Effective Generation of Dynamically Balanced Locomotion with Multiple Non-coplanar Contacts

Nicolas Perrin, Darwin Lau and Vincent Padois

Abstract Studies of computationally and analytically convenient approximations of rigid body dynamics have brought valuable insight into the field of humanoid robotics. Additionally, they facilitate the design of effective walking pattern generators. Going further than the classical Zero Moment Point-based methods, this paper presents two simple and novel approaches to solve for 3D locomotion with multiple non-coplanar contacts. Both formulations use model predictive control to generate dynamically balanced trajectories with no restrictions on the center of mass height trajectory. The first formulation treats the balance criterion as an objective function, and solves the control problem using a sequence of alternating convex quadratic programs. The second formulation considers the criterion as constraints, and solves a succession of convex quadratically constrained quadratic programs.

1 Introduction

Locomotion is a challenging task for humanoid robots, even within completely known environments. Although the dynamics of multibody systems are well understood, direct approaches to resolve the motion of all degrees of freedom for locomotion have remained computationally intractable due to the large number of degrees of freedom and nonlinear behaviour. To generate dynamically balanced locomotion trajectories efficiently, current methods are typically based on simplified models of the robot dynamics.

One of the most widespread model is the Inverted Pendulum Model (*IPM*) [10], where the mass of the robot is assumed to be concentrated at its center of mass (*CoM*). Although more accurate models are sometimes required [11], the simple IPM is suitable for many situations where the rotational inertial effects of the robot

N. Perrin, D. Lau & V. Padois
Institut des Systèmes Intelligents et de Robotique (ISIR), CNRS UMR 7222 & Université Pierre et Marie Curie, Paris, France, e-mail: {perrin, lau, padois}@isir.upmc.fr

arms, legs and torso are negligible or can compensate for each other. An example of such a scenario includes walking at a moderately fast pace.

One of the main properties of multi-contact locomotion is that the robot's CoM acceleration depends only on the contact forces with the environment. Hence, an important objective for the generation of locomotion is to design a CoM trajectory that is dynamically balanced at any point in time. *Dynamically balanced* refers to the existence of contact forces that can produce the desired CoM acceleration while respecting the contact constraints, such as being within the friction cones.

When all the contacts are coplanar, a point of particular interest, sometimes called the Zero Moment Point (*ZMP*) [17], can be defined. Although this name has led to confusion about its physical nature [3], the term ZMP will be used in the remainder of the work. A system is considered dynamically balanced if the ZMP lies within the support region, i.e. the 2D convex hull of the contact points. In the IPM, the relationship between the CoM and ZMP dynamics is defined by nonlinear differential equations. However, if the vertical displacements of the CoM are set in advance, the relationship can be decoupled and expressed as linear differential equations. This substantially simplifies the problem into one that can be more efficiently and easily implemented for locomotion trajectory generation and gait control.

The trajectory generation problem has been studied through analytical approaches [4], and more recently, using convex optimization with constraints on the ZMP in discrete time [7, 19, 20, 6]. The efficiency of convex optimization solvers allows them to be used within model predictive control (*MPC*) schemes and with the potential of real-time implementation for reactive walking. However, the ZMP+MPC approach suffers from two main drawbacks: 1) the trajectory of the CoM height (z-direction) must be known or fixed in advance, and 2) the contact points must always be coplanar, making this approach unsuitable for walking in complex unstructured environments or if the arms are to also be used. Due to these two restrictions, the ZMP+MPC approach can be regarded as a 2D CoM planner that operates in the horizontal xy-plane only.

Several studies have looked into extending the concept of ZMP into 3D conditions, such as GZMP [5] and 3DZMP [9], to handle non-coplanar contact points. These criteria have been used in control algorithms to maintain the dynamical balance of a humanoid robot interacting with its environment. However, no locomotion trajectory generation algorithm has been designed based on these notions. In [8], a more general criterion than the ZMP was proposed to evaluate the balance of contacts during the motion of a legged robot. This criterion was used within a preview control algorithm with additional restrictions, for example, the vertical motion of the CoM must be approximately constant. Furthermore, a preliminary phase is needed to plan the inertia and gravity wrenches appropriately, which is a difficult problem.

In this paper, two simple and novel MPC approaches to solve for 3D locomotion with multiple non-coplanar contacts are presented. The 3D condition for dynamically balanced gait allows for non-coplanar multiple contacts and no restrictions on the CoM height trajectory. Using the proposed 3D dynamically balanced criterion, the first MPC formulation treats the criterion as an objective function, where the resulting non-convex MPC problem is solved using a sequence of alternating (convex) quadratic programs (*AQP*). The second formulation considers the criterion as non-convex constraints to the problem, and is solved through a succession of convex QCQPs. The results for simple locomotion scenarios show the promise in using the proposed 3D condition to generate the CoM trajectory within the control framework of robots with multiple contacts.

Although generalizations, such as allowing for multiple contacts, non-coplanar contacts and not predetermining height trajectory, inherently increase the problem complexity, care has been taken to maintain balance between the computational efficiency and limitations of the model. Indeed, our approach is more general than traditional simplified methods based on the IPM or ZMP, but thanks to our main assumption, namely, that the CoM lies inside all the contact friction cones, the number of variables that have to be taken into account is much smaller than with direct methods, such as [15]. The feature of the proposed MPC schemes is that they can be effectively and efficiently solved through a succession of convex optimization problems, when the problem formulation is of particular forms, such as bilinear problems or non-convex QCQPs.

The remainder of the paper is organised as follows: Section 2 formulates the 3D condition for dynamically balance. The two MPC formulations using the 3D condition as an objective or as constraints are formulated in Sections 3 and 4, respectively. Section 5 presents a brief discussion on the two approaches. Finally, Section 6 concludes the paper and presents areas of future work.

