Deliverable T12: Walking Pattern Generator

P.B. Wieber, A. Herdt, O. Stasse

April 9, 2010

Chapter 1

Introduction

In this deliverable we describe the development made on the problem of generating dynamically balanced gaits for the humanoid robot HRP-2 submitted to perturbations. The original proposal of R-Blink was separated into 3 parts regarding the problem of Real-Time Non Linear Predictive Control:

- Integration of complex model at the control level to solve the problem of equilibrium.
- Flexibility in the ankles.
- Stability with 3D environment.

The activity during this first year of the project focused mostly on the first point. Indeed it turned out to be a very fruitful source of interesting problems for which we designed several novel solutions. Starting from the seminal work of Kajita [8] based on a point mass model of the robot, we have proposed in [23] a new formulation of the problem allowing to compute a CoM reference trajectory from a set of foot-steps. The goal of the work realized in this project has been to extend this formulation to integrate more information from the whole robot model and expand the functionalities provided by [23]. During the first year of this project, we have been working on the following points:

- **Fast resolution scheme**: This work is based on a study realized while solving the problem formulated in [23] with standard off-the-shelf solvers. We realized that most of the constraints are never deactivated. Based on this observation we have proposed and implemented a new solver which out-perform generic state-of-the-art algorithms. This work has been published in [3].
- **Allowing automatic foot placement**: The system is able to decide autonomously the foot-steps to perform in order to follow a reference and cope with perturbations. The reference can be pre-defined foot-steps, or a mean CoM velocity. This approach described in section. 3 open many

interesting leads for instance visual servoing. The result of this work has been published in [7].

- **Planning foot-step trajectory for reduction of dynamical effects**: One reason to deal with complex robot model is to deal with the inertial effect induced by the motion of the flying foot. Such effect are particularly important with robots having heavy legs, or when performing fast leg motions. As the walking pattern generator is in charge of generating this trajectory, we have proposed to find the one minimizing the inertia on a simplified robot model with two masses. This work has been published in [15].

Chapter 2

Fast resolution scheme

2.1 An LMPC scheme generating walking motions

The Model Predictive Control (MPC) scheme introduced in [8, 23] 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 $t_k = kT$,

$$\hat{x}_k = \begin{pmatrix} x(t_k) \\ \dot{x}(t_k) \\ \ddot{x}(t_k) \end{pmatrix},\ \hat{y}_k = \begin{pmatrix} y(t_k) \\ \dot{y}(t_k) \\ \ddot{y}(t_k) \end{pmatrix}, \tag{2.1}$$

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

$$\hat{x}_{k+1} = A\,\hat{x}_k + B\,\dddot{x}(t_k), \tag{2.2}$$

$$\hat{y}_{k+1} = A\,\hat{y}_k + B\,\dddot{y}(t_k), \tag{2.3}$$

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:

$$z_k^x = \begin{pmatrix} 1 & 0 & -h/g \end{pmatrix} \hat{x}_k, \tag{2.4}$$

$$z_k^y = \begin{pmatrix} 1 & 0 & -h/g \end{pmatrix} \hat{y}_k, \tag{2.5}$$

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

Using the dynamics (2.2) recursively, we can derive a relationship between the jerk of the CoM and the position of the ZMP over time intervals of length NT:

$$Z^x_{k+1} = P_{zs}\hat{x}_k + P_{zu}\dddot{X}_k, \tag{2.6}$$

$$Z^y_{k+1} = P_{zs}\hat{y}_k + P_{zu}\dddot{Y}_k, \tag{2.7}$$

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

$$Z^x_{k+1} = \begin{pmatrix} z^x_{k+1} \\ \vdots \\ z^x_{k+N} \end{pmatrix}, \; \dddot{X}_k = \begin{pmatrix} \dddot{x}_k \\ \vdots \\ \dddot{x}_{k+N-1} \end{pmatrix}, \tag{2.8}$$

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:

$$b^l_{k+1} \leq D_{k+1} \begin{pmatrix} Z^x_{k+1} \\ Z^y_{k+1} \end{pmatrix} \leq b^u_{k+1}, \tag{2.9}$$

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 [23, 4], leading to a canonical Quadratic Program (QP)

$$\min_u \; \frac{1}{2}u^TQu + p_k^Tu \tag{2.10}$$

with

$$u = \begin{pmatrix} \dddot{X}_k \\ \dddot{Y}_k \end{pmatrix}, \tag{2.11}$$

$$Q = \begin{pmatrix} Q' & 0 \\ 0 & Q' \end{pmatrix} \tag{2.12}$$

where Q' is a positive definite constant matrix, and

$$p_k^T = \begin{pmatrix} \hat{x}_k^T & \hat{y}_k^T \end{pmatrix} \begin{pmatrix} P_{su} & 0 \\ 0 & P_{su} \end{pmatrix} \tag{2.13}$$

where P_{su} is also a constant matrix (see [2] for more details).

