2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

On Human Motion Imitation by Humanoid Robot

Wael Suleiman*, Eiichi Yoshida†, Fumio Kanehiro†, Jean-Paul Laumond* and André Monin*
IS/AIST-ST2I/CNRS Joint Japanese-French Robotics Laboratory (JRL)
* LAAS - CNRS, university of Toulouse
7 Avenue du Colonel Roche, 31077 Toulouse, France
{suleiman, jpl, monin}@laas.fr
† Intelligent Systems Research Institute, National Institute
of Advanced Industrial Science and Technology (AIST)
AIST Central 2, Umezono 1-1-1, Tsukuba, 305-8568 Japan
{e.yoshida, f-kanehiro}@aist.go.jp

Abstract— **In this paper, the imitation of human captured motions by a humanoid robot is considered. The main objective is to reproduce an imitated motion which should be as close as possible to the original human captured motion. To achieve this goal, the imitation problem is formulated as an optimization problem and the physical limits of the humanoid robot are considered as constraints. The optimization problem is then solved recursively by using an efficient dynamics algorithm, which allows the calculation of the gradient function with respect to the control parameters analytically. The simulation results using OpenHRP platform, which is a dynamical simulator for humanoid robot motions, have pointed out that the imitated motions preserve the salient characteristics of the original human captured motion. Moreover the optimization procedure converges well thanks to the analytical calculation of the gradient function.**

I. Introduction

Imagining a humanoid robot collaborates with humans to execute some daily tasks is now reality. Actually, the ability of humanoid robots to execute complex tasks increases rapidly. In order to increase the autonomous behavior of humanoid robots as well as improving their reactivity, the humanoid robot should be able to imitate human motions.

In recent years, the imitation of human motions by humanoid robots was an active research field. Pollard *et al* [1] have proposed a method to transform a dance captured motion to a motion that the humanoid robot can execute. Nakaoka *et al* [2] have realized a whole body control of humanoid robot to imitate Jongara-Bushi dance that is a traditional Japanese folk dance. To maintain the dynamical stability of humanoid robot, they control the trajectory of Zero Moment Point (ZMP) [3] to be inside the polygon of support. Safonova *et al* [4] use also a pre-recorded human motion to generate optimal motion of the upper body of Sarcos humanoid robot. The function to be minimized is the difference between the recorded and executed motion by the robot. However, the previous methods do not consider some physical limits of humanoid robot, e.g. torque limits.
Ruchanurucks *et al* [5] have proposed a method to optimize upper body motion of humanoid robot in order to imitate a human record motion. Their objective function preserves the main characteristics of the original motion, and at the same time it respects the physical constraints of the humanoid robot. However, the authors have mentioned that the resulting trajectories would meet the latter limits while the former limits are often violated. This is because their method considers the velocity and force constraints separately.

In this paper, our objective is to generate a motion within the humanoid physical capabilities from a human captured motion. The physical limits are the angle, joint velocity and torque limits of the humanoid robot. We focus on the imitation of upper body motion. On the other hand, the motion of lower body can be efficiently generated using leg motion primitives [6], [7]. Concerning the dynamical stability of humanoid robot, it can be guaranteed by controlling the ZMP trajectory [8].

The main contribution of this paper is providing an optimization framework to generate the upper body motion of humanoid robot from human captured motions. The generated motions imitate the original human captured motion, and at the same time they respect the physical limits of humanoid robot.

The remainder of this paper is organized as follows. In Section II the imitation problem is formulated. In Section II a pre-processing procedure for human captured motion is explained. An overview of the algorithm of recursive multibody dynamics is given in Section IV. In Section V the optimization problem is reformulated using the notations of recursive multibody algorithm which is explained in the previous section. In Section VI discretizing the configuration space and solving the optimization problem are explained. Some experimental results are given in Section VII and Section VIII concludes the paper.

II. Imitation problem formulation

The kinematic structure of the humanoid robot HRP-2 [9] is given in Fig. 1. In this structure the degree of freedoms are presented by cylinders. The structure contains 30 degree of freedoms.

The inputs of the imitation procedure are human captured motions. These motions are provided by a motion capture system as a skeleton of virtual actor and a sequence of the angular values of the virtual actor's joints.

978-1-4244-1647-9/08/$25.00 ©2008 IEEE.

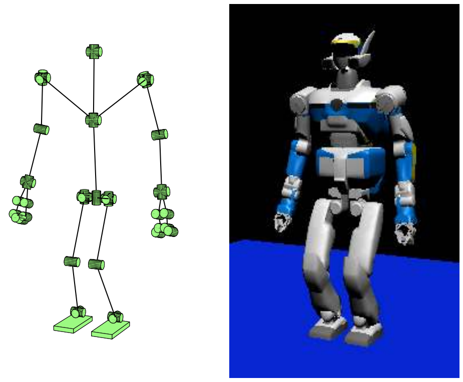


Fig. 1. Description of HRP-2 kinematic structure.

Generally, the virtual actor has more degree of freedoms than the humanoid robot and as well its links lengths are different from those of humanoid robot.

The imitation problem, from a kinematic point of view, is well known in computer graphics and it is called motion retargeting [10]. The motion retargeting problem is formulated as follows

$$\min_{\mathbf{q_t}} \int_{t_0}^{t_f} \left\{ \left(\mathbf{q_t} - \mathbf{q_t^c}\right)^T \left(\mathbf{q_t} - \mathbf{q_t^c}\right) + \sigma \left(P_t - P_t^c\right)^T \left(P_t - P_t^c\right) \right\} dt$$

subject to

$$\begin{cases} \mathbf{q_{t_0}} = \mathbf{q_0} \\ \mathbf{q_{t_f}} = \mathbf{q_f} \\ \mathbf{q}^- \leq \mathbf{q_t} \leq \mathbf{q}^+ \end{cases} \tag{1}$$

where σ is a user defined constant. $\mathbf{q_t}$ and $\mathbf{q_t^c}$ are the joint positions of the upper body of the humanoid robot and the virtual actor respectively. P_t and P_t^c are the Cartesian positions of the hands and the head of the humanoid robot and virtual actor in the pelvis frame. It is defined as follows

$$P_t = \begin{bmatrix} P_{\text{head}} \\ P_{\text{right hand}} \\ P_{\text{left hand}} \end{bmatrix} \tag{2}$$

Note that if the lengths of the virtual actor's links are largely different from those of humanoid robot, the vector P_t^c can be scaled to fit the humanoid robot size. $\mathbf{q}^-$ and $\mathbf{q}^+$ denote the minimal and maximal values of the vector $\mathbf{q}_t$ respectively.