2 Conditions for Dynamically Balanced Locomotion in 3D

Expressing wrenches with respect to the CoM of the robot $\mathbf{x}$, the Newton-Euler equations of motion for a multibody robot system in a fixed world frame can be written as

$$\mathbf{W}_x^{gravity} + \mathbf{W}_x^{contact} = \begin{bmatrix} \dot{\mathbf{L}} \\ M\ddot{\mathbf{x}} \end{bmatrix} , \tag{1}$$

where $\mathbf{W}_x^{gravity}$ is the gravity wrench in 6D vector notation [2], $\mathbf{W}_x^{contact}$ the sum of the contact wrenches, $\mathbf{L}$ is the angular momentum of the whole robot with respect to its CoM, and M the total mass of the robot. The gravity wrench is equal to the vector $\begin{bmatrix}\mathbf{0}^T \ M\mathbf{g}^T\end{bmatrix}^T$, where $\mathbf{g}$ is the gravity vector, and

$$\mathbf{W}_x^{contact} = \sum_j^N \begin{bmatrix} (\mathbf{c}_j - \mathbf{x}) \times \mathbf{f}_j \\ \mathbf{f}_j \end{bmatrix} , \tag{2}$$

where the $\mathbf{c}_j$ and $\mathbf{f}_j$ are the location and force of contact j, respectively, and N is the total number of contacts. Substituting (2) into (1) results in

$$\sum_j \begin{bmatrix} (\mathbf{c}_j - \mathbf{x}) \times \mathbf{f}_j \\ \mathbf{f}_j \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{L}} \\ M(\ddot{\mathbf{x}} - \mathbf{g}) \end{bmatrix} . \tag{3}$$

The system can be regarded as *dynamically balanced* if (3) is satisfied. The goal of the problem of CoM trajectory generation is to compute $\ddot{\mathbf{x}}(t)$ such that the *dynamic wrench*, the right hand side term of (3), can be compensated by the sum of contact force wrenches. It is usually assumed that the contact forces belong to a cone (*friction cone*) spanning from the contact point.

Generally, it is not trivial to compute the set, or even a reasonable subset, of the dynamic wrenches that can be compensated by a given set of contact cones [16]. However, by assuming that all contact cones contain the CoM $\mathbf{x}$, as shown in Fig. 1, it is possible to choose the force of contact j to be

$$\mathbf{f}_j = \alpha_j(\mathbf{x} - \mathbf{c}_j)\ , \alpha_j \geq 0\ . \tag{4}$$

This assumption is equivalent to enforcing the constraint $\mathbf{x} - \mathbf{c}_j \in F_j$, where F_j is the friction cone for the contact point j. Given the contact point $\mathbf{c}_j$, normal vector to the contact surface $\hat{\mathbf{n}}_j$ and the coefficient of friction μ_j, this constraint can be expressed as a second order cone constraint or approximated by a set of linear constraints.

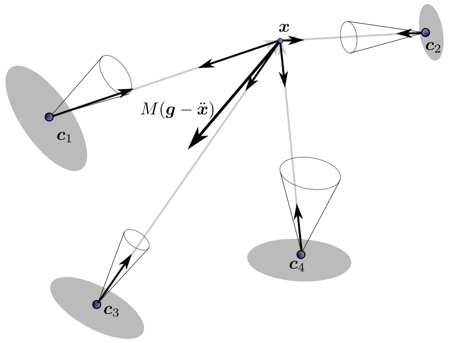


Fig. 1 All the contact cones contain the direction towards the position of the robot CoM $\mathbf{x}$. Choosing the force of contact j equal to $\alpha_j(\mathbf{c}_j - \mathbf{x})$, $\alpha_j \geq 0$, yields: $M(\mathbf{g} - \ddot{\mathbf{x}}) = \sum_j \alpha_j(\mathbf{c}_j - \mathbf{x}),\ \alpha_j \geq 0$.

By this selection, substituting (4) into the criterion for dynamic balance (3) results in

$$\begin{bmatrix} \mathbf{0} \\ \sum \alpha_j(\mathbf{x} - \mathbf{c}_j) \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{L}} \\ M(\ddot{\mathbf{x}} - \mathbf{g}) \end{bmatrix}\ . \tag{5}$$

From (5), the system is dynamically balanced if:

1. The angular momentum $\dot{\mathbf{L}}$ of the whole robot with respect to its CoM is negligible.
2. $M(\mathbf{g} - \ddot{\mathbf{x}})$ can be expressed by a positive linear combination of the vectors $\mathbf{c}_j - \mathbf{x}\ , j = 1, \ldots, N$.

It is worth noting that the proposed condition (5) holds in 3D without the restriction that contacts are coplanar. Furthermore, although the assumption $\mathbf{x} - \mathbf{c}_j \in F_j$

limits range of scenarios in which the simplified conditions for dynamic balance (5) can be used, it is important to note that the restrictions are less limited and can be applied in a larger range of scenarios compared with Zero Moment Point (ZMP) conditions.

3 Model Predictive Control with Dynamically Balancing Objective

3.1 Problem Formulation: Non-Convex Optimization Problem

The aim of this problem is to determine the CoM trajectory for the locomotion of a multi-limbed robot for a given sequence of contacts over a finite time horizon. Although the number of contacts between the robot and the environment can theoretically be infinite, without loss of generality it can be supposed that only a finite number of points N of the robot can be in contact with the environment. For example, the contact points for a rectangular foot can be represented by the 4 vertices of the foot sole. The positions of these time-varying contact points can be denoted by $\mathbf{c}_1(t), \mathbf{c}_2(t), \ldots, \mathbf{c}_N(t)$. A binary variable $\kappa_j(t) \in \{0,1\}$ can be defined to denote whether $\mathbf{c}_j(t)$ is in contact ($\kappa_j(t) = 1$) or not in contact ($\kappa_j(t) = 0$) with the environment at time t. In this work, the contact trajectory $\mathbf{c}_j(t)$ and $\kappa_j(t)$ are predetermined.