With the help of the relationships (2.6) and (2.7), the constraints (2.9) on the position of the ZMP can also be represented as constraints on the jerk u of the CoM:

$$b'^l_{k+1} \leq D_{k+1} \begin{pmatrix} P_{zu} & 0 \\ 0 & P_{zu} \end{pmatrix} u \leq b'^u_{k+1}. \tag{2.14}$$

Since the matrix Q is positive definite and the set of linear constraints (2.14) forms a (polyhedral) convex set, there exists a unique global minimizer u^* [14].

An 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 (2.14) 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 $m_a \leq m/2$ [3].

2.2 Numerical results

Before implementing the algorithm designed in the frame of R-Blink and described more precisely in [3], the computation of our LMPC scheme relied on QL [16], 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 (2.14), 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. 2.1.

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. 2.2, 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 algo-

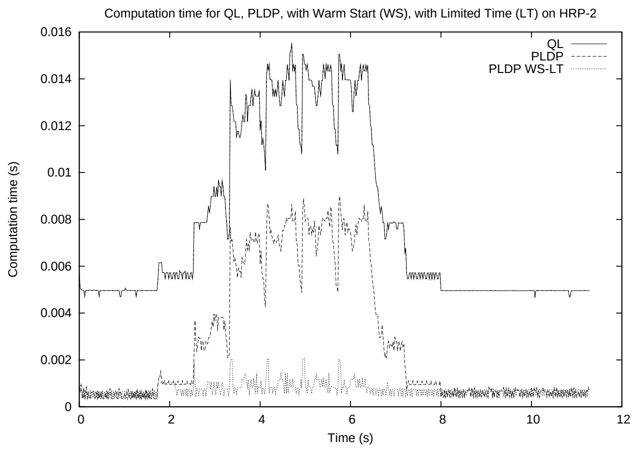

Figure 2.1: Computation time required by a state of the art generic QP solver (QL), our optimized solver (PLDP), and our optimized solver with warm start and limitation of the computation time, over 10 seconds of experiments.

rithm 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.

2.3 Conclusion

During this project, and published in [3], we introduced an optimized algorithm for the fast solution of a particular QP in the context of LMPC for online walking motion generation. We discussed alternative methods for the solution of QPs, and analyzed how do their properties reflect on our particular problem. Our algorithm was designed with the intension to use as much as possible data structures which are pre-computed off-line. In such a way, we are able to decrease the online computational complexity. We made a C++ implementation of our algorithm and presented both theoretical and numerical comparison with state of the art QP solvers. The issue of “warm-starting” in the presence of a real-time bound on the computation time was tested numerically, and we presented quantifiable results.

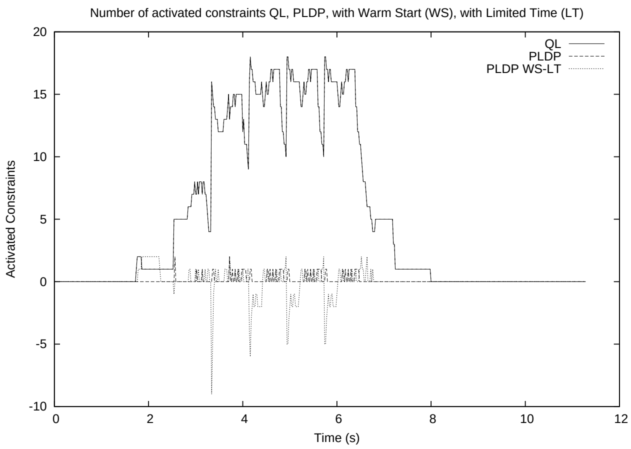

Figure 2.2: Number of active constraints detected by a state of the art solver (QL), difference with the number of active constraints approximated by our algorithm (PLDP), between 0 and 2, and difference with the approximation by our algorithm with warm start and limitation of the computation time, between -9 and 2.

Chapter 3

Automatic foot placement

3.1 Introduction

The difficulty in generating a stable walking motion mostly lies in the fact that the displacement of the Center of Mass (CoM) entirely relies on the contact forces between the feet and the ground, with the constraint that feet can only push on the ground [21, 22]. This restricts the motions that a walking system can realize, strongly limiting its capacity to follow a predefined motion in the presence of perturbations [23]. There is a strong interest therefore in being able to generate walking motions online, continuously adapting them to the current dynamics of the system.

