Bipedal Locomotion Primitive Learning, Control and Prediction from Human Data

Kai Hu, Dongheui Lee

Department of Electrical Engineering and Information Technology, Technical University of Munich, D-80290 Munich, Germany
(e-mail: kai.hu@mytum.de, dhlee@tum.de)

Abstract: At the current stage bipedal robot locomotion is quite different from human walking. Imitation learning framework from human demonstrations is an efficient approach to lead towards human-like behaviors. This paper addresses a framework for real-time whole-body human motion imitation by a humanoid robot. The framework is a structured mixture of whole body motion control, learning and prediction. Human movements are mapped to robot's kinematics in combination with a balancing algorithm in order to ensure the dynamic constraints during different stance phases. Once locomotion primitives are learned from human demonstrations using hidden Markov models, the robot can recognize human's current locomotion state and predict future trajectories using Gaussian regression. The proposed concepts are implemented and evaluated with a small humanoid robot NAO.

Keywords: Motion Imitation, Bipedal Walking, Robot Learning, Control, Prediction

1. INTRODUCTION

The study of human movements often provide great inspiration to scientists and engineers in robotics, biomechanical design, computer animation, and other related fields (Lee et al., 2002; Maufroy et al., 2011). The paradigm of using human motion capture data is known as an efficient method which can avoid tedious hand programming process for humanoid robot motion planning and achieve expressive human-like high dimensional motions. However mapping human motion to robots is not always straightforward due to the kinematic and dynamic inconsistencies between human and humanoid robots. Human reference motion could be infeasible for robots because of physical constraints like joint limits and velocity limits. Further bipedal locomotion is intrinsically instable. Simple kinematic mapping of limb movements will not ensure dynamic stability especially when human reference motion includes stance [1] changes.

The upper-body motion imitation has been successfully realized in many previous works. The joint angles of human skeleton model was adapted to the robot according to joint and velocity limits by local scaling (Pollard et al., 2002). In order to measure quantitatively the human-likeness of the arm motions, Kim et al. (2006) defined the elbow elevation angle and solved the inverse kinematics analytically according to the geometric relations. In (Suleiman et al., 2008) the upper-body motion imitation was formulated as a general optimization problem under the constraints of the physical limits. In (Dariush et al., 2008) a Cartesian tracking control framework was presented where the upper-body motion is imitated by tracking selected motion descriptors in task space. Ott et al. (2009) proposed a Cartesian control method in which a set of control points on the humanoid robot are connected to the corresponding measured points on a human via virtual springs and the spring forces drive a simulation of the robot dynamics. Their approach is applicable for both a torque- and a position-controlled robot and also suitable for tracking large amount of Cartesian markers without specific prioritization. In these above works, however, the lower-body motion is ignored or simple balance control is employed.

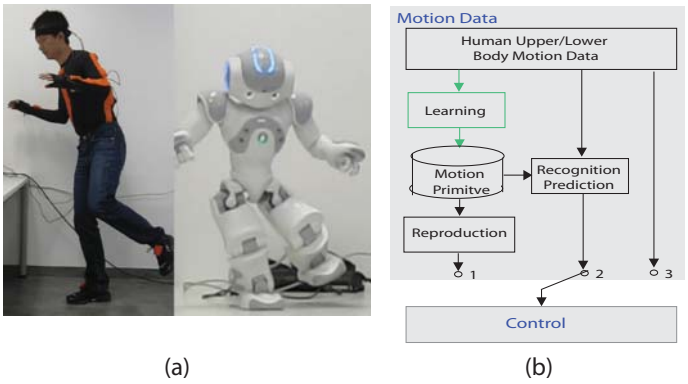

Fig. 1. (a)Whole body motion imitation including stance changes;(b)System overview of learning, control and prediction.

The HRP2 humanoid robot performed Japanese folk dance successfully in (Nakaoka et al., 2003, 2005). Their method requires offline process to define the basic task primitives and their parameter sets, which are necessary for the lower-body motions. Kim et al. (2009) suggested to use a simplified human model in order to obtain complex human ZMP trajectories. However, this approach requires to specify the ZMP human model for different human

[1] Throughout the paper a 'stance' denotes a foot stance, for example, left foot support (LS), right foot support (RS), single foot support (SS), and double feet support (DS).

performers and to use a force plate. Yamane and Hodgins (2010) proposed a motion synthesis method which modifies the reference CoM trajectory according to the predicted ZMP in order to maintain balance when the contact state changes. Combined with their tracking and balancing controller online whole-body motion imitation with stepping is achieved in simulation. However a constant time delay exists due to the ZMP prediction and the implementation and evaluation on a real robot was missing.

Above mentioned works are mainly focused on mimicking human motions, considering kinematics only or together with dynamic stability. Robot learning from observation of human's behavior has become an active research area in robotics. This approach is promising for movements of high degree-of-freedom humanoid robot body. Further, this approach allows recognition and reproduction beyond the simple mimicking. So far imitation learning of arm and upper body motion has been extensively studied (Billard et al., 2008). Recently it has been extended to manipulation task involving physical contact with environment (Koropouli et al., 2011; Kormushev et al., 2011; Schmidts et al., 2011). On the other hand, locomotion skill learning from human data has not been investigated deeply due to its dynamically instable nature.