In this proposed model predictive control (*MPC*) scheme, the *decision variables* are the CoM positions $\mathbf{x}(t)$ and the contact force multipliers $\alpha_j(t)\ \forall j \in \{1, \ldots, N\}$, over the finite time horizon T, where $t \in [t_0, t_0 + T]$. Discretizing the horizon at a time step of δt results in K discrete time instances, where $T = K\delta t$ and $t_i = t_0 + i \cdot \delta t$. Given the contact trajectory information $\mathbf{c}_j(t)\ \forall j \in \{1, \ldots, N\}$ and $\kappa_j(t)\ \forall j \in \{1, \ldots, N\}$, the *objective* is to satisfy the translational components of (5) by minimising

$$J_1 = \sum_{i=1}^{K} \left\| M(\mathbf{g} - \ddot{\mathbf{x}}(t_i)) - \sum_{j}^{N} \kappa_j(t_i)\alpha_j(t_i)(\mathbf{c}_j(t_i) - \mathbf{x}(t_i)) \right\|^2 .$$

The *constraints* for this problem are:

- Initial/current CoM position $\mathbf{x}(t_0) = \mathbf{x}_0$
- Initial/current CoM velocity $\dot{\mathbf{x}}(t_0) = \dot{\mathbf{x}}_0$
- Final CoM position at the end of the horizon $\mathbf{x}(t_K) = \mathbf{x}_f$
- Max acceleration $\|\ddot{\mathbf{x}}(t_i)\|_\infty \leq a_{max},\ i = 0, \ldots, K-1$

Note that any constraints on the CoM velocity $\dot{\mathbf{x}}$ and acceleration $\ddot{\mathbf{x}}$ can be expressed with respect to $\mathbf{x}$ by the linear relationships

$$\dot{\mathbf{x}}(t_i) = \frac{\mathbf{x}(t_i) - \mathbf{x}(t_{i-1})}{\delta t},\ i = 1, \ldots, K \tag{6}$$

$$\ddot{\mathbf{x}}(t_i) = \frac{\dot{\mathbf{x}}(t_{i+1}) - \dot{\mathbf{x}}(t_i)}{\delta t},\ i = 0, \ldots, K-1\ . \tag{7}$$

As a result, all constraints are either linear equality or inequality constraints with respect to $\mathbf{x}$. Thus, the MPC optimization problem to determine the CoM trajectory over the finite time horizon can be expressed as

$$\begin{aligned} \min_{(\mathbf{x}(t_i))_i, (\alpha_j(t_i))_{i,j}} \quad & J_1\left((\mathbf{x}(t_i))_i, (\alpha_j(t_i))_{i,j} \right) \\ \text{s.t.} \quad & \|\ddot{\mathbf{x}}(t_i)\|_\infty \leq a_{max},\ i = 0, \ldots, K-1 \\ & \mathbf{x}(t_0) = \mathbf{x}_0 \\ & \dot{\mathbf{x}}(t_0) = \dot{\mathbf{x}}_0 \\ & \mathbf{x}(t_K) = \mathbf{x}_f\ . \end{aligned} \tag{8}$$

Remark: as it is presented here, to implement the approach as a model predictive controller would require measuring the CoM velocity ($\dot{\mathbf{x}}_0$), which is not always easy. If the contact forces are accessible, one could estimate their total wrench and use it to obtain an approximation of $M(\mathbf{g} - \ddot{\mathbf{x}})$, which could potentially be used to make the approach more robust.

3.2 An algorithm exploiting the objective function structure

On a computational level, the objective function J_1 is non-convex and it is therefore difficult to obtain the global optimum. Finding a local minimum could be relatively quick and provide good results, but in the next section we propose another approach that exploits the form of the objective function. The alternating convex optimization approach is inspired by [12], where successful results for a different application are obtained by using an alternating sequence of convex QPs (*AQPs*) instead of trying to solve head-on an optimization problem with a bilinear objective function. The structure of function J_1 can be expressed as

$$J_1 = \sum_i \left\| \varepsilon_i \left(\mathbf{x}(t_i), (\alpha_j(t_i))_j \right) \right\|^2 , \tag{9}$$

with

$$\varepsilon_i = M(\mathbf{g} - \ddot{\mathbf{x}}(t_i)) - \sum_j^N \kappa_j(t_i)\alpha_j(t_i)(\mathbf{c}_j(t_i) - \mathbf{x}(t_i)).$$

The functions ε_i are bilinear in the variables $\mathbf{x}(t_i)$ and $\alpha_j(t_i)$. Therefore, if $\mathbf{x}(t_i)$ are fixed, the optimization problem (8) becomes a convex QP that can be efficiently solved. Similarly, if the $\alpha_j(t_i)$ variables are fixed, we obtain a convex QP in the $\mathbf{x}(t_i)$ variables. This property can be used to alternatively optimize the $\mathbf{x}(t_i)$ and $\alpha_j(t_i)$

variables, using the solution of each step as the fixed variables of the next step. This ensures that the objective J_1 decreases at each iteration of the algorithm until it converges.

For the first step, an initial guess was made for the CoM trajectory that is consistent with the constraints from (8), and the $\alpha_j(t_i)$ variables are optimized. The most direct motion from $\mathbf{x}_0$ to $\mathbf{x}_f$ is often a natural candidate for this initial guess, so in practice this method does not require any tuning at all.

3.3 Results

The RobOptim [14] framework was used to illustrate the use of the proposed MPC formulation (8) and alternating quadratic programs that exploit the bilinear form (9) on CoM generation for locomotion. RobOptim provides a convenient interface to try various optimization tools. One interesting aspect of the formulation of the proposed MPC is that the matrices describing the constraints are very sparse, with $O(m)$ non-zero elements where m is the total number of variables. Thus, we chose the IPOPT optimizer [18] that can exploit matrix sparsity. Three different scenarios are simulated and presented:

1. 3 step walk with two foot supports (coplanar contacts);
2. 3 step walk with two foot supports and one hand support (non-coplanar contacts);
3. Jump-step with flight phase.