A promising approach making use of a Linear Quadratic Regulator (LQR) has been proposed in [8]. Based on a linear approximation of the dynamics of the system, this approach tries to keep the contact forces in the middle of the feasible set, stabilizing the motion of the CoM of the system by minimizing its jerk over a finite prediction horizon. This LQR based approach allows generating stable walking motions online, with the possibility to continuously take into account the current state of the system [11, 13]. Variants have also been proposed [12]. But ignoring the exact constraints on contact forces limits its capacity to deal with difficult cases such as fast changes in the desired motion or strong perturbations.

In order to overcome this limitation, it has been proposed to introduce these constraints explicitly into the regulator, turning the LQR scheme into a more general Linear Model Predictive Control (LMPC) scheme, what led to a significant improvement in its capacity to deal with these difficult cases [23]. But both of these propositions were designed to work with foot step positions decided and fixed beforehand by a foot step planner, and we know that being able to adapt step positions online can contribute significantly to dealing with these difficult cases. It has been proposed therefore in [2] to introduce new control variables corresponding to the positions of the foot steps, allowing their real time adaptation with only a minimal modification to the existing LMPC scheme. But this scheme still needed that foot steps were planned beforehand, modifying them only to ensure feasibility and deal with perturbations.

The goal of the work briefly presented here and published in [7], is to show that this scheme can once again be slightly modified in order to obtain a fully automatic placement of the foot steps: a reference speed is given to the robot, which can be modified at any time, and according to this reference speed and the current state of the robot, a safe foot step placement is decided. An important feature of this new scheme is that even in case of an external perturbation or in case of a reference speed which is not realizable, the generated walking motion is always entirely safe and stable: the reference speed is tracked only as much as possible within the limits of stability.

Walking motion with automatic foot step placement has already been realized in [19] with a saturated PD regulator of the motion of the CoM, the saturation being used to satisfy the constraints on the Center of Pressure (CoP) and generate the foot step placements accordingly when required. However, feasibility and stability of these foot step placements is not ensured exactly. Moreover, the use of a fixed saturated PD controller for controlling at the same time the motion of the CoM and the foot step placements can only induce sub-optimal reactions. In our case, feasibility and stability constraints are handled explicitly, and both the motion of the CoM and the foot step placements are entirely free to be used to satisfy these constraints optimally.

3.2 AUTOMATIC FOOT STEP PLACEMENT

Since the feet of the robot can only push on the ground, the CoP can lie only within the support polygon; the convex hull of the contact points between the feet and the ground [21]. Any trajectory not satisfying this constraint can't be realized. It has been shown in [23] that the tracking of a reference position of the CoP in the QP (2.10) can be replaced by just enforcing these constraints on the position of the CoP. This allows to avoid specifying explicitly the shape of the walking motion, which then can be generated freely among all feasible motions.

A theoretical analysis of this Predictive Control scheme has been proposed in [24], showing that minimizing any derivative of the motion of the CoM of the robot while enforcing the constraints on the position of the CoP results in stable online walking motion generation. A nice detail there is that any control variable can be handled by this minimization process, including foot step placement. Noteworthy, in the original Predictive Control scheme introduced in [8] as well as in all similar works [11, 13, 20, 23], the foot step placement needs to be decided first of all.

Based on this observation and with a view on increasing the robustness of the walking motion with respect to external perturbations, an online adaptation of the feet positions has been proposed in [2]. The Predictive Scheme (2.10) was globally kept unchanged except for the introduction of new variables (X_k^f, Y_k^f) corresponding to the positions of the m foot steps taking place in the prediction horizon. These variables were adapted then in the minimization process, leading to the adaptation of the foot step placements. But still, reference foot step placements need to be given to this scheme.

We can conclude from all these previous results that it should be possible to generate a stable walking motion by only regulating the speed of the CoM to a desired mean value $(\dot{x}^{ref}, \dot{y}^{ref})$ and letting the foot step placement adapt fully automatically. Note that the dynamics of walking presents unavoidable effects such as the lateral sway motion implying that only a mean desired speed of the CoM can be sought for, as we will see in Section 3.3. We can consider therefore a QP as simple as

$$\min_{u_k} \frac{\beta}{2} \left\| \dot{X}_{k+1} - \dot{X}^{ref}_{k+1} \right\|^2 + \frac{\beta}{2} \left\| \dot{Y}_{k+1} - \dot{Y}^{ref}_{k+1} \right\|^2 \tag{3.1}$$

with variables

$$u_k = \begin{pmatrix} \dddot{X}_k \\ X^f_k \\ \dddot{Y}_k \\ Y^f_k \end{pmatrix}, \tag{3.2}$$

but we will see that better results can be obtained when re-introducing the same terms as in the QP (2.10):

