2013 IEEE International Conference on Robotics and Automation (ICRA)
Karlsruhe, Germany, May 6-10, 2013

Motion Capture and Reinforcement Learning of Dynamically Stable Humanoid Movement Primitives

Rok Vuga[1], Matjaž Ogrinc[1], Andrej Gams[1,3], Tadej Petrič[1], Norikazu Sugimoto[4], Aleš Ude[1,2], and Jun Morimoto[2]

Abstract— **Direct transfer of human motion trajectories to humanoid robots does not result in dynamically stable robot movements due to the differences in human and humanoid robot kinematics and dynamics. We developed a system that converts human movements captured by a low-cost RGB-D camera into dynamically stable humanoid movements. The transfer of human movements occurs in real-time. As need arises, the developed system can smoothly transition between unconstrained movement imitation and imitation with balance control, where movement reproduction occurs in the null space of the balance controller. The developed balance controller is based on an approximate model of the robot dynamics, which is sufficient to stabilize the robot during on-line imitation. However, the resulting movements cannot be guaranteed to be optimal because the model of the robot dynamics is not exact. The initially acquired movement is therefore subsequently improved by model-free reinforcement learning, both with respect to the accuracy of reproduction and balance control. We present experimental results in simulation and on a real humanoid robot.**

I. INTRODUCTION

Motion capture has proven to be an effective way to acquire humanoid trajectories since many years [12], [22]. However, the problem of transferring human motion to humanoid robots becomes much more difficult if the observed human motion should result in dynamically stable humanoid robot movements. Since the human and the humanoid robot kinematics and dynamics are not the same, a copy of human trajectories usually results in dynamically unstable humanoid robot motion. 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.

Stability of humanoid trajectories is usually ensured by controlling the robot's zero moment point (ZMP) [23], which is defined as the point on the ground where the tipping moment acting on the humanoid robot, due to gravity and inertia forces, equals zero [13]. The tipping moment is defined as the component of the moment which is tangential to the ground surface. A biped humanoid robot is dynamically stable at any given time if its ZMP lies within the area defined by the convex hull of one (single support phase) or two (double support phase) supporting feet.

A ZMP compensation filter was developed to enable the stabilization of walking trajectories [8] and the imitation of dancing movements [7]. In both cases the stability of motion was achieved by modifying the horizontal torso trajectory. Kajita et al. [4] designed a control system which minimizes the error between the desired ZMP and the output ZMP by applying a preview controller. Sugihara et al. [20] applied the 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. The real-time transfer of human motion while maintaining balance was studied in [9], [24], but unlike the system described in this paper, these authors used marker-based trackers, which are inherently more precise, and did not utilize prioritized control and model-free reinforcement learning (see below) to improve the transferred movements. Another alternative are off-line, optimization based approaches to reshape human motion, as described for example in [11], who also used a marker-based system for motion acquisition.

While a lot of previous research on stability of humanoid robots was concerned with walking, our first major goal is to integrate balance control with motion capture systems to generate dynamically stable reproductions of human movements in real-time. We propose to apply whole-body prioritized control for this purpose. In the context of humanoid robots, prioritized control was used for example to enable the unified control of centre of mass, operation-space tasks, and internal forces [16]. Prioritized control for locomotion and balance control was also addressed in [6].

Center of pressure (CoP) is defined as [2]

$$\mathbf{x}_{\mathrm{CoP}} = \frac{\int_S \mathbf{x} \mathrm{F}_z(\mathbf{x}) d\mathbf{x}}{\int_S \mathrm{F}_z(\mathbf{x}) d\mathbf{x}}, \tag{1}$$

where F_z is the component of the contact force normal to the sole(s). CoP and ZMP coincide while the dynamic balance is being preserved [23], i. e. as long as they are located within the support polygon. Many humanoid robots (including the two robots used in our experiments) are equipped with pressure sensors (typically four) in each foot. These can be used to estimate $\mathbf{x}_{\mathrm{CoP}}$, or equivalently $\mathbf{x}_{\mathrm{ZMP}}$, even when a model of the robot's dynamics is not available. Exploiting

[1]R. Vuga, M. Ogrinc, A. Gams, T. Petrič, and A. Ude are with the Laboratory of Humanoid and Cognitive Robotics, Department of Automatics, Biocybernetics and Robotics, Jožef Stefan Institute, Ljubljana, Slovenia. rok.vuga at ijs.si, matjaz.ogrinc42 at gmail.com, andrej.gams at ijs.si, tadej.petric at ijs.si, ales.ude at ijs.si

[2]A. Ude, and J. Morimoto are with the Dept. of Brain Robot Interface, ATR Computational Neuroscience Laboratories, Kyoto, Japan. R. Vuga worked at ATR in summer 2012. xmorimo at atr.jp

[3]A. Gams is with Biorobotics Laboratory, École Politechnique Fédérale de Lausanne, Switzerland.

[4]N. Sugimoto is with National Institute of Information and Communications Technology, Japan. xsugi at nict.go.jp

this information, model-free reinforcement learning methods can be applied to improve the initial reproduction of human motion on a humanoid robot. It is important that the initial movement is dynamically stable because in practice, there is little hope that model-free methods would find dynamically stable movements due to the dimensionality of the search space. The second major goal of this paper is to show that reinforcement learning can be used to improve both stability and fidelity of the reproduced motion.

