Model predictive control of a humanoid robot

Andrei Herdt

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

MINES ParisTech

École doctorale n°84 :
Sciences et technologies de l'information et de la communication

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

l'École nationale supérieure des mines de Paris

Spécialité « Informatique temps-réel, robotique et automatique »

présentée et soutenue publiquement par

Andrei HERDT

le 27 janvier 2012

Commande prédictive d'un robot humanoïde
Model predictive control of a humanoid robot

Directeur de thèse : **Bernard ESPIAU**
Co-encadrement de la thèse : **Pierre-Brice WIEBER**

Jury

M. Franck MULTON, Professeur, UFR-STAPS, Université Rennes 2	Président
M. Mazen ALAMIR, Professeur, Gipsa-lab, Dép. Automatique, Univ. de Grenoble	Rapporteur
M. Fathi BEN OUEZDOU, Professeur, LIRIS, Univ. de Versaille St Quentin	Rapporteur
M. Bernard ESPIAU, Directeur de Recherche, INRIA Grenoble - Rhône-Alpes	Examinateur
M. Pierre-Brice WIEBER, Chargé de Recherche, INRIA Grenoble - Rhône-Alpes	Examinateur

MINES ParisTech
INRIA Grenoble - Rhône-Alpes
655 Avenue de l'Europe 38334 Montbonnot - St Ismier, France

T H È S E

The work in this thesis has been carried out within the context of the French national projects R-Blink and Romeo.

Contents

List of Figures

Introduction

This thesis contributes to the domain of Humanoid Robot Motion Control; more precisely, the model-based online generation of walking trajectories. Walking motions can be generated online through the control of reduced dynamic models that are rich enough in motion information and subsequent tracking of the resulting trajectories by the complete robot. By focussing on key characteristics of the gait, the computational burden can be reduced, redistributing it into a global, low-dimensional (decision) part and a local, high-dimensional (tracking) part. Since precomputed reference motions become unnecessary, robustness and reactivity are expected to be superior to the converse approach where adaptation schemes have to be employed to choose or interpolate between static reference motions.

During the last decade, several control schemes have been developed that demonstrate the capacity of simplified models to generate robust and reactive gaits. But, the computational load is small enough only for very simple models that generate relevant deviations from their comprehensive counterparts, and the associated walking style is inefficient and unnatural looking. The realizability of the generated motions and the overall stability of the robot are typically assured by severely constraining the set of possible solutions. Trajectories are obtained with the help of strong assumptions on the motions and the environment that can easily lose their validity. These measures reduce the robot's capacity of taking appropriate decisions and lower the gain due to the reduced computational time and the absence of offline computed motion references.

In this thesis, the problem of trajectory generation is addressed from the perspective of Model Predictive Control (MPC). This point of view will allow us first to generalize and second to improve existing solutions in this do-

main. The main goal of this thesis consists in the development of a generic MPC-based trajectory generation scheme that meets the target flexibility and reactivity required for autonomous walking. Embedded in a real-time control framework, it is designed to ease and to improve the interaction with other control instances and the user by means of a more abstract interface. A second goal consists in improving the efficiency and performance of the control scheme itself by incorporating more complex gait models.

Its main contributions are:

- Development of a Linear Model Predictive Control (LMPC) scheme for real-time walking control.
- Extension of the scheme to natural use of toes and oscillations of the Center of Mass and multiple-mass models for a more precise control of the motions.
- Verification on different robotic platforms, including the HRP-2, the HRP-4c and Romeo in a direct user-robot interface as well as in a visual feedback loop.
- A Nonlinear MPC (NMPC) prototype capable of computing three dimensional walking trajectories in real-time.

In Chapter 1, we will show the key challenges of motion control for humanoid robots. We will give an overview of the major realizability restrictions and a thorough definition of a robot's stability. This will serve us then to justify the suitability of Model Predictive Control for motion generation. A generic MPC scheme will be formulated that will serve as a reference for the definition of control laws throughout the thesis.

Throughout Chapter 2, we will mention several possible linearized formulations of this motion generation scheme. We will discuss, in thorough detail, one single formulation that allows for free foot placement and control through velocity references, therefore capable to be implemented without further modifications in a visual feedback loop, and we will discuss its real-time implementation. Simulations and experiments on robotic platforms will serve us to evaluate the performance of the scheme in particular and the potential of linear MPC for walking control in general.

Chapter 3 will serve as an example for its generalizability. Extensions of the scheme will be introduced that allow for considerably more efficiency and walk-

ing performance through varying waist height and the use of toes. To reduce the modeling error that becomes significant for faster motions, a multiple-mass model is proposed.

In Chapter 4, we introduce a nonlinear MPC scheme capable of controlling nonlinear dynamic models, and we embed it in a real-time computation framework. Extended by compliant legs it will prove its capacity to reproduce major characteristics of the human gait.

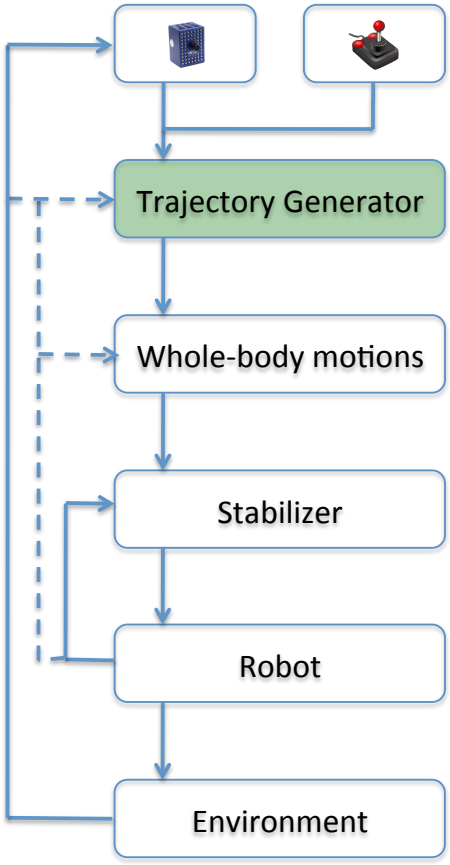


Figure 1: One possible control structure. The emphasis of this thesis lies on the generation of low-dimensional trajectories (filled entity). The solid interconnections between instances have been considered in this thesis, whereas the dashed ones are possible feedback interconnections.

Chapter 1

Controlling the displacement

Among the objectives that a legged robot may fulfill, keeping the balance plays an imperative role. Any other goal, including the locomotion itself, can be achieved only if the robot manages to gain sufficient foothold. The central question that has to be answered when formulating a walking control law can therefore be stated as follows: How to control the displacement of the overall body such that the robot keeps balance and secondary goals can be fulfilled?

This chapter serves to justify the approach taken in this thesis from the theoretical point of view. Leaving the analysis of the actual performance of the developed schemes to subsequent chapters, we will focus in the following on the feasibility and stability shown here as essential requirements on generated motions. In a first part, we will highlight the dynamic structures that govern and restrain the displacement of legged robots. We will, in a second part, adapt a concept of stability, and we will give a sufficient condition for its achievement. This will serve us finally to formulate a general Model Predictive Control scheme for the displacement control of a biped robot.

1.1 Constrained dynamics

Additionally to the conceptual properties of classical robot manipulators, comprehensive dynamics of a biped robot exhibit redundant degrees of freedom and several end-effectors. Complex interactions occur during the unavoidable contact making, generating impact forces that can destabilize the system. Once contact is established, its unilateral nature prohibits the generation of forces

in any direction so that foothold cannot be maintained for arbitrary motions. How to control the displacement of such a complex and highly constrained system?

1.1.1 Internal constraints

Let us first have a look at the equations of motion when the robot is constrained only by its internal structure. With some generalized coordinates q describing the robot's configuration in a global frame, we can express motions by an equilibrium condition between the generalized internal and external efforts:

$$T(q, \dot{q}, \ddot{q}) = \Gamma(q, \tau) + \Lambda(q, f), \tag{1.1}$$

with $T(q, \dot{q}, \ddot{q})$ encompassing the efforts resulting from the robot's dynamic properties, $\Gamma(q, \tau)$ the efforts resulting from the actuator torques τ, and $\Lambda(q, f)$ the efforts applied by external forces f.

Considering the fact that actuators can generate efforts only on a subset of q, the robot's joints, this equation can be decoupled into two parts (cf. [Wieber, 2000]):

$$\begin{bmatrix} T_1(q, \dot{q}, \ddot{q}) \\ T_2(q, \dot{q}, \ddot{q}) \end{bmatrix} = \begin{bmatrix} \Gamma(q, \tau) \\ 0 \end{bmatrix} + \begin{bmatrix} \Lambda_1(q, f) \\ \Lambda_2(q, f) \end{bmatrix}, \tag{1.2}$$

where the first part is actuated by the joint motors and the contact forces, whereas the second part is actuated only by contact forces. As a not surprising consequence: A robot requires (sufficient) contact forces to realize a motion $q(t)$.

1.1.2 External constraints

Now, contrary to the fixed base of a classical robot, the feet of a humanoid robot can only push the ground and friction is finite, i.e. the normal and tangential components f_n and f_t of the reaction forces are bounded:

$$f_n \geq 0, \tag{1.3}$$

$$\|f_t\| \leq f_t^{max}(f_n). \tag{1.4}$$

Hence, the generalized efforts vector $\Lambda_2(q, f)$ is limited, and the robot might not be able to realize a desired motion $q(t)$, even if sufficient efforts can be

generated by the actuators. The less the support base is extended, the more the set of realizable trajectories is restricted. Since the feet of a *human sized* robot are small compared to its overall size, the efforts that can be applied by them are very limited, requiring a frequent change of support positions during walking. And because every unrealizable motion reference might induce a loss of equilibrium, the above two conditions give the second part of (1.2) a particular importance for walking control.

Still, the considerations above are not enough to assure that a generated trajectory can be realized by the system. Like every robot, a humanoid robot is also restricted by the characteristics of the actuators in the joints and its geometric structure:

- The torques that can be generated by the joint actuators are always limited. However, more restricting for walking are usually their limitations in the maximal angle and velocities:

$$(q, \dot{q}) \in \mathcal{K}. \tag{1.5}$$

- By definition, a biped robot has at least two extremities. The motions of the legs have to be planned such that they do not interfere with each other during walking:

$$q \in \mathcal{G}_I. \tag{1.6}$$

- The main motivation that governs the development of humanoid robots is their use in the human environment and in interaction with the humans itself. The designated workspace can thus be very restrained and dynamic. This requires exact knowledge about the limitations posed by the environment when generating motions:

$$q \in \mathcal{G}_E. \tag{1.7}$$

We neglect in the following the behavior of the system during the impact phase. We are supposing for this purpose that impact forces are negligibly small and cannot cause loss of contact or sliding. This supposition is reasonable for walking when shock reducing measures in the mechanical structure and the executed trajectories have been taken.

With the above considerations we get able to define a comprehensive dy-

namic model to be used throughout this thesis:

$$T_2(q, \dot{q}, \ddot{q}) = \Lambda_2(q, f) \tag{1.8}$$

$$(q, \dot{q}) \in \mathcal{K} \tag{1.9}$$

$$q \in \mathcal{G} \tag{1.10}$$

$$f_n \geq 0 \tag{1.11}$$

$$\|f_t\| \leq f_t^{max}(f_n). \tag{1.12}$$

1.2 Stability of a humanoid robot

In related literature, stability and feasibility are very often not strictly separated in the sense that a (finite) reference or the robot is assumed to be stable if it is statically or dynamically equilibrated [Kuffner et al., 2002] [Tedrake, 2004] [Dalibard et al., 2009] [Nishiwaki and Kagami, 2010] [Hirukawa et al., 2006]. We can guess that the crucial observation there is that if a robot is not dynamically equilibrated, then it will fall, so that the intuitively made, underlying definition of stability must be:

Definition 1. *A robot is stable if it will not fall.*

But even if we adapt this definition, two objections can be raised against expressing it directly as a feasibility constraint condition. First, feasibility at a certain point in time does not imply that it will be maintained in the future. Second, and as a consequence, a stability definition has to consider the *totality* of the system, including the employed *control law*.

1.2.1 Feasibility implies stability

On the contrary, we can state approximately that any motion starting from an unfeasible state with respect to the conditions of the previous section will stay unfeasible since the handling of such situations is difficult, and a fall becomes likely. These feasibility conditions define thus the borders to an 'undesirable' set of states $\mathcal{U}$ out of which a fall might be unavoidable.

Therefore, to avoid falling, a *necessary* condition for a robot is to stay outside of $\mathcal{U}$, which is possible only from subsets of the feasible state-space $\mathcal{F}$. The union of this subspaces, the *viability kernel* $\mathcal{V}$, consists of all states

$(q, \dot{q})$ for which at least one motion exists that permits to stay out of $\mathcal{U}$ [Aubin, 1991].

Necessary condition for stability. *A state $(q(t_0), \dot{q}(t_0))$ cannot be stable, in the sense of Definition 1, if it is not viable, i.e., no motion exists such that, starting from this state, the system can avoid to get inside $\mathcal{U}$: $q(t) \in \mathcal{V} \subset \mathcal{F}$ $\forall t \geq t_0$*

Motions inside the viability kernel are always realizable, but, as mentioned in [Wieber, 2000], it is neither clear how to compute the viability kernel of a humanoid robot, nor if this computation is even feasible. What counts more however, is that the knowledge of the actual viability kernel might not even be of practical use, since it is not related to the actually employed control law. The *invariant* set $\mathcal{S}$ of a real control law is always a subspace of the viability kernel. A *sufficient* condition for the stability of a humanoid robot is to avoid motions that lead its state out of $\mathcal{S}$.

Sufficient condition for stability. *A state $(q(t_0), \dot{q}(t_0))$ is stable, in the sense of Definition 1, if it is inside the invariant set $\mathcal{S}$ of the system:*
$(q(t_0), \dot{q}(t_0)) \in \mathcal{S} \subset \mathcal{V} \subset \mathcal{F}$.

Determining whether the state of a robot is inside $\mathcal{S}$ of the employed control law can be achieved by simulating the constrained forward dynamics (1.8)-(1.12) for an infinitely long duration. This leads us directly to the Model Predictive Control approach. Here, at each iteration the evolution of the current state is simulated forward. An optimality criterion serves to compute the control input, and, what is more important with respect to the above stability definition, hard constraints on the state and the control can be considered to respect the limitations of the system.

$$
\begin{aligned}
\text{minimize} \quad & \int_{t_0}^{t_f} \mathcal{L}(t)dt \quad + \quad \Phi(t_f) \\
\text{subject to} \quad & \\
& T_2(q, \dot{q}, \ddot{q}) \quad = \quad \Lambda_2(q, f) \\
& (q, \dot{q}) \quad \in \quad \mathcal{K} \\
& q \quad \in \quad \mathcal{G} \\
& f_n \quad \geq \quad 0 \\
& \|f_t\| \quad \leq \quad f_t^{max}(f_n).
\end{aligned}
\tag{1.13}
$$

If now the predicted time period spans to infinity and no model plant mismatch is present, we can state that the robot is stable if it will stay within the comprehensive set of feasibility conditions for the entire preview period, i.e. a solution to the above program exists $\forall t \in [t_0, \infty]$.

Sufficient condition for stability. *A state* $(q(t_0), \dot{q}(t_0))$ *is stable, in the sense of Definition 1, if, starting from this state, the system will stay inside* $\mathcal{F}$*:* $(q(t), \dot{q}(t)) \in \mathcal{F}\ \forall t \geq t_0$.

However, infinite prediction is in general computationally not tractable, and the length of the prediction horizon has to be reduced, entailing the need of measures to re-establish the properties of an infinitely long horizon. But before addressing the implications of a finite prediction horizon on the stability, let us resume the discussion of the employed model and have a closer look at the equations of motion.

1.2.2 Simplifying the model

As shown in Section 1.1, the second part of Equation (1.2) plays a dominant role in the control of motions. Its basic interpretation is that, in accordance with Newton's laws of motion, the change of the system's momentum is proportional to the efforts applied at the contact points. The same principle is expressed by the following relations between the rate of change of the linear and angular momenta of rigid bodies, and the external forces and torques:

$$\sum_k m_k(\ddot{c}_k - \vec{g}) = \sum_k f_k^c \tag{1.14}$$

$$\sum_k c_k \times m_k(\ddot{c}_k - \vec{g}) + R_k I_k \dot{\omega}_k = \sum_k c_k \times f_k^c + R_k \tau_k, \tag{1.15}$$

where c_k is the position of the k^{th} rigid body of m_k mass, R_k its orientation, and ω_k its angular velocity in an inertial frame of reference. The force acting through c_k and the torque vectors are denoted by f_k^c and τ_k, the constant gravitational acceleration vector as $\vec{g}$.

These so-called Newton-Euler equations allow for efficient solutions to the direct and the inverse dynamics problem (cf. [Siciliano et al., 2009]). A closer look at the Newton equation (1.14) reveals a big advantage of this formalism

for our purpose. We can comprise the weighted sum of particle accelerations to the acceleration $\ddot{c}$ of the Center of Mass (CoM), yielding:

$$m(\ddot{c} - \vec{g}) = \sum_k f_k^c. \tag{1.16}$$

The CoM has long been identified as a crucial feature of the human body and is used to this day as the main indicator of human gait characteristics. Although the CoM has no direct physiological meaning, evidence suggests that it plays an important role in the control of human motions [Scholz and Schöner, 1999]. As for robotics, the above change of variables allows to greatly reduce the complexity of the equations of motion and constitutes therefore the core abstraction of most related approaches (cf. Section 1.3).

Yet this change of variable cannot be done to an equal extent for the non-holonomic Euler equation (1.18). Body motions can generate angular momenta without affecting the state of the CoM. But whereas the CoM's linear momentum is of great importance for walking, the role of these rotational momenta is somewhat less clear. By significantly varying angular momentum around the Center of Mass, humans are able to modulate the reaction forces to increase maneuverability and balance [Hofmann et al., 2009]. However, it is not yet clear what rotational momenta should be generated for walking. Deciding the angular momentum around the CoM decides little about the movements that the system is going to realize [Wieber, 2005], and evidence shows that its value is small during nominal walking [Popovic et al., 2004] [Herr and Popovic, 2008]. For these reasons, we will neglect in the following the rotational momenta and focus on the control of translational CoM motions only:

$$m(\ddot{c} - \vec{g}) = \sum_k f_k^c \tag{1.17}$$

$$mc \times (\ddot{c} - \vec{g}) = \sum_k c_k \times f_k^c + R_k \tau_k. \tag{1.18}$$

Contact forces f_k^c and the gravity force $m\vec{g}$ usually act on a limited set of bodies, so that the consideration of all body centers c_k on the right hand side of the Newtown-Euler equations is not necessary for the control of external angular momenta. Additionally, the external forces and torques f_k^c and τ_k can equally be represented by contact forces f_k that act on the robot through a

limited set of contact points p_k, yielding:

$$m(\ddot{c} - \vec{g}) = \sum_k f_k \tag{1.19}$$

$$mc \times (\ddot{c} - \vec{g}) = \sum_k p_k \times f_k. \tag{1.20}$$

1.2.3 Finite prediction horizon

Still, even with the system simplified in the way described above, an infinite prediction horizon is likely to be computationally not feasible, hence a shortened horizon has to be used. A consequence is that the principle of the optimality of subarcs [Bellman, 1957] does not apply anymore. Even if no model plant mismatch and no disturbances are present, the closed-loop behavior differs then from the open-loop one. This means in general that the performance objective $\mathcal{L}$ might not be achieved by the closed-loop system. What carries more weight, however, is that the closed-loop system might become unstable. Special attention has therefore to be payed to guarantee stability when employing finite prediction horizons.

Here, we review some central ideas on how stability can be established for the nominal case, i.e. without noise or model-plant mismatch. Being only a short summary, more detailed reviews can be found in [Mayne et al., 2000] [Morari and Lee, 1999] [Nicolao et al., 2000] [Primbs et al., 1997].

Terminal equality constraint: A straightforward method to give the solution properties of an infinite horizon consists in employing an equality constraint on the state of the system at the end of the prediction horizon. Then, if the terminal constraint has been fulfilled at the end of the preview period, feasibility is assured for all subsequent time periods, and the above definition of stability applies.

A drawback of this method is that such constraints might limit the flexibility of the controller to achieve secondary goals. Furthermore, the system might not be able to achieve the desired state inside the previewed period such that an extension of the horizon can become necessary.

Terminal inequalities and cost: To alleviate the demand on the system, we can think of expanding the terminal equality constraint to an inequality

constrained region. But then, such a terminal region might not be enough to guarantee stability. A suitable terminal cost Φ can be used to achieve properties of an infinite cost and to guarantee feasibility. This cost function is thereby usually not motivated by a physical restriction but has the sole purpose of enforcing stability. Modifying the original control goal, it can influence the performance of the system and has to be used cautiously. The combination of both the terminal region and cost is the most popular approach.

Contraction constraint: Instead of applying a terminal constraint on the final state, another important stream in the literature consists in employing a contractive constraint on the state in order to force monotonicity.

1.3 Related work

All aspects of the trajectory generation problem that have been discussed above, i.e. feasibility, model simplification, and stability, have been addressed in some way in all of the developed schemes for online walking control. Anticipation is thereby the key aspect of most approaches. It seems to be a common understanding that a predicted evolution of the system in some future is required to achieve stable walking.

The consideration of solely the Center of Mass for walking control has, to the knowledge of the author, been first made in [Kajita and Tani, 1991] showing that the motion information provided by this feature is sufficient enough to generate stable walking motions. The computational simplicity of point-mass models, also referred to as Inverted Pendulum or Cart Table models, has since then led to their dominance in online walking control [Kajita et al., 2003] [Nishiwaki and Kagami, 2009] [Morisawa et al., 2006] [Tajima et al., 2009] [Harada et al., 2004] [Takenaka et al., 2009].

In all of these approaches, dynamic feasibility is enforced by reference patterns for the reaction torque that are modified online by heuristic adaptation schemes. Stability is in most cases assured by equality conditions on the state of the CoM at the end of the control horizon, supposing either an immobile position or a cyclically continuing trajectory [Kajita et al., 2001] [Morisawa et al., 2006] [Takenaka et al., 2009] [Harada et al., 2004] [Nishiwaki et al.,

2002a].

1.4 Conclusion

More than for classical robots, limitations on the realizability of motions play a crucial role in the control of biped robots since respecting these constraints is necessary for avoiding a fall. Model Predictive Control is apt for the control of constrained systems, but the complexity of the robots dynamics can deprive from generating feasible motions online. This complexity can be reduced significantly by focussing on features that are essential for locomotion: the Center of Mass and its interaction with the contact forces, once contact is established. The computational load can be reduced furthermore by limiting the length of the horizon, but measures have to be taken to assure stability. The motion generation scheme that has been established in this chapter will serve in the following for the development of practical control schemes.

1.5 Résumé

Encore plus que pour les robots classiques, les limitations sur la viabilité des mouvements jouent un rôle crucial pour le contrôle des robots bipèdes car respecter ces contraintes est nécessaire pour l'évitement d'une chute. La commande prédictive est appropriée pour le contrôle des systèmes sous contraintes, mais la complexité de la dynamique des robots peut nous priver de générer des mouvements viables en ligne. L'étendu de cette complexité peut être réduite de manière significative si l'on se focalise sur des éléments qui sont essentiels pour la locomotion: le Centre de Masse et son interaction avec les forces de contact, une fois le contact établi. Le charge de calcul peut être réduit davantage en limitant la longueur de l'horizon, mais des mesures doivent être prisent pour assurer la stabilité. Le générateur des mouvements qui a été établi dans ce chapitre va servir dans la suite pour le développement des contrôleurs pratiques.

