Time Parameterization of Humanoid Robot Paths

Wael Suleiman, Fumio Kanehiro, *Member, IEEE*, Eiichi Yoshida, *Member, IEEE*,
Jean-Paul Laumond, *Fellow Member, IEEE*, and André Monin

***Abstract*—This paper proposes a unified optimization framework to solve the time parameterization problem of humanoid robot paths. Even though the time parameterization problem is well known in robotics, the application to humanoid robots has not been addressed. This is because of the complexity of the kinematical structure as well as the dynamical motion equation. The main contribution in this paper is to show that the time parameterization of a statically stable *path* to be transformed into a dynamical stable *trajectory* within the humanoid robot capacities can be expressed as an optimization problem. Furthermore we propose an efficient method to solve the obtained optimization problem. The proposed method has been successfully validated on the humanoid robot HRP-2 by conducting several experiments. These results have revealed the effectiveness and the robustness of the proposed method.**

***Index Terms*—Timing; Robot dynamics; Stability criteria; Optimization methods; Humanoid Robot.**

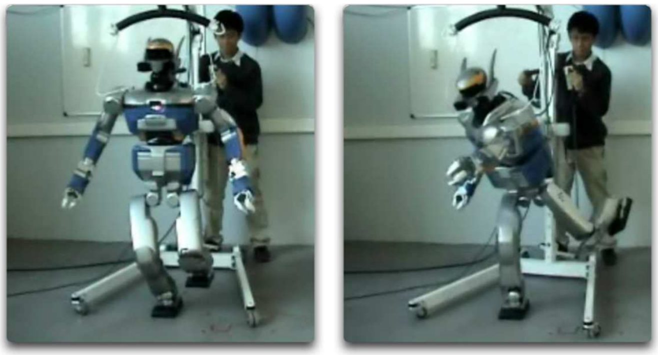

(a) Initial configuration (b) Final configuration

Fig. 1. Dynamically stable motion

I. Introduction

The automatic generation of motions for robots which are collision-free motions and at the same time inside the robots capacities is one of the most challenging problems. Many researchers have separated this problem into several smaller subproblems. For instance, collision-free path planning, time parameterization of a specified path, feedback control along a specified path using vision path planning, etc. In this paper, the problem of finding a time parameterization of a given path for a humanoid robot is investigated.

The time parameterization problem is an old problem in robotic research [1]. In order to better understand the objective of time parameterization of a path, let us start by defining a path and a trajectory.

A *path* denotes the locus of points in the joint space, or in the operational space, the robot has to follow in the execution of the desired motion, and a *trajectory* is a path on which a time law is specified [2].

Generally speaking, the time parameterization of a path is the problem of transforming this path into a trajectory which respects the physical limits of the robot, e.g. velocity limits, acceleration limits, torque limits, etc.

In the research works on manipulators, the time parameterization problem has the objective of reducing the execution time of the tasks, thereby increasing the productivity. Most of these approaches are based on time-optimal control theory [3], [4], [5], [6], [7].

In the framework of mobile robots, the time parameterization problem arises also to transform a feasible path into a feasible trajectory [8], [9]. The main objective, in this case, is to reach the goal position as fast as possible.

Even though the conventional methods which are based on the optimal-control theory have been successfully applied in practice on manipulators and mobile robots, the application of time optimal control theory in the case of humanoid robot is however a difficult task. This is because not only the dynamic equation of the humanoid robot motion is very complex, but also applying time optimal control theory requires the calculation of the derivative of the configuration space vector of humanoid robot with respect to the parameterized path. Although such calculation can be evaluated from differential geometry, it is a very difficult task in the case of systems with large number of degree of freedoms and branched kinematic chains, which is the case of humanoid robot. For that, we propose to solve the time parameterization problem numerically using a finite difference approach.

The remainder of this paper is organized as follows. Section II gives an overview of the dynamic stability notion and the mathematical formulation of Zero Moment Point (ZMP). The time parameterization problem is formulated as an optimization problem under constraints and an efficient method to solve it is explained in Section III. Cases studies and experimental results using the humanoid robot platform HRP-2 are given in Section IV.

II. Dynamic Stability and ZMP: An Overview

Let us start by introducing some definitions:

***Definition** 1:* Statically stable trajectory is a trajectory for which the trajectory of the projection of the Center of Mass

W. Suleiman, F. Kanehiro and E. Yoshida are with CNRS-AIST JRL (Joint Robotics Laboratory), UMI3218/CRT, National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba Central 2, 1-1-1 Umezono, Tsukuba, Ibaraki, 305-8568 Japan. {wael.suleiman, f-kanehiro, e.yoshida}@aist.go.jp

J.-P. Laumond and André Monin are with LAAS-CNRS, 7 Avenue du Colonel Roche, 31077 Toulouse, France. {jpl, monin}@laas.fr

A part of this work has been done at LAAS-CNRS during the Ph.D. thesis of W. Suleiman.

(CoM) of the humanoid robot on the horizontal plane is always inside of the polygon of support (i.e. the convex hull of all points of contact between the support foot (feet) and the ground).

***Definition** 2:* Dynamically stable trajectory is a trajectory for which the trajectory of Zero Moment Point (ZMP) [10] is always inside of the polygon of support.

The generation of a statically stable path deals only with the kinematic constraints of the humanoid robot. It can be obtained by constraining the projection of the Center of Mass (CoM) on the horizontal plane to be always inside of the polygon of support [11], [12], [13], [14].

Theoretically, any statically stable trajectory can be transformed into a dynamically stable one by slowing down the humanoid robot's motion.

Let the ZMP on the horizontal ground be given by the following vector

$$\boldsymbol{p} = \begin{bmatrix} p_x & p_y \end{bmatrix}^T \tag{1}$$

To compute $\boldsymbol{p}$, one can use the following formula

$$\boldsymbol{p} = \boldsymbol{N}\ \frac{\boldsymbol{n} \times \boldsymbol{\tau}^o}{(\boldsymbol{f}^o|\boldsymbol{n})} \tag{2}$$

where the operator $\times$ and $(.|.)$ refer to the cross and scalar products respectively, and

- $\boldsymbol{N}$ is a constant matrix
$$\boldsymbol{N} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{3}$$
.
- the vector $\boldsymbol{n}$ is the normal vector on the horizontal ground $\left(\boldsymbol{n} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T\right)$.
- The vector $\boldsymbol{f}^o$ is the result of the gravity and inertia forces
$$\boldsymbol{f}^o = M\boldsymbol{g} - \sum_{i=0}^{n} m_i\ \ddot{\boldsymbol{X}}^{c_i} \tag{4}$$
where $\boldsymbol{g}$ denotes the acceleration of the gravity ($\boldsymbol{g} = -g\boldsymbol{n}$), and M is the total mass of the humanoid robot. The quantities m_i, $\ddot{\boldsymbol{X}}^{c_i}$ are the mass of the i^{th} link and the acceleration of its center of mass c_i respectively. Note that m_0, $\ddot{\boldsymbol{X}}^{c_0}$ are, respectively, the mass and the acceleration of the free-flyer joint (pelvis joint) of the humanoid robot.
- $\boldsymbol{\tau}^o$ denotes the moment of the force $\boldsymbol{f}^o$ about the origin of the fixed world frame. The expression of $\boldsymbol{\tau}^o$ is the following
$$\boldsymbol{\tau}^o = \sum_{i=0}^{n} \left(m_i\ \boldsymbol{X}^{c_i} \times \left(\boldsymbol{g} - \ddot{\boldsymbol{X}}^{c_i} \right) - \dot{\boldsymbol{\mathcal{L}}}^{c_i} \right) \tag{5}$$
where $\boldsymbol{\mathcal{L}}^{c_i}$ is the angular momentum at the point c_i
$$\dot{\boldsymbol{\mathcal{L}}}^{c_i} = \boldsymbol{R}^i \left(\boldsymbol{I}_{c_i} \dot{\boldsymbol{\omega}}^i - \left(\boldsymbol{I}_{c_i} \boldsymbol{\omega}^i \right) \times \boldsymbol{\omega}^i \right) \tag{6}$$
$\boldsymbol{R}^i$ and $\boldsymbol{I}_{c_i}$ are the rotation matrix associated to the i^{th} link and its inertia matrix respectively. $\boldsymbol{\omega}^i$ is the angular velocity of the i^{th} link can be obtained using the following formula
$$\left[\boldsymbol{\omega}^i\right]^\wedge = \frac{d\boldsymbol{R}^i}{dt} \boldsymbol{R}^{iT} \tag{7}$$
where $[.]^\wedge$ designs the skew operator defined as follows
$$\begin{aligned} &[.]^\wedge : \boldsymbol{\omega} \in \mathbb{R}^3 \rightarrow so(3) \\ &[\boldsymbol{\omega}]^\wedge = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \end{aligned} \tag{8}$$
where $so(3)$ denotes the Lie algebra of $SO(3)$ which is the group of rotation matrices in the Euclidean space. The inverse operator of skew operator can be defined as follows
$$\begin{aligned} &[.]^\vee : \boldsymbol{\Omega} \in so(3) \rightarrow \mathbb{R}^3 \\ [\boldsymbol{\Omega}]^\vee &\triangleq \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}^\vee \\ &= \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \end{aligned} \tag{9}$$
and the angular velocity can be calculated using the inverse operator as follows
$$\boldsymbol{\omega}^i = \left[\frac{d\boldsymbol{R}^i}{dt} \boldsymbol{R}^{iT} \right]^\vee \tag{10}$$
Note that the skew operator and its inverse are linear operators.
Finally, $\dot{\boldsymbol{\omega}}^i$ denotes the angular acceleration of the i^{th} link.
Note that $\boldsymbol{R}^0$ and $\boldsymbol{I}_{c_0}$ are the rotation matrix associated to the free-flyer joint (pelvis joint) and its inertia matrix respectively. $\boldsymbol{\omega}^0$ and $\dot{\boldsymbol{\omega}}^0$ are the angular velocity and the angular acceleration of the free-flyer joint respectively.