In this paper we propose a framework of whole body motion control, learning and prediction for a humanoid robot from human motion data, as shown in Fig. 1(b). This framework consists of two main blocks: *motion data* and *control*. The *motion data* block handles three different approaches: learning from human demonstrations and reproduction (switch 1), recognizing human's state and predicting the future human behaviors (switch 2), and directly passing real-time human data to a controller (switch 3). The use of human data provides efficient programming and human-like high-dimensional motions of a humanoid robot. The learning from human data and reproduction approach extends its advantages to the compact representation and generalization in new situations (Billard et al., 2008). The prediction based approach brings further benefits for advanced robot behaviors, such as proactive behaviors (Medina et al., 2011). The *control* block deals with kinematic mapping and dynamic mapping (Sec. 3.5) from the (measured or learned or predicted) human motion to the robot. It preserves the human-motion likeliness and dynamic stability respectively. The detail of this block is shown in Fig. 2. In this work, based on learning, prediction and control, a new method for a real-time synchronized whole-body human motion imitation including stance changes is realized.

2. PROGRAMMING BY DEMONSTRATIONS

2.1 Motion primitive learning

Periodic Hidden Markov Models (HMMs) (Rabiner, 1989) are used for the representation of the whole body motion primitives. HMMs provide a probabilistic representation of spatiotemporal variabilities of training data with good recognition and reproduction performance. Each primitive is represented as an HMM λ with N states. HMM parameters consist of the initial state probability, the state transition probability, and observation probability distribution: π_i denotes the probability for the initial state to be the ith state, a_{ij} denotes the probability to transit from ith state to the jth state, and $b_i(\boldsymbol{o})$ is the probability density function for the output of a vector $\boldsymbol{o}$ at the ith state. Herein, a Gaussian distribution is used: $b_i(\boldsymbol{o}) = \mathcal{N}(\boldsymbol{o}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$. The mean vector $\boldsymbol{\mu}_i$ and the covariance matrix $\boldsymbol{\Sigma}_i$ for the Gaussian at the ith state are designed as

$$\boldsymbol{\mu}_i = \begin{bmatrix} {}^t\mu_i \\ {}^s\boldsymbol{\mu}_i \end{bmatrix} \quad , \qquad \boldsymbol{\Sigma}_i = \begin{bmatrix} {}^t\Sigma_i & {}^{ts}\boldsymbol{\Sigma}_i \\ {}^{st}\boldsymbol{\Sigma}_i & {}^s\boldsymbol{\Sigma}_i \end{bmatrix}$$

so that explicit temporal information is contained as well as spatial information. The symbol ${}^{ts}\boldsymbol{\Sigma}_i$ is the covariance vector between temporal and spatial data of the Gaussian at state i. The superscript s and t denotes spatial and temporal data respectively. It is shown that a continuous trajectory can be reproduced by learning the correlation between temporal and spatial data in spite of the discrete nature of states in the HMM in Lee and Ott (2010).

From multiple demonstrations of human locomotions $\mathcal{O}$, the parameters of the above mentioned HMM λ are trained in order to maximize the likelihood $P(\mathcal{O}|\lambda)$. First, a spatial data sequence ${}^s\mathcal{O} = \{{}^s\boldsymbol{o}(t)\}$ is used to train the HMM parameters $({}^s\boldsymbol{\mu}_i, {}^s\boldsymbol{\Sigma}_i)$ via the Baum Welch algorithm (Rabiner, 1989). The next step is the learning the correlation between temporal and spatial variables. Given the spatial data, the probability of being in state i at time t for the e-th demonstration sequence is calculated and denoted as $\gamma_i^e(t)$. Then, for each state i the mean ${}^t\mu_i$ and time variance ${}^t\Sigma_i$ of the temporal data are computed as follows:

$${}^t\mu_i = \frac{\sum_{e=1}^{E}\sum_{t=1}^{T^e}\gamma_i^e(t)\,t}{\sum_{e=1}^{E}\sum_{t=1}^{T^e}\gamma_i^e(t)} \tag{1}$$

$${}^t\Sigma_i = \frac{\sum_{e=1}^{E}\sum_{t=1}^{T^e}\gamma_i^e(t)\,(t-{}^t\mu_i^e)^2}{\sum_{e=1}^{E}\sum_{t=1}^{T^e}\gamma_i^e(t)} \tag{2}$$

where T^e denotes the total length of the e-th demonstration and E is the number of demonstrations in the training dataset. The covariance between spatial and temporal data is given by

$${}^{ts}\boldsymbol{\Sigma}_i = \frac{\sum_{e=1}^{E}\sum_{t=1}^{T^e}\gamma_i^e(t)\,(t-{}^t\mu_i^e)\,({}^s\boldsymbol{o}^e(t)-{}^s\boldsymbol{\mu}_i)}{\sum_{e=1}^{E}\sum_{t=1}^{T^e}\gamma_i^e(t)} \quad . \tag{3}$$

The spatial data ${}^s\boldsymbol{o}(t)$ presents whole body motions which can be in joint coordinates and/or Cartesian coordinates (Lee and Ott, 2010). In this work foot stance information (DS: double support, LS: left foot support or RS: right foot support) is added for the compatibility with most locomotion controllers. Note that some of spatial elements are continuous (joint angles and/or Cartesian marker data) and the newly added spatial element is in discrete space (DS, LS, RS). Thus, the used HMM is designed as the combination of continuous HMM and discrete HMM, simply by defining the element ${}^s\mu$ for stance as an integer value.