Chapter 2

Linear MPC

In linear MPC (LMPC), the control of a constrained system can be cast as a convex quadratic program (QP) subject to linear constraints and therefore solved efficiently. This is one of the main reasons for the dominance of LMPC in the process industry. Due to recent numerical and theoretical advances in optimization and control, linear MPC has become tractable in robotics where processes are faster and successful applications still rare.

This chapter provides an extensive coverage of LMPC for walking control. We will extract a globally linear system out of the general framework established in the previous chapter and discuss possible quadratic programming formulations. The focus of the chapter will be given to a scheme that allows for velocity control and free foot placement. Simulation and application results will serve us to evaluate the performance of this scheme with respect to reactivity, robustness and flexibility. Solutions to the numerical and stability issues that are characteristic to Model Predictive Control will be discussed and applied to this scheme.

2.1 Linear walking control problem

In order to cast the general optimization program (1.13) as a QP subject to linear constraints, we will discuss in this section the linearization of the dynamics (1.8)-(1.12). The walking scheme that we will formulate by the end of the section can be found in [Wieber, 2006], although under a formulation that is less advantageous from the computational point of view.

2.1.1 Linearized motion equations

In order to formulate (1.13) as a QP, the following linear system subject to linear state and control constraints has to be found:

$$\begin{aligned} \frac{d}{dt}s = &\quad As + Bu, \\ \text{subject to:} &\quad Es + Fu \leq \epsilon, \end{aligned} \tag{2.1}$$

with s and u the state and control vectors of the system.

Let us focus first on the Newton-Euler equations (1.19) and (1.20) for one single particle, the Center of Mass c, and multiple contact points p_k:

$$m \begin{pmatrix} \ddot{c}^x \\ \ddot{c}^y \\ \ddot{c}^z - g \\ c^x(\ddot{c}^z - g) - c^z\ddot{c}^x \\ c^y(\ddot{c}^z - g) - c^z\ddot{c}^y \\ c^y\ddot{c}^x - c^x\ddot{c}^y \end{pmatrix} = \sum_k \begin{pmatrix} f_k^x \\ f_k^y \\ f_k^z \\ p_k^x f_k^z - p_k^z f_k^x \\ p_k^y f_k^z - p_k^z f_k^y \\ p_k^y f_k^x - p_k^x f_k^y \end{pmatrix}, \tag{2.2}$$

with g the gravitational acceleration constant of $-9.81\ \mathrm{m/s^2}$, m the total mass of the robot, and f_k the force applied at the contact point p_k. Being nonlinear and coupled, this condition cannot be directly considered inside (2.1).

Let us suppose now that:

1. The motions of the point mass and the positions of the contact points are bound to horizontal planes (Figure 2.1) such that:

$$c^z = h, \tag{2.3}$$

$$p_k^z = 0. \tag{2.4}$$

2. No sliding of the contact points can occur:

$$\|f_k^x\| \leq \infty, \|f_k^y\| \leq \infty. \tag{2.5}$$

The horizontal forces f_k^x, f_k^y can then take arbitrary values, and equations (2.2) reduce to the following decoupled linear ordinary differential (ODE) system:

$$m \begin{pmatrix} -g \\ -h\ddot{c}^x - c^x g \\ -c^y g - h\ddot{c}^y \end{pmatrix} = \sum_k \begin{pmatrix} f_k^z \\ p_k^x f_k^z \\ p_k^y f_k^z \end{pmatrix}. \tag{2.6}$$

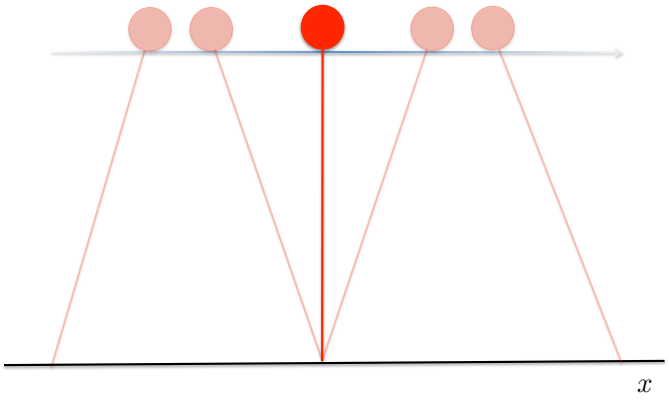


Figure 2.1: Planar case: The motions of the point mass and the positions of the feet are constrained to horizontal planes.

The above two suppositions build the basis of many related linear generation schemes. The main difference with respect to our formulation is that in these works the original model is the inverted pendulum model (e.g. [Kajita et al., 2003]). In this thesis, we derive the simplified model from a system of rigid bodies, which will allow us for a smooth extension to multi-body formulations in Chapter 3 and the general three dimensional case in Chapter 4.

For predefined contact points $p_k(t)$, we can rewrite equation (2.6) in form of the linear system (2.1):

$$\frac{d}{dt}\hat{c}^{x,y}(t) = A\hat{c}^{x,y}(t) + Bu, \tag{2.7}$$

with

$$\hat{c}^{x,y} = \begin{pmatrix} c^{x,y} & \dot{c}^{x,y} \end{pmatrix}^T, \tag{2.8}$$

and

$$u = \ddot{c}^{x,y} \text{ or } u = f^z, \tag{2.9}$$

where f^z comprises the vertical components of all contact forces f_k.

With the assumption that the contact points cannot slide, constraints (1.11) and (1.12) on the contact forces f_k reduce to unilateral constraints (1.11) on the vertical force components f_k^z only:

$$0 \leq f_k^z(u,t) := E_k^f \hat{c}^{x,y}(t) + F_k^f u. \tag{2.10}$$

2.1.2 Steering of the point-mass above predefined step positions

If we neglect for a moment the objective functions $\mathcal{L}$ and Φ together with the implementation details, we can formulate at this point a QP problem for the control of horizontal CoM motions $c^{x,y}(t)$:

$$\begin{aligned}
&\underset{f_k^z}{\text{minimize}} && \int_{t_0}^{t_f} \mathcal{L}(f_k^z, t)dt + \Phi(f_k^z, t_f) \\
&\text{subject to} && \\
&t \in [t_0, t_f] && 0 = \sum_k f_k^z + mg \\
&t \in [t_0, t_f] && \ddot{c}^{x,y}(t) = \frac{1}{h}\left(-c^{x,y}(t)g - \frac{1}{m}\sum_k p_k^{x,y}(t) f_k^z\right) \\
&\forall k && 0 \leq f_k^z,
\end{aligned} \tag{2.11}$$

where the horizontal positions $p_k^{x,y}$ of the contact points are fixed in advance, and the dynamic feasibility is assured by a set of linear equality constraints and bounds on the control variables f_k^z.

A functionally identical MPC scheme has proven in [Wieber, 2006] that stable walking trajectories can be obtained this way. The scheme above however is posed in a computationally more convenient form since the dynamic feasibility is assured by linear equality constraints and bounds on the decision variables of the QP instead of general inequality constraints. We will come back to this formulation in Chapter 4, but in the following, we will focus on a slightly different formulation.

2.2 Free foot placement

To this day, predefined foot steps are at the basis of most online walking control schemes. These approaches require, therefore, the consideration of adaptation strategies (foot step planners) to change support positions online. The heuristics that these adaptation schemes make use of, however, compromise the stability and reactivity of the robot. Several attempts have been made to allow for free foot placement inside the actual trajectory generation scheme, although again, not without using heuristic assumptions [Harada

et al., 2004] [Morisawa et al., 2007] [Buschmann, 2010].

We can see that the right hand side of equation (2.6) is still nonlinear for variable p_k, which means that the above linear control of the particle is possible only for fixed foot positions. Now, we can replace the product $p_k^{x,y} f_k^z$ by a change of abstraction from the vertical reaction forces f_k^z to the horizontal reaction torques $\tau_k^{x,y}$, applied at the contact points:

$$\tau_k^{y,x} = p_k^{x,y} f_k^z, \tag{2.12}$$

or the Center of Pressure (CoP):

$$z^{x,y} = \frac{\sum_k p_k^{x,y} f_k^z}{\sum_k f_k^z} = -\frac{\sum_k p_k^{x,y} f_k^z}{mg}. \tag{2.13}$$

This latter gives us a convenient geometrical interpretation for the violation of the unilateral constraints (1.11): If the CoP lies on the border of the convex hull spanned at the contact points, the robot might lose foothold and fulfill a rotation around the corresponding edge of the foot. Dynamic feasibility can thus be assured by a set of inequality conditions on the position z of the CoP that restrict it to the interior of this convex hull [Wieber, 2006]:

$$E_z(p^{x,y}, p^\theta) z^{x,y} \leq \epsilon_z, \tag{2.14}$$

where $p^{x,y}$ and p^θ denote the positions and orientations of the supporting feet in the transverse plane. These constraints are linear with respect to the CoP and nonlinear with respect to the orientations of the feet, as well as their positions during the double support phases, when both feet are in contact.

A big advantage of the above geometrical interpretation of condition (1.11) has been shown in [Diedam et al., 2008], where a slight modification of the above constraints allowed to modify the foot placements online and in permanent accordance with the feasibility requirements. The idea there is to determine the positions $p^{x,y}$ of future single support phases, i.e. the periods of time $t \in SS \subset [t_0, t_f]$ when the robot is supported by one single foot, by additional parameters in the control vector u such that:

$$E_z(p^\theta(t))(z^{x,y}(u,t) - p^{x,y}(u,t)) \leq \epsilon_z \quad \forall t \in SS. \tag{2.15}$$

Although the constraints on the CoP are generally nonlinear with respect to the positions and orientations of the feet, they are linear with respect to the

foot positions during single support. Hence, by deciding (only) the orientations of the feet in advance and by considering the double support phases implicitly (cf. Section 2.5 for the implications), free foot placement can be achieved within the following linear MPC control scheme:

$$\underset{u}{\text{minimize}} \qquad \int_{t_0}^{t_f} \mathcal{L}(u,t)dt + \Phi(u,t_f) \tag{2.16}$$

subject to

$$t \in [t_0, t_f] \qquad \ddot{c}^{x,y}(u,t) = \frac{g}{h}\left(z^{x,y}(u,t) - c^{x,y}(u,t)\right) \tag{2.17}$$

$$t \in SS \qquad E_z(p^\theta(t))(z^{x,y}(u,t) - p^{x,y}(u,t)) \leq \epsilon_z \tag{2.18}$$

2.2.1 Kinematic feasibility

To ensure that the generated foot step placements are feasible with respect to the geometric and kinematic limitations (1.5) and (1.6), we can derive approximations that can be expressed in the form of linear constraints. We can derive for example linear bounds on the positions of the feet, one with respect to the other, with minimum and maximum values preventing collision on one side and over-stretching of the legs on the other side (cf. Appendix A):

$$E_p(p^\theta(t))p^{x,y}(u,t) \leq \epsilon_p \quad \forall t \in SS. \tag{2.19}$$

Concerning maximum joint speed, simple bounds on the position of the next foot step depending on the current position of the foot in the air $p^t_{air}(t_0)$ and a simple Cartesian maximum speed v_{max} can be sufficient:

$$\|p^{x,y}(u,t_d) - p^{x,y}_{air}(t_0)\| \leq (t_d - t_0)v_{max}, \tag{2.20}$$

with t_d being the instant at which the foot in the air is touching the ground.

The LMPC scheme becomes then:

$$\underset{u}{\text{minimize}} \qquad \int_{t_0}^{t_f} \mathcal{L}(u,t)dt + \Phi(u,t_f) \tag{2.21}$$

subject to

$$t \in [t_0, t_f] \qquad \ddot{c}^{x,y}(u,t) = \frac{g}{h}\left(z^{x,y}(u,t) - c^{x,y}(u,t)\right) \tag{2.22}$$

$$t \in SS \qquad E_z(p^\theta(t))(z^{x,y}(u,t) - p^{x,y}(u,t)) \leq \epsilon_z \tag{2.23}$$

$$t \in SS \qquad E_p(p^\theta(t))p^{x,y}(u,t) \leq \epsilon_p \tag{2.24}$$

$$t = t_d \qquad \|p^{x,y}(u,t) - p^{x,y}_{air}(t_0)\| \leq (t - t_0)v_{max}. \tag{2.25}$$

2.3 Tracking a velocity reference

The free foot placement introduced in the previous section allows to overcome the main obstacle on the way to truly autonomous robots. Due to the ability to adapt the foot placement to feasibility requirements of generated trajectories, be it a contact wrench or a kinematic limitation, trajectory generation can be done exclusively on a more elevated level than foot position planning. The robot can for example be controlled through a desired CoM trajectory or realize a desired contact force other than a CoP reference.

In this section, we are focusing on the control through a given displacement velocity. Since the chosen objective $\mathcal{L}$ together with the constraints have the greatest influence on the performance of the control scheme, we are going to examine the behavior of the system against the background of this two elements, and we will leave the implementation details to subsequent sections.

2.3.1 Minimization of instantaneous velocities.

In [Wieber, 2008], it has been shown that the minimization of any derivative of the state has a stabilizing effect on the system. Together with the free-foot placement introduced in the previous section, we can accordingly think of minimizing the velocity of the CoM with respect to a desired speed:

$$\mathcal{L}(u,t) = \left(\dot{c}^{x,y}(u,t) - \dot{c}^{x,y}_{ref}(t)\right)^2. \tag{2.26}$$

We will examine the performance of the associated control scheme with the help of the following scenario (cf. Figure 2.2, left):
The robot starts from rest in double support, and walks continuously for 20 s, making a step regularly every 0.8 s. The velocity reference $\dot{c}^{x,y}_{ref}$ is switched to 0.3 m/s at the beginning of the first step. To simulate a push, an instantaneous impulse to the left is applied on the CoM 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 m/s on the right. In the beginning of step 15, at time $t = 12$ s, it is switched back to 0.3 m/s forward and back to zero in the middle of step 22, at time $t = 18$ s.

Figure 2.2 shows that the MPC scheme, consisting of the above minimization of *instantaneous* velocities subject to constraints (2.22)-(2.25), manages

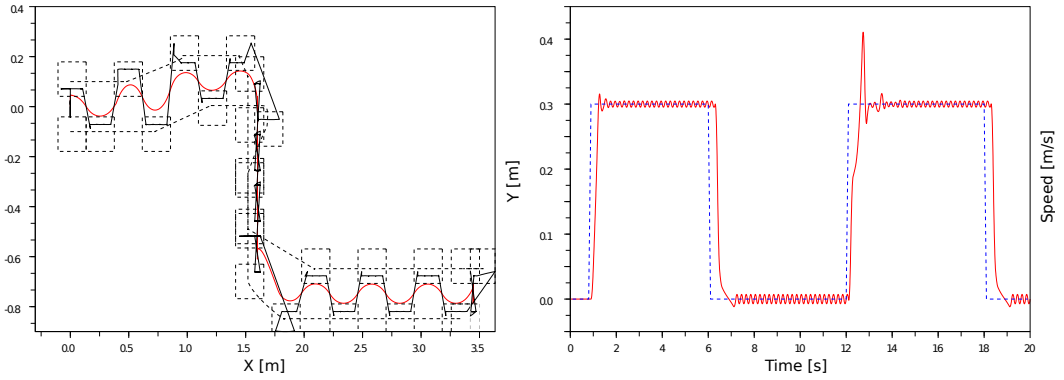


Figure 2.2: Walking scenario obtained with our linear MPC scheme. Left: Foot step placement and ankle motion (dashed grey), Center of Pressure (solid black) and Center of Mass (solid red) positions of the approximate model (2.6). Right: Forward speed of the CoM (solid red) and reference speed (dashed blue).

to perfectly realize the 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 that the CoP always lies 3 cm inside the true boundaries of the support polygon. In fact, the position of the CoP plotted here corresponds to the approximate model, but (as for the HRP-2) the difference with the real CoP is usually less than 2 cm so this motion appears to be completely safe.

Still, this motion is not completely satisfactory. The trajectory of the CoP looks chaotic sometimes, which can lead to difficulties on a real robot. This even has an effect on the speed of the robot, which can be seen to oscillate around its reference value (Figure 2.2, right). In the approximate model (2.6), the position of the CoP appears to be related to the position and acceleration of the CoM, so minimizing the derivative of this acceleration, the jerk $\dddot{c}$, should smoothen the trajectory of the CoP and the speed of the CoM:

$$\mathcal{L}(u,t) = \ldots + \alpha \dddot{c}^{x,y}(u,t)^2. \tag{2.27}$$

We can observe in Figure 2.3 that it is indeed the case when introducing a gain $\alpha = 10^{-6}$.

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

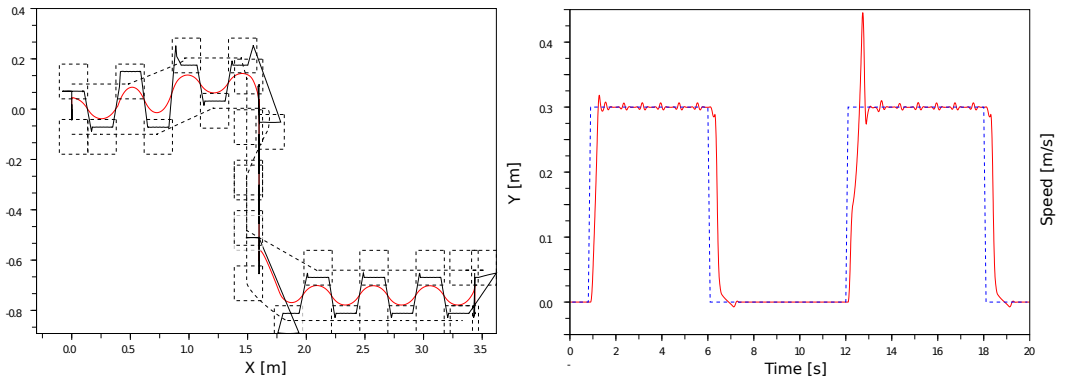


Figure 2.3: Same scenario as in Figure 2.2 but with smoothened trajectories due to (2.27). Left: Foot step placement and ankle motion (dashed grey), position of the Center of Pressure (solid black) and Center of Mass (solid red). Right: Forward speed of the CoM (solid red) and reference speed (dashed blue).

collision 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.

But once again, this motion is not completely satisfactory yet. We can observe in Figure 2.3 that during the lateral walk the CoP is positioned at the front of the feet. Although perfectly correct from the point of view of the dynamics of the system, this position induces difficulties when a perturbation or a change of desired velocity needs to be dealt with. This can be seen at the end of the lateral motion, at time $t = 12$ s; a delay and an overshoot can be observed in Figure 2.3, much more than in the similar situation at time $t = 0.8$ s. To increase the distance of the CoP to the borders of the feet during lateral walk we enforce the centering of the CoP:

$$\mathcal{L}(u,t) = \ldots + \beta \left(z^{x,y}(u,t) - p^{x,y}(u,t)\right)^2 \quad \forall t \in SS. \tag{2.28}$$

A gain $\beta = 10^{-6}$ is enough to solve this problem (cf. Figure 2.4) since no other objective interferes with the positioning of the feet during lateral walking. A

comparison between Figure 2.3 and Figure 2.4 shows that during the lateral motion only the foot step placements have changed, not the trajectories of the CoM and CoP. The interpretation of this term can thus be made as follows: The position of the CoP is decided with respect to the desired motion of the CoM, and the foot step placement is decided then accordingly, centered around the CoP when possible, here during the lateral motion.

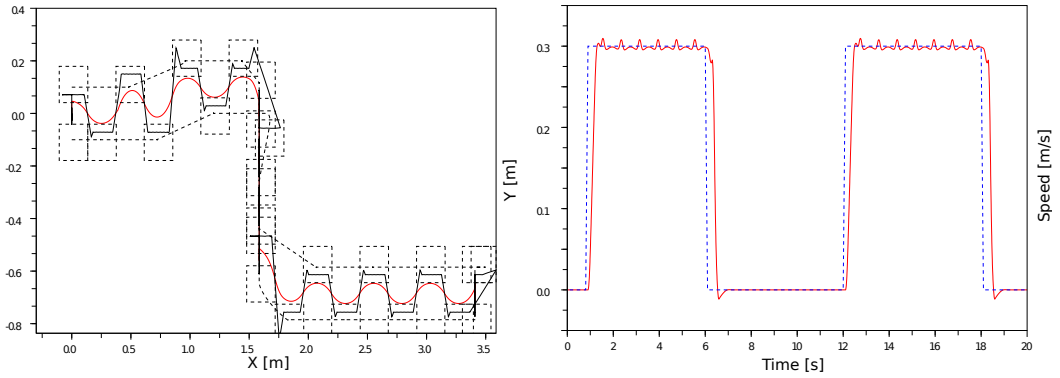


Figure 2.4: Same scenario as in Figure 2.2 but with centered CoP due to (2.28). Left: Foot step placement and ankle motion (dashed grey), position of the Center of Pressure (solid black) and Center of Mass (solid red). Right: Forward speed of the CoM (solid red) and reference speed (dashed blue).

2.3.2 Keeping a mean velocity

Having a look at the lateral speed of the CoM in Figure 2.5 (left), we can observe that because of the unavoidable lateral sway motion, only a mean desired speed of the CoM can be obtained. But having a more precise look at the mean speed over prediction horizons, which appears in black on this figure, we can see that it is very different from the reference speed (in blue) during the lateral motion, between times t = 6 s and t = 12 s. The objective (2.26) is regulating the instantaneous speed $\dot{c}$ to the desired value, but during lateral motion, one step out of two must be realized in a direction opposite to the desired one since crossing legs is not possible for this robot. So one step out of two, the instantaneous speed can be perfectly regulated to the desired value while, one step of two, it can reach only a far lower value, giving in the end a mean speed of about 2/3 of the desired one. An option could be to regulate

instead the mean speed over the duration $2T_s$ of two steps:

$$\mathcal{L}(u,t) = \ldots + \gamma \left(\frac{c^{x,y}(u, t+2T_s) - c^{x,y}(u,t)}{2T_s} - \dot{c}^{x,y}_{ref}(t) \right)^2 . \tag{2.29}$$

We can see in Figure 2.5 that the resulting mean speed corresponds to the

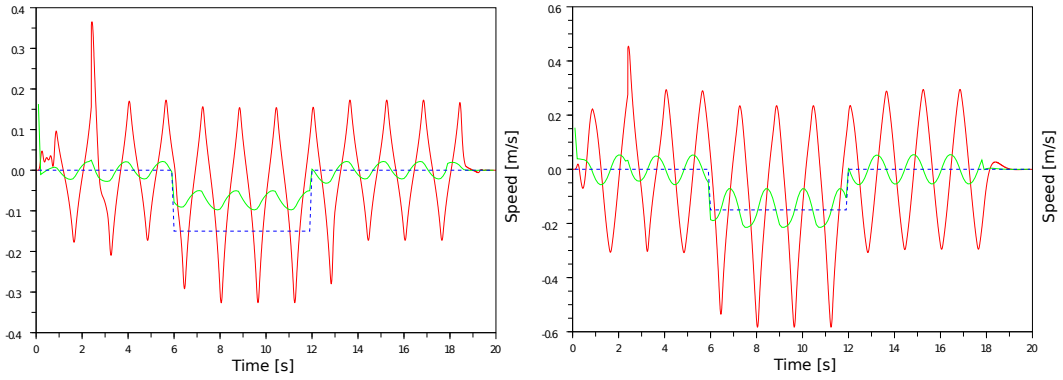


Figure 2.5: Lateral speed of the CoM (solid red), mean value of this speed over prediction horizons (solid green) and reference speed (dashed blue) for the motion of Figure 2.4. Left: Regulation of instantaneous velocities (2.26). Right: Regulation of the mean velocity (2.29).

reference speed, but this variant also brings two difficulties. The first one is that working with a mean speed over two steps instead of an instantaneous speed requires that the prediction horizon has a duration of at least four steps (3.2 s here), what can imply longer computation times. The second one is that regulating only the mean speed generates a stronger sway motion, that can be seen here when comparing the amplitude of the oscillations in Figure 2.5. And this can have undesirable effects such as inducing a higher sensibility to perturbations. However, an infinity of possibilities exist to obtain the desired mean velocity. Reducing the sway motions is possible by minimizing the instantaneous velocity (2.26) term in the null space of this objective function.

