dependent variables are considered:

1. *Task Performance Measure.* The main objective of the analyzed task segment is to align the screwdriver correctly with the screwhead. Thus, the lateral deviation at the point of insertion is an appropriate measure for the task performance. The quantitative performance measure is defined as:

$$TP = \|\boldsymbol{x}_T^{\perp}\|^{-1}. \tag{5.2}$$

   When the position correction in the final phase is applied, the lateral deviation is systematically forced to be zero at the point of insertion such that the performance is optimal ($TP \to \infty$).

2. *Feeling of Presence Measure.* The feeling of presence is measured by two items of the presence questionnaire in [104] (translated to German by [87]): "How natural did your interactions with the environment seem?" and "How compelling was your sense of moving around inside the virtual environment?" Each question is rated on a 7-point rating scale. A sum score of presence rating FP is computed by summing both items.

Experimental Procedure

Participants are given time to become familiar with the test environment and the task. As soon as they can successfully complete the task in three consecutive trials with a maximum deviation in task execution time of one second, the preparation period is finished. Afterwards, they complete their trials depending on their group membership without being informed about the variation of the assistance modes. Each of the above described combinations is presented five times in fully randomized order to avoid learning effects. After each trial, the participant answers both questions regarding his subjective feeling of presence.

5.3.2 Result

Presence and performance measure of each combination are averaged across the 5 repetitions. The averaged baseline measure (no assist function activated) is subtracted from each experimental manipulation to derive information about improvement of applying the assist functions. This results in the relative measures ΔTP and ΔFP. The influence of the assistance mode characteristics on both dependent variables is first tested with multivariate analyses of variance ($MANOVA$) with the test statistics Pillai Spur. Significant effects (as well as question 2) are tested with univariate analyses of variance ($ANOVA$). Partial η^2 is chosen as effect size; if necessary, violations of assumed sphericity are corrected by the Greenhouse-Geisser correction. The significance level is set to 1%.

Position-Based Assistance:

Influence of scaling factor, direction dependence, and position dependence

The *MANOVA* reveals significant effects of damping ($F(2,9) = 35.6$, $p < 0.01$; $\eta^2 = 0.89$) and scaling ($F(4,40) = 5.2$, $p < 0.01$; $\eta^2 = 0.34$). Additionally, the interaction between both factors is statistically significant at the 5%-significance level ($F(4,40) = 3.1$, $p < 0.05$; $\eta^2 = 0.23$). No other effect is statistically significant and therefore not considered further.

1. *Task Performance Measure.* Variation in damping also affects position accuracy ($F(1,10) = 76.2$, $p < 0.01$, $\eta^2 = 0.88$): When inserting the screwdriver using the position-based damping scheme, performance is significantly better (cf. Fig. 5.2a). Scaling does not statistically influence accuracy on the 1%-significance level, but on the 5%-significance level ($F(2,20) = 15.3$, $p < 0.05$; $\eta^2 = 0.60$) indicating best performance with highest scaling; the Bonferroni test shows a significant improvement with the highest scaling compared to both other ones (again on the 5%-significance level). The interaction between damping and scaling is not statistically significant ($F(2,20) = 0.1$, $p < 0.01$).
2. *Feeling of Presence Measure.* Damping influences the presence rating ($F(1,10) = 13.1$, $p < 0.01$; $\eta^2 = 0.57$): Position-based damping is rated to result in a greater feeling of presence (cf. Fig. 5.2b). Additionally, the scaling factor affects the presence feeling indicating a greater increase of the rating with increasing scaling factor (corrected by Greenhouse Geisser: $F(1.2,12.5) = 10.3$, $p < 0.01$; $\eta^2 = 0.51$). However, as can be seen from Fig. 5.2b, this is only true with the constant scaling assistance scheme (significant interaction: $F(2,20) = 8.3$, $p < 0.01$; $\eta^2 = 0.45$).

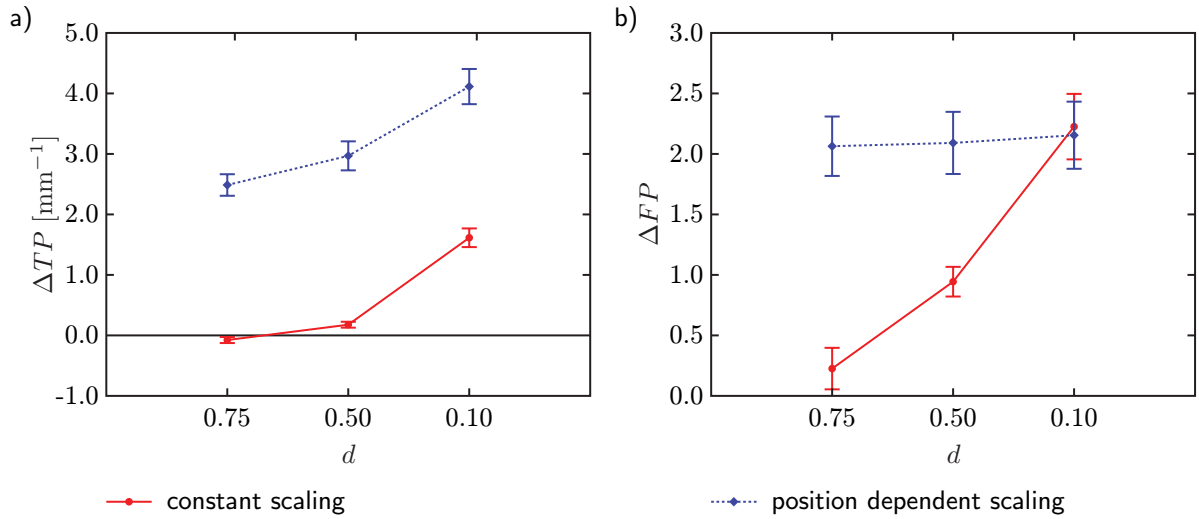

Figure 5.2: Effects of position-based assist functions compared to unassisted trials on a) task performance and b) feeling of presence

Position-Based Assistance:

Influence of Position Correction in Final Phase

Whether adding position correction in the final phase affects the feeling of presence is tested with a univariate *ANOVA* and reveals a significant influence ($F(1, 10) = 11.3$, $p < 0.01$; $\eta^2 = 0.53$): Position correction is rated to improve the feeling of presence. Additionally, an interaction between position correction and damping can be observed at the 5% significance level ($F(1, 10) = 5.7$, $p < 0.05$; $\eta^2 = 0.36$) indicating greater differences in presence ratings without position correction in the final phase just as described before (cf. Fig. 5.3a).

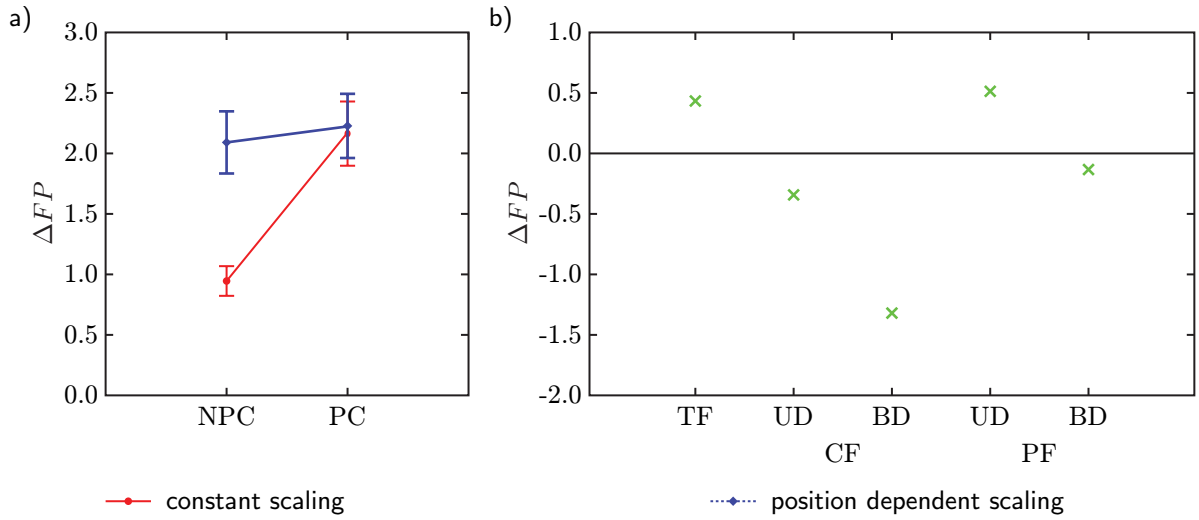

Figure 5.3: a) Influence of position correction in final phase and damping on presence rating (position-based assistance mode); b) Influence of different force-based assistance modes on presence ratings

Force-Based Assistance

The influence of the five investigated force-based assistance schemes (cf. Sec. 5.3.1) is tested with a *MANOVA* and affect the dependent variables ($F(8, 72) = 4.5$, $p < 0.01$; $\eta^2 = 0.33$). Additionally, the interaction between method and scaling factor reaches statistical significance ($F(16, 144) = 5.0$, $p < 0.01$; $\eta^2 = 0.36$). The scaling factor is significant at the 5%-significane level ($F(4, 36) = 3.3$, $p < 0.05$; $\eta^2 = 0.23$). Results of the separate analyses for each dependent variables are described below.

1. *Performance Measure.* Neither the type of force assist function ($F(4, 36) = 0.3$, $p = 0.89$) nor the scaling factor ($F(2, 18) = 0.62$, $p = 0.54$) nor the interaction between both (corrected by Greenhouse Geisser: $F(2.7, 24.5) = 2.9$, $p = 0.06$) influences the task performance in a statistically significant way.

2. *Feeling of Presence Measure.* The type of force assist function significantly affects the feeling of presence (corrected by Greenhouse Geisser: $F(1.9, 16.7) = 15.9$, $p < 0.01$; $\eta^2 = 0.64$). Bonferroni tests reveals a significant difference between the constant and the position dependent bidirectional force scheme ($p < 0.01$) indicating a decrease of the feeling of presence with the constant force mode. A marginally significant

difference between the time dependent force and the constant bidirectional force assist mode ($p = 0.01$) is due to a greater decrease in presence ratings with the constant bidirectional force assistance mode (cf. Fig. 5.3b).

The interaction between type of force assist function and scaling also reaches statistical significance ($F(8, 72) = 7.9$, $p < 0.01$; $\eta^2 = 0.47$): The greatest decrease in presence rating is observed with the constant bidirectional force assistance and the higher force scaling ($F(2, 18) = 7.0$, $p < 0.01$; $\eta^2 = 0.44$).

Position Based Assistance vs. Force Based Assistance

Lastly, position-based and force-based assist functions are compared according to their effect on task performance and feeling of presence. There is a statistically significant influence on performance ($F(1, 19) = 54.1$, $p < 0.01$; $\eta^2 = 0.74$) as well as on presence rating ($F(1, 19) = 11.4$, $p < 0.01$; $\eta^2 = 0.38$). Working with the position-based assist functions result not only in an improved feeling of presence compared to experiencing the force-based assist functions ($\overline{\Delta FP} = 1.6$, $\sigma = 1.6$ vs. $\overline{\Delta FP} = -0.2$, $\sigma = 1.1$) but also in an increased task performance, i.e. an increased positioning accuracy.

5.3.3 Discussion

In the presented experimental evaluation, two different spatial assist concepts, a position-based and a force-based approach, were assessed and compared. Both concepts are intended to reduce lateral position deviations in the approach phase of a peg-in-hole task. The feeling of presence should not be negatively affected by the assist functions.

In summary, the position-based assist functions are considerably superior to the force-based assist functions. The application of a position-based assist function can result in an increased task performance and an increase feeling of presence at the same time. As expected, the increase of task performance and feeling of presence is closely correlated to the scaling factor. In the experimental study, even the most extreme scaling factor did not produce any unnatural behavior, which would have decreased the feeling of presence. The task performance is superior when the position-dependent scaling is employed instead of a constant scaling.

The force-based assist functions do not provide a significant increase in task performance. The effects on feeling of presence are significant, however, none of the examined modes can increase the feeling of presence by the same amount as position-based assist functions can.

In general, increasing task performance and feeling of presence are seen as conflicting objectives: A high performance increase typically requires a strong intervention of the assist system which in turn badly affects the feeling of presence. In contrast, this experimental study shows that a human-adaptive assist system can increase performance and feeling of presence at the same time, because the system only suppresses unwanted errors such that the task is performed successfully and the expectations of the operator are better matched.

5.4 Temporal Assist Functions

In the previous section, evaluation results of the spatial assist functions were presented. In this section, a description of the complementary approach, the temporal assist functions, follows. The temporal assist functions leave the spatial course, i.e. the path of the approach movement unchanged, but apply corrections to the temporal course, i.e. the velocity profile. In this way, high impact velocities, which come along with high kinetic energy and high mechanical momentum, are avoided.

As discussed in Chapter 4, the compensation of time delay plays a major role in the design of the temporal assist functions. Therefore, the effects of this assist functions are tested in a condition with time delay and compared to the undelayed case.

In addition to the temporal reshaping of the approach trajectory, the evaluation investigates the effects of the force augmentation on operator side. This can be operated in two different modes, the *stiffness-based prediction (SBP)* and the *rate-hardness-based prediction (RHBP)* mode.