The 4.5 s trajectory for scenario 1 can be defined by the following sequence as shown in Fig. 2:

- $0 \leq t < 0.6$ s: Robot initially begins in double support
- $0.6 \leq t \leq 1.2$ s: Step 1 moving the left foot
- $1.2 < t < 1.9$ s: Double support phase
- $1.9 \leq t \leq 2.5$ s: Step 2 moving the right foot
- $2.5 < t < 3.2$ s: Double support phase
- $3.2 \leq t \leq 3.8$ s: Step 3 moving the left foot
- $3.8 < t \leq 4.5$ s: Double support phase

With a time horizon of 4.5 s and $\delta t = 0.1$ s, the MPC scheme consists of 45 discrete steps. Without loss of generality, it should be noted that the MPC in this setup solves for the entire trajectory, where in practice the time horizon would be shorter to achieve better computational performance. The dimensions of the feet 10.45 cm × 28.3 cm and the total mass 36.66 kg correspond to the Romeo robot [1]. At $t = 0$ s and $t = 4.5$ s, the CoM is set to be in the middle between the feet at a height of 0.6 m, $\mathbf{x}(0) = [0.14\ \ 0\ \ 0.6]^T$ and $\mathbf{x}(4.5) = [0.85\ \ 0\ \ 0.6]^T$, respectively. The initial guess for the CoM trajectory follows a straight line from its initial to its final position. Fig. 2 shows the CoM motion generated after two steps of the algorithm (i.e. one optimization of the $\alpha_j(t_i)$ variables and then one optimization of the CoM trajectory).

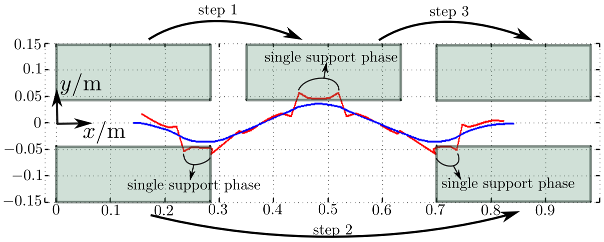


Fig. 2 The resulting trajectory from the MPC scheme (8) for scenario 1. In blue, the horizontal CoM trajectory produced by our algorithm in 95 ms. In red, the projected ZMP trajectory (10).

Fig. 2 shows the resulting CoM trajectory (in blue) and the projection (in red)

$$\mathbf{x}_{zmp} = \lambda_{zmp}(\mathbf{x} + M(\mathbf{g} - \ddot{\mathbf{x}})), \tag{10}$$

where $\lambda_{zmp} > 0$ is a scaling factor such that $z_{zmp} = 0$. The projected point (10) is equivalent to the ZMP point for the scenario 1 in which all contact points are always coplanar. It can be observed that the projected ZMP lies within the foot during single support phases. In this scenario, the results are similar to that of the ZMP+MPC approach, since all contacts are coplanar on the ground plane. For this scenario, the optimization problem (8) with horizon of K steps consists of $11K$ decision variables, comprised of 3 for the CoM position and 8 for the two contact supports (four contact points per support) at each time step, and 9 + K constraints from (8). The trajectory was computed in 95 ms (on a 2.40GHz Intel(R) Core(TM) i7-4700MQ CPU) after solving two alternations of the AQP, and hence 4 convex QPs. Executing more alternations of the algorithm showed that the cost J_1 quickly reached small values, for example, more than 1000 times smaller than the initial value after 4 optimizations. Furthermore, it was observed that the algorithm had almost converged after the 2 first optimizations.

In the second scenario, an additional contact corresponding to the right hand of the robot on a wall is considered. The contact point is at a height of 0.6 m, and activated only during the second step. This example demonstrates the ability of the proposed 3D condition and MPC algorithm to handle non-coplanar contacts. The resulting trajectory in Fig. 3 shows how the proposed algorithm can handle this situation and generated a different CoM trajectory. It could be observed from the results that the additional hand contact point enabled the robot to avoid the sway motion to the left.

Finally, the third scenario illustrates the ability of the algorithm to determine the CoM height without restrictions unlike the classical ZMP+MPC approach. The steps were replaced by one jump defined by the following sequence of actions: at $t = 0.6$ s the left foot leaves the ground, and at $t = 2.0$ s the right foot leaves the ground. After a flight phase of 0.4 s, the left foot lands at $t = 2.4$ s, and then the right foot lands at $t = 3.8$ s. Remark: in this scenario the a_{max} bound must be at least equal to the norm of the gravitational acceleration. Fig. 4 shows the CoM trajectory produced after 2 optimizations. This example clearly shows the benefits in relaxing the fixed height

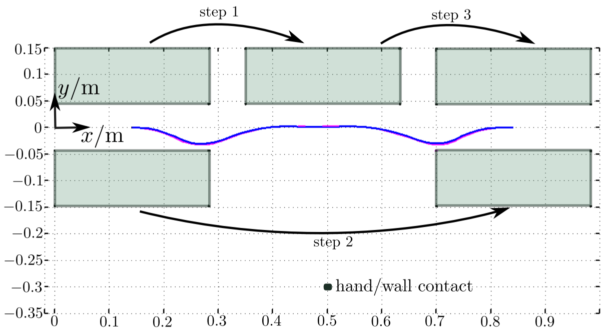


Fig. 3 The resulting trajectory from the MPC scheme (8) for scenario 2 with hand support. In blue, the horizontal CoM trajectory produced after 2 steps of the algorithm (in 96 ms). In purple, the trajectory produced after 6 steps of the algorithm (in 412 ms). These trajectories are almost exactly the same, which shows that the convergence is fast.

trajectory from the ZMP+MPC approach, such that the CoM trajectory generation is able to produce a jump motion in the z-direction.

