2011 11th IEEE-RAS International Conference on Humanoid Robots
Bled, Slovenia, October 26-28, 2011

Constraining Movement Imitation With Reflexive Behavior: Robot Squatting

Andrej Gams, Tadej Petrič, Jan Babič, Leon Žlajpah, Aleš Ude
Jožef Stefan Institute
Ljubljana, Slovenia
Email: {andrej.gams, tadej.petric, jan.babic, leon.zlajpah, ales.ude}@ijs.si

***Abstract*—Direct imitation of human movement with a humanoid robot, which has a similar kinematic structure, does not guarantee a successful completion of the task because of different dynamical properties. Our research starts by showing how to apply a generalization algorithm to extract the desired movement primitives from multiple human demonstrations. The emphasis of the paper is on a method that constrains the extracted movement primitives when mapping them to a robot, taking into account a critical criterion of the task. As a practical example we study the stability of a robot, which is determined through a normalized zero-moment-point. Our approach is based on prioritized task control and allows direct movement transfer as long as the selected criterion is met. It only constrains the movement when the criterion approaches a critical condition. The critical condition thus triggers a reflexive, subconscious behavior, which has higher priority than the desired, conscious movement. We demonstrate the properties of the algorithm on a real, human-inspired leg robot developed in our laboratory.**

I. Introduction

Movement imitation is one of the approaches of transferring human movement to robotic mechanisms [1]. In the paper we show how we can constrain the recorded movement when mapping it to a robot, in order to maintain a chosen criterion. The criterion is in our case the stability of the robot of a human-inspired leg robot.

Different kinematic and dynamic properties of humans and robotic mechanisms do not allow direct transfer or mapping of movement from one to the other [2]. The resulting movement of the robot will most likely not accomplish the same task. For example, recorded joint movement of humans when squatting will, if directly copied to a humanoid robot, result in the robot tipping over. The movement has to be somehow changed or constrained to account for the difference in the kinematic structure and the dynamic properties.

One of the approaches is to transfer the task space movement, for example, the movement of the tip of the arm, thus completely ignoring the joint space movements of the demonstrator. In such case the joint movement is under the control of an inverse kinematics algorithm. Null-space motion and prioritized task control can be used to, for example, add desired joint movement as a secondary task. While the primary task is end-effector movement in task space, the secondary task, if possible, maintains similar poses as the demonstrator [3]. Another common use of the secondary task is to implement obstacle avoidance [4]. Such prioritized task space control is also referred to as Operational space control [5].

In our approach we change the formulation of the primary and the secondary task, so that the desired movement of the robot is in fact a secondary task. The primary task is only observed if we approach a pre-defined threshold. While far from the threshold, the algorithm allows direct control of separate joints. Upon approaching the threshold, the primary task smoothly takes over and only allows joint control in its null-space. The algorithm allows smooth transition in both ways – between observing the primary task with the secondary in its null space, or just the unconstrained secondary task. This allows unconstrained joint movement while not close to the threshold. As such, it mimics the human reflexive mechanism of retaining the stability of the posture. Similarly to the proposed algorithm, the proprioceptive and vestibular systems of the human body trigger the reflexes that ensure postural stability during an arbitrary motion [6], [7].

We demonstrate the approach on a stability task, where we perform squatting movements on a planar leg robot, which has similar kinematical properties as a human in the sagittal plane [8]. We use a normalized zero-moment point (ZMP) location as the criterion of the stability. Control of the stability criterion only takes over when it approaches the threshold, and even then the transition is smooth. Otherwise, we use direct joint control of the mechanism. The demonstrated trajectories, apart from the offsets, are not modified in advance.

A similar manner of robot control with two different tasks, where their desired joint velocities blend together, was presented by Sugiura et al. [9]. Contrary to the proposed method the authors used a blending coefficient that mixes two ”primary” tasks, each with its own null-space task. Our proposed method, on the other hand, allows continuous transition between a primary and a secondary task. The work by Mansard et al. [10] provides a general solution for integration of unilateral constraints. The authors use an activation matrix which modifies the Jacobian of the task.

To acquire the demonstrated trajectories of movement for the desired task we recorded several executions of chosen movements, for example squatting. As human movement can never be completely reproduced in timing and in movement itself, representative movement or movement primitives have to be somehow extracted. We use a generalization algorithm [11] based on a regression technique to acquire joint movement primitives for the demonstrated tasks. We modified the algorithm to evenly weight separate trajectories and thus extract

primitives of movement, encoded in the dynamic movement primitive (DMP) framework [12], [13].

