Full Body Motion Adaption based on Task-Space Distance Meshes

Thomas Nierhoff, Sandra Hirche, Wataru Takano and Yoshihiko Nakamura

***Abstract*— This paper presents a novel robot pose measure for human movement imitation based entirely on the Euclidean distance information between any two links of a robot and any link and object in the robot's environment in a Cartesian task space. A Hidden Markov Model is used to encode the spatio-temporal information of multiple demonstrations. In combination with Gaussian Mixture Regression for extracting the important task properties, feasible full-body motion adaption can be achieved. The method is suited for use with a humanoid robot by considering additional constraints like balance control and collision avoidance. In order to tackle modeling errors occurring due to the human movement demonstration and the robotic reproduction, a manipulability based weighting scheme is proposed. Complexity reduction of the otherwise redundant pose measure is performed based upon a mechanical analogy of an interconnected spring system. Experiments are conducted using a HRP-4 robot and display the applicability of the presented methods for robotic full-body motion imitation tasks.**

I. INTRODUCTION

Programming by demonstration is a promising way for humanoid robots to learn complex tasks in a human-like fashion. Instead of programming every desired movement from scratch, a human movement fulfilling the desired task is adopted to a robot and modified when necessary. Movement modifications are necessary for various reasons, e.g. environmental changes, differences in the pose of human and robot or intentionally altered movements during reinforcement learning of a task.

Various adaption schemes are proposed in literature in order to encode and reproduce task-specific movements. This includes *Dynamic Movement Primitives* (DMP), a method disturbing a stable second order attractor system locally using a weighted set of nonlinear differential equations. Successful application is shown in [1], [2] both to one-link manipulators and humanoid robots being able to react dynamically to changes in the environment. Another approach are *Gaussian Mixture Models/Gaussian Mixture Regression* (GMM/GMR), encoding a time-index trajectory set by a finite number of Gaussian functions, see [3], [4]. This provides information about both, most likely trajectory and corresponding spatial variance. Especially the variance is of interest as it can be used as a measure of importance of different task elements. A more specific approach for precise object manipulation is presented in [5], where the combination of two HMMs triggering first the hand motion and then the full body motion is presented. Last, [6] present an extension to Dynamic Bayesin Networks, being also able to deal with unforeseen perturbations.

A common property of all presented methods is the possible handling of both free space movements and interactive robot tasks with objects/humans in its environment. Problems arise as most methods encode the robot's joint angles even if the quality of imitation is mostly evaluated in task space - either by inspection or using endeffector trajectory deviation measurements. As the mapping from joint to task space is highly nonlinear depending on the robot's geometry, it can be prone to errors in case of large environmental changes. In this case, a measure based upon task space information can be both more robust and intuitive. Hence various approaches grab up this idea, see [7], [8].

Therefore this paper presents an entirely task-space based measure of both the robots pose and the interaction of the robot with its environment. Inspired by the work presented in [9], [10] not the joint angles for describing the robots current pose but instead the Euclidean distance between any two links of the robot is used. In a similar fashion for interacting with the environment, the distance between any link of the robot and any object in the environment is calculated. The importance of each distance is encoded in its spatial variance and evaluated using HMM and GMR. Thus it can be ensured only the important task-specific properties are considered both for the robot pose and the environment interaction. Further analysis tackles the question how the proposed method can be used as a singularity avoidance scheme and to what extent its complexity can be reduced. Experiments performing motion adaption on a HRP-4 humanoid robotic platform show that the proposed approach can be adapted to real life problems by adding constraints like collision avoidance, balance of the robot defined by its center of mass or joint angle limits.

The remainder of this paper is organized as follows: Sec. II displays the general approach and improvements leading to better overall performance. Experimental results using a HRP-4 robotic platform are presented in Sec. III. Sec. IV discusses the presented approach. Last, Sec. V concludes with a final statement and possible expansions for the future.

Notation: Throughout the article scalars are written in non-bold letters (e.g a), vectors in bold lower case letters (e.g. $\mathbf{a}$) and matrices in bold capital letters (e.g. $\mathbf{A}$). Accessing a specific element of a matrix/vector is denoted by square subscript brackets (e.g. $\mathbf{A}_{[3,2]}$ for the third row, second column of $\mathbf{A}$).

Thomas Nierhoff and Sandra Hirche are with the Institute for Information-oriented Control (ITR), Faculty of Electrical Engineering, Technische Universität München, D-80290 München, Germany {tn, hirche}@tum.de. Thomas Nierhoff, Wataru Takano and Yoshihiko Nakamura are with the Department of Mechano-Informatics, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, 113-8656 Tokyo, Japan {takano, nakamura}@ynl.t.u-tokyo.ac.jp.

