Very fast decision making for whole body motion generation with humanoid robots

Olivier Stasse, *Member, IEEE,* Nicolas Perrin, Pierre-Brice Wieber, Nicolas Mansard, and Florent Lamiraux,

***Abstract*—High speed decision making to generate motion for humanoid robot is mandatory to deal with highly changing environments. Such contexts include human-robot interaction or emergency situations. We focus here on the problem of balancing. At first we explain a method allowing to plan very quickly footsteps. It is based on building an approximation function which takes as an input a transition between two steps and output a positive value if the transition is possible. An experiment on human-humanoid robot interaction illustrates the approach. In second we present a solver specifically targeted towards solving optimization problem for walking.**

***Index Terms*—Humanoids, fast decision making**

I. Introduction

THE AIM of this project is to explore high speed decision making by a humanoid robot for motion generation. Humans, such as firemen or sportsmen, are able to take a pertinent decision in the blink of an eye although such decision usually involves a rather time consuming deliberative process. The so- called snap-judgments are usually obtained after training, practice and an extended experience of similar situations. Robots interacting with humans performing collaborative work for instance, face the same kind of challenges. Indeed let us imagine a humanoid robot manipulating a table with a human (such as demonstrated in 2003 by HRG [1]). If the latter one loses grip of the table, the robot will have to quickly move appropriately to avoid putting in danger its human collaborator and itself. Finding in a timely manner a safe sequence of motion to avoid such situation is crucial. There are other fields of applications, less dramatic, where such capabilities would be useful such as entertainment where a small-size robot could imitate a child, or interact with him during a game. In such contexts, obviously it would not be possible to assume that the user is an expert able to program the robot appropriately.

A. Context

Reacting in the blink of an eye for security reason or during collaborative work is a very challenging task for a generic robot such as a humanoid. The problem generally consists in generating motions based on a representation of the world and taking into account the limitation of the robot. The representation of the world is usually partial and noisy, and in a human like environment it can be quite complex. A humanoid robot is considered in general well-suited to evolve in such kind of environments because of its volume and its wide range of possible configurations. The subsequent disadvantages of the versatility and the compactness are the complexity to generate motions where the robot keeps its stability. Two general approaches exist to address this problem: the reactive approach and the deliberative one. In both cases, one of the major difficulties with humanoid robots is to take into account the problem of stability.

B. Balancing

Through the well-known RABBIT [2], [3] [4] [5] and BIP [6] projects, several researchers proposed very advanced stability criteria. They demonstrated how it was possible to take into account the cyclic aspect of walking, impacts and even how to make a robot run. However those researches focused mostly on biped robots and not on humanoid as a generic robot. Moreover the current available humanoid platforms such as HRP-2 are not well suited for the control laws designed from this work. Researchers considering humanoid as a generic platform realize whole body motion in real-time in two steps: first to plan reference paths considering stability criteria and second to generate motion ensuring that the previously found paths are valid. The recent breakthroughs allowing real world applications such as ASIMO [7], Q-RIO [8] and HRP-2 [9] were all based on efficient resolution scheme achieved by simplifying the stability criteria and the dynamical model of the robot. The most widely used stability criterion is the Zero Momentum Point which assumes that the feet of the robot are in contact with a flat and horizontal floor. Regarding a simplification of the robot model, the most significant contribution is the cart-model proposed by Kajita [10], which constraints the CoM of the inverted pendulum on a plane. By considering a window over the near future, it is possible to consider either an analytical solution by adding a constraint on the ZMP functional space, either a preview control scheme [9]. The preview control is a linear optimal control based control strategy, working with a linearized model of the inverted pendulum. Although this window on the future is a key for planning dynamic motion, it generates only CoM trajectory. In order to perform planning, a Generalized Inverted Kinematics scheme is needed to generate the articular values corresponding to the CoM trajectory. Such scheme involves SVD computations which are quite time consuming. We propose therefore to use a machine learning based scheme to test ether or not a foot transition is possible or not considering joint limits, self-collision and dynamical stability. It is based on Recursive Stratified Sampling and Farthest Point Sampling.

O. Stasse and N. Perrin are with the CNRS-AIST JRL (Joint Robotics Laboratory), UMI3218/CRT AIST Central 2, Umezono 1-1-1, Tsukuba 305-8568 Japan (see http://staff.aist.go.jp/olivier.stasse).

P.-B.Wieber is with the BIBOP team, INRIA Rhones-Alpes, Inovalle 655 avenue de l'Europe, Montbonnot, 38 334 Saint Ismier Cedex France .