In this experimental study, the different temporal assist functions are rated w.r.t. their influence on the feeling of presence. An evaluation of task performance is not applicable to this assist function, because the related objective can be perfectly reached under mild assumptions.

5.4.1 Method

Experimental Setup

The scenario of the evaluation is again the screwdriving task [103]. The test persons are assigned to move the screwdriver forward and establish contact with the screwhead. In accordance with the purpose of the temporal assist functions, the transition from freespace to contact is in the focus of the evaluation. As only the longitudinal motion is of interest in this experiment, the task is restricted to one degree of freedom in direction of the approach motion.

The visual feedback is provided by the same virtual reality as used in Sec. 5.3 with the scene shown in Fig. 5.1. The advantage of a virtual reality lies in the possibility to conceal the current experimental condition from the test person. The haptic interaction is implemented by a 1 DOF telepresence system, which is shown in Fig. 5.4. The haptic interface and the teleoperator are both constructed as high-fidelity linear devices. The actuators are linear servo motors, *Thrusttube Modules 2504* by Copley Controls Corp. which are equipped with an optical position encoder (resolution 1 μm). A 1 DOF force sensor is employed to measure the interaction force with human operator and environment, respectively. Detailed information on the experimental hardware can be found in Appendix C.

The teleoperator is equipped with a cylindrical steel pin as end-effector, which is used to probe the remote environment. In order to vary the mechanical properties of the environ-

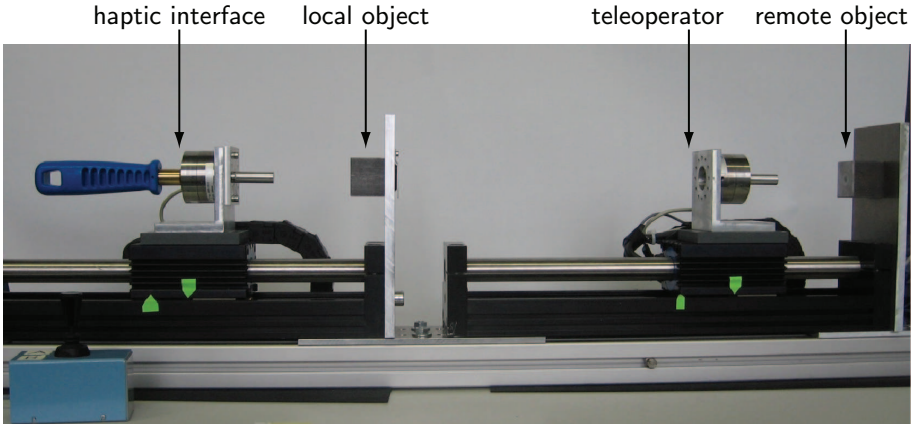

Figure 5.4: Haptic telepresence system with one degree of freedom. The system is composed of two identical linear actuators equipped with position and force sensors. On operator side and teleoperator side, cubes with different mechanical compliance can be mounted such that the operator can probe the environment either in direct or telepresent interaction.

ment, different cubes can be mounted on a plate in front of the teleoperator. The initial distance between end-effector and object is 8 cm.

On the linear device which acts as haptic interface, the handle of a screwdriver is mounted as end-effector. During the experiment, the test person grasps the handle and moves it away from his body toward the target object. In order to allow a comparison of direct and telepresent interaction with a mechanical object, the same steel pin as on the teleoperator is mounted on the backside of the haptic interface. There is also a mounting plate in front of the haptic interface such that the different cubes can be either presented on operator or teleoperator side.

As mentioned above, all external cues which allow the test person to differentiate between direct and telepresent manipulation are carefully hidden. This significantly increases the external validity of the experimental evaluation.

Experimental Design

The experiment is supposed to investigate the influence of the position- and force-based assist functions on the attribute "perceived realism". To this end, the method of paired comparison is employed in a 2×3 mixed-subjects experimental design, where the position-based assistance mode is varied on two levels (active, inactive) and the force-based assistance mode is varied on three levels (SBP, RHBP, none). A non-mediated ("real") condition, in which the teleoperator is omitted, and the physical contact is established directly on operator side, serves as reference and control condition. The systematic combination of all assist modes plus the real condition, yields $2 \times 3 + 1 = 7$ experimental stimuli, which are listed in Table 5.1. In each condition of the experiment, the test persons are presented the reference stimulus (RS) followed by two comparison stimuli arbitrarily selected from the seven possible settings (S1 ... S6, RS). After the presentation of each

		position-based assistance	
		active	**inactive**
force-based assistance	**rate-hardness-based prediction**	S1	S4
	stiffness-based prediction	S2	S5
	none	S3	S6
	real condition	RS	

Table 5.1: Experimental stimuli for the evaluation of temporal assist functions. There are $3 \times 2 = 6$ assistance conditions S1 . . . S6, which are compared to the real condition (RS), i.e. direction manipulation.

condition, the test persons are asked to decide which one of the two presented comparison stimuli feels most like the first presented reference stimulus.

Each stimulus is systematically contrasted with every other stimulus, thus leading to $(7 \times (7-1))/2 = 21$ pairs of stimuli for participants to judge. These stimulus pairs are presented under four different conditions: On the one hand, the stiffness of the remote environment is varied by using either a steel cube or a silicone cube. On the other hand, the time delay in the communication channel is varied between no time delay, i.e. $T_d = 0$ ms, and a time delay of $T_d = 10$ ms. Stiffness is treated as within-group variable, i.e. test persons experience both conditions. In contrast, time delay is treated as between-group variable, i.e. all participating test persons are randomly assigned to one of two groups, where persons of one group only experience conditions without time delay, and those of the other group experience only conditions with time delay.

In order to cancel out ordering effects, the sequence in which the stimulus pairs are presented is varied for each participant using the Latin square method. Furthermore, the order in which the two cubes of different stiffness are presented is also randomized. Although the visual appearance of the two cubes is different, test persons receive no prior knowledge about the current test condition, because the visual feedback generated by the HMD completely hides the experimental hardware.

Participants

The test persons of the experiment are an opportunity sample of 35 participants. The data of one test person is excluded from further analysis because of irregularities during the experiment. The remaining sample ($N = 34$) comprises 9 women and 25 men (mean age: 25 yrs., standard deviation: 4 yrs.). All participants are distributed randomly across the two test groups (see above). The groups are tested for differences in age or experience with 3D computer gaming, which turn out to be non-significant ($t(32) = 0.39$, $p > 0.05$; $t(17, 17) = 1.06$, $p > 0.05$, respectively).

Three participants state to be left-handed, all others state to be right-handed. Depending on the handedness, test persons conduct the experiment either with the right or the left hand, where the virtual reality is adapted accordingly.

Procedure

Before the actual experiment starts, the test persons are introduced to the experimental apparatus. They are given the opportunity to get acquainted with the device and the task. Each experimental condition consists of three trials, in which the reference stimulus and two different stimuli from Table 5.1 are presented. In each trial, the test person moves the virtual screwdriver forward and feeds it into the virtual screw socket, where the haptic feedback comes from the steel pin touching one of the test cubes. After having performed all three trials, the test person is asked to judge, which of the stimuli in the latter two trials feels most similar to the reference stimulus experienced in the first trial. Each test person is exposed to $2 \times 21 = 42$ test conditions such that both cubes are tested under all stimulus pairs. The total duration of the experiment is approx. one hour per test person.

5.4.2 Result

Manipulation check

Before further analysis, the acquired data is checked for validity. The consistency coefficients are calculated for all four experimental conditions, i.e. the two stiffness levels combined with the two time delay levels, and yield a high degree of consistency in judgment. The validity of the individual ratings is also supported by the fact that the reference stimulus is always rated higher than telepresent conditions. This confirms that the test persons judge according to the correct criterion. Furthermore, the concordance of the judgments is assessed for the four experimental conditions by calculating Kendall's concordance coefficient W, which demonstrates a high degree of consistency between test persons.

In Fig. 5.5, the preference judgments are illustrated. Each bar in the figure indicates how often the stimulus is preferred over any other stimulus. As each of the 17 test persons per group experienced 21 conditions per cube, there are a total of $17 \times 21 = 357$ preference judgments, which are distributed across the 7 stimuli.

Selection of BTL model

For the further analysis, the data from the paired comparison trials is translated to a continuous rating scale by using the Bradley-Terry-Luce (BTL) model, see Sec. 5.1 and [19]. As the experiment comprises four experimental conditions, the BTL model is extended to accommodate several preference matrices. A general, multinominal model with all 21 comparisons is compared to a restricted model with only 7 parameters w.r.t. the ability to explain the paired comparison data. In both cases, the model parameters are determined by using a maximum likelihood estimation method. As the general model does not explain the experimental data significantly better than the restricted model, the latter is used, because

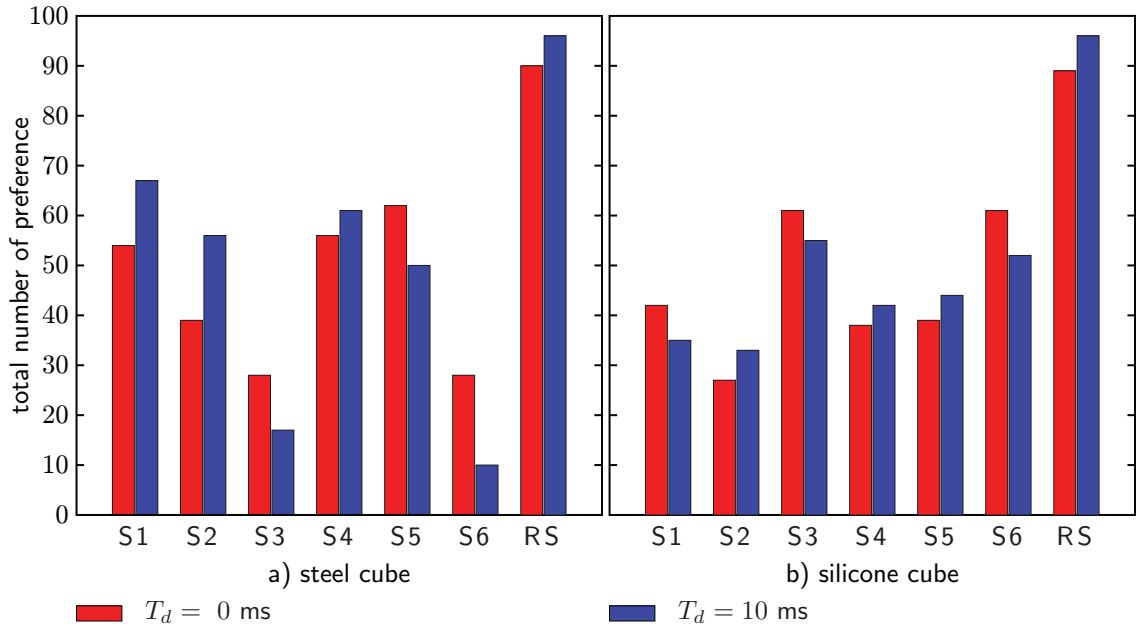

Figure 5.5: Preference judgments for all four experimental conditions. The bars express the number of test condition in which the stimulus is preferred over the comparison stimulus. In Fig. a), the results for the steel cube are presented for the delayed and non-delayed case. In Fig. b), the respective results for the silicone cube are shown.

it has fewer parameters. The numerical data of the comparison for all four experimental conditions is depicted in Table 5.2.

Differences in preference judgments for different experimental conditions

It is tested whether time delay in the communication channel has a significant effect on the preference ratings. To this end, a *conjoint approach* is employed, in which a general model with 12 free parameters (df=12) is compared to a restrictive model with only 6 free parameters (df=6). The comparisons are performed separately for the experimental conditions involving the steel cube and those involving the silicone cube. For the steel

hard contacts	$T_d = 0$ ms	$\chi^2(14) = 9.68, p = 0.79$
	$T_d = 10$ ms	$\chi^2(14) = 9.74, p = 0.78$
soft contacts	$T_d = 0$ ms	$\chi^2(14) = 12.87, p = 0.54$
	$T_d = 10$ ms	$\chi^2(14) = 10.09, p = 0.76$

Table 5.2: Comparison of general and restricted BTL model. The table list the values of the χ^2-distributions comparing both models. The differences are not significant for any experimental condition.

cube, the general model (df=12) explains significantly more variance than the restricted model (df=6). In contrast, for the soft cube, no significant difference between general and restricted model can be observed. Thus, time delay has a significant effect on the rating when stiff objects such as the steel cube are touched, whereas the effect vanishes when soft objects such as the silicone cube are touched.

The relative strengths of preferences for contact settings are displayed in Fig. 5.6. The most preferred settings are normalized to one. Note that preferences of soft contacts are combined into one preference scale because there are no significant differences between the two time delay conditions.