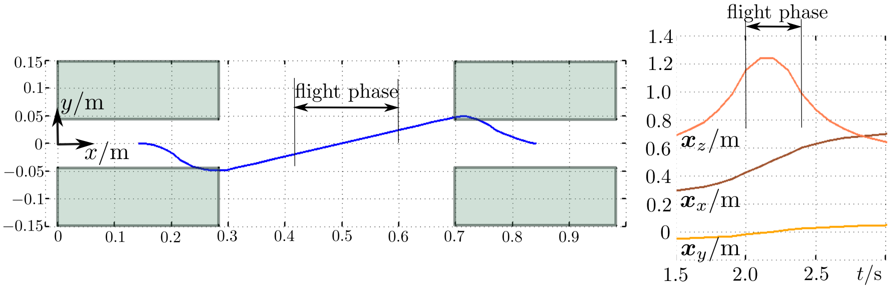


Fig. 4 *On the left:* the horizontal CoM trajectory generated (in 176 ms) for the jump scenario. *On the right:* the evolution of the CoM coordinates during the flight phase.

4 Model Predictive Control with Dynamically Balancing Constraints

4.1 Problem Formulation: Non-Convex Quadratically Constrained Quadratic Program

With the same aim of determining the CoM trajectory as Section 3, the problem in this section will be formulated by considering the dynamically balanced condition

(5) as constraints over the time horizon. Furthermore, the objective to minimise is the tracking error for a reference CoM trajectory $\mathbf{x}_r(t)$. The reference CoM trajectory is pre-planned based on the desired motion. In this MPC problem, the *decision variables* are the CoM jerk vectors $\dddot{\mathbf{x}}(t_i)$, $\forall i \in \{1,\ldots,K\}$. The convex *objective* of the problem is to minimise

$$J_2 = \sum_{i=1}^{K} Q_1 \left\|\mathbf{x}(t_i) - \mathbf{x}_r(t_i)\right\|^2 + Q_2 \left\|\dddot{\mathbf{x}}(t_i)\right\|^2 \ ,$$

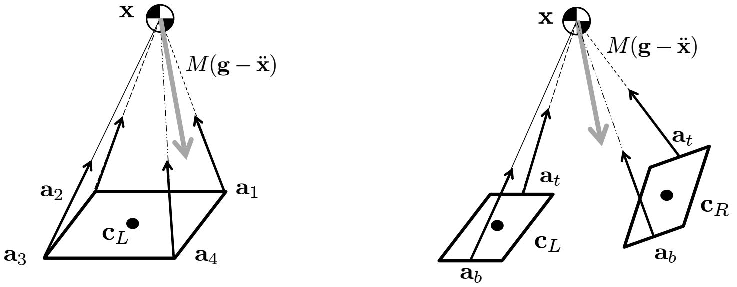
where Q_1 and Q_2 are the relative weights between the tracking and jerk terms, respectively. Note that the relationship between the CoM jerk $\dddot{\mathbf{x}}$ and the CoM position $\mathbf{x}$, velocity $\dot{\mathbf{x}}$ and acceleration $\ddot{\mathbf{x}}$ is linear [19] in the form

$$\begin{bmatrix} \mathbf{x}(t_{i+1}) \\ \dot{\mathbf{x}}(t_{i+1}) \\ \ddot{\mathbf{x}}(t_{i+1}) \end{bmatrix} = A \begin{bmatrix} \mathbf{x}(t_i) \\ \dot{\mathbf{x}}(t_i) \\ \ddot{\mathbf{x}}(t_i) \end{bmatrix} + B\dddot{\mathbf{x}}(t_i) \ .$$

As a result, the objective and all constraints can be expressed with respect to the CoM jerk. As presented in Section 2, the geometrical meaning of the constraint

$$\sum \alpha_j(\mathbf{c}_j - \mathbf{x}) = M(\mathbf{g} - \ddot{\mathbf{x}}) \ , \alpha_j \geq 0 \tag{11}$$

is that the vector $M(\mathbf{g}-\ddot{\mathbf{x}})$ must lie within the positive cone of the vectors $(\mathbf{c}_j - \mathbf{x})$ $\forall j \in \{1,\ldots,N\}$. Such constraints will be derived in the following for the case of one and two supports in contact.



(a) Single support contact (b) Double support contact

Fig. 5 The constraints required to achieve dynamic balance for: (a) when one support is in contact with the environment $\kappa_L = 1, \kappa_R = 0$ and (b) when two supports are in contact $\kappa_L = 1, \kappa_R = 1$.

As shown in Fig. 5(a), when the system is in single support it will be assumed that there are four contact points. The position of the vertices for the rectangular contact surface relative to the center of the contact $\mathbf{c}_j$ can be represented by $\mathbf{a_1}$, $\mathbf{a_2}$, $\mathbf{a_3}$ and $\mathbf{a_4}$. For the single support, the vector $M(\mathbf{g}-\ddot{\mathbf{x}})$ is inside the cone produced by the vectors $(\mathbf{c}_j+\mathbf{a}_1-\mathbf{x}, \mathbf{c}_j+\mathbf{a}_2-\mathbf{x}, \mathbf{c}_j+\mathbf{a}_3-\mathbf{x}, \mathbf{c}_j+\mathbf{a}_4-\mathbf{x}) = (\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, \mathbf{w}_4)$

if the following triple products are positive:

$$\begin{aligned} &\mathbf{w}_1 \times \mathbf{w}_2 \cdot (\mathbf{g} - \ddot{\mathbf{x}}) \geq 0, \quad \mathbf{w}_2 \times \mathbf{w}_3 \cdot (\mathbf{g} - \ddot{\mathbf{x}}) \geq 0 \\ &\mathbf{w}_3 \times \mathbf{w}_4 \cdot (\mathbf{g} - \ddot{\mathbf{x}}) \geq 0, \quad \mathbf{w}_4 \times \mathbf{w}_1 \cdot (\mathbf{g} - \ddot{\mathbf{x}}) \geq 0\ . \end{aligned} \tag{12}$$

