REAL-TIME FULL BODY MOTION IMITATION ON THE COMAN HUMANOID ROBOT

Andrej Gams[1,2], Jesse van den Kieboom[2], Florin Dzeladini[2],
Aleš Ude[1] and Auke Jan Ijspeert[2]
[1]Dept. of Automation, Biocybernetics and Robotics
Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
[2]Biorobotics Laboratory
École Polytechnique Fédérale de Lausanne
Station 14, CH-1015 Lausanne, Switzerland
{andrej.gams, ales.ude}@ijs.si
{jesse.vandenkieboom, florin.dzeladini, auke.ijspeert}@epfl.ch

ABSTRACT- On-line full body imitation with a humanoid robot standing on its own two feet requires simultaneously maintaining the balance and imitating the motion of the demonstrator. In this paper we present a method that allows real-time motion imitation while maintaining stability, based on prioritized task control. We also describe a method of modified prioritized kinematic control that constrains the imitated motion to preserve stability only when the robot would tip over, but does not alter the motions otherwise. To cope with the passive compliance of the robot, we show how to model the estimation of the center-of-mass of the robot using support vector machines. In the paper we give detailed description of all steps of the algorithm, essentially providing a tutorial on the implementation of kinematic stability control. We present the results on a child sized humanoid robot called Compliant Humanoid Platform or COMAN. Our implementation shows reactive and stable on-line motion imitation of the humanoid robot.

Keywords: kinematics, motion imitation, stability, center-of-mass, SVM

1 Introduction

The transfer of human motion to humanoid robots can be accomplished in many manners, one of them being motion capture ([1,2]). Different kinematic and dynamic properties of humans and robotic mechanisms do not allow direct transfer or mapping of movement from one to the other.[3] This becomes even more evident when the robot should be, just as the demonstrator, standing on its own feet. For example, recorded joint movement of humans when squatting will, if directly copied to a humanoid robot, most likely result in the robot tipping over.

Thus the observed human motion needs to be adapted to the properties of the humanoid robot, but this requires the availability of models specifying robot kinematics and dynamics in order to control the robot's stability criterion.

Probably the most commonly used criterion to maintain robotic stability is the zero moment point (ZMP) ([4,5]), defined as the point on the ground where the tipping moment acting on the humanoid robot, due to gravity and inertia forces, equals zero ([6]). A biped humanoid robot is dynamically stable at any given time if its ZMP lies within the area defined by the convex hull of the supporting feet – in the double support phase, or one foot in the single support phase. ZMP is commonly used to evaluate the center of mass (CoM) acceleration boundaries, i.e. to determine the highest possible accelerations of the CoM, which keep the ZMP inside of the support polygon.

This method was for example used by Harada et al.,[7] who developed the ZMP dynamic-evaluation criterion, which enables generalized multicontact locomotion behaviors. Kajita et al.[8] designed a control system which minimizes the error between the desired ZMP and the output ZMP by applying a preview controller. Later Hyon et al.[9] proposed the compliant multicontact behavior using optimal distribution of contact forces. Even before that Sugihara et al.[10] applied an inverted pendulum control to generate dynamically stable walking patterns in real-time. The advantage of inverted pendulum approaches is that they require only a rough model of the robot dynamics to be successful.

One of the above mentioned approaches is commonly used to constrain the movement of the robot, so that the ZMP moves along the desired trajectory or even remains stationary.[11]

Humanoid robots are kinematically redundant.[12] The redundant DOFs can be used to effectively control the stability while performing some other task. The prioritized task control can be used to implement such behaviors. For the case of stability control, the motion of ZMP is considered as a primary task while other tasks or movements are considered as secondary tasks projected onto the null space of the primary task.

The goal of this paper is to show how to integrate stability control with motion capture systems to generate stable reproductions of human movements in real-time. We propose to exploit the kinematical redundancy of a humanoid robot and apply whole-body prioritized control. In the context of humanoid robots, prioritized control was used for example to enable the unified control of center of mass, operation-space tasks, and internal forces.[13] Prioritized control for locomotion and balance control was also addressed by Mistry et al.[14]

Since keeping the stability of a robot is normally the most important motor task, it thus constrains all other tasks to its null space and effectively alters the motions executed on the robot. In this paper we propose and evaluate a method which in certain situations allows unconstrained execution of the secondary task while the robot is securely stable. The primary task of stability control takes over only when approaching a predefined threshold, when the robot is in danger of becoming unstable. On top of that, it also allows smooth, continuous and reversible transition between the two modes. Such behavior, when applied to stability control, allows arbitrary movement of the robot while it is in a stable configuration. Furthermore, it does not interfere with the desired movement, for example the demonstrated movement the robot should track. Once a predefined threshold of a selected criterion, e.g. the location of ZMP is reached, the primary task takes over, and constrains the desired, demonstrated movement.

To demonstrate the applicability of the algorithm we show how it can be applied to real-time motion imitation of a humanoid robot, which at the same time preserves stability by standing on its own two feet. We performed the experiments on the Compliant Humanoid Platform or COMAN, which boasts 14 series-elastic joints, of which 6 in the legs are in the sagittal plane. The discrepancies between the CAD data of the robot and the real robot, and the passive elements in the kinematic chains lead to an error of the estimation of the center-of-mass. We show in the paper how we can model the discrepancies with the use of support vector machines, a supervised machine learning approach.[15,16] Other approaches were demonstrated to account for the behavior of the springs on the same platform. Lee et al.[17] have used a time-delay estimation in their control scheme, focusing on the behavior when carrying load. On the other hand, Mosadeghzad et al.[18] have proposed optimal compliance regulation. The emphasis of the paper was on the control with respect to external impacts. A model free approach, completely excluding the kinematics, was used for postural control of the same compliant robotic platform by Gay et. al.[19] In their approach, the authors used visual flow and gyroscopes as the input into optimized neural networks. In our paper, we show how we can perform postural control and motion imitation online, without of-line optimization.

To implement the real-time motion imitation we used a low cost RGB-D sensor, namely Kinect for the tracking of a human body. A similar approach applied to a dynamic simulation was proposed by.[20] Real-time motion transfer using precise motion capture on a Nao robot was described by.[21] Dynamic motion capture and imitation using motion capture was described by Ramos et al.[22] The paper describes of-line optimizations of motion and uses precise motion capture, while we describe real-time on-line motion imitation, where the possibility of optimizing motions is limited by the time-step of the control loop. Even so, we achieve reactive and stable motion imitation, which we demonstrated on a real robotic platform. In a recent paper, Zheng and Yamane[23] have extended motion tracking with strict contact force constraints, implemented by solving a nonlinear optimization problem with complex constraints in every control-loop step. They demonstrated the results in a dynamics simulator.