The retargeting problem has been extremely studied in computer graphics during the last years, and we have actually many commercial graphic softwares that can solve it efficiently.

However, in the motion imitation by a humanoid robot additional difficulties arise such as the joints velocity and the torque limits.
By taking into account those additional constraints, the motion imitation problem becomes

$$\min_{\mathbf{q_t}, \dot{\mathbf{q}}_\mathbf{t}, \ddot{\mathbf{q}}_\mathbf{t}} \int_{t_0}^{t_f} \left\{ \left(\mathbf{q_t} - \mathbf{q_t^c}\right)^T \left(\mathbf{q_t} - \mathbf{q_t^c}\right) + \sigma \left(P_t - P_t^c\right)^T \left(P_t - P_t^c\right) \right\} dt$$

subject to

$$\begin{cases} \mathbf{M}\left(\mathbf{q_t}\right) \ddot{\mathbf{q}}_\mathbf{t} + \mathbf{C}\left(\mathbf{q_t}, \dot{\mathbf{q}}_\mathbf{t}\right) = \tau_t \\ \mathbf{q_{t_0}} = \mathbf{q_0}, \dot{\mathbf{q}}_{\mathbf{t_0}} = \mathbf{0}, \ddot{\mathbf{q}}_{\mathbf{t_0}} = \mathbf{0} \\ \mathbf{q_{t_f}} = \mathbf{q_f}, \dot{\mathbf{q}}_{\mathbf{t_f}} = \mathbf{0}, \ddot{\mathbf{q}}_{\mathbf{t_f}} = \mathbf{0} \\ \tau^- \leq \tau_t \leq \tau^+ \\ \dot{\mathbf{q}}^- \leq \dot{\mathbf{q}}_\mathbf{t} \leq \dot{\mathbf{q}}^+ \\ \mathbf{q}^- \leq \mathbf{q_t} \leq \mathbf{q}^+ \end{cases} \tag{3}$$

where τ_t is the vector of the applied torques on the humanoid robot's joints. $\mathbf{M}\left(\mathbf{q_t}\right) \ddot{\mathbf{q}}_\mathbf{t} + \mathbf{C}\left(\mathbf{q_t}, \dot{\mathbf{q}}_\mathbf{t}\right) = \tau_t$ is the dynamic equation of the motion.

III. Pre-processing human captured motion

The main challenging issue in the imitation of human captured motion is the fast dynamic. On account of the physical limits of the humanoid robot and the capacities of the motors in its joints, it is not able to follow a fast and highly dynamic motion.
Therefore, taking the human captured motion as an initial solution for the optimization problem (3) might yield a motion, that is very different from the original one. In order to obtain a good initial guess for the optimization problem (3), one can slow down the captured motion.
Let us consider that we have a captured motion of length N samples and the sampling frequency of this motion is f (e,g $f = 120$ Hz). Let us denote $q^c(n)$ the vector of joint values which corresponds to the sample number n.
A simple algorithm to transform the human captured motion into a motion within the joint velocity limits of the humanoid robot is given by the following pseudo code

```
Input: $n \leftarrow 1$
while $n \leq N - 1$ do
    Calculate $\Delta\mathbf{q}^c \leftarrow |\mathbf{q}^c(n+1) - \mathbf{q}^c(n)|$;
    Input: $\Delta t \leftarrow \frac{1}{f}$
    while $\frac{\Delta\mathbf{q}^c}{\Delta t} > \dot{\mathbf{q}}^+$ do
        $\Delta t \leftarrow \Delta t + \frac{1}{f}$;
    end
    $n \leftarrow n + 1$;
end
```

Algorithm 1: Time re-parameterization of the human captured motion

Recall that $\dot{\mathbf{q}}^+$ is the maximal value of humanoid robot's joint velocity vector. Fig. 3.b shows the obtained motion after the application of time re-parameterization on the original human captured motion Fig. 3.a.

IV. Recursive multibody dynamics

Solving the optimization problem (3) is generally difficult on account of the complexity of the dynamic equation of motion, and the implicit relation between the vector of applied torques τ_t and the vector of joints positions $\mathbf{q_t}$.

The objective of this section is giving an overview of an efficient recursive algorithm for multibody dynamics, which allows the calculation of the gradient function of the dynamic equation analytically.

Park *et al* [11] have proposed to write the recursive multibody dynamics for serial open or branched kinematic chains using Lie group and Lie algebra. The main advantage of this formulation is to relate the joint torques and joint angles explicitly. Therefore the differentiation of joint torques with respect to joint angles can be done analytically.

Let us define the Lie groups $SO(3)$ and $SE(3)$, which denote the orthonormal matrix Θ in $R^{3\times 3}$ and the homogeneous transformation group respectively. The Lie algebra of $SO(3)$ and $SE(3)$ are denoted $so(3)$ and $se(3)$ respectively. The operators defined on these groups are: skew, matrix exponential, adjoint map $Ad_G(.)$, dual adjoint $Ad^*_G(.)$, Lie bracket $ad_g(.)$ and dual Lie bracket $ad^*_g(.)$. For more details on Lie group, Lie algebra and the operators definitions see Appendix.