N.Mansard and F.Lamiraux are with the Gepetto team, LAAS, CNRS, 7 avenue du Colonel Roche - 31077 Toulouse Cedex 4, France.

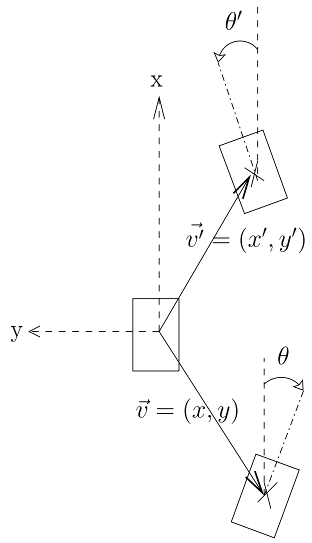


Fig. 1. Steps space. As the legs of the robot are symmetric, only one foot is consider here.

This method allows to identify the space of possible footsteps for any pattern generator. It is detailed in section II. The constraint on dynamical stability is important because the methods introduced in [9] and [11] tracked a ZMP reference trajectory but do not guarantee that it is inside the support polygon. To take into account this problem, and avoid to resort to a desired ZMP trajectory defined *a-priori* we have proposed to solve a constrained Quadratic Problem. However the off-the-shelf solvers available are quite time consuming. We proposed a new solver customized to our problem which is 10 times faster than those off-the-shelf solvers. This solver is briefly presented in section III.

II. Foot step planning

A. Problem statement

Let us assume that the robot is using a specific *g*ait *g*enerator

$$gg : \mathbf{x} \rightarrow \gamma \tag{1}$$

with $\mathbf{x} = [\mathbf{x}_{lf}^\top \mathbf{x}_{rf}^\top]^\top$, and γ is an articular trajectory $\mathbf{q}(t), \forall t \in [t_{begin}, t_{end}]$. A feasible trajectory should not make the robot self-collide, respect the articular limits and have the Foot Rotation Indicator projected into the support polygon. Based on those constraints, let us consider the following function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ such that

$$f(\mathbf{x}) \begin{cases} > 0 \text{ if } \gamma \text{ is feasible,} \\ \leq 0 \text{ if } \gamma \text{ is not feasible.} \end{cases} \tag{2}$$

Such function can be computed from γ by calculating the distance to the above constraints. However the time to generate γ and compute the distance to the constraints can be computationally expensive. In the context of probabilistic roadmap planning, the speed taking to shoot configuration in the footsteps space and checking either or not this configuration is feasible is important. It affects the resolution efficiency. Therefore our approach is to sample f and build an approximation $\hat{f}$ which can be evaluated very efficiently.

B. Overview of related work

The current solution is to only allow a small set of steps for the robot: in that case the generation and verification phases are useless since all trajectories can be memorized and verified off-line ([12], [13]). This approach is not always satisfying for it leads to a gait which has no flexibility, and combined with planning it often results in the robot making a large number of steps to perform a task for which only one or two steps would have been arguably enough. par Pre-computing robot dependent data-structure has been proposed in path planning for multi-body robots in the past [14]–[16]. In these papers a road-map is computed for a multi-body robot without obstacles. Once the robot is placed in an environment with obstacles, the pre-computed road-map is pruned by removing edges in collision with the obstacles. The remaining road-map is then used to plan paths.

Closer to our application, in [17] a 2 dimensional map is built which returns the time necessary to change a HRP-2 step-length during the flying phase of the foot in order to realize an emergency stop. The key-point of this work is to build a map which verifies that the ZMP realized by the robot stays in the support polygon for a given step-length modification done at a given time while walking. Indeed walking pattern generator such as the one proposed by Kajita et al. [18], or Morisawa [11] does not guarantee that the robot ZMP will stay in the support polygon. The main difference between this previous work and our approach is that we consider more constraints, and propose an adaptive partition of the input space well suited for higher dimensions. Indeed our work, taking into account free steps (their work only considers forward walking), has to aim at dealing with higher dimensions.