$$\begin{aligned} \min_{u_k} \; & \frac{\alpha}{2} \left\| \dddot{X}_k \right\|^2 + \frac{\beta}{2} \left\| \dot{X}_{k+1} - \dot{X}^{ref}_{k+1} \right\|^2 + \frac{\gamma}{2} \left\| Z^x_{k+1} - Z^{x\_ref}_{k+1} \right\|^2 \\ & + \frac{\alpha}{2} \left\| \dddot{Y}_k \right\|^2 + \frac{\beta}{2} \left\| \dot{Y}_{k+1} - \dot{Y}^{ref}_{k+1} \right\|^2 + \frac{\gamma}{2} \left\| Z^y_{k+1} - Z^{y\_ref}_{k+1} \right\|^2 . \end{aligned} \tag{3.3}$$

These terms have however a different meaning here than in the original QP (2.10). The minimization of the jerk $(\dddot{X}, \dddot{Y})$ was necessary in (2.10) to generate stable motions whereas this is obtained in the QP (3.1) by just regulating the speed $(\dot{X}, \dot{Y})$. We will see however in the simulation results that a weakly weighted minimization of the jerk helps smoothing the contact forces and therefore the resulting motion. The tracking of a reference position of the CoP was necessary in (2.10) to generate a feasible motion whereas feasibility is obtained now by enforcing directly the constraints on the position of the CoP (which will be discussed in more length in the next section). We will see however that a weakly weighted centering the CoP under the feet allows faster and more robust reactions to changes in the state of the system or in the desired speed of the CoM.

More precisely, instead of having the CoP track a reference position (Z^{x_ref}, Z^{y_ref}) fixed in advance as in the original QP (2.10), we want it lie in the middle of the foot positions actually decided by our algorithm. With (X^{fc}_k, Y^{fc}_k) the current position of the foot on the ground (which can't be changed) and (X^f_k, Y^f_k) the positions of the following m steps which will be decided by our QP (3.1), we basically have

$$Z^{x\_ref}_{k+1} = U^c_{k+1} X^{fc}_k + U_{k+1} X^f_k \tag{3.4}$$

$$Z_{k+1}^{y\_ref} = U_{k+1}^c Y_k^{fc} + U_{k+1} Y_k^f \tag{3.5}$$

with

$$U_{k+1}^c := \begin{pmatrix} 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad U_{k+1} := \begin{pmatrix} 0 & 0 & \\ \vdots & \vdots & \\ 0 & 0 & \\ 1 & 0 & \\ \vdots & \vdots & \\ 1 & 0 & \\ 0 & 1 & \\ \vdots & \vdots & \\ 0 & 1 & \ddots \end{pmatrix}. \tag{3.6}$$

The ones in this vector $U_{k+1}^c \in \mathbb{R}^N$ and matrix $U_{k+1} \in \mathbb{R}^{N \times m}$ simply indicate which sampling times t_k fall into which step, where sampling times correspond to rows and steps to column, and therefore which foot position must be taken into account at what time.

We can express then this new QP in the same canonical form (2.10), but this time with a varying quadratic term because of the varying matrix U_{k+1}:

$$Q_k = \begin{pmatrix} Q'_k & 0 \\ 0 & Q'_k \end{pmatrix}, \tag{3.7}$$

$$Q'_k = \begin{pmatrix} \alpha I + \beta P_{vu}^T P_{vu} + \gamma P_{zu}^T P_{zu} & -\gamma P_{zu}^T U_{k+1} \\ -\gamma U_{k+1}^T P_{zu} & \gamma U_{k+1}^T U_{k+1} \end{pmatrix} \tag{3.8}$$

and

$$p_k = \begin{pmatrix} \beta P_{vu}^T (P_{vs} \hat{x}_k - \dot{X}_{k+1}^{ref}) + \gamma P_{zu}^T (P_{zs} \hat{x}_k - U_{k+1}^c X_k^{fc}) \\ -\gamma U_{k+1}^T (P_{zs} \hat{x}_k - U_{k+1}^c X_k^{fc}) \\ \beta P_{vu}^T (P_{vs} \hat{y}_k - \dot{Y}_{k+1}^{ref}) + \gamma P_{zu}^T (P_{zs} \hat{y}_k - U_{k+1}^c Y_k^{fc}) \\ -\gamma U_{k+1}^T (P_{zs} \hat{y}_k - U_{k+1}^c Y_k^{fc}) \end{pmatrix} \tag{3.9}$$

Hopefully, the matrix U_{k+1} varies cyclically with time so prefactorizing a whole cycle of matrices Q_k would still be possible to minimize online computation time.