III. Time Parameterization Problem Formulation

Generally speaking, the time parameterization problem of a function $f\left(x_t\right)$, where t denotes time, consists in finding a real function $\mathcal{S}_t$ in such a way $f\left(x_{\mathcal{S}_t}\right)$ verifies some temporal constraints.

Mathematically that means

$$h(\mathcal{S}_t) \leq f\left(x_{\mathcal{S}_t}\right) \leq l(\mathcal{S}_t) \tag{11}$$

In order to obtain a causal and feasible motion, the function $\mathcal{S}_t$ should be a strictly increasing function, that means $\frac{d\mathcal{S}_t}{dt} > 0$. Therefore we will express $\mathcal{S}_t$ as the integral of a strictly positive function $s_h > 0$, as follows

$$\mathcal{S}_t = \int_{h=0}^{t} s_h\ dh \tag{12}$$

Our objective is to transform the statically stable path into a dynamically stable trajectory by minimizing a specified criterion. We will be interested in minimizing cost function of the form

$$\begin{aligned} J\left(\mathcal{S}_t\right) &= S_{t_f} + \int_{h=0}^{t_f} L\left(\boldsymbol{q}_{s_h}, \dot{\boldsymbol{q}}_{s_h}, \ddot{\boldsymbol{q}}_{s_h}, \tau_{s_h}, s_h\right)\ dh \\ &= \underbrace{\int_{h=0}^{t_f} s_h\ dh}_{T_1} + \underbrace{\int_{h=0}^{t_f} L\left(\boldsymbol{q}_{s_h}, \dot{\boldsymbol{q}}_{s_h}, \ddot{\boldsymbol{q}}_{s_h}, \tau_{s_h}, s_h\right)\ dh}_{T_2} \end{aligned} \tag{13}$$

where $\boldsymbol{q}_{s_h}, \dot{\boldsymbol{q}}_{s_h}, \ddot{\boldsymbol{q}}_{s_h}$ and τ_{s_h} design the configuration vector, the joint velocity, the joint acceleration and the exerted torque vector respectively. The first term of the cost function T_1 has the purpose of minimizing the final time t_f, the second term T_2 captures the desire to minimize an energy related function such as the exerted joint torques or the jerk function of some specific joints of the humanoid robot.

In order to obtain a motion within the humanoid robot capacities, we will consider two cases:

1) The physical limits which are taken into account are the joint velocity and acceleration limits of the humanoid robot. In this case, the function L in Eq. (13) is defined independently of the exerted torques on humanoid robot's joints.
2) The physical limits which are taken into account are the joint velocity and torques limits of the humanoid robot, and there is no constraint on the function L.

Even though the first case is included in the second one, we will propose an adequate and optimized method to solve each case.

A. Case 1: joint velocity and acceleration limits

Let us suppose that we have a path which consists of K points. At first, we transform this path into a trajectory by considering a uniform time distribution function. In other words, we suppose that $s_t = 1 : \ \forall t$ in Eq. (12). Let the sampling period of the desired trajectory be Δt, we denote the time horizon $T = K\,\Delta t$.

In this case, the time parameterization problem of transforming the initial path into a dynamically stable trajectory within the joint velocity and acceleration limits of the humanoid robot can be expressed as an optimization problem as follow

$$\min_{s_t} J(s_t) = \min_{s_t} \left\{ \int_{t=0}^{T} s_t dt + \int_{t=0}^{T} L\left(\boldsymbol{q}_{s_t}, \dot{\boldsymbol{q}}_{s_t}, \ddot{\boldsymbol{q}}_{s_t}, s_t\right) dt \right\} \tag{14}$$

subject to

$$s_t > 0 \tag{15}$$

$$\boldsymbol{p}_{s_t}^{-} \leq \boldsymbol{p}_{s_t} \leq \boldsymbol{p}_{s_t}^{+} \tag{16}$$

$$\left(\boldsymbol{f}_{s_t}^{o} | \boldsymbol{n}\right) < 0 \tag{17}$$

$$\mu \ \left(\boldsymbol{f}_{s_t}^{o} | \boldsymbol{n}\right) < \left(\boldsymbol{\tau}_{s_t}^{o} | \boldsymbol{n}\right) < -\mu \ \left(\boldsymbol{f}_{s_t}^{o} | \boldsymbol{n}\right) \tag{18}$$

$$\dot{\boldsymbol{q}}^{-} \leq \dot{\boldsymbol{q}}_{s_t} \leq \dot{\boldsymbol{q}}^{+} \tag{19}$$

$$\ddot{\boldsymbol{q}}^{-} \leq \ddot{\boldsymbol{q}}_{s_t} \leq \ddot{\boldsymbol{q}}^{+} \tag{20}$$

where $\boldsymbol{p}_{s_t}$ is the ZMP vector, $\boldsymbol{p}_{s_t}^{-}$ and $\boldsymbol{p}_{s_t}^{+}$ design the polygon of support for the humanoid robot. $\boldsymbol{f}_{s_t}^{o}$ and $\boldsymbol{\tau}_{s_t}^{o}$ denote the applied force on the foot (feet) and the moment of this force about the origin of the fixed world frame respectively.

μ is the coefficient of the static friction. The vector $\dot{\boldsymbol{q}}_{\boldsymbol{s_t}}$ and $\ddot{\boldsymbol{q}}_{\boldsymbol{s_t}}$ denote the joint velocity and acceleration of the humanoid robot respectively. $\ddot{\boldsymbol{q}}^{+}$ and $\dot{\boldsymbol{q}}^{+}$ design the upper limits of acceleration and velocity respectively. $\ddot{\boldsymbol{q}}^{-}$ and $\dot{\boldsymbol{q}}^{-}$ design the lower limits of acceleration and velocity respectively.

The constraints of the optimization problem (14) can be analyzed as follows

- Constraint (16) guarantees that the ZMP is inside of the polygon of support.
- Constraint (17) ensures that the foot is in contact with the ground (the humanoid robot will not jump).
- Constraint (18) prevents the foot from sliding around the Z-axis.
- Constraint (19) guarantees that the obtained trajectory respects the joint velocity limits of the humanoid robot.
- Constraint (20) guarantees that the obtained trajectory respects the joint acceleration limits of the humanoid robot.

Let us write $\boldsymbol{p}_{s_t}$, $\boldsymbol{f}_{s_t}^{o}$ and $\boldsymbol{\tau}_{s_t}^{o}$as functions of s_t

$$\boldsymbol{p}_{s_t} = \boldsymbol{N} \ \frac{\boldsymbol{n} \times \boldsymbol{\tau}_{s_t}^{o}}{\left(\boldsymbol{f}_{s_t}^{o} | \boldsymbol{n}\right)} \tag{21}$$

where

$$\begin{aligned} \boldsymbol{\tau}_{s_t}^{o} &= \sum_{i=0}^{n} \left(m_i \ \boldsymbol{X}_t^{c_i} \times \left(\boldsymbol{g} - \ddot{\boldsymbol{X}}_t^{c_i} \right) - \dot{\boldsymbol{\mathcal{L}}}_t^{c_i} \right) \\ \boldsymbol{f}_{s_t}^{o} &= M\boldsymbol{g} - \sum_{i=0}^{n} m_i \ \ddot{\boldsymbol{X}}_t^{c_i} \end{aligned} \tag{22}$$

in which

$$\begin{aligned} \ddot{\boldsymbol{X}}_t^{c_i} &= \frac{\frac{\Delta \boldsymbol{X}_t^{c_i}}{\Delta t} s_{t-1} - \frac{\Delta \boldsymbol{X}_{t-1}^{c_i}}{\Delta t} s_t}{{s_t}^2 s_{t-1} \Delta t} \\ \dot{\boldsymbol{\mathcal{L}}}_t^{c_i} &= \boldsymbol{R}_t^i \left(\boldsymbol{I}_{c_i} \dot{\boldsymbol{\omega}}_t^i - \left(\boldsymbol{I}_{c_i} \boldsymbol{\omega}_t^i \right) \times \boldsymbol{\omega}_t^i \right) \end{aligned} \tag{23}$$

The angular velocity $\boldsymbol{\omega}_t^i$ can be obtained as follows

$$\begin{aligned} \boldsymbol{\omega}_t^i &= \left[\frac{\Delta \boldsymbol{R}_t^i}{s_t \Delta t} {\boldsymbol{R}_t^i}^T \right]^{\vee} \\ &= \frac{1}{s_t} \left[\left(\frac{\boldsymbol{R}_t^i - \boldsymbol{R}_{t-1}^i}{\Delta t} \right) {\boldsymbol{R}_t^i}^T \right]^{\vee} \\ &= \frac{1}{s_t} \left[\frac{\boldsymbol{I}_3 - \boldsymbol{R}_{t-1}^i {\boldsymbol{R}_t^i}^T}{\Delta t} \right]^{\vee} \end{aligned} \tag{24}$$

where $\boldsymbol{I}_3 \in \mathbb{R}^{3\times 3}$ is the identity matrix.