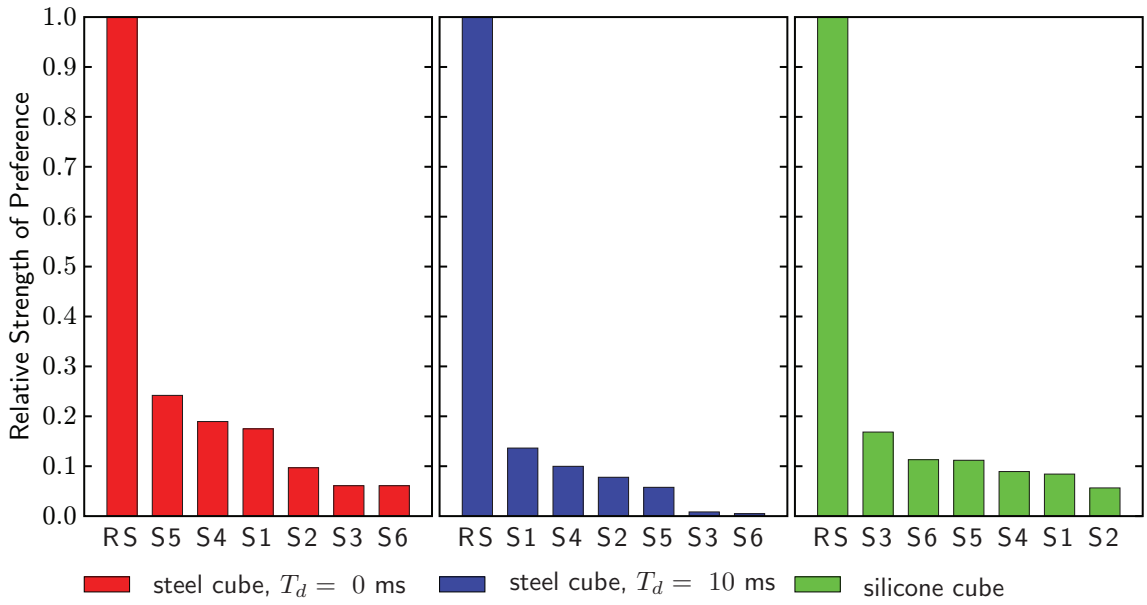

Figure 5.6: Relative strength of preference. The charts illustrate the rating scale obtained through the BTL model for different experimental conditions. In all conditions, the reference stimulus gets the highest ranking.

Perceived realism of contacts with a stiff environment

In order to assess the effect of the various assistance modes, the BTL-models are checked for significant differences between the respective preference ratings. Besides the influence of the individual modes, namely position-based assistance and the two force-based assistance modes (SBP and RHBP), the possible interactions between position-based assistance and force-based assistance are investigated. The analysis is performed separately for contacts with stiff environments as represented by the steel cube and contacts with soft environment as represented by the silicone cube.

In Table 5.3, the results of the relevant comparisons between different experimental conditions are summarized. The significance level for the following tests is adjusted with Bonferroni corrections and set to $p < 0.007$ in order to compensate for multiple comparisons. Because of the differences in the preference ratings between the non-delayed and the delayed condition, separate comparisons are performed ($N = 17$).

stimulus pair	$T_d = 0$ ms	$T_d = 10$ ms
S4, S6	$\chi^2(1) = 15.50,\ p < 0.007$	$\chi^2(1) = 65.97,\ p < 0.007$
S5, S6	$\chi^2(1) = 22.63,\ p < 0.007$	$\chi^2(1) = 43.64,\ p < 0.007$
S4, S5	$\chi^2(1) = 0.73,\ p = 0.39$	$\chi^2(1) = 3.00,\ p = 0.08$
S3, S6	$\chi^2(1) < 0.001,\ p > 0.99$	$\chi^2(1) = 2.18,\ p = 0.14$
S1, S6	$\chi^2(1) = 13.42,\ p < 0.007$	$\chi^2(1) = 79.91,\ p < 0.007$
S2, S6	$\chi^2(1) = 2.53,\ p = 0.12$	$\chi^2(1) = 65.71,\ p < 0.007$
S1, S2	$\chi^2(1) = 4.42,\ p = 0.04$	$\chi^2(1) = 3.04,\ p = 0.08$

Table 5.3: Preference conditions of contacts with a stiff environment. Relevant experimental conditions are checked for significant differences in their preference ratings. As the BTL models are significantly different for experiments with and without time delay, the analyses are performed separately.

In particular, the following qualitative results can be obtained from the comparisons:

- *Force-based assistance modes significantly improve the perceived realism compared to the unassisted condition.* This statement holds for the stiffness-based prediction (S5, S6) as well as the rate-hardness-based prediction (S4, S6) independently of the time delay in the communication channel. The statement is even still valid when position-based assistance is added, because in the combined assist functions, stiffness-based prediction (S2, S6) and rate-hardness-based prediction (S1, S6) are again preferred over the unassisted condition.
- *There is no significant difference between stiffness-based prediction and rate-hardness-based prediction.* This statement is independent of the time delay and holds for conditions without additional position-based assistance (S4, S5) as well as conditions with additional position-based assistance (S1, S2).
- *Position-based assistance does not significantly deteriorate the perceived realism.* The unassisted condition (S6) is not significantly preferred over the condition without position-based assistance alone (S3). Also in combination with force-based assist functions, the position-based assistance did not change the effects on perceived realism. However, in the non-delayed case, the preference of the stiffness-based prediction over the unassisted case drops below significance level when position-based assistance is activated (S2, S6).

Perceived realism of contacts with a soft environment

The same analyses regarding significant differences in the preference ratings are also performed for contacts with a soft environment. The respective results are summarized in Table 5.4. As there are no significant differences between the two time delay conditions,

stimulus pair	
S4, S6	$\chi^2(1) = 9.22,\ p < 0.007$
S5, S6	$\chi^2(1) = 8.64,\ p < 0.007$
S4, S5	$\chi^2(1) = 0.01,\ p = 0.92$
S3, S6	$\chi^2(1) = 0.01,\ p = 0.92$
S1, S6	$\chi^2(1) = 13.90, p < 0.007$
S2, S6	$\chi^2(1) = 28.51, p < 0.007$
S1, S2	$\chi^2(1) = 2.91,\ p = 0.08$

Table 5.4: Preference conditions of contacts with a soft environment. Relevant experimental conditions are checked for relevant differences in their preference ratings. Only one BTL model needs to be considered because there are no significant difference between experiments with and without time delay.

all ratings of contacts with soft environments are combined into one preference matrix ($N = 34$).

The specific results which can be extracted from Table 5.4 are:

- *Force-based assistance modes significantly decrease the perceived realism compared to the unassisted condition.* As opposed to the experiments with stiff environments, the stiffness-based prediction (S5, S6) as well as the rate-hardness-based prediction (S4, S6) significantly decrease the perceived realism. This also holds when the force-based assist functions are combined with position assistance (S2, S6) and (S1, S6).
- *There is no significant difference between stiffness-based prediction and rate-hardness-based prediction.* This observation is in line with the experiments with stiff environments. It is valid when no position-based assistance is added (S4, S5) and when position-based assistance is added (S1, S2).
- *Position-based assistance does not significantly deteriorate the perceived realism.* As indicated by the comparison (S3, S6), the condition without assistance is not significantly preferred over the condition with position assistance.

5.4.3 Discussion

In this section, an experimental evaluation of temporal assist functions was performed. In the evaluation, the influence of position-based and force-based assist function on the realistic portrayal of haptic contacts with surfaces of differing stiffness in a telepresence system was investigated.

Theoretically, the position-based assist functions are prone to reduce the perceived realism for two reasons: On the one hand, the trajectory of the teleoperator before the impact is deliberately altered by the assistance algorithm and is, therefore, not in full accordance with

the commanded trajectory. On the other hand, the change of the impact velocity, which is imposed by the position assistance scheme, is expected to affect the perception during the establishment of contact. The results, however, contradict these expectations and show that position-based assistance did not significantly affect the perceived realism, regardless of the stiffness properties of the environment. Apparently, the position deviations during the approach phase are too small and are built up too slowly to be perceived by the user [86]. Previous studies suggest that stiffness perception when mediated by tools relies primarily on kinesthetic rather than tactile information, e.g. [58], which could explain why the changed impact velocity is not noticed.

The force-based assist functions are designed and expected to increase the perceived realism when physical contact between teleoperator and environment is established. This expectation is plausible, because the environment is rendered locally, thereby omitting the deteriorating and softening effects of the telepresent control loop. In contrast, the force-based assist functions are expected to decrease the perceived realism in interactions with soft environments. This is attributed to the initialization of the stiffness estimation to a stiff environment, which needs some time to adapt to the correct value. The experimental evaluation confirms both hypotheses. The expectations are met for both force-based assistance schemes, the stiffness-based prediction as well as the rate-hardness-based prediction.

As the rate-hardness-based prediction is specifically designed for contacts with stiff environments, it is expected to perform superiorly to the stiffness-based prediction when contact with a stiff environment is actually established. However, this expectation cannot be confirmed by the experiment. Unlike other studies in the literature, see [60], no difference in the perceived realism produced by a stiffness-based and a rate-hardness-based prediction scheme can be observed, here. This disagreement may be attributed to the different characteristics of the employed hardware. In contrast to the hardware used in [60], the experimental device used in this study is able to output a high degree of stiffness. It is, therefore, possible that rate-hardness-based prediction is only superior to mechanical stiffness with devices that tend to render stiff surfaces softer.

For the combination of position-based and force-based assistance, a significantly positive effect is expected for contacts with stiff environments, because force-based assistance is thought to prevail possibly negative effects of position-based assistance. For contacts with soft environments both assist functions are assumed to significantly deteriorate the feeling of perceived realism, leading to an overall negative effect. Except for one comparison, these hypotheses are confirmed by the experiment. The combination of stiffness-based prediction and position assistance for stiff contacts without time delay fails significance when compared with the unassisted teleoperation mode.

The experiments were conducted for a telepresence system without time delay and for a system with time delay in the communication channel. It is expected that time delay does not change the effects of the assist functions qualitatively but quantitatively such that the effects are more prominent in trials with time delay than in trials without time delay. In general, this is confirmed by the experiment. The only exception can be found in conditions with a stiff environment when position and force assistance modes are combined. Apparently, the influence of this combination drops below significance level for the no time delay condition.

5.5 Discussion

As the human operator forms an integral part of a telepresence scenario and as the presented assist system claims to improve the performance and the experience of the human operator using the assisted telepresence system, the extent to which the claim is fulfilled can finally be judged by the human only. In order to get reliable results regarding the acceptance of the system by the user, psychologically founded experiments were conducted.

The evaluations in this chapter attest the successful application of the developed assist functions. More specifically, they show that the task performance of peg-in-hole tasks can be optimized without disturbing the operator. It is even shown that the feeling of presence also increases, when the task performance is increased.

The generalizing interpretation of the results suggests that the human operator is willing to accept corrections of command and feedback signals, when these corrections make the telepresent procedure more resembling to the direct execution of that procedure. In this way, the task performance approaches the level of direct manipulation, and the feeling of presence also increases because the expectations of the operator are better matched.

6 Conclusions and Future Work

The research reported in this thesis is focused on the development and evaluation of assist functions for haptic telepresence systems. The design objective for the assist functions consists in supporting the human operator without degrading the intuitiveness of the telepresence system. In this chapter, the results of the research and the specific contributions are summarized. Finally, future research directions are suggested.

6.1 Summary and Results

In order to properly embed the presented research into the known literature, a classification of existing systems according to the control paradigm and the assistance paradigm is made in Chapter 1. It is shown that there is a strong correlation between time delay, update rate, degree of autonomy, and human-to-robot ratio of common telepresence architectures. Furthermore, a distinction of assist systems into intramodal and crossmodal systems according to the supporting and the supported modality is developed.

The underlying idea of the thesis is to combine the particular strengths of human operator and robotic telemanipulator. While the human operator provides the creativity, decision making, and sense of responsibility, the telemanipulator contributes the endurance, working precision, and resistance to adverse environmental conditions. Consequentially, the assist system does not interfere with the operation planning of the human, but silently corrects inaccuracies in the operation execution. In summary, the application of the assist system makes it easier to fulfill a given task without being noticed as such by the human operator.

The components, which are required to implement this concept, are developed in this thesis, and the validity of the concept is shown in experiments. A screwdriving task serves as application example for both, the development and the evaluation.

As the telemanipulator is required to support the human operator in the currently executed task, it must first and foremost know which task he is aiming to accomplish. In order to not distract the human operator, his intention is silently inferred from his motions and actions. The general aspects of intention recognition are discussed in Chapter 2. For the specific application example, the screwdriving task, an intention estimator is developed. It is based on a model of human point-to-point movements, which constitute the differencing part of the screwdriving task. Fitting observed movement data to the model yields the target of the movement and, thus, the target of the screwdriving task. The fitting process must be performed in real-time so that its results can be used before the task is finished. In Chapter 2, different real-time capable intention estimation algorithms are developed and compared w.r.t. their performance, complexity, and computational effort. As a main result, it is stated that model fitting alone does not provide sufficiently accurate intention

estimates, but that hypotheses on possible target points must be provided. Further, it is shown that a Gauss-Newton approximation working on the full trajectory data provides the best results at the cost of requiring high computational power.

The above postulated need of hypotheses for the intention estimation process can be fulfilled by a proper scene recognition system, which extracts possible targets from visual data of the remote scene. Additionally, this data is required for planning a reference path to the target, once it is selected as the currently intended target. In Chapter 3, a broad overview of literature in the field of image processing and scene recognition is given. As this is a vast research domain by itself, but constitutes only a supporting technology within the scope of this thesis, a simple, marker-based feature extraction algorithm is selected and tailored to the application example. As the main contribution, the orthogonal iteration algorithm, which estimates the relative pose between camera and marker, is extended to a multi-camera setup. It is shown that the developed pose estimator with multiple cameras far outperforms the single camera solution in terms of accuracy and robustness.