II. Motion Learning and Reproduction

A. Data Representation

Let the robot be represented as a kinematic chain with k joints, the corresponding joint angle vector $\boldsymbol{\theta}$ and l links with corresponding position $\mathbf{p}_i^l \in \mathbb{R}^3, i = \{1, \dots, l\}$. The same accounts for a set of objects in the environment. In order to obtain information about the objects orientation, o object points $\mathbf{p}_i^o \in \mathbb{R}^3, i = \{1, \dots, o\}$ are defined depending on the objects' geometry. For easier handling, let $n = l + o$ be the total number and $\mathbf{p}_i \in \mathbb{R}^3, i = \{1, \dots, n\}$ be the total set of position vectors to be considered. The time-varying version of $\mathbf{p}_i$ is denoted $\mathbf{p}_{i,t}$. Then the distance mesh vector $\mathbf{d} \in \mathbb{R}^{n(n-1)/2}$ stores every possible Euclidean distance between the position vectors. For every time step $t \in \{0, \dots, t_{max}\}$ a new distance mesh vector $\mathbf{d}_t$ is calculated, thus encoding the temporal progress of the distance mesh.

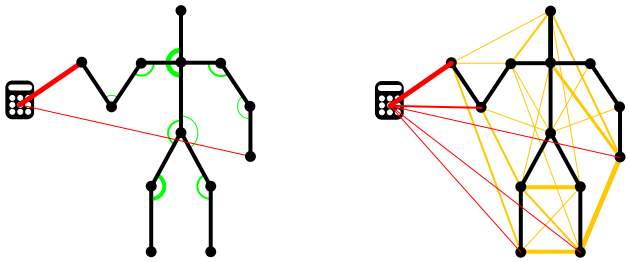

Fig. 1. Conventional approach (left) and new approach (right). Whereas the conventional approach encodes the robot joint angles (green) and distances to objects (red) separately, the method presented in this paper encodes the pose of the robot and the interaction with its environment through the weighted set of Cartesian link-link (orange) and link-object (red) distances.

B. Data Encoding and Reproduction

A CHMM in combination with GMR is used to encode the temporal progress of every element of the distance mesh $\mathbf{d}_t$ based on multiple task repetitions. The CHMM is represented by the parameter set $\{\boldsymbol{\pi}, \mathbf{a}, \mathbf{w}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ with $\boldsymbol{\pi}$ for the initial state probabilities, $\mathbf{a}$ the state transition probabilities and $\mathbf{w}$, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ for weight, mean and covariance of each state of the CHMM. In addition to the spatial mean $\mu_{s,g}$ and variance $\Sigma_{s,g}$ for every Gaussian mixture component g for a single dimension, the temporal information are used in addition to calculate the temporal mean $\mu_{t,g}$, the temporal variance $\Sigma_{t,g}$ and the covariance between temporal and spatial data $\Sigma_{ts,g}$, resulting in the spatio-temporal mean vector $\boldsymbol{\mu}_g$ and covariance matrix $\boldsymbol{\Sigma}_g$. For a single dimension of the CHMM for a single state it is defined as

$$\boldsymbol{\mu}_g = \begin{bmatrix} \mu_{s,g} \\ \mu_{t,g} \end{bmatrix}, \qquad \boldsymbol{\Sigma}_g = \begin{bmatrix} \Sigma_{s,g} & \Sigma_{ts,g} \\ \Sigma_{ts,g} & \Sigma_{t,g} \end{bmatrix}. \tag{1}$$

Data reproduction is performed through GMR based on $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}_g$ for every dimension as described in [11]. This provides a prototypic (i.e. most likely) distance mesh $\hat{\mathbf{d}}_t \in \mathbb{R}^{n(n-1)/2}$ together with its associated prototypic spatial variance $\hat{\boldsymbol{\Sigma}}_{s,t} \in \mathbb{R}^{n(n-1)/2 \times n(n-1)/2}$

$$\hat{\boldsymbol{\Sigma}}_{s,t} = diag(\Sigma_{s,g,1}, \dots, \Sigma_{s,g,n(n-1)/2}), \tag{2}$$

for every time step t. Both $\hat{\mathbf{d}}_t$ and $\hat{\boldsymbol{\Sigma}}_{s,t}$ in combination with the vectors $\mathbf{p}_{i,t}$ provide all information about the robots' pose and its interaction with environmental objects. The time-dependent variance $\hat{\boldsymbol{\Sigma}}_{s,t}$ encodes the relative importance of each element of the prototypic distance mesh, except for those elements of the distance mesh where the robot's geometry imposes a time-invariant constant distance. Whereas generally the inverse of the variance as

$$\mathbf{W}_t = \hat{\boldsymbol{\Sigma}}_{s,t}^{-1}, \tag{3}$$

is used to represent the variance-dependent importance matrix $\mathbf{W}_t \in \mathbb{R}^{n(n-1)/2}$ of every element of the distance mesh, a more general version is used in this paper as

$$\mathbf{W}_t = f(\hat{\boldsymbol{\Sigma}}_{s,t}^{-1}), \tag{4}$$

such that $\mathbf{W}_t$ is positive definite.

By measuring all distances in task space and using only a single weighting term $\mathbf{W}_t$ we overcome the problem of weighting joint space and task space distances relatively to each other, see [3]. As the distance mesh includes all distances between any link of the robot and the task-relevant objects in the environment, information are provided for the time-dependent entire robot pose relative to the objects. Thus the approach differs from other methods which have to specify the important limbs (e.g. hands/feed) manually.

