Planning and Fast Re-Planning of Safe Motions for Humanoid Robots : Application to a Kicking Motion

Sébastien Lengagne, Nacim Ramdani and Philippe Fraisse

Abstract— **Optimal motions are usually used as joint reference trajectories for repetitive or complex motions. In the case of soccer robots, the kicking motion is usually a benchmark motion, computed off-line, without taking into account the current position of the robot or the direction of the goal. Moreover, robots must react quickly to any situation, even if not expected, and cannot spend time to generate a new optimal motion by the classical way. Therefore, we propose a new method for fast motion re-planning based on an off-line computation of a feasible sub-set of the motion parameters, using Interval Analysis.**

INTRODUCTION

Planning optimal motions for humanoid robots is dedicated to complex or repetitive tasks. Humanoid robots are complex systems in which neither geometric structure nor dynamic model are simple. This complexity limits the reactive capabilities of computation for motion planning. Indeed, a set of constraints such as balance, maximal joint torque velocity or position has to be included into the optimization process.

These constraints can be non linear and require a large computation time. Consequently, optimal motions are often generated off-line and used as joint reference trajectories. Motion planning includes as well, the problem of digital actors' locomotion [1], kick motion generation on HRP-2 robot [2], computing a manipulator robot's trajectory [3] or smoothing pre-calculated motions [4]. In previous works, we presented a new method for planning safe motions [5], [6], [7] which uses the Interval Analysis to compute the constraint functions over time-interval, whereas classical methods which compute them over a time grid without any information about the constraint validity between two points of the grid.

Nowadays, motion planning aims to be fast enough to react to unexpected events. For instance, homotopic paths allow to modify a generated trajectory, to address collision avoidance for mobile arm manipulator [8].

In the case of humanoid robots, the computation of the inequality constraint functions is too time-consuming, and motion planning can last from a few minutes to several hours. However, all the motions planned, or re-planned must satisfy a set of inequality constraint functions (balance, maximum torque or angle joint).

The re-planning process starts from an optimal motion parameter set, and computes a feasible sub-set that is an inner approximation of the feasible set of the inequality constraints. Then, it has to find a solution, in this sub-set, which allows to overcome unpredicted situations. By

the way, the re-planning process consider the inequality constraints without computing them online.

In the case of soccer robots [9], the most important motion is the kicking motion, since it allows to goal. Usually, kicking motion is computed off line [10], hence does not take into account the current position of the robot or the direction of the goal. Nevertheless these motions allow the robot to react quickly to the situation, even if the kicking can lead to an accurate trajectory of the ball. In this paper, we propose a method to make the kicking motion more accurate by an off-line planning and a fast re-planning process.

In Section I, we briefly show how to generate optimal motions. Then, Section II presents our algorithm for computing a feasible box around the optimal motion. We apply this method to a kicking motion for the HOAP-3 Humanoid robot in Section III

I. SAFE MOTION PLANNING

A. Modeling

1) B-splines parameterization: Motion planning problem is an infinite programming problem which can be transformed into a Semi-Infinite Problem (SIP) by a joint trajectory parameterization [6].

We choose to compute the joint trajectories thanks to B-splines functions [11]. Thus, we define a motion via the parameter vector $\mathbf{X} = [T, \mathbf{p}_{1,1}, \mathbf{p}_{1,2}, \ldots, \mathbf{p}_{6,5}]$ where T is the motion duration and $\mathbf{p}_i$ the coefficients of the weighted sum of B-splines functions. The joint trajectory $q_i(t)$ is computed as follows:

$$q_i(t) = \sum_{j=0}^{N_s} p_{i,j} \times B_j(t) \qquad (1)$$
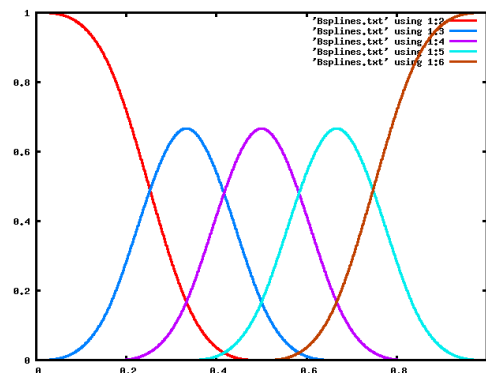

Fig. 1.   Time History of the Bsplines functions

The joint velocity and acceleration are obtained by differentiating the equation(1). In this paper, we consider 5 B-splines functions represented in Figure 1. Those functions allow to get motion with initial and final velocity and acceleration equal to zero.