For the case of double support contacts, the expression of the convex cone would be non-trivial if each support was assumed to consist of 4 contact points. This is because the convex cone would be dependent on the locations of each support. Furthermore, this would increase the number of surfaces of the convex cone for the vector $M(\mathbf{g} - \ddot{\mathbf{x}})$ to be checked, hence increasing the number of constraints. However, by assuming that each contact support only has two contact points, at the top $\mathbf{a}_t$ and bottom $\mathbf{a}_b$, then the resulting convex cone shown in Fig. 5(b) only consists of four surfaces and is a strict subset of the 4 contact point convex hull regardless of the support location. For the double support, the vector $M(\mathbf{g} - \ddot{\mathbf{x}})$ is inside the cone produced by the vectors $(\mathbf{c}_L + \mathbf{a}_t - \mathbf{x}, \mathbf{c}_L + \mathbf{a}_b - \mathbf{x}, \mathbf{c}_R + \mathbf{a}_b - \mathbf{x}, \mathbf{c}_R + \mathbf{a}_t - \mathbf{x}) = (\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, \mathbf{w}_4)$ as with the conditions in (12). In a similar manner, constraint equations for extra scenarios with additional contacts could be considered. For both mathematical and practical considerations, it will be assumed that contacts do not overlap, such that the contact points in Figure 5(b) can form a convex cone.

It can be observed that the constraints (12) for both single and double support are non-convex quadratic inequality constraints if the contact trajectories are known. As a result, the MPC optimization problem to determine the CoM trajectory over the finite time horizon can be expressed as

$$\begin{aligned} \min_{(\ddot{\mathbf{x}}(t_i))_i} \quad & \sum_{i=1}^{K} Q_1 \left\|\mathbf{x}(t_i) - \mathbf{x}_r(t_i)\right\|^2 + Q_2 \left\|\ddot{\mathbf{x}}(t_i)\right\|^2 \\ \text{s.t.} \quad & (12) \text{ for } \kappa_j = 1,\ \kappa_a = 0\ \forall a \neq j \\ & (12) \text{ for } \kappa_L = 1, \kappa_R = 1\ . \end{aligned} \tag{13}$$

The optimization problem (13) is a non-convex quadratically constrained quadratic program (*QCQP*) with a convex objective function.

4.2 Feasible Point Pursuit Successive Convex Approximation

On a computational level, the objective function J_2 has non-convex constraints and therefore it is non-trivial to obtain a feasible solution, let alone the global optimum. The feasible-point-pursuit successive convex approximation (*FPP-SCA*) [13] is an effective approach that solves the non-convex QCQP as a succession of QP convex approximations. Considering the non-convex QCQP of the standard form with the decision variable $\mathbf{u} \in \mathbb{R}^n$

$$\begin{aligned} \min_{\mathbf{u}} \quad & \mathbf{u}^T A_0 \mathbf{u} + \mathbf{q}_0^T \mathbf{u} \\ \text{s.t.} \quad & \mathbf{u}^T A_k \mathbf{u} + \mathbf{q}_k^T \mathbf{u} \le b_k,\ k = 1, \ldots, L\ , \end{aligned} \tag{14}$$

where $A_0 \in \mathbb{R}^{n \times n}$ is a positive semidefinite matrix, if any of $A_k \in \mathbb{R}^{n \times n}$ from the L constraints are not positive semidefinite, then (14) is non-convex and the problem is *NP*-hard in general. By performing a *linear restriction* about any point $\mathbf{z}_i$, the FPP-SCA approach aims to solve the following problem

$$\begin{aligned} \min_{\mathbf{u},\mathbf{s}} \quad & \mathbf{u}^T A_0 \mathbf{u} + \mathbf{q}_0^T \mathbf{u} + \lambda \sum_k s_k^2 \\ \text{s.t.} \quad & \mathbf{u}^T A_k^+ \mathbf{u} + (2\mathbf{z}_i^T A_k^- + \mathbf{q}_k^T)\mathbf{u} \le b_k + \mathbf{z}_i^T A_k^- \mathbf{z}_i + s_k\ , \\ & s_k \ge 0,\ k = 1, \ldots, L \end{aligned} \tag{15}$$

where A_k^+ and A_k^- are the positive semidefinite and negative semidefinite matrices from the decomposition $A_k = A_k^+ + A_k^-$. The terms s_k are slack variables that represent the violation of the k-th quadratic constraint and $\lambda \gg 1$ is a constant that gives priority to minimise the constraint violation. The vector $\mathbf{z}_i \in \mathbb{R}^n$ is the initial guess vector at iteration i. In the FPP-SCA approach, the convex QCQP (15) is repeated, where the variable $\mathbf{z}_{i+1}$ is set as the optimal solution $\mathbf{u}_i^*$ at iteration i.

By using the FPP-SCA approach to solve the MPC problem (13) with horizon of K steps, at each iteration of FPP-SCA the convex QCQP problem consists of 7K decision variables, comprised of 3 for the CoM jerk and 4 constraint slack variables at each time step, and 4K quadratic constraints (12) at each time step. The initial guess $\mathbf{z}_0$ can be any arbitrary vector, and in the simulations the zero vector was chosen. By solving a succession of (15), it was shown in [13] and in the results of Section 4.3 that the algorithm converges within a few successions.

4.3 Results

To demonstrate the MPC formulation (13), the CoM jerk trajectories were determined using the FPP-SCA for the three following scenarios: scenario 1 from Section 3.3, scenario 2 from Section 3.3, and walking up 2 steps of a staircase with the same hand support as in scenario 2 from Section 3.3. Scenarios 1 and 2 allow for a direct comparison between the trajectories resulting from the two different MPC schemes. The number of successive QCQPs solved was set as 5, however it was observed that convergence typically happened in less than 3 successions. The reference trajectory for both scenarios was set to be simply the forward linear motion of travel at a height of 0.6 m. The values for objective function weights in (13) and (15) were set as $Q_1 = 1$, $Q_2 = 10^{-4}$ and $\lambda = 10^4$.