C. Motion Adaption

When executing the learned task in a new environment, the reproduced distance mesh $\tilde{\mathbf{d}}_t(\boldsymbol{\theta}_t)$ at time t generally differs from $\boldsymbol{\theta}_t$. Yet the goal is to find a robot configuration defined by the joint angles $\boldsymbol{\theta}_t$ such that the cost function c_{cf} based on the weighted difference between $\tilde{\mathbf{d}}_t(\boldsymbol{\theta}_t)$ and the prototypic distance mesh $\hat{\mathbf{d}}_t$ becomes minimal

$$\mathbf{d}_{rel[i]} = \frac{\tilde{\mathbf{d}}_{t[i]}(\boldsymbol{\theta}_t) - \hat{\mathbf{d}}_{t[i]}}{\hat{\mathbf{d}}_{t[i]}}, \tag{5}$$

$$c_{cf} = \mathbf{d}_{rel}^T \mathbf{W}_t \mathbf{d}_{rel}, \tag{6}$$

$$\boldsymbol{\theta}_{min} = \underset{\boldsymbol{\theta}_t}{\operatorname{argmin}}(c_{cf}). \tag{7}$$

Differing from [4], the relative deviation $(\tilde{\mathbf{d}}_{t[i]}(\boldsymbol{\theta}_t) - \hat{\mathbf{d}}_{t[i]})/\hat{\mathbf{d}}_{t[i]}$ is used instead of the absolute deviation $\tilde{\mathbf{d}}_{t[i]}(\boldsymbol{\theta}_t) - \hat{\mathbf{d}}_{t[i]}$ in order to increase the influence of deviations of small elements of the distance mesh. The corresponding Jacobian $\mathbf{J}_{cf}$ for the cost function c_{cf} is derived as

$$\mathbf{J}_{cf} = \frac{\partial c_{cf}}{\partial \boldsymbol{\theta}_t}. \tag{8}$$

For execution on a real robot, a two-staged prioritized differential inverse kinematics (IK) approach [12] with included inequality constraints as described in [13] is used. The Jacobians of primary and secondary task are denoted $\mathbf{J}_1$ and $\mathbf{J}_2$ and the pseudoinverse of a Jacobian $\mathbf{J}$ is $\mathbf{J}^{\#}$. The joint space velocities $\dot{\boldsymbol{\theta}}$ are related to the task space velocities $\dot{\mathbf{r}}_{\{1,2\}}$ for primary and secondary task as

$$\dot{\mathbf{r}}_{\{1,2\}} = \mathbf{J}_{\{1,2\}} \dot{\boldsymbol{\theta}}. \tag{9}$$

Then the general solution of the prioritized IK such that the secondary task interacts as least as possible in the least squares sense with the primary task with $\mathbf{E}$ as the identity matrix is

$$\dot{\boldsymbol{\theta}} = \mathbf{J}_1^{\#}\dot{\mathbf{r}}_1 + (\mathbf{J}_2(\mathbf{E} - \mathbf{J}_1^{\#}\mathbf{J}_1))^{\#}(\dot{\mathbf{r}}_2 - \mathbf{J}_2\mathbf{J}_1^{\#}\dot{\mathbf{r}}_1), \quad (10)$$

see [12] for an in-depth discussion about this topic. The motion imitation task is incorporated in the secondary task. With K_{cf} as a variable gain factor controlling the speed of the gradient descent of c_{cf}, it is

$$\mathbf{J}_2 = \mathbf{J}_{cf}, \quad (11)$$
$$\dot{\mathbf{r}}_2 = -K_{cf}c_{cf}. \quad (12)$$

Both Jacobian $\mathbf{J}_1$ and vector $\dot{\mathbf{r}}_1$ consist of three elements, accounting for balance of the robot ($\mathbf{J}_{com}$ and $\dot{\mathbf{r}}_{com}$), fixed position of the feet ($\mathbf{J}_f$ and $\dot{\mathbf{r}}_f$) and self-collision avoidance ($\mathbf{J}_{ca}$ and $\dot{\mathbf{r}}_{ca}$). It is

$$\mathbf{J}_1 = \begin{bmatrix}\mathbf{J}_{com} & \mathbf{J}_f & \mathbf{J}_{ca}\end{bmatrix}^T, \quad (13)$$
$$\dot{\mathbf{r}}_1 = \begin{bmatrix}\dot{\mathbf{r}}_{com} & \dot{r}_f & \dot{\mathbf{r}}_{ca}\end{bmatrix}^T. \quad (14)$$

Given sufficiently slow movements, balance of the robot can be achieved by keeping the *center of mass* (COM) $\mathbf{p}_{com} \in \mathbb{R}^2$ within the support polygon of the robot. A balance controller is used that moves the COM of the robot towards the center $\mathbf{p}_{com}^{ref} \in \mathbb{R}^2$ of the support polygon

$$\dot{\mathbf{r}}_{com} = K_{com}(\mathbf{p}_{com}^{ref} - \mathbf{p}_{com}), \quad (15)$$

with adjustable gain K_{com}.