The rest of the paper is organized as follows. In section two we first show how we extract and encode motion primitives from the demonstrations trough generalization. In section three we present our approach on how to constrain the joint movement of the robot with reflexive behavior and show how to calculate of the ZMP criterion. In section four we present our leg robot and give results of real-world experiments. Conclusions and discussion are given in section five.

II. Motion Primitives

A. Data acquisition

Using motion capture we measured the movement of several subjects performing different tasks. We used 3D investigator motion capture equipment from NDI. Our measuring set-up consisted of three position sensors, where each of these sensors consists of three infrared cameras. We used active markers. Fig. 1 shows the leg robot, developed at our laboratory, and a silhouette of a person, side by side in the sagittal plane. We can see that the kinematical structure of the planar mechanism and a human are similar in this plane. A dynamical model from SimMechanics is presented in the right. The placement of the

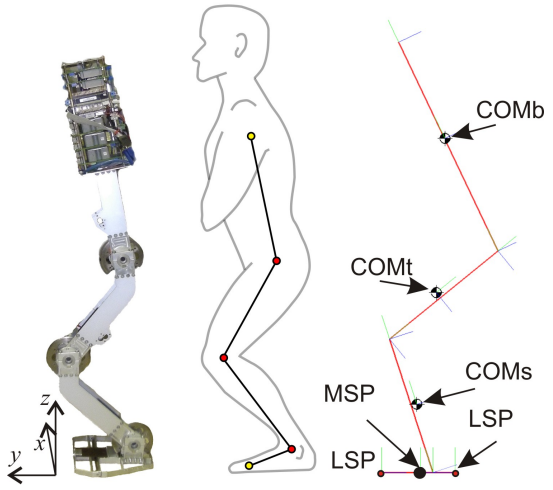

Fig. 1. Similarity in kinematical structures of the leg robot and a person. The dots on the silhouette show the location of the markers for motion capture. A dynamical model from SimMechanics with marked centers of mass (COM) for the body, thigh and shank, is shown on the right. Most stable point (MSP) and the least stable points (LSP) are marked as well.

markers for motion capture is conditioned by the kinematical structure of the robot. In our case we could acquire joint movements of the hip, the knee and the ankle using 5 markers, as shown in Fig. 1. These were attached to the demonstrator at these joints, plus a marker on the foot and one on the body. Using these 5 markers we can calculate the joint angles by

$$\Phi = \arccos \frac{\mathbf{a}^T \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|}, \tag{1}$$

where Φ is the joint angle and $\mathbf{a}, \mathbf{b}$ are vectors defined by adjacent markers.

B. Movement Primitives

We extract movement primitives for separate joints from a set of recordings, and encode them in the dynamic movement primitive (DMP) framework [12], [13]. A set of equations

$$\tau \dot{z} = \alpha_z(\beta_z(g - y) - z) + f(x), \tag{2}$$

$$\tau \dot{y} = z, \tau \dot{x} = -\alpha_x x, \tag{3}$$

where

$$f(x) = \frac{\sum_{i=1}^{N} w_i \Psi_i(x)}{\sum_{i=1}^{N} \Psi_i(x)} x, \tag{4}$$

$$\Psi_i(x) = \exp\left(-h_i \left(x - c_i\right)^2\right), \tag{5}$$

defines what is known as a discrete DMP. In (2) – (5), $\alpha_z = 4\beta_z > 0$, g is the goal of the movement, x is the phase of the movement, N is the number of kernel functions, Ψ are the kernel functions, w_i are the weights of the kernel functions, h_i is their width and c_i is equally distributed on x. The details on the derivation and the use of DMPs are in [12], [13], [14]. Once we have collected movement data, we have sets of movements for each task. One set of movements consists of several recordings of the same movement.

The recordings are of different length, because humans cannot reproduce exactly the same movement for every execution of the task. Movement of different subjects also varies and besides that also depends on the marker placement. To extract the primitives of separate movements we use a generalization algorithm [11]. The algorithm can be used to generalize from different demonstrated trajectories, for example from trajectories of reaching to different points or throwing to different targets [11]. For the algorithm to work, the demonstrated trajectories have to have similar characteristics, for example, all reaching trajectories have to be straight (or all have to be curved, etc.). The result of the generalization algorithm is for such a case a straight trajectory to a given target location - query point.

For the case of generalizing amongst trajectories that all describe the same movement, we had to change the algorithm to put the same weight on all demonstration trajectories. Additionally, the target for the generalization must be representative of the movement. We chose the average final position of the joints. This final position is set as the goal g of the DMP, which encodes the generalized motion primitive. The timing of the movement primitive is also a result of the generalization.