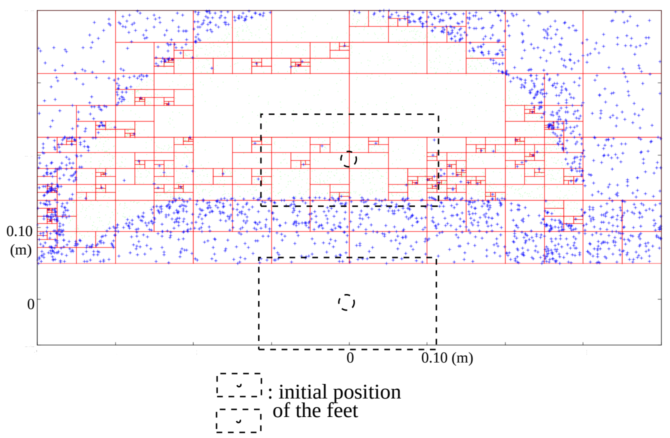


Fig. 2. The construction of the approximation $\hat{f}$

C. Adaptive sampling

In order to cope with the high-dimension aspect of the function the method build an approximation function which

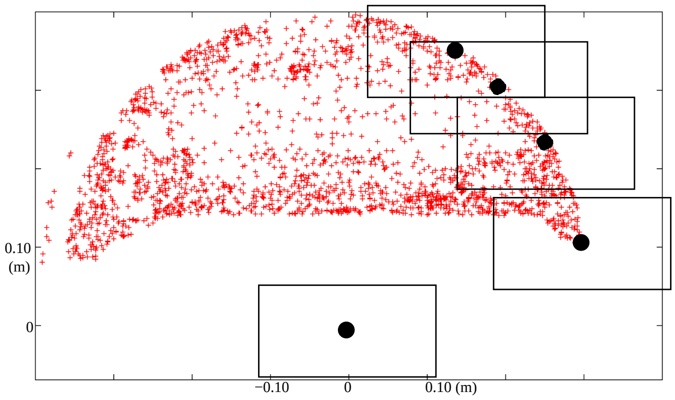


Fig. 3. The positive samples generated and some feasible steps

is not over-complete, but based on a tree representation of the input space. Inside the leaf-boxes, an optimization problem is solved to provide a local approximation similar to the Support Vector Regression method.

The leaf-boxes are separated in two categories:

- Boxes including positive and negative output values are called "frontier boxes" noted $\mathbb{B}_{Frontier}$. Empty boxes also belong to the frontier.
- Boxes including only only negative or positives values are called "regular boxes" noted $\mathbb{B}_{Regular}$.

Without further assumption on f it is quite difficult to decide when the approximation at the frontier is sufficiently well sampled, or when the regular boxes do not contain areas where f is of opposite sign. Therefore we assume that there is a probability ρ_{boost} that the sampling should be perform on $\mathbb{B}_{frontier}$ rather than $\mathbb{B}_{regular}$. This probability can be changed by the user to modify the behavior of the algorithm. Once the set from which to choose has been decided, the box with the lowest confidence is used to generate the sample.

D. Computing f

In this specific application f is computed by starting from the set of steps described in Fig. 1. By setting $\theta = \theta' = 0$ the dimension of the input space can be reduced to 4. The pattern generator used in this specific work is the one described in [9]. It generates a trajectory γ in the articular space. The distance to the constraints is computed at each time step of the trajectories: joint limits, self-collision and deviation of the fictuous ZMP from the desired trajectory. f is finally the smallest value of the distance to the constraints over the all trajectory. A major difficulty for the approximation scheme is to reflect the non-linearities introduces by having constraints both on the task space and on the articular values.

Locally the approximation scheme is using a similar representation to the Support Vector Regression scheme [19] without feature space.

E. Experimental results

1) Approximating feasibility of gaits for a given pattern generator: The pattern generator considered in this experiment is the preview control scheme proposed by Kajita in [9]. The implementation used in this experiment is described in [20]. It has the particularity to add a constraint between the waist and the CoM making sure that for one configuration of steps leads to one and only one γ trajectory. V-Clip was used to compute self-collision between the legs, while the Foot Rotation Indicator is used to compute the degree of stability of the robot.

Figure 2 shows the mechanism of the decision tree which recursively divides the input space into a disjoint union of rectangular cells. It also shows the negative samples (unfeasible steps); we can see some negative samples in what one would picture as the feasible area: this is due to numerical unstability in the computation of f, more specifically when a possible trajectory is close by a constraint. These slow the computation as it creates a new frontier. Several techniques used in machine learning for error tolerance might be helpful if we cannot make our simulation process more reliable. Figure 3 shows the positive samples. They are concentrated near the frontier between feasible and unfeasible arrival footprints, which is indeed the region on which the approximation scheme should focus.