In order to apply the prioritized task control on the robot one needs the complete kinematic description of the robot and the means to control the CoM or ZMP using inverse kinematics. In Section 2 we briefly outline the calculation of kinematic descriptions of humanoid robots. In Section 3 we present motion imitation based on prioritized task control. The paper continues with the algorithm to manipulate the ZMP through the COM and the final prioritized control. Section 4 explains the modified task control, while Section 5 gives the results on the real robot. In Section 6 we describe how we can model the behavior of passive elements of the robot using SVM. Discussion and conclusions are given in Section 7.

2 Kinematics of a Humanoid Robot

When calculating the kinematics of a humanoid robot, one has to take into consideration that the robot is not attached to the ground, as it is the case with conventional industrial manipulators. A humanoid robot is bound to the ground by a one-way constraint, given by the current support plane, for instance with the feet. Defining an inertial frame is necessary in order to describe the position and orientation of the multi-legged kinematic chain with the use of systematical approaches for serial mechanisms.

The humanoid robot can be modeled as a combination of four kinematic chains, one for each limb, which all originate in the same starting point, called the base or root.[24] This point is often in the "abdomen" of the robot. The base frame attached to the robot is then connected to the inertial frame via 6 unactuated DOFs. In a kinematical aspect, using these DOF to calculate the kinematics becomes equivalent to imposing a null velocity reference to the feet.[24] Since these DOFs cannot be directly actuated, the term floating-base systems is often used to describe them.

Systematical approaches for serial mechanisms can be used to describe the kinematics of each of the four chains of a humanoid robot. The four chains consist of the two legs and the two arms (see Fig.1 showing the robot). Any systematical approach, such as the DenavitHartenberg (DH) parameters or the vector parameters[25] can be used for the description of the kinematic description of the chains.

3 Motion Imitation with Stability Control

The task of our algorithm is to allow on-line motion imitation on top of stability control. Therefore we have chosen the primary task to be stability control and the secondary task to be imitation of a demonstrator's movements, extracted with the Kinect sensor. In order to keep the robot stable, we wish to manipulate ZMP through the CoM. The relationship between the velocity of the center of mass in base coordinates (denoted by ^{b}) $^{b}\mathbf{x}_{\mathrm{CoM}}$ and joint angle velocity $\dot{\mathbf{q}}$ is given by the Jacobian of the center of mass $\mathbf{J}_{\mathrm{CoM}} \in \mathbb{R}^3$.

3.1 Center of Mass Jacobian

The center-of-mass Jacobian in base coordinates ${}^b\mathbf{J}_{\mathrm{CoM}}$ is obtained from

$$ {}^b\mathbf{x}_{\mathrm{CoM}} = \frac{\sum_{i=1}^n m_i^b \mathbf{x}_i}{\sum_{i=1}^n m_i} \tag{1} $$

from the relation

$$ {}^b\dot{\mathbf{x}}_{\mathrm{CoM}} = \frac{\sum_{i=1}^n m_i \; {}^b\mathbf{J}_i \dot{\mathbf{q}}}{\sum_{i=1}^n m_i} = \frac{\sum_{i=1}^n m_i \; {}^b\mathbf{J}_i}{\sum_{i=1}^n m_i} \dot{\mathbf{q}} = \; {}^b\mathbf{J}_{\mathrm{CoM}} \dot{\mathbf{q}}. \tag{2} $$

where ${}^b\mathbf{J}_i$ is the geometric Jacobian of the center of mass of body part i in base coordinates. Algorithm 1 gives a pseudo code on how to calculate the CoM Jacobian.

Algorithm 1 Center of Mass Jacobian

```
function J_CoM
    M = Σ_{j=1}^{n} m_j
    for all kinematic chains do
        m_λ = 0;
        for j = n : −1 : 1 do
            m_λ = m_λ + m_j
            p_{CoM,j} = m_j x_{CoM,j} / m_λ − O_j
            J_{CoM,j} = m_λ / M (r_j × p_{CoM,j})        ▷ × cross product
```

Basically, to calculate the center-of-mass Jacobian, one calculates how much a differential motion of a separate joint differentially displaces the center of mass. The pseudocode provided in algorithm 1 starts at the end of a kinematic chain and calculates the effect of moving the last joint, all the way to the first joint in the chain, which moves the mass of the complete chain. In this pseudocode, the variable $\mathbf{p}_{\mathrm{CoM_j}}$ is an auxiliary variable, $\mathbf{O}_j$ refers to the origin of frame j, $\mathbf{r}_j$ is the $j-$th joint axis direction in the base frame, and m_λ is the recursively calculated mass from the current frame to the end of the kinematic chain. The complete $\mathbf{J}_{\mathrm{CoM}}$ is calculated by combining the $\mathbf{J}_{\mathrm{CoM},j}$ columns of all the kinematic chains.

Eq. 2 provides the geometric Jacobian of the center of mass of body part i in base coordinates. However, since we are dealing with a free floating base, one has to take into account that one or two support feet are fixed in the world coordinate system, as they provide the support for the robot. We therefore have to calculate the Jacobian matrix in the corresponding coordinate system of the support foot and take into consideration that the feet do not move. The velocities of the feet are 0, i.e. $\dot{\mathbf{x}}_R = \omega_R = 0$ and $\dot{\mathbf{x}}_L = \omega_L = 0$. The variables $\dot{\mathbf{x}}_{R,L}$ and $\omega_{R,L}$ stand for respectively the linear and the angular velocities of both feet in the world coordinate system. It was shown by[10] that the ${}^{b}J_{\text{CoM}}$ can be transformed to assume the main support foot

$$\mathbf{J}_{\text{CoM},F} = \mathbf{R}(\,{}^{b}\mathbf{J}_{\text{CoM}} - \,{}^{b}\mathbf{J}_{F} + \Omega(\,{}^{b}\mathbf{x}_{\text{CoM}} - \,{}^{b}\mathbf{x}_{F})\,{}^{b}\mathbf{J}_{\omega F}), \tag{3}$$

F being either L or R (i.e. left or right foot). Here $\Omega(\mathbf{v})$ is defined as

$$\Omega(\mathbf{v}) = \begin{bmatrix} 0 & -v(3) & v(2) \\ v(3) & 0 & -v(1) \\ -v(2) & v(1) & 0 \end{bmatrix}. \tag{4}$$

and $\mathbf{R}$ is the orientation of the base of the robot in world coordinates. ${}^{b}\mathbf{J}_{F}$ and ${}^{b}\mathbf{J}_{\omega F}$ are the translational and rotational part of the Jacobian of the foot, while ${}^{b}\mathbf{x}_{F}$ is the position of the foot, all in robot base coordinates.

To maintain the other foot on the ground in double support phase, we have to add the constraint which prevents the other foot from moving. For example, if $F = R$ in eq. (3), we have to add the constraint

$$\mathbf{J}_{L}\dot{\mathbf{q}}_{LW} = 0, \tag{5}$$

where $\mathbf{J}_L \in \mathbb{R}^{6xn}$ is the Jacobian of the left foot in the world coordinates and $\mathbf{q}_{LW}$ the joints that span the chain from the right to the left foot. Figure 2 illustrates the situation. Since we have all the Jacobian matrices calculated in the base coordinate systems, i.e. the kinematic chains originating in the abdomen of the robot, we have to generate the Jacobian (in our case when F $= R$) matrix that defines the relation between the joints of both legs and the tip of the left foot with respect to the tip of the right foot. The transformation can be derived from

$$\mathbf{T}_L^R = \mathbf{J}_L = \begin{bmatrix} \mathbf{R}_R^T\mathbf{R}_L & \mathbf{R}_L^T(\mathbf{x}_R - \mathbf{x}_L) \\ \mathbf{0} & 1 \end{bmatrix}, \tag{6}$$

and deriving separately for the position and the orientation parts. By replacing $\mathbf{x}$ with $\mathbf{J}\dot{\mathbf{q}}$ and expressing separately for the joints of the left and right foot, we get

$$\mathbf{J}_L = \begin{bmatrix} -\mathbf{R}_R\Omega(\mathbf{x}_L - \mathbf{x}_R)^T\mathbf{J}_{\omega R} - \mathbf{R}_R^T\mathbf{J}_{pR} & \mathbf{R}_R^T\mathbf{J}_{pL} \\ -\mathbf{R}_R^T\mathbf{J}_{\omega R} & \mathbf{R}_R^T\mathbf{J}_{\omega R} \end{bmatrix}, \tag{7}$$

$$\mathbf{q}_{LW} = \begin{bmatrix} \mathbf{q}_R \\ \mathbf{q}_L \end{bmatrix}. \tag{8}$$

Considering the constraints of the support feet, the velocity of the center of mass and the kinematic constraints with respect to the joint motion, can now be expressed as

$$\dot{\mathbf{x}}_e = \mathbf{J}_e\dot{\mathbf{q}}, \tag{9}$$

where index $_e$ stands for augmented. The augmented Jacobian accounts for both the stability task and the kinematic constraint with

$$\dot{\mathbf{x}}_e = \begin{bmatrix} \dot{\mathbf{x}}_{\text{CoM}} \\ \mathbf{0} \end{bmatrix}, \tag{10}$$

$$\mathbf{J}_e = \begin{bmatrix} \mathbf{J}_{\text{CoM}} \\ \\ \mathbf{J}_F \end{bmatrix}, \tag{11}$$

for the double support phase. For the single support phase eqs. (10,11) simplify into $\dot{\mathbf{x}}_e = \dot{\mathbf{x}}_{\text{CoM}}$ and $\mathbf{J}_e = \mathbf{J}_{\text{CoM}}$.

An alternative approach to constraining the motion of the non-leading foot would be to simply set the primary task of the robot to maintain the position of the other foot and then map the stability control to the null space of the task. The drawback is mainly in not having the stability as the primary task and therefore the velocities for maintaining the stability are always projected through the null space of the task of keeping the feet stationary.

3.2 ZMP Manipulation Through CoM Jacobian

Controlling the center-of-mass allows for the control of static stability. In order to control the dynamic stability of a humanoid robot we need to control its motion so that ZMP stays within the support polygon. It was shown by Sugihara et al.[10] that, neglecting the inertia matrices, the relationship between the CoM, defined in eq. (1) and given by $\mathbf{x}_{\text{CoM}} = [x_{\text{CoM}}, y_{\text{CoM}}, z_{\text{CoM}}]$, and the ZMP can be expressed by

$$\ddot{x}_{\text{CoM}} = \omega^2(x_{\text{CoM}} - x_{\text{ZMP}}), \tag{12}$$

$$\ddot{y}_{\text{CoM}} = \omega^2(y_{\text{CoM}} - y_{ZMP}), \tag{13}$$

$$\omega = \sqrt{\frac{\ddot{z}_{\text{CoM}} + g}{z_{\text{CoM}} - z_{ZMP}}} \tag{14}$$

Here g is the gravitation constant. Eq. (14) requires desired ZMP planning to calculate the desired z_{CoM}, which can be obtained from an inverted pendulum control. For details on inverted pendulum control see Kajita et al.[26]

Figure 3 shows real robot results of manipulating the measured center of pressure (CoP), which can be assumed to represent the ZMP when within the support polygon,[27] with the use of the CoM Jacobian. The main advantage is that the robot can react to external forces. In the results of Fig. 3 we can see the measured forces, the desired ZMP location, the actual CoP location and the actual (estimated) CoM location if both forward-backward (x) and left-right (y) directions of the robot. We can see that if an external force appears, the CoM is shifted. Due to the passive elements of the robot, the location of the CoP overshoots when external forces disappear and the robot wobbles slightly. The offset of the forces in the y direction show a discrepancy between the model and the real robot.

3.3 Prioritized task control

Stable reproduction of human movements can be formulated using prioritized control. Classically, one defines the stability as the primary task and movement imitation as the secondary task. This leads to the control policy

$$\dot{\mathbf{q}} = \mathbf{J}_e^+ \dot{\mathbf{x}}_e + \mathbf{N}\dot{\mathbf{q}}_{KIN} \tag{15}$$

where $N = (\mathbf{I} - \mathbf{J}_e^+ \mathbf{J}_e)$ defines the null space of $\mathbf{J}_e$ and $\dot{\mathbf{q}}_{KIN}$ are the desired joint angles velocities to account for the Kinect tracking of the human motion, with $\dot{\mathbf{q}}_{KIN} = k_p(\mathbf{q}_{actual} - q_{KIN})$ and k_p a positive gain.

When controlling the non-supporting leg of the robot in the single stance phase, one should exclude some of the degrees of freedom from the above matrices. The other degrees of freedom should preserve the stability.

4 Modified Prioritized Task Control

In the double support phase the robot allows considerable motion of the upper part of the body that does not move the ZMP out of the support polygon. The lower part, namely the feet, are completely constrained and remain motionless on the ground.