The angular acceleration $\dot{\boldsymbol{\omega}}_t^i$ can be calculated as follows

$$\begin{aligned} \dot{\boldsymbol{\omega}}_t^i &= \frac{\boldsymbol{\omega}_t^i - \boldsymbol{\omega}_{t-1}^i}{s_t \Delta t} \\ &= \frac{s_{t-1} \left[\frac{\boldsymbol{I}_3 - \boldsymbol{R}_{t-1}^i {\boldsymbol{R}_t^i}^T}{\Delta t} \right]^{\vee} - s_t \left[\frac{\boldsymbol{I}_3 - \boldsymbol{R}_{t-2}^i {\boldsymbol{R}_{t-1}^i}^T}{\Delta t} \right]^{\vee}}{{s_t}^2 s_{t-1} \Delta t} \end{aligned} \tag{25}$$

In similar way, we obtain

$$\begin{aligned} \dot{\boldsymbol{q}}_{s_t} &= \frac{\Delta \boldsymbol{q}_t}{s_t \Delta t} \\ \ddot{\boldsymbol{q}}_{s_t} &= \frac{\frac{\Delta \boldsymbol{q}_t}{\Delta t} s_{t-1} - \frac{\Delta \boldsymbol{q}_{t-1}}{\Delta t} s_t}{{s_t}^2 s_{t-1} \Delta t} \end{aligned} \tag{26}$$

1) Reformulation of the inequality constraints: It is clear that the constraints ($16 \sim 20$) are rational functions in the time parameterization function (s_t). In order to accelerate the convergence rate of the optimization problem and the accuracy of the obtained solution, one might think of transforming the inequality constraints from rational functions to polynomial functions.

In order to reformulate the inequality constraints, let us start by the constraint of ZMP (16):

$$\begin{aligned}\boldsymbol{p}^-_{s_t} \leq \boldsymbol{p}_{s_t} \leq \boldsymbol{p}^+_{s_t} \Leftrightarrow & \begin{bmatrix} \boldsymbol{p}_{s_t} \\ -\boldsymbol{p}_{s_t} \end{bmatrix} \leq \begin{bmatrix} \boldsymbol{p}^+_{s_t} \\ -\boldsymbol{p}^-_{s_t} \end{bmatrix} \\ \Leftrightarrow & \begin{bmatrix} -\boldsymbol{N}\left(\boldsymbol{n} \times \boldsymbol{\tau}^o_{s_t}\right) + \left(\boldsymbol{f}^o_{s_t}|\boldsymbol{n}\right)\boldsymbol{p}^+_{s_t} \\ \boldsymbol{N}\left(\boldsymbol{n} \times \boldsymbol{\tau}^o_{s_t}\right) - \left(\boldsymbol{f}^o_{s_t}|\boldsymbol{n}\right)\boldsymbol{p}^-_{s_t} \end{bmatrix} \leq 0\end{aligned} \tag{27}$$

As $D(s_t) = {s_t}^2 s_{t-1}\Delta t > 0 : \ \forall s_t$, we multiply the two sides of inequality (27) by $D(s_t)$ and obtain

$$\boldsymbol{p}^-_{s_t} \leq \boldsymbol{p}_{s_t} \leq \boldsymbol{p}^+_{s_t} \Leftrightarrow \begin{bmatrix} -\boldsymbol{N}\left(\boldsymbol{n} \times \tilde{\boldsymbol{\tau}}^o_{s_t}\right) + \left(\tilde{\boldsymbol{f}}^o_{s_t}|\boldsymbol{n}\right)\boldsymbol{p}^+_{s_t} \\ \boldsymbol{N}\left(\boldsymbol{n} \times \tilde{\boldsymbol{\tau}}^o_{s_t}\right) - \left(\tilde{\boldsymbol{f}}^o_{s_t}|\boldsymbol{n}\right)\boldsymbol{p}^-_{s_t} \end{bmatrix} \leq 0 \tag{28}$$

where

$$\begin{aligned}\tilde{\boldsymbol{\tau}}^o_{s_t} &= \sum_{i=0}^{n}\left(m_i\ \boldsymbol{X}^{c_i}_t \times \left(\tilde{\boldsymbol{g}} - \tilde{\ddot{\boldsymbol{X}}}^{c_i}_t\right) - \tilde{\dot{\boldsymbol{\mathcal{L}}}}^{c_i}_t\right) \\ \tilde{\boldsymbol{f}}^o_{s_t} &= M\tilde{\boldsymbol{g}} - \sum_{i=0}^{n} m_i\ \tilde{\ddot{\boldsymbol{X}}}^{c_i}_t\end{aligned} \tag{29}$$

in which

$$\begin{aligned}\tilde{\boldsymbol{g}} &= \boldsymbol{g}\ {s_t}^2 s_{t-1}\Delta t \\ \tilde{\ddot{\boldsymbol{X}}}^{c_i}_t &= \frac{\Delta \boldsymbol{X}^{c_i}_t}{\Delta t}s_{t-1} - \frac{\Delta \boldsymbol{X}^{c_i}_{t-1}}{\Delta t}s_t \\ \tilde{\dot{\boldsymbol{\mathcal{L}}}}^{c_i}_t &= \boldsymbol{R}^i_t\left(\boldsymbol{I}_{c_i}\tilde{\dot{\boldsymbol{\omega}}}^i_t - \frac{s_{t-1}}{\Delta t}\left(\boldsymbol{I}_{c_i}\tilde{\boldsymbol{\omega}}^i_t\right) \times \tilde{\boldsymbol{\omega}}^i_t\right) \\ \tilde{\boldsymbol{\omega}}^i_t &= \left[\frac{\boldsymbol{I}_3 - \boldsymbol{R}^i_{t-1}\boldsymbol{R}^{i\,T}_t}{\Delta t}\right]^\vee \\ \tilde{\dot{\boldsymbol{\omega}}}^i_t &= s_{t-1}\left[\frac{\boldsymbol{I}_3 - \boldsymbol{R}^i_{t-1}\boldsymbol{R}^{i\,T}_t}{\Delta t}\right]^\vee - s_t\left[\frac{\boldsymbol{I}_3 - \boldsymbol{R}^i_{t-2}\boldsymbol{R}^{i\ \ T}_{t-1}}{\Delta t}\right]^\vee\end{aligned} \tag{30}$$

It is clear that $\tilde{\boldsymbol{\tau}}^o_{s_t}$ and $\tilde{\boldsymbol{f}}^o_{s_t}$ are polynomial function with respect to s_t. As a result, the inequality constraint of ZMP trajectory becomes polynomial function with respect to s_t.

In similar way the constraints (17 - 18) can be transformed into the following equivalent ones

$$\left(\tilde{\boldsymbol{f}}^o_{s_t}|\boldsymbol{n}\right) < 0 \tag{31}$$

$$\mu\ \left(\tilde{\boldsymbol{f}}^o_{s_t}|\boldsymbol{n}\right) < \left(\tilde{\boldsymbol{\tau}}^o_{s_t}|\boldsymbol{n}\right) < -\mu\ \left(\tilde{\boldsymbol{f}}^o_{s_t}|\boldsymbol{n}\right) \tag{32}$$

The velocity and acceleration limits can be reformulated as follows

$$s_t\ \dot{\boldsymbol{q}}^- \leq \frac{\Delta \boldsymbol{q}_t}{\Delta t} \leq s_t\ \dot{\boldsymbol{q}}^+ \tag{33}$$

$${s_t}^2 s_{t-1}\ \ddot{\boldsymbol{q}}^- \leq \frac{\Delta \boldsymbol{q}_t}{(\Delta t)^2}s_{t-1} - \frac{\Delta \boldsymbol{q}_{t-1}}{(\Delta t)^2}s_t \leq {s_t}^2 s_{t-1}\ \ddot{\boldsymbol{q}}^+ \tag{34}$$

As a result, all constraints of the optimization problem are transformed into polynomial functions with respect to s_t. Recall that the original constraints of the optimization problem before the reformulation are rational function with respect to s_t. By using this reformulation the convergence rate of the optimization problem has been considerably improved.

2) Discretization of The Solution Space: As it is well known, the space of the admissible solutions of the minimization problem (14) is in fact very large. In order to transform this space to a smaller dimensional space, we can use a basis of shape functions (e.g. cubic B-spline functions).

Let us consider a basis of shape functions $\boldsymbol{B}_t$ that is defined as follows

$$\boldsymbol{B}_t = \begin{bmatrix} B^1_t & B^2_t & \cdots & B^l_t \end{bmatrix}^T \tag{35}$$

where B^i_t denotes the value of shape function number i at the instant t. The dimension of $\boldsymbol{B}_t$ is l which defines the dimension of the basis of shape functions.

The projection of s_t into the basis of shape functions B_t can be given by the following formula

$$s_t = \sum_{i=1}^{l} s^i_B\ B^i_t = \boldsymbol{s}^T_B\ \boldsymbol{B}_t \tag{36}$$

Thus, the optimization problem (14) can be rewritten as follows

$$\min_{\boldsymbol{s}_B}\left\{\sum_{k=1}^{l} s^k_B \int_{t=0}^{T} B^k_t\ dt + \int_{t=0}^{T} L\left(\boldsymbol{q}_{s_B}, \dot{\boldsymbol{q}}_{s_B}, \ddot{\boldsymbol{q}}_{s_B}\right)dt\right\}$$