The generalization algorithm adjusts the weights of the DMP, i.e. w_i from (3), to best fit the measured data. We need triplets of position, velocity and acceleration $\{y_d(t_j), \dot{y}_d(t_j), \ddot{y}_d(t_j)\}, j = 1, ..., T$, where t_j are the sampling times. In our case the data was obtained by motion capture and is in joint space. Equations (2) and (3) can be rewritten in a single equation by replacing z with $\tau \dot{y}$ to get

$$\tau^2 \ddot{y} + \alpha_z \tau \dot{y} - \alpha_z \beta_z (g - y) = f, \tag{6}$$

where f is defined as in (4) and (5). We thus get

$$F(t_j) = \tau^2 \ddot{y}_d(t_j) + \alpha_z \tau \dot{y}_d(t_j) - \alpha_z \beta_z (g - y_d(t_j)), \tag{7}$$

$$\mathbf{f} = \begin{bmatrix} F(t_1) \\ ... \\ F(t_T) \end{bmatrix}, \mathbf{w} = \begin{bmatrix} w_1 \\ ... \\ w_N \end{bmatrix}, \quad (8)$$

and obtain the following set of linear equations

$$\mathbf{X}\mathbf{w} = \mathbf{f}, \quad (9)$$

which needs to be solved to estimate the DMP describing the desired motion. For discrete movements, as is our case, the matrix $\mathbf{X}$ is

$$\mathbf{X} = \begin{bmatrix} \frac{\Psi_1(x_1)}{\sum_{i=1}^{N}\Psi_1(x_1)}x_1 & ... & \frac{\Psi_N(x_1)}{\sum_{i=1}^{N}\Psi_N(x_1)}x_1 \\ ... & ... & ... \\ \frac{\Psi_1(x_T)}{\sum_{i=1}^{N}\Psi_1(x_T)}x_T & ... & \frac{\Psi_N(x_T)}{\sum_{i=1}^{N}\Psi_N(x_T)}x_T \end{bmatrix}. \quad (10)$$

The parameters $\mathbf{w}$ can be calculated with the above system of differential equations in a least-squares sense. Such batch approach is common for discrete movements.

All sampled trajectory points included in the library of demonstrated movements can be used. The optimal parameters $\mathbf{w}$ can be calculated from the available training data by solving the following regression problem:

$$C(\mathbf{w}) = \sum_{k=1}^{M} ||\mathbf{X}_k\mathbf{w} - \mathbf{f}_k||^2 \quad (11)$$

with respect to $\mathbf{w}$.

Figure 2 shows motion capture and generalization results for squatting. Thin lines show the collected data. The generalized trajectories are presented with dashed lines. Marker placement has proved quite an issue, because the recorded trajectories have too much of an offset at initial conditions. We only present the results of measurements with similar initial conditions in all the trials. The generalization algorithm successfully deals with different lengths of separate demonstration movements. It is important to note that the generalization can only be successful for similar movements, so all squatting movements have to be performed in the same manner. In our case the heel had to be on the ground all the time.

III. Constraining Movement with Reflexive Behavior

In this section we show how we can constrain the recorded movement when mapping it to the robotic mechanism, in order to maintain a given criteria. In our case the criteria is the location of a simplified Zero-Moment-Point (ZMP).

A. Calculating the zero moment point

When humans do squats with the feet together, or when doing vertical jumps, large forces act on a narrow (short) support plane, thus reducing the area of stability. Stability is one of the major problems in performing movements of humanoid robots. In the case of our leg robot we are dealing with stability in the sagittal plane (forward-backward). Stability in the lateral plane (left-right) is in our case ensured by a wide foot and the robot is planar and cannot move in the lateral plane.

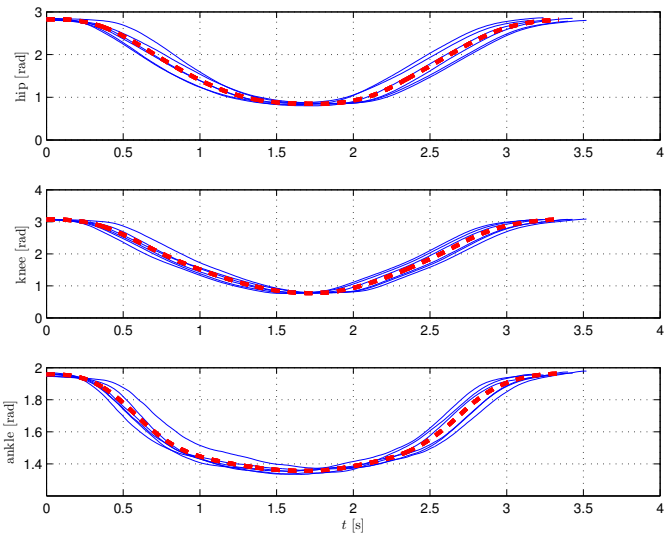

Fig. 2. Motion capture results for squatting. Recorded data is presented with solid lines and the generalized primitives are presented with dashed lines.

