Capture-Point Based Balance and Reactive Omnidirectional Walking Controller

Michael Bombile and Aude Billard

***Abstract*—This paper proposes a capture-point based reactive omnidirectional controller for bipedal locomotion. The proposed scheme, formulated within Model Predictive Control (MPC) framework, exploits concurrently the Center of Mass (CoM) and Capture Point (CP) dynamics. It allows the on-line generation of the CoM reference trajectory and the automatic generation of footstep positions and orientations in response to a given velocity to be tracked, or a disturbance to be rejected by the robot while accounting explicitly for different walking constraints. For instance, in order to cope with disturbance such as a push, the proposed controller not only adjusts the position of the Center of Pressure (CoP) within the support foot, but can also induce at least one step with appropriate length allowing thus to maintain the stability of the robot. Finally, the proposed algorithm is validated through simulations and actual experiments on the humanoid robot iCub.**

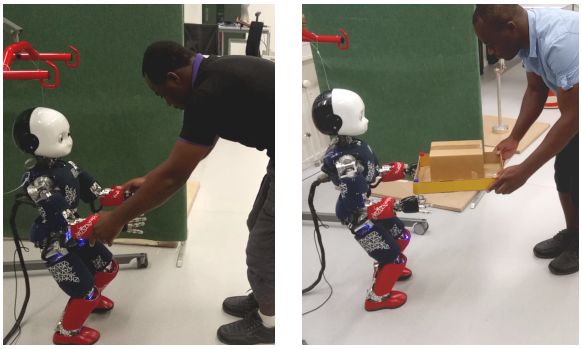

Figure 1. *iCub walking: left- in reaction to persistent pulling force, and right - in reaction to human's intention in a cooperative transporting task*

I. Introduction

THE potential ability of humanoid robots to operate in unstructured environments with narrow passages and limited support areas renders them very useful for service robotics. They could be employed for underground and planetary explorations, rescue operations after disaster, etc. Moreover, their anthropomorphic structure enables them to perform tasks in environment designed for humans and potentially to better collaborate with humans. However, working in such an environment is really challenging; the robots might be pulled or pushed while interacting with humans. They could possibly bump into obstacles, step on small object lying on the floor or walk on different surfaces, etc. Therefore, their balance and gait must be not only stable but also robust in the presence of such disturbances for the robots to successfully complete their missions.

Different concepts have been proposed to ensure the stability of bipedal robots, with the most popular being based on the Zero Moment Point (ZMP). The latter must stay within the support polygon for the robot to be stable. Following this approach, the footsteps are generally planned in advance, which reduces the problem to the generation of CoM trajectories consistent with the robot dynamics or with approximation of this dynamics [1][2] and which satisfy the desired ZMP. For instance in [1] the desired walking patterns were generated using a preview controller. In [3], the constraints on the ZMP were explicitly enforced with a Linear Model Predictive controller.

Michael Bombile is with the Learning Algorithms and Systems Laboratory, EPFL, Lausanne, Switzerland, e-mail: michael.bombile@epfl.ch.

Aude Billard is with the Learning Algorithms and Systems Laboratory, EPFL, Lausanne, Switzerland, e-mail: aude.billard@epfl.ch.

This paper focuses on cases where the footsteps cannot be planned beforehand, but have to be automatically determined on the fly. This reactive behavior is particularly important when the humanoid robot is driven by a high level task-related objective which provides reference commands in form of velocities or accelerations to be followed by the walking robot (e.g. while transporting cooperatively an object, etc.) or tasks which could generate significant change of the robot's momentum (e.g. when handing over or throwing a heavy object to the robot). Moreover, such a reactive behavior of the robot is suitable for the mitigation of the effects of other disturbances such as pushing or pulling forces, ground unevenness, etc., that the robot could be subject to while performing its prescribed task.