By defining three sampling points $\mathbf{p}_{i,j,f}$ and each feet transformation $\mathbf{T}_i \in \mathbb{R}^{3\times3},\ i = 1, 2$ (for example the trihedron $\mathbf{T}_i[\mathbf{e}_x, 1]^T$, $\mathbf{T}_i[\mathbf{e}_x, 1]^T$, $\mathbf{T}_i[\mathbf{e}_z, 1]^T$), a single Jacobian $\mathbf{J}_f$ accounting both for orientational and positional errors of the feet can be calculated based on the cost function

$$c_f = \sum_{i=1}^{2}\sum_{j=1}^{3}\|\mathbf{p}_{i,j,f}^{ref} - \mathbf{p}_{i,j,f}\|_2^2. \quad (16)$$

The scalar $\dot{r}_f$ is then defined in terms of the gradient descent speed $\dot{r}_f = -K_f c_f$ with gain K_f.

Self-collision avoidance between two links of the robot is based upon enclosing cylinders covering all robot links. In this case, if the distance $d_{i,ca}$ between two links falls below a certain threshold $d_{i,ca}^{min}$, the desired collision avoidance velocity is expressed as

$$\dot{r}_{i,ca} = K_{ca}(d_{i,ca}^{min} - d_{i,ca}) \text{ if } d_{i,ca} < d_{i,ca}^{min}, \quad (17)$$

with gain factor K_{ca}, resulting in $\dot{r}_{ca}$ and $\mathbf{J}_{ca}$. Joint angle limits are considered through an underlying controller with bilateral boundaries.

D. Handling Modeling/Measurement Errors

Using only a task-space based measure does not consider the geometry of the robot, see Fig. 2: Shown are two different manipulators, one element of the distance mesh - the distance base-endeffector - and a hypothetical manipulability ellipsoid for the endeffector. If due to a modeling/measurement error the distance base-endeffector has to be increased based on $\mathbf{J}_{cf}$, the joint angle changes are minimal for the left manipulator. The right configuration however may enter a singularity as a consequence of a non-reachable space. When using the singular-robust (SR) pseudoinverse, the additive term of the SR pseudoinverse may cause an unpredictable drift of the joint angles close to a singularity. Thus another option based on the manipulability ellipsoid is proposed that modifies the cost function c_{cf} in such a way that the elements of the distance mesh leading to a singular configuration are weighted less.

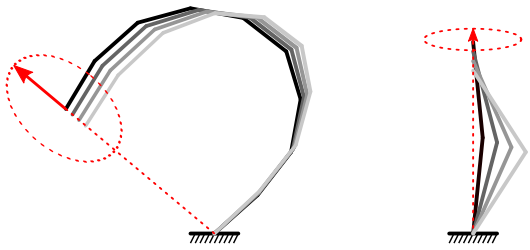

Fig. 2. Effect of different manipulability: Changing the distance base-endeffector (red dotted line) results in higher joint velocities on the right side than on the left side. To incorporate this effect into the cost function c_{cf}, the manipulability along the direction of each mesh distance element is calculated (red arrow)

In general, the manipulability ellipsoid is defined as

$$\dot{\mathbf{r}}^T(\mathbf{J}\mathbf{J}^T)^{-1}\dot{\mathbf{r}} = 1, \quad (18)$$

and used to calculate the manipulability along an arbitrary direction. Yet we are only interested in the manipulability along the direction of each element of the distance mesh, see the red arrows in Fig. 2. Assuming that $\dot{\mathbf{r}}_{i,t} \in \mathbb{R}^3$ is the direction the i-th element of $\mathbf{d}_t$ accounts for, we get

$$\dot{\mathbf{r}}_{i,t}^T(\mathbf{J}_{i,t}\mathbf{J}_{i,t}^T)^{-1}\dot{\mathbf{r}}_{i,t} = 1, \quad (19)$$

for every element of $\mathbf{d}_t$. It is then

$$\mathbf{D}_{man} = diag(\|\dot{\mathbf{r}}_{1,t}\|_2, \ldots, \|\dot{\mathbf{r}}_{n(n-1)/2,t}\|_2), \quad (20)$$
$$\mathbf{W}_t = \mathbf{D}_{man}\hat{\boldsymbol{\Sigma}}_{s,t}^{-1}, \quad (21)$$

i.e. the elements of the distance mesh which change of length according to (8) would cause high joint velocities are both weighted less and considered less during gradient calculation.

