HAL open science

Whole body motion controller with long-term balance constraints

Alexander Sherikov, Dimitar Dimitrov, Pierre-Brice Wieber

To cite this version:

Alexander Sherikov, Dimitar Dimitrov, Pierre-Brice Wieber. Whole body motion controller with long-term balance constraints. IEEE-RAS International Conference on Humanoid Robots (Humanoids), Nov 2014, Madrid, Spain. pp.444-450, 10.1109/HUMANOIDS.2014.7041399 . hal-01418363

HAL Id: hal-01418363

https://inria.hal.science/hal-01418363

Submitted on 16 Dec 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Whole body motion controller with long-term balance constraints

Alexander Sherikov, Dimitar Dimitrov, Pierre-Brice Wieber

***Abstract*— The standard approach to real-time control of humanoid robots relies on approximate models to produce a motion plan, which is then used to control the whole body. Separation of the planning stage from the controller makes it difficult to account for the whole body motion objectives and constraints in the plan. For this reason, we propose to omit the planning stage and introduce long-term balance constraints in the whole body controller to compensate for this omission. The new controller allows for generation of whole body walking motions, which are automatically decided based on both the whole body motion objectives and balance preservation constraints. The validity of the proposed approach is demonstrated in simulation in a case where the walking motion is driven by a desired wrist position. This approach is general enough to allow handling seamlessly various whole body motion objectives, such as desired head motions, obstacle avoidance for all parts of the robot, *etc*.**

I. Introduction

Humanoid robots have to perform tasks that are typically expressed as high-level objectives involving whole body motion control. For example, approaching and manipulating objects, maintaining visual contact, [1] *etc*. The motion of a robot must also respect feasibility constraints, such as collision avoidance, satisfaction of various mechanical constraints, and keeping balance. Fulfillment of the high-level objectives simultaneously with feasibility constraints often requires planning motion in advance, in particular when locomotion is performed. Due to the high complexity of humanoid robots, real-time control schemes typically resort to planning based on approximate models. Though these models often reduce a robot to a single point-mass, they were demonstrated to be very effective in preserving long-term balance [2], and even in choosing appropriate feasible footstep positions [3], [4].

The typical sequence of real-time motion generation [2], [3], [5], [6] includes: (i) first plan the motion of the center of mass with an approximate model to ensure long-term balance, (ii) then control the whole body of the robot to execute this plan taking into account whole body motion objectives and constraints. In this scheme the plan that results from stage (i) may not satisfy all the constraints considered at stage (ii), since they are usually neglected in the approximate model. Though an infeasible plan may still be executed approximately [7], it is always preferable to avoid such situations. Furthermore, inaccuracies in execution of the plan require its periodic corrections [8], [9]. Also, the high-level objectives cannot in general be expressed with the limited approximate model at stage (i), even though they might be required to drive the locomotion of the robot, for example, approaching an object before grasping it [10]. This lead to developing *ad hoc* approaches [10], [11], [12], which are often tailored to specific objectives and may be difficult to combine with footstep adjustments in response to external disturbances. Here, we propose to address the identified problems by omitting stage (i). Since stage (i) is crucial for long-term balance preservation, the whole body motion controller of stage (ii) must be modified to compensate for this omission.

The balance of humanoid robots can be analyzed in terms of viability theory and capturable states [13], [14], [15]. A robot is said to be in a capturable state, if it can eventually stop all motion and avoid falling or violating other feasibility constraints. In order to preserve balance the robot should never leave the set of capturable states, which is unfortunately impossible to determine in general [14]. However, it is possible to check with an approximate model if the robot can stop in K steps, *i.e.*, if a state of this model is K-step capturable [15]. Therefore, in order to preserve capturability, the motion can be previewed for K steps, with a terminal stopping constraint [5], [16], [17]. Clearly, at least one step must be considered to enable walking ($K \geq 1$).

Here we propose a modification of the standard whole body controller in such a way, that it does not require preliminary motion planning. In order to achieve this without deteriorating the long-term balance, the controller is augmented with appropriate constraints, which rely on motion preview with an approximate model to verify K-step capturability. Thus the controller generates such whole body commands that the balance of the previewed motion is preserved. At the same time the previewed motion is compliant with whole body constraints and objectives. The approximate model adopted in this work enables automatic adaptation of footstep positions as in [3], [16], but here, the adaptation is directly driven by the whole body motion objectives instead of a reference velocity for the body. This behavior is similar to the one demonstrated in [18], but does not require offline computations. Simultaneously with realization of whole body motion objectives, our controller automatically adapts footsteps positions when necessary to compensate for external disturbances, which is not addressed in [10], [11], [12]. Some works advocate for motion preview with nonlinear whole body models [19]. While such an approach is potentially more descriptive, due to computation time limits it is still necessary to employ approximations whose validity on real robots has yet to be demonstrated. And, more importantly, long-term balance constraints are not enforced as we explicitly do here.

The proposed whole body motion controller is posed as a hierarchical linear least squares problem with inequality constraints, which is solved using a dedicated active set

solver [20], [21].