2.2 Reproduction

In order to achieve the generation of a continuous trajectory, recently the combination of HMMs and Gaussian regression techniques have been proposed by Lee and Ott (2011) and Calinon et al. (2010). In this work, the learning

approach from (Lee and Ott, 2011) has been modified since all the spatial elements are not continuous in their nature, e.g., foot stance.

In the generation procedure, the learned temporal information is taken into account, defining a Gaussian for each state i, centered in ${}^t\mu_i$ with variance ${}^t\Sigma_i$. The responsibility over time ${}^t\gamma_i(t)$ is defined as

$$ {}^t\gamma_i(t) = \frac{\mathcal{N}(t|{}^t\mu_i, {}^t\Sigma_i)}{\sum_{j=1}^{N}\mathcal{N}(t|{}^t\mu_j, {}^t\Sigma_j)} \quad . \tag{4} $$

Weighted by the responsibilities over time, the spatial data is generated by the Gaussian mixture regression algorithm as

$$ {}^s\hat{\boldsymbol{o}}(t) = \sum_{i=1}^{N} {}^t\gamma_i(t)\left({}^s\boldsymbol{\mu}_i + \frac{{}^{ts}\boldsymbol{\Sigma}_i}{{}^t\Sigma_i}\left(t - {}^t\mu_i\right)\right) \quad . \tag{5} $$

The conditional expectation of the variance of the spatial data is

$$ {}^s\hat{\boldsymbol{\Sigma}}(t) = \sum_{i=1}^{N} {}^t\gamma_i(t)\left({}^s\boldsymbol{\Sigma}_i - \frac{{}^{ts}\boldsymbol{\Sigma}_i {}^{st}\boldsymbol{\Sigma}_i}{{}^t\Sigma_i}\right) \quad . \tag{6} $$

In this way, a smooth motion sequence can be decoded from a learned HMM except for the discrete stance trajectory. For the stance, ${}^s\hat{\boldsymbol{o}}(t)$ is set to the mean value of the state which has the highest ${}^t\gamma_i$ in order to ensure the discrete nature.

2.3 Prediction

Prediction of the future of an observed motion can be realized by extending the generation method in Sec. 2.2. Given the observed time series data so far, the best matching HMM is found by selecting the HMM with the highest likelihood in the learned primitive database. With the chosen best HMM and observed time-series data, the best current state s at the current time t is estimated by the Viterbi algorithm. Herein, we introduce a generalized time index ${}^t\hat{o}(t)$ in addition to a real time index ${}^to(t) = t$ by comparing the duration of the state s in the incoming observations and the generalized output of the model. A generalized time index of the real time index ${}^to(t+L)$, which is the time index of L steps in the future, is defined by

$$ {}^t\hat{o}(t+L) = \left(\frac{L+n_s}{v} + t_s\right) \quad , \tag{7} $$

where t_s is the starting time index of the state s on the generalized state sequence, which is ${}^t\mu_s - \frac{-3+\sqrt{1+48\,{}^t\Sigma_s}}{4}$. The term v is the velocity factor, defined as the ratio between observed and learned velocity. When the velocity is not included as the spatial elements of the HMM, the velocity can be estimated indirectly from the expected durations and the measured durations in the previous states during the observation. The term n_s is the time length stayed at the current state s on the incoming observations state sequence. Once the generalized time index is estimated, the spatial data at the L step in the future can be predicted by

$$ {}^s\hat{\boldsymbol{o}}(t+L) = \sum_{i=1}^{N} {}^t\gamma_i\left({}^t\hat{o}(t+L)\right)\left({}^s\boldsymbol{\mu}_i + \frac{{}^{ts}\boldsymbol{\Sigma}_i}{{}^t\Sigma_i}\left({}^t\hat{o}(t+L) - {}^t\mu_i\right)\right) $$

for the continuous data, and by

$$ {}^s\hat{\boldsymbol{o}}(t+L) = {}^s\boldsymbol{\mu}_k \quad , $$

where k is the index of the state with the highest ${}^t\gamma_i\left({}^t\hat{o}(t+L)\right)$, for the discrete data .

3. WHOLE BODY MOTION CONTROL

3.1 Kinematic Mapping

The kinematic mapping procedure converts whole-body human motion to the robot by tracking Cartesian motion descriptors of each body limb. The goal is to preserve motion likeness as well as enforce kinematic constraints such as singularity avoidance, joint limits and joint velocity limits.

Motion descriptors should be chosen for each limb such that they fully characterize the limb motion. Position and orientation of the end-effectors are good candidates. Because of the geometric differences between human and robot model, the original human motion data must be normalized to suit the robot's kinematic size based on a physiological rescaling of the translational part along the kinematic chain. Additional rotational adjustment should be included if the local coordinate systems of human and humanoid robot body parts are defined differently.