2.3.3 Reactivity due to inequality constraints

Figure 2.6 shows a comparison between the forward speed that we obtained with the above scheme and the speed that is obtained with the scheme proposed in [Morisawa et al., 2007] when the desired speed is changed from 0 to 0.3 m/s (steps of length 24 cm every 0.8 s) at the beginning of a step. The first obvious

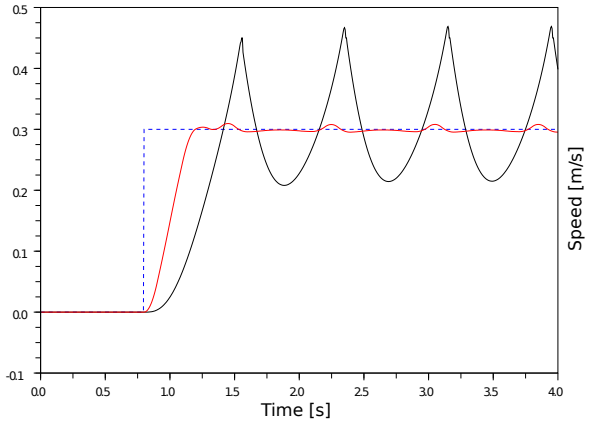


Figure 2.6: Comparison between the forward speed obtained with our method (solid red) and with the method proposed in [Morisawa et al., 2007] (solid black) when the reference speed (dashed blue) is changed from 0 to 0.3 ms^{-1} at the beginning of a step.

observation is that, with the scheme of this section, the speed converges nearly perfectly to the desired value whereas, with the scheme of [Morisawa et al., 2007], only the mean value of the speed is obtained. However, the scheme described there was not designed for such a convergence, so this observation is not very meaningful. The approximately quadratic shape of the speed is a classical result of continuously positioning the CoP in the middle of the feet, whereas with our scheme the CoP moves continuously forward under the feet. More interesting is the observation that the speed of the CoM rises nearly twice more quickly. This is noteworthy since the approach in [Morisawa et al., 2007] has been proposed for fast reaction of the robot. In this sense, the motion of the robot appears to react almost twice faster with our trajectory generator.

These observations are a direct consequence of the inequality constraints that we employ. The inequalities on the Center of Pressure relieve us from the requirement of using a reference and allow for more 'freedom' in the choice of an objective that suits the desired goal. Since the cost function (2.26) can be interpreted as attaining the desired speed as fast as possible, the consequence is an improved reactivity.

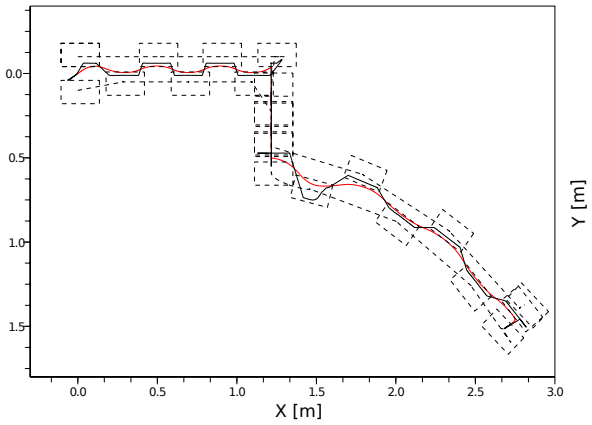


Figure 2.7: Walking motions with a rotation phase. Foot step placement and ankle motion (dashed grey), position of the Center of Pressure (solid black) and Center of Mass (solid red).

2.4 Tracking a rotational velocity reference

Considering also the rotations of the trunk $c^\theta(t)$ around the vertical axis can be done straightforwardly by employing the same linear relationships (2.1) as for the translational motions. The rotation of the trunk can thus be smoothly integrated into the instantaneous-velocity term (2.26). However, as mentioned in Section 2.2, including the orientations p^θ of the feet in contact with the ground as decision variables will give rise to nonlinearities in (2.15) and (2.19). To keep the linear form of the original QP, we choose here to calculate the orientations of the feet and the trunk prior to the generation of translational motions. When rotating the trunk, the robot has to take care of appropriate orientations of the feet. One intuitive solution consists in aligning the feet with the rotating trunk as much as possible for the whole preview period:

$$\underset{u}{\text{minimize}} \ \left(\dot{c}^\theta(u,t) - \dot{c}^{\theta-ref}(t)\right)^2 + \alpha \left(p^\theta(u,t) - c^\theta(u,t)\right)^2. \tag{2.30}$$

Although aligning the feet with the trunk reduces the probability of a violation of kinematic constraints, guaranteeing that the generated foot and trunk trajectories respect the limitations of the concerned joints requires the consid-

eration of a set of constraints on the kinematics of the robot:

$$||p^{\theta}(u,t) - c^{\theta}(u,t)|| < \Delta\theta_{max}, \tag{2.31}$$

$$||\dot{p}^{\theta}(u,t) - \dot{c}^{\theta}(u,t)|| < \Delta\dot{\theta}_{max}, \tag{2.32}$$

$$||\ddot{p}^{\theta}(u,t) - \ddot{c}^{\theta}(u,t)|| < \Delta\ddot{\theta}_{max}. \tag{2.33}$$

2.5 Real-time implementation

Before being able to solve efficiently the optimization problems discussed above, their discretized form has to be obtained. In this section, we are going to discuss the discretization of system (2.1), laying emphasis on the resulting loss of information in between sampled instants and strategies to reduce the computational time. The resulting discrete trajectory generation scheme is described in more detail in Appendix A.

2.5.1 Discretized system

We seek to formulate the following discrete-time model:

$$\begin{aligned} c_{i+1} = \quad & \bar{A}\hat{c}_i + \bar{B}u_i, \\ \text{subject to:} \quad & \bar{E}\hat{c}_i + \bar{F}u_i \leq \bar{\epsilon}. \end{aligned} \tag{2.34}$$

To assure feasibility, the sampling rate has to be chosen in a way that equilibrates the computational complexity against the maximal possible deviation in between the sampling instants, that can be approximated by comparing the profile of the discrete system to an assumed worst-case evolution. A security margin can then be considered to assure feasibility for a given sampling rate. It appears that satisfying the feasibility constraints (3.5)-(**??**) only every 100 ms is largely enough for generating realizable motions. An important observation is that at transition times between single and double support phases the constraints of both single and double support apply, but those of single support are the most restrictive and are therefore sufficient on their own. We choose here therefore to satisfy the constraint on the position of the CoP with a period T in between chosen to be strictly equal to the length of the double support phases ($t_i \in DS \subset [t_0, t_f]$) (0.1 s here, with single support periods of 0.7 s) so

that no sampling time falls strictly inside them. This way, we end up having to consider the constraints only during single support periods ($t_i \in SS$).

The results in Section 2.3 have been obtained with a trivial piecewise constant jerk ($\dddot{c} = const \ \ \forall t \in [t_i, t_{i+1}[$) parametrization and a sampling rate of 100 ms (cf. Appendix A for further details):

$$\bar{A} = \begin{pmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{pmatrix}, \bar{B} = \begin{pmatrix} \frac{T^3}{6} \\ \frac{T^2}{2} \\ T \end{pmatrix}. \tag{2.35}$$

This parametrization has the advantage of a constant sampling gap, but it leads to the number of decision variables being proportional to the number of samplings. With a preview period of 1.6 s in the above examples, the minimal number of variables arises at 32 and the minimal number of constraints at 64. We know for example from [Lau et al., 2009] that the computation time of a QP solver rises polynomially with the number of variables and constraints.

To reduce the dimension of the optimization problem, a strategy might be a parametrization map, as can be found in [Alamir and Murilo, 2008]. An important observation is that instants that are further away in the future are of less influence on the control than closer ones. The sampling frequency can therefore possibly be reduced gradually over the preview window without considerable impact on the result.

2.5.2 Warm-start

As well as the number of variables, the number of inequality constraints, or more precisely, the number of active inequality constraints, has a significant influence on the resolution time [Lau et al., 2009]. The most prominent approaches for the resolution of inequality constrained QP problems are the *active set* and the *interior point* methods. Fast solvers exist for both of the approaches, but the performance can vary drastically depending on the structure of the optimization problems. Whereas interior point methods are relatively insensitive to the number of active inequality constraints, active set methods gain more from their initializations.

In Model Predictive Control, optimization problems are solved in sequences. Depending on the sampling rate, subsequent problems share a more or less

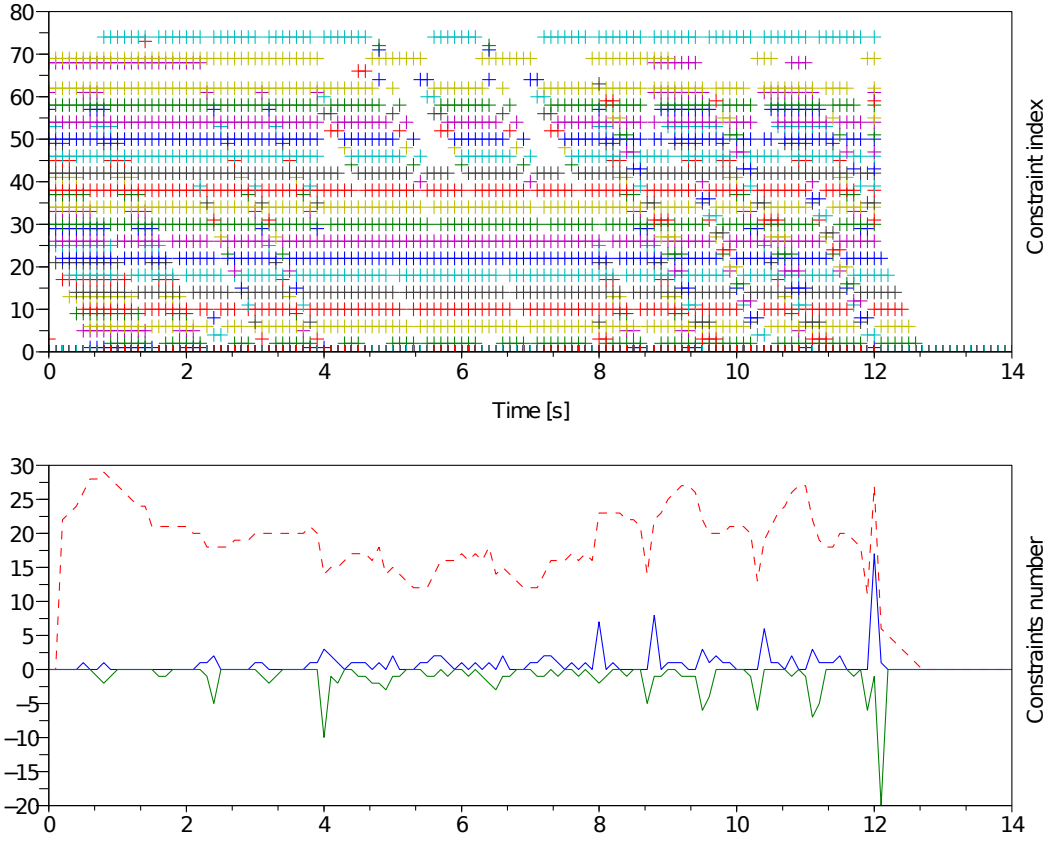


Figure 2.8: Active inequality constraints for a walking scenario similar to the one in Figure 2.7. Upper graph: Activation sequence of inequality constraints. Graph below: Total number of active (dashed red), activated (positive, blue) and deactivated constraints (negative, green) at an instant.

similar structure. The knowledge of this structure can be exploited to *warm-start* the QP solver. The upper graph in Figure 2.8 shows the evolution of the active constraints for a scenario similar to the one in Figure 2.7. We can observe a relatively structured evolution of 'sliding' active CoP constraints (numbers 1-64) and 'static' active foot positioning constraints (numbers 65-80). The lower part of Figure 2.8 shows that, by exploiting this structure, the number of constraints that have to be activated can be reduced significantly. Since the warm-started active set solver has then a reduced combinatorial research phase, we can expect a reduction in computational time [1].

1. In praxis, the computation time could be reduced by 50% in average.

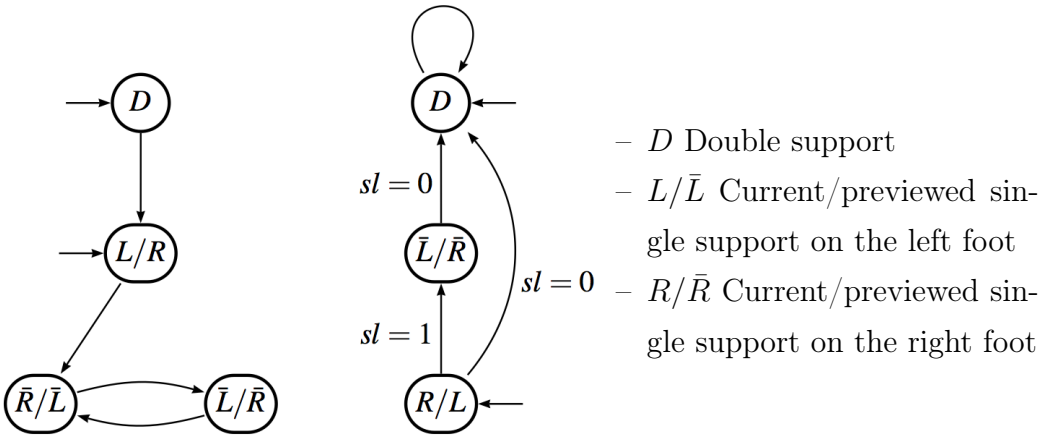


Figure 2.9: Finite state logic for the starting (left) and the stopping phase (right): Every state $D, L, R, \bar{L}, \bar{R}$ corresponds to a different set of constraints, depending on the support phase of the robot at a sampled instant. The parameter sl has been introduced to allow for variable number of steps before stopping in a double support phase.

2.5.3 Gait coordination

In order to avoid nonlinearities, the MPC scheme introduced in this chapter takes into account only single supports. With the sampling synchronized with the contact transitions, this is valid during walking but restricting during a stop. To correct this deficiency we propose to stop only with the feet aligned in the frontal plane of the robot. Due to its rectangular form, the double support phase can then be smoothly integrated in the computation of the CoP constraints.

The assembling of the constraints requires for each instant of the preview period prior knowledge about the associated support phase. With the durations of the support phases fixed beforehand, this knowledge can be provided by the finite state machine (FSM) in Figure 2.5.3. Given a constant speed reference, the sequence of support phases is then defined only by the predefined lengths of the single and double support periods.

2.6 Stability

In the results above, stability has been enforced solely by the minimization of the CoM velocity with respect to a reference. Although sufficient for slower motions, divergence can occur for faster walking or bigger changes in the velocity reference. As mentioned in the previous chapter, stability can be guaranteed by giving the solution properties of an infinite horizon. The common approach of using a terminal constraint on the state of the particle, however, would strongly restrict the set of trajectories, and make the fulfillment of such a constraint difficult. The MPC approach, or more precisely, the possibility of considering inequality constraints, allows us to ease these difficulties. We can thus think of a set of terminal inequalities that limit the final position of the particle above the support polygon. A strongly weighted cost term $\Phi(t_f)$ can additionally be used to enforce zero velocity by the end of the horizon, thus determining the evolution of the state until infinite time.

A (maybe) more elegant, since less restricting, solution can be given with a glimpse at [Pratt et al., 2006]. The Capture Point introduced there gives an estimation of the position of the CoM at $t = \infty$, assuming an instantaneous change of the CoP position to the position of the capture point. By limiting this point to the interior of the support region, we can guarantee that an immobilization of the CoM is possible in infinite time.

2.7 Conclusion

Linear Model Predictive Control allows for fast control of humanoid robots. It is the explicit consideration of inequality constraints that permits to improve the reactivity and robustness and to enhance the level of abstraction such that displacement control only through a velocity reference becomes possible. The additional computational load that is due to the consideration of inequalities can be reduced by exploiting the similarities of subsequent optimization problems as well as by limiting the length and the accuracy of the preview period.

This chapter did not cover the entire potential of LMPC for motion control. Other control goals can be thought of such as following a desired CoM trajectory or applying a contact force. Obstacle avoidance can be achieved by inequality constraints on the positions of the controlled particles, and external forces can be considered for the interaction with the environment as well as the compensation of perturbations.

However, even if LMPC allows to improve the performance of the approach of movement control by means of simplified models, the scheme introduced in this chapter is still far from the flexibility of humans, and the resulting walking motions lack efficiency and naturalness. Strong assumptions have to be made to achieve linearity; in this chapter, the vertical positions of the Center of Mass and the contact points have been fixed to horizontal planes. Orientations have been computed prior to translational motions, and the duration of support phases fixed beforehand. In an attempt to (partly) overcome these deficiencies, the focus of the following two chapters will be kept on the generation of walking motions only.

2.8 Résumé

La commande prédictive linéaire permet le contrôle rapide des robots humanoïdes. C'est la prise en compte explicite des contraintes d'inégalités qui permet d'améliorer la réactivité et la robustesse, ainsi que augmenter le niveau d'abstraction tel que le contrôle du déplacement par uniquement des références en vitesse devient possible. Le temps de calculs supplémentaire qui est du à la considération des inégalités peut être réduit en exploitant des similarités des problèmes d'optimisation subséquentes ainsi qu'en limitant la longueur et la précision de la période de prédiction.

Ce chapitre ne couvre pas le potentiel complet de LMPC pour le contrôle des mouvements. D'autres objectives de contrôle sont possibles tels que suivre une trajectoire du CdM ou appliquer des forces de contact. L'évitement d'obstacles peut être atteint par des contraintes d'inégalités sur la position des particules contrôlés, et des forces externes peuvent être considérées pour assurer l'interaction avec l'environnement ainsi que pour compenser des perturbations.

Même si LMPC permet d'améliorer la performance de l'approche du contrôle des mouvements par moyen des modèles simplifiés, la loi de commande introduit dans ce chapitre est toujours loin de la flexibilité des humains, et mouvement de la marche manque de l'efficacité et du naturel. Des suppositions fortes doivent être fait pour atteindre linéarité; dans ce chapitre, le positions verticales du Centre de Masse et des points de contact ont été fixés apriori. Pour surmonter (partiellement) ces inconvénients, l'accent des chapitres suivants va être maintenu sur uniquement la génération des mouvements de la marche.

2.9 Appendix A: Control scheme in more details

2.9.1 Discrete dynamics

We consider trajectories of the CoM which have piecewise constant jerks $\dddot{c}^x$ and $\dddot{c}^y$ over time intervals of constant length T so that we can compute the state of the CoM at discrete times t_i with:

$$\hat{c}_{i+1}^{x,y} = \bar{A}\,\hat{c}_i^{x,y} + \bar{B}\,\dddot{c}^{x,y}(t_i), \tag{2.36}$$

with

$$\hat{c}_i^{x,y} = \begin{pmatrix} c^{x,y}(t_i) \\ \dot{c}^{x,y}(t_i) \\ \ddot{c}^{x,y}(t_i) \end{pmatrix}, \tag{2.37}$$

and

$$\bar{A} = \begin{pmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{pmatrix},\ \bar{B} = \begin{pmatrix} \frac{T^3}{6} \\ \frac{T^2}{2} \\ T \end{pmatrix}. \tag{2.38}$$

We consider furthermore the position $z^{x,y}$ of the CoP on the ground:

$$z_i^{x,y} = \begin{pmatrix} 1 & 0 & h/g \end{pmatrix} \hat{c}_i^{x,y}. \tag{2.39}$$

Using the dynamics (2.36) recursively, we can derive relationships between the jerk of the CoM, its position and velocity, and the position of the CoP over longer time intervals NT:

$$C_{i+1}^{x,y} = \begin{pmatrix} c_{i+1}^{x,y} \\ \vdots \\ c_{i+N}^{x,y} \end{pmatrix} = S_p \hat{c}_i^{x,y} + U_p \dddot{C}_i^{x,y}, \tag{2.40}$$

$$\dot{C}_{i+1}^{x,y} = \begin{pmatrix} \dot{c}_{i+1}^{x,y} \\ \vdots \\ \dot{c}_{i+N}^{x,y} \end{pmatrix} = S_v \hat{c}_i^{x,y} + U_v \dddot{C}_i^{x,y}, \tag{2.41}$$

$$Z_{i+1}^{x,y} = \begin{pmatrix} z_{i+1}^{x,y} \\ \vdots \\ z_{i+N}^{x,y} \end{pmatrix} = S_z \hat{c}_i^{x,y} + U_z \dddot{C}_i^{x,y}, \tag{2.42}$$

with

$$\dddot{C}_i = \begin{pmatrix} \dddot{c}_i \\ \vdots \\ \dddot{c}_{i+N-1} \end{pmatrix}. \tag{2.43}$$

The matrices S_p, S_v, $S_z \in \mathbb{R}^{N\times 3}$ and U_p, U_v, $U_z \in \mathbb{R}^{N\times N}$ introduced here follow directly from a recursive application of the dynamics (2.36):

$$S_p = \begin{pmatrix} 1 & T & \frac{T^2}{2} \\ \vdots & \vdots & \vdots \\ 1 & NT & N^2\frac{T^2}{2} \end{pmatrix}, \qquad U_p = \begin{pmatrix} \frac{T^3}{6} - T\frac{h}{g} & 0 & 0 \\ \vdots & \ddots & 0 \\ (1+3N+3N^2)\frac{T^3}{6} & \dots & \frac{T^3}{6} \end{pmatrix}, \tag{2.44}$$

$$S_v = \begin{pmatrix} 0 & 1 & T \\ \vdots & \vdots & \vdots \\ 0 & 1 & NT \end{pmatrix}, \qquad U_v = \begin{pmatrix} \frac{T^2}{2} & 0 & 0 \\ \vdots & \ddots & 0 \\ (1+2N)\frac{T^2}{2} & \dots & \frac{T^2}{2} \end{pmatrix}, \tag{2.45}$$

$$S_z = \begin{pmatrix} 1 & T & \frac{T^2}{2} - \frac{h}{g} \\ \vdots & \vdots & \vdots \\ 1 & NT & N^2\frac{T^2}{2} - \frac{h}{g} \end{pmatrix}, \quad U_z = \begin{pmatrix} \frac{T^3}{6} - T\frac{h}{g} & 0 & 0 \\ \vdots & \ddots & 0 \\ (1+3N+3N^2)\frac{T^3}{6} - T\frac{h}{g} & \dots & \frac{T^3}{6} - T\frac{h}{g} \end{pmatrix} \tag{2.46}$$

2.9.2 Optimization problem

The objective function is stated as follows:

$$\begin{aligned} \underset{\dddot{C}_i, \bar{P}_i}{\text{minimize}} \quad & \frac{1}{2}\left\|\dot{C}^x_{i+1} - \dot{C}^{x\_ref}_{i+1}\right\|^2 + \frac{1}{2}\left\|\dot{C}^y_{i+1} - \dot{C}^{y\_ref}_{i+1}\right\|^2 \\ & + \frac{\alpha}{2}\left\|\dddot{C}^x_i\right\|^2 + \frac{\alpha}{2}\left\|\dddot{C}^y_i\right\|^2 \\ & + \frac{\beta}{2}\left\|Z^x_{i+1} - Z^{x\_ref}_{i+1}\right\|^2 + \frac{\beta}{2}\left\|Z^y_{i+1} - Z^{y\_ref}_{i+1}\right\|^2 \\ & + \frac{\gamma}{2}\left\|EC^x_{i+1} - \dot{C}^{x\_ref}_{i+1}\right\|^2 + \frac{\gamma}{2}\left\|EC^y_{i+1} - \dot{C}^{y\_ref}_{i+1}\right\|^2, \end{aligned} \tag{2.47}$$

with the double diagonal matrix:

$$E := \begin{bmatrix} -I & 0 & I \end{bmatrix} / (2\tau_{step}), \tag{2.48}$$

computing the mean speed of the CoM over two steps out of C^x_{i+1} and C^y_{i+1}, with τ_{step} the duration of a step. The CoP reference vector is going through the centers of the current and the previewed support feet $\hat{p}_i$ and $\bar{P}_i$ due to:

$$Z^{x\_ref}_{i+1} = V\hat{p}^x_i + \bar{V}\bar{P}^x_i, \tag{2.49}$$

$$Z_{i+1}^{y\_ref} = V\hat{p}_i^y + \bar{V}\bar{P}_i^y, \tag{2.50}$$

with

$$V = \begin{pmatrix} 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \bar{V} = \begin{pmatrix} 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 1 & 0 \\ \vdots & \vdots \\ 1 & 0 \\ 0 & 1 \\ \vdots & \vdots \\ 0 & 1 \quad \ddots \end{pmatrix}, \tag{2.51}$$

assembled by the finite state machine introduced in Section 2.5. The ones in the vector $V \in \mathbb{R}^N$ and matrix $\bar{V} \in \mathbb{R}^{N\times m}$ simply indicate which sampling times t_i fall into which step, where sampling times correspond to rows and steps to columns, and therefore which foot position must be taken into account at what time.