For calculating the elements of $\mathbf{D}_{man}$ it is necessary to calculate the Jacobian matrices between any two links of the robot. Assuming $\mathbf{J}_{i,t}$ is the Jacobian of link G with respect to link F, i.e. $\mathbf{J}_{i,t} = {}^F\mathbf{J}_G$. Usually, all Jacobians are calculated with respect to a base link 0, resulting in ${}^0\mathbf{J}_G$ and ${}^0\mathbf{J}_F$. The calculation of ${}^F\mathbf{J}_G$ given only ${}^0\mathbf{J}_G$ and ${}^0\mathbf{J}_F$ is (see [14])

$${}^F\mathbf{J}_{G,tr} = \mathbf{R}_0\left\{{}^0\mathbf{J}_{G,tr} - {}^0\mathbf{J}_{F,tr} + ({}^0\mathbf{r}_G - {}^0\mathbf{r}_F)^{\times}\,{}^0\mathbf{J}_{F,\omega}\right\}, \quad (22)$$

with ${}^0\mathbf{r}_G$ and ${}^0\mathbf{r}_F$ being the position of link G and F with respect to the base link 0, the subscript $_{tr}$ respectively $_{\omega}$ denoting the translational and rotational part of the Jacobian, the matrix $\mathbf{R}_0$ accounting for the pose of the base link with

respect to world coordinates and $^\times$ for the cross product and therefore skew-symmetric matrix of a vector.

E. Complexity Reduction

The approach suffers from two large drawbacks: A $\mathcal{O}(n^2)$ complexity for a given set of position vectors and redundancy in the weighting matrix $\mathbf{W}_t$ as its elements can differ by several magnitudes. Two methods are presented in this chapter that consider only a subset of all distance mesh elements when calculating the cost function c_{cf}. The first method considers only a fraction of the largest elements in $\mathbf{W}_t$. In case very few elements are considered, this might cause undesired nullspace movements of robot limbs not included anymore in c_{cf}. The second more elaborate method interprets the distance mesh as a mechanical system of springs keeping the points $\mathbf{p}_i$ in position. By removing springs successively, it is investigated how well the points $\mathbf{p}_i$ can be fixed to a certain spatial position. This is achieved by looking at the volume of the unit energy ellipsoid when displacing every point $\mathbf{p}_i$. Differing from the first attempt, this method encodes not only the weighting $\mathbf{W}_t$ but also the spatial arrangement of the distance mesh.

More in detail, the j-th element that c_{cf} consists of can be seen as a mechanical spring with displacement $\mathbf{s}_{j,t} = \tilde{\mathbf{d}}_{j,t}(\boldsymbol{\theta}_t) - \hat{\mathbf{d}}_{j,t}$ and spring constant $D_{j,t} = \frac{\mathbf{W}_{jj,t}}{\hat{\mathbf{d}}_{j,t}^T \hat{\mathbf{d}}_{j,t}}$.

Under the assumption that one can achieve a perfect reproduction of the movement, it is $\mathbf{s}_{j,t} = 0 \ \forall j, t$. For a single spring between two endpoints and a small displacement of one of the two endpoints $\mathbf{p}_i$ in a random direction $\Delta\mathbf{s}_{i,t}$ (small with respect to $\hat{\mathbf{d}}_{j,t}$), the displacement force $\mathbf{f}_{i,t} \in \mathbb{R}^3$ can be expressed as $\mathbf{f}_{i,t} \approx \mathbf{D}_{j,t}\Delta\mathbf{s}_{i,t}$. Note that we use the symmetric matrix $\mathbf{D}_{j,t} \in \mathbb{R}^{3\times3}$ instead of the scalar $D_{j,t}$, accounting for the fact that $\Delta\mathbf{s}_{i,t}$ and $\hat{\mathbf{d}}_{j,t}$ are usually not coaligned. Consequently, the energy $E_{i,t}$ required to cause the displacement can be written as

$$E_{i,t} = \frac{1}{2}\Delta\mathbf{s}_{i,t}^T \mathbf{D}_{j,t} \Delta\mathbf{s}_{i,t}. \tag{23}$$

if there is more than one spring attached to the point $\mathbf{p}_i$, the displacement energy is the sum

$$E_{i,t} = \frac{1}{2}\Delta\mathbf{s}_{i,t}^T (\sum_j \mathbf{D}_{j,t}) \Delta\mathbf{s}_{i,t}. \tag{24}$$

It is visible that the energy required to displace the point $\mathbf{p}_i$ depends both on the direction of displacement and the amount of displacement. In order to find configurations where a displacement of $\mathbf{p}_i$ is possible without changing of the energy $E_{i,t}$ we look at the size of the constant energy ellipsoid defined by

$$1 = \frac{1}{2}\Delta\mathbf{s}_{i,t}^T (\sum_j \mathbf{D}_{j,t}) \Delta\mathbf{s}_{i,t}, \tag{25}$$

for every point $\mathbf{p}_i$. Similar to the manipulability ellipsoid, its volume $v_{i,t}$ can be calculated up to a constant multiplying factor as

$$v_{i,t} = \sqrt{\det((\sum_j \mathbf{D}_{j,t})^{-1})}. \tag{26}$$

The final cost function $c_{r,t}$ can thus be defined as the sum over all volumes $v_{i,t}$

$$c_{r,t} = \sum_{i=1\ldots n} v_{i,t}. \tag{27}$$

Thus, when selecting the elements of the distance mesh, only those elements are chosen that minimize c_r.

