



HAL
open science

Dynamic Motion Capture and Edition using a Stack of Tasks

Oscar Efrain Ramos Ponce, Layale Saab, Sovannara Hak, Nicolas Mansard

► **To cite this version:**

Oscar Efrain Ramos Ponce, Layale Saab, Sovannara Hak, Nicolas Mansard. Dynamic Motion Capture and Edition using a Stack of Tasks. IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS 2011), Oct 2011, Bled, Slovenia. p. 224-230. hal-00600959

HAL Id: hal-00600959

<https://hal.science/hal-00600959>

Submitted on 16 Jun 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Dynamic Motion Capture and Edition using a Stack of Tasks

Oscar E. Ramos, Layale Saab, Sovannara Hak and Nicolas Mansard
Gepetto Group, CNRS ; LAAS ; Université de Toulouse ; France
{oefrainr, Isaab, shak, nmansard}@laas.fr

Abstract—This paper presents a complete methodology to quickly reshape a dynamic motion demonstrated by a human and to adapt the dynamics of the human to the dynamics of the robot. The method uses an inverse dynamics control scheme with a quadratic programming optimization solver. The motion data recorded using a motion capture system is introduced into the control scheme as a reference posture task to be followed by the joints trajectory respecting the dynamic limitations as well as the contact constraints. The motion is further modified using arbitrary tasks to let the robot imitate the original motion more closely or to make voluntary changes for aesthetic reasons. The results show the method applied to the humanoid robot HRP-2 imitating a human "pop dance".

I. INTRODUCTION

Human motion imitation by a humanoid robot is a challenging task involving coordination, control and stabilization of the robot. Coordination problems are inherent to kinematically redundant robots, and humanoids are known to be highly redundant systems. Control difficulties arise from the complex humanoid tree-like structure as well as its unstable nature due to its vertical position. The robot must not only reproduce some task, but it should not fall while performing the task, keeping its dynamic balance. These constraints make imitation of captured motion a more complicated problem in robotics than in computer animation.

A classical approach for motion imitation in computer graphics is inverse kinematics [1], [2] since the characters usually do not possess physical dynamic characteristics. The main tools are related to motion editing and retargeting captured motions [3]. Even though sometimes dynamics is taken into account [4], its main objective is to give more realism to the animation rather than to make the motion feasible. The motion generated with methods for computer animation is not guaranteed to be suitable for a direct reproduction on a robot.

In robotics, capture and analysis of motion has become an active research area during these recent years to generate motion by imitation [5]. This imitation paradigm that lets the robot acquire the way of performing a task by observing human demonstrations has been called *learning from observation (LFO)* [6] and was initially introduced for robotic manipulators with the name *assembly plan from observation (APO)* [7]. The starting point is usually the motion acquired from a human expert using a motion capture system. Chinese Kungfu is an example of this type of imitation [8]. Also, Japanese traditional dancing has been replayed on a humanoid robot imitating the motion of a professional dancer [9], [10],

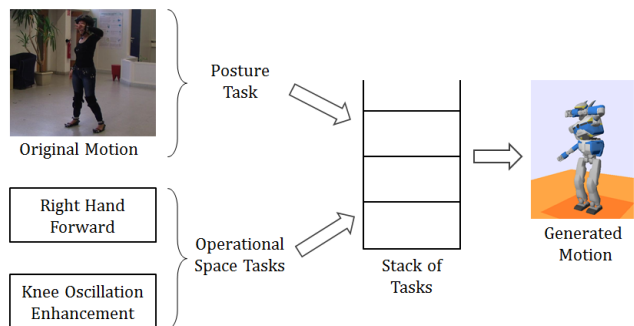


Fig. 1: Inverse Dynamics Cascade Scheme

[6]. Moreover, modifications to the upper body motion have been realized for the Japanese dancing [11]. Another motion sequence resembling a dance was presented in [12] adapting the captured joint trajectories to generate feasible motion in a robot, and robot modeling is recreated in [13] where walking and turning motion are created based on the motion obtained from a professional model.

Optimization is a classical solution for reshaping the captured motion before imitation. For example, upper body motion is produced solving a separate optimization problem for each frame of the motion in [14]. A constraint that uses trajectory optimization and filtering to preserve the main characteristics of the original motion and to respect at the same time the physical limitations of the humanoid robot is proposed in [15]. A similar approach is proposed in [5] but the optimization problem is reformulated using Lie algebra for the dynamic equations of motion. In this case, the ZMP is controlled using the cart table model [16].

In this paper, we propose to use a generic hierarchical optimization solver to solve simultaneously the dynamic reshaping and the motion edition. The considered solver consists of the inverse-dynamics control cascade proposed initially in [17] and extended in [18]. The flexibility of the scheme allows the addition of arbitrary tasks on the operational space that modify the trajectory of the joints, to generate a more similar motion, or to change some part of it, as Figure 1 shows.

This paper is organized as follows. Section II presents the dynamic model considering the contact constraints. In Section III we recall the inverse dynamics solver applied for the generation of motion by resolving a set of tasks. The tasks used for motion imitation are introduced in Section IV. Section V presents the method used to reshape the observed

human motion to the robot kinematics using an optimization algorithm. Finally, the results obtained in simulation for the humanoid robot HRP-2 imitating a dance motion are presented in Section VI.

II. DYNAMIC MODEL

A. Dynamic model with contact conditions

In this work, we consider a humanoid robot in contact with its environment. The full representation of the humanoid robot is given by the *generalized coordinates* $q = (x_b, q_a)$ with x_b as the position and orientation of the robot with respect to the world, and q_a as the vector including the n actuated joints. The arbitrary center of the body in contact (*e.g.* the body center of mass or the center of the joint attached to it) will be represented by x_c and the frame attached to it by $\{C\}$. By abuse, x_c will denote the 6D position (translation and orientation) of the frame $\{C\}$. The contact forces are then expressed in terms of the generalized 6D contact force $\phi_c = (f_c, \tau_c)$ written at $\{C\}$. Considering a single body in contact, the dynamic model is:

$$A\ddot{q} + b + J_c^T \phi_c = S^T \tau \quad (1)$$

where J_c is the contact Jacobian of x_c expressed at the same point as ϕ_c , A is the whole-body inertia matrix, b is the vector including Coriolis, centrifugal and gravity forces, τ is the actuated torque vector, and S is a matrix that selects the actuated joints. This can be generalized for more than two bodies in contact as shown later.