Optimization problem (2.47) can be expressed as a canonical QP:

$$\underset{u_i}{\text{minimize}}\ \frac{1}{2}u_i^T Q_i u_i + r_i^T u_i, \tag{2.52}$$

over the vector

$$u_i = \begin{bmatrix} \dddot{C}_i^x \\ \bar{P}_i^x \\ \dddot{C}_i^y \\ \bar{P}_i^y \end{bmatrix}, \tag{2.53}$$

of motion parameters which gathers the jerk $\dddot{C}_i$ of the CoM and the future steps $\bar{P}_i$, with a cyclically varying quadratic term because of the cyclically varying matrix $\bar{V}$:

$$Q_i = \begin{bmatrix} Q_i' & 0 \\ 0 & Q_i' \end{bmatrix}, \tag{2.54}$$

with

$$Q_i' = \begin{bmatrix} U_v^T U_v + \alpha I + \beta U_z^T U_z + \gamma U_p^T E^T E U_p & -\beta U_z^T \bar{V} \\ -\beta \bar{V}^T U_z & \beta \bar{V}^T \bar{V} \end{bmatrix}, \tag{2.55}$$

and

$$r_i = \begin{bmatrix} U_v^T(S_v\hat{c}_i^x - \dot{C}_{i+1}^{x\_ref}) + \beta U_z^T(S_z\hat{c}_i^x - V\hat{p}_i^x) + \gamma U_p^T E^T(ES_p\hat{c}_i^x - \dot{C}_{i+1}^{x\_ref}) + \\ -\beta\bar{V}^T(S_z\hat{c}_i^x - V\hat{p}_i^x) \\ U_v^T(S_v\hat{c}_i^y - \dot{C}_{i+1}^{y\_ref}) + \beta U_z^T(S_z\hat{c}_i^y - V\hat{p}_i^y) + \gamma U_p^T E^T(ES_p\hat{c}_i^y - \dot{C}_{i+1}^{y\_ref}) \\ -\beta\bar{V}^T(S_z\hat{c}_i^y - V\hat{p}_i^y) \end{bmatrix}. \tag{2.56}$$

2.9.3 Constraints on the Center of Pressure

Considering that the foot on the ground has a polygonal shape, potentially depending on the support foot $s \in \{Left, Right\}$, the unilateral condition can be expressed as a set of linear constraints on the position of the CoP:

$$\begin{bmatrix} d_s^x(p^\theta) & d_s^y(p^\theta) \end{bmatrix} \begin{bmatrix} z^x - p^x \\ z^y - p^y \end{bmatrix} \leq b(p^\theta), \tag{2.57}$$

which are linear with respect to the position (p^x, p^y) of the foot on the ground but nonlinear with respect to its orientation p^θ. The column vectors d^x and d^y gather the x and y coordinates of the normal vectors to the edges of the feet, and the column vector b corresponds to the positioning of these edges.

Considering this constraint at all instants $t_{i+1} \dots t_{i+n}$ of the preview period can be done by:

$$D \begin{bmatrix} Z_{i+1}^x - V\hat{p}_i^x - \bar{V}\bar{P}_i^x \\ Z_{i+1}^y - V\hat{p}_i^y - \bar{V}\bar{P}_i^y \end{bmatrix} \leq b(p^\theta), \tag{2.58}$$

with the simple double block-diagonal matrix:

$$D = \begin{bmatrix} d_{s,i+1}^x(p_{i+1}^\theta) & 0 & d_{s,i+1}^y(p_{i+1}^\theta) & 0 \\ & \ddots & & \ddots \\ 0 & d_{s,i+n}^x(p_{i+n}^\theta) & 0 & d_{s,i+n}^y(p_{i+n}^\theta) \end{bmatrix}, \tag{2.59}$$

and the vector:

$$b(p^\theta) = \begin{bmatrix} b(p_{i+1}^\theta) \\ \vdots \\ b(p_{i+n}^\theta) \end{bmatrix}. \tag{2.60}$$

Expressed with respect to the vector u_i introduced in (2.53), this constraint takes the following form:

$$D \begin{bmatrix} U_z & -\bar{V} & 0 & 0 \\ 0 & 0 & U_z & -\bar{V} \end{bmatrix} u_i \leq b\big(p^\theta\big) + D \begin{bmatrix} V\hat{p}_i^x - S_z\hat{c}_i^x \\ V\hat{p}_i^y - S_z\hat{c}_i^y \end{bmatrix}, \tag{2.61}$$

which can be introduced directly in the QP (2.52).

2.9.4 Constraints on the foot positioning

We have derived simple linear constraints on the positions of the feet one with respect to the other with minimum and maximum values preventing collision on one side and over-stretching of the legs on the other side:

$$G \begin{bmatrix} -1\,1 & & 0 & -1\,1 & & 0 \\ & \ddots & & & \ddots & \\ 0 & & -1\,1 & 0 & & -1\,1 \end{bmatrix} \begin{bmatrix} \hat{p}_i^x \\ \bar{P}_i^x \\ \hat{p}_i^y \\ \bar{P}_i^y \end{bmatrix} \leq h\big(p^\theta\big), \tag{2.62}$$

with

$$G\big(p^\theta\big) = \begin{bmatrix} a_{s,i+1}^x\big(p_{i+1}^\theta\big) & & 0 & a_{s,i+1}^y\big(p_{i+1}^\theta\big) & & 0 \\ & \ddots & & & \ddots & \\ 0 & & a_{s,i+n}^x\big(p_{i+n}^\theta\big) & 0 & & a_{s,i+n}^y\big(p_{i+n}^\theta\big) \end{bmatrix}. \tag{2.63}$$

To do so, we took in consideration the MPC scheme in [Kajita et al., 2003]. Starting from the standard initial position of HRP-2, we used extensive offline calculations to build a point cloud of positions that can be reached by the center of the left foot with one single step.

The offline computation involved random positions, which were used as inputs by the MPC scheme in [Kajita et al., 2003] to generate joint space trajectories. Each trajectory was then checked with an *ad hoc* verification process that declared a trajectory feasible when it did not violate joint limits and avoided self-collisions.

The point cloud in Figure 2.10 shows all the positions that lead to feasible trajectories. We have defined a 5 edges polygon included in this area, as shown in Figure 2.10. It is symmetric about the y-axis in order not to take into account specificities of the HRP-2 robot and the MPC scheme in [Kajita et al., 2003], that tend to result in a slightly larger feasible zone when walking backwards.

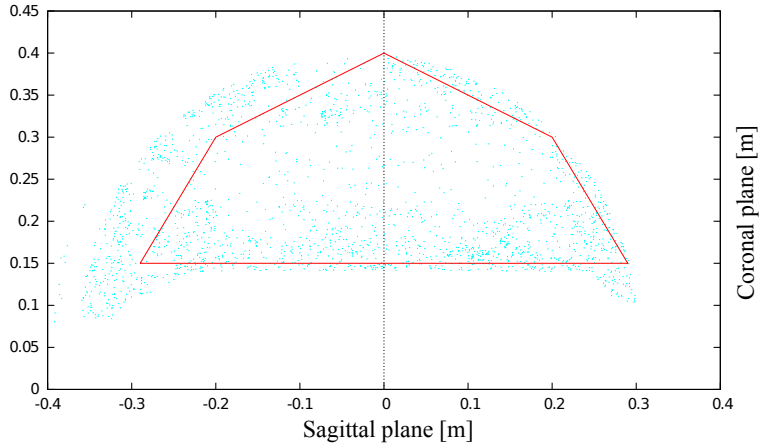


Figure 2.10: Polygonal approximation (solid red) of the feasible region of the final position of one foot relative to the support foot (dotted blue region).

2.9.5 Support and trunk orientations

Although tracking a rotational reference can be done by a QP as described in Section 2.4, our implementation differs in the sense that feasibility is verified analytically. The trajectory of the trunk orientation c^θ consists for this purpose of a fourth-order-polynomial acceleration phase, and a subsequent constant-velocity period that spans until the end of horizon t_f. The trajectories in between previewed foot positions consist of third-order polynomials. These two suppositions allow for a straightforward verification of kinematic feasibility constraints for the vertical hip joint and to avoid self collisions. The overall resolution process consists of the following steps:

1. The desired trunk velocity $\dot{c}^\theta{}^{ref}$ is obtained after a fixed period of time and keeps constant for the rest of the preview period.
2. If $\dot{c}^\theta{}^{ref}$ exceeds the maximal attainable velocity of the hip joint, it is reduced to the maximal value.
3. If the orientation of the trunk with respect to the stance foot at the end of the current support phase exceeds the maximal allowable value, the velocity is reduced such that the maximal angular difference is obtained.
4. The acceleration peak in the middle of the acceleration period is obtained analytically.
5. If this maximal allowable acceleration exceeds the limitation of the hip

joint, $\dot{c}^{\theta}\_^{ref}$ is computed by fixing the acceleration in the middle to the maximal acceleration of the hip joint.

6. As the desired solution, the orientation of the previewed support foot is posed such that the trunk and the foot are aligned in the middle of the support period.
7. The resulting maximal velocity of the hip joint is computed based on the assumption of third order polynomial trajectories, and the orientation corrected if maximal value is exceeded.
8. For self-collision avoidance, the orientations of the feet with respect to each other are verified and corrected, if necessary.
9. The capacity to achieve a rectangular double support phase with maximal velocity is verified and the foot angles corrected, if necessary.
10. Finally, the angular difference between the trunk and the support foot at the end of each support phase is verified. If a violation occurs, the overall process restarts with a reduced velocity reference $\dot{c}^{\theta}\_^{ref}$.

2.10 Appendix B: Application on the Visual Servoing

The following work has been effectuated in a collaboration with Claire Dune.

In the MPC scheme developed in the previous sections, we have shown that walking motions can be generated without predefined foot step placements but only a desired CoM velocity. The interaction with other control instances can be eased drastically this way. The above scheme has been successfully tested in series with a visual servoing controller (cf. [Dune et al., 2010]). The visual servoing scheme that will be introduced in the following computes a desired velocity that is given as a reference to the trajectory generator.

2.10.1 Visual Servoing

The system we consider in the following is an on-board camera rigidly linked to the robot's CoM. Let $\mathcal{C}$ and $\mathcal{K}$ be the frames attached to the CoM and the camera and $\dot{c}$ and $\dot{k}$ their velocities. The visual servoing system regulates to zero the error vector $e = s - s^*$ between some current visual features s and some desired visual features s^* [Chaumette and Hutchinson, 2006]. The key feature of this control scheme is the interaction matrix L which links the time variation of the visual features $\dot{s}$ to the relative camera/object kinematics screw $\dot{k}$. It is defined by:

$$\dot{s} = Lk \tag{2.64}$$

Then, the classical control law that regulates e with an exponential decrease $\dot{e} = -\lambda e$ is:

$$\dot{k} = -\lambda \hat{L}^{+} e \tag{2.65}$$

where $\hat{L}^{+}$ denotes the Moore-Penrose pseudo inverse of an approximation or a model of L, and λ is a positive scalar value.

2.10.2 Control of the trajectory generator

The trajectory generator of Chapter 2 ensures that the CoM tracks a reference velocity yet only in average and in the limits of the feasibility requirements. Let us describe two consequences in detail:

– *Limiting the velocity:* In order to ensure the tracking of the reference velocity, the three velocity components have to be limited to feasible ones, i.e. velocities that respect the feasibility constraints which depend on the robot's geometry and actuator capabilities. The robot's maximal speed $\dot{c}_{limit}$ is determined by the size of the polygonal constraints (2.19) and the stepping frequency. Yet this speed cannot be reached instantly from a stationary position.

We can distinguish three phases in the walking motion: i) an initial state (400 ms) where the robot is standing in double support, i.e. the two feet are on the ground and the robot stands still, ii) a nominal walking phase with a constant period of $\tau_{step} = 800$ ms, and iii) a final phase where the robot stands in double support. To switch from the initial double support state to the nominal walking phase, there is a transitory phase during which the CoP is brought from the center of the two feet to the center of the left foot. Then the robot starts walking, and the velocity of the CoM increases gradually during the first steps to reach a steady state where the reference velocity can be tracked up to $\dot{c}$. We then set a transient maximum velocity for the first two steps. The maximum velocity is then:

$$\dot{c}_{max} = \begin{cases} \frac{t}{2\tau_{step}}\dot{c}_{limit} & \text{if } t \leq 2\tau_{step} \\ \dot{c}_{limit} & \text{else.} \end{cases} \tag{2.66}$$

– *Canceling the sway motion:* Due to self-collision constraints, the stepping motion induces a lateral sway motion that prevents the CoM velocity from following instantaneously the expected one. The sway motion is mandatory for a proper walk and the control law should not compensate for it but to cancel its effects on the visual error computation.

Let us define $\dot{b}$ the additional sway motion of period $T = \tau_{step}/\tau$, such that $\sum_{l=i}^{i+T} \dot{b}_l = 0$. The behavior of the trajectory generator can be approximated by $\dot{c} = \bar{\dot{c}} + \dot{b}_c$ where $\bar{\dot{c}}$ would be the velocity if there where no swaying. This induces a motion of the camera of $\dot{k} = \bar{\dot{k}} + {}^kV_c\dot{b}_c$, where ${}^cV_k$ is the twist matrix associated to the cam-com transform ${}^cM_k$. If we assume ${}^cM_k$ to be constant over the time the camera velocity can be written $\dot{k} = \bar{\dot{k}} + \dot{b}_k$. The features will then oscillate in the image and the feature

variation can be written:

$$\dot{e} = \dot{s} = L\dot{\bar{k}} + L\dot{b}_k. \tag{2.67}$$

Let us define a virtual camera $\bar{\mathcal{K}}$ that corresponds to the position of the on-board camera if there was no sway motion. The velocity $\dot{\bar{k}}$ of this virtual camera is sent as input in the trajectory generator. In order to compute a control law that does not include the sway motion, we will servo this virtual camera $s(\bar{k})$ to $s(\bar{k}^*)$.

We have now to express $\bar{s} = s(\bar{k})$ with regards to the current measurement $s = s(k)$. With (2.64), we can write:

$$s(t) - s(0) = \int_0^t L\dot{k}dt = \int_0^t L(\dot{\bar{k}} + \dot{b}_k)dt, \tag{2.68}$$

and

$$\bar{s}(t) - \bar{s}(0) = \int_0^t L\dot{\bar{k}}dt. \tag{2.69}$$

Then, assuming that $s(0) = \bar{s}(0)$ and the above two equations, we obtain $s(t) = \bar{s}(t) + \int_0^t L\dot{b}_k dt$, from which we can deduce the corrected visual error:

$$\bar{e}(t) = \bar{s}(t) - s^* = e(t) - \int_0^t L\dot{b}_k dt. \tag{2.70}$$

Notice that when $\bar{e} \to 0$ then $e \to \int_0^t L\dot{b}_k$. In this study, we do not expect e to converge to zero but to oscillate around zero with a period T due to the sway motion. The convergence of the control law is then reached when $\int_{t_T}^t e dt = 0$, which is obtained if $\int_{t-T}^t \int_0^t L\dot{b}_k dt = 0$. Let us define $E = \int_{t-T}^t \int_0^t L\dot{b}_k dt$ and note that in general $E \neq 0$. It can be estimated over one period of time T. We can then use a sliding window to define the current error $\bar{e} = e - (\int_0^t L\dot{b}_k - E)$ and deduce the control law:

$$\dot{\bar{k}} = -\lambda L^+(e - \int_0^t L\dot{b}_k - Edt). \tag{2.71}$$

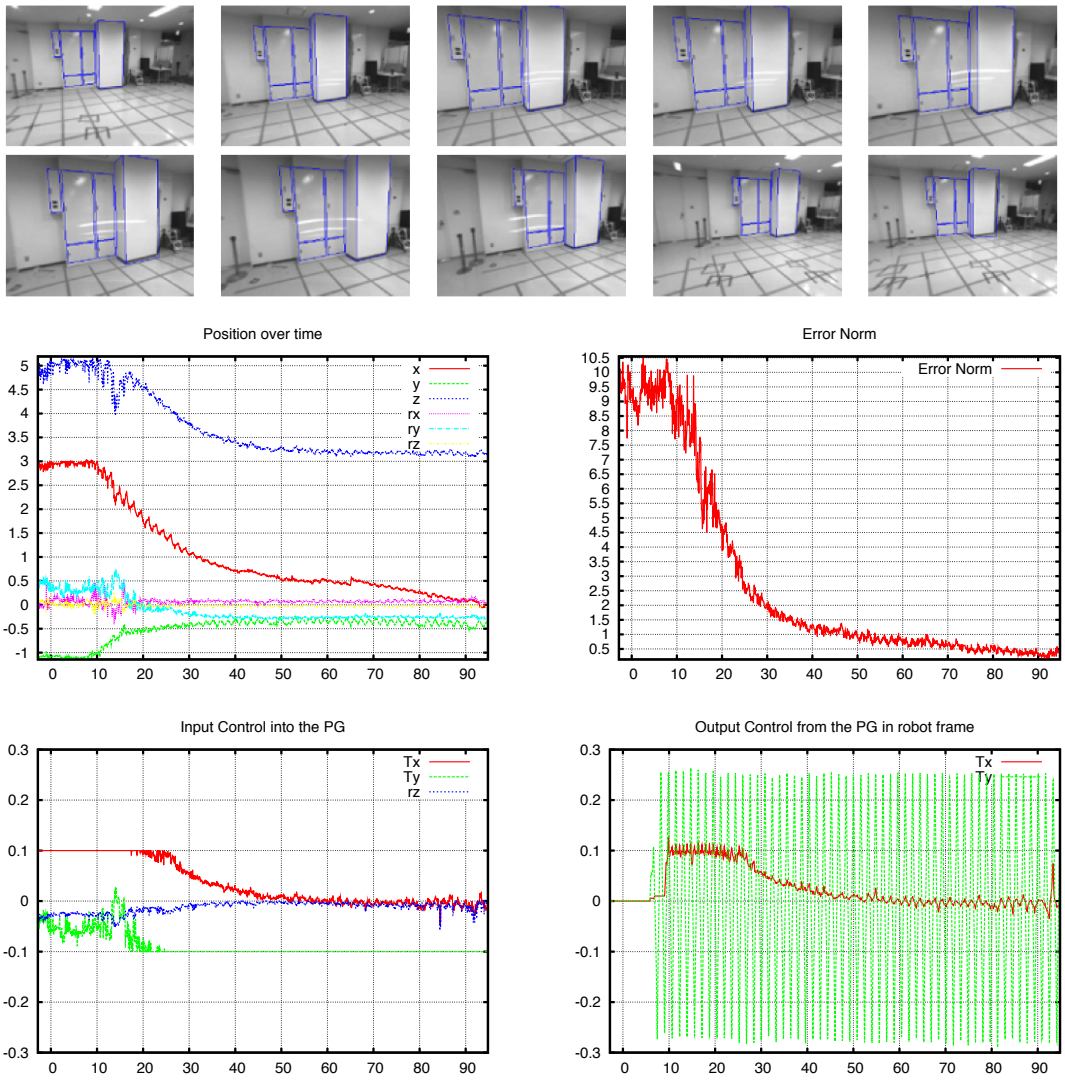


Figure 2.11: Model tracking while walking on a square on the ground with the HRP-2. The robot, firstly, walks forward, then sideways, then backwards, and sideways again to reach its initial position. Top left: Object's position (x, y, z) and orientation (rx, ry, rz) in the camera frame. Top right: Norm of the error. Bottom left: Control input of the pattern generator for desired longitudinal (Tx, Ty) and rotational (rz) velocities. Bottom right: Control output of the pattern generator for translational motions.

Chapter 3

Toward more efficient walking

The MPC scheme that we have introduced in the previous chapter is based on the (commonly made) assumptions that the motion of the Center of Mass is limited to a horizontal plane and that the support foot does not rotate around the heel or the toe during the entire support phase, i.e. foothold is always held with the entire sole. To circumvent the resulting restrictions on the motion of the robot, the height of the CoM has to be reduced, leading to the walk with strongly bent knees that is characteristic to most humanoid robots today.

Biomechanical studies highlight a walking pattern that differs in several points from the one generated by this scheme:

(a) The support leg is kept straightened during the entire stance phase.

(b) Toe flexion extends the effective leg length at the end of single support phases.

(c) The height of the CoM follows a sinusoid-like curve with the lowest position in the middle of the double support phase and the highest in the middle of the single support phase.