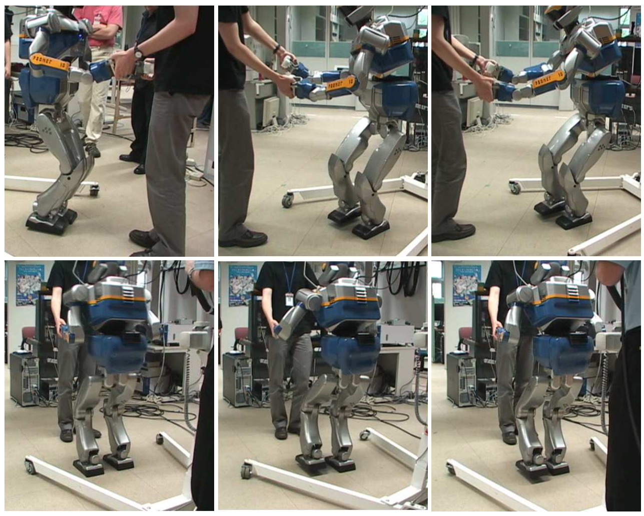


Fig. 4. Experimental results

2) Application: Human-Humanoid robot interaction: We have been able to successfully use this region for online footprints correction in an experiment where HRP2 is guided by a human holding its hands. The footprint to be corrected corresponds to a reference position of the left or right foot relatively to the current position of the (resp. left or right) hand, which the robot endlessly tries to go back to. Figure 4 shows the results of this experiment.

III. Solver to generate dynamically stable CoM trajectories

We recall here briefly the results presented in [21].

A. Problem statement

The Model Predictive Control (MPC) scheme introduced in [9], [22] for generating walking motions works primarily with the motion of the CoM of the walking robot. In order to obtain an LMPC scheme, it is assumed that the robot walks on a constant horizontal plane, and that the motion of its CoM is also constrained to a horizontal plane at a distance h above the ground, so that its position in space can be defined using only two variables (x, y).

Only trajectories of the CoM with piecewise constant jerks $\dddot{x}$ and $\dddot{y}$ over time intervals of constant length T are considered. That way, focusing on the state of the system at the instants $t_k = kT$,

$$\hat{x}_k = \begin{pmatrix} x(t_k) \\ \dot{x}(t_k) \\ \ddot{x}(t_k) \end{pmatrix}, \ \hat{y}_k = \begin{pmatrix} y(t_k) \\ \dot{y}(t_k) \\ \ddot{y}(t_k) \end{pmatrix}, \tag{3}$$

the integration of the constant jerks over the time intervals of length T gives rise to a simple recursive relationship:

$$\hat{x}_{k+1} = A\,\hat{x}_k + B\,\dddot{x}(t_k), \tag{4}$$

$$\hat{y}_{k+1} = A\,\hat{y}_k + B\,\dddot{y}(t_k), \tag{5}$$

with a constant matrix A and vector B.

Then, the position (z^x, z^y) of the ZMP on the ground corresponding to the motion of the CoM of the robot is approximated by considering only a point mass fixed at the position of the CoM instead of the whole articulated robot:

$$z^x_k = \begin{pmatrix} 1 & 0 & -h/g \end{pmatrix} \hat{x}_k, \tag{6}$$

$$z^y_k = \begin{pmatrix} 1 & 0 & -h/g \end{pmatrix} \hat{y}_k, \tag{7}$$

with h the constant height of the CoM above the ground and g the norm of the gravity force.

Using the dynamics (4) recursively, we can derive a relationship between the jerk of the CoM and the position of the ZMP over time intervals of length NT:

$$Z^x_{k+1} = P_{zs}\hat{x}_k + P_{zu}\dddot{X}_k, \tag{8}$$

$$Z^y_{k+1} = P_{zs}\hat{y}_k + P_{zu}\dddot{Y}_k, \tag{9}$$

with constant matrices $P_{zs} \in \mathbb{R}^{N\times 3}$ and $P_{zu} \in \mathbb{R}^{N\times N}$, with

$$Z^x_{k+1} = \begin{pmatrix} z^x_{k+1} \\ \vdots \\ z^x_{k+N} \end{pmatrix}, \ \dddot{X}_k = \begin{pmatrix} \dddot{x}_k \\ \vdots \\ \dddot{x}_{k+N-1} \end{pmatrix}, \tag{10}$$

and similar definitions for Z^y_{k+1} and $\dddot{Y}_k$.