Zero-Moment Point (ZMP) was defined in robotics as a stability criterion by Vukobratovic and Borovac [15]. In short, it is defined as the point on the ground about which the sum of all the moments of the active forces equals zero [16].

The model of our leg robot, as shown in Fig. 1, can be represented as a triple inverted pendulum, with most of the mass in the upper segment - the body. Such structure is inherently unstable. The robot is stable when the forces on the foot are in the center of the support polygon - the foot. We denote this position as the MSP (Most Stable Point) as shown in Fig. 1. Contrary to that, the Least Stable Point (LSP) represents the point when the ZMP is on the very edge of the support polygon - either on the heels or on the tip of the toes, also depicted in Fig. 1. The more the ZMP approaches a LSP, the less stable is the robot. If the ZMP reaches any of the LSPs, the robot tips over.

To calculate the position of the ZMP of the leg robot, we need the dynamical model. We denote the i-th link with the mass m_i at the mass center point $\boldsymbol{r}_i = [x_i, y_i, z_i]^T$ (relative to the inertial frame), inertial tensor $\boldsymbol{I}_i$ and with the angular velocity $\boldsymbol{\omega}_i$. External forces and torques are denoted by $\boldsymbol{F}_{i,k}$ and $\boldsymbol{M}_{i,j}$, where index k tracks all forces and index j tracks all torques acting on i-th link. The overall rotational and translational equation of the system in an arbitrary point $\boldsymbol{p} = [x_p\ y_p\ 0]^T$ on the plain $z = 0$ is

$$\sum_i \{m_i(\boldsymbol{r}_i - \boldsymbol{p}) \times (\ddot{\boldsymbol{r}}_i + \boldsymbol{g}) + \boldsymbol{I}_i\dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times \boldsymbol{I}_i\boldsymbol{\omega}_i\} + \boldsymbol{C} = \boldsymbol{M}_p \quad (12)$$

where

$$\boldsymbol{C} = -\sum_j \boldsymbol{M}_j - \sum_k (\boldsymbol{s}_k - \boldsymbol{p}) \times \boldsymbol{F}_k. \quad (13)$$

Here $\boldsymbol{s}_k$ is the vector that points towards the position where the external force $\boldsymbol{F}_k$ acts on the robot, $\boldsymbol{g}$ is the gravity

acceleration vector (horizontal supporting surface) and $\boldsymbol{M}_p$ is the resulting torque at the observed point $\boldsymbol{p}$.

In accordance to the ZMP definition [15], only the moment $\boldsymbol{M}_p = [0\ 0\ M_z]^T$ acts at the point $\boldsymbol{p}_{zmp} = [x_{zmp}\ y_{zmp}\ 0]^T$, and by ignoring the inertial tensor $\boldsymbol{I}_i$ [17], and with no external forces acting on the robot, the components are

$$x_{zmp} = \frac{\sum_i m_i \left\{x_i \left(\ddot{z}_i + g_z\right) - z_i \left(\ddot{x}_i + g_x\right)\right\}}{\sum_i m_i \left(\ddot{z}_i + g_z\right)}, \quad (14)$$

and

$$y_{zmp} = \frac{\sum_i m_i \left\{y_i \left(\ddot{z}_i + g_z\right) - z_i \left(\ddot{y}_i + g_y\right)\right\}}{\sum_i m_i \left(\ddot{z}_i + g_z\right)}. \quad (15)$$

The relationship between the ZMP and the joint space velocity $\boldsymbol{q}$ is given by

$$u_z = \mathbf{J}_z \dot{\boldsymbol{q}}, \quad (16)$$

where

$$\mathbf{J}_z = \left[\frac{\partial y_{zmp}}{\partial q_i}\right] \quad (17)$$

and u_z is the velocity of the ZMP in general, depending on the task. In the case of the leg robot this becomes $u_z = \dot{y}_{zmp}$. The Jacobian $\mathbf{J}_z$ is a function of the actual joint angles (i.e. the joint angles set at the robot's joints at each given moment), and it essentially maps the velocity vector $\dot{\boldsymbol{q}}$ in the joint space to the velocity of the ZMP.

In the next section we show how the Jacobian inverse is employed to constrain the joint space movement as reflexive movement in order to maintain the stability of the leg robot.