The building blocks of the optimization problem are described in the Sections II–V: Section II briefly reviews standard whole body motion constraints, Section III outlines long-term balance constraints based on an approximate model, Section IV describes the coupling between the approximate and whole-body models, motion objectives are described in Section V. Finally, Section VI introduces the hierarchical least squares formulation of the whole body motion control, while Section VII presents simulation results.

II. Standard whole-body motion constraints

Our whole body controller is based on the standard inverse dynamics approach [22], which allows the definition of the instantaneous whole-body motion feasibility constraints as described in this section.

A. Lagrangian dynamics

The dynamics of a walking robot can be described as

$$\underbrace{\begin{bmatrix}\boldsymbol{H}_1\\ \boldsymbol{H}_2\end{bmatrix}}_{\boldsymbol{H}}\ddot{\boldsymbol{q}}+\underbrace{\begin{bmatrix}\boldsymbol{h}_1\\ \boldsymbol{h}_2\end{bmatrix}}_{\boldsymbol{h}}=\begin{bmatrix}\boldsymbol{\tau}\\ \boldsymbol{0}\end{bmatrix}+\underbrace{\begin{bmatrix}\boldsymbol{J}_{c,1}^T\\ \boldsymbol{J}_{c,2}^T\end{bmatrix}}_{\boldsymbol{J}_c^T}\boldsymbol{f}, \tag{1}$$

where $\boldsymbol{H}$ is the generalized inertia matrix; $\boldsymbol{q}=(\boldsymbol{q}',\boldsymbol{q}'')$ is the vector of generalized coordinates including joint angles $\boldsymbol{q}'$ and position and orientation of the base of the robot $\boldsymbol{q}''$; $\boldsymbol{h}$ is the vector of Coriolis, centrifugal, and gravity effects; $\boldsymbol{\tau}$ is the vector of joint torques; $\boldsymbol{f}$ is the vector of ground contact forces; and $\boldsymbol{J}_c$ is the corresponding Jacobian matrix. Note that given any $\ddot{\boldsymbol{q}}$ and $\boldsymbol{f}$, joint torques $\boldsymbol{\tau}$ satisfying the first part of dynamics can always be found. Consequently, we only need to find $\ddot{\boldsymbol{q}}$ and $\boldsymbol{f}$ such that

$$\boldsymbol{H}_2\ddot{\boldsymbol{q}}+\boldsymbol{h}_2=\boldsymbol{J}_{c,2}^T\boldsymbol{f}. \tag{2}$$

B. Constraints on the contact forces

The constraints on the ground contact forces reflect two aspects: limitations due to Coulomb's friction, and restriction of the Center of Pressure (CoP) to the contact area [22]. For simplicity, we describe the constraints only in single support, while extension to double support is trivial.

The contact forces $\boldsymbol{f}$ are considered as a single wrench, applied at a reference point on the contact surface. The constraints on this wrench are formulated in a local frame located at the reference point, with the z axis normal to the contact surface. Thus, the wrench is expressed as

$$\boldsymbol{f}=\boldsymbol{T}\begin{bmatrix}\boldsymbol{f}_c\\ \boldsymbol{\mu}_c\end{bmatrix}, \tag{3}$$

where $\boldsymbol{f}_c$ and $\boldsymbol{\mu}_c$ are the force and moment applied at the reference point in the local frame, and $\boldsymbol{T}$ is a frame transformation matrix.

We use a linear approximation of the Coulomb's friction cone, considering feasible contact force as a conic combination of unit vectors lying on the edges of the cone

$$\boldsymbol{f}_c=\boldsymbol{G}\boldsymbol{\lambda}, \tag{4}$$

where $\boldsymbol{\lambda}$ is a vector of positive coefficients

$$\boldsymbol{\lambda}\geq\boldsymbol{0}. \tag{5}$$

At the same time, the moment about the z axis is limited with

$$\|\mu_c^z\|\leq\gamma f_c^z, \tag{6}$$

where γ is a torsional friction coefficient ([23], Section 5.5). Position of the CoP is kept within the support area with a constraint of the form [22]

$$\boldsymbol{A}\begin{bmatrix}\boldsymbol{f}_c\\ \boldsymbol{\mu}_c\end{bmatrix}\leq\boldsymbol{b}. \tag{7}$$

C. Maintaining contacts

In order to ensure that the robot does not break contacts with the ground, we need the acceleration of each support foot to be zero in both translation and rotation:

$$\boldsymbol{J}_c\ddot{\boldsymbol{q}}+\dot{\boldsymbol{J}}_c\dot{\boldsymbol{q}}=\boldsymbol{0}. \tag{8}$$

D. Joint limits

The joints angles $\boldsymbol{q}'$ must stay within their mechanical limits. This is taken into account with the constraints

$$\underline{\boldsymbol{q}}'\leq\boldsymbol{q}'+T\dot{\boldsymbol{q}}'+\frac{T^2}{2}\ddot{\boldsymbol{q}}'\leq\overline{\boldsymbol{q}}', \tag{9}$$

where $\underline{\boldsymbol{q}}'$ and $\overline{\boldsymbol{q}}'$ are the vectors of lower and upper joint limits, and T is a positive time parameter.