The centerpiece of the assist system are the actual assist functions, which alter the interchanged position and force signals in such a way that the task performance and the feeling of presence are improved. The assist functions depend on the knowledge of the intended target and a reference path to implement this target. Based on this knowledge, they provide assistance, which is effective but unobtrusive. Different functions are proposed in Chapter 4. They are grouped into temporal and spatial assist functions depending on whether they alter the temporal or the spatial course of the trajectory. Another distinction is made into position-based and force-based assist functions, where the former alter the position signals and the latter alter the force signals. Combined assist functions are proposed which change both signals in the spatial as well as in the temporal domain. It is argued that smooth transitions between commanded trajectory and reference trajectory are favorable and practically realizable.

An experimental evaluation complements the development of the assist functions. In the respective Chapter 5, the hypotheses, which have governed the design of the assist functions, are tested. In separate sections, the spatial and the temporal functions are evaluated. It is shown that in all cases, position-based assist functions show excellent results. As a main finding it is discovered that assist functions can increase task performance and feeling of presence at the same time. This is attributed to the fact that a telepresent task execution with less errors resembles more the direct task execution, which in turn leads to a higher feeling of presence. Furthermore, the temporal force assist function which is based on estimating mechanical properties of the remote environment can increase the feeling of presence when the initial guess of the mechanical properties is sufficiently close to the true values. Altogether, the experimental evaluation proves the benefits of the proposed assist concept to the effect that the performance of the telepresence system is increased, although the human operator does not explicitly notice the assist system.

6.2 Review of Contributions

In this section, the major research contributions of this thesis are revisited. These contributions are seen as the generalizable outcomes of the presented research.

- The assist concept itself, which consists in estimating the operator intention, creating a corresponding reference plan, and finally integrating commanded actions and reference plan into a superior action sequence, is the most significant contribution. While this is applied to a simple example task within this thesis, it is applicable to a broad range of tasks. Fundamentally, the concept presents an attractive way of combining human and machine intelligence.

  Although many assist systems for haptic telepresence systems have been proposed before, this is the first concept which does not depend on prior planning or explicit instructions to the assist system. Therefore, it is the only assist system which fully maintains the feeling of presence in the remote environment

- The intention estimation algorithm for human point-to-point movements extends the current state of the art. Most existing systems rely on trained classifier systems, where the most prominent ones are Hidden Markov Models. In contrast, the presented algorithm is based on a dynamic model of the considered movements. A system, which can estimate trajectories based on a model in real-time and in 3D, has not been presented before.

- The pose estimation algorithm with multiple cameras combines the global convergence and computational efficiency of the classic orthogonal iteration algorithm with the increased accuracy and robustness which can be achieved by combining multiple sources of information. Thus, the presented visual pose estimation algorithm reaches a higher performance level than previous algorithms. The application range far exceeds the context of assisted telepresence.

6.3 Future Research Directions

From the presented research, several future directions emerge. On the one hand, direct extensions of the presented example can be considered, on the other hand, fundamentally new applications of the presented concept are thinkable.

Within the presented application example, two lines of further developments can be followed. On the one hand, the intention estimation process can be embedded into a larger intention recognition process, which first determines the type of action, the human operator is aiming to perform. On the other hand, the scene recognition system can be extended from the current marker-based system to a feature-based system, which is able to detect possible targets from camera images of the scene without markers.

As many efforts are undertaken to provide robots with more autonomy and a higher degree of intelligence, the capabilities of robots will gradually increase in the time to come. However, in the foreseeable future, robots will not replace humans in many areas, but

instead new forms of human-robot collaboration will emerge. Thereby, the application range of the presented assist concept will further increase. A higher degree of intelligence will enable the robots to detect the intention behind more complex action sequences of the human operator. At the same time, the higher degree of intelligence will allow creating appropriate reference plans for these more complex action sequences. Thus, unobtrusive assistance will be implementable at a higher level. Besides from being very effective, this form of human-robot collaboration has the advantage that the human always takes the superior role such that the method earns a high level of human acceptance.

Finally, the idea of dynamic assist functions can be applied to telepresence systems with multiple human operators. Such multi-user systems are a prevailing direction in telepresence research in order to bring the benefits of telepresence to scenarios, in which the support of more than one operator is needed to perform a given task. In these cases, the assist functions can help the operators to coordinate their actions, e.g. when aligning objects. For this purpose, the intentions of all operators must be estimated and integrated into a common goal, a reference plan for this common goal must be developed, and finally this plan must be fused with the commands of the operators to give the actual commands to the teleoperators. Although this setting is far more complex than the single-user case, the basis are the same as presented in this thesis.

A Numerical Considerations of Intention Estimation

Among the different intention estimation algorithms presented in Chapter 2, the hypothesis-based method with unknown start point is the most powerful. However, as it is an iterative method which requires solving a linear least-squares problem in each iteration step, it is computationally expensive. In the following, some simplifications are introduced which make the algorithm real-time capable on decent hardware.

Recalling from Sec. 2.3, the intention estimation is performed by fitting the observed trajectory data to a model function according to the following cost function:

$$G(t_f) = \frac{1}{2}\sum_{i=1}^{n_k} \|\hat{x}_i(t_f, \boldsymbol{p}(t_f)) - x_i\|^2, \tag{A.1}$$

where the estimates $\hat{x}_i(t_f, \boldsymbol{p})$ are calculated by the model function $h(t_i, t_f, \boldsymbol{p})$ as follows:

$$\begin{aligned} \hat{x}_i(t_f, \boldsymbol{p}) &= h(t_i, t_f, \boldsymbol{p}) \\ &= \boldsymbol{\tau}^T \boldsymbol{p} \\ &= \tau^5 p_5 + \tau^4 p_4 + \ldots + \tau p_1 + p_0 \\ &= (t_i - t_f)^5 p_5 + (t_i - t_f)^4 p_4 + \ldots + (t_i - t_f) p_1 + p_0. \end{aligned} \tag{A.2}$$

As the position, velocity, and acceleration are hypothesized for the final time t_f, the following boundary conditions at $t = t_f$ must be ensured:

$$h(t_f, t_f, \boldsymbol{p}) = p_0 = x_f \quad \dot{h}(t_f, t_f, \boldsymbol{p}) = p_1 = \dot{x}_f \quad \ddot{h}(t_f, t_f, \boldsymbol{p}) = 2p_2 = \ddot{x}_f. \tag{A.3}$$

If an estimate of $\hat{t}_f$ is given, the coefficient vector $\boldsymbol{p}$ can be found by linear-regression, which minimizes the following cost function:

$$G(\boldsymbol{p}) = \frac{1}{2}\sum_{i=1}^{n_k} \|\hat{x}_i(\hat{t_f}, \boldsymbol{p}) - x_i\|^2. \tag{A.4}$$

Minimizing (A.4) while observing (A.3), can be done by solving the matrix equation for $\boldsymbol{p}$:

$$\begin{pmatrix} S_{10} & S_9 & S_8 & S_7 & S_6 & S_5 \\ S_9 & S_8 & S_7 & S_6 & S_5 & S_4 \\ S_8 & S_7 & S_6 & S_5 & S_4 & S_3 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \end{pmatrix} \boldsymbol{p} = \begin{pmatrix} X_5 \\ X_4 \\ X_3 \\ x_f \\ \dot{x}_f \\ \ddot{x}_f \end{pmatrix}, \tag{A.5}$$

where S_r and X_r are defined as:

$$S_r = \sum_{i=1}^{n_k} \tau_i^r \quad \text{and} \quad X_r = \sum \tau_i^r x_i. \tag{A.6}$$

As the sum terms S_r and X_r depend on τ, which in turn depends on t_f, they cannot be calculated accumulatively, but must be completely recalculated whenever t_f changes. This creates the major part of the computational costs. However, if the values S_r and X_r are known for final time t_f, the corresponding values S'_r and X'_r for a different final time t'_f can be computed by a simple matrix multiplication. To this end, the values S_r and X_r are grouped in the vectors $\boldsymbol{s}$ and $\boldsymbol{\xi}$, where $\boldsymbol{s}$ is an 11 dimensional vector ($r = 0 \dots 10$), and $\boldsymbol{\xi}$ is a 6 dimensional vector ($r = 0 \dots 5$). The time-shifted vector $\boldsymbol{s}'$ can be calculated by:

$$\boldsymbol{s}' = \boldsymbol{\Delta}\, \boldsymbol{s}, \tag{A.7}$$

using the 11×11 matrix $\boldsymbol{\Delta}$, which is defined as:

$$\boldsymbol{\Delta}_{ij} = \begin{cases} 0 & \text{if} \quad i < j \\ 1 & \text{if} \quad i = j \\ \begin{pmatrix} i \\ j \end{pmatrix} (t'_f - t_f)^{i-j} & \text{if} \quad i > j \end{cases} \tag{A.8}$$

The time-shifted vector $\boldsymbol{\xi}'$ is calculated analogously. The computational costs of shifting the vector $\boldsymbol{s}$ and $\boldsymbol{\xi}$ are independent of the number of data.

As the vector $\boldsymbol{s}$ and the matrix $\boldsymbol{\Delta}$ contain powers up to 10, the high dynamic range can lead to considerable numerical inaccuracies when using double precision floating point numbers. Therefore, it is favorable to continuously shift the vectors $\boldsymbol{s}$ and $\boldsymbol{\xi}$ to $t_f = t_{n_k}$ such that the values of τ_i are small. Additionally, an exponential attenuation $\alpha < 1$ of older data points is introduced to guarantee stability:

$$\boldsymbol{s}' = \alpha^{(t'_f - t_f)/t_\alpha}\, \boldsymbol{\Delta}\, \boldsymbol{s}. \tag{A.9}$$

This attenuation is also a reasonable weighting of the data w.r.t. the estimation process itself, because older data points, which may reflect the current intention less accurately then newer ones, are weighted less.

B Telepresence Architecture

The telepresence architecture defines the structure of the telepresence system. It specifies which components are present in the system and how they are interconnected. Most telepresence systems, including the assisted telepresence system which is under investigation in this thesis, comprise a single haptic interface and a single telemanipulator, which are connected by a communication channel. The interaction between haptic interface and human operator, as well as the interaction between telemanipulator and remote environment, takes place in a single point.

The interconnection of the individual components describes the signals which are interchanged between haptic interface and teleoperator as well as the properties of the communication channel. The interchanged signals are classically positions, velocities, forces, or any combination of these. The most important properties of the communication channel are time delay, jitter, and possibly loss rate of information. It is a well-known fact in telepresence research that the interconnection has an essential effect on the stability and the transparency of the telepresence system, e.g. time delay in the communication channel can make a bilateral telepresence system unstable [28]. For a given architecture, the stability region is determined by the magnitude of time delay and the mechanical properties of human operator and remote environment.

In the context of this thesis, the specific demand on the telepresence architecture consists in integrating the assist functions. As the assist functions alter position and force signals in a closed-loop fashion, they are prone to destabilize the system. The telepresence architecture must, therefore, restrict the intervention of the assist functions in such a way that stability is maintained.

The remainder of this appendix is structured as follows: A review of existing telepresence architectures, which are relevant to the presented system, is given in Sec. B.1. From these architectures, the port-Hamiltonian framework constitutes the most promising approach to the specific needs of this thesis and is, thus, presented in more detail in Sec. B.2.

B.1 State of the Art

A telepresence system typically comprises two robots: the master robot, which interacts with the human operator, and the slave robot, which interacts with the remote environment. The telepresence architecture determines how these two robots are connected and controlled, i.e. which kind of signals are exchanged between the robots over the communication channel, and which signals are inputs of the local controllers. The telepresence architecture, therefore, deals with the control theoretic aspects of the telepresence system.