The trajectory for scenario 1 is shown in Fig. 6. From the projected ZMP points using (10) it is clear that the produced CoM trajectory satisfies the 3D dynamically balanced criterion. To maintain good tracking performance of the reference

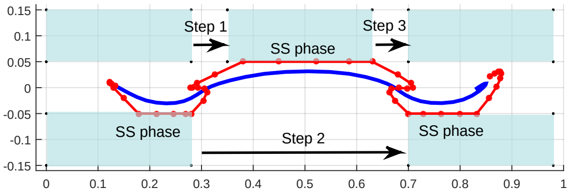


Fig. 6 The resulting trajectory from the MPC scheme (13) for scenario 1. In blue, the horizontal CoM trajectory and in red, the projected ZMP trajectory (10) since all contacts are coplanar.

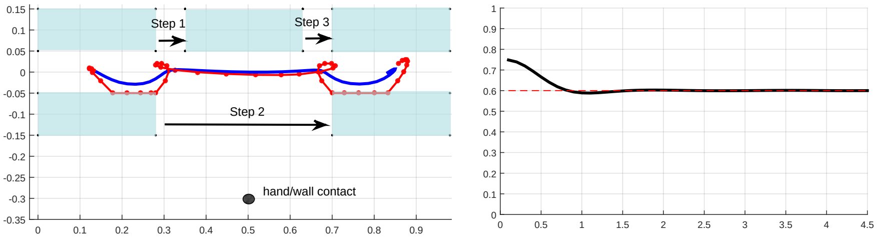


Fig. 7 *On the left:* the resulting trajectory from the MPC scheme (13) for scenario 2 with hand support. In blue, the horizontal CoM trajectory and in red, the projected ZMP trajectory (10). It can be observed that the results are similar to that from Fig. 3. *On the right:* the resulting z-direction trajectory from the MPC scheme (13) for scenarios 1 and 2. In black, the CoM trajectory and in red, the reference height trajectory. The results show that the trajectory successfully lowers from the initial height $z_0 = 0.75$ m to the reference height 0.6 m.

trajectory for $y = 0$, it is expected that the projected ZMP would be as close to the boundary of the single stance (*SS*) support region as possible. Furthermore, it can be observed that this MPC scheme generated very similar CoM trajectory results from Fig. 2.

As with Fig. 2, the natural artifact of left-right swaying motion can be observed in Fig. 6 due to the existence of single support instances. As such, a hand contact was included (scenario 2) during step 2 of the motion. From the resulting trajectories shown in Fig. 7 (on the left), the MPC scheme generated a very similar, in fact near identical, behaviour to that in Fig. 3. The extra hand contact allowed the CoM to better track the reference trajectory as there is no single support phase sway required during step 2. To show the tracking of the height trajectory, the initial height of the CoM was set to be 0.75 m. Fig. 7 (on the right) shows that the MPC CoM planner was able to quickly converge to the desired reference height of 0.6 m. As such, it can be claimed that for this scenario, the FFP-SCA formulation was able to achieve both feasibility of the dynamically balancing constraints and good tracking performance.

Finally, scenario 3 demonstrates the ability for the proposed MPC formulation to solve more complex behaviours, such as walking up stairs while using the hand to hold onto the staircase rails. The steps of the stairs were set as 15 cm high and

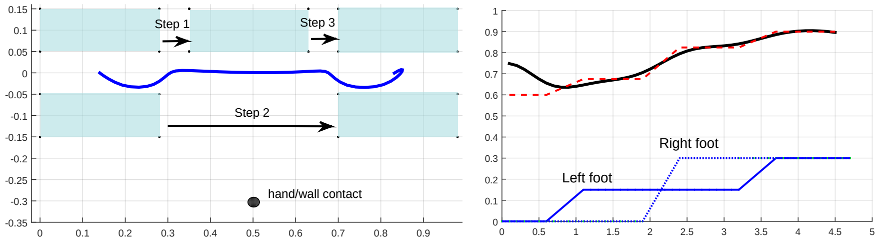


Fig. 8 *On the left:* the resulting trajectory from the MPC scheme (13) for scenario 3 with hand support walking up a staircase. The horizontal CoM trajectory is shown in blue. It can be observed that the results are similar to that from Fig. 7. *On the right:* the resulting z-direction trajectory from the MPC scheme (13) for scenario 3. In black, the CoM trajectory and in red, the reference height trajectory. The blue lines show the left and right foot trajectories indicating that the robot is walking up the staircase. The results show that the trajectory successfully tracks the desired height above the staircase platform.

the reference height trajectory for the CoM was set to be 0.6 m above the surface of stair steps. Fig. 8 (on the left) shows the x and y direction (top-down view) CoM trajectories determined by the MPC. It can be observed that the CoM results for the x and y directions are very similar to that of walking on a flat ground as shown in Fig. 7 (on the left).

However, the resulting z trajectory shown in Fig. 8 (on the right) shows significant differences with scenario 2. In addition to satisfying the dynamically balanced constraints for the locomotion on the non-coplanar contacts, the desired height trajectory represented by the red line in Fig. 8 (on the right) was able to be tracked. This example shows the robustness of the formulation and the QCQP solver to various different scenarios.

5 Discussion

To demonstrate the advantages and use of the 3D model for dynamically balanced locomotion in Section 2, two different example ways to use the criterion for MPC generation of CoM trajectory were presented. Both approaches have a fundamentally common point: the generalisations to allow for 3D non-coplanar multiple contacts naturally result in non-convex problems. However, regardless of the use of the criterion as an objective or constraint, the nature of the criterion is that it is in a bilinear form that can be converted into a convex quadratic function by restricting some variables. Both MPC approaches take advantage of this and then solve a succession of convex optimization problems. This is important in the proposed MPC schemes to ensure that it is still possible to implement them on a robot in real-time. Comparing between the two MPC schemes, it can be observed that using the dynamically balancing criterion in the objective function results in a problem with more decision variables than treating it as a constraint ($11K$ vs. $7K$, where K is the horizon length). However, the number of constraints are significantly less in return ($9 + K$ vs. $4K$), and the resulting problem is only a *QP* rather than *QCQP*. As a result, the approach from Section 3 is expected to be more computationally efficient than the approach from Section 4. In return, the constraint MPC approach provides a stricter notion of feasibility to dynamically balanced locomotion, and is less concerned with optimality.