III. Long-term balance constraints

The whole body constraints reviewed in Section II guarantee only instantaneous feasibility of the motion, and, in particular, they are not sufficient for long-term balance preservation. For this reason, it is common to impose additional constraints, for example, restricting the projection of the Center of Mass (CoM) to the support polygon when standing still [24]. The balance constraints described in this section are based on a preview of the motion of an approximate model and can be used both for walking and standing.

There is a variety of approximate models developed for planning the motion of humanoid robots. Many use a linear single point-mass model [2], some introduce additional point-masses [17], or constant external forces [25], or angular momentum [26]. Even the simple point-mass model can be used differently: as a second or third order model [2], [3], [27]. The latter one appears to be more suitable, since it allows for smooth transition of the contact forces [28]. Our approach is not limited to a certain model, and here, for the sake of illustration, we adopt a simple third order point-mass model. This model enables automatic footstep placement, provided that the duration and sequence of steps are predetermined to keep the model linear [3].

A. Point-Mass Model

The motion of the point-mass model along the x axis is characterized by the continuous time system

$$\begin{bmatrix} \dot{c}^x \\ \ddot{c}^x \\ \dddot{c}^x \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & \frac{g}{c^z} & 0 \end{bmatrix} \begin{bmatrix} c^x \\ \dot{c}^x \\ \ddot{c}^x \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -\frac{g}{c^z} \end{bmatrix} \dot{z}^x, \qquad (10)$$
$$z^x = c^x - \frac{c^z}{g}\ddot{c}^x,$$

where $\boldsymbol{c} = (c^x, c^y, c^z)$ is the position of the CoM of the robot, $\boldsymbol{z} = (z^x, z^y)$ is the position of the Center of Pressure (CoP) on the ground and g is the gravitational acceleration, aligned with the z axis. This system is equivalent for the motion along y axis and is derived under assumptions, that the robot walks on a flat ground and the CoM height c^z is constant.

System (10) is discretized and its evolution is previewed over N time intervals. Since evolution of the system is determined by its initial state $(\boldsymbol{c}_0^{x,y}, \dot{\boldsymbol{c}}_0^{x,y}, \ddot{\boldsymbol{c}}_0^{x,y})$ and the control inputs $(\dot{\boldsymbol{z}}_0, \ldots, \dot{\boldsymbol{z}}_{N-1})$, all constraints and objectives can be expressed using them. The control inputs in turn can be expressed through the CoP and footstep positions in order to simplify constraints as described in the following.

B. Constraints on positions of the CoP

A walking motion on a flat ground is feasible only if the CoP remains inside the support polygon [29]. This requirement can be taken into account explicitly in the form of a constraint on the position $\boldsymbol{Z} \in \mathbb{R}^{2N}$ of the CoP over the next N preview sampling intervals. Provided that the feet are rectangular and the CoP positions $\boldsymbol{Z}$ are expressed in appropriately chosen frames, the constraints on $\boldsymbol{Z}$ can be expressed

$$\underline{\boldsymbol{Z}} \leq \boldsymbol{Z} \leq \overline{\boldsymbol{Z}}, \qquad (11)$$

where $\underline{\boldsymbol{Z}}$ and $\overline{\boldsymbol{Z}}$ are the vectors of lower and upper bounds [30].

C. Constraints on positions of the feet

The key feature of the Model Predictive Control (MPC) scheme proposed in [3] is that it automatically determines appropriate footsteps. In order to realize this behavior, the previewed footstep positions must be constrained to feasible areas with respect to the current foothold. These feasible areas are usually determined experimentally as polygons on the ground [3]. Approximating them by rectangles and choosing reference frames appropriately allows formulating these constraints as simple bounds [4]

$$\underline{\boldsymbol{P}} \leq \boldsymbol{P} \leq \overline{\boldsymbol{P}}, \qquad (12)$$

where $\boldsymbol{P} \in \mathbb{R}^{2K}$ is the vector of positions of the next K footsteps, $\underline{\boldsymbol{P}}$ and $\overline{\boldsymbol{P}}$ the corresponding vectors of lower and upper bounds.

D. Terminal capturability constraint

In order to enforce K-step capturability [15], we impose a terminal stopping constraint at the end of the previewed K steps. It is well known, that for the second order point-mass model, the terminal capturability constraint involves position of the capture point

$$\boldsymbol{\xi}^{x,y} = \boldsymbol{c}_N^{x,y} + \frac{1}{\omega}\dot{\boldsymbol{c}}_N^{x,y}, \qquad (13)$$

which is constrained to stay in the final support area [15], [17], [27]. An equivalent terminal constraint for the third order system (10) must be imposed on the velocity of the capture point

$$\dot{\boldsymbol{\xi}}^{x,y} = \dot{\boldsymbol{c}}_N^{x,y} + \frac{1}{\omega}\ddot{\boldsymbol{c}}_N^{x,y} = \boldsymbol{0}, \qquad (14)$$

from which follows $\boldsymbol{\xi}^{x,y} = \boldsymbol{z}_N^{x,y}$. This implies that the capture point is again limited to the final support area, due to the constraints on the CoP positions (11). The equality (14) can be easily obtained by deriving the condition under which the system (10) is stable with a zero control input $\dot{\boldsymbol{z}} = \boldsymbol{0}$.

Note that in some situations the capturability constraint can be infeasible, since it is impossible to stop in K previewed steps. This, however, does not necessarily imply that the robot will eventually fall. This issue is addressed in the controller proposed in Section VI.