In order to allow upper body to freely move until the ZMP starts approaching the support polygon, we divide the problem per degrees of freedom. While the degrees of freedom of the legs follow the control policy from Section 3.3, we propose using a modified task control for the arms and the body of the robot. The control method is based on the reflexive stability control framework for humanoid robots,[3] which allows unconstrained motion while the ZMP is well within the stability polygon. In this paper we evaluate for the first time the approach on a real robot in 3 dimensions. The modified prioritized control policy suggests

$$\dot{\mathbf{q}} = \eta(\mathbf{x}_{ZMP})^n \mathbf{J}_e^+ \dot{\mathbf{x}}_e + \mathbf{N}_\eta \dot{\mathbf{q}}_{KIN}, \tag{16}$$

with

$$\mathbf{N}_\eta = (1 - \eta(\mathbf{x}_{ZMP})^n)\mathrm{diag}(\mathbf{N}) + \eta(\mathbf{x}_{ZMP})^n \mathbf{N} \tag{17}$$

and $\mathbf{N} = (\mathbf{I} - \mathbf{J}_e^+ \mathbf{J}_e)$. The weighting function $\eta(\mathbf{x}_{ZMP})$ defines the transition between the constrained, i.e. in the null space of the stability, and unconstrained motion imitation. The weighting function takes into account the normalized distance of the ZMP to the edge of the support polygon

$$\eta(\mathbf{x}) = \begin{cases} \dfrac{d(\mathbf{x}_p) - d(\mathbf{x})}{d(\mathbf{x}_p) - d_{min}} & , d(\mathbf{x}) > d_{min} \\ 1, \text{ else} \end{cases} \tag{18}$$

with $\mathbf{x}_p$ defining the center of the support polygon and d_{min} being the minimal allowed distance to the edge of the support polygon.

Alternatively to eq.(17), one can also use

$$\mathbf{N}'_\eta = \mathbf{I} - \eta(\mathbf{x}_{ZMP})^n \mathbf{J}^+ \mathbf{J}' \tag{19}$$

For the details on such use see Petrič et al.[3]

5 Experimental Evaluation

In this section we present both simulation and real-world application of the proposed modified task priority algorithm for stability control.

5.1 Compliant Humanoid Platform COMAN

The Compliant Humanoid Platform COMAN[28,29] approximates the dimensions of a 4 year old child, with the height from the foot to the center of the neck 945mm. The distance between the centers of the shoulders is 312mm. The total weight of the robot is 31.2kg, out of which the legs and the waist module weigh 18.5kg. The complete robot has 25 DOF, but the 2 neck degrees of freedom are not being used at the time. Each leg has 6 DOF: 3 at the hip, 1 at the knee level and 2 at the ankle. For the trunk there is a 3 DOF waist while each arm has currently 4 DOF, i.e. 3 in the shoulder and 1 in the elbow. Passive compliance based on series elastic actuation (SEA) was added to the 14 of the 25 DOF including all flexion/extension DOF of the legs, the flexion/extension of the shoulders and elbows and the shoulder abduction/adduction. The robot is presented in Fig. 1.

In the motion imitation algorithm we used the Kinect sensor to track and imitate the motion of the complete arms (4 DOF) and of the hips and knees of the legs. Additionally, we implemented the rotation of the torso around the vertical axis. This was calculated from the positions of the shoulder joints of the demonstrator.

5.2 Experimental results

The difference when using modified prioritized task control compared to using standard prioritized task space control is that the task with the higher priority is only observed when necessary, so stability is only controlled when necessary. This can be clearly seen in the results of an experiment, where we set the desired hip angles of the robot to sinusoidally oscillate from the original configuration at -0.3 rad to $-\pi/2$ rad, resulting in the robot bending forward and backward periodically. The motion of the hips is presented in the top plot of Fig. 4. In the bottom plot we can see the location of the CoM. It remains stationary when using the classical approach, as reflected in eq. (15), which through the primary task reduces the error of the CoM. On the other hand, when using the modified task space approach, the CoM moves because, as defined in (16), the primary task is pre-multiplied with $\eta(\mathbf{x}_{ZMP})^n$, which is virtually zero when close to the center of the support polygon.

The stability control was set to fully take over 6 cm from the edge of the stability polygon. Fig. 5 shows in the top plot how this affects the behavior of other joints, in the given case the ankles. We can see that when using the modified approach, the joint values remain constant (one instance marked with dashed lines) when the distance from the edge of the support polygon is sufficient, given by $\eta(\mathbf{x}_{ZMP})^5$ as defined in eq. (18). The value of $\eta(\mathbf{x}_{ZMP})^5$ is shown in the bottom plot. In other words, the stability control is not active and does not change the (desired) joint positions when $\eta(\mathbf{x}_{ZMP})^5 \cong 0$.

Figure 6 shows a sequence of photos showing a simulated robot in a dynamic simulator Webots[30] imitating the motion of a human in real time. The sequence shows the robot lifting one foot. When using the modified task priority control, the demonstrator can move the CoM within the support polygon, but has to observe the current location of CoM to perform the required motion. In our case we defined the desired CoM to move under one foot when the tracking detected that the other foot was considerably higher.

Figure 7 shows the real-time motion imitation of COMAN robot. The demonstrator was tracked with the Kinect sensor. We can see imitation with the arms, the body and with the legs when performing a squat and bending over. The robot safely and reliably maintained the stability with very little delay, which can only be observed in very fast demonstrator motions. The algorithm has proven very robust and would only fail in the case of tracking errors. A video showing the real-time motion imitation on the real robot is available at http://biorob.epfl.ch/files/content/sites/biorob/files/public/Coman/KinectDemoVideo.mov.

6 Estimating Robot-Model Discrepancies Using Support Vector Machines

Since we used only the CAD data to describe the mass properties of the robot and since we do not account for the passive elements, there is a discrepancy between the position of the center of mass $\mathbf{x}_{\mathrm{CoM}}^{\mathrm{model}}$ as calculated from the available model data and the actual CoM $\mathbf{x}_{\mathrm{CoM}}$. While the discrepancy between the model and the real CoM is present in both forward-backward (anteroposterior) and left-right (mediolateral) direction of the robot, all of the springs act in the sagittal plane and therefore the discrepancy is larger in the anteroposterior direction. In this section we show how we can account for the discrepancy in the forward-backward direction using support vector machines (SVM).[15] A similar approach using Gausian Process Regression (GPR) was used to correct the estimation of kinematics of a mechanism for manipulation.[31]

In our approach we first record a very slow and stable motion of the robot, which covers the expected human demonstrated motion and maintains postural stability. Due to very slow motion we can assume that the measured center of pressure $\mathbf{x}_{CoP}$ obtained from pressure sensors on the feet is approximately the same as the center of mass $\mathbf{x}_{\mathrm{CoM}}$. They both move within the support polygon. We can model the error between $\mathbf{x}_{\mathrm{CoM}}^{\mathrm{model}}$ and the measured $\mathbf{x}_{\mathrm{CoP}} \approx \mathbf{x}_{\mathrm{CoM}}$ using SVM regression. We perform the estimation and correction only in the anteroposterior (x) direction of the robot. SVM training was implemented using the LIBSVM[15] library in Matlab. After training we can estimate the discrepancy as follows

$$x_{\text{CoM}}^{\text{corrected}} = x_{\text{CoM}}^{\text{model}} + \Delta x \tag{20}$$

$$\Delta x = f_{\text{SVM}}(x_{\text{CoM}}^{\text{model}}, \mathbf{q}), \tag{21}$$

where f_{SVM} is the function estimated by SVM regression and $\mathbf{q}$ are the robot's joint angles. The data for learning consists of $x_{\text{CoP},i}$, $x_{\text{CoM},i}^{\text{model}}$, $\mathbf{q}_i$. $i = 1, \ldots, N$ are the sample indices. The training outputs are calculated as

$$\Delta x_i = x_{\text{CoP},i} - x_{\text{CoM},i}^{\text{model}}. \tag{22}$$

Theoretically, all joint angles affect the stability of the robot. However, it would require a large amount of training data to estimate f_{SVM} if all of the joint angles were considered in the optimization process. To reduce the dimensionality of the input space, we rather use the center of pressure x_{CoM} calculated from the available model and a small number of joints that affect the stability most. These are the leg joints, i. e. ankle, knee, and hip joints. Thus the input joint angles $\mathbf{q}_i$ consist of some subset of the measured joint angles of the legs. The different joint angle combinations we tested are: ankle joints, additionally added knee joints, and finally also with added hip joints. Figure 8 shows the results of using different input data for estimating the discrepancy between the real CoM and the CoM calculated from the model. For testing we used data that was not used for estimating the SVM regression function f_{SVM}. Table 1 shows the standard deviations of the difference between the corrected center of pressure $x_{\text{COM}}^{\text{corrected}}$ and the center of pressure x_{CoP} estimated from the foot pressure sensors, i. e. $x_{\text{CoP}} - x_{\text{COM}}^{\text{corrected}}$. We can see that the standard deviation of the error increases in case D, which is a result of a finite set of training data. The best result was achieved when using ankle and knee joints in addition to the center of mass coordinates as input.

7 Discussion and Conclusion

We have shown that we can effectively apply the modified prioritized task control for simultaneous stability control and motion imitation in real-time. In this aspect, we have shown how to apply the described algorithm for both center-of-mass and center-of-pressure control approach. While the former is somewhat easier to implement, the latter takes into consideration the external forces and can adapt the posture of the robot accordingly.

If ZMP of the robot moves away from the center of the support polygon and approaches the edge of the support polygon, our stability control takes over, if necessary completely overriding the imitation. The primary task at that point only allows motion that would move the ZMP towards the center of the support polygon. The prioritized task control, through the Jacobian and if enough degrees of freedom are available, may also move the other joints so that the secondary task the imitation is observed.

The presented approaches are effective in controlling the stability, yet several issues remain with the applicability to the passively compliant platform used in the experiments. As COMAN boasts series elastic elements, i. e. springs after the motors, the behavior of the springs cannot be directly influenced and specialized controllers need to be developed to account for the spring behavior. While the springs come in handy for interaction with the environment and walking, i. e. to reduce the impact forces, for the task of stability they simply introduce an error in the posture. Nevertheless, we successfully demonstrated that our method can be applied, despite the inaccuracies brought by the springs. They can be partially accounted for by the proposed SVM regression method. For this method, we first acquire a data set of CoM values obtained from the available kinematic model, the center of pressure values estimated from the foot pressure sensors, and the associated joint angles of the robot. In the future we would like to improve these results with a more in-depth analysis of this approach.

The modified stability approach has allowed us to transfer the motion of the demonstrator to the robot in real time, including the lifting of separate legs. This proves that the proposed method enables the transfer of human motion to the robot without the explicit need for the demonstrator to take into consideration the behavior of the robot. Since we do not explicitly control the stability all the time, but only when necessary, and by keeping a well defined prioritized control policy with smooth transitions between the tasks, we can perform a variety of tasks, which are not feasible with the strictly prioritized approach.

8 Acknowledgments

The work presented in this paper was supported by Sciex-NMS$^{\mathrm{CH}}$ project 406 12.018, FP7 project WALK-MAN (FP7-ICT 611832), FP7 project Symbitron (FP7-ICT 661626) and FP7 project Xperience (FP7-ICT 270273).

REFERENCES

[1] Ales Ude, Christopher G. Atkeson, and Marcia Riley. Planning of joint trajectories for humanoid robots using b-spline wavelets. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2223–2228, 2000.

[2] A. Gams, A.J. Ijspeert, S. Schaal, and J. Lenarčič. On-line learning and modulation of periodic movements with nonlinear dynamical systems. *Autonomous Robots*, 27(1):3–23, 2009.

[3] Tadej Petrič, Andrej Gams, Jan Babič, and Leon Žlajpah. Reflexive stability control framework for humanoid robots. *Autonomous Robots*, 34(4):347–361, 2013.

[4] Miomir Vukobratovic and Davor Juricic. Contribution to the synthesis of biped gait. *IEEE Transactions on Biomedical Engineering*, BME-16(1):1 –6, jan. 1969.

[5] Miomir Vukobratovic and Branislav Borovac. Zero-moment point - thirty five years of its life. *International Journal of Humanoid Robotics*, 1(1):157–173, 2004.

[6] P. Sardain and G. Bessonnet. Forces acting on a biped robot. center of pressure-zero moment point. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 34(5):630–637, 2004.

[7] K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa. ZMP analysis for arm/leg coordination. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 75 – 81 vol.1, oct. 2003.

[8] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1620–1626 vol.2, 2003.

[9] Sang-Ho Hyon, J.G. Hale, and G. Cheng. Full-body compliant human-humanoid interaction: Balancing in the presence of unknown external forces. *IEEE Transactions on Robotics*, 23(5):884 –898, oct. 2007.

[10] T. Sugihara, Y. Nakamura, and H. Inoue. Real-time humanoid motion generation through ZMP manipulation based on inverted pendulum control. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1404–1409 vol.2, 2002.