An additional non-holonomic restriction to be satisfied when two rigid bodies are in contact is that no relative motion should occur between them [19]: $\dot{x}_c = 0$ and $\ddot{x}_c = 0$ ¹. At the acceleration level, this is $\ddot{x}_c = J_c \ddot{q} + \dot{J}_c \dot{q} = 0$, which implies

$$J_c \ddot{q} = -\dot{J}_c \dot{q} \quad (2)$$

B. Force constraint for rigid planar contacts

1) *Unilateral model:* In the following, the only contact model considered is the rigid multi-planar contact. For each contact, at least three non aligned points must lie on the planar surface. Consider only one body in rigid contact and let the points belonging to the contact surface be p_i , $i = 1, \dots, n_p$ with $n_p \geq 3$. At each vertex p_i , the force is represented by $f_i = (f_{ix}, f_{iy}, f_{iz})$. The torque expressed at $\{C\}$ is ${}^c\tau_i = {}^c p_i \times f_i = {}^c \hat{p}_i f_i$, where ${}^c \hat{p}_i$ is the skew-matrix obtained from ${}^c p_i$, and ${}^c p_i$ is the point p_i with respect to the reference frame $\{C\}$. The total force at the center of contact is then $f_c = \sum_{i=1}^{n_p} f_i$ and the total torque is $\tau_c = \sum_{i=1}^{n_p} {}^c \hat{p}_i f_i$:

$$\phi_c = \begin{bmatrix} f_c \\ \tau_c \end{bmatrix} = \begin{bmatrix} I & I & \dots & I \\ {}^c \hat{p}_1 & {}^c \hat{p}_2 & \dots & {}^c \hat{p}_{n_p} \end{bmatrix} \begin{bmatrix} f_1 \\ \vdots \\ f_{n_p} \end{bmatrix}. \quad (3)$$

¹We consider rigid contact ($\ddot{x} = 0$, contact of class 6) which implies a minimum of three contact points. For other classes of contact (*e.g.* punctual contact) or non-rigid contact (*i.e.* \ddot{x} is obtained as a non-zero function of the exerted forces), the same structure can be used, but it is not considered here.

A sufficient condition to guarantee a rigid contact between a body of the robot and the planar surface is that the reaction force at the interface must be directed towards the robot; equivalently, the normal components f_{iz} of each contact force must be positive [17]:

$$f^\perp = (f_{1z}, f_{2z}, \dots, f_{n_p z}) \geq 0 \quad (4)$$

2) *Constraint reduction:* The contact can be represented by (3) and inequality (4). Using the planar hypothesis, it is possible to reduce the size of these conditions. Consider a reference frame $\{O\}$ as the projection of $\{C\}$ on the contact surface so that the Z axis of $\{O\}$ is normal to the surface, and let ${}^o c = (0, 0, {}^o c_z)$ be the origin of frame $\{C\}$ expressed with respect to $\{O\}$. The total force and torque with respect to the new frame are given by

$$\phi_o = \begin{bmatrix} f_o \\ \tau_o \end{bmatrix} = \begin{bmatrix} I & 0 \\ {}^o \hat{c} & I \end{bmatrix} \begin{bmatrix} f_c \\ \tau_c \end{bmatrix}. \quad (5)$$

Each contact point in the reference frame $\{O\}$ is ${}^o p_i = ({}^o p_{ix}, {}^o p_{iy}, 0)$, with ${}^o p_{iz} = 0$ as they also lie in the contact surface. Using (5) and the relation $\tau_o = \sum_{i=1}^{n_p} {}^o \hat{p}_i f_i$, it is straightforward to verify that:

$$\begin{bmatrix} f_{c_x} \\ f_{c_y} \\ f_{c_z} \\ -{}^o c_z f_{c_y} + \tau_{c_x} \\ {}^o c_z f_{c_x} + \tau_{c_y} \\ \tau_{c_z} \end{bmatrix} = \begin{bmatrix} \sum_i {}^o p_{iy} f_{ix} \\ \sum_i {}^o p_{ix} f_{iy} \\ \sum_i f_{iz} \\ \sum_i {}^o p_{iy} f_{iz} \\ -\sum_i {}^o p_{ix} f_{iz} \\ -\sum_i {}^o p_{iy} f_{ix} + \sum_i {}^o p_{ix} f_{iy} \end{bmatrix} \quad (6)$$

Since the rigid contact condition on the force only constrains the normal component f_{iz} , the first, second and sixth rows of (6) impose no restriction to the dynamic evolution and can be arbitrarily chosen. With these simplifications, the rigid-contact constraint including (4) and (6) is given by [18]:

$$[Q_{c1} \quad Q_{c2}] \begin{bmatrix} \phi_c \\ f^\perp \end{bmatrix} = 0 \quad \text{and} \quad f^\perp \geq 0 \quad (7)$$

with

$$Q_{c1} = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & {}^o c_z & 0 & -1 & 0 & 0 \\ -{}^o c_z & 0 & 0 & 0 & -1 & 0 \end{bmatrix} \quad (8)$$

$$Q_{c2} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ {}^o p_{1y} & {}^o p_{2y} & \dots & {}^o p_{n_p y} \\ -{}^o p_{1x} & -{}^o p_{2x} & \dots & -{}^o p_{n_p x} \end{bmatrix} \quad (9)$$

Multiple contact: The model presented above is valid for one body in contact with one plane. However, it is straightforward to extend it to multiple bodies in contact with multiple planes, by considering one couple of conditions (2) and (7) and one couple of variables Φ_c and f^\perp for each contact. In that case, the complete dynamic equation of the robot is:

$$A\ddot{q} + b + \sum J_c^T \phi_c = S^T \tau \quad (10)$$

The equations (2) and (7), considered one for each body in contact, complete the dynamic model.