A. Forward Kinematics

The kinematics of an open chain can be modeled as a sequence of homogeneous transformation between consecutive joint frames. Let $T_{i-1,i} \in SE(3)$ be the transformation matrix between the frame of link i and the frame of link $i-1$.
The matrix $T_{i-1,i}$ can be written using matrix exponential notation as follows

$$T_{i-1,i} = M_i e^{S_i q_i} \tag{4}$$

where $S_i \in se(3)$ is the joint screw written in the coordinate of link $i-1$, q_i is the current position of joint i and M_i is the coordinate transformation between link i and link $i-1$.
Using the above definition of transformation matrix, the end-effector of a kinematic chain can be calculated by the product

$$\begin{aligned} T_{0,n} &= T_{0,1} T_{1,2} \cdots T_{n-1,n} \\ &= M_1 e^{S_1 q_1} M_2 e^{S_2 q_2} \cdots M_n e^{S_n q_n} \end{aligned} \tag{5}$$

Note that by expressing the matrix of transformation in exponential form, we can calculate its derivative with respect to q_i analytically.

B. Recursive inverse dynamics of branched chains

Branched chains are serial open chains with two or more branches leading to two or more tip links [11], [12]. In the branched chains two definitions arise :

- *Parent link*: the link inward (towards the base) from a given link.
- *Child link*: the link or links which are outward (towards the tips) from a given link.

Spatial velocity of branched chains:

- **Initialization**: Given V_0.
- **Outward recursion**: loop over all links in depth manner:

$$\begin{aligned} T_{P,i} &= M_i e^{S_i q_i} \\ V_i &= Ad_{T_{P,i}^{-1}} (V_P) + S_i \dot{q}_i \\ a_i &= -ad_{S_i \dot{q}_i} (V_i) \\ b_i &= -ad^*_{V_i} (J_i V_i) \end{aligned} \tag{6}$$

where the index P denotes the parent link of link i, $T_{P,i}$ designs the mapping from the link i to its parent P and V_P denotes the spatial velocity of link P.

Applied torques on the branched chains: In order to calculate the inward recursion of forces and torques, we define the external forces applied on a link j by $\hat{F}_j$.

- **Initialization:** Given the external applied forces on each link $\hat{F}_j$, $\dot{V}_0$ and $\hat{J}_j = 0$ for each tip link.
- **Inward recursion**: loop over all links in reversed breadth

$$\begin{aligned} \dot{V}_i &= Ad_{T_{P,i}^{-1}} \left(\dot{V}_P\right) + S_i \ddot{q}_i + a_i \\ \hat{J}_i &= J_i + \sum_{j \in C} Ad^*_{T_{i,j}^{-1}} \hat{J}_j Ad_{T_{i,j}^{-1}} \\ B_i &= b_i + \sum_{j \in C} Ad^*_{T_{i,j}^{-1}} z_j \\ z_i &= \hat{J}_i \left(S_i \ddot{q}_i + a_i\right) + B_i + \sum_{j \in C} Ad^*_{T_{i,j}^{-1}} \hat{F}_j \\ F_i &= \hat{J}_i \, Ad_{T_{P,i}^{-1}} \left(\dot{V}_P\right) + z_i \\ \tau_i &= S_i^T F_i \end{aligned} \tag{7}$$

where C denotes the child links for link i, and τ_i is the torque applied on the joint i.
The matrix J_i is called the spatial inertia and it is defined as follows

$$J_i = \begin{bmatrix} I_i - m_i [r_i]^2 & m[r_i] \\ -m[r_i] & m.\mathbf{1} \end{bmatrix} \tag{8}$$

where I_i is the inertia of the link i about its center of mass and m is its mass. r_i is the vector from the point of application of the force and the center of mass of the link i. Recall that $[r_i]$ is the skew operator see Appendix for more details.

V. Optimization problem reformulation

Using Eq. (7), the dynamic equation of motion $\mathbf{M}(\mathbf{q_t})\ddot{\mathbf{q}}_\mathbf{t} + \mathbf{C}(\mathbf{q_t}, \dot{\mathbf{q}}_\mathbf{t}) = \tau_t$ can be rewritten as follows

$$S^T F_t = \tau_t \tag{9}$$

where τ_t, F_t and S are defined as follows

$$\tau_t = \begin{bmatrix} \tau_{1,t} \\ \tau_{2,t} \\ \vdots \\ \tau_{n,t} \end{bmatrix}, F_t = \begin{bmatrix} F_{1,t} \\ F_{2,t} \\ \vdots \\ F_{n,t} \end{bmatrix}, S = \begin{bmatrix} S_1 & 0 & \cdots & 0 \\ 0 & S_2 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & S_n \end{bmatrix} \tag{10}$$

τ_t and F_t are the vectors of the applied torques and forces on the humanoid robot's joints respectively. $\tau_{i,t}$ and $F_{i,t}$ denote the value of the applied torque and force on the joint i respectively. In order to transform this optimization problem into a classical optimization problem, let us define

$$\begin{aligned}
X_t &= \begin{bmatrix}\mathbf{q_t}^T & \dot{\mathbf{q}}_\mathbf{t}^T & \ddot{\mathbf{q}}_\mathbf{t}^T\end{bmatrix}^T \\
L(X_t) &= \int_{t_0}^{t_f} \left\{(\mathbf{q_t} - \mathbf{q_t^c})^T (\mathbf{q_t} - \mathbf{q_t^c}) + \cdots \right. \\
&\qquad \left. \sigma (P_t - P_t^c)^T (P_t - P_t^c)\right\} dt \\
G(X_t) &= \begin{bmatrix} \tau_t - \tau^+ \\ -\tau_t + \tau^- \\ \dot{\mathbf{q}}_\mathbf{t} - \dot{\mathbf{q}}^+ \\ -\dot{\mathbf{q}}_\mathbf{t} + \dot{\mathbf{q}}^- \\ \mathbf{q_t} - \mathbf{q}^+ \\ -\mathbf{q_t} + \mathbf{q}^- \end{bmatrix}, \quad H(X_t) = \begin{bmatrix} \tau_t - S^T F_t \\ \mathbf{q_{t_0}} - \mathbf{q_0} \\ \dot{\mathbf{q}}_\mathbf{t_0} \\ \ddot{\mathbf{q}}_\mathbf{t_0} \\ \mathbf{q_{t_f}} - \mathbf{q_f} \\ \dot{\mathbf{q}}_\mathbf{t_f} \\ \ddot{\mathbf{q}}_\mathbf{t_f} \end{bmatrix}
\end{aligned} \tag{11}$$