IV. Coupling the approximate and whole body models

The point-mass and the whole body models presented in Sections II and III are coupled through the current accelerations of the CoM and swing foot. The coupling ensures that the commands for the robot comply with the capturability constraint (14), while the previewed motion fulfills the whole body constraints (Section II) and objectives (Section V).

A. Acceleration of the CoM

The CoM acceleration $\ddot{\boldsymbol{c}}_0$ can be directly obtained from the initial state of the system in (10). The CoM of the robot and the point-mass model are therefore linked by

$$\boldsymbol{J}_{com}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{com}\dot{\boldsymbol{q}} = \ddot{\boldsymbol{c}}_0. \qquad (15)$$

B. Acceleration of the swing foot

Given the next footsteps $\boldsymbol{P}$ and the current state of the swing foot $(\boldsymbol{s}_0, \dot{\boldsymbol{s}}_0)$ (based on the current state of the robot), the swing foot trajectory is obtained using cubic polynomials as in [6]. This determines the current acceleration of the swing foot $\ddot{\boldsymbol{s}}_0$ as a linear function of $(\boldsymbol{P}, \boldsymbol{s}_0, \dot{\boldsymbol{s}}_0)$. Thus, during a single support, the swing foot follows this translational acceleration while maintaining a constant orientation:

$$\boldsymbol{J}_{foot}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{foot}\dot{\boldsymbol{q}} = \begin{bmatrix} \ddot{\boldsymbol{s}}_0 \\ \boldsymbol{0} \end{bmatrix}. \qquad (16)$$

V. Control objectives

Various control objectives can be introduced in the controller, depending on the desired behavior. This section describes a few possible objectives, which were used in the simulations presented in Section VII.

A. Hand motion

A saturated PD controller $\boldsymbol{\pi}_{hand}$ is used to drive the right hand to a desired position. This is realized using a least squares objective

$$\left\| \boldsymbol{J}_{hand}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{hand}\dot{\boldsymbol{q}} - \boldsymbol{\pi}_{hand} \right\|^2 . \tag{17}$$

B. Orientation of the torso

In order to prevent inclination of the body while walking, the orientation of the torso is kept constant with the objective

$$\left\| \boldsymbol{J}_{torso}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{torso}\dot{\boldsymbol{q}} - \boldsymbol{\pi}_{torso} \right\|^2 . \tag{18}$$

C. Controlling joints

A least squares objective based on a simple joint PD-controller

$$\left\| \ddot{\boldsymbol{q}}' + K_d\dot{\boldsymbol{q}}' - K_p(\boldsymbol{q}'_{ref} - \boldsymbol{q}') \right\|^2 \tag{19}$$

is used for two purposes: damping ($K_d > 0$, $K_p = 0$) and maintaining reference configuration $\boldsymbol{q}'_{ref}$ ($K_d > 0$, $K_p > 0$).

D. Penalizing violent swing foot motion

The footstep adjustments are allowed during the whole duration of a step, which may result in violent position changes near the end of a step. A straightforward workaround is to block footstep adaptation when the swing foot approaches the ground. Alternatively, it is possible to minimize the swing foot jerk

$$\| \dddot{\boldsymbol{s}} \|^2 , \tag{20}$$

which is a linear function of $(\boldsymbol{P}, \boldsymbol{s}_0, \dot{\boldsymbol{s}}_0)$.

E. CoP positioning and damping

Also, we employ objectives in order to shape the previewed motion of the approximate model. It is done by minimizing the distance between the CoP positions and the center of the footsteps

$$\| \boldsymbol{Z} - \boldsymbol{V}\boldsymbol{P} \|^2 , \tag{21}$$

where $\boldsymbol{V}$ is a matrix which selects footstep positions appropriately. Additionally, the CoP velocity

$$\left\| \dot{\boldsymbol{Z}} \right\|^2 \tag{22}$$

is minimized.

VI. A HIERARCHICAL LEAST-SQUARES PROBLEM

The constraints and objectives described in Sections II, III, IV, and V are collected in a single hierarchical optimization problem (or, more concisely, a hierarchy), which provides a convenient framework for whole body motion control [20]. All equality and inequality constraints imposed in a hierarchy are satisfied in a least-squares sense, *i.e.*, the norm of their violation is minimized. For this reason, we can express the objectives described in Section V as equality constraints and introduce them at lower levels of a hierarchy.

The proposed whole body motion controller uses a hierarchy with the following three levels:

1) The highest priority level of the hierarchy includes the whole body motion constraints from Section II, long-term balance constraints described in Section III without the capturability constraint, and coupling constraints introduced in Section IV:

$$(2, 5, 6, 7, 8, 9, 15, 16, 11, 12).$$

2) The capturability constraint

$$(14)$$

is considered separately to prevent violation of the primary constraints when it is infeasible. Indeed, infeasibility of the capturability constraint does not always imply that the robot will fall, since it enforces capturability with a fixed preview horizon, which may be overly conservative.

3) The third level collects objectives, which are weighted to obtain the desired behavior:

$$(17, 18, 19, 21, 22, 20).$$

The decision variables of this hierarchy are the vector of accelerations of the generalized coordinates $\ddot{\boldsymbol{q}} = (\ddot{\boldsymbol{q}}', \ddot{\boldsymbol{q}}'')$, the contact wrenches expressed with $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}_c$, the positions of the CoP $\boldsymbol{Z}$ and footsteps $\boldsymbol{P}$ in the preview horizon, and the CoM acceleration $\ddot{\boldsymbol{c}}_0^{x,y}$.

VII. SIMULATIONS

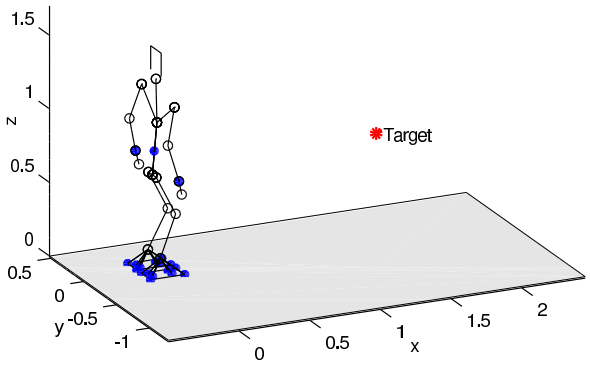
We evaluate the proposed approach in several simulations with an HRP-2 robot assuming perfect model and perfect inertial measurement unit. The commands for the robot are produced by solving the hierarchy introduced in Section VI with a dedicated active set solver on each control iteration. Size of the hierarchy depends on the length of preview horizon and discretization interval of the approximate model, which are set to the duration of two steps $0.8 \cdot 2$ (s) and 100 (ms) respectively as in [3]. Such choice results in a hierarchy with about 120 equality and 80 inequality constraints and the number of decision variables changing between 80 and 90 depending on the state of the robot. In order to achieve high computational performance, the solver is hot-started using a guess of the active set and solution. Furthermore, the solver handles simple bounds on variables more efficiently than general inequality constraints, which is taken into account in the formulation of the hierarchy.

During the tests the robot is exposed to external disturbances whose implementation is described in the next subsection, while the following subsections focus on describing the test setups and results.

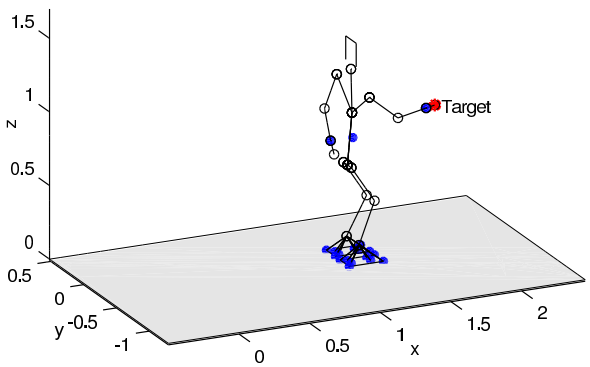
A. Simulating disturbances

External disturbances are modeled as instantaneous impacts leading to change in velocities. The relation between the initial $\dot{\boldsymbol{q}}^-$ and resultant velocities $\dot{\boldsymbol{q}}^+$ and impulses due to interaction with environment can be obtained by integrating the Lagrangian dynamics (1) [31]

$$\boldsymbol{H}(\dot{\boldsymbol{q}}^+ - \dot{\boldsymbol{q}}^-) = \boldsymbol{J}_c^T \boldsymbol{r}_c + \boldsymbol{J}_d^T \boldsymbol{r}_d, \tag{23}$$



(a) Start of the simulation.



(b) End of the simulation.

Fig. 1. Tests 1,2: Initial and final positions of the robot and the target.

where $\boldsymbol{r}_d$ is an impulse representing the disturbance, $\boldsymbol{r}_c$ is an impulse appearing due to ground contacts, $\boldsymbol{J}_c$ and $\boldsymbol{J}_d$ are respective Jacobians. The computed resultant velocities are constrained to maintain contacts

$$\boldsymbol{J}_c \dot{\boldsymbol{q}}^+ = \boldsymbol{0}. \tag{24}$$

All disturbances are applied to the torso of the robot and their directions are parallel to the ground.

B. Test 1: reaching an object

In this test we demonstrate walking driven by a whole-body objective, and more precisely the hand motion objective (17), which controls the right hand of the robot to a target position. This objective influences the walking motion generation, resulting in a sequence of steps taken towards the target. Note that exclusion of (17) from the hierarchy results in walking in place. The desired position of the hand is set to a point which is initially unreachable, while later during the simulation the desired position changes in an unpredictable way. The initial and final positions of the robot and target are shown in Fig. 1. The resulting motions of the CoM, feet and hand are shown in Fig. 3(a) and in the accompanying video. The reaction of the system to changes in the target hand position can be seen in Fig. 4. The robot starts walking since the target is initially unreachable, and continues to walk until the target is reached, around 4 (s). However, due to a change in the x position of the target, the walk is resumed