ZMP-based stability condition: Considering frame $\{O\}$ in the interface between the supporting feet and the ground, and assuming that the robot is in contact with a single plane, the ZMP is computed as the average of the contact points ${}^o p_i$ weighted by the normal component of the contact forces f_{i_z} at each point:

$$p_{zmp} = \left(\frac{\sum_i^{n_p} {}^o p_{i_x} f_{i_z}}{\sum_i^{n_p} f_{i_z}}, \frac{\sum_i^{n_p} {}^o p_{i_y} f_{i_z}}{\sum_i^{n_p} f_{i_z}}, 0 \right) \quad (11)$$

The unilateral condition guarantees that $f^\perp \geq 0$, and therefore, the ZMP will exist inside the support polygon, ensuring the dynamic stability condition. Consequently, (7) ensures the fulfillment of the classical ZMP condition.

III. DYNAMIC INVERSION USING A STACK OF TASKS

The dynamic model presented above allows a dynamic inversion from a reference joint trajectory, that is, it finds the torques to exert on the robot joints given a reference trajectory \tilde{q}^* . However, it is often much straightforward to express the robot objectives in dedicated spaces, as proposed by the task function approach [20]. A classical way to compose a movement from a set of tasks is to introduce a strict hierarchy between tasks to ensure a safe behavior in case of conflict. This is a well known solution in kinematics [21] and in dynamics [22]. In [23] a hierarchical solver including inequalities was introduced for kinematics, and it was extended in [18] for dynamics. In the following, the principles of this solver are quickly recalled, and the classical tasks to be used are briefly introduced.

A. Generic QP cascade scheme

Consider a linear system with n linear inequalities (or equalities) $A_1 x \leq b_1, A_2 x \leq b_2, \dots, A_n x \leq b_n$. The system is solved for x by the hierarchical QP solver using the following procedure. The first stage is:

$$\begin{aligned} \min_{x, w_1} \|w_1\| \\ A_1 x - b_1 \leq w_1 \end{aligned} \quad (12)$$

where w_1 is a slack variable. The second stage minimizes the slack variable w_2 :

$$\begin{aligned} \min_{x, w_2} \|w_2\| \\ A_2 x - b_2 \leq w_2 \\ A_1 x - b_1 \leq w_1^* \end{aligned} \quad (13)$$

where w_1^* is the slack variable determined in the first stage (12). In case of conflict between the constraints, the slack variable w_2 allows to violate $A_2 x \leq b_2$, giving a greater priority to $A_1 x \leq b_1$. The subsequent stages are formulated in a similar way, and the n -th stage can be written as:

$$\begin{aligned} \min_{x, w_n} \|w_n\| \\ A_n x - b_n \leq w_n \\ A^a x - b^a \leq w^{a*} \end{aligned} \quad (14)$$

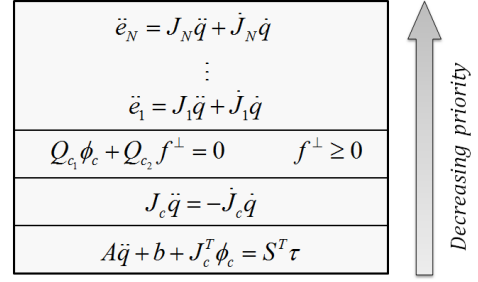


Fig. 2: Dyanmic QP Cascade Scheme.

where $A^a = (A_1, \dots, A_{n-1})$, $b^a = (b_1, \dots, b_{n-1})$ and $w^{a*} = (w_1^*, \dots, w_{n-1}^*)$. The augmented slack variable w^{a*} contains the slack variables obtained in the previous $n - 1$ stages. This hierarchical QP is denoted as (12) \prec (13) \prec \dots \prec (14).

B. Operational tasks

For any observable $s(q)$ of the robot configuration, a desired position and orientation s^* can be specified without loss of generality by the task $e = s - s^*$. A usual choice for the task rate of change is $\dot{e} = -\lambda_p e - \lambda_v \dot{e}$ with $\lambda_p > 0$ and $\lambda_v = 2\sqrt{\lambda_p}$ to make the error decrease exponentially. At the acceleration level, the relation between the task space and the joint space is:

$$J\ddot{q} = \ddot{e} - \dot{J}\dot{q} \quad (15)$$

where $J = \frac{\partial e}{\partial q}$ is the task Jacobian.

C. QP for the dynamic inversion

With the notation presented, the generic hierarchical QP proposed in [18] for the resolution of the inverse dynamics using the task space is written as (1) \prec (2) \prec (7) \prec (15), and is shown in Figure 2. To deal with all these constraints, the optimization vector determined by the QP solver is $x = (\ddot{q}, \tau, \phi_c, f^\perp)$. The operational space dynamic tasks correspond to the accelerations of the controlled task e , which is \ddot{e} , and Figure 2 shows the case of N tasks e_1, \dots, e_N . These tasks can be added or removed at any time and with any priority. Additionally, the contact constraints can also be added or removed at any time making the whole cascade very flexible when handling whole-body motion.

IV. TASK SET FOR IMITATION

The solver detailed previously can be used for various kinds of tasks. Examples can be found in [18]. In the frame of imitation and edition, we have mainly used four types of tasks:

- The first task regulates the posture (actuated part of the robot configuration) to follow the motion of the demonstrator.
- The other tasks were used to edit the motion and include: the control of the position and orientation of an operational point, the control of the trajectory of a particular joint, and the control of the position on the ground plane of one foot sliding.