[11] W. Suleiman, F. Kanehiro, K. Miura, and E. Yoshida. Improving ZMP-based control model using system identification techniques. In *9th IEEE-*

RAS International Conference on Humanoid Robots (Humanoids), pages 74 –80, dec. 2009.

[12] N. Mansard and F. Chaumette. Task sequencing for high-level sensor-based control. *Robotics, IEEE Transactions on*, 23(1):60 –72, feb. 2007.

[13] L. Sentis, Jaeheung Park, and O. Khatib. Compliant control of multicontact and center-of-mass behaviors in humanoid robots. *IEEE Transactions on Robotics*, 26(3):483–501, 2010.

[14] M. Mistry, J. Nakanishi, and S. Schaal. Task space control with prioritization for balance and locomotion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007)*, pages 331–338, 2007.

[15] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[16] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[17] Jinoh Lee, Houman Dallali, Nikolaos Tsagarakis, and Darwin Caldwell. Robust and Model-Free Link Position Tracking Control for Humanoid COMAN with Multiple Compliant Joints. In *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 56 – 61, 2013.

[18] Mohamad Mosadeghzad, Zhibin Li, Nikos Tsagarakis, Gustavo A. Medrano-Cerda, Houman Dallali, and Darwin G. Caldwell. Optimal Ankle Compliance Regulation for Humanoid Balancing Control. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4118 – 4123, 2013.

[19] Sebastian Gay, Jesse van den Kieboom, Jose Santor-Victor, and Auke Jan Ijspeert. Model-Based and Model-Free Approaches for Postural Control of a Compliant Humanoid Robot using Optical Flow. In *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 1 – 7, 2013.

[20] Van Vuong Nguyen and Joo-Ho Lee. Full-body imitation of human motions with kinect and heterogeneous kinematic structure of humanoid robot. In *2012 IEEE/SICE International Symposium on System Integration (SII)*, pages 93–98, 2012.

[21] J. Koenemann and M. Bennewitz. Whole-body imitation of human motions with a nao humanoid. In *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 425–425, 2012.

[22] O.E. Ramos, L. Saab, S. Hak, and N. Mansard. Dynamic motion capture and edition using a stack of tasks. In *2011 11th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 224–230, 2011.

[23] Yu Zheng and Katsu Yamane. Human Motion Tracking Control with Strict Contact Force Constraints for Floating-Base Humanoid Robots. In *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 1 – 7, 2013.

[24] Agostino Santis, Giuseppe Gironimo, Luigi Pelliccia, Bruno Siciliano, and Andrea Tarallo. Multiple-point kinematic control of a humanoid robot. In *Advances in Robot Kinematics: Motion in Man and Machine*, pages 157–168. Springer Netherlands, 2010.

[25] T. Bajd, M. Mihelj, J. Lenarčič, A. Stanovnik, and M. Munih. *Robotics*, volume 43 of *Intelligent Systems, Control and Automation: Science and Engineering*. Springer Science+Business Media B.V., 2010.

[26] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa. The 3d linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. In *2001 IEEE/RSJ International Conference on Intelligent Robots and Systems.*, volume 1, pages 239–246 vol.1, 2001.

[27] Ambarish Goswami. Postural stability of biped robots and the foot-rotation indicator (FRI) point. *The International Journal of Robotics Research*, 18(6):523 – 533, 1999.

[28] F.L. Moro, N.G. Tsagarakis, and D.G. Caldwell. A human-like walking for the compliant humanoid coman based on com trajectory reconstruction from kinematic motion primitives. In *2011 11th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 364–370, 2011.

[29] L. Colasanto, N.G. Tsagarakis, and D.G. Caldwell. A compact model for the compliant humanoid robot coman. In *2012 4th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 688–694, 2012.

[30] O. Michel. Webots: Professional mobile robot simulation. *Journal of Advanced Robotics Systems*, 1(1):39–42, 2004.

[31] Peter Pastor, Mrinal Kalakrishnan, Jonathan Binney, Jonathan Kelly, Ludovic Righetti, Gaurav Sukhatme, and Stefan Schaal. Learning task error models for manipulation. In *2013 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2612–2618, 2013.

List of Figures

Figure 1: COmpliant HuMANoid Platform – COMAN, developed by IIT, and used in the experiments to demonstrate the possibility of using modified task space control for motion imitation.