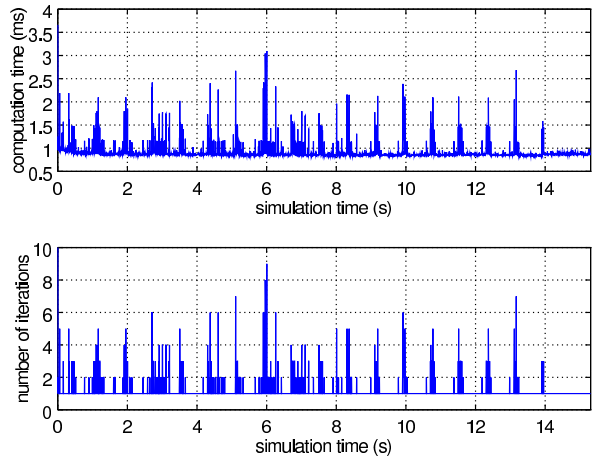


Fig. 2. Test 1: Time required for solution of the optimization problem (above) and the number of iterations of the solver, *i.e.*, the number of active set changes (below).

(Fig. 4(a)). The lateral motion of the target also influences the walk, as can be seen in Fig. 4(b). This experiment clearly demonstrates that the walking motion (both CoM and feet motion) is determined taking into account whole-body objectives directly, which cannot be achieved with other schemes like the one in [3]. It is demonstrated here with the desired hand position, but with respect to earlier similar results such as in [12], the approach proposed here is not limited to the hand: it is completely generic and can handle seamlessly all sorts of whole-body objectives, such as a desired head motion, obstacle avoidance for all parts of the robot, *etc*. Moreover, we can see in Fig. 4(a) that with our approach, the walking motion reacts immediately, including footstep modifications even if the swing foot is in mid-air, as happens around 4 (s) and 8 (s).

In order to assess applicability of the proposed control approach to real-time applications, we measured computation time of the solver at each control iteration (see Fig. 2). Most of the time solver does only one iteration due to successful guess of the active set. The number of iterations increases during transition between the left and right supports. The average time required for solution of the optimization problem is equal to 0.92 (ms), when the solver is executed on Intel Core i5-3360M (2.80GHz) CPU.

C. Test 2: reaching an object and reacting to disturbances

The setup of this test is the same as for the previous one, but additionally the robot is pushed, while it is trying to approach the hand target. Disturbances of 15 Ns are applied to the robot at 2.5 (s) from the right, forcing the robot to make a step to the left, and at 7 (s) from the front, forcing the robot to make a step backward. The results can be seen in Figs. 3(b) and 5, and in the accompanying video. We can see, especially in comparison with the previous simulation without perturbations, that the controller immediately reacts to disturbances and successfully compensates for them by adjusting footsteps in mid-air.

The average time required for solution of the optimization

problem is the same as before, but there are peaks up to 20 (ms) when disturbances are applied due to increased number of iterations of the solver. This issue can be addressed by limiting the number of iterations.

VIII. Conclusion

We presented a new approach to real-time whole body motion control, which, contrary to the standard approach [2], [3], [5], [6], does not perform preliminary motion planning with approximate models. The proposed controller design allows for generation of walking motions that satisfy long-term balance constraints and whole body motion objectives. This property is verified in simulations performed with HRP-2 robot. The computational performance demonstrated during the simulations is comparable to the performance of the state of the art real-time whole body motion controllers, which require preplanned footstep positions [32].

References

[1] N. Mansard, O. Stasse, F. Chaumette, and K. Yokoi, "Visually-guided grasping while walking on a humanoid robot," in *Robotics and Automation, 2007 IEEE International Conference on*, 2007, pp. 3041–3047.

[2] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, vol. 2, Sept 2003, pp. 1620–1626 vol.2.

[3] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online Walking Motion Generation with Automatic Foot Step Placement," *Advanced Robotics -Utrecht-*, vol. 24, no. 5-6, pp. 719–737, 2010.

[4] H. Diedam, D. Dimitrov, P. B. Wieber, K. Mombaur, and M. Diehl, "Online walking gait generation with adaptive foot positioning through linear model predictive control," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, 2008, pp. 1121–1126.

[5] M. Morisawa, K. Harada, S. Kajita, S. Nakaoka, K. Fujiwara, F. Kanehiro, K. Kaneko, and H. Hirukawa, "Experimentation of humanoid walking allowing immediate modification of foot place based on analytical solution," in *Robotics and Automation, 2007 IEEE International Conference on*, April 2007, pp. 3989–3994.

[6] K. Nishiwaki and S. Kagami, "Online walking control system for humanoids with short cycle pattern generation," *The International Journal of Robotics Research*, vol. 28, no. 6, pp. 729–742, 2009.

[7] F. Kanehiro, W. Suleiman, K. Miura, M. Morisawa, and E. Yoshida, "Feasible pattern generation method for humanoid robots," in *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, Dec 2009, pp. 542–548.

[8] T. Buschmann, R. Wittmann, M. Schwienbacher, and H. Ulbrich, "A method for real-time kineto-dynamic trajectory generation," in *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*, Nov 2012, pp. 190–197.

[9] T. Takenaka, T. Matsumoto, and T. Yoshiike, "Real time motion generation and control for biped robot -3rd report: Dynamics error compensation-," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, Oct 2009, pp. 1594–1600.

[10] E. Yoshida, O. Kanoun, C. Esteves, and J.-P. Laumond, "Task-driven support polygon reshaping for humanoids," in *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, Dec 2006, pp. 208–213.