A. Related Works

Reactive stepping has been largely addressed in the humanoid community particularly in dealing with large perturbations such as pushes, collisions or tripping. Besides moving the ZMP or Center of Pressure (CoP) within the support polygon [4][5][6], the robot could either accelerate its angular momentum (through trunk or upper limbs motions) or even take a step in order to prevent a fall or to come to a stop [2][7][8]. In this particular case, the "Capture-Point" (CP) [2] also called "Extrapolated Center of Mass" [9] and defined as the point on the ground where a biped robot should step to in order to come to a complete stop, has proven to be very effective. In [10], it was even used to define a concept for the stability analysis of legged locomotion, namely the "N-step capturability", which was then validated in [11].

Besides push recovery, Englsberger et al. [12] suggested the control of the unstable dynamics of the CP and used it to ease the generation of walking patterns. Dynamically consistent reference trajectories of the CP were generated

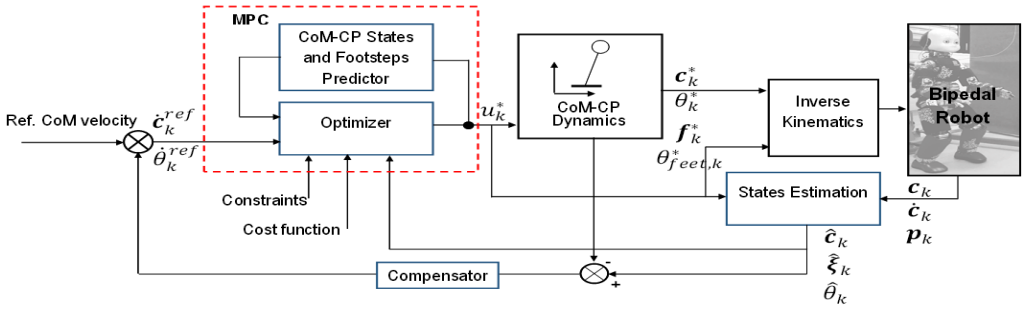

Figure 2. *Proposed MPC based reactive walking controller. Given the estimated CoM and CP states* $(\hat{\boldsymbol{c}}_k, \hat{\boldsymbol{\xi}}_k, \hat{\boldsymbol{\theta}}_k)$ *of the robot (under perturbation or not), the controller generates motions of the CoM* $(\boldsymbol{c}^*_k, \theta^*_k)$ *and of the feet* $(\boldsymbol{f}^*_k, \theta^*_{feet,k})$ *to steer the CoM translational and rotational velocities towards their reference* $(\dot{\boldsymbol{c}}^{ref}_k, \dot{\theta}^{ref}_k)$ *(feedforward + feedback coming from the Compensator block).* u^*_k *denotes the optimized decision variables and* p_k *the ZMP.*

thanks to a backward recursive method proposed in [13]. The stability constraints, however, were not always met by the computed ZMP, which was then projected onto the support polygon, yielding thus some discontinuities in the controller output. To address the above problem, Krausse et al., [14] suggested a formulation of the CP controller within MPC framework. Using MPC, the 3D extension of the CP called divergence component of motion (DCM) is controlled in [15] for the ajustment of footstep positions and orientations in the presence of disturbance. In [16], a CP based MPC is used for push recovery. The ZMP is controlled to stay within the support polygon while the CP is steered towards its desired location by modulating the Centroidal Moment Pivot (CMP) through the angular momentum. This approach was extented in [17] in order to include the ajustment of footstep positions while walking. Although benefiting from the robustness inherent to the CP control, the above MPC based works still rely on predefined footstep placements, and therefore cannot be used, as they are, for reactive walking.

B. Contribution

In order to develop a reactive walking controller, this work draws inspiration from *Herdt et al.*, [18] automatic footstep placement algorithm, and of the footstep orientations [19]. It extends these algorithms designed for the ZMP tracking to the CP tracking so as to leverage the robustness inherent to the latter. As a result, this paper proposes a CP based controller that can not only adapt the CoP's position within the support polygon, but also generate automatically reference trajectories of the CP and their related footstep positions and orientations in response to perturbations or to reference velocities to be tracked by the robot. Also, besides the theoretical development, this work validates experimentally the proposed controller.

The remainder of this paper is organized as follows. After briefly recalling the CoM-CP dynamics, Section II presents the proposed reactive omnidirectional balance and walking controller. In Section III, experimental results validating the proposed controller are presented and briefly discussed. Finally Section IV concludes the paper and provides some future perspectives.

II. Capture Point Based Reactive Omnidirectional Walking Controller

This section presents the development of the proposed walking controller able to generate reactively not only the footstep positions, but also their orientations.

A. Problem Formulation

Following the MPC approaches [18], [20], [21] and [14], this paper's objective can be mathematically synthesized with a cost function of the form

$$J = J_{V_{CoM}} + J_{f_{Pose}} + J_{ctrl}$$

subject to balance and walking constraints. When optimizing J over a horizon N:

- $J_{V_{CoM}}$ ensures that the actual velocity of the CoM frame tracks its reference velocity (translation + rotation),
- $J_{f_{Pose}}$ must ensure that the actual CP position and footstep orientation follow their auto-generated references,
- J_{ctrl} must ensure that the control effort to achieve the previous objectives is kept minimal,

Solving the above problem will require: *1)* a CoM-CP dynamics and a rotational dynamics of the robot with respectively translational velocity and angular velocity as one of their state variables, *2)* a model for the auto-generation of the CP reference trajectory, and *3)* a formulation of walking constraints that have to be respected by the robot.

In the sequel, each of the above requirements will be addressed. A block diagram illustrating the proposed MPC based solution is given in Figure 2.

B. CoM-CP Dynamics

1) CoM-CP System Dynamics: This is the dynamics that will be used to propagate the states over the MPC's prediction horizon. Exploiting the 3D Linear Inverted Pendulum Model (3D-LIPM) [22], [1] it is possible to derive different CoM-CP dynamical models that have the CoM velocity as state. However, the walking motion being characterized by an unavoidable sway motion, the CoM should better track its reference on average rather than instantaneously [18].

Thus, if the position of the CoM with respect to a fixed inertial frame $\mathcal{W}$ is denoted by $\boldsymbol{c} = \begin{bmatrix} c_x & c_y & c_z \end{bmatrix}^\top$, the

CP will be defined as [2] $\xi_h \triangleq c_h + \frac{1}{\omega}\dot{c}_h$, where $(h \equiv x,\, y)$, $\omega \triangleq \sqrt{g/c_z}$, with g denoting the gravity acceleration.

The overall CoM-CP dynamics will be given by [12]

$$\begin{bmatrix} \dot{c}_h(t) \\ \dot{\xi}_h(t) \end{bmatrix} = \begin{bmatrix} -\omega & \omega \\ 0 & \omega \end{bmatrix} \begin{bmatrix} c_h(t) \\ \xi_h(t) \end{bmatrix} + \begin{bmatrix} 0 \\ -\omega \end{bmatrix} p_h(t) \quad (1)$$

where t denotes the time and p_h is the position of the ZMP.

2) CoM-CP Prediction Model: After discretizing (1) with the state vector defined as $\boldsymbol{x}_k \triangleq \begin{bmatrix} c_k & \xi_k \end{bmatrix}^\top$(index h dropped to simplify notation), the prediction model over a horizon N of each state of (1) can be written as

$$\begin{aligned} \underrightarrow{\boldsymbol{c}}_k &= S_c \boldsymbol{x}_k + U_c \underrightarrow{\boldsymbol{p}}_{k-1} \\ \underrightarrow{\boldsymbol{\xi}}_k &= S_\xi \boldsymbol{x}_k + U_\xi \underrightarrow{\boldsymbol{p}}_{k-1} \end{aligned} \quad (2)$$

where the notation $\underrightarrow{\cdot}_k$ means a stack of N *predicted* values of the considered variable, starting from but not including the time index k. Here $S_j \in \mathbb{R}^{N\times 2}$ and $U_j \in \mathbb{R}^{N\times N}$ $(j \equiv c, \xi)$ represent respectively the sub-matrices associated with the state c or ξ of the matrices $\mathbf{S}_s$ and $\mathbf{U}_p$ given by

$$\mathbf{S} \triangleq \begin{bmatrix} \mathbf{A} \\ \vdots \\ \mathbf{A}^N \end{bmatrix}, \quad \mathbf{U} \triangleq \begin{bmatrix} \mathbf{A}^0\mathbf{B} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{A}^{N-1}\mathbf{B} & \cdots & \mathbf{A}^0\mathbf{B} \end{bmatrix}$$

where the state transition matrix $\mathbf{A}$ and the control vector $\mathbf{B}$ of the discrete model of (1) are given by

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} e^{-\omega T} & \frac{1}{2}e^{\omega T}\left(1 - e^{-2\omega T}\right) \\ 0 & e^{\omega T} \end{bmatrix} \\ \mathbf{B} &= \begin{bmatrix} 1 - \frac{1}{2}e^{\omega T}\left(1 + e^{-2\omega T}\right) \\ 1 - e^{\omega T} \end{bmatrix} \end{aligned} \quad (3)$$

with T is the sampling time.

3) CoM Average Velocity: Due to the sway motion of the robot while walking, the average CoM velocity would be conveniently computed between every two footsteps. If, for instance, four footsteps are considered within the MPC horizon N, the average velocity can be written as

$$\underrightarrow{\dot{\boldsymbol{c}}}_k = E \underrightarrow{\boldsymbol{c}}_k \quad \text{with} \quad E = \frac{1}{2T_{sp}} \begin{bmatrix} -I & I \\ -I & I \end{bmatrix} \quad (4)$$

Here $I \in \mathbb{R}^{\frac{N}{2}\times\frac{N}{2}}$ is a unit matrix and T_{sp} a step's duration.

C. Robot's Orientation

1) Angular Trajectory of the robot: In order to follow a rotational velocity, we attach a frame to the CoM and consider only the yaw angle θ (assuming an upright posture). Using the jerk $\dddot{\theta}_k$ as a control variable, the rotational motion of the CoM frame can be described by [18]

$$\begin{bmatrix} \theta_{k+1} \\ \dot{\theta}_{k+1} \\ \ddot{\theta}_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A}_\theta} \underbrace{\begin{bmatrix} \theta_k \\ \dot{\theta}_k \\ \ddot{\theta}_k \end{bmatrix}}_{\boldsymbol{\theta}_k} + \underbrace{\begin{bmatrix} \frac{T^3}{6} \\ \frac{T^2}{2} \\ T \end{bmatrix}}_{\mathbf{B}_\theta} \dddot{\theta}_k \quad (5)$$

Similarly to (2), the prediction model related to the first two states of (5) over the horizon N can be written as

$$\begin{aligned} \underrightarrow{\boldsymbol{\theta}}_k &= S_\theta \boldsymbol{\theta}_k + U_\theta \underrightarrow{\ddot{\boldsymbol{\theta}}}_{k-1} \\ \underrightarrow{\dot{\boldsymbol{\theta}}}_k &= S_{\dot{\theta}} \boldsymbol{\theta}_k + U_{\dot{\theta}} \underrightarrow{\ddot{\boldsymbol{\theta}}}_{k-1} \end{aligned} \quad (6)$$

Computed as in (2) but with $(\mathbf{A}_\theta,\, \mathbf{B}_\theta)$ instead of $(\mathbf{A},\, \mathbf{B})$, $S_j \in \mathbb{R}^{N\times 3}$ and $U_j \in \mathbb{R}^{N\times N}$ $(j \equiv \theta, \dot{\theta}, \ddot{\theta})$ represent the sub-matrices associated with the state θ, $\dot{\theta}$ and $\ddot{\theta}$ over the horizon N. Thus, from (4) and (6) $J_{V_{CoM}}$ can now be written as

$$J_{V_{CoM}} = \sum_{h=x,y} \frac{\beta}{2} \left\| \underrightarrow{\dot{\boldsymbol{c}}}_k - \underrightarrow{\dot{\boldsymbol{c}}_h}_k^{ref} \right\|^2 + \frac{\alpha_\theta}{2} \left\| \underrightarrow{\dot{\boldsymbol{\theta}}}_k - \underrightarrow{\dot{\boldsymbol{\theta}}}_k^{ref} \right\|^2 \quad (7)$$

where $\underrightarrow{\dot{\boldsymbol{c}}_h}_k^{ref} \in \mathbb{R}^{N\times 1}$ and $\underrightarrow{\dot{\boldsymbol{\theta}}}_k^{ref} \in \mathbb{R}^{N\times 1}$ are the translational and rotational reference velocity of the CoM frame. β and α_θ are weights of the cost function.

D. Self-generated Capture-Point Reference Trajectories

The reference trajectory of the CP, $\xi^*(t)$, is derived from the solution of the CP dynamics (1). Hence, for constant ZMP position, $\xi^*(t)$ can be written as

$$\xi^*(t) = e^{\omega t}\xi_{ini} + (1 - e^{\omega t})f \quad (8)$$

where ξ_{ini} and f are respectively the initial CP and the footstep position (fixed ZMP). Hence, for a sequence of m footsteps, if the CP at the end of the step i is denoted by $\xi_{eos,i}$, the dynamically consistent initial CP, $\xi_{ini,i}$, can be computed with a backward recursion as follows [23], [14]

$$\xi_{ini,i} = \xi_{eos,i}e^{-\omega T_{sp}} + (1 - e^{-\omega T_{sp}})f_i \quad (9)$$

$$\xi_{eos,i-1} = \xi_{ini,i} \quad (10)$$

Thus, starting from the final preplanned footstep (f_m and $\xi_{eos,m}$), the $\xi_{ini,i}$ are computed down to the current footstep.

Unlike in [23], [14], [15], [17] where the reference footsteps were predetermined, in this work, they are formulated as variables such that they can be generated automatically on-line. To that end, let us consider that there are only four $(m = 4)$ footsteps (f_1, f_2, f_3, f_4) falling within the receding horizon N (it only eases the computation of the CoM's average velocity in the rest of these developments). Thus, starting from the fourth footstep and using (8), it can be shown that all $\xi_{ini,i}$ will be given by

$$\boldsymbol{\xi}_{ini,1:4} = N_e\, \xi_{eos,4} + \mathbf{M}_e\, \boldsymbol{f}_{1:4} \quad (11)$$

where $\xi_{eos,4} = \xi_{eos,m} = \xi_N$ is the end of horizon CP and

$$\begin{aligned} \boldsymbol{\xi}_{ini,1:4} &\triangleq \begin{bmatrix} \xi_{ini,1} & \xi_{ini,2} & \xi_{ini,3} & \xi_{ini,4} \end{bmatrix}^\top \\ \boldsymbol{f}_{1:4} &\triangleq \begin{bmatrix} f_1 & f_2 & f_3 & f_4 \end{bmatrix}^\top \\ N_e &\triangleq \begin{bmatrix} e^{-4\omega T_{sp}} & e^{-3\omega T_{sp}} & e^{-2\omega T_{sp}} & e^{-\omega T_{sp}} \end{bmatrix}^\top \\ \mathbf{M}_e &\triangleq \delta_{e_\omega} \begin{bmatrix} 1 & e^{-\omega T_{sp}} & e^{-2\omega T_{sp}} & e^{-3\omega T_{sp}} \\ 0 & 1 & e^{-\omega T_{sp}} & e^{-2\omega T_{sp}} \\ 0 & 0 & 1 & e^{-\omega T_{sp}} \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

with $\delta_{e_\omega} \triangleq (1 - e^{-\omega T_{sp}})$. Substituting now (11) in (8), the four CP reference trajectories can be shown to be given by

$$\boldsymbol{\xi}^*(t) = \Big[\underbrace{e^{\omega t}\mathbf{M}_e + (1-e^{\omega t})\mathbf{I}_m}_{\boldsymbol{\Xi}_{\boldsymbol{f}_m}} \quad \underbrace{e^{\omega t}N_e}_{\boldsymbol{\Xi}_{\xi_N}}\Big] \begin{bmatrix} \boldsymbol{f}_{1:m} \\ \xi_N \end{bmatrix} \tag{12}$$

Separating the known current footstep position f_1 of $\boldsymbol{f}_{1:m}$ from the future footsteps $\boldsymbol{f}_{2:m}$, the latter with ξ_N being unknowns to be determined through optimization, the reference CP $\boldsymbol{\xi}^*_{1:m}(t)$ for the m footsteps can now be written as

$$\boldsymbol{\xi}^*(t) = \boldsymbol{\Xi}_{\boldsymbol{f}_m} V_c f_1 + \boldsymbol{\Xi}_{\boldsymbol{f}_m} \mathbf{V}_f \boldsymbol{f}_{2:m} + \boldsymbol{\Xi}_{\xi_N} \xi_N \tag{13}$$

where the substitution $\boldsymbol{f}_{1:m} = V_c f_1 + \mathbf{V}_f \boldsymbol{f}_{2:m}$ was used and where $V_c \in \mathbb{R}^{m\times 1}$ and $\mathbf{V}_f \in \mathbb{R}^{m\times(m-1)}$ are constant selection vector and matrix given by

$$V_c \triangleq \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \ \mathbf{V}_f \triangleq \begin{bmatrix} 0 & \cdots & 0 \\ 1 & \ddots & \vdots \\ 0 & \ddots & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{14}$$

Note that (13) defines a piecewise continuous trajectory whose discontinuity stems from the discrete footsteps. Between two successive footsteps, this trajectory is continuous and can be obtained for each footstep (each row of (13)) by varying the time t between $[0,\ T_{sp}]$ (reinitialized due to change of initial condition for each footsteps). Over the prediction horizon N, the overall CP reference trajectory $\underrightarrow{\boldsymbol{\xi}}^*_k \in \mathbb{R}^{N\times 1}$ will be obtained by superimposing the discretized CP reference trajectories of each of the m footsteps.

E. Footstep Positions and Orientations

1) Footstep Orientations: During bipedal locomotion, the footstep positions are discrete over time and so are their orientations. If it is assumed that there is no slippage between the foot and the ground, then the current footstep orientation denoted $\theta^{\mathrm{w}}_{f,1}$ is fixed and known with respect to the inertial frame $\mathcal{W}$. In such a case, only the future footstep orientations denoted $\boldsymbol{\theta}^{\mathrm{w}}_{f,2:m}$ have to be determined. Hence, the footstep orientations over the prediction horizon can be written as

$$\underrightarrow{\boldsymbol{\theta}}^{*\mathrm{w}}_k = H^c_{k+1}\theta^{\mathrm{w}}_{f,1} + \boldsymbol{H}^f_{k+1}\boldsymbol{\theta}^{\mathrm{w}}_{f,2:m} \tag{15}$$

where $H^c_{k+1} \in \mathbb{R}^{N\times 1}$ and $\boldsymbol{H}^f_{k+1} \in \mathbb{R}^{N\times(m-1)}$ are cyclic vector and matrix associating each sample instant with a footstep. They are the same as those used for translations in [18], [20]. The superscript w indicates a reference to the frame $\mathcal{W}$. Thus, in order to ensure the desired orientation of the robot, the variables to optimize are the future footstep orientations $\boldsymbol{\theta}^{\mathrm{w}}_{f,2:m}$ in (15) and the jerk $\underrightarrow{\ddot{\boldsymbol{\theta}}}_{h,k-1}$ in (6).

2) Footstep Positions: When accounting for the rotational motion of the robot, the X and Y translations which were previously independent will now be coupled through a non-linear mapping function of the orientation angle. For instance, consider Fig. 3 depicting a sequence of three footsteps. If f^{w}_1 has an orientation θ_1 with respect to the inertial frame $\mathcal{W}$, the position of f^{w}_2 with respect to $\mathcal{W}$ will be given by

$$\begin{aligned} x^{\mathrm{w}}_2 &= x^{\mathrm{w}}_1 + cos\theta_1 x^1_2 - sin\theta_1 y^1_2 \\ y^{\mathrm{w}}_2 &= y^{\mathrm{w}}_1 + sin\theta_1 x^1_2 + cos\theta_1 y^1_2 \end{aligned} \tag{16}$$

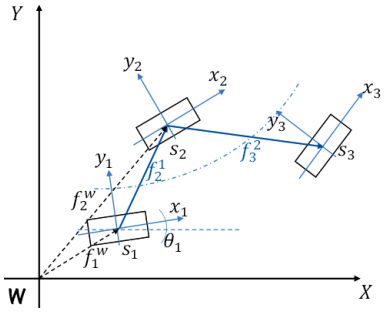

Figure 3. *Example of footstep positions when the robot performs a rotation.* $f^{\mathrm{w}}_1 = [x^{\mathrm{w}}_1 \quad y^{\mathrm{w}}_1]^\top$ *and* $f^{\mathrm{w}}_2 = [x^{\mathrm{w}}_2 \quad y^{\mathrm{w}}_2]^\top$ *denote respectively the positions of the footsteps* s_1 *and* s_2 *with respect to the fixed inertial frame* $\mathcal{F}_{\mathrm{w}}$ *(the world frame), while* $f^i_j = [x^i_j \quad y^i_j]^\top$ *represents the position of the footstep* s_j *relative to the footstep* s_i.

Equation (16) shows clearly the introduced nonlinearity. However, keeping the MPC linear or reducing the induced non-linearity will be beneficial for a real-time implementation of the controller. To that end, let us rewrite (16) as follows

$$f^{\mathrm{w}}_2 = f^{\mathrm{w}}_1 + R^{\mathrm{w}}_{s_1} f^1_2 \tag{17}$$

where $R^{\mathrm{w}}_{s_1} = \begin{bmatrix} cos\theta_1 & -sin\theta_1 \\ sin\theta_1 & cos\theta_1 \end{bmatrix}$.

The absolute footstep position (with respect to fixed inertial frame) f^{w}_2 is written as affine transformation of the relative footstep f^1_2. Thus, considering four footsteps ($m = 4$) over the horizon N, starting from the i^{th} footstep denoted s_i and following (17), the positions of the four footsteps $\boldsymbol{f}^{\mathrm{w}}_{i:i+3}$ and the end of horizon CP (end of the $i^{th}+3$ step), $\boldsymbol{\xi}^{\mathrm{w}}_N$, can be written as

$$\boldsymbol{f}^{\mathrm{w}}_{i:i+3} = \mathbf{1}_4 \boldsymbol{f}^{\mathrm{w}}_{s_i} + \mathbf{R}_{xy} \triangle \boldsymbol{f}_{i+1:3} \tag{18}$$

$$\boldsymbol{\xi}^{\mathrm{w}}_N = \boldsymbol{f}^{\mathrm{w}}_{s_i} + \mathbf{R}_{xy(3:)} \triangle \boldsymbol{f}_{i+1:3} + R^{\mathrm{w}}_{s_{i+2}} \boldsymbol{\xi}^{s_{i+2}}_N \tag{19}$$