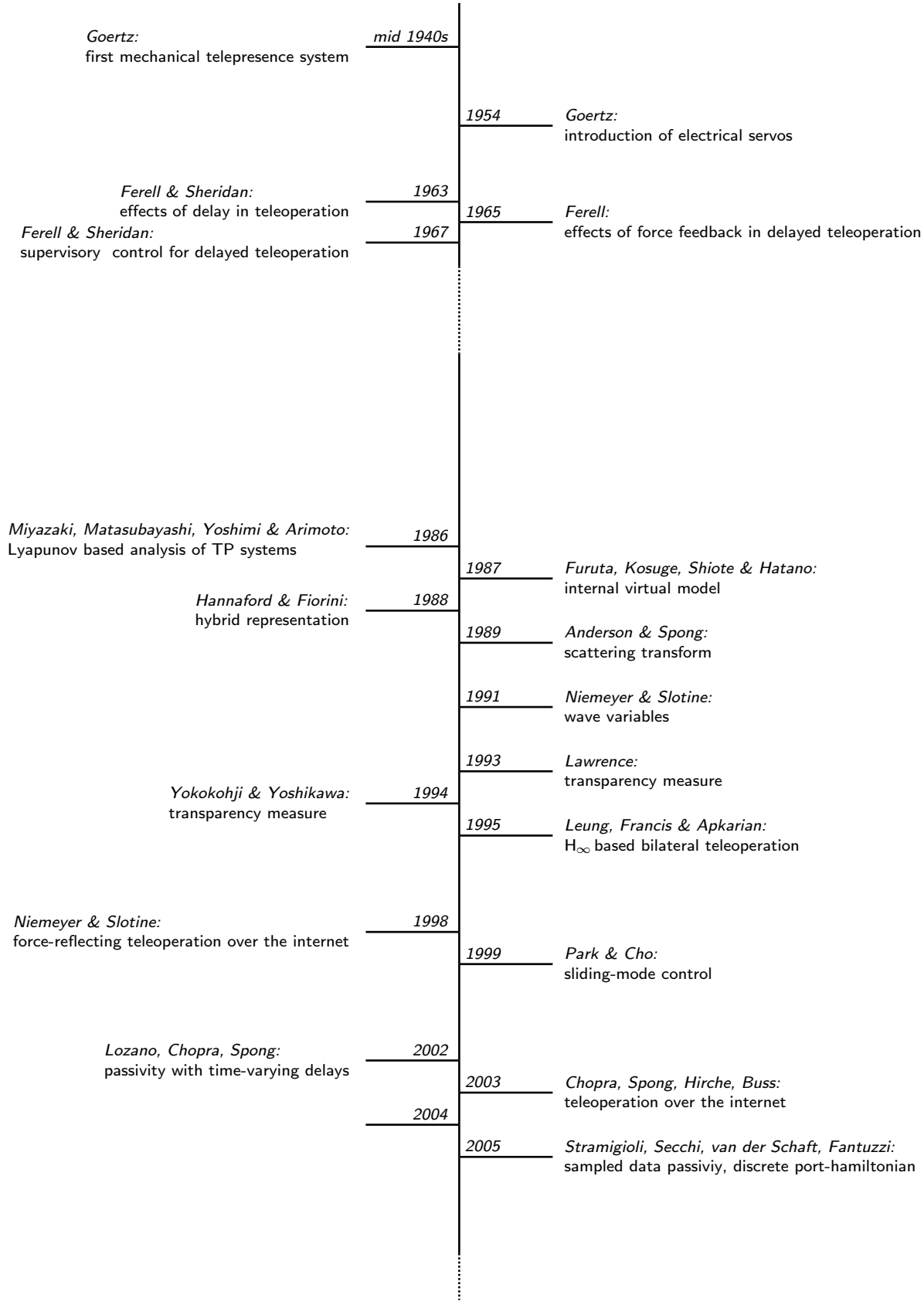

Figure B.1: Milestones in the research on telepresence systems.

In order to achieve a natural interaction via the telepresence system, signals must be exchanged bilaterally between the two robots, i.e. forces which are applied to the master robot result in forces applied to the remote environment, and forces applied to the slave robot result in forces applied to the human operator. This force reflection can provide a realistic impression of the mechanical properties of the remote environment. However, due to the closed control loop over the communication channel, it is prone to create instabilities when the communication channel induces a delay on the exchanged signals.

Throughout the history of telepresence (cf. Fig. B.1), the two objectives *stability* and *transparency* have dominated the research. As in any control system, stability is indispensable for a correct system behavior and, finally, for avoiding harm to human operator, remote environment, and the telepresence system itself. In the ideal case, the telepresence system cannot be destabilized no matter how the human operator or the remote environment behaves. Transparency is a desired property of the system, meaning that the human operator perceives the remote environment as if he was in direct contact with it. While true transparency cannot be technically achieved, the goal is to make the feeling as realistic as possible. The two afore-described objectives, stability and transparency, are hard to reconcile such that an appropriate balance must be found for a specific application.

After telepresence had made the transition from purely mechanical systems to electrically controlled servo-systems in the mid 1950s, the early research concentrated on the effects of delay in the communication channel [28, 89]. The results suggested that the execution time of a given task increases linearly with the amount of time delay. This insight triggered a plethora of investigations on new telepresence paradigms such as shared and supervisory control [29, 90]. The main idea of supervisory control consists in sending high-level commands to the teleoperator, which can be executed without human intervention such that the time delay in the communication channel plays a minor role. For the description of these high-level commands, many new languages were developed [32, 61, 68, 85]. A similar approach is given by predictive displays, which build a model of the remote environment as intermediate representation [14, 15, 43, 94]. If the manipulations are virtually executed in the intermediate representation and then transferred to the remote environment, the concept is called the hidden-robot concept [53].

After 1980, control theoretic aspects gained more attention in the telepresence research. In [71], a Lyapunov-based analysis of telepresence systems is conducted. A control approach based on an internal virtual model is presented in [34].

A major advance in stabilizing telepresence systems with time delay was reached in 1988 with the introduction of network theory and impedance representations of the mechanical properties [39]. Subsequently, more and more concepts were borrowed from network theory such as wave transmission, scattering theory, and passivity.

B.1.1 Two-Port Networks

The main building block from network theory, which was applied to telepresence system, is the two-port network. A two-port network interacts with its environment through two separate ports (see Fig. B.2). The two physical quantities entering a port always constitute

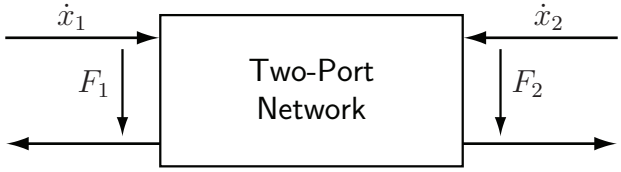

Figure B.2: A Two-Port Network. Each of the ports consist of an effort-flow pair, which is given by force F and velocity $\dot{x}$ in a mechanical system. The arrows indicate that power entering the system is counted positive.

an *effort-flow pair*. For mechanical systems, the effort-flow pair is formed by force F and velocity $\dot{x}$, for electrical systems, it is formed by voltage U and current I.

A whole telepresence system can be modeled as two port, where the interaction between human operator and telepresence systems takes place via one port, and the interaction between telepresence system and remote environment takes place via the other port. This two-port model can be refined by subdividing the telepresence system into several two-ports, namely the human-system interface, the communication channel, and the teleoperator. In a further refinement, the single devices can be split into one or more control blocks and the hardware, which are each modeled as two-ports.

Two-port networks inherently exhibit the energy flow between interconnected subsystems. And thus, the energy exchange between human operator and remote environment, which is an important property of a telepresence system, can also be easily modeled. The behavior of the communication channel with time delay can be modeled by a transmission line, which is a special type of two-port network.

For linear systems, the relationship between the signals at both ports can be expressed in the frequency domain by using impedance, admittance, or hybrid matrices, which are also well-known from network theory. The impedance matrix $\boldsymbol{Z}(s)$ takes the form:

$$\begin{pmatrix} F_1(s) \\ F_2(s) \end{pmatrix} = \boldsymbol{Z}(s) \begin{pmatrix} \dot{x}_1(s) \\ \dot{x}_2(s) \end{pmatrix} = \begin{pmatrix} z_m(s) - c_{11}(s) & -c_{12}(s) \\ -c_{21}(s) & z_s(s) - c_{22}(s) \end{pmatrix} \begin{pmatrix} \dot{x}_1(s) \\ \dot{x}_2(s) \end{pmatrix} \tag{B.1}$$

The hybrid matrix $\boldsymbol{H}(s)$ representation takes the form:

$$\begin{pmatrix} F_1(s) \\ -\dot{x}_2(s) \end{pmatrix} = \boldsymbol{H}(s) \begin{pmatrix} \dot{x}_1(s) \\ \dot{x}_2(s) \end{pmatrix} = \begin{pmatrix} h_{11}(s) & h_{12}(s) \\ h_{21}(s) & h_{22}(s) \end{pmatrix} \begin{pmatrix} \dot{x}_1(s) \\ F_2(s) \end{pmatrix} \tag{B.2}$$

B.1.2 Passivity Concept

Passivity has become the tool of choice for proving stability of telepresence systems. The passivity concept is based on the intuitive physical quantities power and energy and thereby exploits the nature of two-port networks. As only input-output properties of the system are considered, without the need of an exact dynamic model of the system, it is applicable to a wide range of system classes including linear and non-linear, time-continuous and time-discrete systems.

Fundamentally, the idea of passivity consists in monitoring the energy flow of a system. As long as the system absorbs more energy than it produces, it is said to be passive. Formally, the energy flow takes place through ports, which are defined by a pair of power variables e and f. Assuming that the effort variables of all ports entering the system are aggregated in the effort vector $\boldsymbol{e}$ and the corresponding flow variables are analogously aggregated in the flow vector $\boldsymbol{f}$, the total power entering the system is described by:

$$P_{in} = \boldsymbol{e}^T \boldsymbol{f}. \tag{B.3}$$

The system is *passive*, if it does not generate energy, i.e. if there exists a lower bounded, continuously differentiable energy storage function V such that

$$\dot{V} \leq \boldsymbol{e}^T \boldsymbol{f}. \tag{B.4}$$

If $\dot{V} < \boldsymbol{e}^T \boldsymbol{f}$, the system is *strictly passive*, and if $\dot{V} = \boldsymbol{e}^T \boldsymbol{f}$, the system is *lossless*.

The enormous usefulness of the passivity concept in network theory and telepresence systems stems from the fact that the interconnection of two passive systems, either in serial, parallel, or feedback connection, is again passive. A passive systems can be easily shown to be stable by using the sum of all energy storage functions as Lyapunov function.

B.1.3 Scattering Transform

With the introduction of the passivity concept, the instability of a telepresence system with time delay can be easily attributed to the non-passive behavior of the communication channel. In order to passivate the communication channel, it can be implemented as a transmission line, which is a well-known passive interconnection of systems in network theory.

In a transmission line, the energy exchange between master and slave is decomposed into an incident and a reflective wave, which are represented by the scattering variables $F(t)+\dot{x}(t)$ and $F(t) - \dot{x}(t)$, respectively. Both waves are related by the scattering matrix $\boldsymbol{S}$:

$$F(t) - \dot{x}(t) = \boldsymbol{S}(F(t) + \dot{x}(t)). \tag{B.5}$$

For a two-port network in hybrid representation, the scattering matrix can be directly obtained from the hybrid matrix (B.2):

$$\boldsymbol{S} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} (\boldsymbol{H}(s) - \boldsymbol{I})(\boldsymbol{H}(s) + \boldsymbol{I})^{-1} \tag{B.6}$$

In order to guarantee passivity and thus stability, the energy of the reflected wave must be smaller than or equal to the energy of the incident wave. This is equivalent to the statement that the infinity norm of the scattering matrix must be smaller than or equal to 1:

$$\|S(j\omega)\|_\infty \leq 1 \tag{B.7}$$

The scattering transform can be successfully applied to telepresence systems in order to stabilize systems with arbitrary, constant time-delay in the communication channel [8]. The guaranteed stability, however, comes at the cost of an unbounded position drift between operator and teleoperator.

B.1.4 Port-Hamiltonian Systems

Port-Hamiltonian systems constitute a recent framework which enables a unified description of concepts such as network interconnection, power ports, passivity, and sampled data systems [95]. Therefore, it is well-suited for modeling a general telepresence system as well as the assisted telepresence system described in this thesis.

An in-depth description of the mathematical foundations of port-Hamiltonian system can be found in [102]. A description which is tailored to the application in the assisted telepresence follows in Sec. B.2.

B.2 Port-Hamiltonian Based Telepresence Architecture

The port-Hamiltonian framework provides an intuitive description of interconnected physical systems, where the interconnection is described in terms of energy exchange. In this section, an introductory overview of the port-Hamiltonian formalism is given. A more comprehensive description can be found in [102].

B.2.1 Mathematical Background

Intuitively, a port-Hamiltonian system is modelled as an n-port system, where each port is associated with a pair of power-conjugated variables. The fundamental distinction between port-Hamiltonian system models and classical control models is, therefore, that the interconnection between systems is modelled as energy exchange rather than a mere signal interchange. Port-Hamiltonian systems are lossless systems and, thus, inherently passive and stable. Typical examples can be borrowed from network theory such as ideal transformers, gyrators, and transmission lines, as well as such complex systems as a complete ideal telepresence system.

The most general mathematical description consists of four ingredients:

- A vector space of the flow variables and a dual vector space of the corresponding effort variables. The effort-flow pairs represent the power ports through which the system interacts with its environment. These variables do not impose any causality on the signals, i.e. input and outputs are not explicitly assigned.
- A vector space of the state variables. The state variables represent the memory of the system and are generally associated with energy storage.

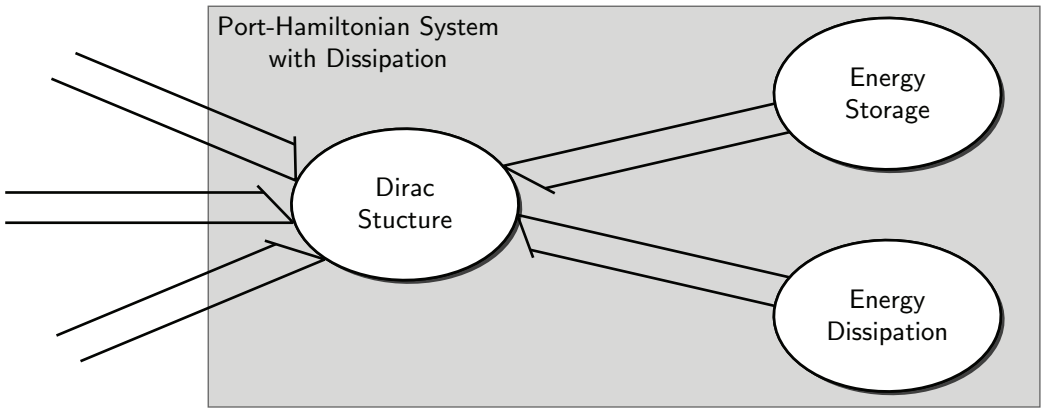

Figure B.3: Graphical representation of a port-Hamiltonian system with dissipation. The external power ports are directly connected to the Dirac structure. Additionally, internal ports for energy storage and energy dissipation are connected to the Dirac structure. The Dirac structure distributes the power among external and internal ports in a power-preserving way. The interconnection behavior is generally time-varying.