Thus the optimization problem (3) can be transformed into the following classical form

$$\begin{gathered}
\min_{X_t} \; L(X_t) \\
\text{subject to} \\
H(X_t) = 0 \\
G(X_t) \le 0
\end{gathered} \tag{12}$$

The above optimization problem has been extremely studied in the literature of optimization theory. To solve this optimization problem, one can use the augmented Lagrange multiplier method, which is a very efficient and reliable method [13]. Using the augmented Lagrange multiplier method transforms the optimization problem (12) to the minimization of the following function

$$\min_{X_t,\lambda} \tilde{L}(X_t,\lambda) = L(X_t) + \lambda_\psi^T \psi + \frac{1}{2}\sigma \psi^T \psi + \lambda_H^T H + \frac{1}{2}\sigma H^T H \tag{13}$$

where $\lambda = \begin{bmatrix}\lambda_\psi^T & \lambda_H^T\end{bmatrix}^T$, $\psi = \max\left\{G(X_t), \frac{-1}{\sigma}\lambda_\psi\right\}$. Then there exist λ^* such that X_t^* is a local minimum of the unconstrained $\tilde{L}(X_t,\lambda^*)$ for all σ smaller than some finite $\bar{\sigma}$.

To solve the unconstrained optimization problem of $\tilde{L}(X_t,\lambda)$ with respect to X_t, one can use Gauss-Newton method. Note that the function $\tilde{L}(X_t,\lambda)$ is differentiable in X_t if and only if $L(X_t)$, $H(X_t)$ and $G(X_t)$ are differentiable in X_t, and in this case we can write

$$\begin{aligned}
\frac{\partial \tilde{L}(X_t,\lambda)}{\partial X_t} =& \frac{\partial L(X_t)}{\partial X_t} + (\lambda_H + \sigma H)^T \frac{\partial H(X_t)}{\partial X_t} + \\
& \max\left\{0, \lambda_\psi + \sigma G(X_t)\right\}^T \frac{\partial G(X_t)}{\partial X_t}
\end{aligned} \tag{14}$$

As λ^* is unknown, an update rule is used

$$\begin{aligned}
\lambda_H^{k+1} &= \lambda_H^k + \sigma H(X_t^k) \\
\lambda_\psi^{k+1} &= \lambda_\psi^k + \sigma \psi(X_t^k)
\end{aligned} \tag{15}$$

where X_t^k is the unconstrained minimum of $\tilde{L}(X_t,\lambda^k)$. Such updating rule will generate a sequence λ^k converges to λ^* [14]. In practice, a good schedule is to choose a moderate σ^0, and increase it as follows

$$\sigma^{k+1} = \alpha \sigma^k \tag{16}$$

where α is between 5 and 10. A threshold $\bar{\sigma}$ is chosen and the update rule of σ stops when σ^k becomes higher than $\bar{\sigma}$.

For more details on the algorithm of augmented Lagrange multiplier method see [15], [13], [14].

Approximating the gradient function $\frac{\partial \tilde{L}(X_t,\lambda)}{\partial X_t}$ by a numerical difference method is usually used in practice. However, this approach is not only a time consuming method on account of the evaluation of the gradient calculation, but also may not converge well because of the approximation.

As we have mentioned the main advantage of using the recursive dynamic algorithm explained in Section IV-B is calculating the gradient function analytically in a recursive way.

A. Gradient calculation

The objective is to calculate the gradient of the dynamic quantities. By considering the vector of parameters $X_t = \begin{bmatrix}\mathbf{q_t}^T & \dot{\mathbf{q}}^T & \ddot{\mathbf{q}}^T\end{bmatrix}^T$, let us start by calculating the derivatives of the operators with respect to an element x of X_t

$$\begin{aligned}
\frac{\partial T_{0,n}}{\partial x} =& T_{0,i}\left(S_i \delta_{x,q_i}\right) T_{i,n} \\
\frac{\partial Ad_{T_{i-1,i}^{-1}}(Y)}{\partial x} =& ad_{Ad_{T_{i-1,i}^{-1}}(Y)}\left(S_i \delta_{x,q_i}\right) + Ad_{T_{i-1,i}^{-1}}\left(\frac{\partial Y}{\partial x}\right) \\
\frac{\partial Ad^*_{T_{i,i+1}^{-1}}(Y)}{\partial x} =& ad^*_{Ad_{M_{i+1}}\left(S_{i+1}\delta_{x,q_{i+1}}\right)}\left(Ad^*_{T_{i,i+1}^{-1}}(Y)\right) + \cdots \\
& Ad^*_{T_{i,i+1}^{-1}}\left(\frac{\partial Y}{\partial x}\right) \\
\frac{\partial ad_Z(Y)}{\partial x} =& ad_{\frac{\partial Z}{\partial x}}(Y) + ad_Z\left(\frac{\partial Y}{\partial x}\right) \\
\frac{\partial ad^*_Z(Y)}{\partial x} =& ad^*_{\frac{\partial Z}{\partial x}}(Y) + ad^*_Z\left(\frac{\partial Y}{\partial x}\right)
\end{aligned} \tag{17}$$

where δ_{x_1,x_2} is the Kronecker delta defined as follows

$$\delta_{x_1,x_2} = \begin{cases} 1 & \text{if } x_1 = x_2 \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

The calculation of the gradient with respect to X_t can be done in a recursive way analogously to the recursive dynamic calculation.