subject to

$$\begin{aligned} & \boldsymbol{s}^T_B\ \boldsymbol{B}_t > 0 \\ & \boldsymbol{N}\left(\boldsymbol{n} \times \tilde{\boldsymbol{\tau}}^o_{\boldsymbol{s}_B}\right) + \left(\tilde{\boldsymbol{f}}^o_{\boldsymbol{s}_B}|\boldsymbol{n}\right)\boldsymbol{p}^+_{\boldsymbol{s}_B} \leq 0 \\ & \boldsymbol{N}\left(\boldsymbol{n} \times \tilde{\boldsymbol{\tau}}^o_{\boldsymbol{s}_B}\right) - \left(\tilde{\boldsymbol{f}}^o_{\boldsymbol{s}_B}|\boldsymbol{n}\right)\boldsymbol{p}^-_{\boldsymbol{s}_B} \leq 0 \\ & \left(\tilde{\boldsymbol{f}}^o_{\boldsymbol{s}_B}|\boldsymbol{n}\right) < 0 \\ \mu\ \left(\tilde{\boldsymbol{f}}^o_{\boldsymbol{s}_B}|\boldsymbol{n}\right) < \ & \left(\tilde{\boldsymbol{\tau}}^o_{\boldsymbol{s}_B}|\boldsymbol{n}\right)\ < -\mu\ \left(\tilde{\boldsymbol{f}}^o_{\boldsymbol{s}_B}|\boldsymbol{n}\right) \\ \dot{\boldsymbol{q}}^-\ \boldsymbol{s}^T_B\ \boldsymbol{B}_t \leq \ & \frac{\Delta \boldsymbol{q}_t}{\Delta t}\ \leq \dot{\boldsymbol{q}}^+\ \boldsymbol{s}^T_B\ \boldsymbol{B}_t \\ \ddot{\boldsymbol{q}}^-_{\boldsymbol{s}_B} \leq \ & \frac{\Delta \boldsymbol{q}_t}{(\Delta t)^2}\boldsymbol{s}^T_B\ \boldsymbol{B}_{t-1} - \frac{\Delta \boldsymbol{q}_{t-1}}{(\Delta t)^2}\boldsymbol{s}^T_B\ \boldsymbol{B}_t\ \leq \ddot{\boldsymbol{q}}^+_{\boldsymbol{s}_B}\end{aligned} \tag{37}$$

in which

$$\begin{aligned}\ddot{\boldsymbol{q}}^+_{\boldsymbol{s}_B} &= \boldsymbol{B}^T_t\boldsymbol{s}_B\boldsymbol{s}^T_B\boldsymbol{B}_t\boldsymbol{s}^T_B\boldsymbol{B}_{t-1}\ \ddot{\boldsymbol{q}}^+ \\ \ddot{\boldsymbol{q}}^-_{\boldsymbol{s}_B} &= \boldsymbol{B}^T_t\boldsymbol{s}_B\boldsymbol{s}^T_B\boldsymbol{B}_t\boldsymbol{s}^T_B\boldsymbol{B}_{t-1}\ \ddot{\boldsymbol{q}}^-\end{aligned} \tag{38}$$

Thus, the optimization problem has been transformed into finding the vector $\boldsymbol{s}_B \in \mathbb{R}^l$.

In order to transform this optimization problem into a classical optimization problem, let us introduce a constant

$\epsilon \in \mathbb{R} : 0 < \epsilon \ll 1$ and the following definitions

$$J\left(\boldsymbol{s}_B\right) = \left\{ \sum_{k=1}^{l} s_B^k \int_{t=0}^{T} B_t^k \, dt + \int_{t=0}^{T} L\left(\boldsymbol{q}_{s_B}, \dot{\boldsymbol{q}}_{s_B}, \ddot{\boldsymbol{q}}_{s_B}\right) dt \right\}$$

$$G(\boldsymbol{s}_B) = \begin{bmatrix} -\boldsymbol{s}_B^T \ \boldsymbol{B}_t + \epsilon \\ \boldsymbol{N}\left(\boldsymbol{n} \times \tilde{\boldsymbol{\tau}}^o_{\boldsymbol{s}_B}\right) + \left(\tilde{\boldsymbol{f}}^o_{\boldsymbol{s}_B} | \boldsymbol{n}\right) \boldsymbol{p}^+_{\boldsymbol{s}_B} \\ \boldsymbol{N}\left(\boldsymbol{n} \times \tilde{\boldsymbol{\tau}}^o_{\boldsymbol{s}_B}\right) - \left(\tilde{\boldsymbol{f}}^o_{\boldsymbol{s}_B} | \boldsymbol{n}\right) \boldsymbol{p}^-_{\boldsymbol{s}_B} \\ \left(\tilde{\boldsymbol{f}}^o_{\boldsymbol{s}_B} | \boldsymbol{n}\right) + \epsilon \\ -\left(\tilde{\boldsymbol{\tau}}^o_{\boldsymbol{s}_B} | \boldsymbol{n}\right) + \mu \left(\tilde{\boldsymbol{f}}^o_{\boldsymbol{s}_B} | \boldsymbol{n}\right) + \epsilon \\ \left(\tilde{\boldsymbol{\tau}}^o_{\boldsymbol{s}_B} | \boldsymbol{n}\right) + \mu \left(\tilde{\boldsymbol{f}}^o_{\boldsymbol{s}_B} | \boldsymbol{n}\right) + \epsilon \\ \frac{\Delta \boldsymbol{q}_t}{\Delta t} - \dot{\boldsymbol{q}}^+ \ \boldsymbol{s}_B^T \ \boldsymbol{B}_t \\ -\frac{\Delta \boldsymbol{q}_t}{\Delta t} + \dot{\boldsymbol{q}}^- \ \boldsymbol{s}_B^T \ \boldsymbol{B}_t \\ \frac{\Delta \boldsymbol{q}_t}{(\Delta t)^2} \boldsymbol{s}_B^T \ \boldsymbol{B}_{t-1} - \frac{\Delta \boldsymbol{q}_{t-1}}{(\Delta t)^2} \boldsymbol{s}_B^T \ \boldsymbol{B}_t - \ddot{\boldsymbol{q}}^+_{\boldsymbol{s}_B} \\ -\frac{\Delta \boldsymbol{q}_t}{(\Delta t)^2} \boldsymbol{s}_B^T \ \boldsymbol{B}_{t-1} + \frac{\Delta \boldsymbol{q}_{t-1}}{(\Delta t)^2} \boldsymbol{s}_B^T \ \boldsymbol{B}_t + \ddot{\boldsymbol{q}}^-_{\boldsymbol{s}_B} \end{bmatrix} \tag{39}$$

Thus the optimization problem (37) can be transformed to the following classical form

$$\begin{aligned} & \min_{\boldsymbol{s}_B} \ J(\boldsymbol{s}_B) \\ \text{subject to} & \\ & G(\boldsymbol{s}_B) \leq 0 \end{aligned} \tag{40}$$

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 [15], [16]. Based on the augmented Lagrange multiplier method, the optimization problem (40) is transformed into the minimization of the following function

$$\min_{\boldsymbol{s}_B, \lambda_\psi} \tilde{J}(\boldsymbol{s}_B, \lambda_\psi) = J(\boldsymbol{s}_B) + \lambda_\psi^T \psi + \frac{1}{2} \sigma \psi^T \psi \tag{41}$$

where $\psi = \max\left\{G(\boldsymbol{s}_B), \frac{-1}{\sigma}\lambda_\psi\right\}$, and $\sigma > 0$. Then there exist λ_ψ^* such that $\boldsymbol{s}_B^*$ is an unconstrained local minimum of $\tilde{J}(\boldsymbol{s}_B, \lambda_\psi^*)$ for all σ smaller than some finite $\bar{\sigma}$.
To solve the unconstrained optimization problem of $\tilde{J}(\boldsymbol{s}_B, \lambda_\psi)$ with respect to $\boldsymbol{s}_B$, one can use Gauss-Newton method.

Note that the function $\tilde{J}(\boldsymbol{s}_B, \lambda_\psi)$ is differentiable in $\boldsymbol{s}_B$ if and only if $J(\boldsymbol{s}_B)$ and $G(\boldsymbol{s}_B)$ are differentiable in $\boldsymbol{s}_B$.

As we have prove $G(\boldsymbol{s}_B)$ is polynomial with respect to $\boldsymbol{s}_B$ and its derivative can be calculated easily. The cost function $J(\boldsymbol{s}_B)$ is defined by the user and it is supposed to be derivable with respect to $\boldsymbol{s}_B$ and its derivative is continuous.

So in this case we can write

$$\frac{\partial \tilde{J}(\boldsymbol{s}_B, \lambda_\psi)}{\partial \boldsymbol{s}_B} = \frac{\partial J(\boldsymbol{s}_B)}{\partial \boldsymbol{s}_B} + \max\left\{0, \lambda_\psi + \sigma G(\boldsymbol{s}_B)\right\}^T \ \frac{\partial G(\boldsymbol{s}_B)}{\partial \boldsymbol{s}_B} \tag{42}$$

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

$$\lambda_\psi^{k+1} = \lambda_\psi^k + \sigma \psi(\boldsymbol{s}_B^k) \tag{43}$$

where $\boldsymbol{s}_B^k$ is the unconstrained minimum of $\tilde{J}(\boldsymbol{s}_B, \lambda_\psi^k)$. Such updating rule will generate a sequence λ_ψ^k converges to λ_ψ^* [17]. In practice, a good schedule is to choose a moderate σ^0, and increase it as follows

$$\sigma^{k+1} = \alpha \sigma^k \tag{44}$$

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

For more details on the algorithm of augmented Lagrange multiplier method see [18], [15], [17].

3) Implementation Algorithm: The algorithm of the implementation can be summarized as follows

1) Given a path which is supposed to be statically stable.
2) Split the path into various time segments depending on the place and shape of support polygon during the motion. The support polygon for each time segment is fixed.
3) Choose Δt, this value is usually determined by the sampling period of the humanoid robot's control loop. For instance, for the humanoid robot HRP-2 $\Delta t = 5.0 \times 10^{-3} sec$.
4) Transform each time segment of the initial path into a trajectory by considering a uniform time distribution function $(s_t = 1, \forall t)$.
5) Calculate $\Delta \boldsymbol{q}_t$, $\boldsymbol{X}_t^{c_i}$, $\Delta \boldsymbol{X}_t^{c_i}$ and $\boldsymbol{R}_t^i$ $(i = 0, 1, \cdots, n)$, for each time segment of the path. Recall that the path is defined by the parameters $\left\{\boldsymbol{q}_t, \boldsymbol{X}_t^{c_0}, \boldsymbol{R}_t^0\right\}$, where $\boldsymbol{q}_t$ designs the configuration vector of the humanoid robot joints, $\boldsymbol{X}_t^{c_0}$ and $\boldsymbol{R}_t^0$ denote the position and the rotation matrix in the Euclidean space of the free-flyer joint (pelvis joint).
6) Calculate the cubic B-spline functions.
7) Solve the optimization problem (37) for each time segment with the initial solution obtained by applying the above steps.