Assuming that the motion capture system provides real-time data of the relative rotation and translation between human body segments connected by rotational joints, we can transform a human marker configuration $\boldsymbol{x}_k = [\boldsymbol{R}_k^h \;\; \boldsymbol{p}_k^h;\;\; \mathbf{0}\;\; 1] \in \mathbb{R}^{4\times4}$ attached to the k-th segment into the desired marker pose $\boldsymbol{x}_{d,k} \in \mathbb{R}^{4\times4}$ for the humanoid by

$$ \boldsymbol{x}_{d,k} = \boldsymbol{T}_0\prod_{i=0}^{k} {}^i\boldsymbol{T}_{i+1} = \boldsymbol{T}_0\prod_{i=0}^{k-1}\begin{bmatrix} {}^i\boldsymbol{R}_{i+1}^h\boldsymbol{R}_{i+1}^{rh} & {}^i\boldsymbol{P}_{i+1}^h\dfrac{L_{r,i}}{L_{h,i}} \\ \mathbf{0}^T & 1 \end{bmatrix} \quad . \tag{8} $$

$\boldsymbol{T}_0$ is the homogeneous transformation between the local frame of the first joint of this limb and the base frame. ${}^i\boldsymbol{T}_{i+1}$ is the normalized relative transformation matrix between the ith and $i+1$th joints of this limb. The human relative rotation matrix ${}^i\boldsymbol{R}_{i+1}^h$ does not need to be rescaled, but the difference between the human and robot's local frames should be adjusted by $\boldsymbol{R}_{i+1}^{rh}$, which is the initial relative rotational matrix of the $i+1$th joint between the human and robot kinematic model. The human relative translational part ${}^i\boldsymbol{P}_{i+1}^h$ should be rescaled using the ith human and robot segment lengths, $L_{h,i}$ and $L_{r,i}$.

We use closed loop inverse kinematics (CLIK) (Siciliano et al., 2009) to track the normalized motion descriptors. The CLIK algorithm is based on the inverse instantaneous kinematics which relates the task space and joint space in velocity level. The regularized right pseudo-inverse of Jacobian is used to avoid singularities (Wampler, 1986) and joint limits (Chan and Dubey, 1995). Although the whole-body limb motions are tracked using this method, the lower-body inverse kinematics should also include balance control part from dynamic mapping.

3.2 Stance Estimation

As the bipedal balance condition the robot's ZMP should stay strictly inside the support polygon. For the stable

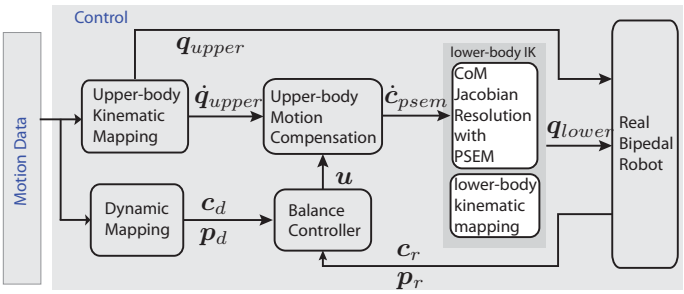

Fig. 2. whole-body motion control and balance control

dynamic mapping, the reference ZMP and CoM trajectories are planned according to human stances. For the lower-body controller we assume the ground to be level and flat. Two conditions are used to estimate the human stance: if the height difference between two feet is larger than a certain threshold, the higher foot is considered as not-supporting.; the foot velocity should below certain threshold for a certain period in order to be recognized as supporting.

The two conditions can cover wider spectrum than walking motions, for example when a human stands still on one foot and one of the motionless feet is not supporting, and when one foot sliding motion during double support, in which the moving foot is actually supporting.

3.3 Inverted Pendulum Model

The linear inverted pendulum model (LIPM) (Kajita et al., 2001) is taken as the simplified control model of the humanoid robot

$$\boldsymbol{p} = \boldsymbol{c} - \frac{c_z}{g}\ddot{\boldsymbol{c}} \quad , \tag{9}$$

in which $\boldsymbol{p} = [p_x \; p_y]^T$ [2] is the ZMP position vector, $\boldsymbol{c} = [c_x \; c_y]^T$ the CoM position vector, c_z the CoM height and g the gravity constant. This model relates the robot body motion to the robot's ZMP and avoids complex multi-link dynamics computation. Despite its simplicity, this model has successfully been applied to state of the art walking control approaches (Kajita et al., 2003; Choi et al., 2007).

3.4 ZMP and CoM Planning

The reference ZMP and CoM trajectory should be planned according to the estimated human stance, current robot feet positions and human pelvis motions. The ZMP must stay in the support polygon and the CoM should be a smooth trajectory which is used to control the robot body motion. The time delay due to solving the dynamic equation (9) from ZMP to CoM is undesirable. Therefore we distinguish two different situations, namely *stance static phase* and *stance change phase*, and suggest different planning methods respectively, aiming at reducing the time delay in the system.

Stance static phase denotes the duration that human stance does not change. During the single support (SS) the support polygon is relatively small, thus we set the position of desired ZMP to be constant in the center of the support polygon. The resulting CoM is identical to the ZMP. During the double support (DS) the support polygon is large and the planar movements of the pelvis are imitated for better human motion imitation. The pelvis motion is approximated by the motion of the CoM. In order to reserve more stability margin, the pelvis motion is projected onto the segment connecting the foot centers. If the projected CoM is outside of the segment, the closest foot center is selected as a desired CoM. Assuming slow motions, we set the reference ZMP trajectory to be identical with the reference CoM trajectory for simplification.

[2] Throughout this paper, the subscript x denotes the direction in saggital plane and y is in lateral plane.

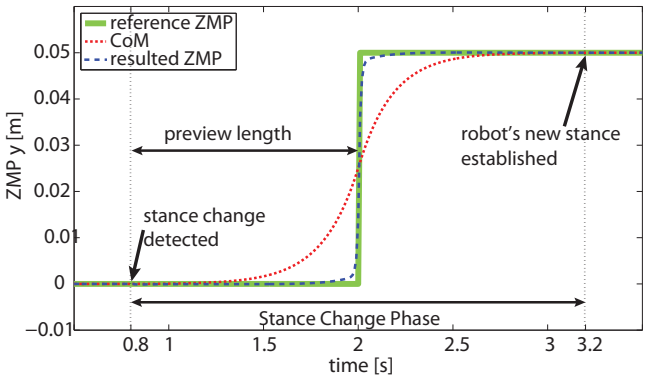

Fig. 3. Reference ZMP and CoM trajectory during the stance change phase from DS to SS. The reference ZMP is delayed for the time of preview length after the detection of human stance change. Optimal CoM trajectory is generated through preview controller.

Stance change phase denotes the time interval from the detection of the human stance change to the establishment of robot's new single support stance. In case of change from from SS to DS, the foot motion can be directly imitated since the new feet contact will extend the support polygon. On the other hand, when the stance changes from DS to SS (e.g. lifting one of the feet), the robot cannot follow this motion immediately, since the reduction of support polygon will make the balance constraint violated. In such a case, we plan the ZMP to move to the new desired position as a step change and calculate CoM trajectory by using the preview controller with an appropriate preview length (Kajita et al., 2003), see Fig. 3. When the stance change is detected, robot's ZMP step change must be delayed for a preview length (green solid line) due to the ZMP preview. The robot foot motion is delayed as well in order to keep the support polygon unchanged. Corresponding CoM trajectory (red dotted line) is generated using the preview controller and executed by the robot. After the time interval of a preview length, the resulted ZMP (blue dashed line) reaches the new desired position, which is located in the new future support polygon. Thus the use of this algorithm alone produces a constant delay robot of the preview length during walking, more specifically speaking at the transition from DS to SS.

Although the inverted pendulum model assumes a fixed CoM height, motions with varying CoM height (e.g., squatting) are handled in stance static phase, by resetting the model's CoM height equal to the robot's CoM height at the beginning of this stance change.

3.5 Whole-body motion coordination and balance control

The CoM Jacobian with partially specified embedded motion (PSEM) method (Choi et al., 2007) allows to specify a desired CoM trajectory by lower-body joints while upper-body motion and foot motion can be taken into account. This method is adopted to combine the kinematic and dynamic mapping results and achieve whole-body motion synthesis (see Fig. 2).

For the balancing against disturbances, the CoM-ZMP feedback controller

$$\boldsymbol{u} = \dot{\boldsymbol{c}}_d - \boldsymbol{K}_p(\boldsymbol{p}_d - \boldsymbol{p}_r) + \boldsymbol{K}_c(\boldsymbol{c}_d - \boldsymbol{c}_r) \qquad (10)$$

is used, where $\dot{\boldsymbol{c}}_d$ is the reference CoM velocity vector, $\boldsymbol{p}_d$ and $\boldsymbol{c}_d$ are desired ZMP and CoM positions. $\boldsymbol{p}_r$ and $\boldsymbol{c}_r$ are the real ZMP and CoM positions calculated from the force sensors on the soles and joint encoders. The error gain matrices $\boldsymbol{K}_p$ and $\boldsymbol{K}_c$ should obey the stability condition given in (Choi et al., 2007).

In particular the torso orientation is important for the balance of our top-weight bipedal robot. The torso orientation calculated from forward kinematics may deviate from the real value and cause serious problem of balance control. Therefore we use the sensor value from the inertial unit mounted in the torso for the feedback of CLIK algorithm.

4. EXPERIMENTAL RESULTS

4.1 Experimental environment setting

The proposed framework integrating learning, prediction and humanoid whole body control has been implemented on a 58 cm high and 4.8 kg weight humanoid NAO, shown in Fig. 1. The robot has 23 DoFs, force sensors on the soles to measure ZMP and inertial unit in the torso to measure the real torso orientation. The algorithm runs on the robot's embedded computer and the control loop frequency is 100 Hz. Xsens MVN motion capture suit is used to capture human motion data in real-time. The suit consists of 16 IMUs (inertia measurement units) and provides the 6D positions and orientations of 23 human body parts.