These tasks are described in detail in this section.

A. Posture Task

The desired trajectory for the joint angles is obtained using the motion capture system (as it will be explained in Section V). This acquired motion can be directly replayed on a human-like character used in the field of computer graphics. However, it is not straightforward to regenerate the data by a robot as it does not consider the dynamic model or the constraints associated with it. In this case, the inverse dynamics control scheme can be used as it considers the contact constraints and the dynamic stability conditions. Particularly, a task that follows the joints evolution as closely as possible while satisfying the prior dynamic conditions is implemented and will be called the *posture task*. Let $q_{jk} = (q_j, \dots, q_k) \in \mathbb{R}^{k-j+1}$ with $k, j \in \mathbb{N}$, $1 \leq j, k \leq n$ and $k \geq j$ represent the vector containing the angular values that need to be controlled, and q_{jk}^* the corresponding desired configuration. A posture task is expressed as $e = q_{jk} - q_{jk}^*$. At the velocity level, this will lead to $\dot{e} = J_{jk}\dot{q}$ with the Jacobian $J_{jk} \in \mathbb{R}^{(k-j+1) \times (6+n)}$ selecting only the desired joints:

$$J_{jk} = \begin{bmatrix} 0_{(k-j+1) \times (6+j-1)} & I_{k-j+1} & 0_{(k-j+1) \times (n-k)} \end{bmatrix} \quad (16)$$

where $I_{k-j+1} \in \mathbb{R}^{(k-j+1) \times (k-j+1)}$ is the identity matrix and $0_{k_1 \times k_2} \in \mathbb{R}^{k_1 \times k_2}$ are zero matrices. In the special case of controlling all the actuated joints, the identity matrix will have its largest size, lying in $\mathbb{R}^{n \times n}$. Finally, the reference acceleration is given as a tracking of the joint trajectory obtained by the motion capture system, using a PD controller:

$$\ddot{q}_{jk}^* = -\lambda_p(q_{jk} - q_{jk}^*) - \lambda_v(\dot{q}_{jk} - \dot{q}_{jk}^*) \quad (17)$$

with λ_p and λ_v defined as before.

B. Arbitrary Edition of the Generated Motion

The posture task reproduces the desired motion at the joint level satisfying the dynamic constraints. However, the PD acts as a low-pass filter, generating non desired movements at some points or erasing some delicate or very dynamic movements, typically due to fast oscillatory motions. Nevertheless, the structure of the stack of tasks can be used to overcome this problem by adding operational tasks that enhance or even modify the original motion. Moreover, the priorities of the tasks can be modified to better achieve the desired motion.

An operational task controls directly the position and/or orientation of different operational points. In our case, we consider the head, chest, waist, feet and hands of the robot as the controlled end-effectors. Each operational point $x_i = (p_{ix}, p_{iy}, p_{iz}, \varphi_{ix}, \varphi_{iy}, \varphi_{iz}) \in \mathbb{R}^6$ is expressed in Cartesian coordinates to specify its position and the roll, pitch, yaw angles for its orientation. Despite this 6D representation, we are free to control either both position and orientation, or only some axis of interest. For this sake, a diagonal selection matrix S_x is defined as:

$$S_x = \text{diag}(s_1, s_2, s_3, s_4, s_5, s_6) \quad (18)$$

where s_j is a binary, 1 or 0, the former one to control that particular element of position or orientation, and the latter

one to leave it unconstrained. Thus, the differential relation is expressed as:

$$S_x \dot{e}_i = S_x J \dot{q} \quad (19)$$

and the reference x_i^* to be followed only considers the controlled elements of position and/or orientation.

We use this operational task to edit the motion in two cases:

- *Specification of target points*: A new desired target for a chosen operational point can be specified without defining the desired trajectory to reach it. This point can be determined using forward kinematics applied on a certain operational point to compute its position from the kinematic optimization, which will be described in Section V, or it can be arbitrarily set.
- *Specification of a trajectory*: Let the trajectory for the operational point x be called $x_o(t)$. The new trajectory $x_n(t)$ that will be set as the desired trajectory for that operational point will be $x_n(t) = x_o(t) + x_m(t)$, where $x_m(t)$ is the trajectory modification that can be done on any of the six degrees of freedom of x . This trajectory modification $x_m(t)$ can be time varying or constant, according to the requirements.

It is important to point out that the added operational task must have higher priority than the joints posture task it would interfere with. For instance, if an operational task is added to the hand, the priority of the arm posture task must be reduced. Alternatively, the task can be removed, but it is preferred to be kept, as it will serve as a “guide” for the new trajectory. If these priority considerations are not taken into account, the desired motion would be blinded by the solutions satisfying the posture task with higher priority, then, the desired effect would not be achieved. Other tasks (typically, the gaze) could have been considered, even if not meaningful in our case of pop dancing.

C. Foot Sliding

At some points, the observed motion revealed slight changes in the contact which becomes not exactly rigid. Thus, to let a foot slide, the rigid contact constraints stating $\dot{x}_c = 0$ and $\ddot{x}_c = 0$ can be relaxed constraining the motion to the horizontal XY plane. For the contact i , this restriction at the velocity level can be formulated as $v_z = 0$, $\omega_x = 0$, $\omega_y = 0$ allowing the other velocity elements to take arbitrary values. This guarantees no translation on the vertical axis Z or rotation about it. To account for these constraints, the dynamic model would need to be reformulated. Alternatively, another simpler and more flexible approach is to remove the contact constraint and add a task that restricts the motion in Z direction, imposing no restrictions to the X or Y axis, which will limit the motion to the XY plane. Considering x^* as the desired task, this particular case of the 6D task (19) is $x^* = (k_x, k_y, H_z, 0, 0, k_{rz})$ and $S_x = \text{diag}(0, 0, 1, 1, 1, 0)$. where S_x is the selection matrix defined in (18), $k_x, k_y, k_{rz} \in \mathbb{R}$ are arbitrary values and H_z is the known height of the foot operational point with respect to the world frame. When the foot is in contact with the ground, this constraint restricts the

height of the foot to remain constant, and the rotation about the X and Y axis to be null.