The matrix $\mathbf{D}_{j,t}$ is the combination of the scalar stiffness value $D_{j,t}$ and a matrix $\mathbf{M}$ only considering the amount of $\Delta\mathbf{s}_{i,t}$ that points in the direction of $\hat{\mathbf{d}}_{j,t}$. Using the rotation matrix $\mathbf{R}_{j,t}$ that coaligns the vector $[1,0,0]^T$ with $\hat{\mathbf{d}}_{j,t}$, it can be calculated as

$$\mathbf{D}_{j,t} = D_{j,t}\mathbf{R}_{j,t} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{R}_{j,t}^T. \tag{28}$$

Whereas theoretically the subset of distance mesh elements has to be recalculated at every time step, it leads to discontinuities as the distance mesh constantly changes its appearance. Thus only one single subset is calculated based on the summed values of $\mathbf{W}_t$ and $\hat{\mathbf{d}}_t$, resulting in the cost function c_r. Note that Delaunay tetrahedralization as used originally in [9] is suboptimal as it cannot consider the weighting $\mathbf{W}_t$ for all possible distance mesh elements.

III. Experimental Evaluation

Experiments are conducted using a Cortex motion capture system tracking the movements of a human test person at 200Hz and providing realistic full-body movements for a given task [15]. The free space motion is subsequently mapped onto a HRP-4 humanoid robotic platform, providing the link-link distances and overcoming the problem of different link lengths between human and robot. In addition, the link-object distances are measured directly from the human test run, causing some inevitable modeling errors due to the different figure of robot and human. Simulations are performed in Openrave [16]. Real life experiments are conducted using a HRP-4 robotic platform. The used parameterization is $K_{cf} = 0.2$, $K_{com}, K_f = 0.3$, $K_{ca} = 0.5$. In total, five demonstrations are performed for each scenario.

Two scenarios are analyzed, see Fig. 3: In the first scenario, named clap, the human puts his hands first on the lap while standing and claps after that with his hands risen up. This way, the capability of the link-link distances for human movement imitation is investigated. For the second scenario, named box, a box has to be lifted from the right side to the left side of the table. Here also the link-object distances between human limbs and the box are taken into account.

The comparison with a conventional method encoding only the joint angle movement, e.g. [17], is shown in Fig. 4. On the left side, reproduction of the clap scenario is displayed for a standing robot (top) and a sitting robot (bottom) with fixed leg joints using the method described in Sec. II-C. For comparison, the right side shows the similar robot

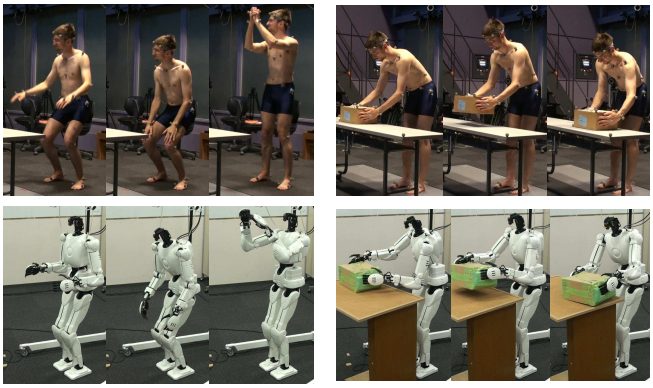

Fig. 3. Human demonstration and HRP4 robot imitation for the clap and box scenario

movement based on HMM-encoded joint angles, resulting in the time varying prototypic joint angle vector $\hat{\boldsymbol{\theta}}_t$. To ensure collision avoidance and a stable stand while keeping the positional joint error $\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_t$ small, a prioritized IK in the form of

$$\dot{\boldsymbol{\theta}} = \mathbf{J}_1^{\#}\dot{\mathbf{r}}_1 + K_\theta(\mathbf{E} - \mathbf{J}_1^{\#}\mathbf{J}_1)(\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_t), \tag{29}$$

with $\mathbf{J}_1$, $\dot{\mathbf{r}}_1$ and $\mathbf{E}$ similar to Sec. II-B, is used.

The reproduction of the original clap scenario differs slightly for both cases due to the different control loop used. Whereas the distance mesh based approach tends to lean backwards when sitting due to the link-link distances between legs and upper body, the joint based approach fails to place the robot's hands on the lap as it considers only joint angles but not the task space relation between lap and hand. Two additional plots provide quantitative results of the reproduction quality with respect to the reference robot motion. The left plot illustrates the imitation quality in terms of c_{cf} (6), that is the similarity of the underlying distance mesh whereas the right plot shows the cumulative positional joint error $\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_t$. Clearly the mesh-based method outperforms the joint-based method in terms of c_{cf} and vice-versa when it comes to the positional joint error.

Fig. 5 displays the adaption capability of the proposed approach to varied environments. Shown in each row are the generated robot movements for different box positions, thus indicating that the prototypic movements can be used for tasks involving object manipulation with varied object position.