2) Inverse Dynamic Model: We model the humanoid robot as an arborescent chain with n degrees of freedom. We assume that the right foot is the reference body, and add some constraint during the motion planning to make sure that is motionless. We use HuManS software [12] to generate the models in the form of C-files functions. Thanks to the definition of Denavit-Hartenberg parameters [13], center of mass and inertia values, this software generates C-functions that compute the inertial effect matrix $M(q)$ and the Non-linear effects vector due to gravity and Coriolis $H(q,\dot{q})$.

Starting from the vectors $\{q(t) \in \mathbb{R}^{n+6}, \dot{q}(t) \in \mathbb{R}^{n+6}, \ddot{q}(t) \in \mathbb{R}^{n+6}\}$ which contain the angle value of the n joints and the position of the reference body, we compute the joint torques, $\Gamma(t) \in \mathbb{R}^n$, and the effort applied by the reference body (the right foot) to the ground $\mathbf{F}(t) \in \mathbb{R}^6$,

$$\begin{bmatrix} \Gamma(t) \\ \mathbf{F}(t) \end{bmatrix} = M(q(t))\ddot{q}(t) + H(q(t),\dot{q}(t)) \qquad (2)$$

The HOAP-3 Humanoid Robot, represented in Figure 2, has 21 degrees of freedom. We assume that the upper parts will not move during motion. Thus we compute the joint trajectories only for the lower limbs. We have to plan the trajectories for the 12 joints of the legs.
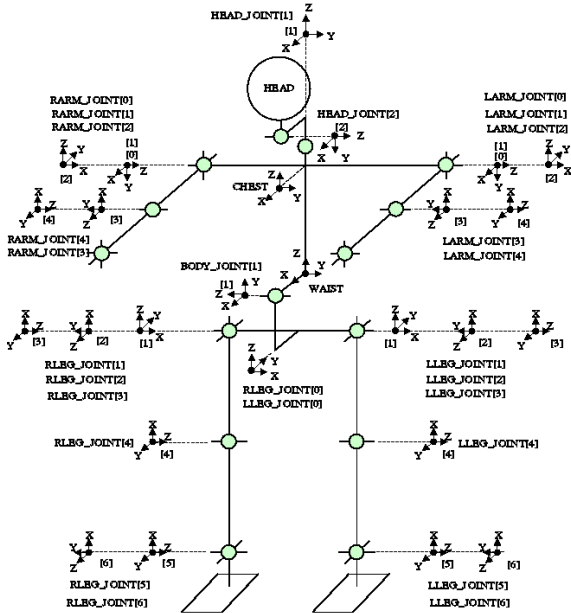

Fig. 2. Denavit-Hartenberg parameters definition coordinate of the HOAP-3 Robot

3) Balance: The balance of humanoid robots can be defined thanks to the Zero Moment Point (ZMP). The ZMP is defined in [14] as a point, on the contact surface, where the total inertia force is equal to 0. If this point stays within the base of support, the robot keeps its balance. The position of the ZMP depends on the ground reaction effort.

$$\begin{bmatrix} ZMP_s(t) \\ ZMP_f(t) \end{bmatrix} = f(\mathbf{F}(t)). \qquad (3)$$

$ZMP_s(t)$ and $ZMP_f(t)$ are the time history of the ZMP projected in the sagital and frontal planes and rely on the ground reaction effort, and hence on the joint trajectories $\{q(t),\dot{q}(t),\ddot{q}(t)\}$.