In order for a motion of the CoM to be feasible, we need to ensure that the corresponding position of the ZMP always stays within the convex hull of the contact points of the feet of the robot on the ground [23]. This constraint can be expressed at the instants t_k for a whole time interval of length NT as:

$$b^l_{k+1} \leq D_{k+1} \begin{pmatrix} Z^x_{k+1} \\ Z^y_{k+1} \end{pmatrix} \leq b^u_{k+1}, \tag{11}$$

with a $D_{k+1} \in \mathbb{R}^{m\times 2N}$ a matrix varying with time but extremely sparse and well structured, with only $2m$ non zero values on 2 diagonals.

The LMPC scheme involves then a quadratic cost which is minimized in order to generate a "stable" motion [22], [24], leading to a canonical Quadratic Program (QP)

$$\min_u \ \frac{1}{2}u^T Q u + p_k^T u \tag{12}$$

with

$$u = \begin{pmatrix} \dddot{X}_k \\ \dddot{Y}_k \end{pmatrix}, \tag{13}$$

$$Q = \begin{pmatrix} Q' & 0 \\ 0 & Q' \end{pmatrix} \tag{14}$$

where Q' is a positive definite constant matrix, and

$$p_k^T = \begin{pmatrix} \hat{x}_k^T & \hat{y}_k^T \end{pmatrix} \begin{pmatrix} P_{su} & 0 \\ 0 & P_{su} \end{pmatrix} \tag{15}$$

where P_{su} is also a constant matrix (see [25] for more details).

With the help of the relationships (8) and (9), the constraints (11) on the position of the ZMP can also be represented as constraints on the jerk u of the CoM:

$$b'^l_{k+1} \leq D_{k+1} \begin{pmatrix} P_{zu} & 0 \\ 0 & P_{zu} \end{pmatrix} u \leq b'^u_{k+1}. \tag{16}$$

Since the matrix Q is positive definite and the set of linear constraints (16) forms a (polyhedral) convex set, there exists a unique global minimizer u^* [26].

The number of variables in the minimization problem (12) is equal to $n = 2N$ and the number of constraints (16) is of the same order, $m \approx 2N$. Typical uses of this LMPC scheme consider $N = 75$ and $T = 20\,ms$, for computations made on a time interval $NT = 1.5\,s$, approximately the time required to make 2 walking steps [24]. This leads to a QP which is typically considered as small or medium sized.

Another important measure to take into account about this QP is the number m_a of *active constraints* at the minimum u^*, the number of inequalities in (16) which hold as equalities. We have observed that at steady state, this number is usually very low, $m_a \leq m/10$, and even in the case of strong disturbances, we can observe that it remains low, with usually $m_a \leq m/2$ [24].

B. An optimized QP solver

1) Design choices: The solved developed in this work is based on an active set method, using a primal formulation and the range space of the constraints matrix. Because quite few constraints are active when solving the problem the active set method is faster than interior point method. The primal formulation has the advantage that the algorithm can be stopped and still provide a feasible solution even so it is sub-optimal. The constraint to provide a feasible solution when starting the algorithm can be easily tackled by solving a linear problem. The range space formulation of the constraint is motivated by the fact that its complexity is directly related to the number of active constraints which is quite small. Moreover, as the related matrices are not ill-conditioned the resolution do not perform poorly.

2) Off-Line Change of variables : The first action of a range space active set method is usually to make a Cholesky decomposition of the matrix $Q = L_Q L_Q^T$ and make an internal change of variable

$$v = L_Q^T u. \tag{17}$$

That way, the Quadratic Problem (12) simplifies to a Least Distance Problem (LDP) [27]

$$\min_{v} \ \frac{1}{2}\|v + L_Q^{-T} p_k\|^2.$$

In our case, we need to solve online a sequence of QPs (12)-(16) where the matrices Q', P_{zu} and P_{su} are constants. We can therefore make this change of variable completely off-line and save a lot of online computation time by directly solving online the LDP:

$$\min_{v} \ \frac{1}{2}\|v + p'_k\|^2 \tag{18}$$

with

$$p_k'^T = \begin{pmatrix} \hat{x}_k^T & \hat{y}_k^T \end{pmatrix} \begin{pmatrix} P_{su}L_Q^{-T} & 0 \\ 0 & P_{su}L_Q^{-T} \end{pmatrix} \tag{19}$$

and constraints

$$b_{k+1}'^{l} \leq D_{k+1} \begin{pmatrix} P_{zu}L_Q^{-T} & 0 \\ 0 & P_{zu}L_Q^{-T} \end{pmatrix} v \leq b_{k+1}'^{u}. \tag{20}$$

Realizing this change of variable off-line allows saving n^2 *flops* at each iteration of our algorithm. Note that, we measure computational complexity in number of floating-point operations, flops. We define a flop as one multiplication/division together with an addition. Hence, a dot product $a^T b$ of two vectors $a,\, b \in \mathbb{R}^n$ requires n flops.