[11] K. Nishiwaki, S. Kagami, J. Kuffner, M. Inaba, and H. Inoue, "Online humanoid walking control system and a moving goal tracking experiment," in *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, vol. 1, Sept 2003, pp. 911–916 vol.1.

[12] Y. Fukumoto, K. Nishiwaki, M. Inaba, and H. Inoue, "Hand-centered whole-body motion control for a humanoid robot," in *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 2, Sept 2004, pp. 1186–1191 vol.2.

[13] J.-P. Aubin, *Viability theory*. Birkhuser, 1991.

[14] P.-B. Wieber, "Viability and Predictive Control for Safe Locomotion," in *IEEE-RSJ International Conference on Intelligent Robots & Systems*, Nice, France, 2008.

[15] T. Koolen, T. de Boer, J. Rebula, A. Goswami, and J. Pratt, "Capturability-based analysis and control of legged locomotion, part 1: Theory and application to three simple gait models," *The International Journal of Robotics Research*, vol. 31, no. 9, pp. 1094–1113, 2012.

[16] J. Urata, K. Nishiwaki, Y. Nakanishi, K. Okada, S. Kagami, and M. Inaba, "Online walking pattern generation for push recovery and minimum delay to commanded change of direction and speed," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, Oct., pp. 3411–3416.

[17] T. Takenaka, T. Matsumoto, and T. Yoshiike, "Real time motion generation and control for biped robot -1st report: Walking gait pattern generation-," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, oct. 2009, pp. 1084 –1091.

[18] O. Kanoun, J.-P. Laumond, and E. Yoshida, "Planning foot placements for a humanoid robot: A problem of inverse kinematics," *The International Journal of Robotics Research*, 2010.

[19] T. Erez, K. Lowrey, Y. Tassa, V. Kumar, S. Kolev, and E. Todorov, "An integrated system for real-time model predictive control of humanoid robots," in *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on*, 2013.

[20] L. Saab, O. E. Ramos, F. Keith, N. Mansard, P. Souères, and J.-Y. Fourquet, "Dynamic whole-body motion generation under rigid contacts and other unilateral constraints," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 346–362, 2013.

[21] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *The International Journal of Robotics Research*, 2014.

[22] Y. Fujimoto and A. Kawamura, "Proposal of biped walking control based on robust hybrid position/force control," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 3, Apr 1996, pp. 2724–2730 vol.3.

[23] R. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Taylor & Francis, 1994.

[24] M. de Lasa, I. Mordatch, and A. Hertzmann, "Feature-based locomotion controllers," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 131:1–131:10, July 2010.

[25] A. Ibanez, P. Bidaud, and V. Padois, "Unified preview control for humanoid postural stability and upper-limb interaction adaptation," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 1801–1808.

[26] S.-H. Lee and A. Goswami, "Reaction mass pendulum (rmp): An explicit model for centroidal angular momentum of humanoid robots," in *Robotics and Automation, 2007 IEEE International Conference on*, April 2007, pp. 4667–4672.

[27] J. Englsberger, C. Ott, M. Roa, A. Albu-Schaffer, and G. Hirzinger, "Bipedal walking control based on capture point dynamics," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, sept. 2011, pp. 4420 –4427.

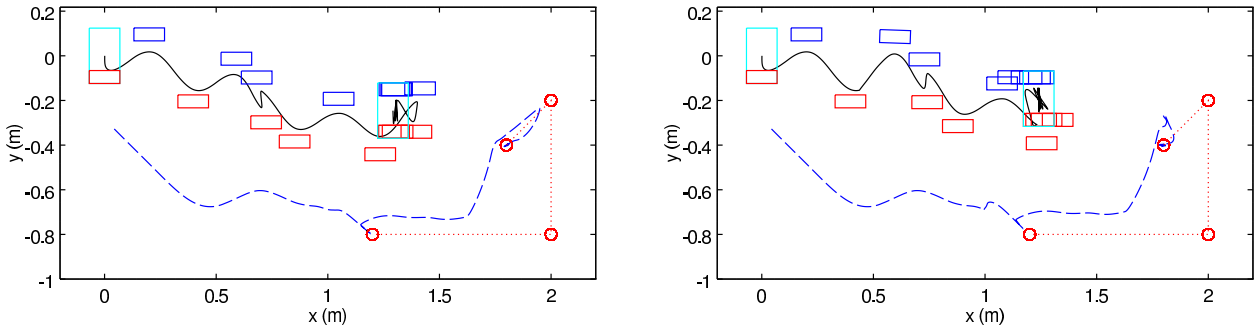
[28] S. Kajita, M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro, and K. Yokoi, "Biped walking stabilization based on linear inverted pendulum tracking," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, oct. 2010, pp. 4489 –4496.

[29] P.-B. Wieber, "On the stability of walking systems," in *Proceedings of the International Workshop on Humanoid and Human Friendly Robotics*, Tsukuba, Japon, 2002.

[30] D. Dimitrov, A. Paolillo, and P. B. Wieber, "Walking motion generation with online foot position adaptation based on ℓ_1- and ℓ_∞-norm penalty formulations," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May, pp. 3523–3529.