This task does not exactly correspond to a sliding, since sliding can accept forces orthogonal to the motion. The proposed solution is thus more restrictive than necessary. However, the visual effect would be the same, and it will keep the motion feasible and dynamically consistent.

V. MULTIBODY MOTION ACQUISITION

Our motion capture system is a 10-camera with marker tracking system [24]. It provides the 6D position, translation and orientation, of a set of unconstrained bodies in space, typically the limbs of the demonstrator, at a frequency of 200 Hz and with a precision of 2 mm. For this set of 6D position, we need to recompute the joint position of the demonstrator, knowing its geometric model. This is achieved by solving a non linear optimization problem where the geometric model of the demonstrator is known, and the joint position is the optimization variable.

A. Optimization problem formulation

The geometric model gives the position and orientation of each joint q_i with respect to the robot world reference frame $\{W\}$, as the transformation matrix ${}^W T_{q_i}(q)$ which is a function of the joint vector q . Then, the problem of finding the suitable joint values $q^*(t)$ for the robot at time t can be reduced to minimizing the difference between these transformation matrices as:

$$q^*(t) = \arg \min_q \sum_i^n {}^W T_{q_i}^*(t) \ominus {}^W T_{q_i}(q) \quad (20)$$

$$s.t. \quad \underline{q}_i \leq q_i \leq \bar{q}_i \quad (21)$$

where ${}^W T_{q_i}^*(t)$ is the 6D position of the body i given by the motion capture system, $\underline{q}_i, \bar{q}_i$ are the minimum and maximum angular values of a joint i , respectively, and the condition to satisfy the joint limits for every joint $i \in \{1, \dots, n\}$ at every optimization process has been added. The symbol \ominus represents a distance operator in the matrix group. We used a weighted norm of the translation-axis-angle vectorial writing of the transformation matrices given as follows. The homogeneous matrices can be decomposed in rotation and translation parts:

$${}^W T_{q_i}^*(t) = \begin{bmatrix} R_{q_i}^*(t) & p_{q_i}^*(t) \\ 0 & 1 \end{bmatrix} \quad (22)$$

$${}^W T_{q_i}(q) = \begin{bmatrix} R_{q_i}(q) & p_{q_i}(q) \\ 0 & 1 \end{bmatrix}. \quad (23)$$

For the part corresponding to the position, the difference to minimize is directly represented by the $L - 2$ norm as $\| p_{q_i}^*(t) - p_{q_i}(q) \|_2$. For the orientation part, the difference is measured by the rotation angle θ_i , which is obtained using the axis/angle representation of the product $R_{d_i} = R_{q_i}(q)R_{q_i}^*(t)^{-1} \in SO(3)$. If the rotation matrices are close

enough, the angle of R_{d_i} about the arbitrary axis is very small. Representing the product of the rotation matrices as

$$R_{d_i} = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix} \quad (24)$$

the angle will be

$$\theta_i = \text{atan2}(s_a, c_a) \quad (25)$$

where

$$s_a = \frac{\sqrt{(n_y - o_x)^2 + (n_z - a_x)^2 + (o_z - a_y)^2}}{2} \quad (26)$$

$$c_a = \frac{n_x + o_y + a_z - 1}{2}. \quad (27)$$

The angle θ_i is always between 0 and π . Finally, the operator \ominus in (20) is given by:

$${}^W T_{q_i}^*(t) \ominus {}^W T_{q_i}(q) = w_{p_i} \| p_{q_i}^*(t) - p_{q_i}(q) \|_2 + w_{o_i} \theta_i \quad (28)$$

where w_{p_i} is the weight corresponding to the position part of the joint i and $w_{o_i} = 1 - w_{p_i}$ is the weight for the orientation part. The weights are included to provide more flexibility to the model and are experimentally determined to give more priority to one of the parts.

The ${}^W T_{q_i}(q)$ should be computed from the geometric model of the demonstrator. In our case, we have used directly the model of the robot to compute ${}^W T_{q_i}(q)$. The optimization method has proved to be robust enough to handle the approximation, and the dynamic solver is sufficient to recover all the resulting noise and inaccuracy.

B. Calibration of the body frame

In (20) it is supposed that the same frames are used to express both the motion capture observation and the geometric model. This is of course not the case in practice, since the motion-capture system uses its own arbitrary frames to express the multi-body motion. The observation of the motion capture has to be then reformulated in the proper frame. To this end, the person whose motion wants to be recovered starts with a position that is well known for the robot. Using this known configurations, the transformation matrices between each node m_i and the corresponding joint q_i in the robot is obtained from a classical ‘‘calibration’’ step and represented by ${}^{m_i} T_{q_i}$, which remains constant for all the process as long as the markers do not have relative motion with respect to the body they are attached to. This matrix considers the differences in orientation between the markers and the frames defined in the kinematic model of the robot for each joint, as well as the differences in the segment lengths of the robot and the dancer. Another matrix that is obtained during the calibration process is the one relating the origin of the motion capture system $\{W_m\}$ and the robot reference frame $\{W\}$ represented by ${}^W T_{W_m}$. Let the position and orientation of each node m_i with respect to the motion capture reference frame be represented by the homogeneous transformation ${}^{W_m} T_{m_i}(t)$ which varies in time according to the motion to be imitated. With these

transformation matrices, the desired configuration of the robot joint i is:

$${}^W T_{q_i}^*(t) = {}^W T_{W_m} {}^{W_m} T_{m_i}(t)^{m_i} T_{q_i} \quad (29)$$

VI. RESULTS

To validate the proposed method, the motion of a dancer was acquired using the motion capture system, which provides the position and orientation of the frames associated with each node. The motion was retargeted to a modified-HRP2 model (*ie* we added one degree of freedom to both chest and neck joints, obtaining a kinematic structure closer to the one of HRP4, to obtain a nicer final motion), and edited to corrected the retargeting error and introduced new undemonstrated features.