B. Case 2: joint velocity and torque limits

In this case, we suppose that we have a path which consists of K points. Similarly to the precedent case, we transform this path into a trajectory by considering a uniform time distribution function. We denote the time horizon $T = K \, \Delta t$, where Δt is the sampling time.

In this case, the time parameterization problem of transforming the initial path to a dynamically stable trajectory within the joint velocity and torque limits of the humanoid robot can be expressed by the following optimization problem

$$\min_{s_t} J(s_t) = \min_{s_t} \left\{ \int_{t=0}^{T} s_t dt + \int_{t=0}^{T} L\left(\boldsymbol{q}_{s_t}, \dot{\boldsymbol{q}}_{s_t}, \tau_{s_t}, s_t\right) dt \right\}$$

subject to

$$
\begin{aligned}
s_t &> 0\\
\boldsymbol{p}^-_{s_t} \le &\boldsymbol{p}_{s_t} \le \boldsymbol{p}^+_{s_t}\\
\left(\boldsymbol{f}^o_{s_t}|\boldsymbol{n}\right) &< 0\\
\mu\,\left(\boldsymbol{f}^o_{s_t}|\boldsymbol{n}\right) < \left(\boldsymbol{\tau}^o_{s_t}|\boldsymbol{n}\right) &< -\mu\,\left(\boldsymbol{f}^o_{s_t}|\boldsymbol{n}\right)\\
\dot{\boldsymbol{q}}^- \le &\dot{\boldsymbol{q}}_{s_t} \le \dot{\boldsymbol{q}}^+\\
\boldsymbol{\tau}^- \le &\boldsymbol{\tau}_{s_t} \le \boldsymbol{\tau}^+
\end{aligned}
\tag{45}
$$

where $\boldsymbol{\tau}_{s_t}$ designs the vector of exerted torques on the humanoid robot's joints. $\boldsymbol{\tau}^-$ and $\boldsymbol{\tau}^+$ denote the lower and upper limits of the torque vector.

By including the torque limits into the time parameterization problem, the obtained motion will not damage the motors located in the articulated joints, and the robot will not stop because of high exerted torques.

However, solving the time parameterization problem, in this case, becomes more difficult. This is because the equation of motion of the humanoid robot should be taken into account, this equation is a complicate and high nonlinear equation.

The motion equation has the following form

$$\mathbf{M}\left(\mathbf{q_t}\right)\ddot{\mathbf{q}}_\mathbf{t} + \mathbf{C}\left(\mathbf{q_t},\,\dot{\mathbf{q}}_\mathbf{t}\right) = \boldsymbol{\tau}_t \tag{46}$$

where $\mathbf{M}\left(\mathbf{q_t}\right)$ is the mass matrix, $\mathbf{C}\left(\mathbf{q_t},\,\dot{\mathbf{q}}_\mathbf{t}\right)$ is the Coriolis matrix, and includes gravity and other forces . At a glance, Eq. (46) might appear to be simple; nevertheless the analytical expressions for a simple six-axis industrial robotic arm are extremely complex.

In order to solve the optimization problem (45), the derivative of $\boldsymbol{\tau}_{s_t}$ with respect to s_t should be calculated. This derivative can be calculated as follows

$$\frac{d\boldsymbol{\tau}_{s_t}}{ds_t} = \frac{\partial\boldsymbol{\tau}_{s_t}}{\partial\boldsymbol{q}_{s_t}}\frac{d\boldsymbol{q}_{s_t}}{ds_t} + \frac{\partial\boldsymbol{\tau}_{s_t}}{\partial\dot{\boldsymbol{q}}_{s_t}}\frac{d\dot{\boldsymbol{q}}_{s_t}}{ds_t} + \frac{\partial\boldsymbol{\tau}_{s_t}}{\partial\ddot{\boldsymbol{q}}_{s_t}}\frac{d\ddot{\boldsymbol{q}}_{s_t}}{ds_t} \tag{47}$$

Because of the path of the vector q_{s_t} is expressed as discreet points in the configuration space, and the time parameterization algorithm will not change the positions of these points in the configuration space, therefore $\frac{d\boldsymbol{q}_{s_t}}{ds_t} \approx 0$. The quantities $\frac{d\dot{\boldsymbol{q}}_{s_t}}{ds_t}$ and $\frac{d\ddot{\boldsymbol{q}}_{s_t}}{ds_t}$ can be calculated easily by using the finite difference approximation of Eq. (26).

The main difficulty is the calculation of the quantities $\frac{\partial\boldsymbol{\tau}_{s_t}}{\partial\dot{\boldsymbol{q}}_{s_t}}$ and $\frac{\partial\boldsymbol{\tau}_{s_t}}{\partial\ddot{\boldsymbol{q}}_{s_t}}$. Although, in principle, these quantities can be numerically approximated analogously by using the finite difference, we have observed that this approximation leads to illconditioning, and poor convergence behavior. This is because of the high non-linearity of the motion equation (46).

To overcome this problem, an analytical formulation can be derived of the quantities $\frac{\partial\boldsymbol{\tau}_{s_t}}{\partial\dot{\boldsymbol{q}}_{s_t}}$ and $\frac{\partial\boldsymbol{\tau}_{s_t}}{\partial\ddot{\boldsymbol{q}}_{s_t}}$ by using the recursive dynamic algorithm proposed in [19], [20], which is based on Lie groups and algebras. (for more details see [19], [20], [21], [22]).

Note that the quantities $\frac{\partial\boldsymbol{\tau}_{s_t}}{\partial\dot{\boldsymbol{q}}_{s_t}}$ and $\frac{\partial\boldsymbol{\tau}_{s_t}}{\partial\ddot{\boldsymbol{q}}_{s_t}}$ are calculated only one time and then they are used as constants in the time parameterization algorithm. Therefore the torque limit constraint can be reformulated as polynomial function with respect to s_t analogously to the previous case.

Thus, the time parameterization problem can be transformed into the classical optimization problem

$$
\begin{aligned}
&\min_{s_t}\;J(s_t)\\
\text{subject to}\quad&\\
&G(s_t) \le 0
\end{aligned}
\tag{48}
$$

Similarly to the previous case, the above problem can be solved using the discretization of solution space and the augmented Lagrange multiplier method.

C. Discussion

In this section, we discuss how to split the initial path into time segments according to the support polygon and the global optimal solution of the optimization problem.

- The support polygon is a function of $\boldsymbol{s}_B$, and it depends on the horizontal position of CoM. However, as the given path is a statically stable one, it can be split into various sections. Each section is a statically stable path which has a fixed support polygon that is independent of $\boldsymbol{s}_B$. Note that the polygon of support can be determined according to the position of feet and by verifying if the feet are in contact with the ground. This determination can be done independently from $\boldsymbol{s}_B$, this is because the position vector of the joints $\boldsymbol{X}_t$ is an invariant quantity in the time parameterization algorithm.
- The optimal solution obtained by solving the optimization problem of time parameterization is a local optimum. However, if we prove that the optimization problem is *convex*, then the obtained solution is the global optimum. To this end, we should prove that the function $\tilde{J}$ of Eq. (41) is a convex function. The functions J and ψ of Eq. (41) are convex, because the first one is defined by the user and it is supposed to be convex and the second one is convex on account of the convexity of $G(\boldsymbol{s}_B)$. Consequently, the function $\tilde{J}$ is convex and the obtained local optimum solution is the *global optimum*.

IV. Case Studies and Experimental Results

The considered example is a collision-free reaching motion in a cluttered environment. Fig. 2 shows snapshots of the simulated motion. In the initial configuration, the robot is standing on its right foot and surrounded by a torus, a cylinder and a box. In this example, the task for the humanoid robot is to move its left hand to a specified position, and at same time to keep the projection of its CoM inside of the support polygon (statically stable motion). Producing such kind of motion is a very challenging task. This is because the environment is very cluttered, and the support polygon is small (standing on the right foot). During the motion the humanoid robot should avoid the collision between its left leg and an obstacle which consists of a cylinder and a box, at the same time the left hand should avoid the collision with the torus to reach the desired position. This collision-free path is created using an efficient method proposed by Kanehiro *et al* [14].

Once the collision-free path is available, this path is converted to a trajectory using a uniform time distribution function.