Forward recursion:

- **Initialization**: Given $\frac{\partial V_0}{\partial X_t}$.
- loop over all links in depth manner:

$$\begin{aligned}
\frac{\partial V_i}{\partial X_t} &= \frac{\partial Ad_{T_{P,i}^{-1}}(V_P)}{\partial X_t} + S_i \frac{\partial \dot{q}_i}{\partial X_t} \\
\frac{\partial a_i}{\partial X_t} &= -\frac{\partial ad_{S_i \dot{q}_i}(V_i)}{\partial X_t} \\
\frac{\partial b_i}{\partial X_t} &= -\frac{\partial ad^*_{V_i}(J_i V_i)}{\partial X_t}
\end{aligned} \tag{19}$$

Backward recursion:

- **Initialization:** Given $\frac{\partial \hat{F}_j}{\partial X_t}$, $\frac{\partial \dot{V}_0}{\partial X_t}$.

- loop over all links in reversed breadth

$$
\begin{aligned}
\frac{\partial \dot{V}_i}{\partial X_t} =& \frac{\partial Ad_{T_{P,i}^{-1}}\left(\dot{V}_P\right)}{\partial X_t} + S_i \frac{\partial \ddot{q}_i}{\partial X_t} + \frac{\partial a_i}{\partial X_t} \\
\frac{\partial \hat{J}_i}{\partial X} =& \sum_{j \in C} \frac{\partial Ad^*_{T_{i,j}^{-1}}}{\partial X_t} \hat{J}_j Ad_{T_{i,j}^{-1}} + \cdots \\
& Ad^*_{T_{i,j}^{-1}} \hat{J}_j \frac{\partial Ad^*_{T_{i,j}^{-1}}}{\partial X_t} + Ad^*_{T_{i,j}^{-1}} \frac{\partial \hat{J}_j}{\partial X_t} Ad_{T_{i,j}^{-1}} \\
\frac{\partial B_i}{\partial X_t} =& \frac{\partial b_i}{\partial X_t} + \sum_{j \in C} \frac{\partial Ad^*_{T_{i,j}^{-1}} z_j}{\partial X_t} \\
\frac{\partial z_i}{\partial X_t} =& \frac{\partial \hat{J}_i}{\partial X_t} \left(S_i \ddot{q}_i + a_i\right) + \hat{J}_i \left(S_i \frac{\partial \ddot{q}_i}{\partial X_t} + \frac{\partial a_i}{\partial X_t}\right) + \cdots \\
& \frac{\partial B_i}{\partial X_t} + \sum_{j \in C} \frac{\partial Ad^*_{T_{i,j}^{-1}} \hat{F}_j}{\partial X_t} \\
\frac{\partial F_i}{\partial X_t} =& \frac{\partial \hat{J}_i}{\partial X_t} Ad_{T_{P,i}^{-1}} \left(\dot{V}_P\right) + \hat{J}_i \frac{\partial Ad_{T_{P,i}^{-1}}\left(\dot{V}_P\right)}{\partial X_t} + \frac{\partial z_i}{\partial X_t} \\
\frac{\partial \tau_i}{\partial X_t} =& S_i^T \frac{\partial F_i}{\partial X_t}
\end{aligned}
\tag{20}
$$

where as we mentioned C denotes the child links for link i.

VI. Discretization of configuration space

It is well known that the space of the admissible solutions of the minimization problem (3) is very large. In order to transform this infinite dimensional space to a finite one, we can use a basis of shape functions.

Let us consider a basis of shape functions B_t that is defined as follows

$$B_t = \begin{bmatrix} B_t^1 & B_t^2 & \cdots & B_t^l \end{bmatrix}^T \tag{21}$$

where B_t^i denotes the value of shape function number i at the instant t, the dimension of B_t is l defines the dimension of the shape function basis.

The projection of the vector of angular values $\mathbf{q_t}$ into the basis of shape functions B_t can be given by the following formula

$$\mathbf{q_t} = Q_B B_t \tag{22}$$

where Q_B is a constant matrix.

The derivative $\dot{\mathbf{q}}_\mathbf{t}$ and $\ddot{\mathbf{q}}_\mathbf{t}$ can be written as follows

$$
\begin{aligned}
\dot{\mathbf{q}}_\mathbf{t} =& Q_B \dot{B}_t \\
\ddot{\mathbf{q}}_\mathbf{t} =& Q_B \ddot{B}_t
\end{aligned}
\tag{23}
$$

In this case, the derivative with respect to each element $Q_B(i,j)$ of the matrix Q_B can be computed using the following formula

$$
\begin{aligned}
\frac{\partial Y_t}{\partial Q_B(i,j)} &= \frac{\partial Y_t}{\partial X_t} \times \frac{\partial X_t}{\partial Q_B(i,j)} \\
&= \frac{\partial Y}{\partial X_t} \times \left(e_i \otimes \begin{bmatrix} B_t^j \\ \dot{B}_t^j \\ \ddot{B}_t^j \end{bmatrix} \right)
\end{aligned}
\tag{24}
$$

where $e_i \in \mathscr{R}^n$,

$$e_i = \begin{bmatrix} 0 & \ldots & 0 & \underset{\substack{\uparrow \\ i}}{1} & 0 & \ldots & 0 \end{bmatrix}^T$$

and $\otimes$ denotes Kronecker's product operator.

By using the discretization of the configuration space, the problem of optimization transforms into the problem of finding the optimal matrix Q_B, which minimizes the function $\tilde{L}(X_t, \lambda)$ in Eq. (13).

It remains to define the shape functions B_t^i. In our case, the shape functions should verify the following properties:

1) They are continuous.
2) Their first and second derivatives are continuous.

Therefore, we use the *quartic B-spline functions*.