[31] S. Chareyron and P.-B. Wieber, "Stability and regulation of nonsmooth dynamical systems," INRIA, Rapport de recherche RR-5408, 2004.

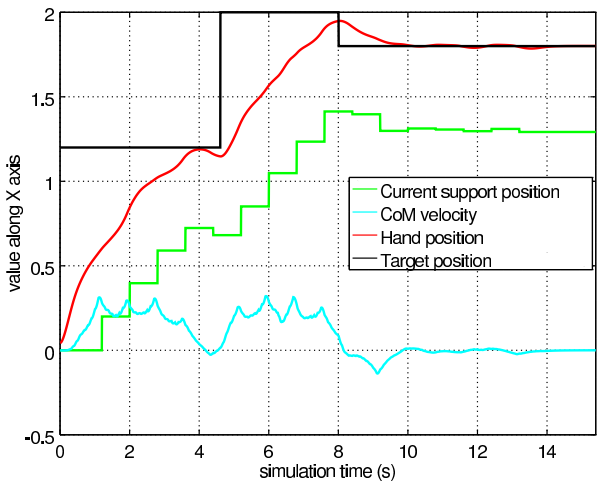
[32] S. Kuindersma, F. Permenter, and R. Tedrake, "An efficiently solvable quadratic program for stabilizing dynamic locomotion," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, May 2014.



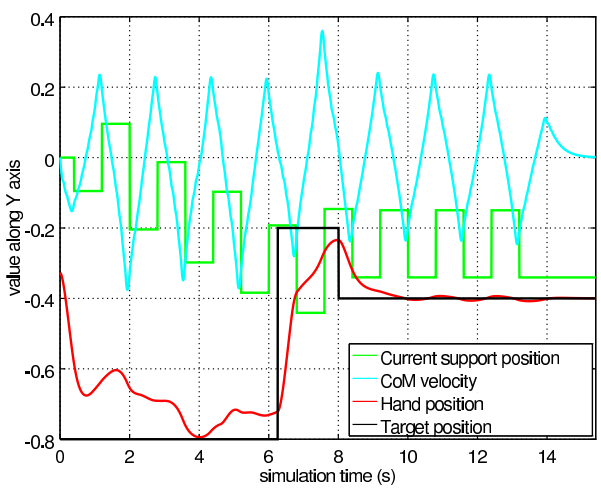
(a) Test 1: No disturbances.

(b) Test 2: With disturbances.

Fig. 3. Tests 1,2: Top view of the walking motion. Footsteps are represented by rectangles, trajectory of the CoM is in black. Trajectory of the hand is in dashed blue, while the trajectory of the target is in dotted red.

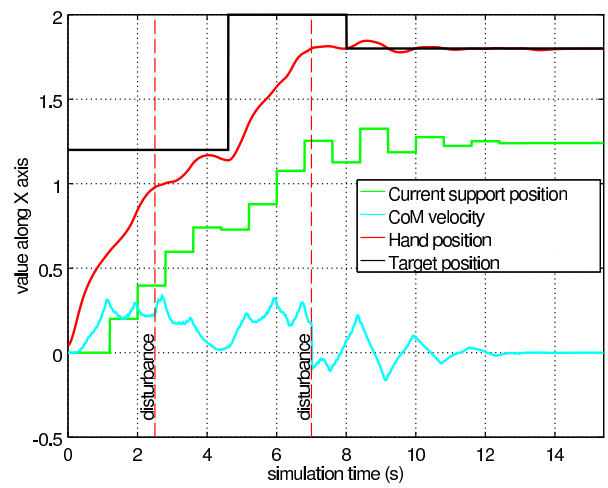


(a) Evolution with time of the x components of the target, hand, and current support positions and CoM velocity.

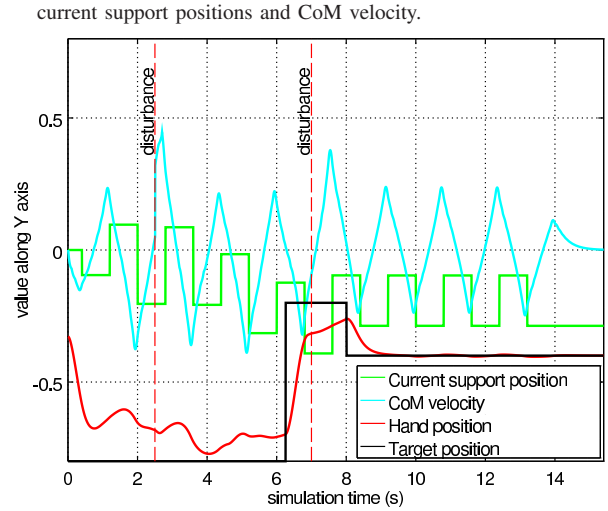


(b) Evolution with time of the y components of the target, hand, and current support positions and CoM velocity.

Fig. 4. Test 1: Reaction to changes in target position.



(a) Evolution with time of the x components of the target, hand, and current support positions and CoM velocity.



(b) Evolution with time of the y components of the target, hand, and current support positions and CoM velocity.

Fig. 5. Test 2: Reaction to changes in target position and external disturbances. The instants of disturbances are indicated with vertical red lines.