3.3 Simulations

We'll consider the following sample motion on a simulated HRP-2 humanoid robot [10]: the robot starts from rest in double support and will walk continuously for 20 s, making a step regularly every 0.8 s. The robot starts with a zero reference velocity which is switched to 0.3 ms^{-1} forward at the beginning of the first step. The robot is pushed to the left at the beginning of step 3, at time $t = 2.4$ s. Then, in the middle of step 7, at time $t = 6$ s, the reference velocity is switched to 0.2 ms^{-1} on the right. Then, in the beginning of step 15,

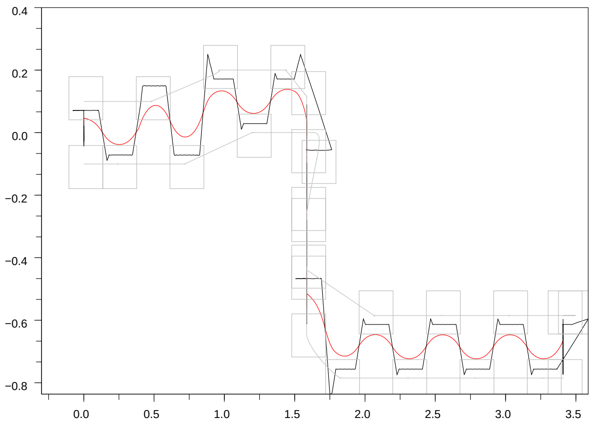

Figure 3.1: Foot step placement and ankle motion (grey), Center of Pressure (black) and Center of Mass (red) positions for the sample motion described in Section 3.3, obtained with the Predictive Control scheme (3.3).

at time $t = 12$ s, it is switched back to 0.3 ms^{-1} forward, and back to zero in the middle of step 22, at time $t = 18$ s.

With $N = 16$ time intervals of length $T = 0.1$ s, the prediction horizon is $NT = 1.6$ s, what corresponds to two steps. Assembling and solving the full QP (3.3) in this case takes less than 1 ms in average with state of the art solvers such as QL [16] or qpOASES [6, 5, 1], notwithstanding the possible optimizations presented in [4] or the optimized solver presented in [3] which can help reduce furthermore the computation time.

We can observe in Fig. 3.1 that the QP (3.3) manages to perfectly realize this desired motion and absorb the perturbation while always maintaining the CoP within the boundaries of the support polygon. More precisely, we have considered a safety margin, so the CoP always lie 3 cm inside the true boundaries of the support polygon. In fact, the position of the CoP plotted here corresponds to the approximate model (2.4)-(2.5), but the difference with the real CoP is usually less than 2 cm so this motion appears to be completely safe.

We can observe that when the push on the left occurs at the beginning of step 3, the robot is just beginning a single support on the left leg, which can not be moved therefore. And since it is forbidden for the robot to cross legs because of the risk of collisions between them, it is only at the end of step 4 that the left leg can be moved to the left in order to absorb the perturbation and recover a motion forward. In the mean time, the robot drifts to the left. This demonstrates one of the most valuable properties of this walking motion generation scheme: safety prevails, in the sense that the generated motion is always kept feasible, even if that means not realizing the desired motion. Here, the goal of the robot is to move forward, but this goal is fulfilled only when possible.

3.4 Conclusion

The LQR scheme originally proposed in [8] opened the way to online walking motion generations which could adapt to the state of the robot. But this scheme was designed to work with foot step positions decided and fixed beforehand by a foot step planner. We have shown here that a minimal modification to this LQR scheme, introducing the exact constraints on the Center of Pressure and new control variables corresponding to the positions of the foot steps, allows a fully automatic placement of these foot steps. We obtain then an online walking motion generation that can track a given reference speed of the robot, which can be modified at any time, while keeping the safety of the system under control even in the case of strong perturbations. We are working now on demonstrating these results on the real HRP-2 humanoid robot.

Chapter 4

Planning foot-step trajectory for reduction of dynamical effects

4.1 Introduction

Most of the human size humanoid robot includes in their ankle a compliant material to absorb the force resulting from the impact when the swinging foot is landing on the floor. A controller is then generally provided to compensate the effect of this compliant material [9]. Often on top of this controller a walking pattern generator provides articular (or torque) trajectories ensuring that the robot is balanced. Current real-time walking pattern generator assumes a simplified "inverted pendulum" model to simplify the problem of finding a CoM trajectory following a given ideal ZMP trajectory or satisfy the constrained related with the ZMP [8]. This simplified model does not take into account the compliant material, and let the underlying controller compensating the passivity of this material. Although this controller works effectively for moderate walking speeds, when performing fast motion or extended stepping over an object it is not sufficient. The compliant material deformation is such that the robot is leaning forward and might hit the floor sooner that expected and created large impact forces. Classically, this is addressed by adding masses to the model to take into account the inertia effect of the legs. However instead of modifying the CoM trajectory and fixing a 3rd order polynomial trajectory for the feet, we want to modify the foot trajectory itself. Indeed for a service robot, the upper body part is used to perform task such as grasping, holding, while the swinging foot trajectory main task is to move towards the next support foot position. Therefore modifying the swinging foot trajectory seems to be a rational choice if the upper body is constrained by other tasks. In the frame of this deliverable, we have proposed a method to plan the swinging foot trajectory in order