The experiments were conducted with a human performer wearing this motion capture suit and the robot imitating the human motion in real-time. The corresponding points of the human and the robot are shown in Fig. 4. The human demonstrator performed a motion sequence consisting of upper- and lower-body motions including squat, bow, torso roll, stance changes to test our online whole-body motion imitation algorithm. Even during the single support phase, various motions like single foot squat and kick motions are performed. Fig. 5 shows snapshots of real-time human whole body motion imitation by NAO during the experiment.

4.2 Learning and prediction

From the human walking data, captured with the sampling rate of 60Hz, has been divided into a training dataset and a test dataset. The training dataset consists of nine demonstrations, 48.15s long in total. The test dataset consists of five demonstrations, in total 45.725s long. The

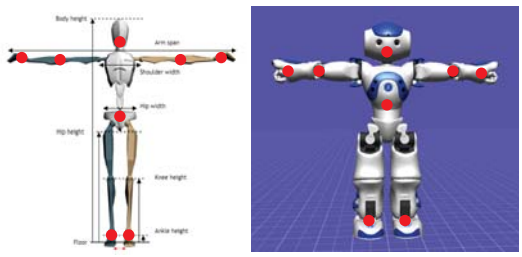

Fig. 4. Selected markers of a human and NAO

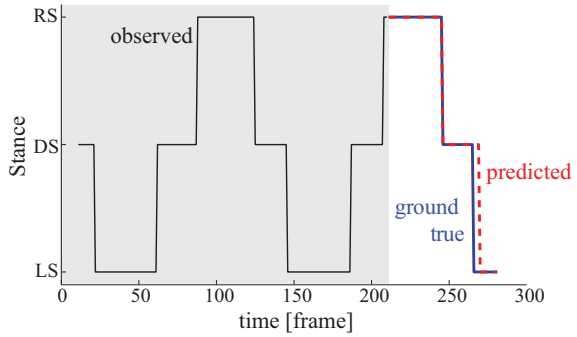

Fig. 6. An example of prediction of human stance trajectory. (black) observed one, (red) predicted result, (blue) ground true

dataset contains walking motions with different speeds. The walking motions used in the experiment are rather quasi-static walking motion; the DS phase exists always between RS and LS. In the test data of 45.725s, there were 20 stance changes from DS to SS (either RS or LS).

A walking motion primitive is learned from nine demonstrations in the training dataset using the method discussed in Section 2. Given observed current human motion data, the current human state is recognized and future human motions (foot position trajectories as well as stance information) are predicted online by using the learned locomotion primitives. An example is shown in Fig. 6, where the observed, predicted and ground true of human foot stance are depicted with the black, red dashed, and blue solid line.

Table 1. Prediction error and delay

	prediction error	delay
average	-0.0451s	0.0725s
standard deviation	0.1691	0.1079
max	0.4s	0.3667s
min	-0.4333s	-

We evaluated the (signed valued) prediction errors of stance change timing, which is calculated as the predicted timing minus the ground true timing of corresponding stance change. During online prediction, each time frame, the algorithm performed prediction of future 1.2s. The average prediction error was -0.0451s with the standard deviation of 0.1691. The max and min error was 0.4s and -0.4333s.

4.3 Combination of prediction and ZMP/CoM planning

In order to evaluate the delay of real-time human walking motion imitation, we conducted two approaches: walking

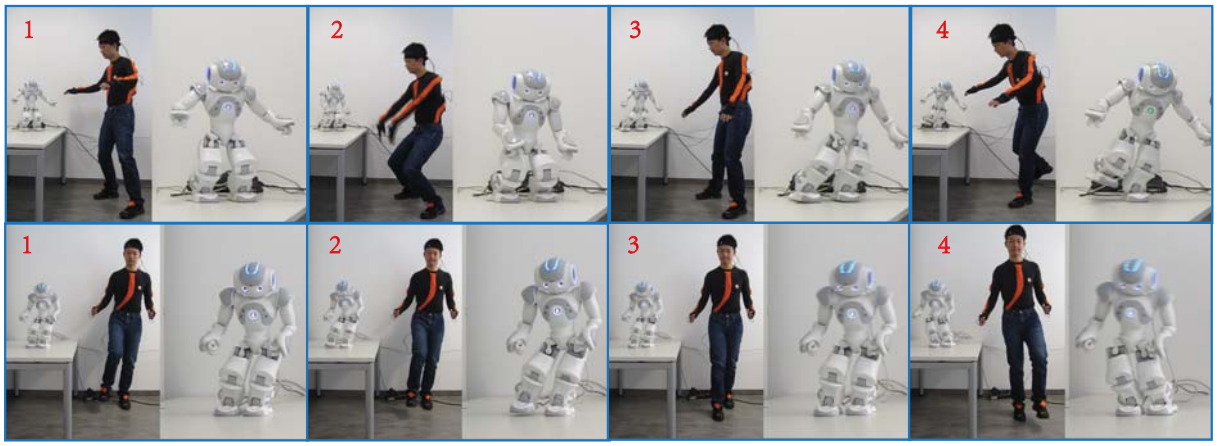

Fig. 5. Snapshots of experiments of real time whole-body motion imitation including walking imitation (bottom). In case of stance change the lift foot motion is temporarily delayed when the prediction was not used.

imitation without the use of prediction results (switch 3 in Fig. 1) and walking imitation by using the motion prediction (switch 2 in Fig. 1). In the first approach (switch 3 in Fig. 1) we plan the ZMP and generate the CoM trajectory using the preview controller by delaying the motion commands for a time interval of preview length which is set to 1.2s according to the criterion in (Kajita et al., 2003). This approach resulted in delayed behaviors but the human walking is imitated faithfully.