In order to measure the influence of the approach presented in Sec. II-D, two different box scenarios with the weighting matrix $\mathbf{W}_t$ set either to $\mathbf{D}_{man}\hat{\boldsymbol{\Sigma}}_{s,t}^{-1}$ ("with manipulability") or $\hat{\boldsymbol{\Sigma}}_{s,t}^{-1}$ ("without manipulability") are analyzed. For the first scenario no COM, feet or collision constraints have been considered. For the second scenario previously mentioned constraints are taken into account. Plotted results of the fastest moving joint versus execution time are shown in Fig. 6. Close-to-singular configurations exist at 250ms and 2100ms, resulting in sudden joint speed peaks. However, the effect (and thus the difference between the two graphs) decreases with increasing number of additional constraints

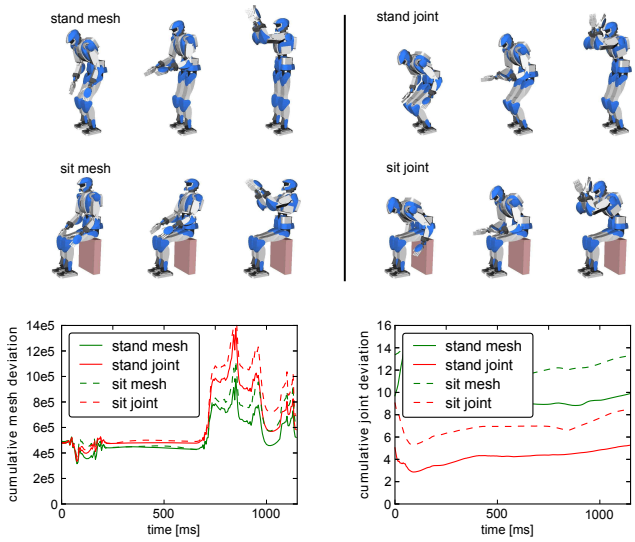

Fig. 4. Comparison between the method proposed in this paper (left) and a joint angle based approach (right) for the clap scenario for a standing and sitting robot

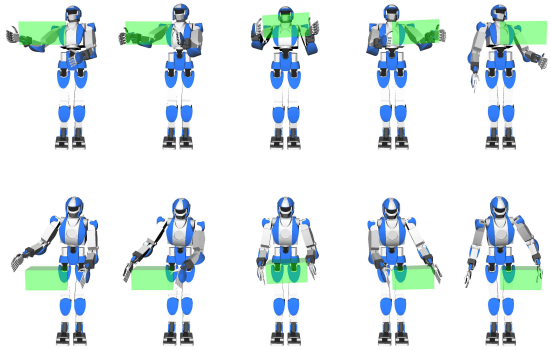

Fig. 5. Box scenario reproduction for two different box positions

as the weighting $\mathbf{W}_t$ is encoded in the secondary task and may be overridden by higher-prioritized tasks.

The amount of redundancy in the distance mesh is investigated in Fig. 7. Shown are results of both scenarios using only a fraction of the elements of the distance mesh as described in Sec. II-E. For the box scenario, 32 points are used to construct the distance mesh, resulting in a maximum of $(32 * 31)/2 = 496$ elements for the distance mesh. For the clap scenario, 21 points are used, resulting in up to 210 elements to be calculated. Given a reference scenario using all elements of the distance mesh, the average task space error of each point $\mathbf{p}_{i,t}$ with respect to its position of the reference scenario is shown when using only a fraction of 50%, 20%, 10% and 5% distance mesh elements. The blue and red bars correspond to the simple and elaborate method as described in Sec. II-E.

Final experiments using the HRP4 robot platform are displayed in Fig. 3. In order to compensate for the entirely position-controlled robot, a smaller box has been used for movement reproduction and compliant elements (sponges) have been attached to each side of the box to make it more

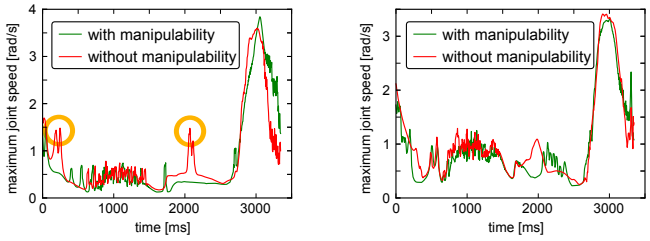

Fig. 6. Cumulative speed of the fastest moving joint versus execution time with reduced number of constraints (left side) and when considering all constraints (right side). Close to singular configurations are highlighted with orange circles

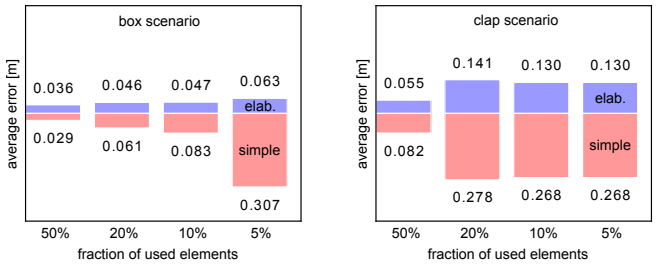

Fig. 7. Average positional error of all distance mesh points depending on the fraction of distance mesh elements used

elastic and prevent damage to the robot.