B. Motion planning as SIP

The motion planning problem can be defined as a Semi-Infinite Programming (SIP) problem [15]. A SIP problem is an optimization problem with a finite number of variables and an infinite number of constraints [16]. It consists in finding the parameter vector $\tilde{X}$ that:

$$\begin{array}{lll} \text{minimizes} & J(\tilde{X},t) & \\ \text{subject to} & \forall j, \forall t \in [0,T] & g_j(\tilde{X},t) \le 0 \\ \text{and} & \forall k & h_k(\tilde{X}) = 0 \end{array} \qquad (4)$$

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 $J(\mathbf{X},t)$ for motion planning must take into account the features of the robot and the desired application. For robot manipulators, some authors minimize motion duration [17], or jerk [3]. For humanoid robots, the energy consumption, taking into account the parameters of the motors (friction, ...) [2], or biological inspired, such as the minimum torque change [18], cost function can be considered. To improve the autonomy of the robots, we choose the cost function as the sum of the torque square:

$$F = \int \sum_{i=1}^{n} \Gamma^2(t).dt \qquad (5)$$

2) Equality constraint functions: The set of the equality constraint functions $h_j(\mathbf{X})$ allows to define the motion. These functions are usually used to impose to constraints on some system state variables at given time instants such as the beginning or the end of a motion. For humanoid robot, we consider equality constraints as the position of the flying foot at the beginning, at the end and at mid-duration of the motion:

$$\forall t_k \in \{0; \tfrac{T}{2}; T\} \quad h(\mathbf{X}, t_k) = 0 \qquad (6)$$

3) Inequality constraint functions: The set of the inequality constraints $g_i(\mathbf{X})$ translates the physical limits of the system. For kicking motion we consider limits about joint position, velocity and torque and about sagital and frontal balance, as presented in Equation (7).

$$\begin{array}{ll} \forall t \in [0,T] & \underline{q_k} \le q_k(t) \le \overline{q_k} \\ \forall k \in \{1,2,\ldots,n\} & \underline{\dot{q}_k} \le \dot{q}_k(t) \le \overline{\dot{q}_k} \\ & \underline{\Gamma_k} \le \Gamma_k(t) \le \overline{\Gamma_k} \\ \forall t \in [0,T] & \underline{ZMP_s} \le \dot{ZMP}_s(t) \le \overline{ZMP_s} \\ & \underline{ZMP_f} \le \dot{ZMP}_f(t) \le \overline{ZMP_f} \end{array} \qquad (7)$$

It is possible to take into account other constraint functions to avoid sliding or self-collision [19], for instance.

C. Time-Interval Discretization

In [7], we presented a new way for dealing with the inequality constraints : the time-interval discretization, which ensures constraint validity over whole motion duration and allows to use state-of-the-art algorithm such as IPOPT [20]. This method uses Interval Analysis to ensure the inequality constraints validity over whole motion duration, by computing minimum and maximum values for the set of functions $g_i(t)$ when t is defined over a given interval $[t]$.

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 $[t]$ [21]. The inequality constraints in (4) are replaced by:

$$\forall i, \forall [t] \in \mathbf{IT} \quad Sup[g]_i(\mathbf{X}, [t]) \leq 0$$
$$\text{with} \qquad \mathbf{IT} = \{[0, t_1], [t_1, t_2] \ldots [t_k, T]\} \tag{8}$$

In practice, the bounds thus derived may be too large because of over approximation in interval computing process. 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 [22].

D. Optimal kicking motion

We planned an optimal motion to make the robot kick a ball at the mid-duration of its motion at $0.6m/s$ speed. We impose that the collision will be at the position $x = 1cm$, $h = 3cm$ as shown in Figure 3.
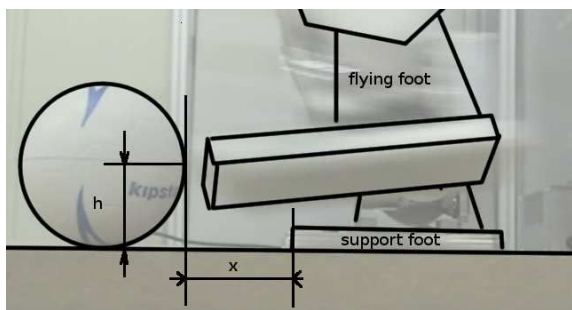
Fig. 3.   Scheme of the collision between the ball and foot of the robot

The dimension of the parameter vector, which characterizes the motion, X is 61 (5×12 B-splines parameters plus motion duration T), and we consider the constraint functions such as defined in Sections I-B.2, I-B.3. When the optimization process was succeeded, we get the motion presented in Figure 4 with the time history of the Zero Moment Point in the frontal and sagital plane in Figures 5, 6. The computation time is about two hours.