C. Constraint activation

We have observed that not considering removing constraints does not affect the result we obtain from our LMPC scheme in a noticeable way. From the implementation viewpoint this allow to implement very efficient updates of the Cholesky decomposition of the constraint matrix. By observing the Lagrangian multipliers we can guess which constraint will be activated for the next iteration. Our final guess for the active set when doing so is in most cases correct or includes only one, and in rare cases two unnecessarily activated constraints. This leads to slightly sub-optimal solutions, which nevertheless are feasible. Furthermore, we have observed that, this does not affect the stability of our scheme: the difference in the generated walking motions is negligible.

D. Numerical results

Before implementing the algorithm described in this publication, the computation of our LMPC scheme relied on QL [**?**], a state of the art QP solver implementing a dual active set method with range space linear algebra. The fact that it implements a dual strategy implies that it can not be interrupted before reaching its last iteration since intermediary iterates are not feasible. Furthermore, no possibilities of warm starting are offered to the user. However, since it relies on a range space algebra, comparisons of computation time with our algorithm without warm starting are meaningful.

We naturally expect to gain n^2 flops at each iteration thanks to the off-line change of variable. Furthermore, QL does not implement double sided inequality constraints like the ones we have in (20), so we need to double artificially the number m of inequality constraints. Since computing the step α requires nm flops at each iteration and $m \approx n$ in our case, that's a second n^2 flops which we save with our algorithm. The mean computation time when using QL is 7.86 ms on the CPU of our robot, 2.81 ms when using our Primal Least Distance Problem (PLDP) solver. Detailed time measurements can be found in Fig. 5.

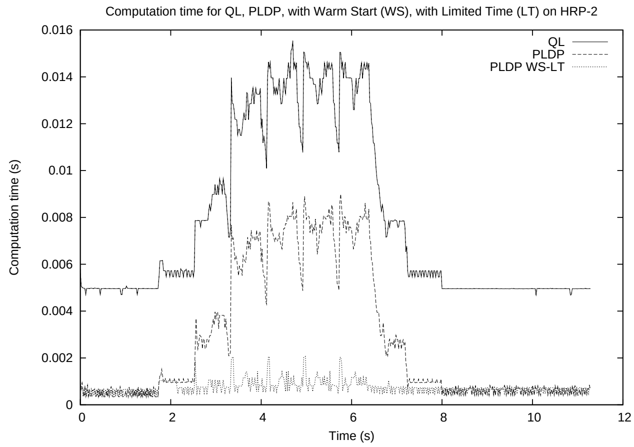


Fig. 5. Computation time required by a state of the art generic QP solver (QL), our optimized solver (PLDP), and our optimized solver with warm start and limitation of the computation time, over 10 seconds of experiments.

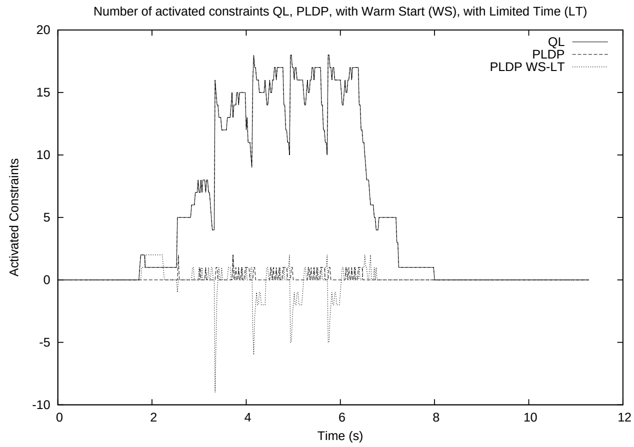


Fig. 6. Number of active constraints detected by a state of the art solver (QL), difference with the number of active constraints approximated by our algorithm (PLDP), between 0 and 2, and difference with the approximation by our algorithm with warm start and limitation of the computation time, between -9 and 2.

Even more interesting is the comparison with our warm start scheme combined with a limitation to two iterations for solving each QP. This generates short periods of sub-optimality of the solutions, but with no noticeable effect on the walking motions obtained in the end: this scheme works perfectly well, with a mean computation time of only 0.74 ms and, most of all, a maximum time less than 2 ms!