IV. Discussion

Experiments show how the proposed approach can be used both for free space movements and constrained manipulation.

The existing redundancy of the distance mesh constitutes a tradeoff between a minimal movement representation and versatile adaption capabilities regarding object manipulation and postural changes. Even though its $\mathcal{O}(n^2)$ complexity is disadvantageous for large-scale scenarios, only a fraction of all distance mesh elements has to be considered in order to produce satisfying results. A last problem all variance-based weighting approaches have in common is the incorporation of hard positional constraints. Due to measurement noise, probabilistic encoding and reproduction and pseudoinverse-based differential inverse kinematics, the adaption to new environments for tasks involving closed kinematic chains becomes challenging.

The approach is robust towards changes of the parameters $K_{com}, K_f, K_{ca}, K_{cf}$ as all of them can be varied by more than one magnitude without severe alterations of the reproduction result. The main challenge so far is the large discrepancy of the elements of the variance matrix $\hat{\mathbf{\Sigma}}_{s,t}$, prioritizing occasionally the wrong (i.e. from a human perspective task-irrelevant) distance mesh elements. Similar to other HMM/GMR-based movement imitation approaches the method is highly sensitive to the number of HMM/GMR states.

V. Conclusion and Future Work

This paper presents a novel method to encode and reproduce a prototypic robot movement for either free space or constrained movements using a purely task space based imitation measure. The advantage over an entirely joint space based approach in modified environments is demonstrated in experiments. As the proposed approach allows one to take also higher prioritized objectives like balance control or collision into consideration, execution on a humanoid robot is discussed in theory and shown by experiments.

Future work will be focused on an improved movement adaption scheme, enabling one to detect hard positional constraints automatically and to take them into account properly.

Acknowledgement

This work is supported in part within the DFG excellence research cluster *Cognition for Technical Systems - CoTeSys* (`www.cotesys.org`) and by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (S), 2008-2012, 20220001, “Establishing Human-Machine Communication through Kinesiology and Linguistics Integration” (PI: Y. Nakamura)

References

[1] D.-H. Park, H. Hoffmann, P. Pastor, and S. Schaal, “Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields,” in *IEEE Humanoids*, 2008, pp. 91–98.

[2] F. Stulp, J. Buchli, E. Theodorou, and S. Schaal, “Reinforcement learning of full-body humanoid motor skills,” in *IEEE Humanoids*, 2010, pp. 405–410.

[3] S. Calinon, F. Guenter, and A. Billard, “On learning the statistical representation of a task and generalizing it to various contexts,” in *IEEE ICRA*, 2006, pp. 2978–2983.

[4] ——, “On learning, representing and generalizing a task in a humanoid robot,” *IEEE SMC*, vol. 37, no. 2, pp. 286–298, 2007.

[5] H. Kunori, D. Lee, and Y. Nakamura, “Associating and reshaping of whole body motions for object manipulation,” in *IEEE IROS*, 2009, pp. 5240–5247.

[6] C. Eppner, J. Sturm, M. Bennewitz, C. Stachniss, and W. Burgard, “Imitation learning with generalized task descriptions,” in *IEEE ICRA*, 2009, pp. 3968–3974.

[7] D. Kulic and Y. Nakamura, “Comparative study of representations for segmentation of whole body human motion data,” ser. IEEE IROS, 2009, pp. 4300–4305.

[8] N. Jetchev and M. Toussaint, “Task space retrieval using inverse feedback control,” in *ICML*, 2011, pp. 449–456.

[9] E. S. L. Ho, T. Komura, and C.-L. Tai, “Spatial relationship preserving character motion adaptation,” in *ACM SIGGRAPH*, 2010, pp. 33:1–33:8.

[10] T. Moulard, E. Yoshida, and S. Nakaoka, “Optimization-based motion retargeting integrating spatial and dynamic constraints for humanoid,” in *ISR*, 2013, pp. FA2–4.

[11] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, “Robot programming by demonstration,” in *Springer Handbook of Robotics*, 2008, pp. 1371–1394.

[12] Y. Nakamura, *Advanced robotics - redundancy and optimization*, 1991.

[13] K. Yamane and Y. Nakamura, “Natural motion animation through constraining and deconstraining at will,” *IEEE TVCG*, vol. 9, pp. 352–360, 2003.

[14] T. Sugihara, Y. Nakamura, and H. Inoue, “Realtime humanoid motion generation through zmp manipulation based on inverted pendulum control,” in *IEEE ICRA*, 2002, pp. 1404–1409.

[15] G. Venture, K. Ayusawa, and Y. Nakamura, “Motion capture based identification of the human body inertial parameters,” in *IEEE EMBS*, 2008, pp. 4575–4578.

[16] R. Diankov, “Automated construction of robotic manipulation programs,” Ph.D. dissertation, Carnegie Mellon University, Robotics Institute, 2010.

[17] D. Kulic, W. Takano, and Y. Nakamura, “Online segmentation and clustering from continuous observation of whole body motions.” *IEEE Transactions on Robotics*, vol. 25, no. 5, pp. 1158–1166, 2009.