In the case of prediction, the prediction results are used by the preview controller to generate walking pattern. In order to perform real-time human walking motion imitation without delay, it is required to predict the stance transition from DS to SS minimum the preview length ahead. The measured delay for 20 stance changes in the test dataset were 0.0725s on average with the standard deviation of 0.1079. Ten out of twenty had no delay. The largest delay was 0.3667s. This result shows the delay has been greatly reduced from 1.2s to 0.0725s on average.

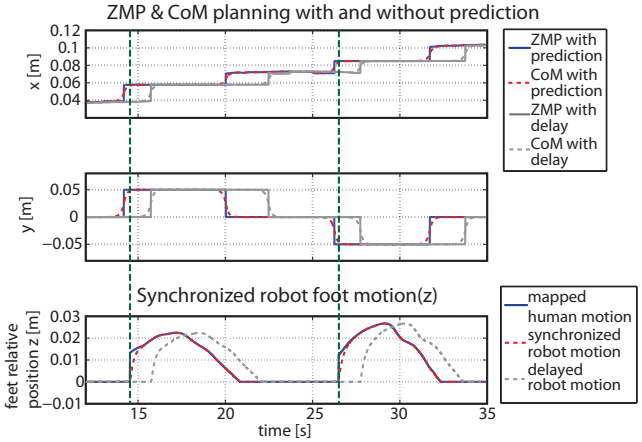

Fig. 7. Comparison with and without prediction. (top, middle) the ZMP and CoM planning results . (bottom) the foot motion of the human and the robot. The green lines indicate the time of human lifting the foot.

Both approaches (with and without prediction) have been implemented on a real robot. The top and middle graphs in Fig. 7 show the ZMP and CoM planning results using the

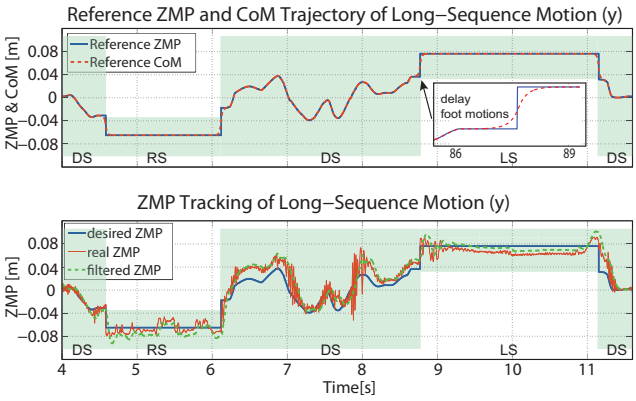

Fig. 8. Dynamic mapping and control. (top) ZMP and CoM planning results from the dynamic mapping in lateral plane. (bottom) the tracking of the desired ZMP trajectory.

predicted stance information (blue solid and red dotted lines) as well as without the prediction (gray solid and dotted lines). The bottom graph in Fig. 7 illustrates the targetting human foot trajectory and the real robot foot trajectory with and without prediction cases. The human stance is estimated by the explanation in Sec. 3.2 and the green vertical lines depicts the time lines when the human lifts his foot. At the moment of human's foot lifting, in the case of prediction, the robot already moved its ZMP and CoM into the (predicted) new supporting area. Although the stance predictions are not very precise (top and middle graph), it is enough for preparing the ZMP shift before the contact state changes and it leads to stable synchronized stepping imitation (bottom graph).

4.4 ZMP and CoM control

The upper graph of Fig. 8 illustrates the planning results of the reference ZMP and CoM trajectories by the dynamic mapping procedure in lateral direction. The green area marks the corresponding support polygon of each stances. During the double support phases, the CoM is approximated by the projected pelvis motion and the reference ZMP and CoM are set to be identical. During the single

support phases the ZMP and CoM are in the middle of the supporting polygon.

The lower graph of Fig. 8 shows the corresponding ZMP tracking results. The ZMP measured from the force sensors is very noisy and leads to small oscillations of robot motions. A low-pass filter is applied to smooth the ZMP trajectory (green dotted line). Although it brings delay into the sensor feedback, the oscillations are well suppressed. The balance controller successfully maintains the ZMP inside the supporting polygon even under critical situations, such as single foot squat.

5. CONCLUSION

In this paper, we proposed a new integrated framework for whole body motion learning, prediction and control for a humanoid robot from captured human motions. Motion primitives are learned from human demonstrations using hidden Markov models, and the robot can recognize human's current locomotion state and predict future trajectories using Gaussian regression. The combination of prediction of future human movements with whole body balancing control leads to synchronized human stepping motion imitation by a humanoid robot. All proposed concepts are implemented on a real humanoid robot. The experimental results showed good performance of faithful whole body movement imitation and balancing control using force-sensing resistors with a rather slow sampling rate (100Hz). Furthermore, the proposed learning and prediction method was evaluated with a new test dataset (45.725s long). The average prediction error of stance transition timing was 0.0451s, and the delay of stance changes from 1.2s to 0.0725s on average.

ACKNOWLEDGEMENTS

This research is partly supported by the DFG excellence initiative research cluster "Cognition for Technical Systems CoTeSys" and Institute for Advanced Study.

REFERENCES

Billard, A., Calinon, S., Dillmann, R., and Schaal, S. (2008). *Robot Programming by Demonstration*. Siciliano, B. and Khatib, O. (eds.). Handbook of Robotics, Springer.

Calinon, S., D'halluin, F., Sauser, E., Caldwell, D., and Billard, A.G. (2010). Learning and reproduction of gestures by imitation: An approach based on hidden markov model and gaussian mixture regression. *IEEE Robotics and Automation Magazine*, 17(2), 44–54.

Chan, T. and Dubey, R. (1995). A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators. *IEEE Transactions on Robotics and Automation*, 11(2), 286–292.

Choi, Y., Kim, D., Oh, Y., and You, B. (2007). Posture/walking control for humanoid robot based on kinematic resolution of com jacobian with embedded motion. *IEEE Trans. on Robotics*, 23(6), 1285–1293.

Dariush, B., Gienger, M., Jian, B., Goerick, C., and Fujimura, K. (2008). Whole body humanoid control from human motion descriptors. In *IEEE Int. Conf. on Robotics and Automation*, 2677 –2684.

Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., and Hirukawa, H. (2003). Biped walking pattern generation by using preview control of zero-moment point. In *IEEE Int. Conf. on Robotics and Automation*, 1620–1626.

Kajita, S., Kanehiro, F., Kaneko, K., Yokoi, K., and Hirukawa, H. (2001). The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, volume 1, 239–246.

Kim, S., Kim, C., and Park, J. (2006). Human-like arm motion generation for humanoid robots using motion capture database. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 3486–3491.

Kim, S., Kim, C., You, B., and Oh, S. (2009). Stable whole-body motion generation for humanoid robots to imitate human motions. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2518–2524.

Kormushev, P., Calinon., S., and Caldwell, D. (2011). Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input. *Advanced Robotics*, 23, 581–603.

Koropouli, V., Lee, D., and Hirche, S. (2011). Learning interaction force skills by demonstration. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 344–349.

Lee, D. and Ott, C. (2010). Incremental motion primitive learning by physical coaching using impedance control. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 4133–4140.

Lee, D. and Ott, C. (2011). Incremental kinesthetic teaching of motion primitives using the motion refinement tube. *Autonomous Robots*, 31(2), 115–131.

Lee, J., Chai, J., Reitsma, P., Reitsma, P., Hodgins, J., and Pollard, N. (2002). Interactive control of avatars animated with human motion data. In *ACM Transactions on Graphics*, volume 21, 491–500.

Maufroy, C., Maus, H.M., Radkhah, K., Scholz, D., von Stryk, O., and Seyfarth, A. (2011). Dynamic leg function of the biobiped humanoid robot. In *Proc. 5th Intl. Symposium on Adaptive Motion of Animals and Machines (AMAM)*.

Medina, J., Lawitzky, M., Moertl, A., Lee, D., and Hirche, S. (2011). An experience-driven robotic assistant acquiring human knowledge to improve haptic cooperation. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2416–2422.

Nakaoka, S., Nakazawa, A., Kanehiro, F., Kaneko, K., Morisawa, M., and Ikeuchi, K. (2005). Task model of lower body motion for a biped humanoid robot to imitate human dances. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 3157–3162.

Nakaoka, S., Nakazawa, A., Yokoi, K., Hirukawa, H., and Ikeuchi, K. (2003). Generating whole body motions for a biped humanoid robot from captured human dances. In *IEEE Int. Conf. on Robotics and Automation*, volume 3, 3905–3910.

Ott, C., Lee, D., and Nakamura, Y. (2009). Motion capture based human motion recognition and imitation by direct marker control. In *IEEE-RAS Int. Conf. on Humanoid Robots*, 399–405.

Pollard, N., Hodgins, J., Riley, M., and Atkeson, C. (2002). Adapting human motion for the control of a humanoid robot. In *IEEE Int. Conf. on Robotics and Automation*, volume 2, 1390–1397.

Rabiner, L.R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2), 257–286.

Schmidts, A., Lee, D., and Peer, A. (2011). Imitation learning of human grasping skills from motion and force data. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1002– 1007.

Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2009). *Robotics: Modelling, Planning and Control*. Springer Verlag.

Suleiman, W., Yoshida, E., Kanehiro, F., Laumond, J.P., and Monin, A. (2008). On human motion imitation by humanoid robot. In *IEEE Int. Conf. on Robotics and Automation*, 2697 –2704.

Wampler, C. (1986). Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Transactions on Systems, Man and Cybernetics*, 16(1), 93 –101.

Yamane, K. and Hodgins, J. (2010). Control-aware mapping of human motion data with stepping for humanoid robots. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 726 – 733.