A better understanding of how these three options relate can be obtained from Fig. 6, which shows the number of constraints activated by QL for each QP, which is the exact number of active constraints. This figure shows then the difference between this exact number and the approximate number found by PLDP, due to the fact that we decided to never check the sign of the Lagrange multipliers. Most often, the two algorithms match or PLDP activates only one constraint in excess. The difference is therefore very small.

This difference naturally grows when implementing a maximum of two iterations for solving each QP in our warm starting scheme: when a whole group of constraints needs to be activated at once, this algorithm can identify only two of them each time a new QP is treated. The complete identification of the active set is delayed therefore over subsequent QPs: for this reason this algorithm appears sometimes to miss identifying as many as 9 active constraints, while still activating at other times one or two constraints in excess. Note that, regardless of how far we are from the real active set, the solution obtained in the end is always feasible.

IV. Conclusion

We have presented our current results in achieving fast decision making to generate whole-body motion generation. It is consists in designing an approximation function which can evaluate the feasibility of a foot-step in 300 μs. In addition we have proposed a solver of constrained Quadratic Program specifically designed for walking which is 10 times faster than well-known general QP-solver.

Acknowledgment

This work is supported by the French Agence Nationale de la Recherche, project R-Blink, grant reference ANR-08-JCJC-0075-01.

References

[1] K. Yokoyama, H. Handa, T. Isozumi, Y. Fukase, K. Kaneko, F. Kanehiro, Y. Kawai, F. Tomita, and H. Hirukawa, “Cooperative works by a human and a humanoid robot,” in *ICRA*, vol. 3, Sep. 14–19, 2003, pp. 2985–2991.

[2] F. Grognard and C. C. de Wit, “Design of orbitally stable zero dynamics for a class of nonlinear systems,” *Systems & Control Letters*, vol. 51, no. 2, pp. 89 – 103, 2004.

[3] S. Miossec and Y. Aoustin, “A simplified stability study for a biped walk with underactuated and overactuated phases.” *IJRR*, vol. 24, no. 7, pp. 537–551, 2005.

[4] J. Foret, O. Bruneau, and J.-G. Fontaine, “Dynamics and m-stability of legged robots,” *IJHR*, vol. 2(1), pp. 1–20, 2005.

[5] N.Khraief and N.K.M’sirdi, “Energy based control for the walking of a 7-dof biped robot,” *Acta Electronica, special issue ’Advanced Electromechanical motion systems’*, 2004.

[6] P.-B. Wieber and C. Chevallereau, “Online adaptation of reference trajectories for the control of walking systems,” *Robotics and Autonomous Systems*, vol. 54, no. 7, pp. 559–566, 2006.

[7] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, “The development of honda humanoid robot,” in *ICRA*, vol. 2. 1998, pp. 1321–1326.

[8] K. Nagasaka, Y. Kuroki, S. Suzuki, Y. Itoh, and J. Yamaguchi, “Integrated motion control for walking, jumping and running on a small bipedal entertainment robot,” in *ICRA*, vol. 4, 2004, pp. 3189–3194.

[9] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, “Biped walking pattern generation by using preview control of zero-moment point,” in *ICRA*, 2003, pp. 1620–1626.

[10] S. Kajita, T. Yamaura, and A. Kobayashi, “Dynamic walking control of a biped robot along a potential energy conserving orbit,” *IEEE Transactions on Robotics and Automation*, vol. 8, no. 4, pp. 431–438, Aug. 1992.

[11] M. Morisawa, K. Harada, S. Kajita, S. Nakaoka, K. Fujiwara, F. Kanehiro, K. Kaneko, and H. Hirukawa, “Experimentation of humanoid walking allowing immediate modification of foot place based on analytical solution,” in *ICRA*, 2007, pp. 3989–3994.

[12] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, “Motion planning for humanoid robots,” in *11th Intl Symp. of Robotics Research*, 2003.

[13] J. Chestnutt, J. Kuffner, K. Nishiwaki, and S. Kagami, “Planning biped navigation strategies in complex environments,” in *IEEE Intl Conf. on Humanoid Robotics*, 2003.

[14] P. Leven and S. Hutchinson, “Toward Real-Time Path Planning in Changing Environments,” *Algorithmic and Computational Robotics: New Directions: the Fourth Workshop on the Algorithmic Foundations of Robotics*, 2001.