II. MOTION CAPTURE WITH BALANCE CONTROL

The spread of low-cost RGB-D cameras like Kinect and skeleton trackers based on these cameras has contributed to a significant improvement of markerless human motion capture in recent years [18], [21]. Rough, video rate (30 Hz) body trackers are now generally available and can be used to reconstruct and transfer human motion to humanoid robots. Their output are typically the positions and orientations of body parts including torso, head, lower and upper arms and lower and upper legs. To transfer this motion to a humanoid robots that consist of sequential, rotational joints, it is only necessary to transform the relative positions and orientations of successive body parts into the appropriate sequences of Euler angles [17]. We do not reproduce the formulas here because they depend on the actual humanoid robot. An example reproduction of human motion captured by Kinect and reproduced by a 38 degrees of freedom humanoid robot CB-i [1] is shown in Fig. 1.

To ensure the dynamic stability of a humanoid robot, we need to control its motion so that ZMP stays within the support polygon during the reproduction of the observed human motion. Neglecting the inertia matrices, the following relation can be obtained between the center of gravity (CoG)

$$\mathbf{x}_{\mathrm{CoG}} = \frac{\sum_{i=0}^{N} m_i \mathbf{x}_i}{\sum_{i=0}^{N} m_i}, \tag{2}$$

and the zero moment point $\mathbf{x}_{\mathrm{ZMP}}$ [20]

$$\ddot{x}_{\mathrm{CoG}} = \omega^2 \left(x_{\mathrm{CoG}} - x_{\mathrm{ZMP}}\right), \tag{3}$$

$$\ddot{y}_{\mathrm{CoG}} = \omega^2 \left(y_{\mathrm{CoG}} - y_{\mathrm{ZMP}}\right), \tag{4}$$

where

$$\omega = \sqrt{\frac{\ddot{z}_{\mathrm{CoG}} + g}{z_{\mathrm{CoG}} - z_{\mathrm{ZMP}}}}, \tag{5}$$

g is the gravity constant, $\mathbf{x}_i$ is the position of the center of mass of robot body part i, m_i its mass, N is the number of degrees of freedom, and z_{ZMP} the height of the ground surface. Thus to compute the desired motion of $\mathbf{x}_{\mathrm{COG}}$ from $\mathbf{x}_{\mathrm{ZMP}}$, we have two equations ((3) and (4)) but they contain three unknowns because ω depends on z_{CoG}. To resolve this ambiguity, the desired z_{CoG} is first determined independently using an inverted pendulum controller. See [20] for details. To keep the dynamic stability, the desired x_{ZMP} and y_{ZMP} should be moved towards the centre of the support polygon, using for example a simple P-controller

$${}^{\mathrm{des}}\dot{x}_{\mathrm{ZMP}} = \gamma_x \left(x_{\mathrm{P}} - x_{\mathrm{ZMP}}\right), \tag{6}$$

$${}^{\mathrm{des}}\dot{y}_{\mathrm{ZMP}} = \gamma_y \left(y_{\mathrm{P}} - y_{\mathrm{ZMP}}\right), \tag{7}$$

where $\mathbf{x}_{\mathrm{P}}$ is the center of the support polygon. The desired motion of $\mathbf{x}_{\mathrm{CoG}}$ can then be calculated by integrating Eq. (3) and (4).

In the following we denote by superscript ^{0} all entities given in robot base coordinates, whereas the entities in world coordinates are written without the superscript. The relationship between the velocity of the centre of gravity in base coordinates ${}^0\dot{\mathbf{x}}_{\mathrm{CoG}}$ and joint angle velocity is given by the Jacobian of the center of gravity ${}^0\mathbf{J}_{\mathrm{CoG}} \in \mathbb{R}^{3\times N}$, which is obtained from Eq. (2) as

$${}^0\dot{\mathbf{x}}_{\mathrm{CoG}} = \frac{\sum_{i=1}^{N} m_i {}^0\mathbf{J}_i \dot{\theta}}{\sum_{i=0}^{N} m_i} = \frac{\sum_{i=1}^{N} m_i {}^0\mathbf{J}_i}{\sum_{i=0}^{N} m_i}\dot{\theta} = {}^0\mathbf{J}_{\mathrm{CoG}}\dot{\theta}, \tag{8}$$

where ${}^0\mathbf{J}_i$ is the geometric Jacobian of the centre of mass of body part i in base coordinates. This relationship, however, does not take into account that one or two support feet are fixed in the world coordinate systems, i. e. $\dot{\mathbf{x}}_{\mathrm{L}} = \omega_{\mathrm{L}} = 0$ and $\dot{\mathbf{x}}_{\mathrm{R}} = \omega_{\mathrm{R}} = 0$, where $\dot{\mathbf{x}}_{\mathrm{L,R}}$ and $\omega_{\mathrm{L,R}}$ are the linear and angular velocities of both feet. Sugihara et al. [20] have proven that if the left or right foot is assumed to be the main support foot (F = R or L), which does not move in world coordinates, then the Jacobian of the centre of gravity in world coordinates can be calculated as

$$\mathbf{J}_{\mathrm{CoG}} = \mathbf{R}\left({}^0\mathbf{J}_{\mathrm{CoG}} - {}^0\mathbf{J}_{\mathrm{F}} + \mathbf{\Omega}\left({}^0\mathbf{x}_{\mathrm{CoG}} - {}^0\mathbf{x}_{\mathrm{F}}\right){}^0\mathbf{J}_{\omega\mathrm{F}}\right), \tag{9}$$

where

$$\mathbf{\Omega}(\mathbf{x}) = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix},$$

$\mathbf{R}$ is the orientation of the robot base in world coordinates, ${}^0\mathbf{J}_{\mathrm{F}} \in \mathbb{R}^{3\times N}$ and ${}^0\mathbf{J}_{\omega\mathrm{F}} \in \mathbb{R}^{3\times N}$ are respectively the translational and orientational part of the Jacobian of the foot, and ${}^0\mathbf{x}_{\mathrm{F}}$ is the position of the foot, all in robot base coordinates. In double support case when both feet are on the ground we can take for example F = L in Eq. (9) and add the constraint

$$\mathbf{J}_{\mathrm{R}}\dot{\theta} = 0, \tag{10}$$

where $\mathbf{J}_{\mathrm{R}} \in \mathbb{R}^{6\times N}$ is the Jacobian of the right foot in world coordinates.