We know that this pattern is energetically advantageous [Donelan et al., 2002] [Gordon et al., 2009] [Ortega and Farley, 2005] [Adamczyk et al., 2006] [Kuo, 2007] [Franken et al., 2008] [Srinivasan and Ruina, 2006] [Tlalolini et al., 2009] since little muscle force is needed to support the body if the knee is kept straightened. However, overstating point (a) leads to stiff legs and inverted-pendulum-like walking motions that increase the vertical velocity at the end of single support phases [Kuo et al., 2005] and consequently the impact forces together with the amount of positive and negative work that is necessary to

redirect the Center of Mass during the double support phase. The production of both the positive and negative work requires energy expenditure in human muscles. The extension (b) of the stance leg during the redirection phase reduces this vertical velocity of the CoM together with the work that is necessary to redirect it to a new single support phase [Adamczyk and Kuo, 2009] [Kuo, 2002]. These two characteristics, almost straight stance legs and toe flexion, are the major reasons for the sinusoid-like evolution (c) of the CoM.

Even if evidence exists that the above two strategies increase the efficiency and performance of human walking motions, the same conclusion cannot be directly applied to robots. Joint actuators might be capable of producing electric power from negative mechanical work and energy can be stored for longer periods than in the human body. However, straight leg walking obviously alleviates the load on the joints by reducing the velocities and torques, especially in the knees. And toe flexion allows to increase the achievable stride length, as well as the height of the CoM during the double support phase. Imitating the above features of the human walk is therefore likely to improve the performance of walking robots.

As mentioned in the previous chapter, the assumption on the 'flatness' of the CoM motion has been made to avoid nonlinearities in the optimization problem. Additionally, if we are to avoid heuristics, the use of toes requires the consideration of a notion of the above mentioned gains in performance in the trajectory generation procedure. All existing approaches lack such a criterion and are based on predetermined strategies.

In this chapter, we achieve the imitation of the above characteristics by our online trajectory generation scheme. In Section 3.1, we show how variations of the CoM height can be achieved without losing the linearity of the horizontal motion equations. The proposed solution can be found in similar forms in [Buschmann, 2010] [Terada and Kuniyoshi, 2007] [Tajima et al., 2009], although either in a more restricting or a numerically less advantageous form. We address in Section 3.2 the geometric feasibility constraints that become important when the robot is walking with straightened legs. The offered solution is extended in Section 3.3 to a criterion for a more efficient and versatile use of toes than what can be found in the literature, for example in [Buschmann, 2010] [Sellaouti et al., 2006] [Kajita et al., 2007] [Nishiwaki et al., 2002b] [Miura

et al., 2011].

3.1 Vertical displacement of the waist

We have seen in the previous chapter that the motion of the CoM c of a robot walking on a perfectly horizontal ground is linked to the position of the Center of Pressure z on the ground by the following nonlinear second order differential equations:

$$c^{x,y} - \frac{c^z}{\ddot{c}^z - g}\ddot{c}^{x,y} = z^{x,y}. \tag{3.1}$$

This equation is linear w.r.t. the horizontal motion $c^{x,y}$ and even Linear Time-Invariant (LTI) when the height c^z of the CoM is fixed, what allowed as to define the LMPC scheme of the previous chapter. In what follows, we exploit the fact that when the height of the CoM varies as a predefined function of time $t \to c^z$, equation (3.1) becomes Linear Time-Variant (LTV), but can be handled equally well within the same control scheme.

Similar approaches for achieving vertical variations of the CoM without loosing the linearity can be found in [Buschmann, 2010] [Tajima et al., 2009] [Terada and Kuniyoshi, 2007] [Nagasaki et al., 2004] [Nagasaka et al., 2004]. There however, the right hand side is predefined prior to the resolution of the ODEs. The consequence is that no analytical solution can be given for arbitrary vertical variations of the Center of Mass so that numerical integration methods have to be employed or violations of boundary conditions accepted. In our case, the above dynamics make part of our Quadratic Programming formulation, therefore allowing a direct implementation of predefined vertical CoM trajectories.

3.2 Ensuring realizability

Obviously, predefining the vertical motion of the CoM must be done with care: frequency and amplitude have to be synchronized with the step duration and length. With the periods of the different support phases fixed in advance, as already required in the scheme of the previous chapter, the synchronization with the step sequence can be achieved easily. To obtain a desired gain in

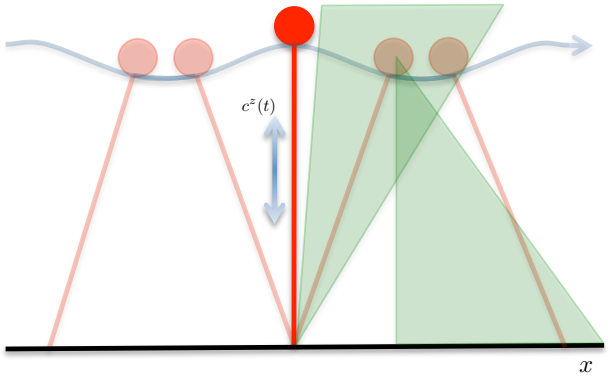


Figure 3.1: Shematic figure: Polyhedric constraints on the CoM position at the end of a double support phase with respect to the precedent support position, and constraints on the ensuing support position relative to the CoM position at the beginning of the double support phase.

efficiency, however, the amplitude of the vertical motion has to be chosen depending on the stride length and the walking direction, making the use of heuristic adaptation strategies necessary. A database of predefined trajectories for such a heuristic adaptation scheme can be obtained with a comprehensive set of simulations, what we will do here in Section 3.4.

3.2.1 Geometric feasibility

Whereas, the constraints (2.15) keep valid for a varying CoM height, constraints (2.19) are not associated with the position of the CoM, which does have an impact on the feasibility of leg motions. These constraints have therefore to be reconsidered from scratch.

Given the positions and orientations p of the support feet, we can define the set of reachable CoM positions in the cartesian space:

$$c \in \mathcal{C}(p). \tag{3.2}$$

In the following, we make the common assumption that the CoM of the robot doesn't move much with respect to its hip when walking. A simple forward kinematics procedure taking into account only the kinematics of the support legs is then sufficient to compute this reachable set. Figure 3.2 shows a sagittal

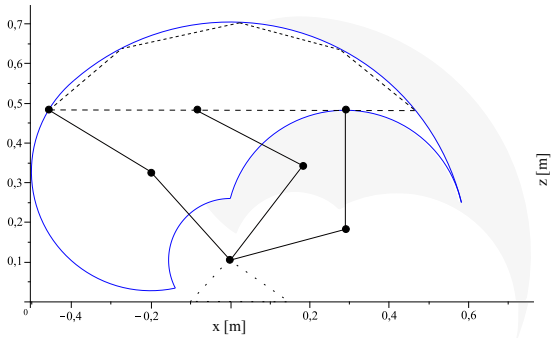


Figure 3.2: Feasible region in the sagittal plane of a straight HRP-2 waist before and after 45° rotation around the toe. The height indicated by the dashed line maximizes the horizontal reachability. The area above, on the contrary, leads to more 'comfortable' leg configurations that are, for this reason, more relevant for walking.

cut through $\mathcal{C}(0)$ for the HRP-2 robot. Being nonlinear and nonconvex, this set cannot be considered directly inside our linear control scheme. This entire set, however, is not of equal interest for walking: lower CoM positions lead to strongly flexed legs and consequently to high torques and high velocities in the knee joints, and the corresponding walk looks unnatural. We are interested therefore in CoM positions that lead to relatively stretched legs, i.e. the higher, convex part of the feasibility region (cf. Figure 3.2). Approximated by a convex polyhedron, this subset can be expressed by a set of inequalities linear with respect to the position of the Center of Mass:

$$E_c(p_i^\theta)(c_i^{x,y,z} - p_i^{x,y,z}) \leq e_c, \tag{3.3}$$

and nonlinear w.r.t. to the orientations p^θ of the supporting foot.

3.2.2 Constraint sampling

For the HRP-2 robot that we used to obtain the results in Section 3.4, an accurate approximation of the feasible area could be achieved by a polyhedron with ten faces. This number has to be multiplied by the number N of previewed samples if these constraints are verified for each sampling instant. Disregarding that this does not necessarily increase the resolution time significantly, verifying the above constraints that frequently might not be required to ensure the

realizability of generated motion references. Considering these constraints is the more important the closer the CoM is to the limits of $\mathcal{C}$, which can imply that ensuring feasibility at the beginning and end $t_i \in [SS^+, {}^-SS] \subset SS$ of a support phase alone can imply feasibility for all instants in between.

We then get the following optimization problem:

$$\underset{\ddot{c}^{x,y}_{1..N}, p^{x,y}_{1..S}}{\text{minimize}} \quad \sum_{i=1}^{N} (\dot{c}_i^{x,y} - \dot{c}_{i,ref}^{x,y})^2 \tag{3.4}$$

subject to

$$t_i \in [t_0, t_f] \qquad c_i^{x,y} - \frac{c_i^z}{\ddot{c}_i^z - g} \ddot{c}_i^{x,y} = z_i^{x,y} \tag{3.5}$$

$$t_i \in SS \qquad E_z(p_i^\theta)(z_i - p_i^{x,y}) \leq e_z \tag{3.6}$$

$$E_p(p_j^\theta)(p_{j+1}^{x,y} - p_j^{x,y}) \leq e_p \tag{3.7}$$

$$t_i \in SS^+ \qquad E_c(p_{i+1}^\theta)(c_i^{x,y,z} - p_{i+1}^{x,y,z}) \leq e_c \tag{3.8}$$

$$t_i \in {}^-SS \qquad E_c(p_{i-1}^\theta)(c_i^{x,y,z} - p_{i-1}^{x,y,z}) \leq e_c. \tag{3.9}$$

Note that now constraints (3.8) and (3.9) prohibit the over-stretching of the legs, but constraints of the form (3.7) are still necessary to prevent self collisions.

3.3 Purposeful toe flexing

As mentioned in the Introduction, straight legs alone do not lead to more efficient or natural looking motions. Although straightening the stance legs lowers the load on the joints and allows for a greater maximal horizontal displacement velocity, it also increases the vertical velocity of the height of the CoM at the beginning and end of its redirection phase around the double support phase, and therefore increases the required positive and negative mechanical work. This drawback can be circumvented by toe flexion at the end of the single support phase, what allows to increase the effective length of the stance leg at this period and thus reduce the vertical oscillation of the CoM. The question then is how to ensure that the generated walking motions exhibit this feature, or equivalently, how to express the gain due to the use of toes in our walking generation scheme.

3.3.1 Dynamic and geometric feasibility

Toe support phases require the CoP to lie under the toe, and in the case of non-actuated, compliant toe joints the position of the CoP relative to the position of the toe joint p_T is even directly related to the angle of toe flexion p_T^ϕ:

$$p_T^\phi := \begin{cases} Kz^x & z^x > p_T^x \\ 0 & else \end{cases}, \tag{3.10}$$

with x pointing in the sagittal direction of the foot.

Toe motions cannot therefore be generated independently from the position of the CoP but forcing it to be under the toe will influence the motion of the CoM due to (3.1), what may perturb the realization of other objectives such as walking with a desired speed or following a position reference of the CoM.

An elegant solution to this problem can be found by looking closer at the feasibility constraint (3.3) and how it evolves in case of a rotation around the stance toe of angle p_T^ϕ (gray in Figure 3.2):

$$E_c(p_T^\theta)R_c(-p_T^\phi)(c^{x,y,z} - p^{x,y,z}) \leq e_c, \tag{3.11}$$

where $R_c(-p_T^\phi)$ rotates the original constraints around the toe joint by the angle p_T^ϕ, which is equivalent to the rotation of the CoM by the angle $-p_T^\phi$. Taking into account the rotation of these constraints would allow the above control scheme to decide upon the toe angle depending on the desired position of the CoM. However, even in the case of a globally linear approximation of equation (3.10):

$$p_T^\phi = K(z^x - p_T^x) \tag{3.12}$$

inequalities (3.11) stay nonlinear, and this prohibits their straightforward consideration by the above linear QP. In order to do so, we require a linear approximation of these inequalities of the following form:

$$E_c(p_T^\theta)(c^{x,y,z} - p^{x,y,z} + \frac{\Delta c^{x,y,z}}{\Delta p_T^\phi} K(z^x - p_T^x)) \leq e_c, \tag{3.13}$$

but then this approximation has to work in the general case.

One useful fact in this regard is that we are mainly interested in toe flexion by the end of the single support phase. Extending inequalities (3.8) to the form of (3.13) only at $t_i \in SS^+$ is enough to achieve this goal and has the

consequence that the CoM is not over-constrained at earlier instants. In the upper part of figure 3.3, we can see the evolution of the frontal part of the constraints (3.11) for a given CoM position and a stiffness K that allows to reach the maximal toe angle (here 45°) with the CoP at the frontal border of the supporting toe (here 0.135 m). In its lower part, positions of the CoM relative to the center of one foot are plotted for several walking speeds and directions. The linear approximation (3.13) of the nonlinear constraints in dashed enforces the use of toes but at the same time does not over-restrict the set of feasible CoM positions when walking backwards. Contrary to (3.11), such approximations do reduce the feasible positions of the CoM the more the closer the CoP approaches the heel so that, dependent on the chosen K, several of the shown maneuvers might not be realizable with the associated control scheme.

Because the maximal attainable sagittal position of the CoM depends also on its deviation in the frontal plane, the gradient $\Delta c^{x,y,z}/\Delta p^{\phi}$ has to be chosen such that it is valid for a relevant set of CoM deviations in the frontal plane. In the case of the HRP-2 robot, an accurate, conservative estimation of this gradient can be given for a large set of inclinations p^{ψ} around the ankle in the frontal plane ($p^{\psi} \in [-25°, 25°]$) by translating the original polyhedron by the distance that is covered by the toe ankle after a rotation of $p^{\phi} = 45°$. The results that are presented in the next section have been obtained with this approximation and a stiffness K that allows to fulfill a maximal rotation around the toe joint by 45°.

Here, we are focussing on the use of active toes. Active toes can compensate for intermediate violations because of the capacity to generate the additional torque that is necessary for a desired rotation. Then, even if the orientation of the toe is assumed to be linear with respect to the relative position of the CoP z^{+}, the actual orientation of the toe does not have to follow this reference. Full-body trajectories can consequently be obtained inside a classic inverse kinematics procedure, with the only supposition that the CoP stays underneath the stance toe during the entire rotation. An actuated toe joint introduces furthermore a redundancy to a leg with three pitch joints. During toe flexion, the mapping from the joints to the position of the Center of Mass is then not unique in the sagittal plane which offers the possibility to fulfill

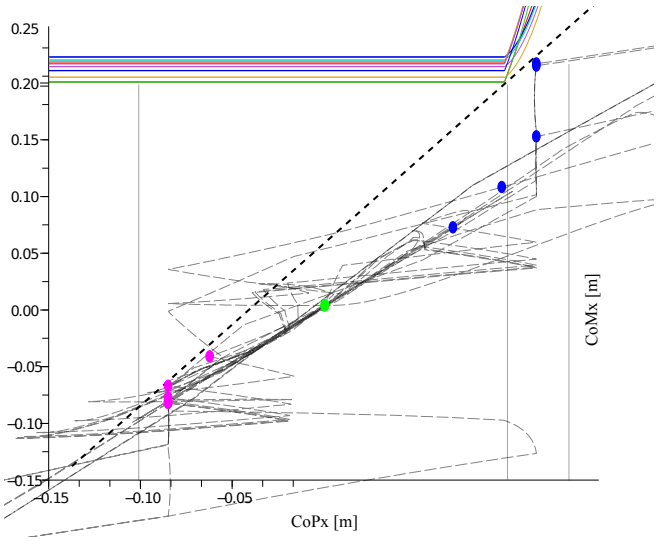


Figure 3.3: Upper part: Evolution of constraints (3.11) in the sagittal direction x for a given position of the CoM and a supposed linear relation between the sagittal position z^x of the CoP and the toe angle p^ϕ. Lower graph: Trajectories of the CoM with respect to the CoP during stance phases of the left foot (indicated by vertical lines) for several walking speeds and directions. The points are the positions of the CoM at $t_i \in SS^+$ when the robot is walking forward (blue), sideways (green), and back (red). The linear approximation (dashed black) respects all constraints at the minimal and the maximal toe orientation but reduces the set of feasible positions (at $t_i \in SS^+$) so that walking backwards is possible only with a slower speed.

supplementary objectives at the joints. Similarly to what can be found for example in [Buschmann, 2010], we choose the norm of the leg joint velocities $\|\dot{q}_L\|$ as a minimization objective subject to constraints for the tracking of the CoM:

$$\text{minimize} \qquad \|\dot{q}_L\|^2 \tag{3.14}$$

$$\text{subject to}$$

$$\dot{c} - J(q_L)\dot{q}_L = 0, \tag{3.15}$$

with $J(q_L)$ the Jacobian of a given configuration of q_L.

3.4 Walking efficiency

To study the influence of extensions (3.3) and (3.13) on the efficiency of the complete control scheme, i.e. trajectory generation and tracking, we have performed a series of dynamic simulations with varying walking speeds and vertical oscillations of the CoM. The vertical oscillation $c^z(t)$ of each simulation is the result of fifth order polynomial interpolations between predetermined vertical positions, velocities, and accelerations of the CoM at redirection instants $t_r \in [t_0, t_f]$ around the double support phase:

$$\begin{aligned} c^z(t_r) &\in [0.8, 0.9]m, \\ \|\dot{c}^z(t_r)\| &\in [0, 0.1]\frac{m}{s}. \end{aligned} \tag{3.16}$$

The horizontal displacement of each simulation is due to a constant reference $\dot{c}^{x,y}_{ref}$ in the forward direction:

$$\dot{c}^{x,y}_{ref} \in [0.1, 0.8]\frac{m}{s}. \tag{3.17}$$

In each simulation, the HRP-2 robot is walking forward for 10 s, trying to realize the prior specified velocity reference for the given vertical evolution of the CoM. Every simulation is performed twice: with the (static) polyhedral constraints introduced in Section 3.2, and the enforced use of toes as discussed in Section 3.3. The resulting motions have been verified against the limitations of the robot, and the realizable motions have been recorded and analyzed.

Figure 3.4 traces two different approximations of the energy consumption against the realized average horizontal velocity:

$$\int_0^{T_{sim}} \sum_j \Gamma(q_1^j(t), \ddot{q}_1^j(t))^2 dt, \tag{3.18}$$

$$\int_0^{T_{sim}} \sum_j \|\Gamma(q_1^j(t), \ddot{q}_1^j(t))\dot{q}_1^j(t)\| dt, \tag{3.19}$$

where $q_1^j(t)$ is the position of an actuated joint j and T_{sim} the duration of a simulation. The first criterion reflects the ohmic drop in the joint motors by the integral of the square of torques, whereas the second measures directly the mechanical work effectuated by the joints during the entire simulation.

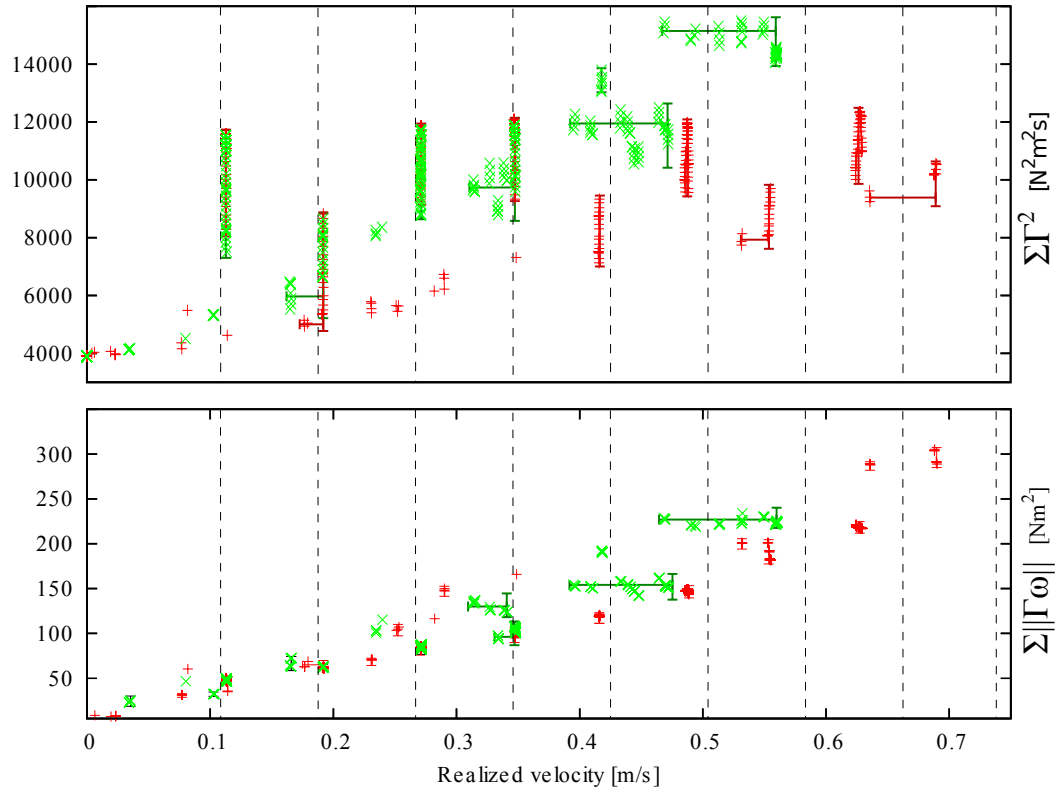


Figure 3.4: Energy consumption for realized mean velocities and different vertical oscillations of the CoM, with the use of toes (red +) and without (green x) as well as different velocity references (dashed black).

Contrary to the second criterion, the first one shows a considerable sensitivity to the variation of the vertical CoM oscillations, showing that a significant reduction of torque load can be achieved by a proper adaptation of the vertical evolution. The observed difference between the two criterions is due to the fact that the first one is minimized by an equal distribution of the joint torques, whereas the second rises proportionally to the velocities and torques in every joint. Therefore, the vertical evolution alone does not affect the total amount of work to a considerable extent.

This is different when toes are used, what can be observed for higher CoM positions, but especially for horizontal displacement velocities above 0.3 m/s. There again however, the reduction is significantly stronger for the first criterion, as toe joints permit to lower the maximal torques in other joints of the leg. We can expect a higher gain in terms of the second criterion when using

passive toes, what however, has not been addressed in this thesis.

Both criterions show, on the contrary, that the inclusion of toe motions inside our scheme can allow for a decrease in energy consumption for a wider range of displacement velocities if the vertical oscillations have been chosen properly. The strong fluctuation of the minima in the upper graph of Figure 3.4 shows, however, that with the chosen set of vertical oscillations (3.16), this has not been achieved for all realized average velocities. A larger set of vertical trajectories is needed to complete these results and to provide a more comprehensive basis for the adaptation scheme mentioned in Section 3.1.