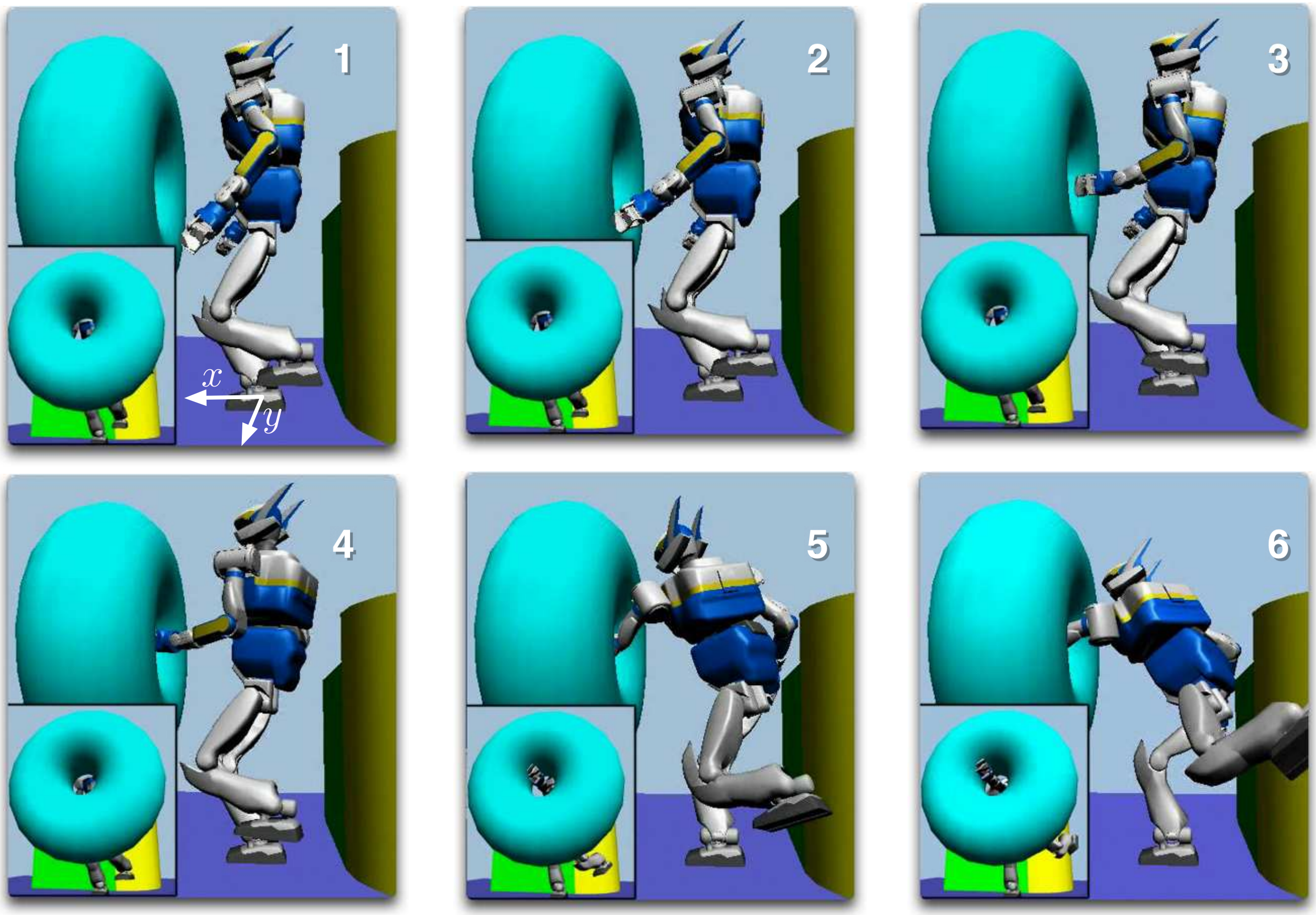


Fig. 2. Snapshots of the simulated whole body reaching with collision avoidance

The sampling time Δt has been chosen to be equal to $5.0 \times 10^{-3} sec$. The duration of the initial trajectory is equal to $77 sec$.

In this section, *we will consider three scenarios:*

A. First Scenario:

The time parameterization problem has the form of Eq. (37), in which $L\left(\boldsymbol{q}_{s_B}, \dot{\boldsymbol{q}}_{s_B}, \ddot{\boldsymbol{q}}_{s_B}\right) = 0$. In this scenario, the objective of time parameterization is to transform the initial path into a minimum time and dynamically stable trajectory.

The effect of the number of B-spline functions on the duration of the obtained trajectory and the required number of iterations to reach the optimal solution is reported in Table I.

TABLE I
DURATION OF OBTAINED TRAJECTORIES AND NUMBER OF ITERATIONS

Number of B-spline	Duration of obtained trajectory (sec)	Number of iterations
10	18.35	4
20	14.81	9
40	12.68	19
120	8.92	27

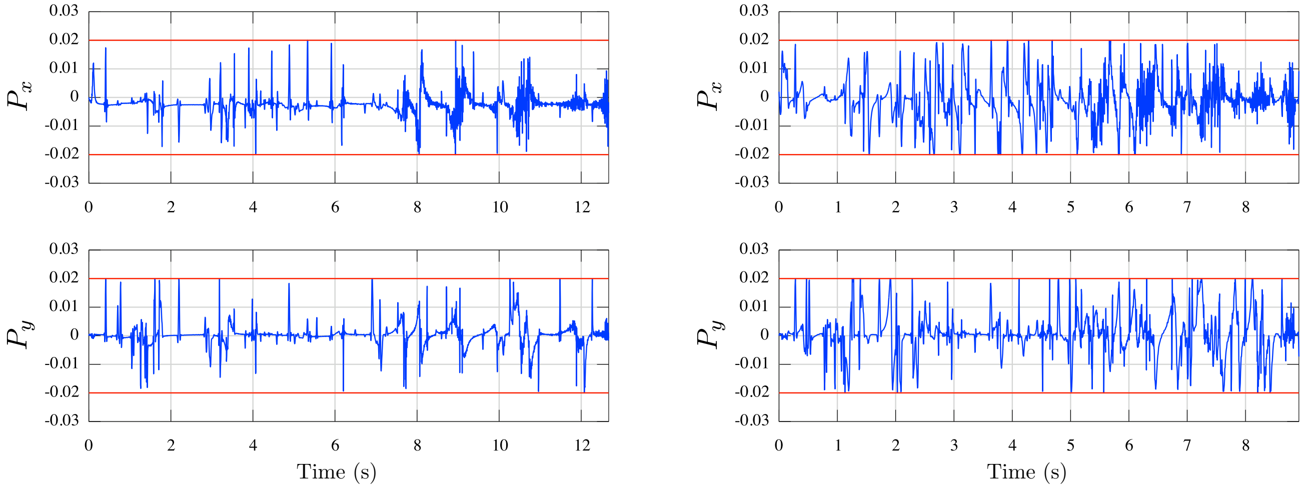
We have chosen to report the number of iteration instead of the computational time. This is because we are using MATALB language to solve the optimization problems. However, the number of iterations of the augmented Lagrange multiplier method gives a good idea of the computational time [23], [24], [17]. In fact, the augmented Lagrange multiplier method is a fast and reliable method because it does not require the inversion of matrices which have, in our case, very huge dimensions.

From Table I, it is clear that the duration of the obtained trajectory decrease by increasing the number of B-spline functions. However, the number of iterations to reach the optimal solution increases while increasing the number of B-spline functions.

The trajectories of ZMP (P_x and P_y) corresponding to basis of B-spline functions of 20 and 120 functions are given in Fig. 3. The directions x and y for the humanoid robot are drawn in Fig. 2.

From Fig. 3, we observe that the trajectories of ZMP are always maintained inside of the polygon of support. However the fluctuations of ZMP occur more frequently when a basis of 120 Bsplines is used, this is because the motion is faster than the motion obtained by using a basis of 40 Bsplines.

In Fig. 4, the initial, the reference and the executed trajectories of the roll axis of right shoulder joint are given. The initial trajectory is directly obtained from the statically



(a) Using a basis of 40 Bsplines

(b) Using a basis of 120 Bsplines

Fig. 3. First scenario: ZMP trajectories, and the safety zones for dynamical stability which are designed by the red lines. The time scale of the two figures is not the same because the time parameterization functions are different

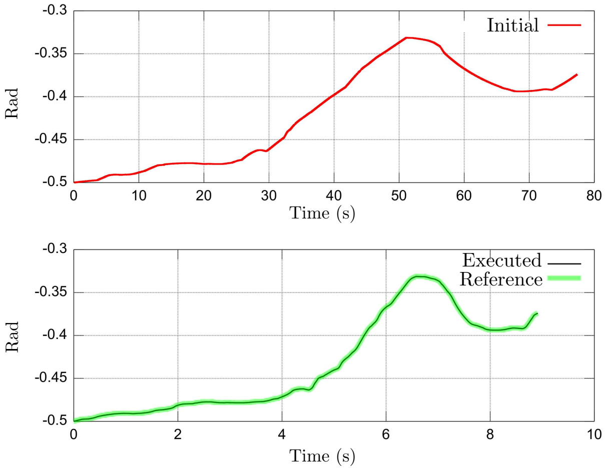


Fig. 4. First scenario (minimum time trajectory): The initial trajectory of the roll axis of right shoulder joint which is obtained directly from the statically stable path (figure above). The reference and the executed trajectories of the real experiment on the humanoid robot HRP-2 (figure below). The time scale after time parameterization (figure below) is much smaller than that of the initial trajectory (figure above)

stable path by considering a uniform distribution of the time parameterization function (s_t). The reference trajectory is the the minimum time trajectory obtained by using a basis of 120 B-spline functions. We can observe that the executed trajectory of the real experiment on the humanoid robot HRP-2 is exactly following the reference trajectory. The humanoid robot HRP-2 is a position controlled robot, that means the trajectories of the humanoid robot's joints are tracked using a high-gain PID controller.

The time parameterization functions s_t which correspond to basis of B-spline functions of 20, 40 and 120 functions are given in Fig. 5.

Snapshots of the conducted motion applied to the humanoid robot HRP2 are given in Fig. 9.