[15] M. Kallmann and M. Mataric, “Motion planning using dynamic roadmaps,” in *ICRA*, 2004, pp. 4399- 4404.

[16] A. Nakhaei and F. Lamiraux, “Motion planning for humanoid robots in environments modeled by vision,” in *IJHR*, 2008, pp. 197-204.

[17] T. Takubo, T. Tanaka, K. Inoue, and T. Arai, “Emergent walking stop using 3-d zmp modification criteria map for humanoid robot,” in *ICRA*, 2007, pp. 2676–2681.

[18] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Yokoi, and H. Hirukawa, “A realtime pattern generator for biped walking,” in *ICRA*, 2002, pp. 31–37.

[19] A. J. Smola and B. Schölkopf, “A tutorial on support vector regression,” *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.

[20] O. Stasse, B. Verrelst, P.-B. Wieber, B. Vanderborght, P. Evrard, A. Kheddar, and K. Yokoi, “Modular architecture for humanoid walking pattern prototyping and experiments,” *Advanced Robotics, Special Issue on Middleware for Robotics –Software and Hardware Module in Robotics System*, vol. 22, no. 6, pp. 589–611, 2008.

[21] D. Dimitrov, P.-B. Wieber, H. Diedam, and O. Stasse, “On the application of linear model predictive control for walking pattern generation in the presence of strong disturbances,” in *ICRA*, May 2009, pp. 1171–1176.

[22] P.-B. Wieber, “Trajectory free linear model predictive control for stable walking in the presence of strong perturbations,” in *International Conference on Humanoid Robots*, 2006, pp. 137–142.

[23] ——, “Somme comments on the structure of the dynamics of articulated motion,” in *Fast Motions in Biomechanics and Robotics*, 2005.

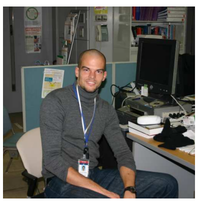
[24] D. Dimitrov, P.-B. Wieber, H. J. Ferreau, and M. Diehl, “On the implementation of model predictive control for on-line walking pattern generation,” in *ICRA*, 2008, pp. 2685–2690.

[25] H. Diedam, D. Dimitrov, P.-B. Wieber, K. Mombaur, and M. Diehl, “Online walking gait generation with adaptive foot positioning through linear model predictive control,” in *IROS*, 2008, pp. 1121–1126.

[26] J. Nocedal and S. J. Wright, *Numerical optimization 2nd edition*. Springer Series in Operations Research, 2000.

[27] R. Fletcher, *Practical Methods of Optimization*. John Wiley & Sons, 1981.

O. Stasse received the MSc degree in operations research (1996) and the PhD degree in intelligent systems (2000), both from Univer- sity of Paris 6. He is assistant professor of Paris 13. His research interests include humanoids robots as well as distributed and real-time computing applied to vision problems for complex robotic systems. Since 2003, he is a member of the Joint French-Japanese Robotics Laboratory (JRL) in a secondment position as a CNRS researcher.



N. Perrin received the MSc degree in theoritical computer science (2007) from University of Paris 7. He is currently a Phd student from University of Paul Sabatier Toulouse, at the Joint Robotics Laboratory, Tsukuba, Japan. His research interests include machine learning and humanoid motion planning.

P.-B. Wieber graduated from the Ecole Polytechnique, Paris, France, in 1996 and received his Ph.D. degree in Robotics from the Ecole des Mines de Paris, France, in 2000. Since 2001, he has been with the INRIA Rhône-Alpes in the BIPOP team. His research interests include the modeling and control of artificial walking.

N. Mansard graduated from École Nationale Supérieure d'Informatique et de Mathématiques Appliquées, Grenoble, France, in 2003, and received the M.S. (DEA) degree the same year in robotics and image processing from the University Joseph Fourier, Grenoble. He received the Ph.D. degree in computer science from the University of Rennes, Rennes, France, in 2006. He is currently a CNRS researcher with LAAS, Toulouse, in the Gepetto group. His research interests are concerned with sensor-based motion generation, and more specifically, the integration of servoing schemes into real robot applications. This includes visual servoing for mobile robot and sensor-based control for humanoid robots.

F. Lamiraux graduated from the Ecole Polytechnique, Paris, France, in 1993 and received his Ph.D. degree in Robotics from University of Toulouse, France, in 1997. Since 1999, he has been with the LAAS Toulouse in the Gepetto team. His research interests include Humanoid Robots Motion Planning and Control for Nonholonomic Robots.