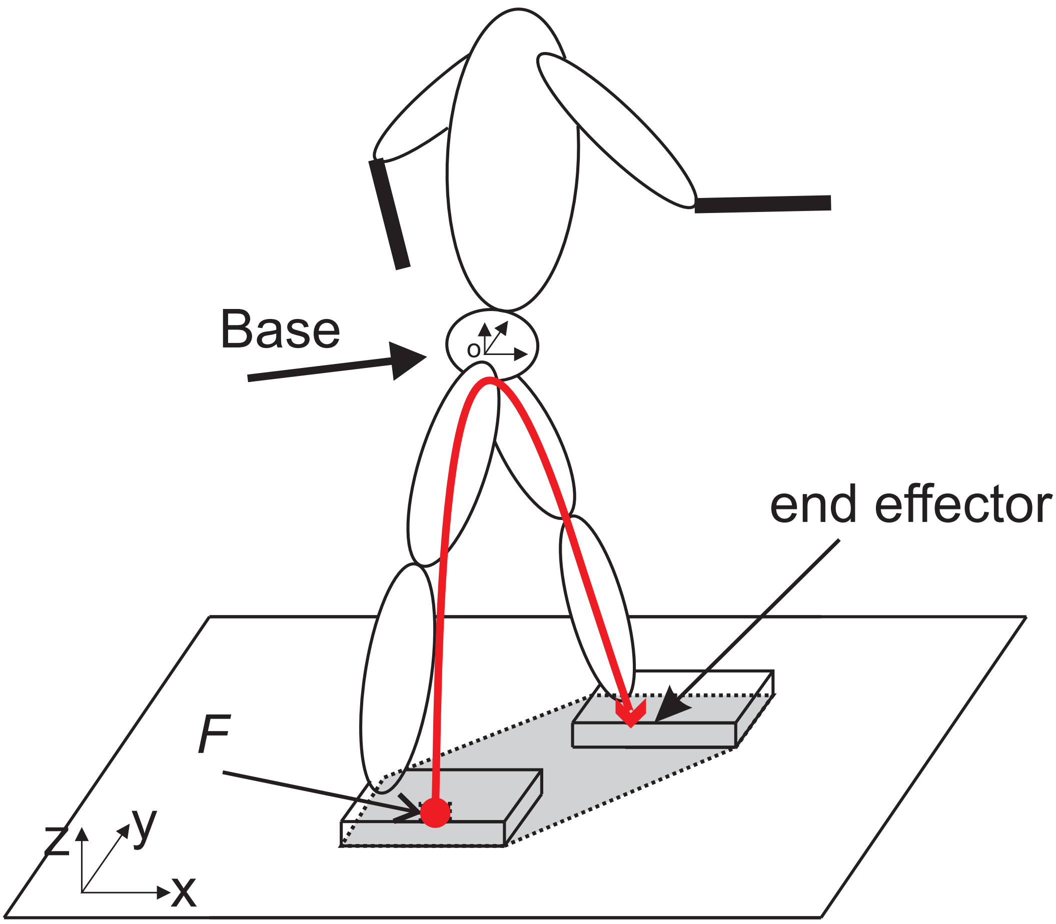

Figure 2: Based on the assumption that the feet do not move when the robot is standing, one foot F is considered as the new, fictional base of the robot. The Jacobian of the CoM has to be transformed so that it assumes the new base. The same goes for the other foot. When maintaining the other, non-leading foot stationary, one can consider the chain from one foot to the other as a serial mechanism, given by the red arrow.

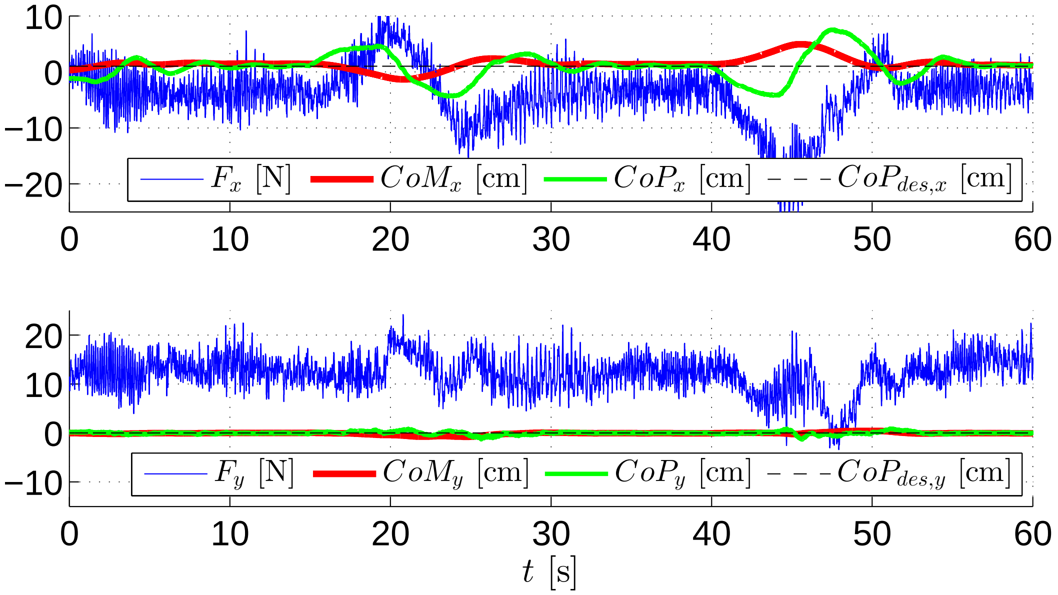

Figure 3: The locations of the CoM (red), CoP (green), desired CoP (dotted) and the measured forces (blue) in the x direction of the robot (forward-backward) in the top plot. The same for the y direction in the bottom plot. The forces are in N while the locations are in cm (for scale) relative to the most stable point of the support polygon.

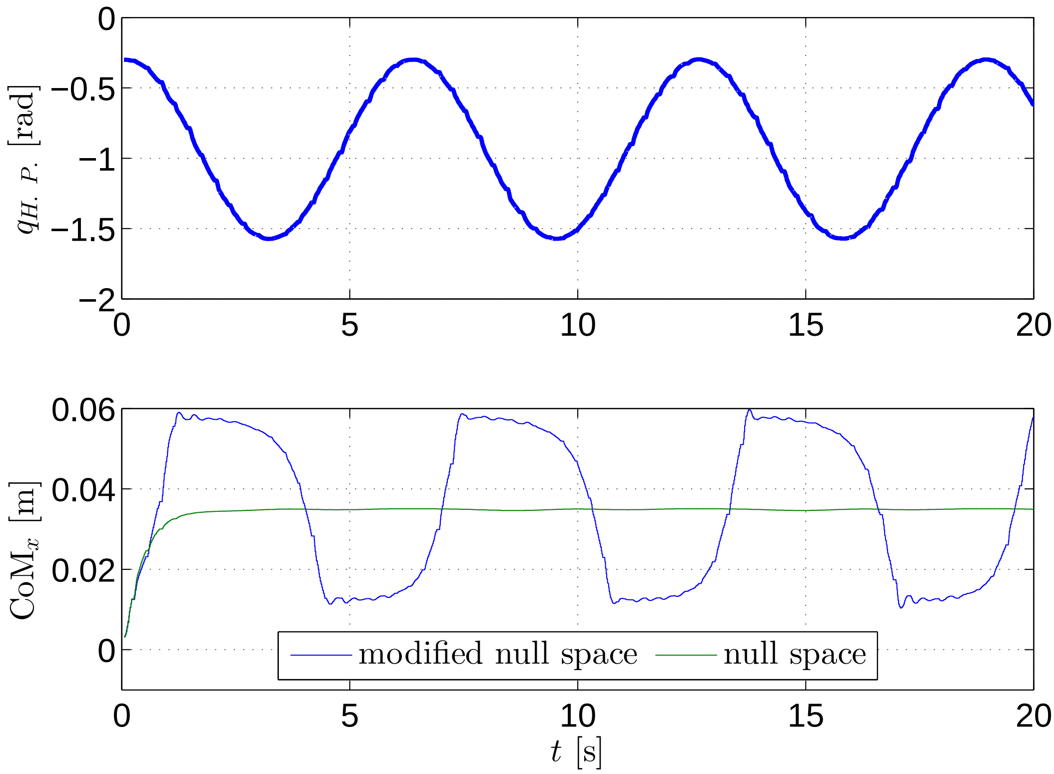

Figure 4: Top: the hip pitch angles for both both hips ($q_{H.\ P.}$). Bottom: the resulting forward-backward location (x-direction) of the CoM projected along the z-axis when using the modified (in blue) or classical (in green) prioritized task control. While the classical prioritized task control does not allow any movement of the CoM, the modified null space controller only prevents the CoM from leaving the support polygon, at the same time allowing stable displacement of the CoM due to the movement of the hips.