B. Constraining the movement with reflexive behavior

The task of the leg robot is to do squats, where the robot joint movement is given by the extracted joint movement primitives for squatting, i.e. to perform the recorded joint movements. Performing the recorded squatting movement only makes sense if the robot maintains stability. Maintaining stability is in this case the primary task. The issue is how to map the demonstrated movements (the extracted joint primitives) to the robot in joint space, and only change them when the robot would lose stability.

The difference to a classical approach is in the control of the ZMP. The movement of the ZMP is not defined and can freely move along the supporting polygon, as long as it stays within a defined boundary. While within the boundary, the desired joint movement (from the demonstration) is completely observed. As the ZMP approaches the defined boundary, which is close to the LSP, the primary task takes over and only allows joint movement, which would not produce instability. As this movement takes command only when necessary, it acts sort of reflexive, higher-priority behavior.

To control the stability we define the stability index as a function of a normalized ZMP to the power of an odd number by

$$z_n = \left(\frac{y_{zmp} - y_{MSP}}{|y_{LSP} - y_{MSP}|}\right)^{2n-1}. \quad (18)$$

The stability control is defined with

$$\dot{y} = u_z = K_p e_z = K_p \left(z_d - z_n\right) = K_p(-z_n), \quad (19)$$

because the desired (referential) normalized ZMP location $z_d = 0$. K_p is the proportional gain. The commanded joint space velocity of the leg robot is defined with

$$\dot{\boldsymbol{q}} = \mathbf{J}^\dagger{}_z u_z + K_p \mathbf{N}'_z \dot{\boldsymbol{q}}_n \quad (20)$$

where $\dot{\boldsymbol{q}}_n = \boldsymbol{q}_n - \boldsymbol{q}$ is the velocity that tracks the desired trajectory, $\mathbf{J}^\dagger{}_z$ is the generalized inverse of (17) and matrix $\mathbf{N}'_z$ is given by

$$\mathbf{N}'_z = \mathrm{diag}(\mathbf{N}_z) + |z_n|(\mathbf{N}_z - \mathrm{diag}(\mathbf{N}_z)). \quad (21)$$

Here $\mathbf{N}_z$ is the null space matrix associated with $\mathbf{J}_z$, given by $\mathbf{N}_z = \mathbf{I} - \mathbf{J}^\dagger{}_z \mathbf{J}_z$.

When $|z_n|$ approaches 1, the zero moment point approaches the boundary of the stability region and matrix $\mathbf{N}'_z$ takes the form $\mathbf{N}'_z \approx \mathbf{N}_z$. Hence in this case the secondary task only generates movements in the null space of the stability task. We can still move any arbitrary degree of freedom, but only if that motion does not affect stability control, i. e. the motion of z_n toward the desired value $z_d = 0$. This is the classic null space control. Power n in Eq. (18) controls how close to the boundary of the support polygon we allow the ZMP to move before triggering stability control. If n is large, then z_n is close to zero over a large portion of interval $-1 \leq z_n \leq 1$, whereas a smaller n triggers stability control earlier.

On the other hand, while $|z_n|$ remains close to zero, we can allow trajectory tracking even if the resulting movement interferes with stability, since in this case the zero moment point is safely within the boundaries of the support polygon. Nevertheless, we multiply the desired velocity $\dot{\boldsymbol{q}}_n$ by $\mathbf{N}'_z \approx \mathrm{diag}(\mathbf{N}_z)$, which has the effect that the robot follows the desired trajectory more accurately with the degrees of freedom that affect stability less. To prove that this is indeed the case, we first show that coefficients of $\mathrm{diag}(\mathbf{N}_z)$ are always nonnegative. This is due to the properties of *Moore-Penrose pseudoinverse* [18], which make $\mathbf{J}_z\mathbf{J}^\dagger{}_z$ an idempotent matrix, i. e. $(\mathbf{J}_z\mathbf{J}^\dagger{}_z)^2 = \mathbf{J}_z\mathbf{J}^\dagger{}_z$. It follows that

$$\begin{aligned}\mathbf{N}_z^2 &= (\mathbf{I} - \mathbf{J}_z\mathbf{J}^\dagger{}_z)^2 = \mathbf{I} - \mathbf{J}_z\mathbf{J}^\dagger{}_z - \mathbf{J}_z\mathbf{J}^\dagger{}_z + \left(\mathbf{J}_z\mathbf{J}^\dagger{}_z\right)^2 = \\ &= \mathbf{I} - \mathbf{J}_z\mathbf{J}^\dagger{}_z = \mathbf{N}_z, \end{aligned} \quad (22)$$

thus $\mathbf{N}_z$ is idempotent, too. Since $\mathbf{N}_z$ is also symmetric, the diagonal elements of $\mathbf{N}_z$, denoted here by a_{ii}, are given by

$$a_{ii} = a_{ii}^2 + \sum_{j=1, j\neq i}^{n} a_{ji}^2 \quad \geq \quad 0. \quad (23)$$

As explained above, $\mathbf{J}_z\mathbf{J}^\dagger{}_z$ is also an idempotent, symmetric matrix, thus its diagonal elements $b_{ii} \geq 0, \forall i$, too. Consequently, $a_{ii} = 1 - b_{ii} \leq 1$ and in summary, $0 \leq a_{ii} \leq 1$.