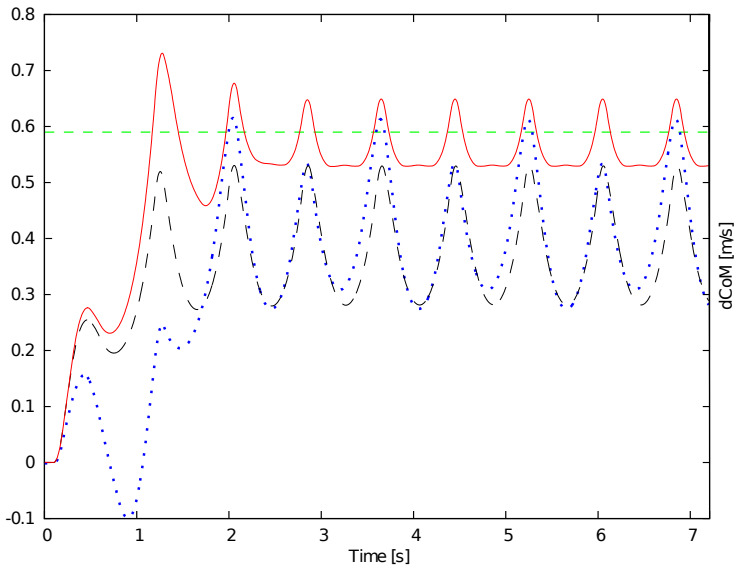


Figure 3.5: Attained sagittal CoM velocities with (solid red) and without the use of toes (dotted blue) or polyhedric constraints (dashed black) for a given velocity reference (dashed green).

We can see, furthermore, that the use of toes allows to realize considerably faster walking speeds $\dot{c}_{ref}^{x,y}$ in the forward direction. If, on the contrary, the desired velocity is not realizable for the chosen oscillation of the CoM, the scheme of manages reduce the displacement velocity. The polyhedric constraints restrict the set of achievable CoM and foot positions such that only geometrically feasible trajectories are generated.

This can be observed in more thorough detail in Figures 3.5 and 3.6 that compare three different schemes introduced in this and the previous chapter.

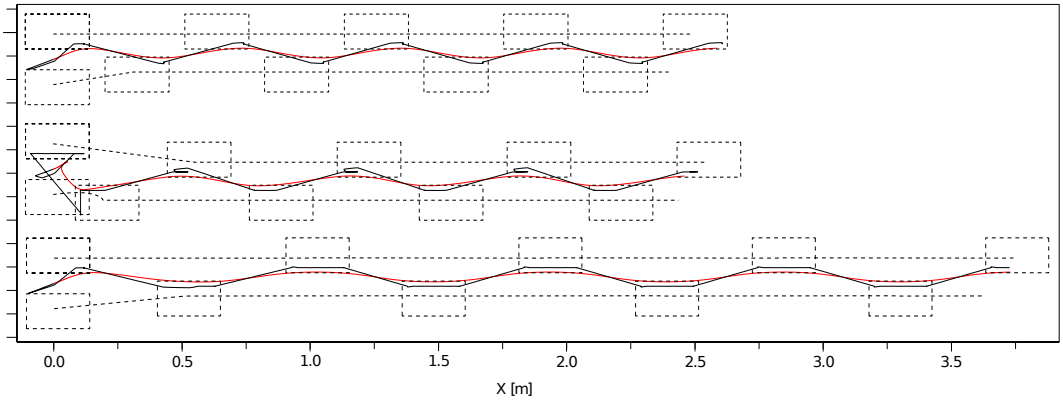


Figure 3.6: CoM (solid red), CoP (solid black) and foot trajectories (dashed black) generated during the simulations in Figure 3.5. From up to down: Polygonal constraints on the support positions, polyhedral constraints on the CoM and the support positions, and polyhedral constraints extended to the use of toes.

Although the velocity obtained with the scheme of the previous chapter is negligibly smaller than the one of Section 3.2, an obvious difference in the horizontal evolutions of the CoM and CoP with respect to the positions of the feet can be observed in Figure 3.6. Due to polyhedric constraints, the scheme of Section 3.2 places the feet further ahead of the CoM and the CoP to allow for a realization of the desired CoM trajectory. The absence of these constraints (3.9), and the resulting 'unconsciousness' of the geometric limitations cause the scheme of the previous chapter to generate unrealizable motions for higher speed references $\dot{c}_{ref}^{x,y}$ or CoM positions. It is for this reason that these constraints constitute an important completion of our trajectory generation scheme. By rendering it more robust to unrealizable references they enforce the most valuable property of our trajectory generation scheme: The feasibility requirements are always respected, even if it means not to realize a desired motion.

3.5 Conclusion

Reproducing characteristics of the human walk can significantly improve the walking performance of robots. The combination of both the natural oscillation of the CoM and the use of toes allows to reduce the load on the leg joints and to augment the walking speed. As an essential element of this chapter, polyhedral constraints on the position of the CoM allow first to ensure the geometric realizability of the CoM and leg motions, and second, to consider the use of toes inside the preview of walking trajectories.

In this chapter, vertical oscillations of the CoM have been given prior to the generation of horizontal trajectories, but the question how to adapt the vertical variations such that natural and efficient walking motions can always be obtained has not been answered. All predefinitions made in this and the previous chapter helped to avoid nonlinearities but made the use of adaptation schemes necessary. The finite state machine for gait coordination and the decoupled QP for the computation of foot orientations are examples of such schemes. The following chapter is an attempt to avoid the precomputation of vertical trajectories by a unified, nonlinear approach for the generation of three-dimensional walking trajectories.

3.6 Résumé

Reproduire les caractéristiques de la marche humaine peut améliorer de manière significative la performance des robots. La combinaison des deux, l'oscillation naturelle du CdM et l'utilisation des orteils, permet de réduire les efforts dans les articulations des jambes et d'augmenter la vitesse de la marche. En tant qu'élément essentiel de ce chapitre, des contraintes polyédriques sur la position du CdM permettent d'abord d'assurer la faisabilité géométrique des mouvement du CdM et des jambes, et deuxièmement, prendre en considération l'utilisation des orteils lors de la prédiction des trajectoires de la marche.

Dans ce chapitre, les mouvements verticales du CdM ont été donnés avant de générer les trajectoires horizontales, mais la question de comment adapter ces variations verticales tel que des mouvement naturels et efficaces peuvent toujours être atteints n'a pas été répondu. Toutes prédéterminations faites dans ce et les chapitres précédents aidaient à éviter des non-linéarités mais rendaient nécessaire l'utilisation des algorithmes d'adaptation. La machine à état fini pour la coordination de la marche et le problème d'optimisation découplé pour le calcul des orientation des pieds sont des exemples pour tel chemins. Le chapitre suivant est une tentative d'éviter des prédéterminations des trajectoires verticales par une approche unifiée, non-linéaire des trajectoires de la marche à trois dimensions.

3.7 Appendix C: Multi-body motions

We can observe that for faster walking with the HRP-2 robot the Centers of Pressure of the comprehensive and the point-mass models deviate strongly and that his deviation becomes critical when the robot is supported only by the stance toe. A point-mass model is able to capture a crucial part of the robots dynamics - the Center of Mass. The state of the CoM can give very accurate information on the contact forces for slower walking motions, but this model loses its accuracy when bodies of the robot generate non-negligible variations in the momentum of the robot. This change of momentum can result from rotational and longitudinal motions of body parts like the legs, the arms, or the trunk, and requires the consideration of additional point-masses for longitudinal motions or rigid bodies for rotational motions.

Here, we show how to improve the accuracy of the CoP prediction inside our linear MPC scheme. We review the motion equations of multiple rigid bodies introduced in Chapter 1 and show how to reduce them to a linear form. A three-point-mass model will serve us thereupon to improve the prediction of the CoP for faster motions.

3.7.1 System of rigid bodies

Let us restart from equation (1.14) and (1.15) the series of simplifications made in Chapter 1:

$$\sum_k m_k(\ddot{c}_k - \vec{g}) = \sum_k f_k^c \tag{3.20}$$

$$\sum_k c_k \times m_k(\ddot{c}_k - \vec{g}) + \sum_k R_k I_k \dot{\omega}_k = \sum_k c_k \times f_k^c + R_k \tau_k. \tag{3.21}$$

But instead of simplifying the equations by introducing the Center of Mass, we will assume that the bodies are perfectly symmetric with respect to their origins c_k so that their orientations do not affect the torques generated during rotational momenta changes $\dot{\omega}_k$:

$$\sum_k m_k(\ddot{c}_k - \vec{g}) = \sum_k f_k^c \tag{3.22}$$

$$\sum_k c_k \times m_k(\ddot{c}_k - \vec{g}) + \sum_k I_k \dot{\omega}_k = \sum_k c_k \times f_k^c + \tau_k, \tag{3.23}$$

By reintroducing the contact points p_k, we get:

$$\sum_k m_k(\ddot{c}_k - \vec{g}) = \sum_k f_k^c, \tag{3.24}$$

$$\sum_k c_k \times m_k(\ddot{c}_k - \vec{g}) + \sum_k I_k\dot{\omega}_k = \sum_k p_k \times f_k, \tag{3.25}$$

and with the assumption of horizontally aligned, non-sliding contact points as well as the CoP equation (2.13), we get the following two nonlinear, ordinary differential equations for the motions of rigid bodies:

$$\frac{1}{\sum_k m_k(\ddot{c}_k^z - g)} \sum_k m_k \begin{pmatrix} (\ddot{c}_k^z - g)c_k^x - c_k^z\ddot{c}_k^x + I_k\dot{\omega}_k^y \\ (\ddot{c}_k^z - g)c_k^y - c_k^z\ddot{c}_k^y + I_k\dot{\omega}_k^x \end{pmatrix} = \begin{pmatrix} z^x \\ z^y \end{pmatrix}. \tag{3.26}$$

Here again, with predefined vertical motions for each body, we end up with linear relationships between the horizontal body motions and the CoP.

$$\frac{1}{\sum_k f_k^z(t)} \sum_k m_k \begin{pmatrix} (\ddot{c}_k^z(t) - g)c_k^x - c_k^z(t)\ddot{c}_k^x + I_k\dot{\omega}_k^y \\ (\ddot{c}_k^z(t) - g)c_k^y - c_k^z(t)\ddot{c}_k^y + I_k\dot{\omega}_k^x \end{pmatrix} = \begin{pmatrix} z^x \\ z^y \end{pmatrix}. \tag{3.27}$$

3.7.2 Three-point-mass model for fast walking

Walking motions are usually obtained with the trunk of the robot kept relatively straight. The observed deviations are thus not the result of rotational momenta changes but mainly the translational accelerations of the legs. The HRP-2 has indeed relatively heavy legs, making together about 30 % of the total mass. Faster walking is therefore likely to generate angular momenta that can lead to strong deviations of the real CoP from the previewed one if only a single point-mass model is used. The motions of the upper body and the legs have then the greatest influence on the evolution of the CoP. We choose here, therefore, to replace the CoM model by three point-masses $c_{w,l,r}$ that represent the longitudinal motions of the waist and both legs [1].

With the angular moment τ_k defined by:

$$\tau_k^y = m_k\gamma_k^x = m_k(\ddot{c}_k^z - g)c_k^x - m_k c_k^z\ddot{c}_k^x,$$
$$\tau_k^x = m_k\gamma_k^y = m_k(\ddot{c}_k^z - g)c_k^y - m_k c_k^z\ddot{c}_k^y,$$

1. As a positive side effect, the generation of waist motions offers the possibility to improve the accuracy of the polyhedral constraints (3.3) introduced above.

and the vertical reaction force f_k^z due to the vertical acceleration of the particle k:

$$f_k^z = m_k(\ddot{c}_k^z - g),$$

we get:

$$\frac{1}{f_c^z + f_r^z + f_l^z} \begin{pmatrix} \tau_c^y + \tau_l^y + \tau_r^y \\ \tau_c^x + \tau_l^x + \tau_r^x \end{pmatrix} = \begin{pmatrix} z_k^x \\ z_k^y \end{pmatrix}. \tag{3.28}$$

3.7.3 Identification of leg masses

Our goal is to reduce the deviations of the previewed CoP $z_p(t)$ from the realized one $z_r(t)$ [2]. Different criteria can be thought of for the identification of the point-masses, depending on whether the maximal or the mean deviation is to be minimized. In the following we will focus on the mean deviation.

To identify the three-mass model we are fitting the CoP generated by the simplified model to the CoP of the complete model. We can see that contrary to (3.27), equation (3.26) is linear with respect to the masses. Therefore, to identify the three masses, the following linear least squares problem can be formulated:

$$\underset{M_p}{\text{minimize}} \ \|\Gamma_p M_p - T_r\|^2, \tag{3.29}$$

with M_p being the vector of masses to be identified and $\Gamma_p M_p$, T_r the previewed and the realized evolution of torques:

$$\Gamma_p M_p = \begin{pmatrix} \gamma_w(t_1) & \gamma_r(t_1) & \gamma_l(t_1) \\ \ddots & & \\ \gamma_w(t_N) & \gamma_r(t_N) & \gamma_l(t_N) \end{pmatrix} \begin{pmatrix} m_w \\ m_r \\ m_l \end{pmatrix}, \tag{3.30}$$

$$T_r = \begin{pmatrix} z_r(t_1) \sum f_z(t_1), \\ \vdots \\ z_r(t_N) \sum f_z(t_N) \end{pmatrix}, \tag{3.31}$$

2. The identification of the masses can be made only with respect to a limited set of trajectories such that model errors are unavoidable. Model errors however are different from perturbations in the sense that they are systematic. Inside an MPC framework, model errors can be compensated for by assuming that the measured deviations from the prediction will stay constant for the following preview period. Compensating control can be computed if the deviation is taken into account.

where N is the number of recorded samples.

Being linear this least squares problem can be solved analytically with:

$$M_p = (\Gamma_p^T \Gamma_p)^{-1} \Gamma_p^T T_r. \tag{3.32}$$

3.7.4 Preview Control of the three-mass model

In order to modify the previous control scheme as little as possible, we control the jerk of the waist $\dddot{c}_w$, and the trajectories of the feet $c_p(t)$ are obtained by single fifth-order polynomial [3] interpolations between the current position and previewed landing positions $p^{(i)}$ (cf. Appendix A). With the polynomial equations

$$c_p^h(t) = at^5 + bt^4 + ct^3 + dt^2 + et + f, \tag{3.33}$$

$$\dot{c}_p^h(t) = 5at^4 + 4bt^3 + 3ct^2 + 2dt + e, \tag{3.34}$$

$$\ddot{c}_p^h(t) = 20at^3 + 12bt^2 + 6ct + 2d, \tag{3.35}$$

$$\tag{3.36}$$

$p_0^h, \dot{p}_0^h, \ddot{p}_0^h$ defining the horizontal state of a foot at an instant t_0, and p_n^h the previewed position of the following support foot at the touchdown instant t_d, the foot trajectories are determined by the following equations:

$$a = -\frac{1}{2}\frac{\ddot{p}_0^h(t_d - t_0)^2 + 6\dot{p}_0^h(t_d - t_0) + 12p_0^h - 12p_n^h}{(t_d - t_0)^5}, \tag{3.37}$$

$$b = \frac{1}{2}\frac{3\ddot{p}_0^h(t_d - t_0)^2 + 16\dot{p}_0^h(t_d - t_0) + 30p_0^h - 30p_n^h}{(t_d - t_0)^4}, \tag{3.38}$$

$$c = -\frac{1}{2}\frac{3\ddot{p}_0^h(t_d - t_0)^2 + 12\dot{p}_0^h(t_d - t_0) + 20p_0^h - 20p_n^h}{(t_d - t_0)^3}, \tag{3.39}$$

$$d = \frac{1}{2}\ddot{p}_0^h, \tag{3.40}$$

$$e = \dot{p}_0^h, \tag{3.41}$$

$$f = p_0^h, \tag{3.42}$$

that are all linear with respect to the previewed landing position p_n^h.

3. The interpolation with order 5 polynomials can cause high acceleration peaks between the sampling instants. A solution can be to consider several polynomials of a lower order as in the case of trajectories with a piecewise constant jerk or splines with second-order continuous derivatives.

For a fixed sampling rate we can formulate the following recursive relation that determines the foot trajectories for the preview period:

$$C_p = S_p \begin{pmatrix} c_p^h \\ \dot{c}_p^h \\ \ddot{c}_p^h \end{pmatrix} + U_p \begin{pmatrix} p_1^h \\ \vdots \\ p_s^h \end{pmatrix}. \tag{3.43}$$

The constraints on the CoP become consequently:

$$\bar{E}_z \left(Z - C_k \right) \leq \bar{\epsilon}_z, \tag{3.44}$$

with the evolution of the CoP being determined by:

$$\begin{aligned} Z = & S_w^z \hat{c}^w + U_w^z \dddot{C}_w + \\ & S_l^z \hat{c}_l + U_l^z P_l + \\ & S_r^z \hat{c}_r + U_r^z P_r, \end{aligned}$$

and the one of the support positions by:

$$\begin{aligned} C_k = & V_l (S_l \hat{c}_l + U_l P_l) + \\ & V_r (S_r \hat{c}_r + U_r P_r), \end{aligned}$$

where the selection matrices V indicate the foot in the air.

Chapter 4

3D motions

In our purpose to obtain linear MPC formulations, we were obliged in the previous two chapters to predetermine the vertical evolutions of the particles and the contact points. Such measures obviously limit the capacity of the control scheme to react to varying control goals or external conditions. Adaptation schemes have to be employed on top of the actual trajectory generator to reduce this deficiency. The question of how to switch between trajectories to obtain smooth transitions and robust motions is not easy to answer though (cf. [Wieber and Chevallereau, 2004]). Also, the set of trajectories that can be precomputed offline is inevitably limited so that no guarantee can be given that the chosen trajectory is an appropriate one (cf. Section 3.4).

Prior to generating three dimensional walking trajectories, however, several newly arising questions have to be reconsidered again. Since the vertical evolutions, that had been predetermined in the previous chapters, are to be generated within the optimization problem, the feasibility conditions have to be reconsidered, and, what is more, an appropriate control goal has to be found. As has been shown in the previous chapters, the motions of point-masses or rigid bodies in the three-dimensional space are related to the contact forces by nonlinear differential equations, nonlinearities arising even for a single point mass system. Linear approximations might not be acceptable for stronger vertical CoM variations that are not avoidable for tasks like climbing stairs or walking down a slope. Generating three dimensional motion trajectories by means of Model Predictive Control requires therefore the resolution of nonlinear optimization problems.

Several related optimization-based approaches exist in computer graphics that manage to generate three-dimensional walking trajectories by means of simplified models. By considering the vertical CoM motions independently from horizontal ones, [Mordatch et al., 2010] succeed to generate walking motions on nonplanar ground. Although decoupled computation of vertical variations can lead to naturally looking walking motions, their feasibility can not be guaranteed anymore, the more so as in this approach the Zero Moment Point (ZMP) is used to assure dynamic feasibility on nonplanar ground. In [Van De Panne, 1997] point-mass motions are generated through the minimization of two objectives, although again, without explicitly ensuring physical plausibility [1]. As concluded in Chapter 1, feasibility is the most important exigence on walking motion references in robotics, and, as shown in Chapters 2 and 3, considering the limitations of a system by the control law can improve its performance.

The trajectory generator introduced in this chapter can be seen as an extension of both the second approach mentioned above and the MPC schemes developed throughout the previous chapters. Building on algorithms for fast computation of neighboring optimization problems, simulations of a point-mass model will serve us to prove the capacity of this scheme to achieve stable and natural walking motions.

4.1 Numerical NMPC methods

Nonlinear MPC allows to make use of comprehensive dynamic models like the one developed in Chapter 1, but the computational load can limit the online applicability. The supposition that underlies the approach taken in the following sections is that numerical methods for NMPC have achieved a status where real-time control of a robot by means of dimensionality-reduced nonlinear models has become feasible on a modern computer [Diehl et al., 2005b] [Houska et al., 2011]. Several generic real-time computation schemes

1. Interestingly, the approach taken in [Van De Panne, 1997] can be seen as the three-dimensional version of Kajita's ZMP Preview Control scheme [Kajita et al., 2003], which is in many ways the basis of this thesis. The contribution of Section 4.3 has therefore similarities to the extension of Kajita's ZMP Preview Control [Kajita et al., 2003] by [Wieber, 2006], where dynamic feasibility has first been considered explicitly.

have been developed in the near past that have the potential to extend the application domain from process industries to robotics, where time scales are in the range of milli- or even microseconds. An extensive introduction to these approaches can be found in [Diehl et al., 2009]. The following is a short summary of strategies that are applied by them:

- Offline pre-computations: Subsequent optimization problems vary, but some variations might be predictable, and state invariant parts might exist that can be precomputed offline.
- Delay compensation by prediction: As the system's state evolves during the computation of a control, the accuracy of the solution can be improved by starting the resolution procedure from a predicted state instead of the current one.
- Division into preparation and feedback phases: Since solutions of consecutive optimization problems of a sufficiently fast sampled process share similarities, it can be possible to compute a first solution with an old state and, subsequently, use the current state to make a quick approximation of the current optimal solution.
- Iterating while the problem changes: Instead of iterating until the solution converges to a minimum, the behavior of a plant might be improved by applying suboptimal control but more frequently.

The results of this chapter have been obtained with the Real Time Iteration Scheme [Diehl et al., 2005a] that exhibits two crucial features:

1. To augment the sampling frequency, only one SQP-type iteration per sampling time is performed.
2. To achieve additionally short feedback delays, each iteration is divided into a preparation and a short, approximating feedback phase.

4.2 Nonlinear Optimal Control Problem

Still, even with adapted numerical methods as discussed above, the dimension and the form of the optimization problem play a decisive role for the computation time. In this section, we are going to develop a numerically convenient, nonlinear formulation of the general MPC scheme in Chapter 1.

4.2.1 Dynamic equilibrium of a system of particles

Let us recall the approximation of a robot by a limited number of point-masses c_i and a set of contact points p_k. The set of dynamically feasible states of such a system is defined by the following equilibrium conditions between the system and the environment:

– Change of linear momenta:

$$\sum_i m_i(\ddot{c}_i - g) - \sum_k f_k^i = 0. \tag{4.1}$$

– Change of angular momenta:

$$\sum_i m_i(\ddot{c}_i - g) \times c_i - \sum_k p_k \times f_k^i = 0, \tag{4.2}$$

with p_k being the k^{th} contact point and $f_k = \sum_i f_k^i$ the force acting at it.

– The contact forces have to respect the unilateral and bilateral inequality conditions (1.11) and (1.12), that prevent the contact body from taking off or sliding:

$$f_k \in \mathcal{A}_k. \tag{4.3}$$

By combining the left and the right sides of equations (4.1) and (4.2) in the internal and the external dynamic wrenches W^{int} and W_{ext}, the above equilibrium conditions can be equally written as:

$$W^{int}(c, \ddot{c}) = W^{ext}(p, f), \tag{4.4}$$

stating that the internal and external wrenches have to be in a permanent equilibrium.

4.2.2 Optimization Problem

With the above feasibility conditions, an optimization problem for the generation of dynamically equilibrated motions can be formulated as follows:

$$\begin{aligned}
&\underset{u}{\text{minimize}} && \int_{t0}^{t_f} \mathcal{L}(u,t)dt + \Phi(u,t_f) \\
&\text{subject to} && \\
&t \in [t_0, t_f] && W^{int}(u,t) - W^{ext}(u,t) = 0 \qquad (4.5)\\
&t \in [t_0, t_f] && f(u,t) \in \mathcal{A}(u,t). \qquad (4.6)
\end{aligned}$$

However, condition (4.5) is a set of nonlinear equality constraints, which can pose computational difficulties. To solve a problem with nonlinear equality constraints either penalty functions have to be employed or dual problems formulated. The choice of an appropriate penalty function is not an easy one though, and the resulting dual problems are not differentiable. One possible solution consists in reformulating the problem such that the nonlinearities are transferred either to inequality constraints or to the objective function $\mathcal{L}$.