Thus the relationship between the desired center of gravity velocity and joint angle velocities under constraint that the support foot or feet do not move can always be expressed as

$$\dot{\tilde{\mathbf{x}}} = \tilde{\mathbf{J}}\dot{\theta}. \tag{11}$$

In the double support phase we have

$$\dot{\tilde{\mathbf{x}}} = \begin{bmatrix} \dot{\mathbf{x}}_{\mathrm{CoG}} \\ 0 \end{bmatrix}, \quad \tilde{\mathbf{J}} = \begin{bmatrix} \mathbf{J}_{\mathrm{CoG}} \\ \mathbf{J}_{\mathrm{R}} \end{bmatrix}, \tag{12}$$

whereas in the single support phase we simply take $\dot{\tilde{\mathbf{x}}} = \dot{\mathbf{x}}_{\mathrm{CoG}}$ and $\tilde{\mathbf{J}} = \mathbf{J}_{\mathrm{CoG}}$.

We can now formulate dynamically stable reproduction of human movements using prioritized control. The standard approach is to define stability as primary task and movement reproduction as secondary task. This leads to the following control policy

$$\dot{\theta} = \tilde{\mathbf{J}}^{+}\dot{\tilde{\mathbf{x}}} + \mathbf{N}\dot{\theta}_{\mathrm{K}}, \tag{13}$$

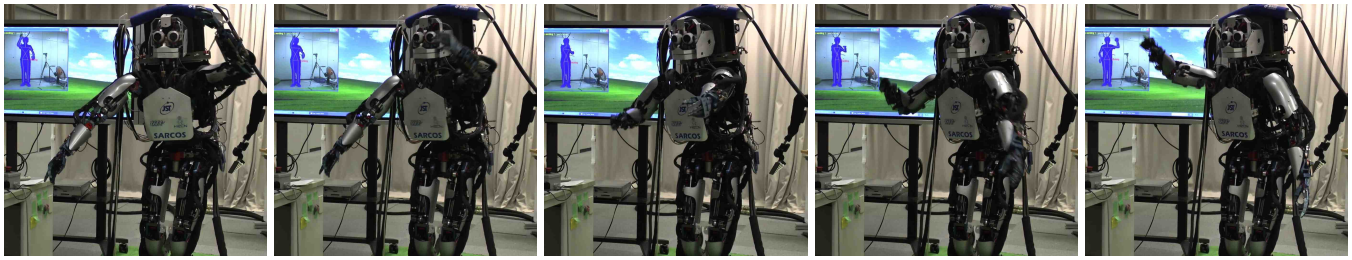

Fig. 1. Real-time transfer of human motion observed by RGB-D camera. Human figure demonstrating the motion on the screen is mirrored; the robot uses the same arm as the human.

where $\mathbf{N} = \left(\mathbf{I} - \tilde{\mathbf{J}}^{+}\tilde{\mathbf{J}}\right)$ is the null space matrix of $\tilde{\mathbf{J}}$ and θ_{K} are the joint angles estimated by observing human motion. Joint angles that are not estimated by the body tracker are set to fixed values. Note that if some of the degrees of freedom should be controlled independently without considering the stability of the robot, e. g. the non-supporting leg in the single support case, then the columns corresponding to the respective degrees of freedom should simply be excluded from the above matrices. It is the task of the remaining degrees of freedom to ensure stability.

III. SMOOTH TRANSITION BETWEEN UNCONSTRAINED AND NULL-SPACE IMITATION

The movement of the lower body is quite constrained by the condition that one or both feet must remain motionless on the ground surface. The upper body motion, however, is much less constrained as long as the ZMP is well within the bounds of the support polygon. This is especially true in the double support phase. Thus in the double support phase we allow the upper body to move freely until the ZMP starts approaching the edge of the support polygon. We divide the control problem into two parts: the lower body control that includes the movement of all leg degrees of freedom and always follows the scheme of Section II, and the upper body control that includes torso, head and arms motion, where we can allow the robot to follow the observed human motion more closely.

The proposed control scheme for upper body is based on the idea of smooth transition between tasks as developed in [10]. Since the upper body motion does not cause the feet to move – at least as long as the robot remains stable – the relationship between the upper body joint velocities and $\dot{\mathbf{x}}_{\mathrm{CoG}}$ can be expressed simply as $\dot{\mathbf{x}}_{\mathrm{CoG}} = \mathbf{J}_{\mathrm{CoG}}\dot{\theta}$, where $\mathbf{J}_{\mathrm{CoG}}$ is the Jacobian of the centre of gravity in world coordinates, with pelvis as the base link at the beginning of the kinematic chain. We now define a controller that smoothly transitions between unconstrained upper-body movement reproduction and reproduction in the null space of stability controller

$$\dot{\theta} = \lambda(\mathbf{x}_{\mathrm{ZMP}})^{n}\mathbf{J}^{+}_{\mathrm{CoG}}\dot{\mathbf{x}}_{\mathrm{CoG}} + \mathbf{N}_{\lambda}\dot{\theta}_{K}, \tag{14}$$

where

$$\mathbf{N}_{\lambda} = (1 - \lambda(\mathbf{x}_{\mathrm{ZMP}})^{n})\,\mathrm{diag}(\mathbf{N}) + \lambda(\mathbf{x}_{\mathrm{ZMP}})^{n}\mathbf{N}. \tag{15}$$

and $\mathbf{N} = \mathbf{I} - \mathbf{J}^{+}_{\mathrm{CoG}}\mathbf{J}_{\mathrm{CoG}}$.

The transition between unconstrained imitation and imitation in the null space of stability controller is regulated through the weighting function $\lambda(\mathbf{x}_{\mathrm{ZMP}})$. Let $d(\mathbf{x})$ be the distance of the point within the support polygon to the boundary of the polygon. We denote by $d_{\min}$ the distance to the boundary of the support polygon where the robot should start imitating the motion in the null space of stability controller only. Then we can define

$$\lambda\left(\mathbf{x}\right) = \begin{cases} \dfrac{d(\mathbf{x}_{\mathrm{P}}) - d(\mathbf{x})}{d(\mathbf{x}_{\mathrm{P}}) - d_{\min}}, & d(\mathbf{x}) > d_{\min} \\ 1, \text{ otherwise} \end{cases}, \tag{16}$$

where $\mathbf{x}_{\mathrm{P}}$ is the center of the support polygon. Since by definition $d(\mathbf{x}) \leq d(\mathbf{x}_{\mathrm{P}})$ for all $\mathbf{x}$ within the support polygon, we have $0 \leq \lambda(\mathbf{x}_{\mathrm{ZMP}}) \leq 1$. Exponent n of Eq. (15) controls how quickly the weighting function $\lambda(\mathbf{x})^{n}$ drops to zero as the distance of ZMP to the boundary of the support polygon increases. In our experiments we used $n = 3$. Matrix $\mathbf{N}_{\lambda}$ is equal to the null space matrix $\mathbf{N}$ when ZMP is close to the boundary of the support polygon. It becomes approximately equal to $\mathrm{diag}(\mathbf{N})$ as ZMP moves to the centre ($\mathbf{x}_{\mathrm{P}}$) of the support polygon. In the following we show why it is appropriate to set $\mathbf{N}_{\lambda} \approx \mathrm{diag}(\mathbf{N})$ when the robot is stable, i e. when ZMP is close to the center of the support polygon.

Any null space matrix $\mathbf{N} = \mathbf{I} - \mathbf{J}^{+}\mathbf{J}$ is symmetric and idempotent, i. e. $\mathbf{N}^{2} = \mathbf{N}$. Consequently

$$n_{ii} = \sum_{j=1}^{N} n_{ij}^{2} \geq 0. \tag{17}$$

By definition, a pseudoinverse of $\mathbf{J}$ satisfies the condition $\mathbf{J}\mathbf{J}^{+}\mathbf{J} = \mathbf{J}$. It follows that $(\mathbf{J}^{+}\mathbf{J})^{2} = \mathbf{J}^{+}(\mathbf{J}\mathbf{J}^{+}\mathbf{J}) = \mathbf{J}^{+}\mathbf{J}$, thus $\mathbf{J}^{+}\mathbf{J}$ is also a symmetric, idempotent matrix. Let c_{ij} be the coefficients of this matrix. As it was shown above, the diagonal elements of symmetric, idempotent matrix are nonnegative, i. e. $c_{ii} \geq 0$. Since $\mathbf{N} = \mathbf{I} - \mathbf{J}^{+}\mathbf{J}$, we can write

$$n_{ii} = 1 - c_{ii} \leq 1. \tag{18}$$

Combining Eq. (17) and (18) shows that the diagonal elements of $\mathbf{N}$ are between 0 and 1.

Let us assume that $n_{ii} = 1$. It follows from Eq. (17) and (18) that in this case $n_{ij} = 0, \forall j \neq i$. This means that 1 is the eigenvalue of $\mathbf{N}$ with the associated eigenvector $\mathbf{e}_{i}$, i. e. $\mathbf{N}\mathbf{e}_{i} = \mathbf{e}_{i}$, where $\mathbf{e}_{i}$ is the unit vector along the i-th coordinate axis. Thus in this case the motion caused

by the i-th degree of freedom lies in the null space of $\mathbf{J}$. Any movement performed by this degree of freedom will therefore not cause any troubles to the stability of the robot. On the other hand, if $n_{ii} = 0$, then it follows from Eq. (17) that $n_{ji} = n_{ij} = 0, \forall j$. Thus in this case $\mathbf{N}\mathbf{e}_i = 0$, which means that $\mathbf{e}_i$ is orthogonal to the null space of $\mathbf{J}$. Any motion caused by the i-th degree of freedom pulls the ZMP directly towards the edge of the support polygon.

We have thus proven that it is reasonable to scale the transferred human motion by the diagonal elements of $\mathbf{N}$. Such scaling causes the reproduction to slow down if it would make the ZMP to move towards the boundary of the support polygon, thus helping the robot to avoid sudden movements that could disturb its balance even before full null-space control is triggered. Since we use feedback control to reproduce the observed human motion, this slow-down effect is only temporary and prevents the ZMP to move too quickly towards the edge of the support polygon. If the difference between the human and the robot motion continues to increase, this increase compensates for the reduction in the gain factor and the robot continues to track the movement. If the ZMP keeps moving towards the edge of the support polygon, the overall controller (14) switches to the standard prioritized control to ensure stability.

Note that the diagonal matrix $\mathrm{diag}(\mathbf{N})$ in Eq. (15) can be replaced by $\mathbf{I}$, resulting in

$$\mathbf{N}_\lambda = (1 - \lambda(\mathbf{x}_{\mathrm{ZMP}})^n)\,\mathbf{I} + \lambda(\mathbf{x}_{\mathrm{ZMP}})^n\mathbf{N}, \qquad (19)$$

which is equivalent to the method proposed in [10]. With this method, the matrix $\mathbf{N}_\lambda$ smoothly transitions between free mimicking (characterized by $\mathbf{N}_0 = \mathbf{I}$) and mimicking in the null space of the stability controller (characterized by $\mathbf{N}_1 = \mathbf{N}$) without the scaling effects caused by the diagonal elements of $\mathbf{N}$.

IV. Improving Motion Transfer by Reinforcement Learning

As noted in the introduction, the location of the ZMP can be estimated by using pressure sensors in the feet, even when a model of the robot's dynamics is not available. Hence we use pressure sensors to compute the ZMP during on-line reproduction of the demonstrated human motion. However, any balance controller, including the one described in Section II, is based on a (approximate) model of the robot's dynamics. While the controller of Section II is successful at generating dynamically stable robot movements from the observed human movements, the resulting motion is not optimal. We propose to improve it by reinforcement learning, which does not require that a model of the robot's dynamics is known in any form.

To apply reinforcement learning, we encode the initially transferred motion with a suitable formal representation system. We chose dynamic movement primitives (DMPs) developed by Ijspeert et al. [3], [15]. With DMPs, the motion of every degree of freedom y is calculated by integrating the equation system

$$\tau\dot{z} = \alpha_z(\beta_z(g - y) - z) + f(x), \qquad (20)$$
$$\tau\dot{y} = z, \qquad (21)$$

where

$$f(x) = \frac{\sum_{i=1}^{M} w_i\Psi_i(x)}{\sum_{i=1}^{M}\Psi_i(x)}x,\ \Psi_i(x) = \exp\left(-h_i\,(x - c_i)^2\right). \qquad (22)$$

Here c_i are the centers of radial basis function distributed along the trajectory and $h_i > 0$. If the constants $\alpha_z,\ \beta_z,\ \tau > 0$ are set appropriately, e. g. $\alpha_z = 4\beta_z$, this system has a unique attractor point at $y = g,\ z = 0$. A phase variable x is used in Eq. (22) instead of time to make the dependency of f on time more implicit. Its dynamics is also defined by a differential equation

$$\tau\dot{x} = -\alpha_x x. \qquad (23)$$

Like time, the phase variable x, $x(0) = 1$, must be common across all the degrees of freedom. Parameters that should be calculated from a human demonstration and the resulting robot motion include the weights w_i, the goal parameter g, and the time constant τ. In our experiments we set τ to be equal to the duration of movement. The goal parameter g is different for every degree of freedom and is set to the final human joint configuration as estimated by Kinect. The weights w_i are also determined separately for every degree of freedom. We use locally weighted regression [15] to compute w_i so that the resulting DMP encodes the joint angle trajectories as executed by the robot during on-line reproduction, which – unlike the original human motion – is dynamically stable on the robot.

While g is already equal to the desired final configuration and therefore does not need to be improved, the parameters w_i are optimal neither with respect to the stability of motion nor with respect to the fidelity of reproduction. Previous research has shown that probabilistic reinforcement learning algorithms can be employed to learn full-body movement primitives of high degree of freedom humanoid robots [19]. Here we propose to apply reinforcement learning to the problem of simultaneous imitation and balance control. In our experiments we utilized the expectation-maximization based policy learning by weighting exploration with the returns algorithm (PoWER) [5]. An important advantage of probabilistic reinforcement learning algorithms such as PoWER is that they can make use of initial approximations for the desired robot motion, which can be obtained from user demonstrations. Moreover, the amount of exploration, i. e. how much different the trial trajectories may be compared to the initial movement, can be controlled by setting the only free parameter of the algorithm, i. e. the exploration noise. This way the robot can be prevented from trying to perform physically infeasible movements, which can lead to damage.

To bring the transferred robot movement closer to the observed human movement, the reward function for each trial trajectory ν should evaluate the distance of the transferred

Fig. 2. Real-time transfer of a walking movement.

trajectory (encoded by a DMP) to the human movement as estimated by Kinect in joint coordinates, i. e. $\{\mathbf{q}_{\mathrm{KIN}}(t_2, i)\}$, where $t_{2,i}$ are the measurement times of the Kinect system. In addition, the transferred movement should be as dynamically stable as possible. Thus we compute also the distance of ZMP, or equivalently the center of pressure CoP, to the center of the support polygon. This results in the following reward function

$$r(\nu) = \frac{\gamma}{1 + a\Delta_{\mathrm{ZMP}}(\nu)} + \frac{1-\gamma}{1 + b\Delta_{\mathrm{KIN}}(\nu)}, \qquad (24)$$

where ν denotes the trajectory, $0 \leq \gamma \leq 1$, $a,\ b > 0$,

$$\Delta_{\mathrm{ZMP}}(\nu) = \frac{1}{n_1}\sum_{i=1}^{n_1} \|\mathbf{x}_{\mathrm{ZMP}}(t_{1,i}) - \mathbf{x}_{\mathrm{P}}\|^2, \qquad (25)$$

and

$$\Delta_{\mathrm{KIN}}(\tau) = \frac{1}{n_2}\sum_{i=1}^{n_2} \|\mathbf{q}(t_{2,i}) - \mathbf{q}_{\mathrm{KIN}}(t_{2,i})\|^2. \qquad (26)$$

$\mathbf{q}(t_{2,i})$ and $\mathbf{x}_{\mathrm{ZMP}}(t_{1,i})$ respectively denote the robot configurations as calculated by integrating a DMP and the associated ZMPs. The support polygon and its center point $\mathbf{x}_{\mathrm{P}}$ remain constant if the supporting feet do not move, but could otherwise be re-calculated at each time step. We have to use different sampling rates $t_{1,i}$ and $t_{2,i}$ because the control rate of our robots and Kinect are not the same. α and β are automatically determined so that the values of Δ_{ZMP} and Δ_{KIN} are in the same range. They can be set for example to $a = s/\|\mathbf{q}_{\max} - \mathbf{q}_{\min}\|^2$ and $b = s/d(\mathbf{x}_P)^2$, where $\mathbf{q}_{\min}$ and $\mathbf{q}_{\max}$ are the joint limits, $d(\mathbf{x}_P)$ is the distance from the center of the support polygon to its boundary, and s is the common desired scale. γ is a free parameter that can be selected by a user and balances the weighting between the fidelity of movement reproduction and stability. The reward is defined as necessary for importance sampling, which provides the basis of the reinforcement learning algorithm PoWER; it becomes small if the value of error functions (25) and (26) increases, and it is equal to 1 if there are no errors.

The only other free parameter in PoWER besides the reward function is the exploration noise, which must be decided by the user. See [5] for details. We set different exploration noise for every degree of freedom, reflecting the fact that some degrees of freedom affect the stability significantly more than others.

V. Experimental Results

Our real world experiments were conducted on a small-size humanoid robot HOAP-3 (see Fig. 2) and on a full-size humanoid CB-i [1] (see Fig. 1). The simulation experiments were performed using the Simulation Laboratory software package [14] with CB-i as a model.

In our first experiment we transferred a human walking pattern to HOAP-3. The results are shown in Fig. 2 and 3. As HOAP-3 is relatively slow, there was a rather large delay between human demonstration and execution on the robot. Therefore the human demonstrator had to perform his movements rather slowly. The robot automatically inferred from a human demonstration when to switch between the left and right leg as the main supporting leg and between single and double support phase. See also the video attached to this paper.

In our next experiment we transferred human motion to a full-size humanoid robot CB-i. Here there was much less delay between the observed and the reproduced movement, as can be seen in the attached video. We successfully transferred the movement of 15 degrees of freedom (2 in each leg, 3 in the torso, and 4 in each arm). The upper-body degrees of freedom were further improved in simulation by reinforcement learning. The results are shown in Fig. 4

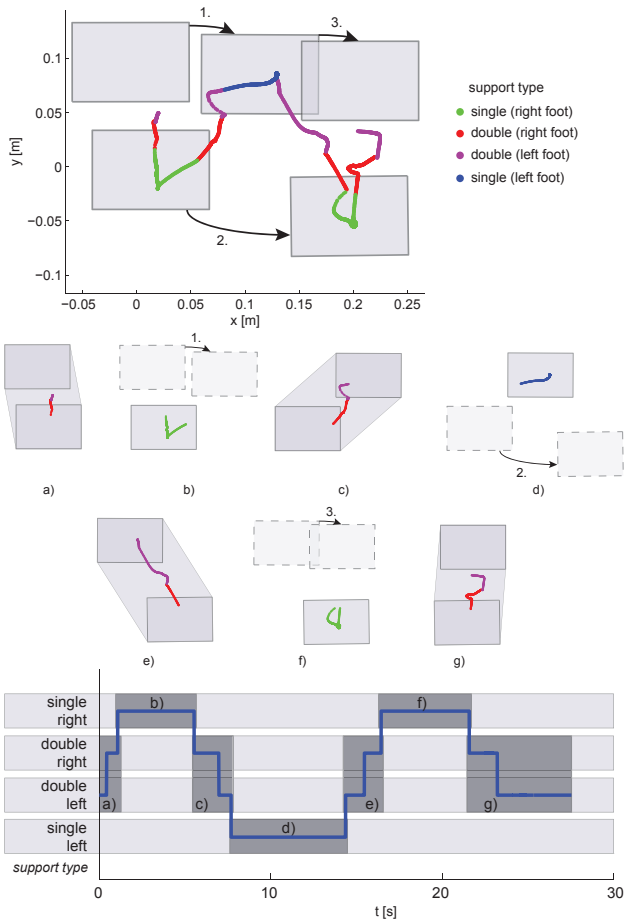

Fig. 3. The evolution of ZMP while transferring a human walking pattern to HOAP-3. In the top row different colors indicate how the robot switches between a single support and double support phase and how a different leg is selected as the main supporting leg in Eq. (9). The middle row shows the evolution of ZMP within each current support polygon. The bottom row shows the time evolution of the walking pattern.

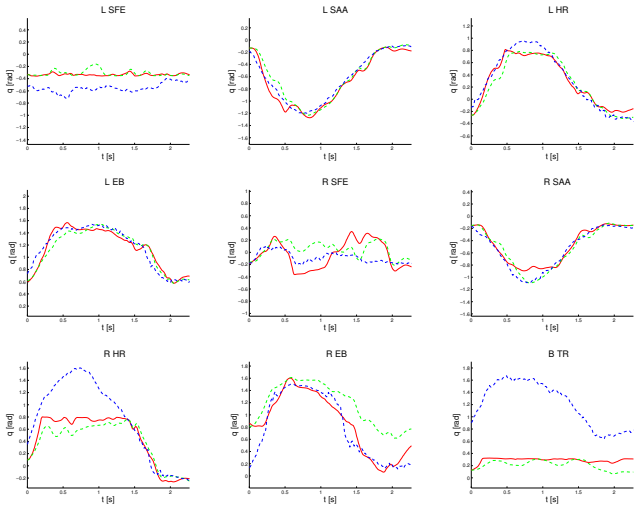

Fig. 4. The initial robot movements transferred from Kinect while maintaining stability (green), the learned, stable DMP (red), and the trajectory measured by Kinect (blue) for waving while squatting movement.

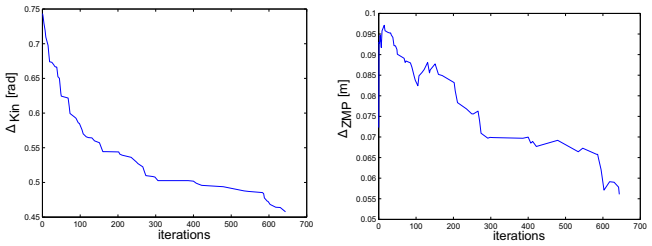

Fig. 5. Improvement achieved by reinforcement learning for waving while squatting movement. Left is Δ_{KIN} and right Δ_{ZMP}.

- 7. In simulation the ZMP was calculated by simulating pressure sensors on both feet of the robot. In the case of waving, a larger weight was given to the reward based on ZMP, while in the boxing example, we gave more importance to the reward that evaluates the fidelity of reproduction. As a result, the ZMP error was significantly reduced by reinforcement learning in the first case, while in the latter case it changed little. Note that since we discard all unstable movements during learning, the final trajectory is stable even if we perform learning based mainly on the trajectory part of the reward. The fidelity of reproduction was improved in both cases, but the improvement was more significant and faster in the case of boxing movement. A significant portion of the difference between the final robot trajectory and the observed human trajectory is caused by the physical limits of the robot.

VI. Conclusion

In this paper we showed that it is possible to transfer human movements observed by low cost RGB-D cameras to humanoid robots in real time, resulting in dynamically stable humanoid robot movements. We proposed a new approach based on prioritized control to simultaneously transfer human movements and control the stability of the robot. The proposed approach is able to automatically switch between free imitation and imitation with balance control. With the developed systems we were able to transfer movements as difficult as human walking patterns to a small-size humanoid robot.

In general it is very difficult to acquire accurate models of the robot dynamics. Thus movements obtained by utilizing robot dynamics models are usually suboptimal due to the discrepancies between the model and the real dynamics. In simulation we showed that by applying a probabilistic reinforcement learning algorithm PoWER, both the stability and the fidelity of reproduction can be improved.

The proposed approach enables the learning of dynamically stable humanoid movement primitives. Up to now we focused on movements and tasks that do not involve the manipulation of other objects in the environment. However, the proposed integration of imitation and reinforcement learning has the potential to be applied also to object manipulation in a model-free way. This is the next major goal of our research.

Acknowledgment

The research leading to these results has received funding from the European Community's Seventh Framework Pro-

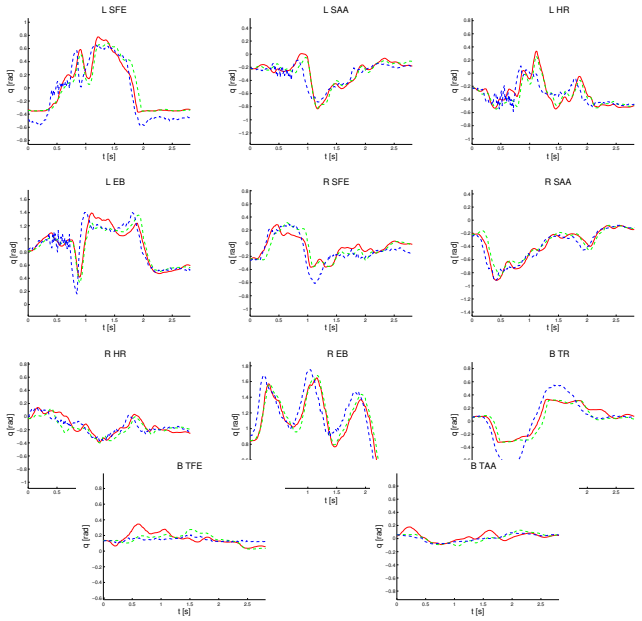

Fig. 6. The initial robot movements transferred from Kinect while maintaining stability (green), the learned, stable DMP (red), and the trajectory measured by Kinect (blue) for boxing movement.