If $a_{ii} = 0$, then it follows from Eq. (23) that $a_{ji} = 0, \forall j$. In this case we have $\mathbf{N}_z \mathbf{e}_i = 0$, where $\mathbf{e}_i$ is the i-th coordinate vector, i. e. $\mathbf{e}_i = [0, \ldots, 0, 1, 0, \ldots, 0]^T$. This means that $\mathbf{e}_i$ is orthogonal to the null space of $\mathbf{J}_z$, thus any motion in the i-th degree of freedom directs the robot straight out of the stability polygon. For this reason we suppress the movement in the i-th degree of freedom if the i-th diagonal element of $\mathbf{N}_z$ becomes small. Note that the suppression is only temporary; if the difference between the desired movement and the actual execution becomes large, then the desired change $\dot{\boldsymbol{q}}_n = \boldsymbol{q}_n - \boldsymbol{q}$ in (20) becomes large and the commanded motion starts tracking the desired motion regardless as long as the ZMP remains within the stability polygon.

On the other hand, if $a_{ii} = 1$, then it follows from Eq. (23) that $a_{ji} = 0, \ i \neq j$. In this case we have $\mathbf{N}_z \mathbf{e}_i = \mathbf{e}_i$, which means that $\mathbf{e}_i$ belongs to the null space of $\mathbf{J}_z$. Consequently, motion in the i-th degree of freedom does not affect the stability of the robot at all. It therefore makes sense to allow the robot to freely follow the trajectory for such degrees of freedom. In summary, the multiplication by $\mathrm{diag}(\mathbf{N}_z)$ is equivalent to weighting the degrees of freedom with a factor that is inversely proportional to the amount to which the motion in each degree of freedom interferes with the primary task. Since the elements of $\mathrm{diag}(\mathbf{N}_z)$ are non-negative, the resulting controller always causes the robot to move in the right direction, but slows down the motion if it significantly affects stability control.

IV. Experimental evaluation

A. Leg robot

We developed a leg robot, which was inspired by the anatomic properties or the human body. The robot is presented in Fig. 1. It is a planar robot composed of four segments which represent the foot, shank, thigh and trunk of a human. The segments are connected together by rotational hinge joints whose axes are perpendicular to the sagittal plane. Each joint is activated by a servo drive mounted inside the proximal segment with regard to the joint. We use Maxon RE 40 DC servo drives. The gear ratio we chose is $1 : 8$, which allows high torques, but squatting overloads the motors, so only a few squats can be performed at a time. The largest part of the robot weight is in the trunk segment where are the motor that activates the hip joint, the computer, motion controller and all power amplifiers. Altogether, the robot weights $\sim$4.6 kg. The robot is connected to an external power supply.

B. Squatting movement

We performed squatting movement with our leg robot. The desired joint movement is given with movement primitives, extracted from human demonstrations. One-to-one mapping of the movement to the robot, accounting for the offsets to compensate the initial position of the robot, results in the robot tipping over. This is clearly shown in Fig. 4, where we can see that unconstrained movement (red), moves the ZMP out of the area of stability. Reflexive movement (blue), on the other hand, successfully maintains stability. Fig. 3 shows

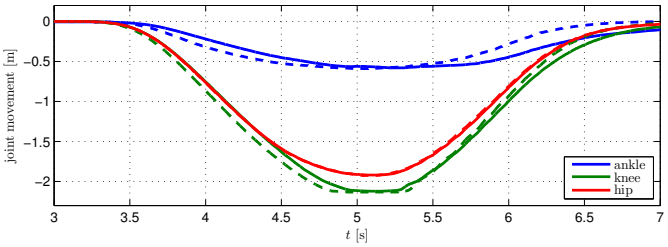

Fig. 3. Joint movement during successful (solid) and unsuccessful (dashed) squatting. The difference in movement is small, but critical. In comparison to the extracted primitives in Fig. 2, we had to add offsets to account for initial position of the leg robot.