Figures 4(b) show the collision between the foot and the ball. The foot hits the ball at $3cm$-high as required by the optimization process.
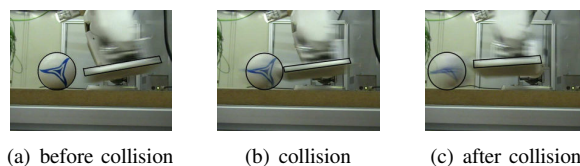
(a) before collision	(b) collision	(c) after collision

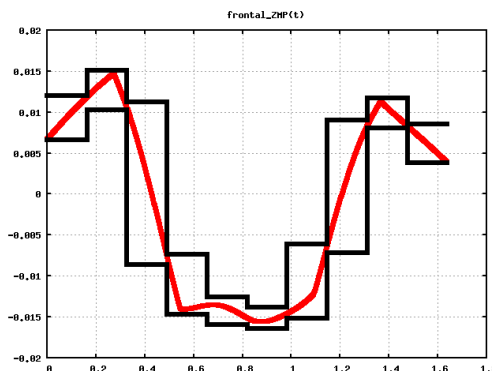Fig. 4.   Optimal kicking motion with a ball far from $1cm$

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

II. FAST MOTION RE-PLANNING

A. The Problem Under Study

In the previous section, we generated a kicking motion while assuming the location of the ball at $x = 1cm$. What happens if the ball is not at the expected position ? Figure 7 shows the result obtained when this optimal motion is used with the ball at $x = 3cm$. The foot hits the ball higher than expected. Thus the energy transmitted to the ball may be insufficient to reach the desired goal.

To improve the kicking motion, one solution could be to generate a new optimal motion with this new equality constraint. However, this solution is too time-consuming (about two hours for the previous one). In this sequel, we introduce a method which modifies the previous optimal kicking motion, in a very small CPU time while ensuring constraint satisfaction.

Our idea consists in replacing the set of inequality constraint $\forall t \in [0, T], \ g(X, t) < 0$ by a set of bounds on the
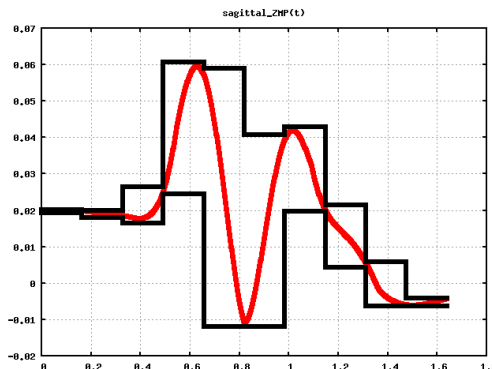
Fig. 6.   Time history of the ZMP in the frontal plane

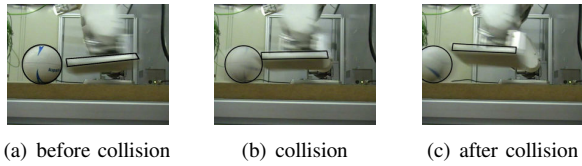(a) before collision   (b) collision   (c) after collision

Fig. 7.   Optimal kicking motion with a ball far from 3*cm*

parameter $X \in [X]$, when $[X]$ is the feasible set of parameters. This allows not to compute the inequality constraints which can be nonlinear and time-consuming, whereas constraints on the parameter are linear and fast to compute.