- A Hamiltonian function describing the total energy. The Hamiltonian function is a lower bounded, smooth function that maps variables from the state space to a scalar value, which specifies the energy of a given state.
- An interconnection structure describing the energy flow in the system. The interconnection structure takes the mathematical form of a Dirac structure. The Dirac structure defines how the energy is distributed between the power ports and the internal storage in a power-preserving way.

A graphical representation of this system description is depicted in Fig. B.3.

By introducing coordinates, the port-Hamiltonian formalism can be imposed on state-space system models, which increases its applicability to practical control problems. In this case, the state variables $\boldsymbol{x}$ are expressed in an n-dimensional vector space $\mathcal{X}$. The power variables are expressed by row- and column-vectors of equal dimensions, and their duality product which yields the transported power is given by the intrinsically defined matrix product. Furthermore, a causality relation is imposed on the power variables. Input variables $\boldsymbol{u}$ live in the vector space $\mathcal{U}$ and output variables $\boldsymbol{y}$ live in the vector space $\mathcal{Y} = \mathcal{U}^*$. The Hamiltonian function is denoted $H : \mathcal{X} \to \mathcal{R}$. The Dirac structure can be implemented by a skew-symmetric matrix $\boldsymbol{J}$ and the power dissipation by a symmetric, positive semidefinite matrix $\boldsymbol{R}$.

With these definitions the state space description of the system can be expressed as:

$$\dot{\boldsymbol{x}} = (\boldsymbol{J}(\boldsymbol{x}) - \boldsymbol{R}(\boldsymbol{x}))\frac{\partial H}{\partial \boldsymbol{x}} + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u} \tag{B.8}$$

$$\boldsymbol{y} = \boldsymbol{g}^T(\boldsymbol{x})\frac{\partial H}{\partial \boldsymbol{x}} \tag{B.9}$$

B.2.2 Control of Admittance-Type Devices

Most existing telepresence architectures are primarily designed for impedance-type haptic interfaces. These can be approximately regarded as force sources because of their low mass, low friction, and good backdrivability. However, their application is restricted to areas with small workspace and low interaction forces lest mass and friction increase. In contrast, admittance-type devices are characterized by heavy-weight constructions with non-negligible mass and friction. In order to compensate these unwanted system properties, typically a force-sensor is added at the end-effector and used in an admittance control.

An *admittance control* is composed of a virtual dynamics equation and a low-level position controller (see Fig. B.4). The virtual dynamics is represented by an admittance equation which computes desired positions from measured forces. A natural choice for the admittance equation are the rigid-body dynamics of a mass-damper element. The position controller is usually implemented as PD-controller and implements the desired position. By this combination, the controlled device behaves according to the virtual dynamics thereby hiding the unwanted physical properties of the device itself.

A stability analysis of admittance-type devices can be found in [76]. It shows that low values of virtual mass and virtual damping are prone to destabilize the control. The exact stability boundaries depend on the environment the device is interacting with. A physically intuitive explanation for these instabilities is given below.

As explained above, the control of port-Hamiltonian systems is favorably modeled as energy-based interaction of the plant system and the control system such that the amalgamated system exhibits the desired properties. As port-Hamiltonian systems are always passive, the controller can only shape the dynamics of the plant, but cannot inject energy into the plant, which is necessary to produce some active behavior. Thus, the controller must be provided with energy through an additional interaction port. By these considerations, the controller is split into a dynamic shaping part and an energy supplying part. The former part is also known as *intrinsically passive controller (IPC)*.

In Fig. B.5, the mechanical representation of an intrinsically passive controller of an admittance type device is shown. The mass m_d is called *virtual object* and sets the desired

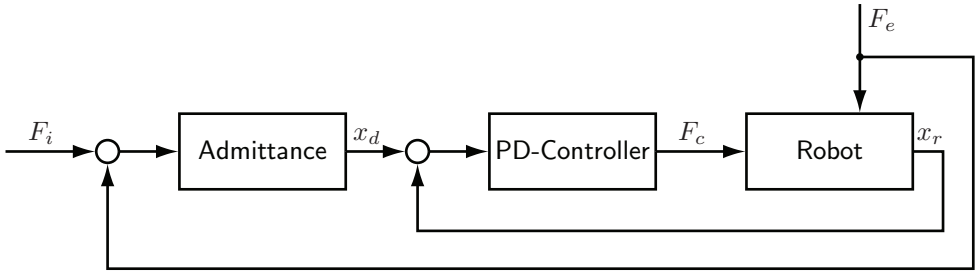

Figure B.4: Block diagram of admittance conrol scheme for one degree of freedom. The desired position x_d is calculated based on the combined interal and external force F_i and F_e in accordance with the virtual dynamics, which is represented in admittance form. The robot is controlled to this position by a linear PD-controller.

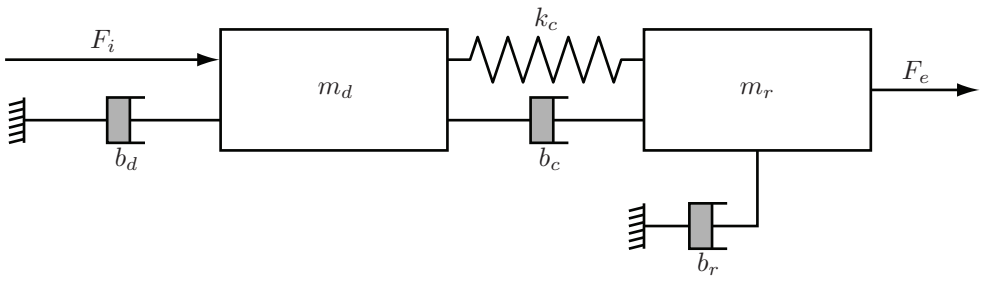

Figure B.5: Mechanical equivalent of the intrinsically passive control scheme. The desired dynamics are modelled by the mass-damper element composed of m_d and b_d which is actuated by the internal force F_i. The actual robot dynamics are modelled by mass m_r and damper b_r, and the coupling between them by the spring-damper element composed of m_c and b_c.

behavior of the robot. By means of the damper b_d some damping can be injected to stabilize the system. The virtual spring k_c and damper b_c couple the virtual object to the physical robot, which is modelled by m_r and b_r. The robot interacts with the environment by the external force F_e. Additionally, the model is coupled to the internal force F_i, which is used to interact with a virtual world or remote scene. As all elements of the intrinsically passive controller are passive, hence its name, the combined system is also passive.

Comparing the admittance control and the intrinsically passive control scheme reveals many similarities. The damped virtual object in the IPC is excited by the internal force and indirectly by the external force, thus producing a similar behavior as the admittance equation in the admittance control. Likewise, the spring-damper coupling in the IPC exerts a force on the robot which is identical to the force commanded by the PD-controller in the admittance control. However, there is a fundamental difference: While the force generated by the spring k_c and b_c also acts on the virtual object m_d, the PD-controller has no direct repercussions on the virtual dynamics. In other words, the PD-controller can draw from an infinite supply of energy, whereas the IPC can only invest the energy stored in the mass m_d, which must be resupplied through F_i. By monitoring and resupplying the amount of energy which is dissipated in the damping of the admittance-type haptic device, the haptic device can be made more transparent without getting unstable.

B.2.3 Intrinsically Passive Telepresence System

Connecting an intrinisically passive controlled haptic interface with an intrinisically passive controlled teleoperator through a passive communication channel brings forth a completely passive telepresence system. For a constant time delay, the passive communication channel can be realized by applying the scattering transform to the transmitted variables. If admittance-type devices are used as haptic interface and teleoperator, the previously presented intrinsicially passive controllers with energy-resupply must be used to increase the level of transparency.

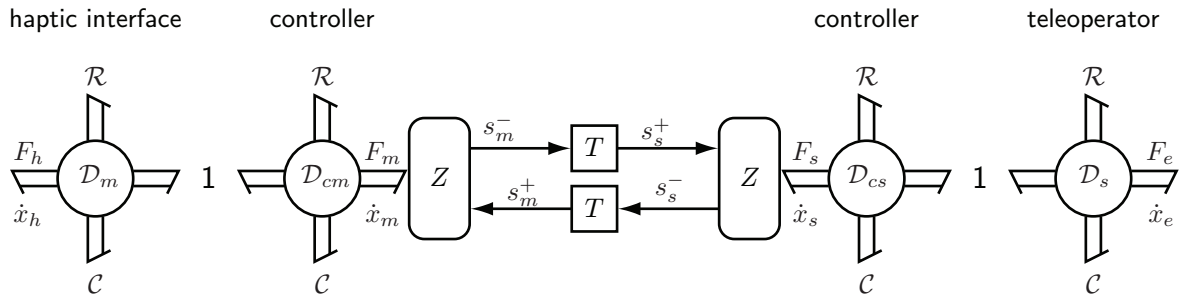

Figure B.6: Port-Hamiltonian representation of an intrinsically passive telepresence system. The structure comprises an intrisically passive controller for haptic interface and teleoperator. These are intercontected by a passified communication channel with constant time delay T and line impedance Z. Devices as well as controller are connected to energy dissipating elements $\mathcal{R}$ and energy storing elements $\mathcal{C}$.

The intrinsically passive telepresence system is shown in Fig. B.6 in a port-Hamiltonian representation. The human interacts with the system via the power port of the haptic interface which is formed by F_h and $\dot{x}_h$. On the other side, the haptic interface is conntected to an intrinisically passive controller in a negative feedback connection. The symmetric structure can be found on teleoperator side where the teleoperator interacts with the environment through the power port formed by F_e and $\dot{x}_e$. Both controllers are interconnected through a scattering-based communication channel. Haptic interface, teleoperator, and the two intrinsically passive controllers have energy dissipating elements $\mathcal{R}$ and energy storing elements $\mathcal{C}$.

B.2.4 Passive Implementation of Assist Functions

The assist functions presented in Chapter 4 are based on scaling velocities or forces on teleoperator side. While this does not render the system non-passive in freespace motions, passivity can be violated in contact situations. The port-Hamiltonian formulation of the intrinsically passive telepresence system offers an elegant way to passivate this scaling.

A transformer is a two-port network which scales the input quantities of both ports. As efforts and flows are scaled with an inverse scaling, the behavior of the transformer is passive and lossless. Therefore, the transformer can be represented as a port-Hamiltonian system without energy storage and dissipation. Even when the scaling ratio is time-variant, the passivity property is not violated, which makes the transformer the ideal element to implement scaling-based assist functions.

In Fig. B.7, the teleoperator side of the intrinsically passive telepresence system with added transformer is shown. The transformer is inserted between the transmission line and the controller of the teleoperator.

Recalling the velocity scaling presented in Sec. 4.1, the output velocity of the teleoperator can be expressed as the product of the operator velocity and a dynamically changing scaling factor:

$$\Delta\dot{\boldsymbol{x}}_T^{\perp} = f(\Delta\boldsymbol{x}_T, \Delta\dot{\boldsymbol{x}}_O^{\perp})\,\Delta\dot{\boldsymbol{x}}_O^{\perp}. \tag{B.10}$$

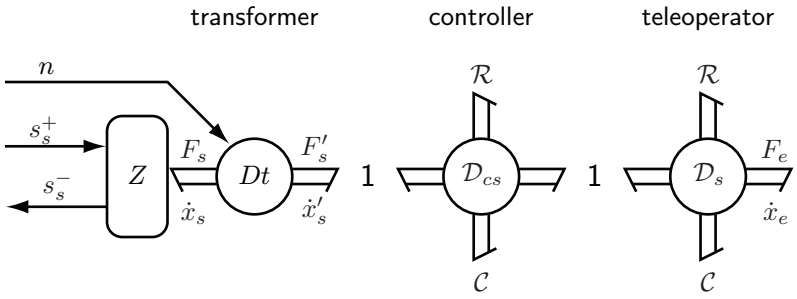

Figure B.7: Port-Hamiltonian diagram of an assisted passive telepresence system.

By setting the scaling ratio n to $f(\Delta \boldsymbol{x}_T, \Delta \dot{\boldsymbol{x}}_O)$, the behavior of the assist function is implemented in a passive way. In freespace motions the resulting inverse scaling of the forces has no effect, because there are no environment forces. However, in scenarios with damping, e.g. underwater applications, attenuated velocities on teleoperator side lead to attenuated forces on operator side and amplified velocities lead to amplified forces in order to keep the power balance.

C Experimental Hardware

C.1 Haptic Devices

The haptic devices are chosen in response to the requirements of the evaluation experiments. In order to keep unwanted disturbances at a minimum, the simplest possible hardware which fulfills the given requirements is employed.

C.1.1 Haptic Devices with 1 DOF

A haptic device with one linear degree of freedom is used in one experiment as both, haptic interface and telemanipulator. The device is composed of a high-performance linear direct drive *ThrustTube ME2504* by Copley Corp., a high-resolution optical encoder, and a force-sensor, which measures the interaction forces with the environment. The technical specifications are listed in Table C.1.

property	value	unit
workspace size	1,612	mm
position resolution	1.0	μm
maximum peak force	624	N
maximum continuous force	44.6	N
maximum acceleration	589	m/s^2
maximum velocity	14.5	m/s

Table C.1: Technical Data of Linear Device. ThrustTube ME2504, Copley Corp.