The kinematic optimization was applied to obtain the corresponding joint trajectories, which were then validated dynamically using the posture task in the inverse dynamics solver. The joints trajectory is first obtained from the motion capture, and it is, of course, not stable nor dynamically consistent as Figure 3b shows. A first dynamic motion is then obtained using only the joint trajectory as references. This motion is displayed in Figure 3c. The motion is stable; however, the geometric and dynamic retargeting have lost some data and produced some errors compared to the initial demonstrated trajectory. Three editions were thus applied:

- The knee oscillations (smoothed by the PD) were enhanced.
- The right hand motion was corrected.
- An additional motion of the left foot consisting in sliding on the ground, which was not present in the initial demonstration, was introduced.

The sequence showing the time moments when these modifications were added is shown in Figure 4. The following subsections provide more details of these three modifications.

A. Knee Oscillation

The knees constitute a particular case as the dancer permanently moved them but at the dynamic level this motion was strongly weakened. To correct this problem we referred to the task approach where the motion of the knee joint was analyzed. Figure 5a shows the joint evolution obtained after the kinematic optimization for the right knee. Between iterations 2000 and 2800, and between 6200 and 7300, the motion of the joint is oscillatory and those are the moments corresponding directly to the observed motion at the dancer's right knee. A scalogram using the Gaussian wavelet was constructed and is shown in Figure 5b, where it is observed (in red circle) that there are salient frequencies at those points. It was determined that the scale a corresponding to the maximum values at the desired positions is 36. The frequency and the scales are related by $f = \frac{f_s f_w}{a}$, where f_s is the sampling frequency and f_w is the center frequency of the wavelet. For the Gaussian derivative of order 4, $f_w = 0.5$, and considering that the sampling frequency used during the acquisition is 200 Hz, the resulting frequency is 2.7 Hz. Then, a task on the knee was added at that frequency to resemble more the motion.

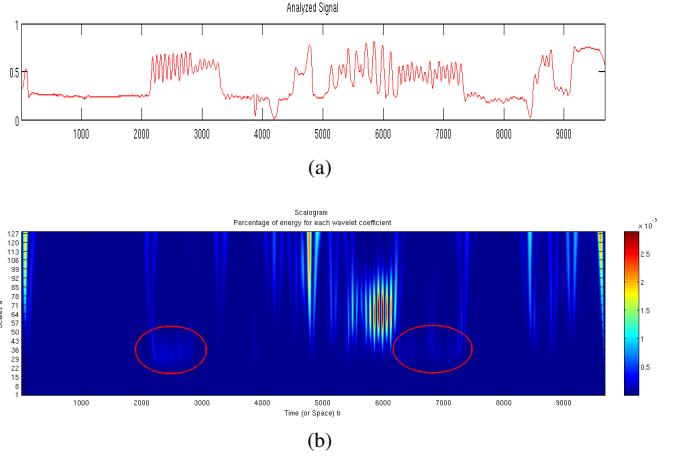


Fig. 5: Scalogram of the right knee joint evolution

Figure 6a shows the evolution of the joint at the right knee (joint number 4) in the joint space as a function of the iteration time, and Figure 6b shows the evolution in the X axis of the operational space. Note that there is a difference in the iteration time of Figure 5 and Figure 6 as the former one corresponds to the kinematic optimization of the motion sampled at 200Hz and the latter corresponds to the dynamic level that runs at 1000Hz (the iteration relation is 5 to 1). The red line shows the evolution of the joint when there is only the posture task in the leg, and the blue line shows the evolution when the operational task is added to the knee, but both lines are obtained using the dynamic control scheme. Even though the joint is not directly controlled, it is observed in Figure 6a that with the operational task addition, the joint at the knee presents an oscillation with higher amplitude, whereas with only the task posture the oscillation is weak. The results in the X axis of the operational space show a clear consistent oscillation with similar amplitude where the knee task was added. Both the joint space and the operational space show the effect of the task at the knee.

B. Right hand motion

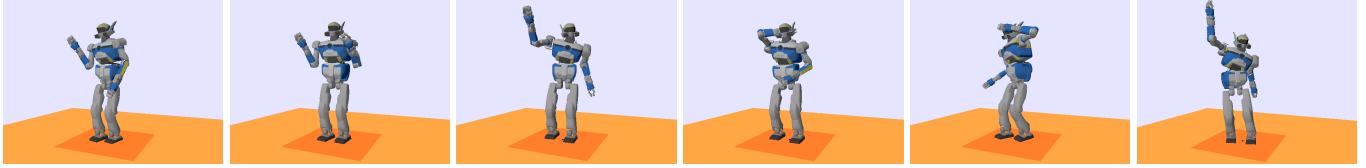
The fast up and down motion of the right arm was also smoothed as a consequence of the PD used. This was especially noted when the arm could not reach the upper positions that the dancer performed. Then, an operational task to raise more the arm was introduced. The result is shown in Figure 7. The trajectory of the right arm in the Z axis using only the posture task is shown in red, whereas the trajectory with the operational correction is shown in blue. The corrected trajectory, improved the upper positions of the right hand. The task in the right hand was also used to avoid the auto collision of the hand with the head, as can be noticed of Fig. 3, fourth thumbnails.