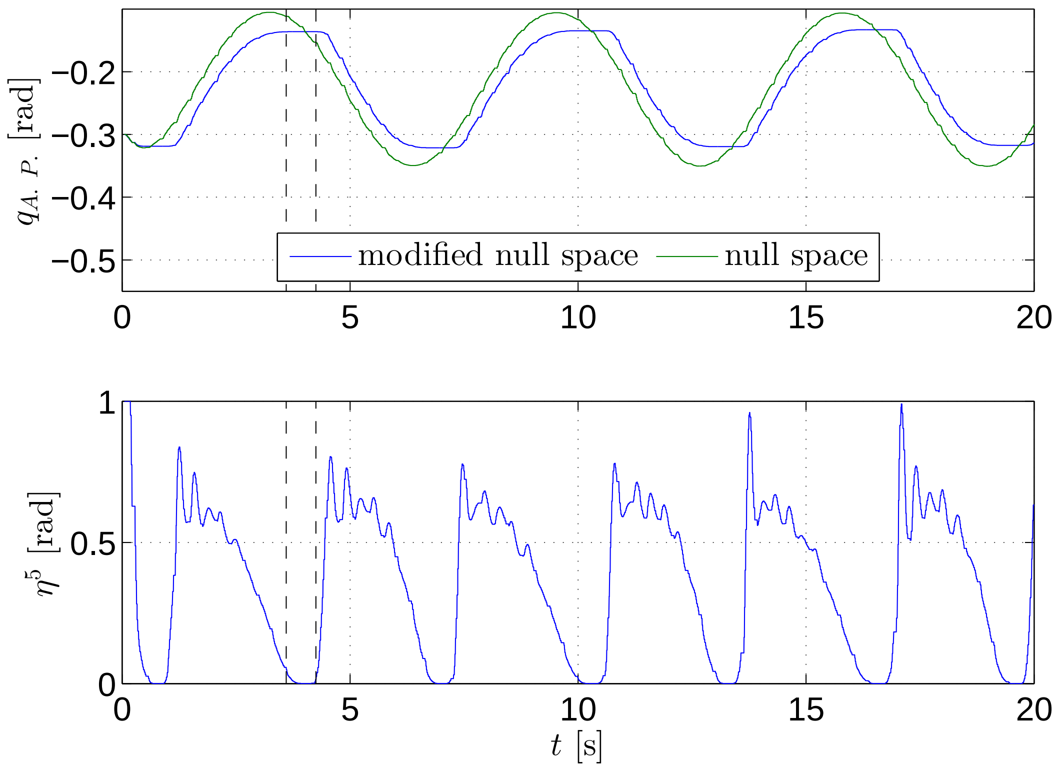

Figure 5: The motion of the ankles ($q_{A.\ P.}$) when maintaining the stability during the experiment presented also in Fig. 4 in the top plot. The value of η^5 is shown in the bottom plot. Vertical dashed lines mark a time span when the primary task, i.e. stability, is not controlled.

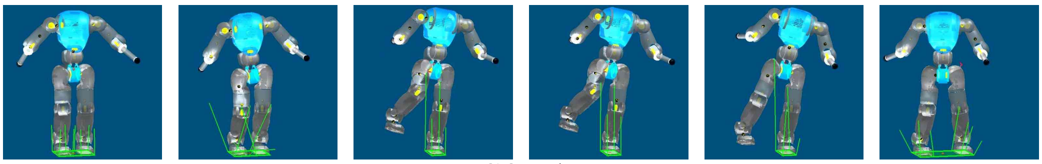

Figure 6: Images showing a simulated COMAN robot while imitating human behavior in real time. The sequence shows the example where the demonstrator performs a side-step.

Figure 7: Sequence of images showing real-time motion imitation with the robot while maintaining stability. The demonstrator performed random waving, squatting and bending motions, but maintained the double feet support at all times.

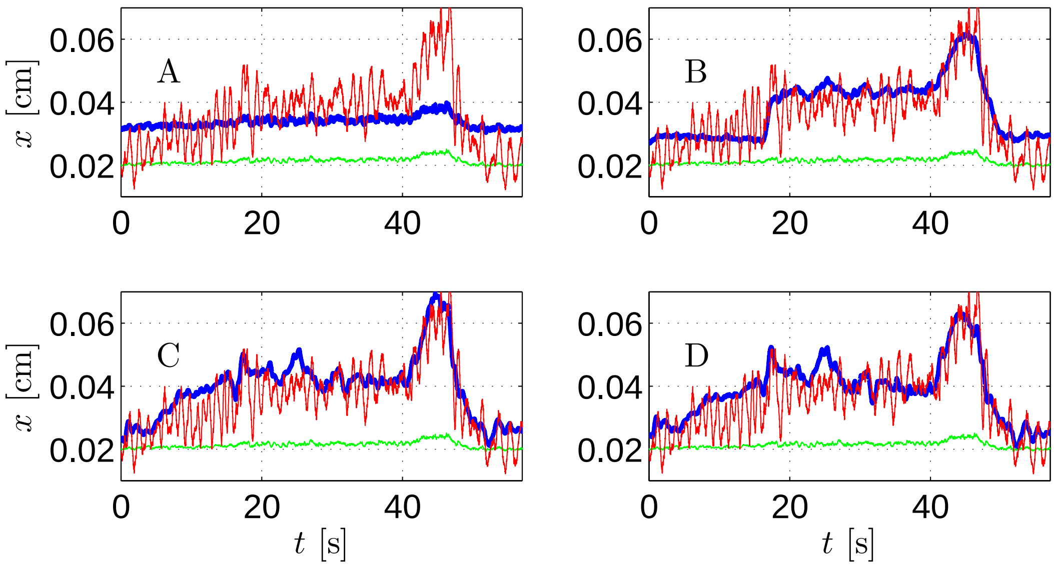

Figure 8: The results of modeling the robot–model discrepancy using different input data, presented on the test data. In all four plots, the trajectories are green for the x_{CoM}, red for the x_{CoP} and blue for the $x_{\mathrm{CoM}^{\mathrm{corrected}}}$. In case A we use only the estimated CoM as the input. In case B, we add the ankle joint values, in C we add also the knee joint values and in D also the hip joint values.

List of Tables

Table 1: Standard deviation of the error (in meters) of x_{COM} estimation using different input and training data

Input data	Standard Deviation
A	0.0102
B	0.0067
C	0.0062
D	0.0065