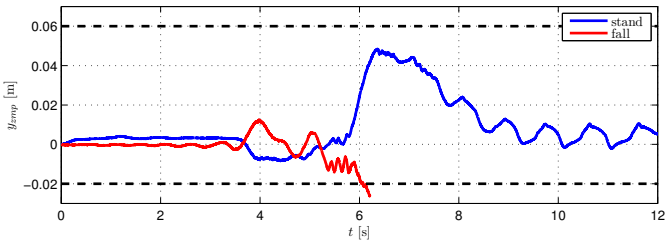

Fig. 4. ZMP location, calculated from the model, during successful (blue) and unsuccessful (red) squatting. The dashed lines give the LSPs.

constrained (solid) and unconstrained (dashed) joint movement of the robot. We can see that the trajectories are very similar. Fig. 5 shows a sequence for successful squatting with reflexive behavior in the top row, and unsuccessful squatting from simple one-to-one mapping in the bottom row.

Fig. 6 shows the comparison of the ZMP movement, calculated from the model, and the projection of the center of gravity on the floor, measured with a Kistler force place, for three consecutive squats. We can see that the model provides an accurate stability criterion.

V. Conclusion

In the paper we have shown how we can use a modified prioritized task control to implement reflexive movement of a robot. Higher priority movement only takes over when the desired movement approaches a given threshold, and thus acts as reflexive movement. We have shown how to apply the algorithm to a leg robot performing squatting. The algorithm can constrain the demonstrated movement to maintain stability. In the future we would like to implement the algorithm in the lowest level of control of our mechanism, and try similar reflexive behavior in other scenarios. We have also shown how we can effectively apply a generalization algorithm to extract movement primitives from sets of recorded movements.

Acknowledgment

This paper was partially funded by Slovenian Research Agency grant Z2-3672. It has also received funding from the European Communitys Seventh Framework Programme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under grant agreement no. 269959, IntellAct.

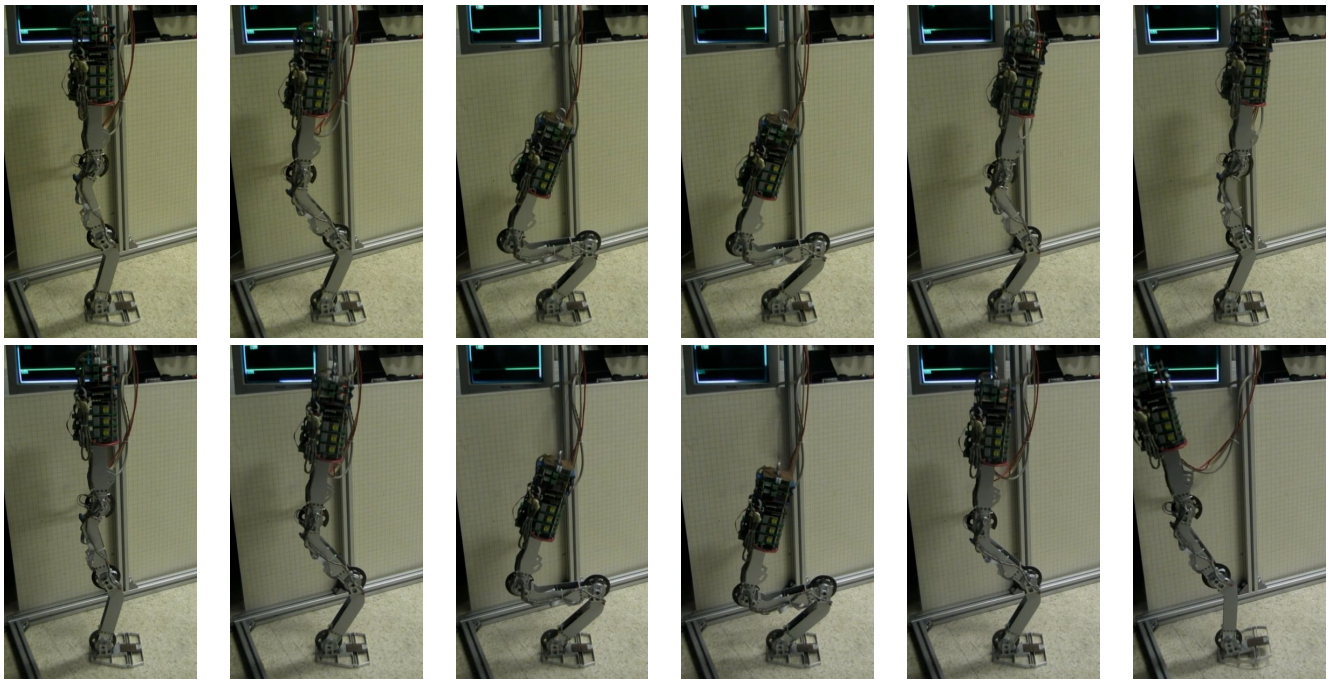

Fig. 5. A sequence of still photos showing squatting movement. The top row show successful execution of the movement using reflexive movement. The bottom row show execution of unconstrained demonstrated movement.