The controller of the device is implemented in software, which is executed on standard PC hardware (see below).

C.1.2 Haptic Devices with 6 DOF

Experiments which require motions with more degrees of freedom are performed by using the large-scale haptic display ViSHaRD10 and an anthropomorphic telemanipulator, which both provide six degrees of freedom of the end-effector.

The *Virtual Scenario Haptic Rendering Device with 10 actuated degrees of freedom (ViSHaRD10)* is a custom-made haptic interface, which was designed to provided a large, singularity-free workspace and large interaction forces. This is achieved by a serial kinematic structure with 10 actuated degrees of freedom, which provides sufficient redundancy to avoid singularities. The technical specifications are summarized in Table C.2, more in-depth information can be found in [99].

The anthropomorphic telemanipulator is a robot arm with human-like shape. The design comprises seven actuated degrees of freedom distributed over two spherical joints at wrist and shoulder and one revolute joint at the elbow. The robot arm was constructed to approximately match the force and motion capabilities of the human arm. The resulting specifications are shown in Table C.2. A comprehensive description of the robot arms is given in [93].

property	ViSHaRD10	Telemanipulator	unit
workspace size	cylinder	half-sphere	
	$r = 0.85$, $h = 0.6$	$r = 0.86$	m
position resolution	< 1.0	< 1.0	μm
maximum peak force	170	220	N
maximum velocity	> 1.0	> 1.0	m/s

Table C.2: Technical Data of 6-DOF Haptic Devices. Haptic Display ViSHaRD10. Anthropomorphic Telemanipulator.

C.1.3 Control of Haptic Devices

All haptic devices are digitally controlled by appropriate software controllers. These run on standard PC hardware which is equipped with I/O-extension cards to connect to sensors and actuators. In order to allow executing the algorithm under real-time conditions, a Linux operating system is used with an RTAI patched kernel. For details, refer to *www.linux.org* and *www.rtai.org*. The actual control algorithms are either programmed directly in C/C++ or generated in Matlab/Simulink by using RealTime Workshop.

The control structure is depicted in Fig. C.1. Due to the relatively large mass and friction of the employed haptic devices, an admittance control scheme is used to control the devices. This requires to measure the interaction force between device and environment F_e with a *Force/Torque Sensor (FTS)*. This force is combine with the internal force F_i, which is generated by the peer device or the assist system. The sum force is input to the admittance controller. The admittance controller computes a desired position from the input force by rendering a virtual dynamics equation. For devices with multiple degrees of freedom, the desired position is transformed into joint space by an inverse kinematics. Finally, the desired joint angles are controlled by a PD-controller, which commands motor torques τ_d.

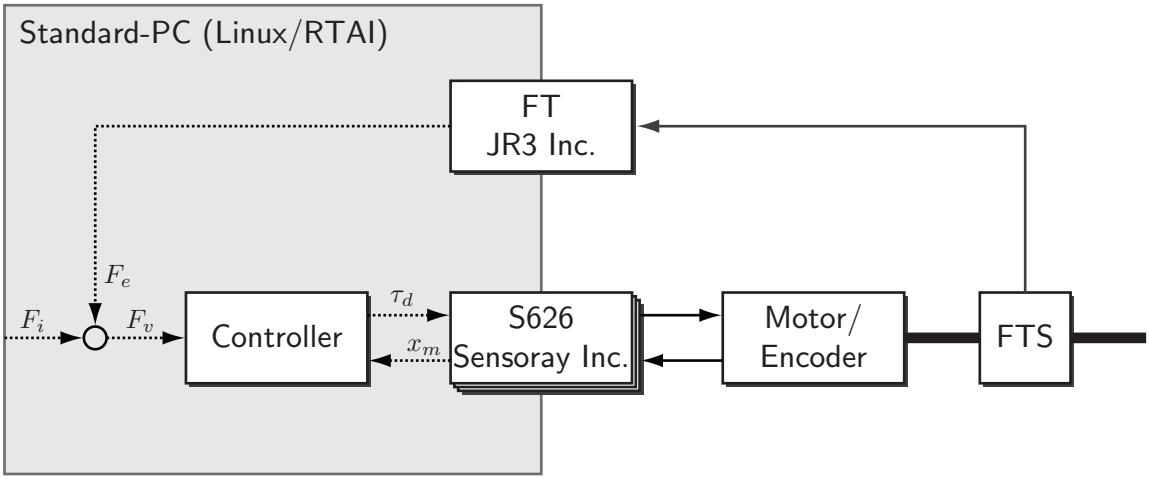

Figure C.1: Control Structure.

The force measurements are interfaced to the PC by a proprietary sensor card from JR3 Inc. The position measurements of the devices, which are obtained by quadrature encoders, and the commanded motor torque are exchanged by using S626 multi-function DAQ boards by Sensoray Inc.

C.2 Graphic Devices

Visual feedback is presented to the human operator via a head-mounted display (HMD) nVisor SX by NVIS, inc. It displays images of SXGA resolution with 1280×1024 pixels at a maximum frame-rate of 60 Hz. The field-of-view covers approx. 50°.

In order to better control the visual feedback, a virtual reality environment is used instead of the real camera images delivered by the teleoperator. The virtual reality consists of a model of the real teleoperator equipped with a screwdriver and some basic objects such as a table. The scene is rendered by the free *Open Inventor* implementation *Coin3D* running on a 64-bit Linux machine with dual-port graphics card. The scene is updated at a rate of 30 Hz.

The virtual scene is provided with measurement data of the end-effector position and the head pose of the operator by an ethernet connection with negligible time delay. The new measurement values are incorporated in the next rendered image such that no perceivable delay occurs.

Bibliography

[1] Intuitive Surgical, Inc.
http://www.intuitivesurgical.com, 2009. Sunnyvale, California, USA.

[2] telerob gesellschaft für fernhantierungstechnik mbh.
http://www.telerob.de, 2009. Ostfildern, Germany.

[3] Jake J. Abbott, Gregory D. Hager, and Allison M. Okamura. Steady-hand teleoperation with virtual fixtures. In *Proceedings of the 12th IEEE International Workshop on Robot and Human Interactive Communication.*, pages 145–151, 2003.

[4] Jake J. Abbott, Panadda Marayong, and Allison M. Okamura. *Robotics Research*, chapter Haptic Virtual Fixtures for Robot-Assisted Manipulation, pages 49–64. Springer Berlin / Heidelberg, 2007.

[5] Jake J. Abbott and Allison M. Okamura. Virtual fixture architectures for telemanipulation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 2798–2805, 2003.

[6] Jake J. Abbott and Allison M. Okamura. Pseudo-admittance bilateral telemanipulation with guidance virtual fixtures. In *Proceedings of the 14th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 169–175, 2006.

[7] Mehdi Ammi and Antoine Ferreira. Robotic assisted micromanipulation system using virtual fixtures and metaphors. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 454–460, 2007.

[8] Robert J. Anderson and Mark W. Spong. Bilateral control of teleoperators with time delay. *IEEE Transactions on Automatic Control*, 34(5):494–501, May 1989.

[9] Rafael Aracil and Manuel Ferre. Telerobotics for aerial live power line maintenance. In Manuel Ferre, Martin Buss, Rafael Aracil, Claudio Melchiorri, and Carlos Balaguer, editors, *Springer Tracts in Advanced Robotics: Advances in Telerobotics*, volume 31. Springer Berlin / Heidelberg, 2007.

[10] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, Sep. 1987.

[11] Ronald T. Azuma. A survey of augmented reality. *Presence Teleoperators and Virtual Environments*, 6(4):355–385, August 1997.

[12] H. C. Barrett, Peter M. Todd, Geoffrey F. Miller, and Philip W. Blythe. Accurate judgments of intention from motion cues alone: A cross-cultural study. *Evolution and Human Behavior*, 26(4):313–331, July 2005.

[13] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, June 2008.

[14] Antal K. Bejczy and Won S. Kim. Predictive displays and shared compliance control for time-delayed telemanipulation. In *Proceedings of the 1990 IEEE International Workshop on Intelligent Robots and Systems*, 1990.

[15] Antal K. Bejczy, Won S. Kim, and Steven C. Venema. The phantom robot: predictive displays for teleoperation with timedelay. In *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, Cincinnati, Ohio, USA, May 1990.

[16] Dimitri P. Bertsekas. *Nonlinear programming*. Athena Scientific, 2003.

[17] Sarah-Jayne Blakemore and Jean Decety. From the perception of action to the understanding of intention. *Nature Reviews Neuroscience*, 2:561–567, August 2001.

[18] Jürgen Bortz and Nicola Döring. *Forschungsmethoden und Evaluation: für Human- und Sozialwissenschaftler*. Springer, Berlin, 2006.

[19] R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs. *Cognitive Psychology*, 39:324–335, 1952.

[20] Michael Bratman. Intention and personal policies. *Philosophical Perspectives*, 3:443–469, 1989.

[21] Martin Buss, Kwang-Kyu Lee, Norbert Nitzsche, Angelika Peer, Bartlomiej Stanczyk, and Ulrich Unterhinninghofen. Advanced telerobotics: Dual-handed and mobile remote manipulation. In Manuel Ferre, Martin Buss, Rafael Aracil, Claudio Melchiorri, and Carlos Balaguer, editors, *Advances in Telerobotics*, volume 31. Springer Tracts in Advanced Robotics: Advances in Telerobotics, 2007.

[22] Michael A. Peshkin Carl A. Moore and and J. Edward Colgate. Cobot implementation of virtual paths and 3-d virtual surfaces. *IEEE Transactions on Robotics and Automation*, 19(2):347–351, April 2003.

[23] B. Corteville, E. Aertbelien, H. Bruyninckx, J. De Schutter, and H. Van Brussel. Human-inspired robot assistant for fast point-to-point movements. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3639–3644, 2007.

[24] Paolo Dario, Blake Hannaford, and Arianna Menciassi. Smart surgical tools and augmenting devices. *IEEE Transactions on Robotics and Automation*, 19(5):782–792, October 2003.

[25] Daniel DeMenthon and Larry S. Davis. Exact and approximate solutions of the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(11):1100–1105, Nov 1992.

[26] Michel Dhome, Marc Richetin, Jean-Thierry Lapreste, and Gerard Rives. Determination of the attitude of 3d objects from a singleperspective view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(12):1265–1278, Dec 1989.

[27] Paul Evrard, Nicolas Mansard, Olivier Stasse, Abderrahmane Kheddar, Thomas Schauš, Carolina Weber, Angelika Peer, and Martin Buss. Intercontinental, multimodal, wide-range tele-cooperation using a humanoid robot. In *Proceedings of the International Conference on Intelligent Robots and Systems*, St. Louis, Missouri, USA, 2009.

[28] W. R. Ferrell. Remote manipulation with transmission delay. *IEEE Transactions on Human Factors in Electronics*, 6:24–32, 1965.

[29] W. R. Ferrell and T. B. Sheridan. Supervisory control of remote manipulation. *IEEE Spectrum*, pages 81–88, 1967.

[30] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[31] Tamar Flash and Neville Hogan. The coordination of arm movements: An experimentally confirmed mathematical model. *The Journal of Neuroscience*, 5(7):1688–1703, July 1985.

[32] C.P. Fong, R.S. Dotson, and A.K. Bejczy. Distributed microcomputer control system for advanced teleoperation. In *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, 1986.

[33] Jan-Michael Frahm, Kevin Köser, and Reinhard Koch. Pose estimation for multicamera systems. *Lecture Notes in Computer Science, Pattern Recognition*, 3175:286–293, 2004.

[34] Katsuhisa Furuta, Kazuhiro Kosuge, Yoshinori Shiote, and Hiromu Hatano. Master-slave manipulator based on virtual internal model following control concept. In *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, March 1987.

[35] Sundaram Ganapathy. Decomposition of transformation matrices for robot vision. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 130–139, March 1984.

[36] Weston Blaine Griffin. *Shared Control for Dexterous Telemanipulation with Haptic Feedback*. PhD thesis, Stanford University, Palo Alto, USA, June 2003.

[37] Chuanfan Guo, Tzyh-Jong Tarn, Ning Xi, and Antal K. Bejczy. Fusion of human and machine intelligence for telerobotic systems. In *Proceedings of the 1995 IEEE International Conference on Robotics and Automation*, Nagoya, Japan, May 1995.

[38] Blake Hannaford. A design framework for teleoperators with kinesthetic feedback. *IEEE Transactions on Robotics and Automation*, 5(4):426–434, August 1989.

[39] Blake Hannaford and Paolo Fiorini. A detailed model of bi-lateral teleoperation. In *Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics*, 1988.

[40] Robert M. Haralick, Chung nan Lee, Karsten Ottenberg, and Michael Nölle. Analysis and solutions of the three point perspective poseestimation problem. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 592–598, Maui, Hawaii, USA, June 1991.

[41] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Proceedings of The Fourth Alvey Vision Conference*, 1988.

[42] Sandra Hirche. *Haptic Telepresence in Packet Switched Communication Networks*. PhD thesis, Technische Universität München, München, June 2005.

[43] G. Hirzinger, J. Heindl, and K. Landzettel. Predictive and knowledge-based telerobotic control concepts. In *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, Scottsdale, Arizona, USA, May 1989.