By this way, online adaptation consists in an optimization process with only bounds on the parameters, the new equality constraints h'_k and possibly the cost function J'.

$$
\begin{array}{ll}
\text{minimizes} & J'(\hat{X},t) \\
\text{subject to} & \hat{X} \in [X] \\
\text{and} & \forall k \quad h'_k(\hat{X}) = 0
\end{array} \quad (9)
$$

B. Computation of the Feasible Sub-set

To make the robot able to adapt its motion to as many situations as possible we have to compute a feasible sub-set $[X]$ that contains the optimal vector $\tilde{X}$ and satisfy all the inequality constraint functions. Recent studies addressed the computation of feasible sets, using Interval Analysis, for the design of parallel or serial robots [23], [24]. In fact, we do not compute whole feasible set, but only an inner approximation of it. The sub-set $[X]$ will be contained in the feasible set. We define a box as large as possible, then we solve the following problem:

$$
\begin{array}{ll}
\text{maximize} & \delta \in \mathbb{R}^+ \\
\text{such as} & \forall i \; [X_i] = \tilde{X}_i + \delta \times [\mathbb{W}_i] \\
\text{with} & \forall i \; 0 \in [\mathbb{W}_i] \\
\forall j, \; \forall X \in [X], \; \forall t \in [0,T] & g_j(X,t) < 0
\end{array} \quad (10)
$$

Where δ is the normalized width of the box and $[\mathbb{W}]$ a weighted interval vector that allows to ignore or give priority to some components of the box $[X]$. In this case, we propose that $[\mathbb{W}]$ is computing by using the distance between the optimal vector $\tilde{X}$ and the first constraint violation along each direction.

As presented in Section I-C, we propose to use the time-interval discretization which ensures constraint validity over whole motion duration. The inequality constraint in Equation (10) is replaced by:

$$
\begin{array}{ll}
\forall i, \forall [t] \in \mathbf{IT} & Sup[g]_i(\mathbf{X},[t]) \leq 0 \\
\text{with} & \mathbf{IT} = \{[0,t_1],[t_1,t_2]\ldots[t_k,T]\}
\end{array} \quad (11)
$$

C. Algorithm

The principle of the algorithm is to start from a large value of δ, and to reduce it by rejecting all the solutions in the corresponding box which violate a constraint.

Figure 8 shows the principle of this algorithm computing the feasible sub-set $[X]$. Using ALIAS software [25], a
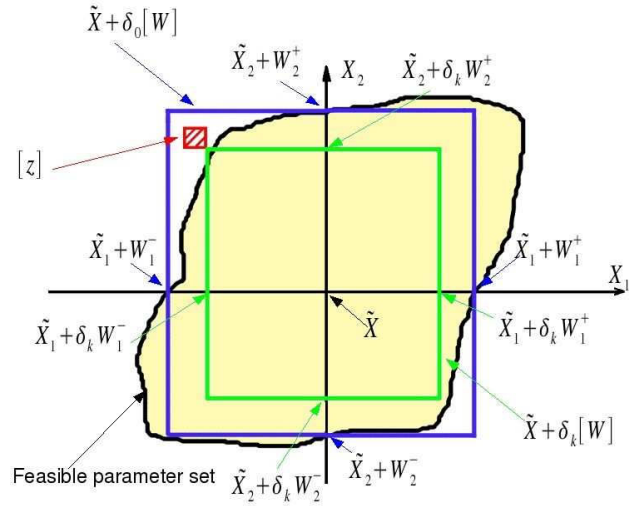

Fig. 8.   Example of a feasible set and of its inner approximation : the feasible sub-set $[X]$

branching algorithm with consistence tests, we search a box $[z]$ that satisfies:

$$
\begin{array}{ll}
\text{with} & [X] = \tilde{X} + \delta_k[\mathbb{W}] \\
\text{find} & [z] \subset [X] \\
\text{such as } \exists j, \exists t \in [t] & Sup[g]_j([z],t) > 0
\end{array} \quad (12)
$$

Once the software finds a solution, $[z]$, it stops and δ is chosen such that:

$$
[z] \; \cap \; \tilde{X} + \delta_{k+1}[\mathbb{W}] = \emptyset \quad (13)
$$

The algorithm stops when there is no solution to the problem (12). Eventually, computed $\tilde{X} + \delta_{final}[\mathbb{W}]$ is the largest box contained in the feasible set.