4.2.3 Control parameterization

As already seen in Chapter 2, appropriate variables for the control of particles can be either their accelerations $\ddot{c}_i$ or the associated contact forces f_k^i:

$$u_i = \ddot{c}_i \leftrightarrow u_k^i = f_k^i, \tag{4.7}$$

and as well as for the linear case, the choice on the decision variables has a consequence on the well-posedness of the optimization problems.

Depending on the contact model, the set of feasible contact forces $\mathcal{A}$ can take different forms. For a linear friction model, $\mathcal{A}$ is a cross-product of revolution cones and can be approximated by a polytope A. The decision upon the parameterization (4.7) modifies its consideration inside the optimization problem. In the first case ($u = \ddot{c}$) conditions (4.5) and (4.6) can be transformed into a set of nonlinear inequalities (cf. Appendix A):

$$Af(u,t) \geq 0. \tag{4.8}$$

The other way around ($u_k^i = f_k^i$), feasible forces can be expressed as positive linear combinations of force vectors lying on the edges of the contact polyhedrons (cf. Figure 4.1):

$$f_k^i = \alpha_1 f_{k\_1}^i + ... + \alpha_M f_{k\_M}^i, \tag{4.9}$$

with M being its number of edges. The resulting contact torques τ_k, then also feasible, are equally defined by:

$$\tau_k^i = \alpha_1 \tau_{k\_1}^i + ... + \alpha_M \tau_{k\_M}^i, \tag{4.10}$$

so that the external wrench w_k applied at a contact point k can be written as:

$$w_k^i = \alpha_1 w_{k\_1}^i + ... + \alpha_M w_{k\_M}^i. \tag{4.11}$$

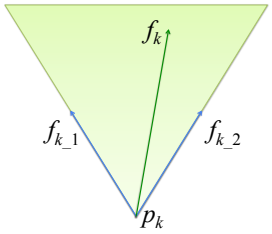


Figure 4.1: Every vector inside a convex polyhedral set can be expressed as a linear combination of the vectors lying on the edges of this set.

This second way offers us thus the possibility to satisfy condition (4.6) without the use of general inequalities but by variable bounds that are easier to handle inside an optimization routine. Then, however, we still face optimization problems with nonlinear equality constraints.

4.2.4 Eliminating the nonlinear coupling

Let's have a closer look at condition (4.5) for one single particle c and multiple, nonplanar contact points p_k. All conclusions made in the following also apply in the more general case of multiple particles. Condition (4.5), written out in full, is:

$$m \begin{pmatrix} \ddot{c}^x \\ \ddot{c}^y \\ \ddot{c}^z - g \\ c^x(\ddot{c}^z - g) - c^z \ddot{c}^x \\ c^y(\ddot{c}^z - g) - c^z \ddot{c}^y \\ c^y \ddot{c}^x - c^x \ddot{c}^y \end{pmatrix} = \sum_k \begin{pmatrix} f_k^x \\ f_k^y \\ f_k^z \\ p_k^x f_k^z - p_k^z f_k^x \\ p_k^y f_k^z - p_k^z f_k^y \\ p_k^y f_k^x - p_k^x f_k^y \end{pmatrix}. \tag{4.12}$$

We substitute for convenience the nonlinear terms on the left side by additional variables τ_+, τ_-, yielding:

$$m \begin{pmatrix} \ddot{c}^x \\ \ddot{c}^y \\ \ddot{c}^z - g \\ \tau_+^y - \tau_-^y \\ \tau_+^x - \tau_-^x \\ \tau_+^z - \tau_-^z \end{pmatrix} = \sum_k \begin{pmatrix} f_k^x \\ f_k^y \\ f_k^z \\ p_k^x f_k^z - p_k^z f_k^x \\ p_k^y f_k^z - p_k^z f_k^y \\ p_k^y f_k^x - p_k^x f_k^y \end{pmatrix}. \tag{4.13}$$

Then, the acceleration of a particle in one dimension, can be expressed in two different ways:

$$\ddot{c}^x = \frac{1}{m}\sum_k f_k^x, \tag{4.14}$$

and

$$\ddot{c}^x = \frac{\tau_+^z}{c^y} = \frac{\tau_+^z}{\tau_+^x}(\ddot{c}^z - g) = \frac{\tau_+^z}{\tau_+^x}\frac{1}{m}\sum_k f_k^z. \tag{4.15}$$

We can get a closer look at the inner coupling between the forces with:

$$\sum_k f_k^x = \frac{\tau_+^z}{\tau_+^x}\sum_k f_k^z, \tag{4.16}$$

or further developed:

$$\sum_k f_k^x = \frac{\tau_+^z}{\tau_+^x}\frac{\tau_+^y}{\tau_-^z}\frac{\tau_-^x}{\tau_-^y}\sum_k f_k^x, \tag{4.17}$$

and equivalently:

$$\tau_+^x\tau_-^z\tau_-^y = \tau_+^z\tau_+^y\tau_-^x. \tag{4.18}$$

As mentioned above, nonlinear equality constraints are difficult to handle inside an optimization problem, requiring this inner coupling to be eliminated.

In Chapter 2, this elimination has been achieved by assuming that friction is infinite. Since the vertical positions of the contact points were aligned with the inertial frame of reference ($p_k^z = 0$), any vertical contact torque τ^z could be generated by horizontal contact forces $f_k^{x/y}$ without affecting horizontal torques $\tau^{x/y}$. Reciprocally, any horizontal contact force was admissible, and the optimization problem (2.11) could be formulated by considering only the vertical contact forces f_k^z together with the following subset of the dynamic constraints (4.12):

$$m\begin{pmatrix} \ddot{c}^z - g \\ c^x(\ddot{c}^z - g) - c^z\ddot{c}^x \\ c^y(\ddot{c}^z - g) - c^z\ddot{c}^y \end{pmatrix} = \sum_k \begin{pmatrix} f_k^z \\ p_k^x f_k^z \\ p_k^y f_k^z \end{pmatrix}. \tag{4.19}$$

We can consequently express motions in three dimensions by three ordinary, decoupled differential equations:

$$\ddot{c}^z = \frac{1}{m}\sum_k f_k^z + g \tag{4.20}$$

$$\ddot{c}^x = -\frac{1}{c^z}(\frac{1}{m}\sum_k p_k^x f_k^z - c^x \sum_k f_k^z) \tag{4.21}$$

$$\ddot{c}^y = -\frac{1}{c^z}(\frac{1}{m}\sum_k p_k^y f_k^z - c^y \sum_k f_k^z). \tag{4.22}$$

Since we are interested in avoiding nonlinear equalities or inequalities, we will adapt in the following these suppositions and choose the contact forces as the control parameters, but, in order to consider also foot placements on non-planar ground, we have to extend the above equations by vertically not aligned contact points.

Again, in order for the above decoupling to be valid, we have to be able to generate any horizontal contact force $f_k^{x/y}$ without applying a torque in the chosen inertial reference frame. With the supposition that contact points cannot slide, such forces can equally be generated by one single or a subset of the total number of contact points, as soon as they are aligned with the origin. The above decoupling can therefore be kept valid also for multiple contact points on non-planar surfaces:

$$m\begin{pmatrix} \ddot{c}^z - g \\ c^x(\ddot{c}^z - g) - c^z\ddot{c}^x \\ c^y(\ddot{c}^z - g) - c^z\ddot{c}^y \end{pmatrix} = \sum_k \begin{pmatrix} f_k^z \\ p_k^x f_k^z \\ p_k^y f_k^z \end{pmatrix} + \sum_l \begin{pmatrix} f_l^z \\ p_l^x f_l^z - p_l^z f_l^x \\ p_l^y f_l^z - p_k^z f_l^y \end{pmatrix}, \tag{4.23}$$

where p_l are non-aligned contact points.

The horizontal forces $f_l^{x/y}$ that are applied at these contact points generate a torque and therefore have to be added to the control vector u, but, a positive side-effect of this is that friction limits can be considered for the associated solid [2].

Now, if we preview walking trajectories over several steps, we will inevitably encounter cases where the robot is supported by one single foot on a surface that is not aligned with the origin. All non-zero horizontal forces generate

2. The assumption of non-sliding contact points is common to all CoP-based walking generation approaches. Here however, this assumption has to be made only for a subset of contact points, allowing for a consideration of friction limits for the rest.

then a torque, such that no compensation can be made, and the generated trajectories might be not realizable. However, let's suppose that the contact points of the perfectly rigid foot are aligned horizontally, as is the case when the robot is mounting stairs. Then all horizontal contact forces share the same line of action, so that, with p_c^z the height of the points, we can state:

$$\sum_k p_k^z f_k^x = p_c^z \sum_k f_k^x = p_c^z m \ddot{c}^x, \tag{4.24}$$

$$\sum_k p_k^z f_k^y = p_c^z \sum_k f_k^y = p_c^z m \ddot{c}^y. \tag{4.25}$$

We get then:

$$m \begin{pmatrix} \ddot{c}^z - g \\ c^x(\ddot{c}^z - g) - \ddot{c}^x(c^z - p_c^z) \\ c^y(\ddot{c}^z - g) - \ddot{c}^y(c^z - p_c^z) \end{pmatrix} = \sum_k \begin{pmatrix} f_k^z \\ p_k^x f_k^z \\ p_l^y f_k^z \end{pmatrix}, \tag{4.26}$$

and consequently:

$$\ddot{c}^z = \frac{1}{m} \sum_k f_k^z + g \tag{4.27}$$

$$\ddot{c}^x = -\frac{1}{c^z - p_c^z} (\frac{1}{m} \sum_k p_k^x f_k^z - c^x \sum_k f_k^z) \tag{4.28}$$

$$\ddot{c}^y = -\frac{1}{c^z - p_c^z} (\frac{1}{m} \sum_k p_k^y f_k^z - c^y \sum_k f_k^z). \tag{4.29}$$

4.3 3D Control of the Center of Mass

We consider now the control of point-mass motions in the three-dimensional space with predefined foot positions. As in the previous chapters, the double support phases are taken into account implicitly, so that according to the considerations of the previous sections, the CoM dynamics are described by the following three nonlinear second-order differential equations:

$$\ddot{c}^z = \frac{1}{m} \sum_k f_k^z + g \tag{4.30}$$

$$\ddot{c}^x = -\frac{1}{c^z - p_c^z} (\frac{1}{m} \sum_k p_k^x f_k^z - c^x \sum_k f_k^z) \tag{4.31}$$

$$\ddot{c}^y = -\frac{1}{c^z - p_c^z} (\frac{1}{m} \sum_k p_k^y f_k^z - c^y \sum_k f_k^z). \tag{4.32}$$

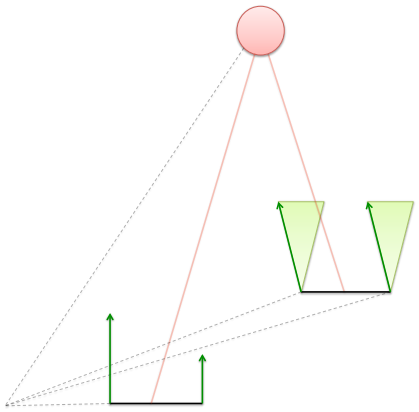


Figure 4.2: By neglecting maximum friction forces for one foot, we can assume that any tangential force can be compensated for, so that the external and internal momenta (4.1) and (4.2) are always equilibrated.

We subsume the position c and the velocity $\dot{c}$ of the CoM particle in the state vector s, and the components of the control vector u are the vertical contact forces. We can formulate then a set of discrete first-order difference equations:

$$s(t_{i+1}) = h(s(t_i), u), \tag{4.33}$$

together with the following bounds on the control vector u:

$$0 \leq u_k \leq \infty. \tag{4.34}$$

4.3.1 Inverted Pendulum Walk

A question, that has been answered in the previous chapters by the requirements on the linearity of the system, now has to be answered again: How to generate walking motions with one single particle? Or more precisely: What optimization criterions serve to generate CoM motions that lead to feasible and natural walking motions? Energy consumption has long been identified as a minimization criterion of the human walk, but one single particle lacks the complexity to reflect the exchange of work during walking.

Considering the simplicity of the CoM model, we are obliged to employ heuristics that are considered to be essential features of the human gait. So, what are these features? Two major observations have contributed to the

development of simplified models for walking. The first observation is that the human walk resembles a 'compass gait' where the CoM vaults over rigid legs. The second observation is that biological systems seem to prefer a walking style that leads to the characteristic two peaks of the vertical ground reaction force (GRF).

Since the observation that the human walk resembles a compass gait, the inverted pendulum model, which embodies the compass gait, has been recognized to be able to partially explain the dynamics of walking. Its capacity to generate stable locomotion has first been proven by [McGeer, 1990]. Several passive dynamic walkers have been conceived since then that are capable of walking down a slope with straight stance legs. This paradigm has been influential in [Van De Panne, 1997] for the motion control of computer animated walking figures. By minimizing the deviation of the leg length with respect to a nominal value inside an optimization-based routine, plausible walking, turning, leaping, and running motions could be generated. Finally, the Linearized Inverted Pendulum, that has first been introduced in [Kajita and Tani, 1991] and has become since then the basis of most online walking generation approaches in humanoid robotics, is a derivation of the inverted pendulum model.

However, due to the incompleteness of this model, we choose here only to imitate the straight leg walking. We are defining for this purpose a dominant minimization term on the leg length, which is approximated by $\|c - p\|$, with respect to the nominal length l_0:

$$\mathcal{L}(u, t_i) = \left(\|c(u, t_i) - p(t_i)\| - l_0\right)^2 . \tag{4.35}$$

An additional, weakly weighted 'inertial' term on the derivative of the forces serves to smoothen the evolution of the CoM trajectory (cf. Chapter 2):

$$\mathcal{L}(u, t_i) = ... \; + \frac{\alpha}{2} \left(u(t_i) - u(t_{i-1})\right)^2 . \tag{4.36}$$

To enforce stability, the position of the CoM is driven above the center of the foot by the end of the preview horizon:

$$\Phi(u, t_N) = \frac{\gamma}{2} \left(c(u, t_N) - p(t_N) - c_0\right)^2 , \tag{4.37}$$

with c_0 being an upright vector of the nominal leg length l_0.

The resulting optimal control problem becomes:

$$\underset{u}{\text{minimize}} \quad \sum_{i=1}^{N} \left(\|c(u,t_i) - p(t_i)\| - l_0 \right)^2 + \frac{\alpha}{2} \left(f(u,t_i) - f(u,t_{i-1}) \right)^2 + \quad (4.38)$$

$$\frac{\gamma}{2} \left(c(u,t_N) - p(u,t_N) - c_0 \right)^2, \quad (4.39)$$

subject to the model difference equation:

$$s(t_{i+1}) = g(s(t_i), u), \quad (4.40)$$

and bounds on the variables:

$$0 \leq u \leq u_{max}, \quad (4.41)$$

that assure the dynamic feasibility.

During the following simulation, the robot is walking forward, making a step every 0.8 s of 20 cm length and width. Three vertical steps of 4 cm and 2 cm height have been considered between the second and the sixth step. As in the previous chapters, the double support phase of 100 ms is neglected by sampling only inside single support phases. The total mass m_c of the robot is 50 kg. The nominal leg length is 1 m. The foot length is 20 cm and the foot width 10 cm. The simulation parameters have been chosen as follows:

- $\alpha = 10^{-6}$
- $\gamma = 10^3$
- $N = 10$

We can observe in Figure 4.3 the expected inverted pendulum like evolution of the vertical CoM position. The robot was able to mount the steps in the middle of the course. For higher steps however, a modification of the cost-function needs to be done, since the dominant leg-length term prohibits a greater variation of the leg length.

The ability of the above MPC scheme to handle nonlinear differential motion equations makes the predefinitions of the previous chapters unnecessary. We can thus think of liberating the foot positions, making the support durations variable, turning without decoupling the preview of the orientations, considering the double support phases, etc. In the following, we are going to focus on the second important feature of the human walk that has been mentioned in the beginning of this section: the natural ground reaction force pattern.

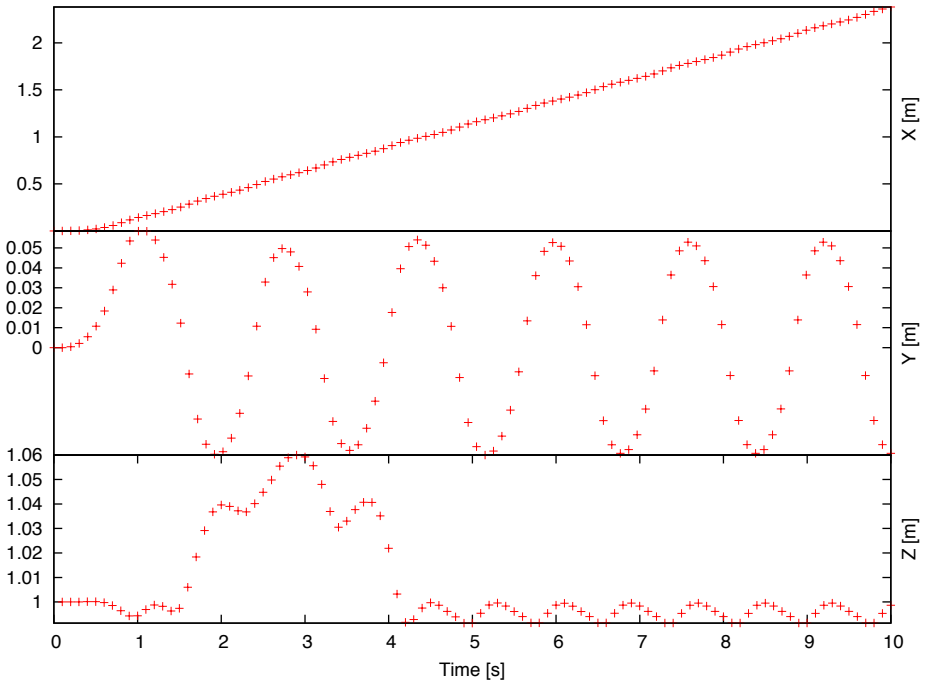


Figure 4.3: The evolution of the CoM for an enforced nominal leg length and predefined foot positions.

4.3.2 Compliant leg behavior

The force curve of the motions generated above shows pronounced peaks between single support phases (Figure 4.4). As a consequence of the inverted-pendulum-like behavior, the redirection of the CoM requires high efforts during a relatively short time. Biological walking shows a considerably smoother force profile. The natural CoM curve follows a sinusoidal-like vertical path that leads to the characteristic double-peaked ground force pattern under the stance leg. Compliant leg behavior has been identified as being able to explain this observation, and the natural CoM motions fit the behavior of the *spring loaded inverted pendulum* (SLIP) especially well for running motions:

$$m\ddot{c} = k\left(\frac{l_0}{\|c-p\|} - 1\right)(c-p) + m\vec{g}. \tag{4.42}$$

[Geyer et al., 2006] have shown that the same model can reproduce the vertical and horizontal GRF patterns also for normal walking, although not equally well for arbitrary walking speeds. By generating high impact forces, it would additionally affect the stability of the system. Here again, we are therefore not

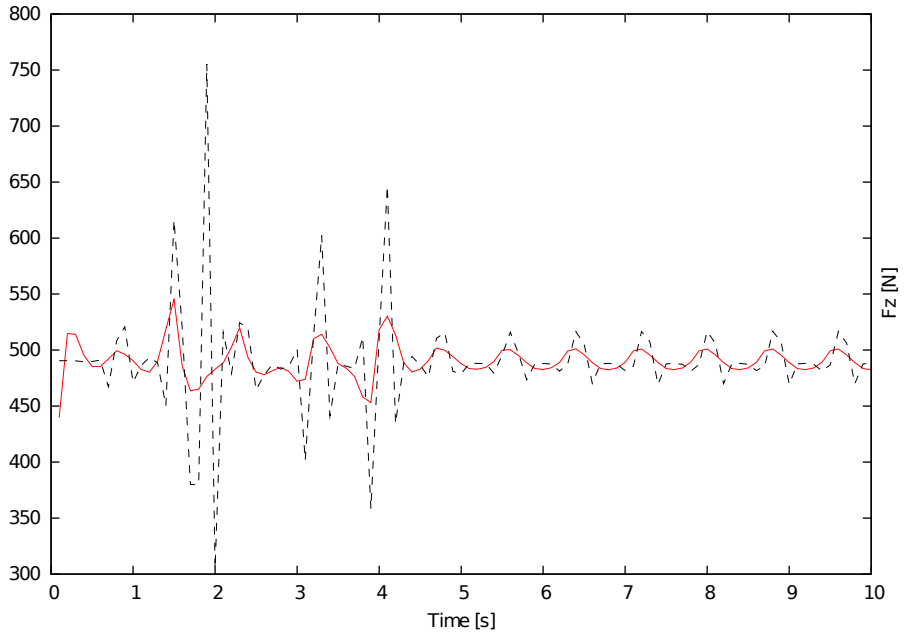


Figure 4.4: Sum of vertical ground reaction forces with (solid red) and without (dashed black) enforced compliance.

interested in reproducing the entire dynamics of a spring.

There is, however, no doubt that compliance in the legs explains an important part of the leg behavior not only during running but also for walking. We are going therefore to enforce the behavior of a spring to some extent by a weakly weighted minimization of the difference between the acceleration of the CoM particle and the acceleration of the above mass-spring system:

$$\mathcal{L}(u,t_i) = ... + \\ \frac{\beta}{2}\left(\ddot{c}(u,t_i) - \frac{k}{m}\left(\frac{l_0}{\|c(u,t_i) - p(t_i)\|} - 1\right)(c(u,t_i) - p(u,t_i)) - \vec{g}\right)^2. \quad (4.43)$$

Figure 4.4 shows the evolution of the ground reaction forces for the same scenario as above, but with $k = 2000$ kg/s^2 and $\beta = 4 * 10^{-2}$. We can observe that the force peaks are smaller for the entire simulation, but especially during climbing stairs. Additionally, we find the sinusoidal force shape indicating a sinusoidal shape also for $c^z(t)$.

The control objectives chosen in this section respond to the observed characteristics of the walk, but observations are different for running, and very

simple models, like those introduced in this thesis, lack the complexity to allow for a handing of both modes of locomotions through a unique control law. Running, climbing stairs, or sitting down on a chair, therefore, require different approaches. The control parametrization and the simplifications of this chapter, on the contrary, are more generic and can serve as the basis for efficient generation of three-dimensional motions via NMPC.

To obtain stable motions in the above simulations, two SQP iterations where necessary, making an average computational time of 30 ms on a 2.25 GHz Intel Core 2 Duo. Although real-time capable on a modern computer, the computational load is still too large for embedded hardware used in contemporary robots such that the question of how to reduce the computational time has to be addressed.

Conclusion

Generic motion control by means of Model Predictive Control involves the resolution of nonlinear optimization problems. In this chapter, an efficient formulation for the generation of motions in the three-dimensional space has been proposed. By controlling the contact wrench of a particle, dynamic feasibility can be assured by bounds on the control variables. The consideration of friction forces is possible for a subset of the contact points.

Stable walking has been achieved by enforcing a nominal leg length. Additionally, enforced compliance in the legs served to generate more natural patterns of the ground reaction forces and CoM motions.

Although the goal of real-time trajectory generation has been achieved on a modern PC, the computational time is still beyond the available machine time of embedded hardware, and further insight into the numerics is necessary.