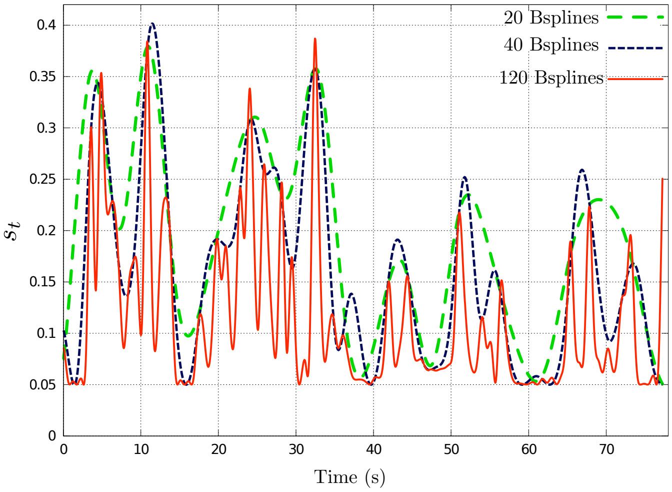


Fig. 5. First scenario (minimum time trajectory): Time parameterization function (s_t)

B. Second Scenario:

In this scenario, the time parameterization problem has also the form of Eq. (37), in which $L\left(\boldsymbol{q}_{s_B}, \dot{\boldsymbol{q}}_{s_B}, \ddot{\boldsymbol{q}}_{s_B}\right) = \beta\, \dddot{\boldsymbol{X}}_t^{left\ hand}$. The function L captures the jerk function of the left hand joint, this joint plays the role of end-effector in the computation of the initial path. The jerk function can be approximated as follows

$$\dddot{\boldsymbol{X}}_t = \frac{\ddot{\boldsymbol{X}}_t - \ddot{\boldsymbol{X}}_{t-1}}{s_t \Delta t} \qquad (49)$$

where $\ddot{\boldsymbol{X}}_t$ is expressed as a function of s_t as in Eq. (23). The duration of the obtained trajectories and the required number of iteration to reach the optimal solution is reported in Table II.

TABLE II
DURATION OF OBTAINED TRAJECTORIES AND NUMBER OF ITERATIONS

β	Duration of obtained trajectory (sec)	Number of iterations
10^{-5}	10.73	16
10^{-4}	14.13	31
10^{-3}	23.18	39

The time parameterization functions s_t which correspond to $\beta = 10^{-4}$ and 10^{-3} and the two configurations which are related to high variation in the jerk function are given in Fig. 6.

The high variation of the jerk function in the first configuration occurs because the left hand stops near the torus in order to avoid the collision, in the second configuration the left hand is inside the torus and it changes the direction of its motion to avoid collision and at the same time to reach the desired position.

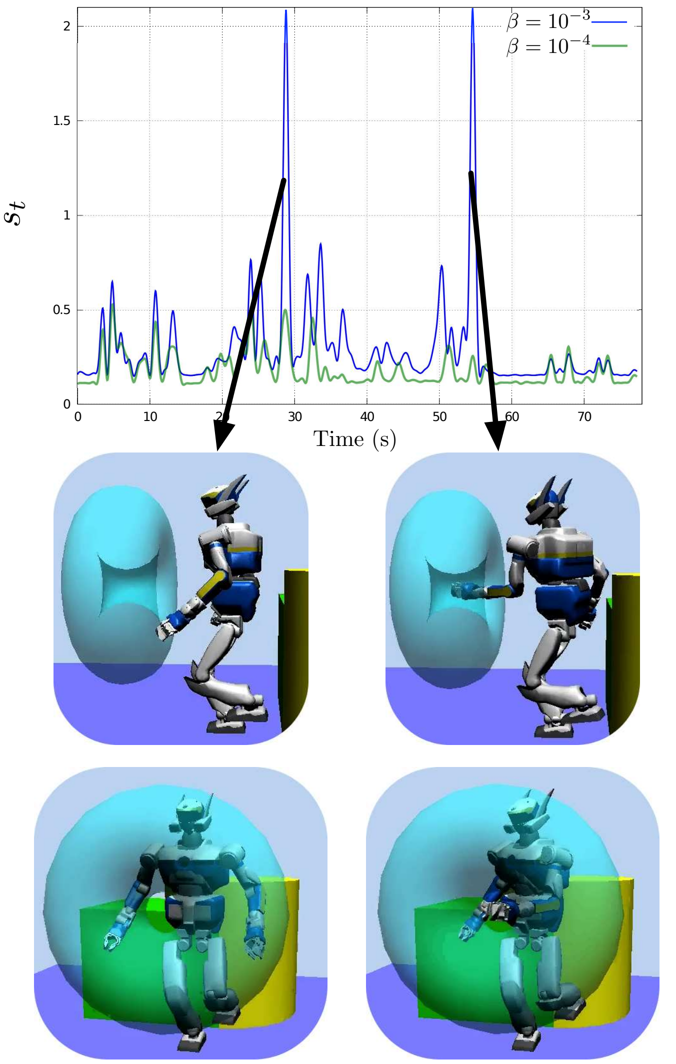


Fig. 6. Second scenario: time parameterization function (s_t) and the two configurations (front and side views) which are related to high variation in the jerk function

C. Third Scenario:

In this scenario, the time parameterization problem has the form of Eq. (45), in which $L\left(\boldsymbol{q}_{s_B}, \dot{\boldsymbol{q}}_{s_B}, \ddot{\boldsymbol{q}}_{s_B}\right) = 0$. The objective is also to find the minimum time trajectory and dynamically stable. The difference between this scenario and the first one is that the torque limits of the humanoid robot are taken into account in this scenario.

A comparison between the time parameterization function s_t which is obtained in this scenario and that one of the first scenario is presented in Fig. 7.

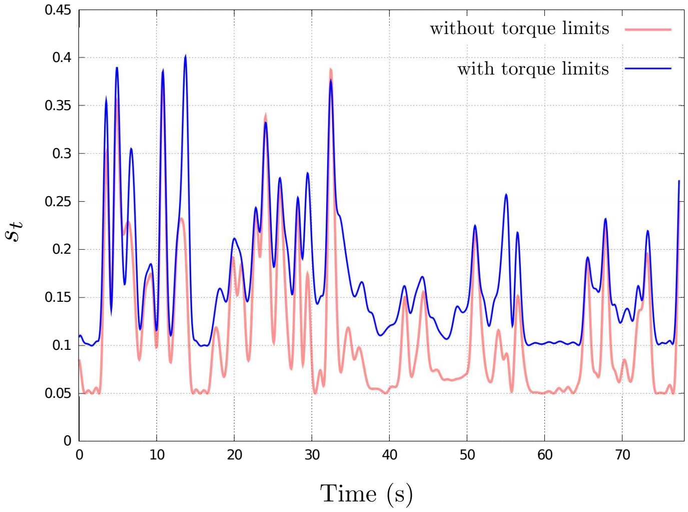


Fig. 7. Third scenario: time parameterization function (s_t)

In order to obtain a safe motion which is not near the physical limits of the humanoid robot, we used a safety margin of 20 percent of the humanoid robot's torque limits.

A comparison between the applied torque on the yaw axis of the hip joint with and without the consideration of torque limits is given in Fig. 8. This figure shows that the constraints on the torque limits have been successfully respected, on the other hand the duration of the obtained trajectory is more than that one of the first scenario. This is in order to respect the torque limits.

V. CONCLUSION

In this paper, we proposed a numerical method to solve the time parameterization problem of humanoid robot paths. The main contribution of this method is transforming a statically stable path into a minimum time and dynamically stable trajectory which respects the physical limits of the humanoid robot's joints. We have shown that not only minimizing the trajectory time but also some energy criteria such as the jerk function can be considered.

The initial statically stable path can be calculated using inverse kinematic methods or the motion planning methods. This path, by definition, is a pure geometric description of the motion.

The effectiveness of the proposed method has been validated using the humanoid robot HRP-2.

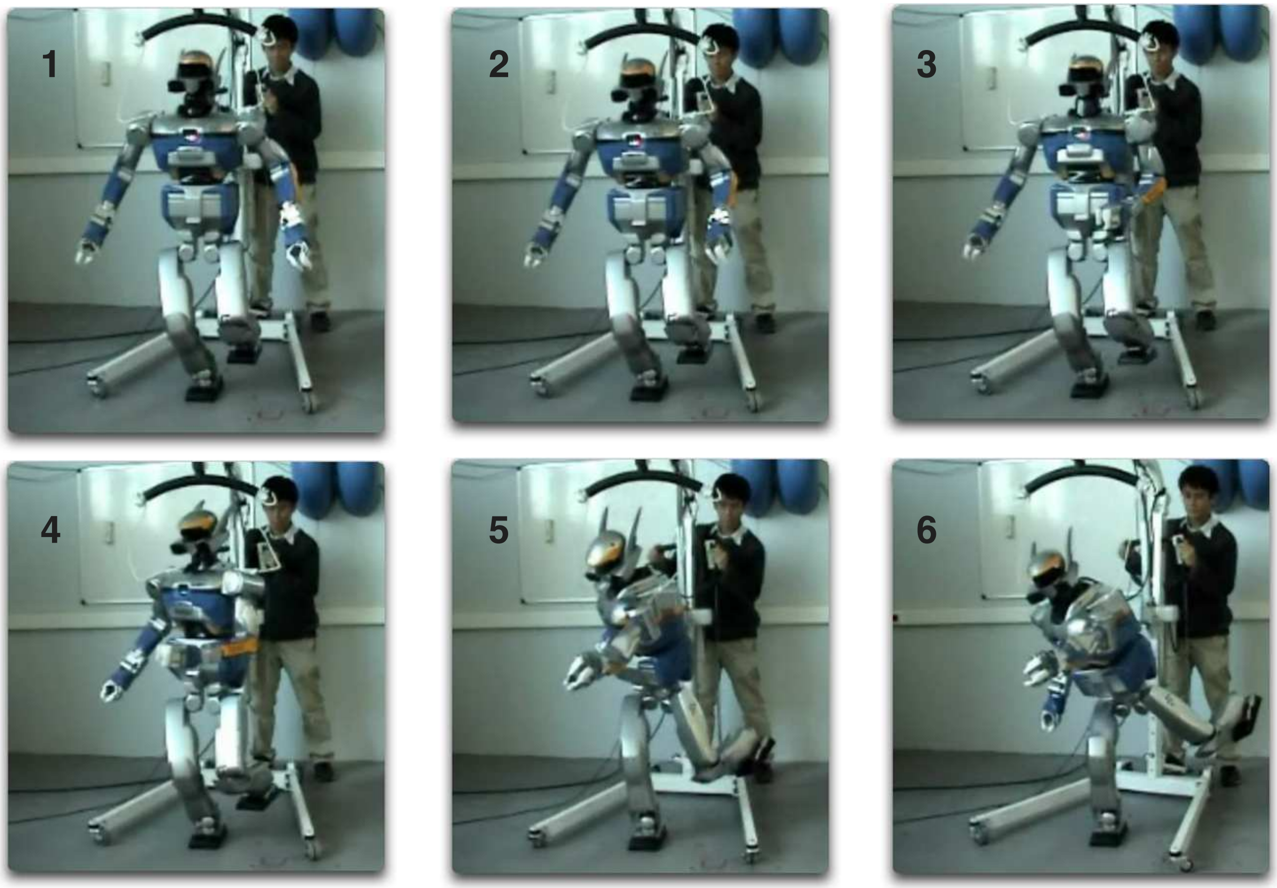


Fig. 9. Snapshots of the real experiment using the humanoid robot HRP-2

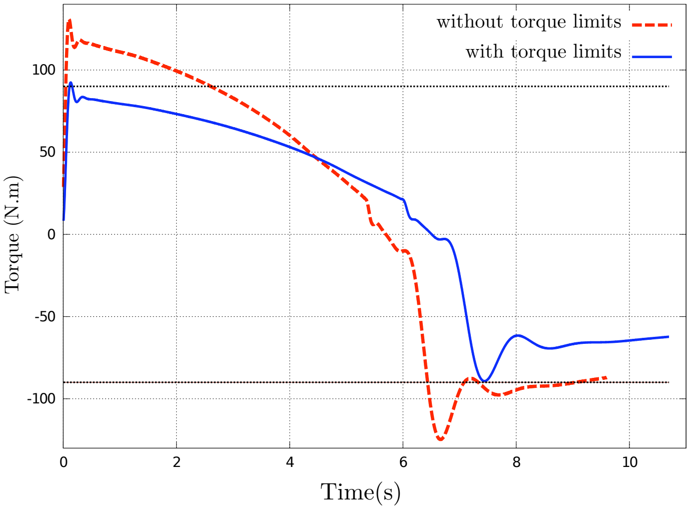


Fig. 8. Third scenario: applied torque on the yaw axis of the hip joint.

Acknowledgment

This research was partially supported by a Grant-in-Aid for Scientific Research from the Japan Society for the Promotion of Science (20-08816), and JST-CNRS Strategic Japanese-French Cooperative Program "Robot motion planning and execution through online information structuring in real-world environment". The first author thanks the JSPS for a postdoctoral research fellowship.

References

[1] L. Sciavicco and B. Siciliano, *Modelling and Control of Robot Manipulators*, 2nd, Ed. Springer-Verlag, 2000.

[2] ——, *Modelling and Control of Robot Manipulators*. Springer-Verlag, 2000, ch. Trajectory Planning, pp. 185–212.

[3] K. Shin and N. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Trans. Automatic Control*, vol. 30, no. 6, pp. 531–541, Jun 1985.

[4] V. Rajan, "Minimum Time Trajectory Planning," in *IEEE International Conference on Robotics and Automation*, vol. 2, 1985, pp. 759–764.

[5] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths," *International Journal of Robotics Research*, vol. 4, no. 3, 1985.

[6] J. Slotine and H. Yang, "Improving the efficiency of time optimal path following algorithms," *IEEE Trans. Automatic Control*, vol. 5, no. 1, pp. 118–124, 1989.

[7] M. Renaud and J. Y. Fourquet, "Time-optimal motions of robot manipulators including dynamics," *The robotics review 2, MIT Press*, pp. 225–259, 1992.

[8] F. Lamiraux and J.-P. Laumond, "From Paths to Trajectories for Multi-body Mobile Robots," in *The Fifth International Symposium on Experimental Robotics V*. London, UK: Springer-Verlag, 1998, pp. 301–309.

[9] K. Jiang, L. D. Senevirante, and S. E. Earles, "Time-optimal smooth-path motion planning for a mobile robot with kinematic constraints," *Robotica*, vol. 15, no. 5, pp. 547–553, 1997.

[10] 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.

[11] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, "Motion Planning for Humanoid Robots Under Obstacle and Dynamic Balance Constraints," in *IEEE International Conference on Robotics and Automation (ICRA2001)*, 2001.

[12] J. Lenarcic and B. Roth, *Advances in Robot Kinematics: Mechanisms and Motion*. Springer, 2006.

[13] F. Kanehiro, W. Suleiman, F. Lamiraux, E. Yoshida, and J.-P. Laumond, "Integrating Dynamics into Motion Planning for Humanoid Robots," in *IEEE/RSJ International Conference on Intelligent RObots and Systems (IROS)*, Nice, France, September 2008.

[14] F. Kanehiro, F. Lamiraux, O. Kanoun, E. Yoshida, and J.-P. Laumond, "A Local Collision Avoidance Method for Non-strictly Convex Objects," in *2008 Robotics: Science and Systems Conference*, Zurich, Switzerland, June 2008.

[15] R. Rockafellar, "Augmented Lagrange Multiplier Functions and Duality in Nonconvex Programming," *SIAM J.Control*, vol. 12, no. 2, May 1974.

[16] J. Lo and D. Metaxas, "Recursive Dynamics and Optimal Control Techniques for Human Motion Planning," *Computer Animation*, pp. 220–234, 1999.

[17] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1995.

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

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

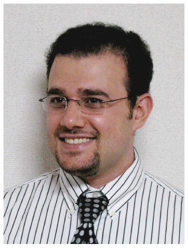
[20] G. Sohl and J. E. Bobrow, "A Recursive Multibody Dynamics and Sensitivity Algorithm for Branched Kinematics Chains," Department of Mechanical Engineering, University of California, Tech. Rep., 2000.

[21] W. Suleiman, E. Yoshida, J.-P. Laumond, and A. Monin, "On Humanoid Motion Optimization," in *IEEE-RAS 7th International Conference on Humanoid Robots*, Pittsburgh, Pennsylvania, USA, December 2007.

[22] W. Suleiman, E. Yoshida, F. Kanehiro, J.-P. Laumond, and A. Monin, "On Human Motion Imitation by Humanoid Robot," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2008.

[23] R. T. Rockafellar, "Lagrange multipliers in optimization," *SIAM-AMS Proceedings*, vol. 9, pp. 145–168, 1976.

[24] ——, "Lagrange multipliers and optimality," *SIAM Review*, vol. 35, no. 2, pp. 183–238, 1993.



Wael Suleiman received his Master and PhD degrees in Automatic Control from Paul Sabatier University in Toulouse, France, in 2004 and 2008 respectively. He is now postdoctoral fellowship of the Japan Society for the Promotion of Science (JSPS) at CNRS-AIST JRL, UMI3218/CRT, at National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. His research interests include nonlinear system identification and control, numerical optimization and humanoid robots.

Fumio Kanehiro was born in Toyama, Japan, in 1971. He received the B.E., M.E., and Ph.D. degrees in 1994, 1996, and 1999, respectively, all from the University of Tokyo, Tokyo, Japan. He was a Research Fellow of the Japan Society for the Promotion of Science (JSPS) from 1999 to 2000 and a visiting research of LAAS-CNRS, Toulouse, France in 2007. He is now a member of the Intelligent Systems Research Institute, National Institute of Advanced Industrial Science and Technology Tsukuba, Japan. His research interests include humanoid robots and developmental software systems.

Eiichi Yoshida received M.E and Ph.D degrees on Precision Machinery Engineering from Graduate School of Engineering, the University of Tokyo in 1993 and 1996 respectively. In 1996 he joined former Mechanical Engineering Laboratory, currently National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. From 1990 to 1991, he was visiting research associate at Swiss Federal Institute of Technology at Lausanne (EPFL). He served as Co-Director of AIST/IS-CNRS/ST2I Joint French-Japanese Robotics Laboratory (JRL) at LAAS-CNRS, Toulouse, France, from 2004 to 2008. He is currently Co-Director of CNRS-AIST JRL (Joint Robotics Laboratory), UMI3218/CRT, AIST, Japan, since 2009. His research interests include robot task and motion planning, modular robotic systems, and humanoid robots.

Jean Paul Laumond (IEEE Fellow) is Directeur de Recherche at LAAS-CNRS in Toulouse, France. With his group Gepetto (www.laas.fr/gepetto) he is exploring the computational foundations of anthropomorphic motion. He has been a Coordinator for two European Esprit projects, PROMotion (1992–1995) and MOLOG (1999–2002), both dedicated to robot motion planning technology. During 2001–2002, he created and managed Kineo CAM, a spin-off company from the LAAS-CNRS to develop and market motion planning technology. From 2005 to 2008 he was a co-director of JRL, a French-Japanese CNRS-AIST laboratory dedicated to humanoid robotics. He teaches robotics at the ENSTA and Ecole Normale Superieure in Paris. He published more than 150 papers in international journals and conferences in computer science, automatic control, robotics and neurosciences. He is a member of the IEEE RAS AdCom.

André Monin was born in Le Creusot, France in 1958. He graduated from the Ecole Nationale Supérieure d'Ingénieurs Electriciens de Grenoble in 1980. From 1981 to 1983, he was a teaching assistant in the Ecole Normale Supérieure de Marrakech, Morocco. Since 1985, he has been with the Laboratoire d'Automatique et d'Analyse des Systèmes of the Centre Nationale de la Recherche Scientifique (LAAS-CNRS), France, as "Chargé de Recherche" from 1989. After some work on non linear systems representation for his Doctoral thesis at Université Paul Sabatier, Toulouse, France, in 1987, his interests moved to the area of non linear filtering, systems realization and identification. He received the Habilitation from the University Paul Sabatier at Toulouse in 2003..