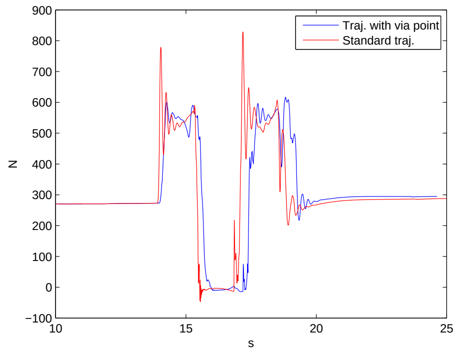

Figure 4.1: Force read at the left foot

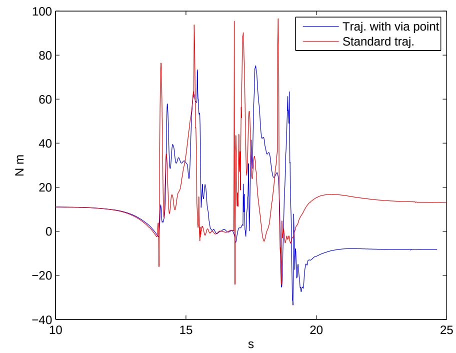

Figure 4.2: Torque reading at the left foot

to minimize the inertia effect on the compliant material. This work has been published in [15].

4.2 Problem statement

Let us consider a foot embedding a compliant material between the sole and a force sensor as depicted in Fig.4.3. Following [9] the floor reaction torque τ is given by:

$$\tau = -k_e(q - q_e) \tag{4.1}$$

When this foot is the support foot and the robot is moving forward, a desired behavior is to avoid any backdrivability, i.e. any deformation of q_e Fig.4.3. This can happen for robots having a mass distribution not concentrated on the waist or when the robot is performing large step forward. For instance in [18], HRP-2

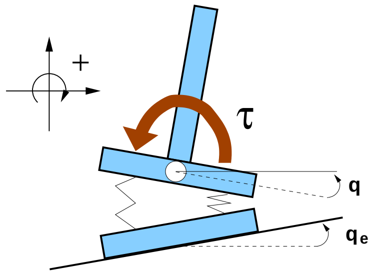

Figure 4.3: The flexible material is located between the sole and the force sensor.