Résumé

Le contrôle générique des mouvements par moyen de la Commande Prédictive inclut la résolution des problèmes d'optimisation non-linéaires. Dans ce chapitre, une formulation efficace pour la génération des mouvements dans l'espace 3D a été proposé. En commandant les torseurs de contact d'un particule, la faisabilité dynamique peut être assuré par des bornes sur les variables de contrôle. La prise en compte des forces de frottement est possible dans un sous-ensemble des points de contact.

La marche stable a été atteinte en renforçant la longueur nominale de la jambe. De plus, une flexibilité dans les jambes a servi de générer des caractéristiques des forces de réaction et des mouvements du CdM plus naturels.

Même si l'objectif de générer des trajectoires en temps réels a été atteint sur un ordinateur moderne, le temps de calcul est toujours en dehors le temps de calcul des systèmes embarqués, et un aperçu plus précis dans la résolution numérique des problèmes posés ici est nécessaire.

4.4 Appendix D: Feasibility by inequality constraints

We approximate the friction cones

$$\|f_k^t\| \leq \mu f_k^n, \tag{4.44}$$

that emerge when considering the linear friction law, by inscribed four-sided pyramids. Then, sliding of one single contact point k can be avoided if:

$$A' f_k = \begin{pmatrix} 1 & 0 & \mu \\ -1 & 0 & \mu \\ 0 & 1 & \mu \\ 0 & -1 & \mu \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f_k^x \\ f_k^y \\ f_k^z \end{pmatrix} \geq 0, \tag{4.45}$$

and the totality of points if:

$$Af = \begin{pmatrix} \mathbb{1} & \mathbb{0} & \mu\mathbb{1} \\ -\mathbb{1} & \mathbb{0} & \mu\mathbb{1} \\ \mathbb{0} & \mathbb{1} & \mu\mathbb{1} \\ \mathbb{0} & -\mathbb{1} & \mu \\ \mathbb{0} & \mathbb{0} & \mathbb{1} \end{pmatrix} \begin{pmatrix} f^x \\ f^y \\ f^z \end{pmatrix} \geq 0, \tag{4.46}$$

where f subsumes the forces vectors f_k of all points.

With condition 4.12 written as:

$$\mathbb{C} = \mathbb{W} \begin{pmatrix} f^x \\ f^y \\ f^z \end{pmatrix}, \tag{4.47}$$

where

$$\mathbb{C} = \begin{pmatrix} \ddot{c}^x \\ \ddot{c}^y \\ \ddot{c}^z - g \\ c^x(\ddot{c}^z - g) - c^z \ddot{c}^x \\ c^y(\ddot{c}^z - g) - c^z \ddot{c}^y \\ c^x \ddot{c}^y - c^y \ddot{c}^x \end{pmatrix} \text{ and } \mathbb{W} = \begin{pmatrix} \mathbb{1} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{1} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{1} \\ -\mathbb{P}_z & \mathbb{0} & \mathbb{P}_x \\ \mathbb{0} & -\mathbb{P}_z & \mathbb{P}_y \\ \mathbb{P}_y & \mathbb{P}_x & \mathbb{0} \end{pmatrix}, \tag{4.48}$$

the dynamic equilibrium is assured if:

$$A\mathbb{W}^+\mathbb{C} \geq 0. \tag{4.49}$$

Final Discussion

The ZMP Preview Control scheme, first introduced in [Kajita et al., 2003], was unique in several ways. For the first time Model Predictive Control was applied successfully to online generation of walking trajectories. Stable motions were obtained by the sole minimization of multiple performance criterions for a short preview period, without strong suppositions on the CoM trajectory. Due to the linearity of this scheme, the control could be computed sufficiently fast to allow for robust and reactive walking (cf. [Nishiwaki and Kagami, 2008]). This scheme, however, didn't break with the common practice of assuring realizability of motions. Reference patterns for the evolution of contact forces still had to be determined prior to obtaining walking motions.

A shift in this paradigm had been initiated in [Wieber, 2006]. Explicit consideration of feasibility requirements had allowed to improve the robustness and to generate motions without predefined CoP references. This contribution made use of the most valuable property of Model Predictive Control, the explicit consideration of limitations, but still, motions were determined to a large extent by the placement of the feet. This deficiency could be overcome in [Diedam et al., 2008]. For the first time, foot placement could be decided online, in permanent accordance with the dynamic feasibility requirements of the generated motions, but again, foot position references had to be given.

In Chapter 2, we have completed this change of approach, away from predetermined contact force and foot references, toward a more elevated control of the Center of Mass. We have shown that real-time control of a robot by a desired velocity reference is not only possible, but also improves its robustness and reactivity. The such enhanced level of abstraction allowed for a seamless implementation inside a Visual Servoing scheme in [Dune et al., 2010].

We have improved the efficiency of the LMPC-based trajectory generator in

Chapter 3 by incorporating the explicit consideration of geometric limitations. Polyhedral constraints on the position of the CoM help to ensure the realizability of walking trajectories for arbitrary vertical oscillations of the CoM or unfeasible displacement references. Subsequently, these constraints made it possible to generate feasible toe motions. As a consequence, a larger maximal displacement speed, as well as more naturally looking and more energetically efficient walking motions, have been achieved. Additionally, multi-point models can improve the accuracy of the prediction and thus the achievable walking performance.

To reduce the dependency on restricting predefinitions, our approach of generating walking motions by means of simplified models has been extended to nonlinear MPC in Chapter 4. We have established a formulation that allows to consider the dynamic limitations of the robot through simple bounds on the variables. This scheme has proven to be capable of generating motion trajectories in real-time, and we were able to achieve three-dimensional walking motions on non-planar ground. Due to enforced compliance in the legs, the scheme managed additionally to reproduce the natural profiles of the CoM and the contact forces observed in human walking.

Various possibilities to carry on the MPC approach taken in this thesis can be distinguished on the horizon. Explicit consideration of external constraints allows to cope with the highly structured and dynamic environment of humans. The consideration of contact forces allows for interaction with external objects other than the ground. The high efficiency of existing resolution methods and the resulting reactivity might allow for stabilizing control on the level of the trajectory generation. Although, in order to lower the computational time to the microsecond range, the numerical aspect of nonlinear Model Predictive Control requires special attention, with its capacity to handle nonlinear equations, NMPC has the potential to further release from predefinitions that limit the flexibility of the approach. But still, the full potential of linear MPC for walking control has not yet been fully exploited, as could be shown also by [Dimitrov et al., 2011], and contributions that the author does not foresee will surely follow.

Bibliography

[Adamczyk et al., 2006] Adamczyk, P. G., Collins, S. H., and Kuo, A. D. (2006). The advantages of a rolling foot in human walking. *Journal of Experimental Biology*, 209(20):3953–3963.

[Adamczyk and Kuo, 2009] Adamczyk, P. G. and Kuo, A. D. (2009). Redirection of center-of-mass velocity during the step-to-step transition of human walking. *Journal of Experimental Biology*, 212(16):2668–2678.

[Alamir and Murilo, 2008] Alamir, M. and Murilo, A. (2008). Swing-up and stabilization of a twin-pendulum under state and control constraints by a fast nmpc scheme. *Automatica*, 44(5):1319 – 1324.

[Aubin, 1991] Aubin, J. P. (1991). *Viability Theory*. Birkhäuser.

[Bellman, 1957] Bellman, R. (1957). *Dynamic Programming*. Princeton, N. J. : Princeton University Press.

[Buschmann, 2010] Buschmann (2010). *Simulation and Control of Biped Walking Robots*. PhD thesis, Technische Universität München.

[Chaumette and Hutchinson, 2006] Chaumette, F. and Hutchinson, S. (2006). Visual servo control, Part I: Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4):82–90.

[Dalibard et al., 2009] Dalibard, S., Nakhaei, A., Lamiraux, F., and Laumond, J.-P. (2009). Whole-body task planning for a humanoid robot: a way to integrate collision avoidance. In *IEEE/RAS Int. Conf. on Humanoid Robots (Humanoids'09)*, pages 355 –360.

[Diedam et al., 2008] Diedam, H., Dimitrov, D., Wieber, P.-B., Mombaur, K., and Diehl, M. (2008). Online Walking Gait Generation with Adaptive Foot Positioning through Linear Model Predictive Control. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'08)*, Nice, France.

[Diehl et al., 2005a] Diehl, M., Bock, H., and Schlode, J. (2005a). A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM Journal on Control and Optimization*, 43(5):1714–1736.

[Diehl et al., 2005b] Diehl, M., Bock, H. G., Diedam, H., and Wieber, P.-B. (2005b). Fast Direct Multiple Shooting Algorithms for Optimal Robot Control. In *Fast Motions in Biomechanics and Robotics*, Heidelberg, Allemagne.

[Diehl et al., 2009] Diehl, M., Ferreau, H. J., and Haverbeke, N. (2009). Efficient numerical methods for nonlinear mpc and moving horizon estimation.

[Dimitrov et al., 2011] Dimitrov, D., Paolillo, A., and Wieber, P.-B. (2011). Walking motion generation with online foot position adaptation based on l1- and l∞-norm penalty formulations. In *IEEE Int. Conf. on Robotics and Automation (ICRA'11)*, Shanghai, Chine.

[Donelan et al., 2002] Donelan, J. M., Kram, R., and Kuo, A. D. (2002). Mechanical work for step-to-step transitions is a major determinant of the metabolic cost of human walking. *Journal of Experimental Biology*, 205(23):3717–3727.

[Dune et al., 2010] Dune, C., Herdt, A., Stasse, O., Wieber, P.-B., Yokoi, K., and Yoshida, E. (2010). Cancelling the sway motion of dynamic walking in visual servoing. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'10)*, Taipei, Taiwan, Province Of China.

[Franken et al., 2008] Franken, M., van Oort, G., and Stramigioli, S. (2008). Analysis and simulation of fully ankle actuated planar bipedal robots. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'08)*, pages 634–639.

[Geyer et al., 2006] Geyer, H., Seyfarth, A., and Blickhan, R. (2006). Compliant leg behaviour explains basic dynamics of walking and running. *Proceedings of the Royal Society of London, Series B: Biological Sciences*, 273:2861–2867.

[Gordon et al., 2009] Gordon, K. E., Ferris, D. P., and Kuo, A. D. (2009). Metabolic and mechanical energy costs of reducing vertical center of mass movement during gait. *Archives of Physical Medicine and Rehabilitation*, 90(1):136–144.

[Harada et al., 2004] Harada, K., Kajita, S., Kaneko, K., and Hirukawa, H. (2004). An analytical method on real-time gait planning for a humanoid robot. In *IEEE/RAS Int. Conf. on Humanoid Robots (Humanoids'04)*, pages 640–655.

[Herr and Popovic, 2008] Herr, H. and Popovic, M. (2008). Angular momentum in human walking. *J Exp Biol*, 211(4):467–481.

[Hirukawa et al., 2006] Hirukawa, H., Hattori, S., Harada, K., Kajita, S., Kaneko, K., Kanehiro, F., Fujiwara, K., and Morisawa, M. (2006). A universal stability criterion of the foot contact of legged robots - adios zmp. In *IEEE Int. Conf. on Robotics and Automation (ICRA'06)*, pages 1976 –1983.

[Hofmann et al., 2009] Hofmann, A., Popovic, M., and Herr, H. (2009). Exploiting angular momentum to enhance bipedal center-of-mass control. In *IEEE Int. Conf. on Robotics and Automation (ICRA'09)*, pages 4423 –4429.

[Houska et al., 2011] Houska, B., Ferreau, H., and Diehl, M. (2011). An auto-generated real-time iteration algorithm for nonlinear mpc in the microsecond range. *Automatica*.

[Kajita et al., 2007] Kajita, S., Kajita, S., Kaneko, K., Morisawa, M., Nakaoka, S., and Hirukawa, H. (2007). ZMP-based biped running enhanced by toe springs. In *IEEE Int. Conf. on Robotics and Automation (ICRA'07)*, pages 3963–3969.

[Kajita et al., 2003] Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., and Hirukawa, H. (2003). Biped walking pattern generation by using preview control of zero-moment point. In *IEEE Int. Conf. on Robotics and Automation (ICRA'03)*, volume 2, pages 1620–1626.

[Kajita et al., 2001] Kajita, S., Matsumoto, O., and Saigo, M. (2001). Real-time 3d walking pattern generation for a biped robot with telescopic legs. In *IEEE Int. Conf. on Robotics and Automation (ICRA'01)*, pages 2299–2306.

[Kajita and Tani, 1991] Kajita, S. and Tani, K. (1991). Study of dynamic biped locomotion on rugged terrain-theory and basic experiment. In *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, pages 741 –746 vol.1.

[Kuffner et al., 2002] Kuffner, J., Kagami, S., Nishiwaki, K., Inaba, M., and Inoue, H. (2002). Dynamically-stable motion planning for humanoid robots. *Autonomous Robots*, 12(1):105–118.

[Kuo, 2007] Kuo, A. (2007). The six determinants of gait and the inverted pendulum analogy: A dynamic walking perspective. *Human Movement Science*, 26(4):617–656.

[Kuo, 2002] Kuo, A. D. (2002). Energetics of actively powered locomotion using the simplest walking model. *Journal of Biomechanical Engineering*, 124(1):113–120.

[Kuo et al., 2005] Kuo, A. D., Donelan, J. M., and Ruina, A. (2005). Energetic consequences of walking like an inverted pendulum: Step-to-step transitions. *Exercise and Sport Sciences Reviews*, 33(2).

[Lau et al., 2009] Lau, M., Yue, S., Ling, K., and Maciejowski, J. (2009). A comparison of interior point and active set methods for fpga implementation of model predictive control. In *Proc. European Control Conference*, Budapest. European Union Control Association.

[Mayne et al., 2000] Mayne, D., Rawlings, J., Rao, C., and Scokaert, P. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789 – 814.

[McGeer, 1990] McGeer, T. (1990). Passive dynamic walking. *The International Journal of Robotics Research*, 9(2):62–82.

[Miura et al., 2011] Miura, K., Morisawa, M., Kanehiro, F., Kajita, S., Kaneko, K., and Yokoi, K. (2011). Human-like walking with toe supporting for humanoids. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'11)*, pages 4428–4435.

[Morari and Lee, 1999] Morari, M. and Lee, J. H. (1999). Model predictive control: past, present and future. *Computers and Chemical Engineering*, 23(4-5):667 – 682.

[Mordatch et al., 2010] Mordatch, I., Lasa, M. D., and Hertzmann, A. (2010). Robust physics-based locomotion using low-dimensional planning. *ACM Transactions on Graphics*, 29(4):1.

[Morisawa et al., 2006] Morisawa, M., Harada, K., Kajita, S., Kaneko, K., Kanehiro, F., Fujiwara, K., Nakaoka, S., and Hirukawa, H. (2006). A biped

pattern generation allowing immediate modification of foot placement in real-time. In *IEEE/RAS Int. Conf. on Humanoid Robots (Humanoids'06)*, pages 581–586.

[Morisawa et al., 2007] Morisawa, M., Harada, K., Kajita, S., Nakaoka, S., Fujiwara, K., Kanehiro, F., Kaneko, K., and Hirukawa, H. (2007). Experimentation of humanoid walking allowing immediate modification of foot place based on analytical solution. In *IEEE Int. Conf. on Robotics and Automation (ICRA'07)*, pages 3989 –3994.

[Nagasaka et al., 2004] Nagasaka, K., Kuroki, Y., Suzuki, S., Itoh, Y., and Yamaguchi, J. (2004). Integrated motion control for walking, jumping and running on a small bipedal entertainment robot. In *IEEE Int. Conf. on Robotics and Automation (ICRA'04)*, volume 4, pages 3189 – 3194 Vol.4.

[Nagasaki et al., 2004] Nagasaki, T., Kajita, S., Kaneko, K., Yokoi, K., and Tanie, K. (2004). A running experiment of humanoid biped. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'04)*, volume 1, pages 136 – 141 vol.1.

[Nicolao et al., 2000] Nicolao, G. D., Magni, L., and Scattolini, R. (2000). Stability and robustness of nonlinear receding horizon control.

[Nishiwaki and Kagami, 2008] Nishiwaki, K. and Kagami, S. (2008). Short cycle pattern generation for online walking control system of humanoids.

[Nishiwaki and Kagami, 2009] Nishiwaki, K. and Kagami, S. (2009). Online walking control system for humanoids with short cycle pattern generation. *Int. J. Rob. Res.*, 28:729–742.

[Nishiwaki and Kagami, 2010] Nishiwaki, K. and Kagami, S. (2010). Strategies for adjusting the zmp reference trajectory for maintaining balance in humanoid walking. In *IEEE Int. Conf. on Robotics and Automation (ICRA'10)*, pages 4230 –4236.

[Nishiwaki et al., 2002a] Nishiwaki, K., Kagami, S., Kuniyoshi, Y., Inaba, M., and Inoue, H. (2002a). Online generation of humanoid walking motion based on a fast generation method of motion pattern that follows desired zmp. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'02)*, pages 2684–2689.

[Nishiwaki et al., 2002b] Nishiwaki, K., Kagami, S., Kuniyoshi, Y., Inaba, M., and Inoue, H. (2002b). Toe joints that enhance bipedal and fullbody motion of humanoid robots. In *IEEE Int. Conf. on Robotics and Automation (ICRA'02)*, pages 3105–3110.

[Ortega and Farley, 2005] Ortega, J. D. and Farley, C. T. (2005). Minimizing center of mass vertical movement increases metabolic cost in walking. *Journal of Applied Physiology*, 99(6):2099–2107.

[Popovic et al., 2004] Popovic, M., Hofmann, A., and Herr, H. (2004). Angular momentum regulation during human walking: biomechanics and control. In *IEEE Int. Conf. on Robotics and Automation (ICRA'04)*, volume 3, pages 2405 – 2411 Vol.3.

[Pratt et al., 2006] Pratt, J., Carff, J., Drakunov, S., and Goswami, A. (2006). Capture point: A step toward humanoid push recovery. In *IEEE/RAS Int. Conf. on Humanoid Robots (Humanoids'06)*, pages 200 –207.

[Primbs et al., 1997] Primbs, J. A., Primbs, J. A., and Nevistic, V. (1997). Constrained finite receding horizon linear quadratic control. Technical report, California Institute of Technology.

[Scholz and Schöner, 1999] Scholz, J. P. and Schöner, G. (1999). The uncontrolled manifold concept: identifying control variables for a functional task. *Experimental Brain Research*, 126(3):289–306.

[Sellaouti et al., 2006] Sellaouti, R., Stasse, O., Kajita, S., Yokoi, K., and Kheddar, A. (2006). Faster and smoother walking of humanoid HRP-2 with passive toe joints. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'06)*, pages 4909–4914.

[Siciliano et al., 2009] Siciliano, B., Sciavicco, L., and Villani, L. (2009). *Robotics: modelling, planning and control*. Advanced textbooks in control and signal processing. Springer.

[Srinivasan and Ruina, 2006] Srinivasan, M. and Ruina, A. (2006). Computer optimization of a minimal biped model discovers walking and running. *Nature*, 439(7072):72–75.

[Tajima et al., 2009] Tajima, R., Honda, D., and Suga, K. (2009). Fast running experiments involving a humanoid robot. In *IEEE Int. Conf. on Robotics and Automation (ICRA'09)*, pages 1571–1576.

[Takenaka et al., 2009] Takenaka, T., Matsumoto, T., and Yoshiike, T. (2009). Real time motion generation and control for biped robot -1st report: Walking gait pattern generation-. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'09)*, pages 1084–1091.

[Tedrake, 2004] Tedrake, R. (2004). *Applied optimal control for dynamically stable legged locomotion*. PhD thesis, Massachusetts Institute of Technology. Dept. of Electrical Engineering and Computer Science.

[Terada and Kuniyoshi, 2007] Terada, K. and Kuniyoshi, Y. (2007). Online gait planning with dynamical 3d-symmetrization method. In *IEEE/RAS Int. Conf. on Humanoid Robots (Humanoids'07)*, pages 222–227.

[Tlalolini et al., 2009] Tlalolini, D., Chevallereau, C., and Aoustin, Y. (2009). Comparison of different gaits with rotation of the feet for a planar biped. *Robotics and Autonomous Systems*, 57(4):371–383.

[Van De Panne, 1997] Van De Panne, M. (1997). From footprints to animation. *Computer Graphics Forum*, 16(4):211–223.

[Wieber, 2000] Wieber, P.-B. (2000). *Modélisation et commande d'un robot marcheur anthropomorphe*. These, École Nationale Supérieure des Mines de Paris.

[Wieber, 2005] Wieber, P.-B. (2005). Holonomy and nonholonomy in the dynamics of articulated motion. In *Proceedings of the Ruperto Carola Symposium on Fast Motion in Biomechanics and Robotics*.

[Wieber, 2006] Wieber, P.-B. (2006). Trajectory free linear model predictive control for stable walking in the presence of strong perturbations. In *IEEE/RAS Int. Conf. on Humanoid Robots (Humanoids'06)*.

[Wieber, 2008] Wieber, P.-B. (2008). Viability and Predictive Control for Safe Locomotion. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'08)*, Nice, France.

[Wieber and Chevallereau, 2004] Wieber, P.-B. and Chevallereau, C. (2004). Online adaptation of reference trajectories for the control of walking systems. Rapport de recherche RR-5298, INRIA.

Commande prédictive d'un robot humanoïde

Résumé : L'étendue des mouvements que les robots humanoïdes peuvent réaliser est fortement limitée par des contraintes dynamiques. Une loi de commande qui ne prend pas en compte ces restrictions, d'une manière ou d'une autre, ne va pas réussir à éviter une chute. La commande prédictive est capable de considérer les contraintes sur l'état et le contrôle de manière explicite, ce qui la rend particulièrement appropriée pour le contrôle des mouvements des robots marcheurs.

Nous commençons par dévoiler la structure spécifique de ces contraintes, démontrant notamment l'importance des appuis au sol. Nous développons ensuite une condition suffisante pour l'évitement d'une chute et nous proposerons une loi de commande prédictive qui y réponde. Cette formulation nous sert ensuite pour la conception des contrôleurs pratiques, capables d'un contrôle plus efficace et plus robuste de la marche humanoïde.

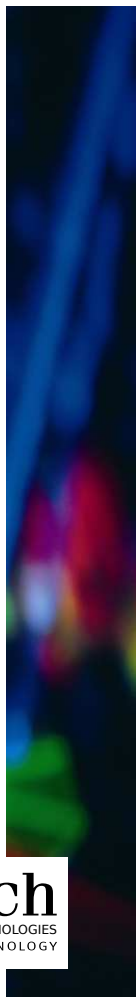
Mots clés : Robots Marcheurs Humanoïdes, Commande Prédictive, Masse Ponctuelle, Orteils, Génération des Trajectoires, Stabilité

Model Predictive Control of a humanoid robot

Abstract: The range of motions that humanoid robots are able to realize is strongly limited by inherent dynamical constraints so that any control law that does not consider these limitations, in one way or another, will fail to avoid falling. The Model Predictive Control (MPC) technique is capable of handling constraints on the state and the control explicitly, which makes it highly apt for the control of walking robots.

We begin by unveiling the specific structure of these constraints, stressing especially the importance of the supports on the ground. We give thereupon a sufficient condition for keeping balance and formulate an MPC law that complies with it. This formulation serves us then for the design of practicable controllers capable of more efficient and more robust control of the humanoid walk.

Keywords: Humanoid Walking Robots, Model Predictive Control, Point Mass, Toes, Trajectory Generation, Stability



MINES ParisTech

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY