

Safe Motion Planning Computation for Databasing Balanced Movement of Humanoid Robots

Sebastien Lengagne, Nacim Ramdani, Philippe Fraisse

► **To cite this version:**

Sebastien Lengagne, Nacim Ramdani, Philippe Fraisse. Safe Motion Planning Computation for Databasing Balanced Movement of Humanoid Robots. ICRA: International Conference on Robotics and Automation, May 2009, Kobe, Japan. pp.1669-1674, 10.1109/ROBOT.2009.5152637 . lirmm-00385625

HAL Id: lirmm-00385625

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00385625>

Submitted on 19 May 2009

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.

Safe motion planning computation for databasing balanced movement of Humanoid Robots

Sébastien Lengagne^{1,2}, Nacim Ramdani^{1,3}, Philippe Fraisse^{1,4}

¹ LIRMM UMR 5506 CNRS Montpellier-France : www.lirmm.fr

² project team DEMAR, INRIA Sophia Antipolis Méditerranée : www-sop.inria.fr/demar

³ project team COPRIN, INRIA Sophia Antipolis Méditerranée : www-sop.inria.fr/coprin

⁴ University of Montpellier 2 : www.univ-montp2.fr

Mail : lengagne,ramdani,fraisse@lirmm.fr

Abstract—Motion databasing is an important topic in robotics research. Humanoid robots have a large number of degrees of freedom and their motions have to satisfy a set of constraints (balance, maximal joint torque velocity and angle values). Thus motion planning cannot efficiently be done on-line. The computation of optimal motions is performed off-line to create databases that transform the problem of large computation time into a problem of large memory space.

Motion planning can be seen as a Semi-Infinite Programming problem (SIP) since it involves a finite number of variables over an infinite set of constraints. Most methods solve the SIP problem by transforming it into a finite programming one using a discretization over a prescribed grid. We show that this approach is risky because it can lead to motions which may violate one or several constraints. Then we introduce our new method for planning safe motions. It uses Interval Analysis techniques in order to achieve a safe discretization of the constraints. We show how to implement this method and use it with state-of-the-art constrained optimization packages. Then, we illustrate its capabilities for planning safe motions dedicated to the HOAP-3 humanoid robot.

Index Terms—Motion planning, Semi Infinite Programming, Discretization, Interval Analysis, Constraints.

I. INTRODUCTION

Motion databasing is an important topic in robotics research. Humanoid robots have a large number of degrees of freedom and their motions have to satisfy a set of constraints (balance, maximal joint torque velocity and angle values). Thus, computing motions which are optimal, is time consuming and is often done off-line to create a database. Databased motions allow to realize global navigation: [1], [2], [3] start from a set of possible step motions and plan a sequence of steps to reach the goal thanks to algorithms such as Rapidly-exploring randomized Trees (RRT) algorithms [4]. Motion planning also includes the problem of digital actors' locomotion [5], kick motion generation on HRP-2 robot [6], computing a manipulator robot's trajectory [7] or smoothing pre-calculated motions [8]. The motions of the database are used as the joint angle reference trajectories. These motions are supposed to minimize a cost function and validate sets of equality and inequality continuous constraints. This problem can be seen as Semi-Infinite Programming problems (SIP), since it involves a finite number of variables which define

the motion, such as B-splines parameters, over an infinite continuous set of inequality constraints. To solve a SIP problem, the set of continuous inequality constraints are usually discretized by picking up several values over a given grid. Therefore, the obtained motions will satisfy the constraints only for the grid nodes. However, between two nodes the retained motion may violate some constraints which may have disastrous consequences on the integrity of the systems.

This paper presents a new method for planning safe motions, i.e. motions which ensure that the inequality constraints remain satisfied all over the motion duration. Our method uses the same optimization algorithms as classical one but replace the point discretization by a safe discretization that computes the constraints over time-intervals using Interval Analysis [9]. In this paper, we focus on the time-discretization, but this method can be extend to other variables discretization. Interval Analysis has already been used in order to solve, in a guaranteed way, the problems of self-collision avoidance and prevention for the arms of a 2-degrees of freedom robot [10] or to solve the problem of finding collision-free paths [11].

A preliminary study of our work was tested successfully on a two degrees of freedom pendulum where an optimal one-step motion was generated [12] and validated on 6 degrees of freedom model of the HOAP-3 Humanoid robot in the sagittal plane [13]. This paper addresses motion planning issues for a more complex 3-D system with twelve degrees of freedom.

We describe how to generate motions to be added to a database for 3D global navigation which ensures the balance and the integrity of the robot over the whole duration. In Section II we present the inverse dynamic model of a 3-D humanoid robot and how to characterize the balance. Section III presents the motion planning, considered as a Semi-infinite Programming (SIP) problem, and show how it is usual solved and how our new method uses Interval Analysis to ensure the balance and the integrity of the robot.

A. Inverse Dynamic Model

We model the humanoid robot as an arborescent chain with the contact foot as the reference body (Fig II-A).

Starting from the external forces F_{ext} and the joint position $q(t)$, velocity $\dot{q}(t)$ and acceleration $\ddot{q}(t)$ the Inverse Dynamic Model (IDM) computes the joint torques $\Gamma(t)$ and the forces applied by the reference body to the environment F_{ref}

$$\begin{bmatrix} \Gamma(t) \\ F_{ref}(t) \end{bmatrix} = IDM(q(t), \dot{q}(t), \ddot{q}(t), F_{ext}) \quad (1)$$

We use the Newton-Euler algorithm described in [14] which is composed of two recursions:

- The first recursion (Fig 1(a)) starts from the reference body and computes, through the waist, the position, velocity and acceleration of all the bodies in the Cartesian space. That allows to compute the forces due to the acceleration .
- the Second recursion starts from the extremity of the limbs to go to the reference body through the waist. It sends back the sum of the effects of the external forces and the forces due to the acceleration through the joints.

These equations can be formulated thanks to Lie Groups [15], [16] or using the notation of [14].

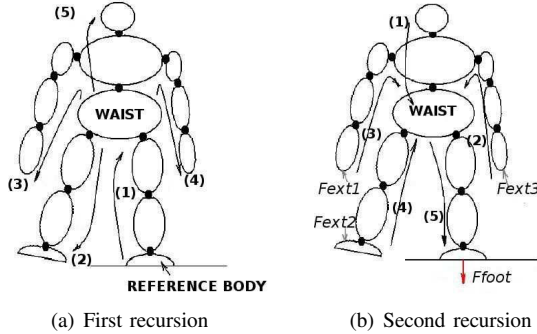


Fig. 1. Recursion for inverse dynamic model of a humanoid robot

B. Balance

The balance of humanoid robots can be defined thanks to the Zero Moment Point (ZMP). The ZMP is defined in [17] as a point, on the contact surface, where total inertia force is equal to 0. If this point stays within the base of support, the robot will keep its balance. The position of the ZMP depends on the force of the reference body.

$$\begin{bmatrix} ZMP_s(t) \\ ZMP_f(t) \end{bmatrix} = f(F_{ref}(t)). \quad (2)$$

$ZMP_s(t)$ and $ZMP_f(t)$ are the time history of the ZMP projected in the sagittal and frontal planes.

A. Semi-Infinite Programming (SIP) problem

The motion planning problem can be defined as a Semi-Infinite Programming (SIP) problem [18]. A SIP problem is an optimization problem with a finite number of variables to optimize and an infinite number of constraints to satisfy [19]. It consists in finding the parameter vector \mathbf{X} that:

$$\text{minimizes} \quad F(\mathbf{X}, t) \quad (3)$$

$$\text{subject to} \quad \forall i, \forall t \in [0, T] \quad g_i(\mathbf{X}, t) \leq 0 \quad (4)$$

$$\text{and} \quad \forall j, \forall \tau \in \{\tau_0, \dots, \tau_k\} \quad h_j(\mathbf{X}, \tau) = 0 \quad (5)$$

Where F denotes the cost (or objective) function, g_i the set of inequality constraint functions, h_j the set of equality constraint functions.

1) *Cost function*: The choice of the cost function $F(\mathbf{X})$ for motion planning must take into account the features of the robot and the desired application. Some authors minimize motion duration [20] or jerk [7] for robot manipulators. In [6], the energy consumption taking into account the parameters of the motors (friction, ...) is considered for humanoid robots. Biological inspired cost function can also be considered, for example the minimum torque change [21].

2) *Equality constraint functions*: The set of the equality constraint functions $h_j(\mathbf{X})$ allows to define the motion. These functions usually correspond to constraints on some system state variables at given time instants $\tau \in \{\tau_0, \dots, \tau_k\}$ such as the beginning or the end of a motion. For humanoid robots, we consider equality constraints as the position of the flying foot at the beginning and at the end of the motion.

3) *Inequality constraint functions*: The set of the inequality constraints $g_i(\mathbf{X})$ translates the physical limits of the system. Hence the integrity and the balance of the robot rely on the validity of these constraints.

These inequality constraints must be satisfied over the whole motion duration: $\forall t \in [0, T]$.

However, classical optimization algorithms, such as IPOPT [22] use a finite number of discrete constraints. Thus the inequality constraints must be discretized.

We present the classical way of discretizing the inequality constraints function in Section (III-B) and emphasize the fact that some constraints can be violated. In Section (III-C) we explain how our new method ensures the integrity and the balance of the robot thanks to Interval Analysis.

B. Solving SIP

In the context of SIP problems, discretization usually consists in picking up the functions value over several time points in a grid [23], [19]. This leads to replace the inequality constraints in Equation (5) by:

$$\forall i, \forall t_k \in \mathbf{T} \quad g_i(\mathbf{X}, t_k) \leq 0 \quad (6)$$

$$\text{where } \mathbf{T} = \{t_0 = 0, t_1, \dots, t_{N-1}, t_N = T\} \quad (7)$$

Consequently, the continuous problem (5) $\forall t \in [0, T]$ becomes a discrete one: $\forall t_k \in \mathbf{T}$ where the constraints are only considered for discrete values over the time-grid \mathbf{T} . There are several methods which run several optimization processes

and modify the grid \mathbf{T} in order to get better results [19]. Therefore, the optimal value depends on the number of time-point considered [24].

This way of discretization ensure the constraints satisfaction only for the instant on the time-grid [25]. Furthermore, no information is given as regarding the constraint satisfaction between two points of the time grid. Therefore the constraints can be violated during the motion. So we propose a new method for a safe constraint discretization using Interval Analysis.

C. Solving SIP via Interval Analysis

1) *Interval Analysis*: Interval analysis was initially developed to account for the quantification errors introduced by the floating point representation of real numbers with computers and was extended to validated numerics [26], [9], [27].

A real interval $[a] = [\underline{a}; \bar{a}]$ is a connected and closed subset of \mathbb{R} . With $\underline{a} = \text{Inf}([a])$, $\bar{a} = \text{Sup}([a])$ and $\text{Mid}([a]) = \frac{\underline{a} + \bar{a}}{2}$. The set of all real intervals of \mathbb{R} is denoted by \mathbb{IR} . Real arithmetic operations are extended to intervals. Consider an operator $\circ \in \{+, -, *, \div\}$ and $[a]$ and $[b]$ two intervals. Then:

$$[a] \circ [b] = [\text{Inf}_{u \in [a], v \in [b]} u \circ v, \text{Sup}_{u \in [a], v \in [b]} u \circ v] \quad (8)$$

Consider a function $\mathbf{m} : \mathbb{R}^n \mapsto \mathbb{R}^m$; the range of this function over an interval vector $[a]$ is given by:

$$\mathbf{m}([a]) = \{\mathbf{m}(\mathbf{u}) \mid \mathbf{u} \in [a]\} \quad (9)$$

The interval function $[\mathbf{m}] : \mathbb{IR}^n \mapsto \mathbb{IR}^m$ is an inclusion function for \mathbf{m} if

$$\forall [a] \in \mathbb{IR}^n, \mathbf{m}([a]) \subseteq [\mathbf{m}]([a]) \quad (10)$$

An inclusion function of \mathbf{m} can be obtained by replacing each occurrence of a real variable by the corresponding interval and each standard function by its interval counterpart. The resulting function is called the natural inclusion function. The performances of the inclusion function depend on the formal expression of \mathbf{m} .

SIP problems were solved with global optimization methods based on Interval Analysis [11], [28], [29]. The optimization process starts with a large interval for parameters value and reduces them until it finds intervals small enough which satisfy the SIP problem (5).

2) *Safe Discretization*: In this paper we present our method which uses Interval Analysis to ensure the inequality constraints validity over the whole motion duration [12] by computing the minimum and the maximum values for the set of functions $g_i(t)$ when t is defined over a given interval $[t]$. An *upper* bound for the maximum value $\max_{t \in [t]}(g_i(t))$ is given by $\text{Sup}[g_i]([t])$ and a *lower* bound for the minimum value $\min_{t \in [t]}(g_i(t))$ is given by $\text{Inf}[g_i]([t])$.

Therefore the upper bound of $g_i(t)$: $\max g_i$ are obtained in an easy and practical way by computing the upper bound of the inclusion function $[g_i]$ for a time interval $[t]$. Therefore, the inequality constraints in (5) are replaced by:

$$\forall i, \forall [t] \in \mathbf{IT} \quad \text{Sup}[g_i](\mathbf{X}, [t]) \leq 0 \quad (11)$$

In practice, the bounds thus derived may be too large because of the wrapping and dependence effects. Still, there are several techniques that can be used to obtain tighter enclosures by using for instance Taylor series expansion or some global optimization techniques [30].

Our method was tested with a 2-D model of the HOAP-3 humanoid robot in the sagittal plane [13]. In this paper we validate this new method for a more complex system such as the 3-D model of the HOAP-3 humanoid robot.

D. Gradient computation

Some optimization algorithms allow to decrease the computation time by using the gradient of the functions (for both cost and constraints function) $\frac{\partial g(\mathbf{X}, t)}{\partial \mathbf{X}}$ with respect to the parameter vector.

With a grid discretization (Cf Section III-B) this gradient is computed at the grid instant. and can be computed either via formal methods [8], [16] or automatic differentiation [31].

Our method presented in section III-C.2 for solving equation (11) subdivides a given time interval $[t]_k$ into N_b subintervals $[t]_k = \cup_{l=1, \dots, N_b} [t]_{k,l}$. Then there exists a subinterval $[t]_{k,l_{\max}}$ which contains the maximum of $g_i(\mathbf{X}, t)$ for $t \in [t]_k$ (Cf Figure 2)

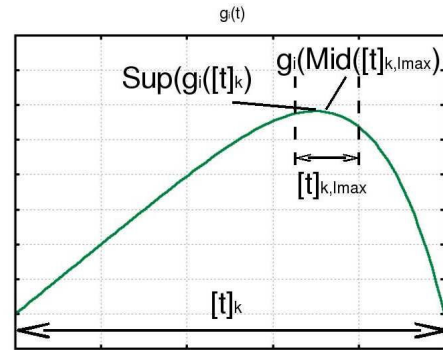


Fig. 2. Representation of the approximation of $\text{Sup}[g_i]([t]_k)$ by $g_i(\text{Mid}[t]_{k,l_{\max}})$ uses for the computation of the gradient

Since we do not know the instant when occurs the maximum within $[t]_{k,l_{\max}}$, we choose to approximate the gradient of the maximum of $g_i(\mathbf{X}, t)$ by the value of the gradient obtained at the middle of the subinterval $[t]_{k,l_{\max}}$. This is formulated in the following equation:

$$\frac{\partial}{\partial \mathbf{X}} \text{Sup}[g_i](\mathbf{X}, [t]_k) \approx \frac{\partial}{\partial \mathbf{X}} g_i(\mathbf{X}, \text{Mid}[t]_{k,l_{\max}}) \quad (12)$$

We can infer from this that the size of $[t]_{k,l_{\max}}$ impacts the accuracy of the computed gradient and hence the efficiency of the optimization algorithm used. The smaller is the interval $[t]_{k,l_{\max}}$ the better is the approximation of the gradient but the longer is the computation time.

E. Guaranteed Discretization Library

For an easy implementation of the planning motion we developed the Guaranteed Discretization Library (GDL) available on <http://www.lirmm.fr/~lengagne/GDL>.

GDL allows to compute the constraint functions for time instants or over time intervals and the gradient of these functions with respect to the parameters by automatic differentiation. All these abilities are based on a single model file in C++.

IV. RESULTS

A. Global navigation

To illustrate our method with a simple example, we propose to create a database that allows to walk straightaway with a fix step length l (HOAP-3 experimentations require $l = 7\text{cm}$). In a future study, we will address the case of the 3D navigation. We need to compute three motions:

- a start motion: the flying foot starts next to the stance foot to finish l cm ahead of the stance foot (position 1 to 2).
- a cycle motion: the flying foot starts l cm behind the stance foot to finish l cm ahead of the stance foot (position 3 to 2).
- an end motion: the flying foot starts l cm behind the stance foot to finish next to the stance foot (position 3 to 1).

The three positions are presented in Figure 4.

We use these three motions to make the robot walks (Cf. Figure 3)

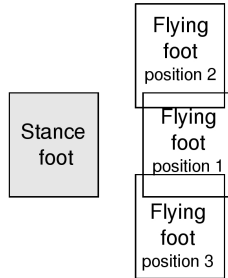


Fig. 4. Representation of the possible foot position.

B. Motions

We want to plan step motions for the HOAP-3 humanoid robot. We present only the single support phase motions which allows us to emphasize the constraint violation that appear with the usual way of discretization, especially the balance constraint function (ZMP).

We can obtain the Double support motions using the same method. In fact, the model have just to consider the force applied to the right foot proportional to the distance between the Center Of Mass (COM) projection and the foot [15].

We consider a model of the HOAP-3 humanoid taking into account only the legs assuming the upper part as a single body. This model contains 12 degrees of freedom. We assume a motion without any impact, by setting the initial and final velocity and acceleration equal to zero.

C. SIP problem

1) *Parameters*: We define a motion via the vector $\mathbf{X} = [T, \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_6]$ where T is the motion duration and \mathbf{p}_i the coefficients of the weighted sum of B-spline functions which model the i^{th} joint position trajectory $q_i(t)$, as follows:

$$q_i(t) = \sum_{j=0}^{N_s} p_{i,j} \times B_j(t) \quad (13)$$

The joint velocity and acceleration are obtained by differentiating (13).

For each degree of freedom we use 5 coefficients. The vector \mathbf{X} is composed of $5 \times 12 + 1 = 61$ parameters.

2) *Cost function*: In this paper we choose to minimize the motion duration:

$$F(\mathbf{X}) = \int_0^T 1 dt = T \quad (14)$$

We want to get fast motion to emphasize the constraints violation for SIP problem solved with usual discretization method [13].

3) *Equality constraints*: The equality constraints are used to define the position and orientation of the flying foot at the beginning and at the end of the motion. Therefore we consider $2 \times 6 = 12$ equality constraints.

4) *Inequality constraints*: In these experiments we consider limitations on the joint position, velocity and torque values and also on the ZMP location in order to ensure the robot balance. The set of inequality constraint functions is as follows:

$$\left\{ \begin{array}{l} \underline{q} \leq q(t) \leq \bar{q} \\ \underline{\dot{q}} \leq \dot{q}(t) \leq \bar{\dot{q}} \\ \underline{\Gamma} \leq \Gamma(t) \leq \bar{\Gamma} \\ \underline{ZMP}_s \leq ZMP_s(t) \leq \bar{ZMP}_s \\ \underline{ZMP}_f \leq ZMP_f(t) \leq \bar{ZMP}_f \end{array} \right. \quad (15)$$

Each of the constraints is decomposed as follow:

$$\underline{y} \leq y(t) \leq \bar{y} \equiv \left\{ \begin{array}{l} y - y(t) \leq 0 \\ -\bar{y} + y(t) \leq 0 \end{array} \right. \quad (16)$$

Therefore we have to deal with $(12 + 12 + 12 + 2) \times 2 = 76$ continuous constraint functions.

D. SIP solved via usual techniques

We use a time grid of 25 time instants and solve the optimization problem (5) thanks to the IPOPT algorithm [22]. We obtain the Figures 5, 6 and 7 which present the evolution of the ZMP function for the optimal motion obtained via usual discretization method. The crosses represent the selected value sent to the algorithm.

On Figures 5 and 6, these values do not correspond to the extrema of the function. Therefore the algorithm do not deal with the real extrema of the constraint functions. Therefore the optimal motion retained can violate the constraints.

This is highlighted in Figure 7 where the considered point are within the contact surface whereas the continuous function can be external to the contact surface. This will lead to the fall of the robot.

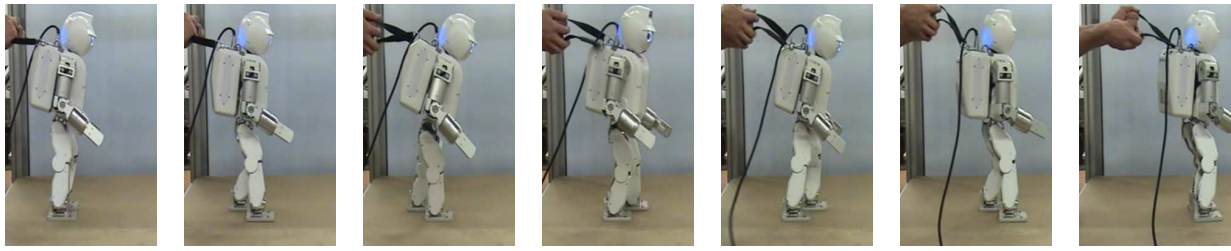


Fig. 3. Global navigation of the HOAP-3 Humanoid robot.

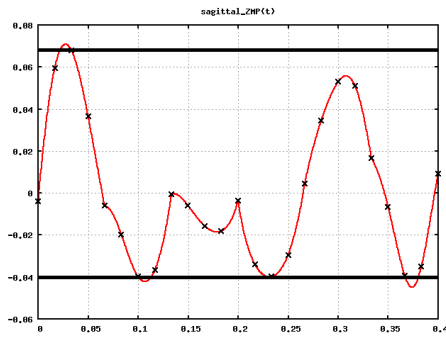


Fig. 5. Time history of the ZMP in the sagittal plane.

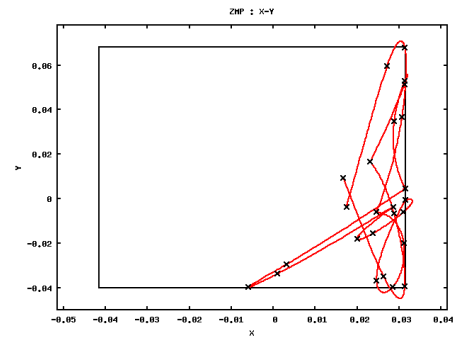


Fig. 7. Representation of the evolution of the ZMP on the contact surface.

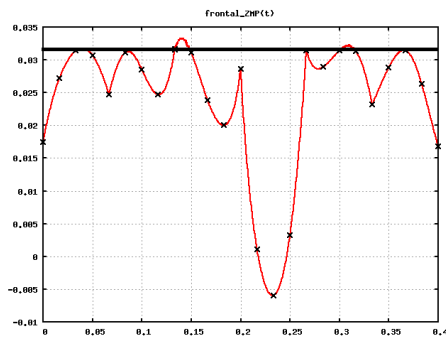


Fig. 6. Time history of the ZMP in the frontal plane.

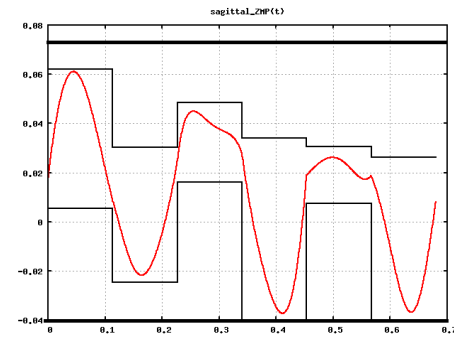


Fig. 8. Time history of the ZMP in the sagittal plane.

E. SIP solved via Interval Analysis

We compute the constraint functions over 6 time intervals and solve the optimization problem (5) thanks to the IPOPT algorithm [22]. We obtain the Figures 8, 9 and 10 which present the evolution of the ZMP function for the optimal motion obtained via safe discretization method.

The enclosures are the values returned to the algorithm. On Figures 8 and 9, the enclosures give a conservative evaluation of the extrema of the function. Therefore the algorithm is aware of the extrema of the constraint functions. Thus the produced solution satisfies the constraints.

This is highlighted in Figure 10 where the ZMP enclosures and the continuous function stays within the contact surface. This motion will not break nor make the robot fall.

V. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

The creation of motion databases allows to perform on-line optimal motions that are computed off-line. These motions have to minimize a cost function and validate a set of equality and continuous inequality constraints.

Motion planning is usually seen as Semi-Infinite Programming problems (SIP) and is solved by transforming it into a finite programming problem thanks to a time-grid discretization. Unfortunately time-grid discretization can lead to some constraint violations which may impact the integrity and the balance of the robot. To the contrary, our new method for safe motion planning uses Interval Analysis to compute the maximum for constraint functions over time-intervals, thus avoids any constraint violation.

We validated our method with a 12-dof model of the HOAP-3 Humanoid robot in 3D, and show that safe discretization ensures the integrity and the balance of the robot.

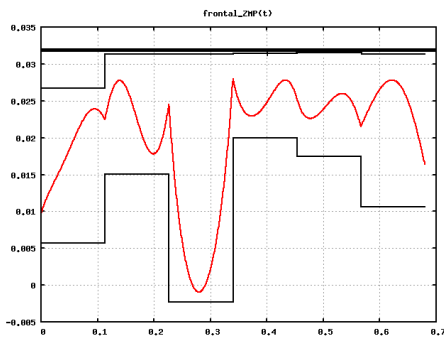


Fig. 9. Time history of the ZMP in the frontal plane.

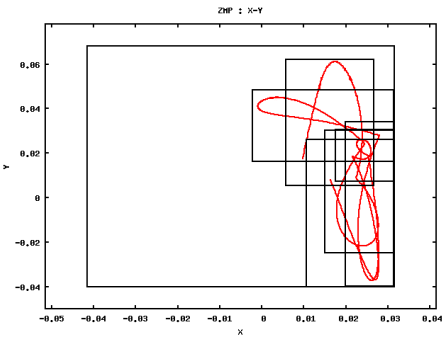


Fig. 10. Representation of the evolution of the ZMP on the contact surface.

B. Future Works

Here we addressed a one-dimension time discretization issue but the same approach can be used for other systems which need a N-dimensions space discretization. We presented the motion planning for each motion of the database. So we have optimal motions but we do not ensure an optimal global navigation. The next step of our work will be to use safe discretization to plan whole motions for the database regarding that the link between two motions is also an optimal motion.

REFERENCES

- [1] J. J. J. Kuffner, K. Nishiwaki., S. Kagami., M. Inaba, and H. Inoue, "Footstep planning among obstacles for biped robots," in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 1, Maui, HI, USA, 2001, pp. 500–505.
- [2] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, "Motion planning for humanoid robots." in *20th Int'l Symp. Robotics Research (ISR'03)*, 2003.
- [3] Z. Xia, Y. He, and K. Chen, "Modeling and motion planning of the infant-size humanoid robot thbip-ii," in *IEEE-RAS 7th International Conference on Humanoid Robots, 2007*.
- [4] S. M. LaValle and J. J. Kuffner, "Rapidly-exploring random trees : Progress and prospects," in *4th Int'l Workshop on the Algorithmic Foundations of Robotics*, 2000.
- [5] E. Yoshida, I. Belousov, C. Esteves, and J.-P. Laumond, "Humanoid motion planning for dynamic tasks," in *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, Dec. 2005, pp. 1–6.
- [6] S. Miossec, K. Yokoi, and A. Kheddar, "Development of a software for motion optimization of robots - application to the kick motion of the hrp-2 robot," in *Proceedings of the 2006 IEEE International Conference on Robotics and Biomimetics*, 2006, pp. 299–304.
- [7] A. Piazzzi and A. Visioli, "Global minimum-jerk trajectory planning of robot manipulators," in *IEEE Transactions on Industrial Electronics*, vol. 47, febr 2000, pp. 140–149.
- [8] W. Suleiman, E. Yoshida, J.-P. Laumond, and A. Monink, "On humanoid motion optimization," in *IEEE-RAS 7th International Conference on Humanoid Robots, 2007*.
- [9] R. E. Moore and F. Bierbaum, *Methods and Applications of Interval Analysis (SIAM Studies in Applied and Numerical Mathematics) (SIAM Studies in Applied Mathematics, 2.)*. Soc for Industrial & Applied Math, 1979.
- [10] H. Fang and J. P. Merlet, "Dynamic interference avoidance of 2-DOF robot arms using interval analysis," in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, Aug. 2005, pp. 3809–3814.
- [11] L. Jaulin, "path planning using intervals and graphs," *Reliable Computing*, vol. 7, no. 1, pp. 1–15, fevrier 2001.
- [12] S. Lengagne, N. Ramdani, and P. Fraisse, "Guaranteed computation of constraints for safe path planning," in *IEEE-RAS 7th International Conference on Humanoid Robots, 2007*.
- [13] —, "A new method for generating safe motions for humanoid robots," in *submitted to IEEE-RAS International Conference on Humanoid Robots, 2008*.
- [14] W. Khalil and E. Dombre, *Modeling, Identification & Control of Robots*, 3rd ed., E. B. Heinemann, Ed. Hermes Sciences Europe, march 2002.
- [15] W. Suleiman, E. Yoshida., J.-P. Laumond., and A. A. Monin, "Optimizing humanoid motions using recursive dynamics and lie groups," in *Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. 3rd International Conference on*, Damascus., Apr. 2008, pp. 1–6.
- [16] S.-H. Lee, J. Kim, F. Park, M. Kim, and J. E. Bobrow, "Newton-type algorithms for dynamics-based robot movement optimization," in *IEEE Transactions on robotics*, vol. 21, 2005, pp. 657– 667.
- [17] M. Vukobratovic and D. Juricic, "Contribution to the synthesis of biped gait," *IEEE trans. Bio-Med Eng.*, vol. BME-16, pp. 1–6, 1969.
- [18] A. I. F. Vaz, E. M. Fernandes, and M. P. S. Gomes, "Robot trajectory planning with semi-infinite programming," *European Journal of Operational Research.*, vol. 153, no. 3, pp. 607–617, 2004.
- [19] R. Hettich and K. O. Kortanek, "Semi-infinite programming: theory, methods, and applications," *SIAM Rev.*, vol. 35, no. 3, pp. 380–429, 1993.
- [20] A. Piazzzi and A. Visioli, "Global minimum-time trajectory planning of mechanical manipulators using interval analysis," *International Journal of Control*, vol. 71, pp. 631–652(22), 10 November 1998.
- [21] Y. Uno, M. Kawato, and R. Suzuki, "Formation and control of optimal trajectory in human multijoint arm movement," *Biological Cybernetics*, vol. 6, no. 2, pp. 89–101, juin 1989.
- [22] *Introduction to IPOPT : a tutorial for downloading, installing and using IPOPT*, april 7th 2006.
- [23] O. von Stryk, "Numerical solution of optimal control problems by direct collocation," 1993.
- [24] R. Reemtsen, "Semi-infinite programming: discretization methods," 1998.
- [25] O. von Stryk and R. Bulirsch, "Direct and indirect methods for trajectory optimization," *Ann. Oper. Res.*, vol. 37, no. 1-4, pp. 357–373, 1992.
- [26] T. Sunaga, "Theory of interval algebra and its application to numerical analysis," *RAAG Memoirs, Ggujutsu Bunken Fukuy-kai*, vol. 2, pp. 547–564, 1958.
- [27] A. Neumaier, *Interval methods for systems of equations*. Cambridge: Cambridge university press, 1990.
- [28] L. Jaulin and Éric Walter, "Guaranteed tuning, with application to robust control and motion planning," *Automatica*, vol. 32, no. 9, pp. 1217–1221, 1996.
- [29] L. Kolev and D. Penev, "An interval method for global inequality constraint optimization problems," in *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*, vol. 4, MAY 2000, pp. 617–620.
- [30] E. Hansen and G. Walster, *Global optimization using interval analysis*, 2nd ed. Marcel Dekker, 2004.
- [31] C. Bendtsen and O. Stauning, *FADBAD, a flexible c++ package for automatic differentiation*.