VII. Experimental results

We have chosen a boxing captured motion to validate our proposed method. As the movements of the upper body of the boxing motion is complex, so the imitation by a humanoid robot is really a challenging issue. The vertical movement of the pelvis joint is also considered, and the movements of the lower body is calculated using inverse kinematic. Snapshots of the conducted motion using OpenHRP platform [16] are presented in Fig. 2. In order to ensure the dynamical stability of humanoid robot, the trajectory of ZMP is controlled using the method of cart table model proposed by Kajita *et al* [8] as follows

1) The ZMP trajectory of the optimized motion is calculated.
2) The calculated ZMP trajectory is then modified and restricted to be inside the polygon of support feet.
3) Once the modified ZMP trajectory is available, we use the cart table model [8] to calculate the trajectory of the center of mass.
4) The horizontal trajectory of the free flyer (pelvis joint) is then calculated using the trajectory of the center of mass.

Fig. 3.a shows the angular position trajectory of the virtual actor's right elbow, Fig. 3.b shows the modified trajectory after the application of the time re-parameterization algorithm explained in Section III. The optimized trajectory for the humanoid robot is given in Fig. 3.c. The optimized trajectory respects the physical limits of HRP-2 humanoid robot, which are not only the angle limits but also the joint velocity and torque limits.

However, the self collision problem is not considered in this work as shown in Fig. 4.a. Although, approximating the humanoid robot's links by cylinders and spheres, and then consider the distance between them as an additional constraint can solve the problem of self collision, this procedure might yield an imitated motion largely different from the original human captured motion on account of the approximation.

F. Kanehiro *et al* [17] have proposed an efficient method to avoid the collision for a non-strictly convex objects. The method makes use of non-strictly convex polyhedra as geometric models of the robot and the environment without any approximation. Applying this method as post-processing task can solve the problem of self collision and it yields a collision-free motion. Fig. 4.b shows the self collision avoidance of the humanoid robot's hands by applying the method proposed by F. Kanehiro *et al* [17].

VIII. CONCLUSION

In this paper, the human motion imitation by a humanoid robot is considered. In order to generate an imitated motion within the humanoid robot capabilities, the imitation problem is formulated as an optimization problem. The physical limits of the humanoid robot are transformed into constraints of the optimization problem, and the objective function to be minimized is the difference between the angular values of the humanoid robot' s joints and those of the virtual actor.

The experimental results have pointed out that the proposed method yields motions that preserve the salient characteristics of the original human captured motion, and at the same time they respect the physical limits of the humanoid robot.

Future work will focus on the integration of self collision avoidance into the optimization problem as an additional constraint.

IX. ACKNOWLEDGMENT

The human captured data used in this paper was obtained from mocap.cs.cmu.edu. The database was created with funding from NSF EIA-0196217.

APPENDIX

A *Lie group* is a differentiable manifold. An example of Lie group is the orthonormal matrix Θ in $R^{3\times3}$, which is called $SO(3)$. Note that this group consists of the rotation matrices in Euclidean space. Another example of Lie group is the group of homogeneous transformation which is the special Euclidean group or $SE(3)$. Given a rotation $\Theta \in SO(3)$ and translation $b \in R^3$, the homogeneous matrix is defined as follows

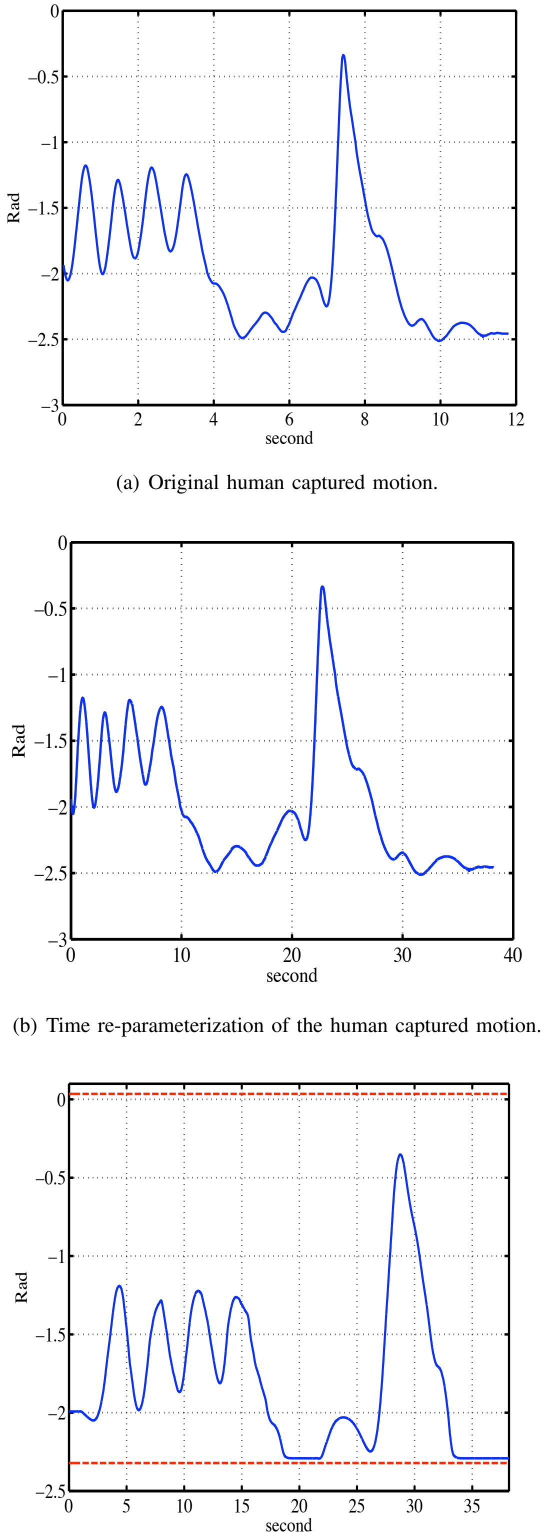
$$G = \begin{bmatrix} \Theta & b \\ 0 & 1 \end{bmatrix} \qquad \text{(A-1)}$$

An important concept associated with each Lie group is the notation of *Lie algebra*. The tangent space at the identity element of a Lie group is called the Lie algebra for that group. The Lie algebra of $SO(3)$ and $SE(3)$ are denoted $so(3)$ and $se(3)$ respectively.

Let us define some notations and operations on Lie groups and Lie algebra:

1) Skew operator:

$$[.] : \omega \in R^3 \rightarrow so(3)$$
$$[\omega] = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \qquad \text{(A-2)}$$



(a) Original human captured motion.

(b) Time re-parameterization of the human captured motion.

(c) Obtained motion after the optimization and considering the physical limits of the humanoid robot (HRP-2). Dash lines denote the angular limits of the elbow joint.

Fig. 3. Angular position of the right elbow.

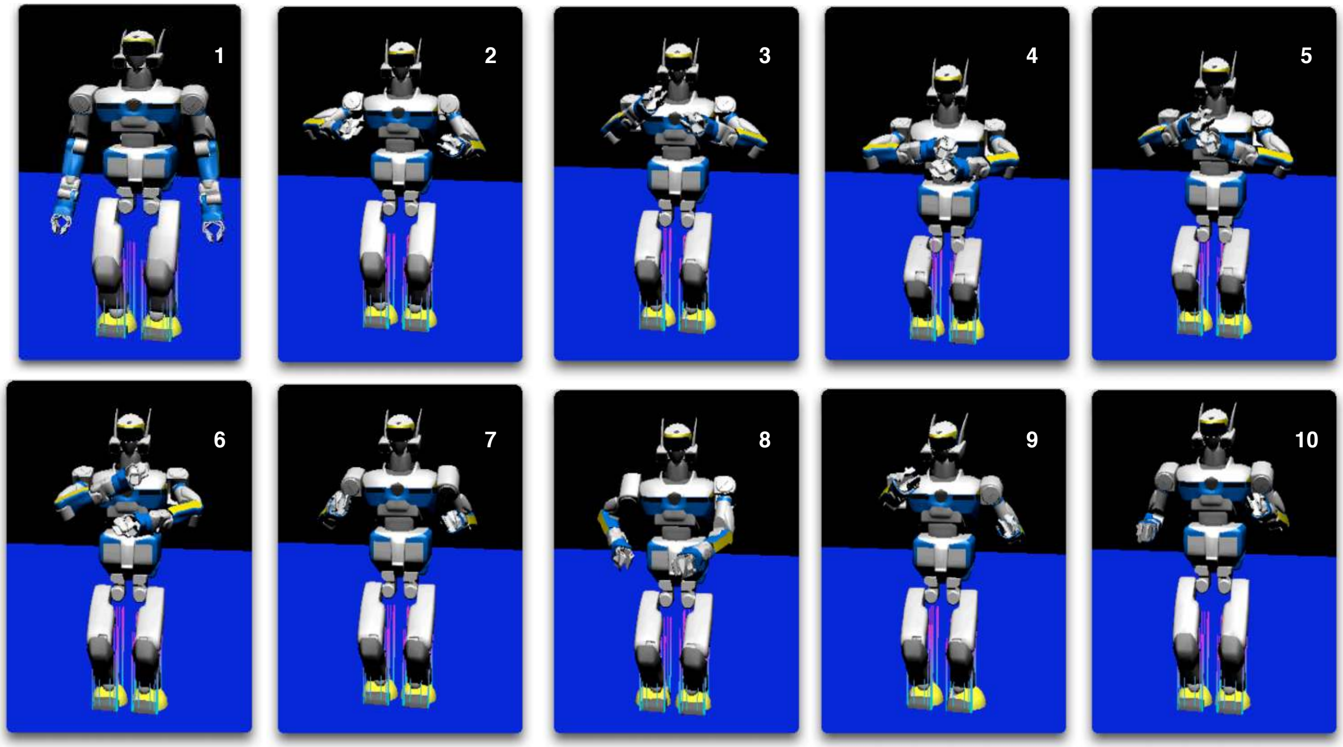


Fig. 2. Snapshots of the conducted motion.

2) $(.,.)$ operator:

$$(.,.):\{\omega, v\} \in R^3 \rightarrow se(3)$$

$$(\omega, v) = \begin{bmatrix} [\omega] & v \\ 0 & 0 \end{bmatrix} \tag{A-3}$$

3) Matrix exponential:

$$e^{(\omega,v)} = \exp\begin{bmatrix} [\omega] & v \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \exp([\omega]) & Av \\ 0 & 1 \end{bmatrix} \tag{A-4}$$

where

$$\exp([\omega]) = I + \frac{\sin\phi}{\phi}[\omega] + \frac{1-\cos\phi}{\phi^2}[\omega]^2,\ \phi = \|\omega\|$$

$$A = I + \frac{1-\cos\phi}{\phi^2}[\omega] + \frac{\phi - \sin\phi}{\phi^3}[\omega]^2 \tag{A-5}$$

4) Adjoint map on $SE(3)$:

$$Ad_G(h) : se(3) \rightarrow se(3)$$

$$Ad_G(h) = \begin{bmatrix} \Theta & 0 \\ [b]\Theta & \Theta \end{bmatrix} \begin{bmatrix} h_\omega \\ h_v \end{bmatrix} \tag{A-6}$$

where $G \in SE(3)$ is defined as in (A-1), and $h = (h_\omega, h_v) \in se(3)$.

5) Dual adjoint operator:

$$Ad^*_G(h^*) : se(3)^* \rightarrow se(3)^*$$

$$Ad^*_G(h^*) = \begin{bmatrix} \Theta^T & \Theta^T[b]^T \\ 0 & \Theta^T \end{bmatrix} \begin{bmatrix} M \\ F \end{bmatrix} \tag{A-7}$$

where $G \in SE(3)$, and $h^* = (M, F) \in se(3)^*$.

6) Lie bracket operator:

$$ad_g(h) = [g,h] = \begin{bmatrix} [g_\omega] & 0 \\ [g_v] & [g_\omega] \end{bmatrix} \begin{bmatrix} h_\omega \\ h_v \end{bmatrix} \tag{A-8}$$

where $g, h \in se(3)$. $h = (h_\omega, h_v)$ and $g = (g_\omega, g_\omega)$.

7) Dual Lie bracket operator:

$$ad^*_g(h^*) = [g, h^*] = \begin{bmatrix} [g_\omega]^T & [g_v]^T \\ 0 & [g_\omega]^T \end{bmatrix} \begin{bmatrix} M \\ F \end{bmatrix} \tag{A-9}$$

where $g = (g_\omega, g_v) \in se(3)$ and $h^* = (M, F) \in se(3)^*$.

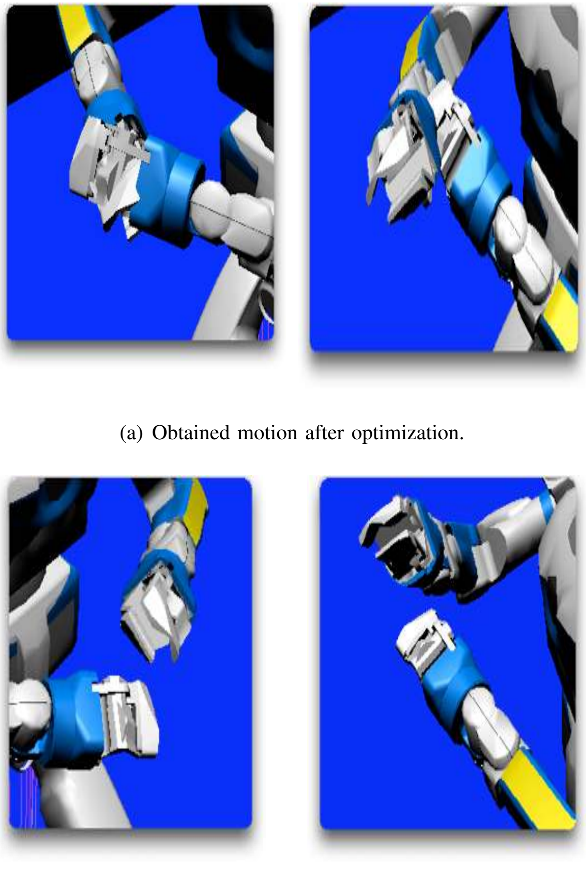
References

[1] 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*, 2002.

[2] S. Nakaoka, A. Nakazawa, K. Yokoi, H. Hirukawa, and K. Ikeuch, "Generating Whole Body Motions for a Biped Humanoid Robot from Captured Human Dances," in *IEEE International Conference on Robotics and Automation*, pp. 3905–3910, 2003.

[3] M. Vukobratović and B. Borovac, "Zero-Moment Point—Thirty Five Years of its Life," *International Journal of Humanoid Robotics*, vol. 1, no. 1, pp. 157–173, 2004.

[4] A. Safonova, N. Pollard, and J. Hodgins, "Optimizing Human Motion for the Control of a Humanoid Robot," in *Proc. Applied Mathematics and Applications of Mathematics*, 2003.



(a) Obtained motion after optimization.

(b) Applying self collision avoidance [17] as post-processing task.

Fig. 4. Self collisions avoidance.

[5] M. Ruchanurucks, S. Nakaoka, S. Kudoh, and K. Ikeuchi, "Humanoid Robot Motion Generation with Sequential Physical Constraints," in *Proc. IEEE International Conference on Robotics and Automation*, pp. 2649–2654, 2006.

[6] 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*, vol. 1, pp. 610–615, 2004.

[7] S. Nakaoka, A. Nakazawa, F. Kanehiro, K. Kaneko, M. Morisawa, and K. Ikeuchi, "Task Model of Lower Body Motion for a Biped Humanoid Robot to Imitate Human Dances," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3157–3162, 2005.

[8] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point," in *Proc. IEEE International Conference on Robotics and Automation*, pp. 1620–1626, 2003.

[9] K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukawa, T. Kawasaki, M. Hirata, K. Akachi, and T. Isozumi, "Humanoid Robot HRP-2," in *Proc. IEEE International Conference on Robotics and Automation*, pp. 1083–1090, 2004.

[10] M. Gleicher, "Retargetting Motion to New Characters," in *ACM SIGGRAPH*, pp. 33–42, 1998.

[11] F. Park, J. Bobrow, and S. Ploen, "A Lie Group Formulation of Robot Dynamics," *International Journal of Robotics Research*, vol. 14, no. 6, pp. 1130–1135, 1995.

[12] G. Sohl and J. Bobrow, "A Recursive Multibody Dynamics and Sensitivity Algorithm for Branched Kinematics Chains," tech. rep., Department of Mechanical Engineering, University of California, June 2000.

[13] R. Rockafellar, "Augmented Lagrange Multiplier Functions and Duality in Nonconvex Programming," *SIAM J.Control*, vol. 12, May 1974.

[14] D. P. Bertsekas, *Nonlinear programming*. Athena Scientific, 1995.

[15] R. Rockafellar, "Penalty Methods and Augmented Lagrangians in Nonlinear Programming," in *Proc. 5th IFIP Conference on Optimization techniques*, 1973.

[16] F. Kanehiro, H. Hirukawa, and S. Kajita, "OpenHRP: Open architecture Humanoid Robotics Platform," *International Journal of Robotics Research*, vol. 23, no. 2, pp. 155–165, 2004.

[17] F.Kanehiro, F.Lamiraux, O.Kanoun, E.Yoshida, and J.-P. Laumond, "A Local Collision Avoidance Method for Non-strictly Convex Objects," Tech. Rep. 08025, LAAS - CNRS, January 2008.