[44] Bruce Hoff and Michael A. Arbib. Models of trajectory formation and temporal interaction of reach and grasp. *Journal of Motor Behavior*, 25(3):175–192, September 1993.

[45] Radu Horaud, Bernard Conio, Olivier Leboulleux, and Bernard Lacolle. An analytic solution for the perspective 4-point problem. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*, pages 500–507, San Diego, California, USA, June 1989.

[46] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4:629–642, 1987.

[47] Berthold K. P. Horn, Hugh M. Hilden, and Shahriar Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America*, 5:1127–1136, 1988.

[48] Tomotaka Itoh, Kazuhiro Kosuge, and Toshio Fukuda. Human-machine cooperative telemanipulation with motion and force scaling using task-oriented virtual tool dynamics. *IEEE Transactions on Robotics and Automation*, 16(5):505–516, October 2000.

[49] L. A. Jones and I. W. Hunter. A perceptual analysis of stiffness. *Experimental Brain Research*, 79(1):150–156, January 1990.

[50] Dong-Joong Kang, Jong-Eun Ha, and Mun-Ho Jeong. A method for camera pose estimation from object of a known shape. *Lecture Notes in Control and Information Sciences, Intelligent Computing in Signal Processing and Pattern Recognition*, 345:606–613, 2006.

[51] Mitsuo Kawato. Trajectory formation in arm movements: minimization principles and procedures. In Howard N. Zelaznik, editor, *Advances in Motor Learning and Control*, pages 225–259. Human Kinetics, 1996.

[52] Fakheredine Keyrouz and Klaus Diepold. Binaural source localization and spatial audio reproduction for telepresence applications. *Presence Teleoperators and Virtual Environments*, 16(5):509–522, October 2007.

[53] A. Kheddar, C. Tzafestas, and P. Coiffet. The hidden robot concept-high level abstraction teleoperation. In *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1818–1825, Grenoble, France, September 1997.

[54] Abderrahmane Kheddar. Teleoperation based on the hidden robot concept. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 31(1):1–13, January 2001.

[55] Alexander Kron, Günther Schmidt, Bernd Petzold, Michael Zäh, Peter Hinterseer, and Eckehard Steinbach. Disposal of explosive ordnances by means of a bimanual haptic telepresence system. In *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2004.

[56] Katherine J. Kuchenbecker, Jonathan Fiene, and Günter Niemeyer. Improving contact realism through event-based haptic feedback. *IEEE Transactions on Visualization and Computer Graphics*, 12(2):219–230, March/April 2006.

[57] Katherine J. Kuchenbecker and Günter Niemeyer. Improving telerobotic touch via high-frequency acceleration matching. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, Orlando, Florida, USA, May 2006.

[58] R. H. LaMotte. Softness discrimination with a tool. *Journal of Neurophysiology*, 83:1777–1786, 2000.

[59] Dale A. Lawrence. Stability and transparency in bilateral teleoperation. *IEEE Transactions on Robotics and Automation*, 9(5):624–637, October 1993.

[60] Dale A. Lawrence, Lucy Y. Pao, Anne M. Dougherty, Mark A. Salada, and Yiannis Pavlou. Rate-hardness: A new performance metric for haptic interfaces. *IEEE Transactions on Robotics and Automation*, 16(4):357–371, August 2000.

[61] S. Lee, G. Bekey, and A. K. Bejczy. Computer control of space-borne teleoperators with sensory feedback. In *Proceedings of the 1985 IEEE International Conference on Robotics and Automation.*, March 1985.

[62] Ming Li and Allison M. Okamura. Recognition of operator motions for real-time assistance using virtual fixtures. In *Proceedings of the 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, March 2003.

[63] Henry C. Lin, Izhak Shafran, Todd E. Murphy, Allison M. Okamura, David D. Yuh, and Gregory D. Hager. Automatic detection and segmentation of robot-assisted surgical motions. In *Proceedings of the Medical Image Computing and Computer-Assisted Intervention MICCAI 2005*, 2005.

[64] Tony Lindeberg. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. *International Journal of Computer Vision*, 11(3):283–318, December 1993.

[65] David G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395, Mar 1987.

[66] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[67] Chien-Ping Lu, Gregory D. Hager, and Eric Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):610–622, June 2000.

[68] Azad M. Madni, Yee yeen Chu, and Amos Freedy. Intelligent interface for remote supervision and control of underwater manipulation. In *Proceedings of the Oceans*, August 1983.

[69] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, January 2004.

[70] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, October 2005.

[71] F. Miyazaki, S. Matsubayashi, T. Yoshimi, and S. Arimoto. A new control methodology toward advanced teleoperation of master-slave robot systems. In *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, 1986.

[72] Eri Nakano, Hiroshi Imamizu, Rieko Osu, Yoji Uno, Hiroaki Gomi, Toshinori Yoshioka, and Mitsuo Kawato. Quantitative examinations of internal representations for arm trajectory planning: Minimum commanded torque change model. *Journal of Neurophysiology*, 81(5):2140–2155, May 1999.

[73] Allison M. Okamura, Jack T. Dennerlein, and Robert D. Howe. Vibration feedback models for virtual environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 1998.

[74] Tobias Ortmaier, Barbara Deml, Bernhard Kbler, Georg Passig, Detlef Reintsema, and Ulrich Seibold. Robot assisted force feedback surgery. In Manuel Ferre, Martin Buss, Rafael Aracil, Claudio Melchiorri, and Carlos Balaguer, editors, *Springer Tracts in Advanced Robotics: Advances in Telerobotics*, volume 31. Springer Berlin / Heidelberg, 2007.

[75] Young S. Park, Hyosig Kang, Thomas F. Ewing, Eric L. Faulring, J. Edward Colgate, and Michael A. Peshkin. Enhanced teleoperation for d&d. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, New Orleans, Louisiana, USA, April 2004.

[76] Angelika Peer. *Design and Control of Admittance-Type Telemanipulation Systems*. PhD thesis, Technische Universität München, 2008.

[77] Angelika Peer, Bartlomiej Stanczyk, and Martin Buss. Haptic Telemanipulation with Dissimilar Kinematics. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2483–2487, Edmonton, Canada, 2005.

[78] Helena Pongrac, Jan Leupold, Stephan Behrendt, Berthold Färber, and Georg Färber. Human factors for enhancing live video streams with virtual reality: Performance, situation awareness, and feeling of telepresence. *Presence: Teleoperators and Virtual Environments*, 16(5):488–508, October 2007.

[79] Detlef Reintsema, Klaus Landzettel, and Gerd Hirzinger. Dlr's advanced telerobotic concepts and experiments for on-orbit servicing. In Manuel Ferre, Martin Buss, Rafael Aracil, Claudio Melchiorri, and Carlos Balaguer, editors, *Springer Tracts in Advanced Robotics: Advances in Telerobotics*, volume 31. Springer Berlin / Heidelberg, 2007.

[80] Detlef Reintsema, Carsten Preusche, Tobias Ortmaier, and Gerd Hirzinger. Toward high-fidelity telepresence in space and surgery robotics. *Presence Teleoperators and Virtual Environments*, 13(1):77–98, March 2004.

[81] P. Richard and P. Coiffet. Human perceptual issues in virtual environments: Sensory substitution and information redundancy. In *Proceedings of the 4th IEEE International Workshop on Robot and Human Communication*, pages 301–306, Tokyo, Japan, July 1995.

[82] Magnus J. E. Richardson and Tamar Flash. Comparing smooth arm movements with the two-thirds power law and the related segmented-control hypothesis. *The Journal of Neuroscience*, 22(18):8201–8211, September 2002.

[83] Pere Ridao, Marc Carreras, Emili Hernandez, and Narcis Palomeras. Underwater telerobotics for collaborative research. In Manuel Ferre, Martin Buss, Rafael Aracil, Claudio Melchiorri, and Carlos Balaguer, editors, *Springer Tracts in Advanced Robotics: Advances in Telerobotics*, volume 31. Springer Berlin / Heidelberg, 2007.

[84] Volker Rodehorst, Matthias Heinrichs, and Olaf Hellwich. Evaluation of relative pose estimation methods for multi-camera setups. In *Proceedings of the XXI Congress of the International Society for Photogrammetry and Remote Sensing*, pages 135–140, Beijing, China, July 2008.

[85] Tomomasa Sato and Shigeoki Hirai. Language-aided robotic teleoperation system (larts) for advanced teleoperation. *IEEE Journal of Robotics and Automation*, 3(5):476–481, October 1987.

[86] Jeffrey A. Saunders and David C. Knill. Visual feedback control of hand movements. *The Journal of Neuroscience*, 13(24):3223–3234, March 2004.

[87] Rainer Scheuchenpflug. Measuring presence in virtual environments. In M. J. Smith, G. Salvendey, and M. R. Kasdorf, editors, *Proceedings of the 9th International Conference on Human-Computer Interaction*, pages 56–58, 2001.

[88] Martin Schrepp, Theo Held, and Patrick Fischer. Untersuchung von designpräferenzen mit hilfe von skalierungsmethoden. *MMI-Interaktiv*, 13:72–82, 2007.

[89] T. B. Sheridan and W. R. Ferrell. Remote manipulative control with transmission delay. *IEEE Transactions on Human Factors in Electronics*, HFE-4(1):25–29, September 1963.

[90] Thomas B. Sheridan. *Telerobotics, Automation, and Human Supervisory Control*. MIT Press, 1992.

[91] Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 245–254, San Francisco, California, USA, July 1985.

[92] J. F. Soechting and F. Lacquaniti. Invariant characteristics of a pointing movement in man. *Journal of Neuroscience*, 1:710–720, 1981.

[93] Bartlomiej Stanczyk. *Development and Control of an Anthropomorphic Telerobotic System*. PhD thesis, Technische Universität München, 2006.

[94] Lawrence Stark, Won-Soo Kim, Frank Tendick, Blake Hannaford, Stephen Ellis, Mark Denome, Mary Duffy, Tim Hayes, Ted Jordan, Mark Lawton, Tim Mills, Robert Peterson, Kathleen Sanders, Mitch Tyler, and Steven van Dyke. Telerobotics: Display, control, and communication problems. *IEEE Journal of Robotics and Automation*, 3(1):67–75, February 1987.

[95] Stefano Stramigioli, Cristian Secchi, Arjan J. van der Schaft, and Cesare Fantuzzi. Sampled data systems passivity and discrete port-hamiltonian systems. *IEEE Transactions on Robotics*, 21(4):574–583, August 2005.

[96] Alireza Tavakkoli, Richard Kelley, Christopher King, Mircea Nicolescu, Monica Nicolescu, and George Bebis. A vision-based architecture for intent recognition. *Lecture Notes in Computer Science, Advances in Visual Computing*, 4842:173–182, 2007.

[97] Michael Tomasello, Malinda Carpenter, Josep Call, Tanya Behne, and Henrike Moll. Understanding and sharing intentions: The origins of cultural cognition. *Behavioral and Brain Sciences*, 28(5):675–735, October 2005.

[98] Nicolas Turro, Oussama Khatib, and Eve Coste-Maniere. Haptically augmented teleoperation. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, Seoul, Korea, May 2001.

[99] Marc-Walter Ueberle. *Design, Control, and Evaluation of a Family of Kinesthetic Haptic Interfaces*. PhD thesis, Technische Universität München, 2006.

[100] Y. Uno, M. Kawato, and R. Suzuki. Formation and control of optimal trajectory in human multijoint arm movement. *Biological Cybernetics*, 61(2):89–101, June 1989.

[101] Ulrich Unterhinninghofen, Franziska K. B. Freyberger, and Martin Buss. Study on computer assistance for telepresent reaching movements. In Manuel Ferre, editor, *Haptics: Perception, Devices and Scenarios*, volume 5024 of *LNCS*, pages 745–754. Springer, 2008.

[102] Arjan van der Schaft. Port-hamiltonian systems: an introductory survey. In *Proceedings of the International Congress of Mathematicians*, Madrid, Spain, 2006.

[103] Carolina Weber, Verena Nitsch, Ulrich Unterhinninghofen, Berthold Färber, and Martin Buss. Position and force augmentation in a telepresence system and their effects on perceived realism. In *Proceedings of the World Haptics 2009 - Third Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 226–231, 2009.

[104] B. G. Witmer and M. J. Singer. Measuring presence in virtual environments: A presence questionnaire. *Presence: Teleoperators and Virtual Environments*, 7(3):225–240, 1998.

[105] Tomonori Yamamoto, Michael Bernhardt, Angelika Peer, Martin Buss, and Allison M. Okamura. Techniques for environment parameter estimation during telemanipulation. In *Proceedings of the 2nd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics*, October 2008.

[106] Wentao Yu, Redwan Alqasemi, Rajiv Dubey, and Norali Pernalete. Telemanipulation assistance based on motion intention recognition. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain, April 2005.

[107] Michael F. Zäh, Stella Clarke, Bernd Petzold, and Johannes Schilp. Achieving flexible micro-assembly systems through telepresence. In *Proceedings of the Mechatronics & Robotics Conference*, pages 1473–1477, Aachen, September 2004.

[108] Michael F. Zäh and Andrea Reiter. Precise positioning in a telepresent microassembly system. In *Proceedings of the IEEE International Workshop on Haptic Audio Visual Environments and their Applications*, pages 44–47, Ottawa, Canada, November 2006.