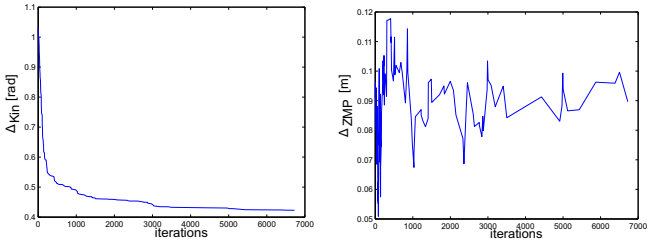

Fig. 7. Improvement achieved by reinforcement learning for boxing movement. Left is Δ_{KIN} and right Δ_{ZMP}.

gramme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under grant agreements no. 269959, IntellAct, and no. 270273, Xperience. It has also been supported by SRPBS, MEXT, by MEXT KAKENHI Grant Number 23120004, by a contract with the Ministry of Internal Affairs and Communications entitled, 'Novel and innovative R&D making use of brain structures' and by Strategic International Cooperative Program, JST.

REFERENCES

[1] G. Cheng, S.-H. Hyon, J. Morimoto, A. Ude, J. G. Hale, G. Colvin, W. Scroggin, and S. C. Jacobsen. CB: a humanoid research platform for exploring neuroscience. *Advanced Robotics*, 21(10):1097–1114, 2007.

[2] A. Goswami. Postural stability of biped robots and the foot-rotation indicator (FRI) point. *Int. J. Robot. Res.*, 18(6):523–533, 1999.

[3] A. J. Ijspeert, J. Nakanishi, T. Shibata, and S. Schaal. Nonlinear dynamical systems for imitation with humanoid robots. In *IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, pages 219–226, Tokyo, Japan, 2001.

[4] 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 Int. Conf. Robotics and Automation (ICRA)*, pages 1620–1626, Taipei, Taiwan, 2003.

[5] J. Kober and J. Peters. Policy search for motor primitives in robotics. *Machine Learning*, 84(1-2):171–203, 2011.

[6] M. Mistry, J. Nakanishi, and S. Schaal. Task space control with prioritization for balance and locomotion. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, pages 331–338, San Diego, CA, 2007.

[7] S. Nakaoka, A. Nakazawa, F. Kanehiro, K. Kaneko, M. Morisawa, H. Hirukawa, and K. Ikeuchi. Learning from observation paradigm: Leg task models for enabling a biped humanoid robot to imitate human dances. *Int. J. Robot. Res.*, 26(8):829–844, 2007.

[8] K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Inaba, and H. Inoue. Online generation of humanoid walking motion based on a fast generation method of motion pattern that follows desired ZMP. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, pages 2684–2689, Lausanne, Switzerland, 2002.

[9] C. Ott, D. Lee, and Y. Nakamura. Motion capture based human motion recognition and imitation by direct marker control. In *IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, pages 399–405, Daejeon, Korea, 2008.

[10] T. Petrič and L. Žlajpah. Smooth transition between tasks on a kinematic control level: Application to self collision avoidance for two Kuka LWR robots. In *IEEE Int. Conf. Robotics and Biomimetics (ROBIO)*, pages 162–167, Pukhet, Thailand, 2011.

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

[12] M. Riley, A. Ude, and C. G. Atkeson. Methods for motion generation and interaction with a humanoid robot: case studies of dancing and catching. In *Workshop Interactive Robotics and Entertainment (WIRE-2000)*, pages 35–42, Pittsburgh, PA, 2000.

[13] P. Sardain and G. Bessonnet. Forces acting on a biped robot. Center of pressure – zero moment point. *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, 34(5):630–637, 2004.

[14] S. Schaal. The SL simulation and real-time control software package. Technical report, Computational Learning and Motor Control Laboratory, University of Southern California, Los Angeles, CA, 2009.

[15] S. Schaal, P. Mohajerian, and A. Ijspeert. Dynamics systems vs. optimal control – a unifying view. *Prog. Brain Res.*, 165(6):425–445, 2007.

[16] L. Sentis, J. Park, and O. Khatib. Compliant control of multicontact and center-of-mass behaviors in humanoid robots. *IEEE Trans. Robot.*, 26(3):483–501, 2010.

[17] K. Shoemake. Euler angle conversion. In P. Heckbert, editor, *Graphic Gems IV*, pages 222–229. Academic Press, 1994.

[18] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *IEEE Conf. Computer Vision and Pattern Recognition (ICPR)*, pages 1297–1304, Colorado Springs, Colorado, 2011.

[19] F. Stulp, J. Buchli, E. A. Theodorou, and S. Schaal. Reinforcement learning of full-body humanoid motor skills. In *10th IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, pages 405–410, Nashville, Tennessee, 2010.

[20] T. Sugihara, Y. Nakamura, and H. Inoue. Realtime humanoid motion generation through ZMP manipulation based on inverted pendulum control. In *IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 1404–1409, Washington, DC, 2002.

[21] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon. The Vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *IEEE Conf. Computer Vision and Pattern Recognition (ICPR)*, pages 103–110, Providence, RI, 2012.

[22] A. Ude, C. G. Atkeson, and M. Riley. Planning of joint trajectories for humanoid robots using B-spline wavelets. In *IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 2223–2228, San Francisco, California, 2000.

[23] M. Vukobratović, B. Borovac, and V. Potkonjak. Towards a unified understanding of basic notions and terms in humanoid robotics. *Robotica*, 25(1):87–101, 2007.

[24] K. Yamane and J. Hodgins. Simultaneous tracking and balancing of humanoid robots for imitating human motion capture data. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, pages 2510–2517, St. Louis, USA, 2009.