C. Foot sliding introduction

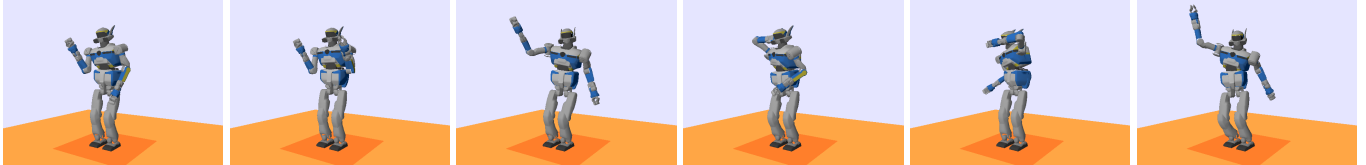
We artificially introduced a sliding movement of the right foot, to prove that extra features can be added as desired.



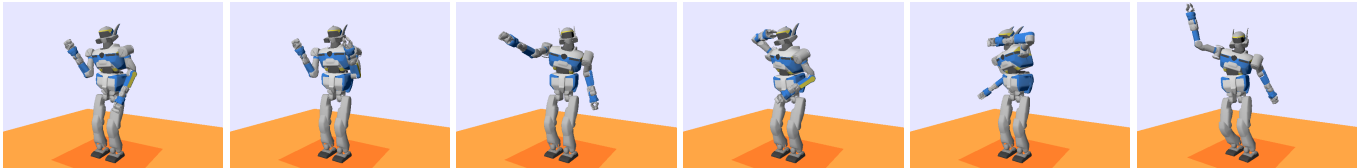
(a) Motion performed by the dancer (<http://homepages.laas.fr/nmansard/humanoid11-ramos-mocap/initial.avi>)



(b) Motion obtained with the geometric model (<http://homepages.laas.fr/nmansard/humanoid11-ramos-mocap/geometric.avi>)



(c) Motion after the posture task (<http://homepages.laas.fr/nmansard/humanoid11-ramos-mocap/dynamic.avi>)



(d) Final motion (<http://homepages.laas.fr/nmansard/humanoid11-ramos-mocap/final.avi>)

Fig. 3: Results for the robot imitating the dance performed by a human.

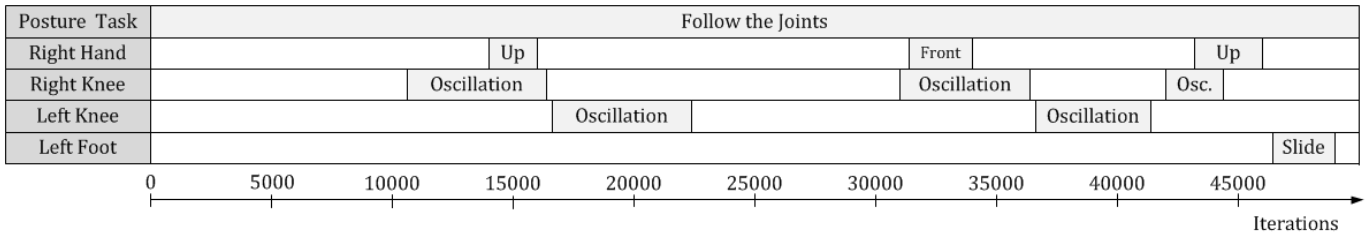
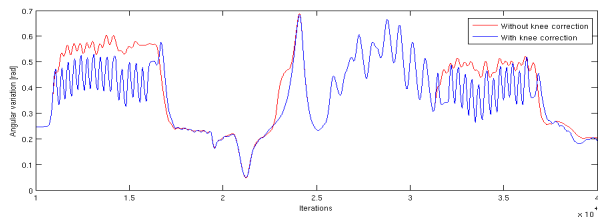


Fig. 4: Task sequence

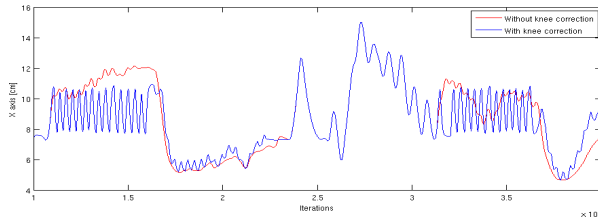
To introduce the sliding effect on a foot, the ZMP of the motion obtained using only the dynamic posture was analyzed. Figure 8 shows in blue the trajectory of the ZMP in the Y axis as a function of the time. The wide red lines show the limits of the support polygon in the Y axis, and the dashed red lines show the boundaries of the foot (inside the support polygon). Between the iteration times 46940 and 48310, the ZMP lies completely in the area corresponding to the right foot. Then, it was at this time that the sliding task was introduced for the left foot, guaranteeing the dynamic stability of the robot, as the sliding foot cannot be considered anymore part of the support polygon.

VII. CONCLUSION

A method for the imitation of whole-body motion for humanoid robots has been presented. The contribution of this work is to propose a complete methodology to quickly reshape a dynamic motion demonstrated by a human expert, adapt the dynamics of the human body to the own dynamics of the robot and modify or edit as desired the initial motion to introduce extra features that were not demonstrated. It allows to build complex dynamic behaviors, based on a composition of tasks and constraints that are used as basic bricks for motion generation. The method was satisfactorily applied to the imitation of dancing motion, but it can be, in general, applied to the imitation of any type of motion. The obtained motion is dynamically consistent, and could be directly applied on the real humanoid robot.



(a) Joint Space (joint number 4)



(b) Task Space (X axis)

Fig. 6: Temporal evolution of the right knee

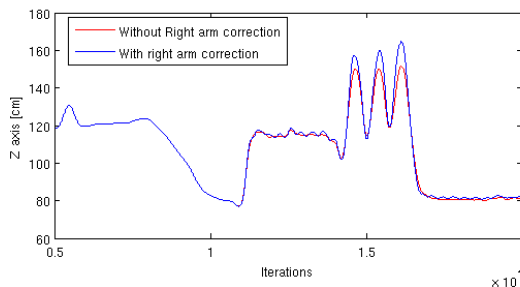


Fig. 7: Right hand evolution in the operational space (Z axis).

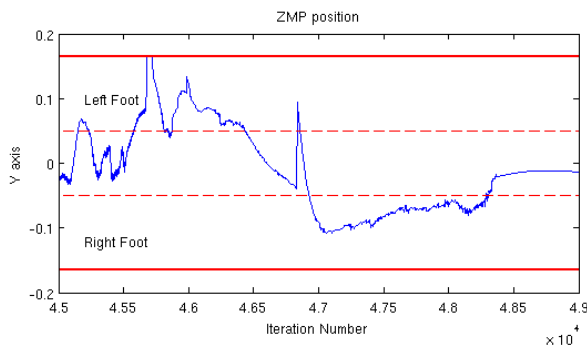


Fig. 8: Evolution of the ZMP with only the posture task

REFERENCES

- [1] M. Gleicher, "Retargetting motion to new characters," in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques (SIGGRAPH 98)*, Orlando, FL, USA, July 1998, pp. 33–42.
- [2] K. Grochow, S. Martin, A. Hertzmann, and Z. Popović, "Style-based inverse kinematics," *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3, pp. 522–531, August 2004.
- [3] K. Yamane and Y. Nakamura, "Natural motion animation through constraining and deconstraining at will," *IEEE Transactions on visualization and computer graphics*, vol. 9, no. 3, pp. 352–360, July 2003.
- [4] Z. Popovic, "Editing dynamic properties of captured human motion," in *IEEE International Conference on Robotics and Automation (ICRA'00)*, San Francisco, CA, USA, April 2000.
- [5] W. Suleiman, E. Yoshida, F. Kanehiro, J. Laumond, and A. Monin, "On human motion imitation by humanoid robot," in *IEEE International Conference on Robotics and Automation (ICRA 2008)*, Pasadena, California, USA, May 2008, pp. 2697–2704.
- [6] 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," *The International Journal of Robotics Research*, vol. 26, no. 8, p. 829, 2007.
- [7] K. Ikeuchi and T. Suehiro, "Toward an assembly plan from observation. Task recognition with polyhedral objects," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 3, pp. 368–385, 1994.
- [8] X. Zhao, Q. Huang, Z. Peng, and K. Li, "Kinematics mapping and similarity evaluation of humanoid motion based on human motion capture," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, Sendai, Japan, September 2004, pp. 840–845.
- [9] S. Nakaoka, A. Nakazawa, K. Yokoi, H. Hirukawa, and K. Ikeuchi, "Generating whole body motions for a biped humanoid robot from captured human dances," in *IEEE International Conference on Robotics and Automation (ICRA'03)*. Taipei, Taiwan: IEEE, September 2003, pp. 3905–3910.
- [10] S. Nakaoka, A. Nakazawa, K. Yokoi, and K. Ikeuchi, "Leg motion primitives for a dancing humanoid robot," in *IEEE International Conference on Robotics and Automation (ICRA'04)*, USA, April 2004.
- [11] T. Shiratori, S. Kudoh, S. Nakaoka, and K. Ikeuchi, "Temporal scaling of upper body motion for sound feedback system of a dancing humanoid robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007)*, San Diego, USA, October 2007, pp. 3251–3257.
- [12] N. Pollard, J. Hodgins, M. Riley, and C. Atkeson, "Adapting human motion for the control of a humanoid robot," in *IEEE International Conference on Robotics and Automation (ICRA'02)*, Washington D.C., USA, May 2002, pp. 1390–1397.
- [13] K. Miura, M. Morisawa, S. Nakaoka, F. Kanehiro, K. Harada, K. Kaneko, and S. Kajita, "Robot motion remix based on motion capture data towards human-like locomotion of humanoid robots," in *9th IEEE-RAS International Conference on Humanoid Robots*, Paris, France, December 2009, pp. 596–603.
- [14] A. Safonova, N. Pollard, and J. Hodgins, "Optimizing human motion for the control of a humanoid robot," *Proc. Applied Mathematics and Applications of Mathematics*, 2003.
- [15] M. Ruchanurucks, S. Nakaoka, S. Kudoh, and K. Ikeuchi, "Humanoid robot motion generation with sequential physical constraints," in *IEEE International Conference on Robotics and Automation (ICRA 2006)*, Orlando, FL, USA, May 2006, pp. 2649–2654.
- [16] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *IEEE International Conference on Robotics and Automation (ICRA'03)*, Taipei, Taiwan, September 2003.
- [17] L. Saab, N. Mansard, F. Keith, J. Fourquet, and P. Soueres, "Generation of Dynamic Motion for Anthropomorphic Systems under Prioritized Equality and Inequality Constraints," in *IEEE International Conference on Robotics and Automation (ICRA'11)*, Shanghai, May 2011.
- [18] L. Saab, O. Ramos, N. Mansard, P. Soueres, and J.-Y. Fourquet, "Generic Dynamic Motion Generation with Multiple Unilateral Constraints," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2011) "submitted"*, San Francisco, USA, September 2011.
- [19] K. Yamane and Y. Nakamura, "Dynamics filter-concept and implementation of online motion generator for human figures," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 3, pp. 421–432, 2003.
- [20] C. Samson, B. Espiau, and M. Borgne, *Robot control: the task function approach*. Oxford University Press, 1991.
- [21] B. Siciliano and J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *IEEE International Conference on Advanced Robotics (ICRA'91)*, Pisa, Italy, June 1991, pp. 1211–1216.
- [22] L. Sentis and O. Khatib, "Control of free-floating humanoid robots through task prioritization," in *IEEE International Conference on Robotics and Automation (ICRA 2005)*, Barcelona, Spain, April 2005, pp. 1718–1723.
- [23] A. Escande, N. Mansard, and P. Wieber, "Fast resolution of hierarchized inverse kinematics with inequality constraints," in *IEEE International Conference on Robotics and Automation (ICRA 2010)*, Anchorage, USA, May 2010, pp. 3733–3738.
- [24] "Motion analysis."