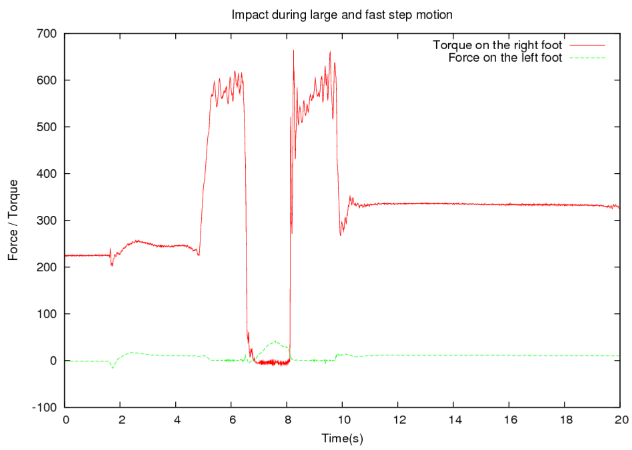

Figure 4.4: Force/Torque sensor read in case of the trajectory with via point

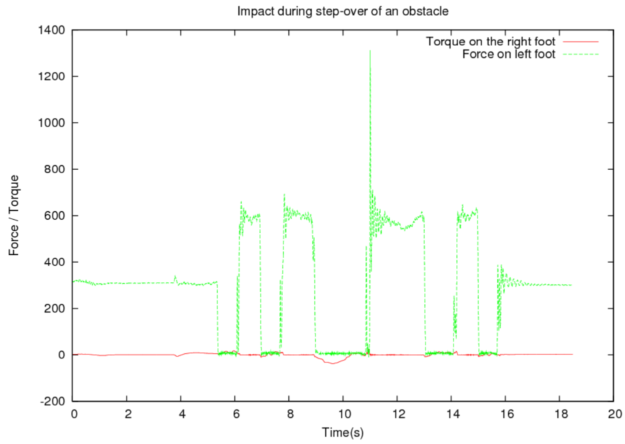

Figure 4.5: Force/Torque sensor read in case of a standard trajectory

is stepping over an obstacle and is performing a step which twice longer than normal standard mode. The use of the classical controller to compensate for the compliant material, the inverted pendulum for the CoM trajectory, and a 3rd polynomial for the feet trajectory generate impact forces about twice the weight of the robot. In [18] the foot trajectory has been changed to a Clamped-Spline to minimize the deformation of the compliant material. But this modification was mostly heuristically motivated. In [15] we give a more well-grounded way of finding swinging foot trajectories which minimizes the angular momentum relative to the CoM.

4.3 Approach

The floating leg may be modeled in a simplified way as a two link kinematic chain (double pendulum). Within the interest of the paper, this model is sufficient for modeling the robot leg in the sagital plane during the single support phase. In fact, the joint 1 represents the waist and joint 2 the knee and their movements have to be considered relative with respect to the CoM (Center of Mass). In our model, we resume the mass of the whole link 1 with m_1 and regarding the foot and the leg with m_2 . The lengths of the links are l_1 and l_2 . The system is free to move in a vertical plane. The local acceleration of gravity is g. The equations of motion for the double pendulum are well documented in literature [17], and can be calculated using the Lagrange formulation. The joint space dynamic model can be written in a compact matrix form:

$$\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \tau \tag{4.2}$$

where $\mathbf{B}(\mathbf{q})$ represents the inertia matrix which is:

$$\mathbf{B}(\mathbf{q}) = \begin{bmatrix} (m_1 + m_2)l_1^2 & m_2 l_1 l_2 cos(\theta_1 - \theta_2) \\ m_2 l_1 l_2 cos(\theta_1 - \theta_2) & m_2 l_2 \end{bmatrix} \tag{4.3}$$

$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ represents the quadratic velocity terms:

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} 0 & m_2 l_1 l_2 sin(\theta_1 - \theta_2)\dot{\theta}_2 \\ -m_2 l_1 l_2 sin(\theta_1 - \theta_2)\dot{\theta}_2 & 0 \end{bmatrix} \tag{4.4}$$

and $\mathbf{g}(\mathbf{q})$:

$$\mathbf{g}(\mathbf{q}) = \begin{bmatrix} (m_1 + m_2)l_1 g sin(\theta_1) \\ m_2 l_2 g sin(\theta_2) \end{bmatrix} \tag{4.5}$$

In order to find a trajectory for the double pendulum, we can suppose to use a via point through which the trajectory should pass at time t_c. This way, the trajectory will be defined by a piece-wise polynomial of the desired order, depending on the initial and final conditions we want to specify. If it is desired to reduce the torque to the CoM due to the movement of the foot, the following vector from Eq. 4.3 should be considered:

$$\tau_d(\mathbf{q}, \dot{\mathbf{q}}) = [\ 1\ 0\ ]\,[\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}] \tag{4.6}$$

The function we want to minimize is

$$f^* = \frac{1}{T}\left| \int_0^{t_c} \tau_{d,1} dt + \int_{t_c}^{T} \tau_{d,2} dt \right| \tag{4.7}$$

This cost function has to be minimized given constraints such as null speed and acceleration in the task space. Details on finding a solution minimizing Eq. 4.7 can be found in [15].

4.4 Experimental results

4.4.1 Simulated experiments

In this section, the results of the simulation using a trajectory specification in the task space are presented.

In Fig. 4.1, we show the force in the z direction (i.e. the direction perpendicular to the ground) representing the impact effect on the ground. As it can be seen from the graph, the impact is considerably reduced using the via point. This results is almost straightforward since the trajectory is specified in order to start and arrive with zero acceleration.

The results in Fig. 4.2 show the validity of the work presented in [15]. The torque acting on the left foot during the right foot floating is actually related to the torque acting on the CoM. It is possible to see that this torque is greatly reduced with respect to a standard trajectory.

4.4.2 Experiences on the humanoid robot HRP-2 and conclusion

The proposed algorithm has been tested over the robotic platform HRP-2. In Fig. 4.5 we present the results of the impact on the ground based on [18]. In this case, huge impact of about 1200 N has been reduced down to about 650 N (as shown in fig. 4.4). Also, the torque measured by the force/torque sensor shows the validity of the proposed work.

Chapter 5

Conclusion

During the first year of the project, we have mostly investigated the integration of complex models when solving the problem of equilibrium. First we are now able to take a reference CoM speed and generate the appropriate set of CoM, ZMP and foot trajectories without any additional information [**?**]. This open the way to integrate other criteria such as visual information. For this reason, Claire Dune, who is now a JSPS fellow at JRL, is currently working on integrating visual servoing using the approach presented in this deliverable. A paper has been submitted to IROS on this specific topic. In second, we also have addressed the problem of solving constrained QP in a limited amount of time [3]. Finally we have bridge some gaps between planning and control by proposing a method taking into account dynamic consideration to reinforce the stability of the system. This has been achieved by using a two masses model of the robot lower part[**?**]. During this second year we will pursue our current goal, while investigating the problem of controlling the compliant material. A student will be in charge of implementation the appropriate algorithms.

5.1 Bibliography

[1] http://homes.esat.kuleuven.be/ optec/software/qpoases/.

[2] 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1121–1126, 2008.

[3] 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 *IEEE/RAS International Conference on Robotics and Automation*, pages 1171–1176, 2009.

[4] 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

IEEE/RAS International Conference on Robotics and Automation, pages 2685–2690, 2008.

[5] H.J. Ferreau, H.G. Bock, and M. Diehl. An online active set strategy to overcome the limitations of explicit mpc. *International Journal of Robust and Nonlinear Control*, 18(8):816–830, 2008.

[6] H.J. Ferreau, P. Ortner, P. Langthaler, L. del Re, , and M. Diehl. Predictive control of a real-world diesel engine using an extended online active set strategy. *Annual Reviews in Control*, 31(2):293–301, 2007.

[7] A. Herdt, D. Holger, P.B. Wieber, D. Dimitrov, K. Mombaur, and D. Moritz. Online walking motion generation with automatic foot step placement. *Advanced Robotics*, 2010. accepted.

[8] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *IEEE/RAS International Conference on Robotics And Automation*, pages 1620–1626, 2003.

[9] S. Kajita, K. Yokoai, M. Saigo, and K. Tanie. Balancing a humanoid robot using backdrive concerned torque control and direct angular momentum feedback. In *IEEE/RAS International Conference on Robotics and Automation*, page 33763382, 2001.

[10] K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukawa, T. Kawasaki, M. Hirata, K. Akachi, and T. Isozumi. Humanoid robot hrp-2. In *IEEE/RAS International Conference on Humanoid Robotics*, pages 1083–1090, 2004.

[11] S. Kanzaki, K. Okada, and M. Inaba. Bracing behavior in humanoid through preview control of impact disturbance. In *IEEE/RAS International Conference on Humanoids Robots*, pages 301–305, 2005.

[12] 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 *IEEE/RAS International Conference on Robotics and Automation*, volume 4, pages 3189–3194, 2004.

[13] K. Nishiwaki and S. Kagami. High frequency walking pattern generation based on preview control of zmp. In *IEEE/RAS International Conference on Robotics and Automation*, pages 2667–2672, 2006.

[14] J. Nocedal and S. J. Wright. *Numerical Optimization*. Series in Operations Research. Springer, 2 edition, 2000.

[15] P. Pierro, O. Stasse, A. Kheddar, K. Yokoi, and C. Balaguer. Humanoid feet trajectory generation for the reduction of the dynamical effects. In *IEEE/RAS International Conference on Humanoid Robotics*, pages 454–458, 2009.

[16] K. Schittkowski. Ql: A fortran code for convex quadratic programming - user's guide, report. Technical report, Department of Mathematics, University of Bayreuth, 2003.

[17] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: Modelling, Planning and Control*. Springer-Verlag, 2009.

[18] O . Stasse, B. Verrelst, B. Vanderborght, and K. Yokoi. Strategies for humanoid robots to dynamically walk over large obstacles. *IEEE Transactions on Robotics*, 25:960–967, 2009.

[19] T. Sugihara. Simulated regulator to synthesize zmp manipulation and foot location for autonomous control of biped robots. In *IEEE/RAS International Conference on Robotics and Automation*, pages 1264 – 1269, 2008.

[20] H. Takeuchi. Real time optimization for robot control using receding horizon control with equal constraint. *Journal of Robotic Systems*, 20(1):3–13, 2003.

[21] P.-B. Wieber. On the stability of walking systems. In *Proceedings of the International Workshop on Humanoid and Human Friendly Robotics*, 2002.

[22] P.-B. Wieber. Some comments on the structure of the dynamics of articulated motion. In *Fast Motions in Biomechanics and Robotics*, 2005.

[23] P.-B. Wieber. Trajectory free linear model predictive control for stable walking in the presence of strong perturbations. In *IEEE/RAS International Conference on Humanoid Robotics*, pages 137–142, 2006.

[24] P.-B. Wieber. Viability and predictive control for safe locomotion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1103–1108, 2008.