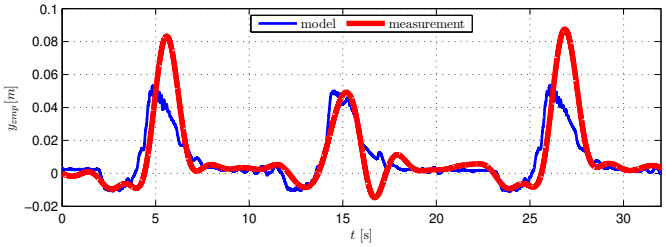

Fig. 6. Movement of the ZMP, calculated from the model (blue), and smoothed measured projection of the center of gravity on the floor, measured with a Kistler force plate. We can see resemblance in the shape of the two signals.

REFERENCES

[1] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in cognitive sciences*, vol. 6, pp. 233–242, 1999.

[2] A. Ude, C. Atkeson, and R. M., "Programming full-body movements for humanoid robots by observation," *Robotics and Autonomous Systems*, vol. 47, no. 2-3, pp. 93–108, 2004.

[3] L. Sentis, J. Park, and O. Khatib, "Modeling and control of multi-contact centers of pressure and internal forces in humanoid robots," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, oct. 2009, pp. 453 –460.

[4] L. Žlajpah and B. Nemec, "Kinematic control algorithms for on-line obstacle avoidance for redundant manipulators," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 2, 2002, pp. 1898 – 1903 vol.2.

[5] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *Robotics and Automation, IEEE Journal of*, vol. 3, no. 1, pp. 43 –53, february 1987.

[6] S. Grillner, "Locomotion in vertebrates: central mechanisms and reflex interaction," *Physiol Rev*, vol. 55, pp. 247 – 304, 1975.

[7] J. Babic, J. G. Hale, and E. Oztop., "Human sensorimotor learning for humanoid robot skill synthesis," *Adaptive Behavior*, 2011.

[8] J. Babic, L. Bokman, D. Omrcen, J. Lenarcic, and F. Park, "A biarticulated robotic leg for jumping movements : theory and experiments," *Journal of mechanisms and robotics*, vol. 1, no. 1, 2009.

[9] H. Sugiura, M. Gienger, H. Janssen, and C. Goerick, "Real-time collision avoidance with whole body motion control for humanoid robots," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, 29 2007-nov. 2 2007, pp. 2053 –2058.

[10] N. Mansard, O. Khatib, and A. Kheddar, "A unified approach to integrate unilateral constraints in the stack of tasks," *Robotics, IEEE Transactions on*, vol. 25, no. 3, pp. 670 –685, june 2009.

[11] A. Ude, A. Gams, T. Asfour, and J. Morimoto, "Task-specific generalization of discrete and periodic dynamic movement primitives," *Robotics, IEEE Transactions on*, vol. 26, no. 5, pp. 800 –815, oct. 2010.

[12] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proc. IEEE Int. Conf. Robotics and Automation*, Washington, DC, 2002, pp. 1398–1403.

[13] S. Schaal, P. Mohajerian, and I. A., "Dynamics systems vs. optimal control – a unifying view," *Progress in Brain Research*, vol. 165, no. 6, pp. 425–445, 2007.

[14] A. Gams, A. Ijspeert, S. Schaal, and J. Lenarčič, "On-line learning and modulation of periodic movements with nonlinear dynamical systems," *Auton. Robots*, vol. 27, no. 1, pp. 3–23, 2009.

[15] M. Vukobratovic and B. Borovac, "Zero-moment point - thirty five years of its life," *I. J. Humanoid Robotics*, vol. 1, no. 1, pp. 157–173, 2004.

[16] Q. Huang, K. Yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi, and K. Tanie, "Planning walking patterns for a biped robot," *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 3, pp. 280 –289, jun 2001.

[17] T. Furuta, T. Tawara, Y. Okumura, M. Shimizu, and K. Tomiyama, "Design and construction of a series of compact humanoid robots and development of biped walk control strategies," *Robotics and Autonomous Systems*, vol. 37, pp. 81–100(20), 30 November 2001.

[18] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications*. New York: John Wiley & Sons, 1974.