Finally, it is also worth noting that compared to the traditional ZMP+MPC approach, several restrictions have been removed, such as coplanar contacts and predetermined height trajectory. The interesting point is that if any of these are relaxed, the problem complexity is identical to that if all are relaxed. As such, the 3D formulation proposed relaxes many conditions from the ZMP+MPC approach while still maintaining a balance with the computational cost of the resulting method. As with the ZMP+MPC approach, the focus is typically more on generating dynamically balanced motion rather than optimal gait behaviour. Hence, the development of methods such as AQP and FPP-SCA provides the opportunity to generate feasible motion for more general locomotion scenarios in real-time control.

6 Conclusion and Future Work

We proposed a novel model for dynamically balanced trajectory generation, more general than the classical IPM+ZMP approach, but simple enough to enable fast computations of CoM trajectories through an iterative resolution of convex QPs or convex QCQPs. This claim is supported by the low number of decision variables and constraint equations shown in the problem analysis. The generalizations gained from the proposed model and MPC approach include the ability to allow for multiple non-coplanar contacts and not having to predefine the CoM height trajectory. The results of the two proposed MPC approaches support the belief that the proposed 3D model of dynamically balanced locomotion is a good candidate for real-time model predictive control for multi-contact locomotion.

In future work, we will focus on performing experiments on a real humanoid robot, and address the following points:

1. Both the AQP and FPP-SCA approaches are observed to work well in practice and converge quickly. However there is no mathematical guarantee on the optimality of the solution, hence better understanding and analysis of such methods on the particular structure of the proposed MPC formulations should be more precisely studied.
2. In the optimisation of the CoM trajectories, the contact locations $\mathbf{c}_i$ could also be optimized without losing the convex QP structure. This allows the potential to not only compute the CoM trajectory, but also optimised contact locations.

References

1. Aldebaran: The Romeo project. URL http://projetromeo.com/
2. Featherstone, R.: Rigid Body Dynamics Algorithms. Springer (2008)
3. Goswami, A.: Postural stability of biped robots and the foot-rotation indicator (FRI) point. The International Journal of Robotics Research **18**(6), 523–533 (1999)
4. Harada, K., Kajita, S., Kaneko, K., Hirukawa, H.: An analytical method on real-time gait planning for a humanoid robot. In: IEEE/RAS International Conference on Humanoid Robots (Humanoids'04), pp. 640–655 (2004)
5. Harada, K., Kajita, S., Kaneko, K., Hirukawa, H.: Dynamics and balance of a humanoid robot during manipulation tasks. IEEE Transactions on Robotics **22**(3), 568–575 (2006)
6. Herdt, A., Perrin, N., Wieber, P.: Walking without thinking about it. In: IEEE International Conference on Intelligent Robots and Systems (IROS'10), pp. 190–195 (2010)
7. Herdt, A., Perrin, N., Wieber, P.: LMPC based online generation of more efficient walking motions. In: IEEE/RAS International Conference on Humanoid Robotics (Humanoids'12), pp. 390–395 (2012)
8. Hirukawa, H., Hattori, S., Harada, K., Kajita, S., Kaneko, K., Kanehiro, F., Fujiwara, K., Morisawa, M.: A universal stability criterion of the foot contact of legged robots - adios ZMP. In: IEEE International Conference on Robotics and Automation (ICRA'06), pp. 1976–1983 (2006)
9. Inomata, K., Uchimura, Y.: 3DZMP-based control of a humanoid robot with reaction forces at 3-dimensional contact points. In: IEEE International Workshop on Advanced Motion Control, pp. 402–407 (2010)
10. Kajita, S., Kanehiro, F., Kaneko, K., Yokoi, K., Hirukawa, H.: The 3D linear inverted pendulum mode: A simple modeling for a biped walking pattern generation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'01), pp. 239–246 (2001)
11. Lee, S., Goswami, A.: Reaction Mass Pendulum (RMP): An explicit model for centroidal angular momentum of humanoid robots. In: IEEE International Conference on Robotics and Automation (ICRA'07), pp. 4667–4672 (2007)
12. Majumdar, A., Ahmadi, A.A., Tedrake, R.: Control design along trajectories with sums of squares programming. In: IEEE International Conference on Robotics and Automation (ICRA'13), pp. 4054–4061 (2013)
13. Mehanna, O., Huang, K., Gopalakrishnan, B., Konar, A., Sidiropoulos, N.D.: Feasible point pursuit and successive approximation of non-convex QCQPs. IEEE Singal Processing Letters **22**(7), 804–808 (2015)
14. Moulard, T., Lamiraux, F., Bouyarmane, K., Yoshida, E., et al.: RobOptim: an optimization framework for robotics. In: The Robotics and Mechatronics Conference (ROBOMEC'13) (2013)
15. Posa, M., Cantu, C., Tedrake, R.: A direct method for trajectory optimization of rigid bodies through contact. The International Journal of Robotics Research **33**(1), 69–81 (2014)
16. Trinkle, J.C., Pang, J.S., Sudarsky, S., Lo, G.: On dynamic multi-rigid-body contact problems with coulomb friction. ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik **77**(4), 267–279 (1997)
17. Vukobratović, M., Borovac, B.: Zero-moment point—thirty five years of its life. International Journal of Humanoid Robotics **1**(1), 157–173 (2004)
18. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical programming **106**(1), 25–57 (2006)
19. Wieber, P.: Trajectory free linear model predictive control for stable walking in the presence of strong perturbations. In: IEEE-RAS International Conference on Humanoid Robots (Humanoids'06), pp. 137–142 (2006)
20. Wieber, P.: Viability and predictive control for safe locomotion. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'08), pp. 1103–1108 (2008)