III. KICKING-MOTION ADAPTATION

A. Choice of the parameter to adapt

Obviously, it is not necessary to adapt all the motion parameters. Since, we are interesting in the collision location along the x-axis, we propose to adapt the trajectories of the knee, hip pitch and ankle pitch (named LEG_JOINT[3,4,5] in Figure 2) which influence the motion in the sagital plane. The collision occurs at the half of the motion, thus we will only change the third B-splines parameters.

B. Feasible Sub-Set

Table I presents the result of our computation. Where the optimal value is $\tilde{X}_i$, the lower bound of the feasible set $\underline{X}_i$ and the upper bound of the feasible set $\overline{X}_i$. The width of the feasible sub-set depends on the parameter, but it is interesting to see that some parameters can be changed within an interval of 5 degrees without making the robot falls (since it ensures no constraint violation).

Figures 9 shows the time history of a joint value and the ZMP in the sagital plane for the optimal motion and their limits for all the motions contained in the feasible sub-set. In Figure 9(b), the ZMP in the sagital plane is presented,

position	$\check{X}_i$	$\underline{X}_i$	$\overline{X}_i$
Right Hip Pitch	-22.60	-26.58	-20.34
Right Knee	39.68	38.20	41.21
Right Ankle Pitch	-17.33	-18.88	-16.05
Left Hip Pitch	-17.12	-19.13	-13.97
Left Knee	19.56	13.97	22.90
Left Ankle Pitch	4.51	-3.66	5.90

TABLE I

TABLE OF THE VALUE FOR THE OPTIMAL MOTION, FOR THE FEASIBLE
SUB-SET (ANGLE ARE GIVEN IN DEGREE)

all the motions of the feasible sub-set will make the robot keeping its balance, since their limits stay within the size of the foot ($-0.04 < ZMP_s(t) < 0.068$). Thus, we can choose any motion of the feasible sub-set without computing the inequality constraint functions to make sure that there is no constraint violation. All the motions of the feasible sub-set are safe.

C. Re-planned motion

| position	$\check{X}_i$	$\hat{X}_i$
Right Hip Pitch	-22.60	-21.90
Right Knee	39.68	40.17
Right Ankle Pitch	-17.33	-18.82
Left Hip Pitch	-17.12	-18.12
Left Knee	19.56	19.21
Left Ankle Pitch	4.51	5.34

TABLE II

TABLE OF THE VALUE FOR THE OPTIMAL MOTION AND FOR THE
RE-PLANNED MOTION (ANGLE ARE GIVEN IN DEGREE)

Figure 11 shows the feasible set of the couple (x, h) for all the motions in the feasible sub-set $[X]$. Unfortunately, it appears that we cannot achieve a kicking motion for the collision location ($x = 3cm, h = 3cm$). If we want the robot kicks the ball at $3cm$ high, the ball must be located between $-1cm$ and $1.7cm$. We choose to re-plan the optimal motion to make the robot kicks a ball at $3cm$ high and located at the position $x = 1.5cm$. Thus, we proceed to the optimization of the problem presented in Equation(9) with these equality constraints:
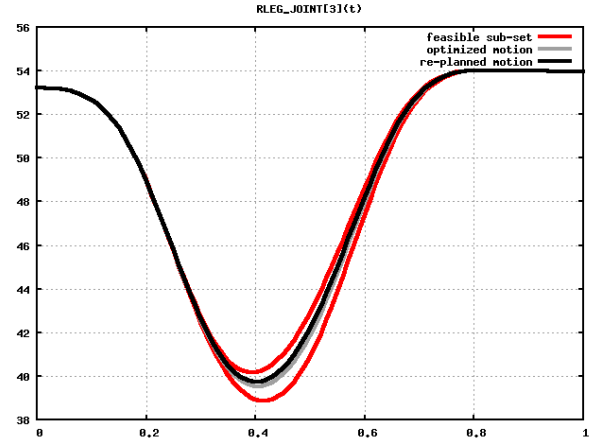
$$
\begin{aligned}
\text{find} \quad & \hat{X} \in [X] \\
\text{such as} \quad & h(\tfrac{T}{2}) = 3cm \\
& x(\tfrac{T}{2}) = 1.5cm
\end{aligned}
\tag{14}
$$

The optimization software spent less than one second of CPU time to find a solution. This solution is presented in Table II and the re-planned motion in Figure 12.
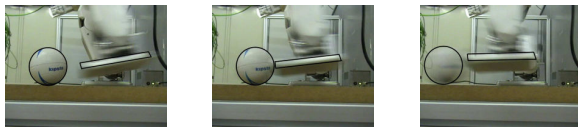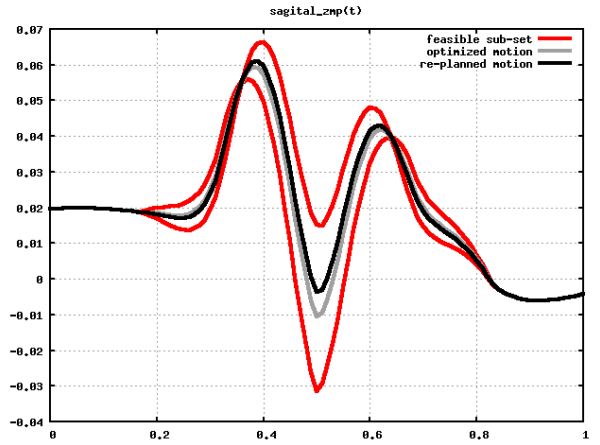

(a) before collision   (b) collision   (c) after collision

Fig. 12.   Re-planned kicking motion

(a) Right Hip Pitch angle value

(b) ZMP in the sagital plane : zmp_s(t)

Fig. 9.   Time History of the Right Hip Pitch angle and of the ZMP in the sagital plane for the optimal motion, for the re-planned motion, and their limits for all the solutions in the feasible sub-set

CONCLUSION

In this paper we presented the planning and fast re-planning of safe motions. We applied our method to a kicking motion for a humanoid robot. The safe motion planning consists in solving a Semi Infinite Programming problem, using a time-interval discretization. Unfortunately, this method requires a large CPU time (2 hours), and hence cannot be done online. We generate a safe kicking motion which makes the HOAP-3 Humanoid Robot kick a ball located at $1cm$ from its foot. We showed that this motion is not good enough, if the ball is farther than expected. As a result we propose a safe re-planning method, which starting from the optimal motion, computes off-line a feasible sub-set of the motion parameters. By the way, we can achieve a fast re-planning which consists in finding, in this feasible sub-set, a solution that will validate a new set of equality constraint. In the future we will test this method for other motions, for example to adapt optimal step motions to a new position or to slopes.
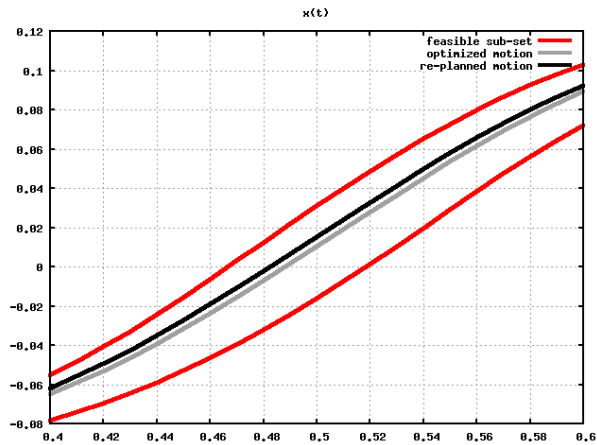
Fig. 10. Time history for the x-position of the flying foot for the optimal motion, for the re-planned motion, and its limits for all the solution in the feasible sub-set
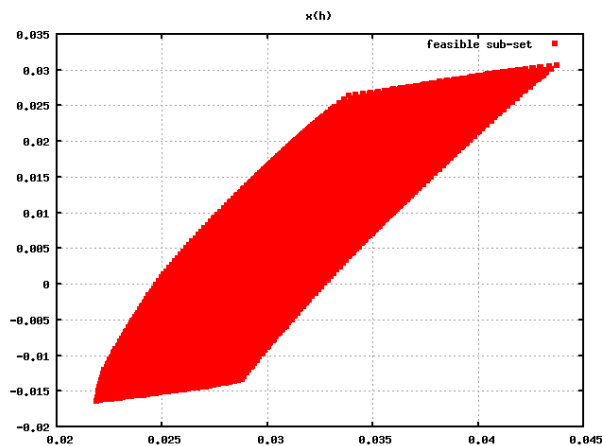

Fig. 11. Representation of the feasible couple $x(h)$

REFERENCES

[1] 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.

[2] 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.

[3] A. Piazzi and A. Visioli, "Global minimum-jerk trajectory planning of robot manipulators," in *IEEE Transactions on Industrial Electronics*, vol. 47, febru 2000, pp. 140–149.

[4] W. Suleiman, E. Yoshida, J.-P. Laumond, and A. Monink, "On humanoid motion optimization," in *IEEE-RAS 7th International Conference on Humanoid Robots*, 2007.

[5] 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.

[6] ——, "A new method for generating safe motions for humanoid robots," in *IEEE-RAS Internationnal conference on Humanoid robots*, 2008.

[7] ——, "Safe motion planning computation for databasing balanced movement of humanoid robots," in *submitted to IEEE International Conference on Robotics and Automation, ICRA*, 2009.

[8] O. Brock and O. Khatib, "Real-time re-planning in high-dimensional configuration spacesusing sets of homotopic paths," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 1, San Francisco, CA, USA, 2000, pp. 550–555.

[9] S. Behnke, M. Schreiber, J. Stuckler, R. Renner, and H. Strasdat, "See, walk, and kick: Humanoid robots start to play soccer," in *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, Genova,, Dec. 2006, pp. 497–503.

[10] S. Carpin and E. Pagello, "The challenge of motion planning for human robots playing soccer," in *Workshop on Humanoid Soccer Robots of the 2006 IEEE-RAS International Conference on Humanoid Robots*, December 2006, pp. 71–77.

[11] C. de Boor, *A Pratical Guide to Splines*. New York: Springer-Verlag, 1978, vol. 27.

[12] P.-B. Wieber, F. Billet, L. Boissieux, and R. Pissard-Gibollet, "The humans toolbox, a homogeneous framework for motion capture, analysis and simulation," in *the ninth ISB Symposium on 3D analysis of human movement*, 2006.

[13] W. Khalil and E. Dombre, *Modeling, Identification & Control of Robots*, 3rd ed., E. B. Heinemann, Ed. Hermes Sciences Europe, march 2002.

[14] M. Vukobratovic and D. Juricic, "Contribution to the synthesis of biped gait," *IEEE trans. Bio-Med Eng.*, vol. BME-16, pp. 1–6, 1969.

[15] 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.

[16] R. Hettich and K. O. Kortanek, "Semi-infinite programming: theory, methods, and applications," *SIAM Rev.*, vol. 35, no. 3, pp. 380–429, 1993.

[17] A. Piazzi 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.

[18] 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.

[19] A. Escande, S. Miossec, and A. Kheddar, "Continuous gradient proximity distance for humanoids free-collision optimized-postures," in *IEEE-RAS 7th International Conference on Humanoid Robots*, 2007.

[20] *Introduction to IPOPT : a tutorial for downloading, installing and using IPOPT*, april 7th 2006.

[21] 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.

[22] E. Hansen and G. Walster, *Global optimization using interval analysis*, 2nd ed. Marcel Dekker, 2004.

[23] N. Ramdani, M. Gouttefarde, F. Pierrot, and J. P. Merlet, "First results on the design of high speed parallel robots in presence of uncertainty," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, Sept. 2008, pp. 2410–2415.

[24] D. Oetomo, D. Daney, and J. P. Merlet, "Design strategy of serial manipulators with certified constraint satisfaction," *IEEE Transactions on Robotics*, vol. 25, no. 1, pp. 1–11, Feb. 2009.

[25] *ALIAS-C++ A C++ Algorithms Library of Interval Analysis for equation Systems*. [Online]. Available: http://www-sop.inria.fr/coprin/logiciels/ALIAS/ALIAS-C++/ALIAS-C++.html