with

$$\triangle \boldsymbol{f}_{i+1:3} \triangleq \begin{bmatrix} \boldsymbol{f}^{s_i}_{s_{i+1}} \\ \boldsymbol{f}^{s_{i+1}}_{s_{i+2}} \\ \boldsymbol{f}^{s_{i+2}}_{s_{i+3}} \end{bmatrix}, \ R^w_{s_i} \triangleq \begin{bmatrix} cos\theta_{s_i} & -sin\theta_{s_i} \\ sin\theta_{s_i} & cos\theta_{s_i} \end{bmatrix} \tag{20}$$

$$\boldsymbol{f}^{\mathrm{w}}_{i:i+3} \triangleq \begin{bmatrix} \boldsymbol{f}^{\mathrm{w}}_{s_i} \\ \boldsymbol{f}^{\mathrm{w}}_{s_{i+1}} \\ \boldsymbol{f}^{\mathrm{w}}_{s_{i+2}} \\ \boldsymbol{f}^{\mathrm{w}}_{s_{i+3}} \end{bmatrix}, \mathbf{R}_{xy} \triangleq \begin{bmatrix} 0 & 0 & 0 \\ R^{\mathrm{w}}_{s_i} & 0 & 0 \\ R^{\mathrm{w}}_{s_i} & R^{\mathrm{w}}_{s_{i+1}} & 0 \\ R^{\mathrm{w}}_{s_i} & R^{\mathrm{w}}_{s_{i+1}} & R^{\mathrm{w}}_{s_{i+2}} \end{bmatrix}, \tag{21}$$

where $\mathbf{1}_4 \in \mathbb{R}^{4\times 1}$ is a unit vector and in (19) $\mathbf{R}_{xy(3:)} \in \mathbb{R}^{2\times 6}$ is the fourth row of $\mathbf{R}_{xy}$. $\boldsymbol{f}^{s_i}_{s_{i+1}} \triangleq [\ f^{s_i}_{x,s_{i+1}} \quad f^{s_i}_{y,s_{i+1}}]^\top$ and $\boldsymbol{\xi}^{s_i}_N \triangleq [\xi^{s_i}_{x,N} \quad \xi^{s_i}_{y,N}]^\top$ are respectively the relative positions of the step s_{i+1} and the end of horizon CP with respect to the step s_i.

Using equations (18) and (19) in conjunction with the definitions (20)-(21), and separating the x from the y components of the future steps $\left(\boldsymbol{f}^{\mathrm{w}}_{s_{i+1}}, \boldsymbol{f}^{\mathrm{w}}_{s_{i+2}}, \boldsymbol{f}^{\mathrm{w}}_{s_{i+3}}\right)$, it can be shown that the considered four footstep positions expressed in a fixed inertial frame can be written as

$$\begin{cases} \boldsymbol{f}^{\mathrm{w}}_{x,i:i+3} &= (V_c + \mathbf{V}_f \mathbf{1}_3)\, f^{\mathrm{w}}_{x,i} + V_f \mathbf{R}_x \triangle \boldsymbol{f}_{i+1:3} \\ \boldsymbol{f}^{\mathrm{w}}_{y,i:i+3} &= (V_c + \mathbf{V}_f \mathbf{1}_3)\, f^{\mathrm{w}}_{y,i} + V_f \mathbf{R}_y \triangle \boldsymbol{f}_{i+1:3} \end{cases} \tag{22}$$

The matrices $\mathbf{R}_x$ and $\mathbf{R}_y \in \mathbb{R}^{3\times 6}$ are defined as

$$\mathbf{R}_x \triangleq \begin{bmatrix} c\theta_{s_i} & -s\theta_{s_i} & 0 & 0 & 0 & 0 \\ c\theta_{s_i} & -s\theta_{s_i} & c\theta_{s_{i+1}} & -s\theta_{s_{i+1}} & 0 & 0 \\ c\theta_{s_i} & -s\theta_{s_i} & c\theta_{s_{i+1}} & -s\theta_{s_{i+1}} & c\theta_{s_{i+2}} & -s\theta_{s_{i+2}} \end{bmatrix},$$

$$\mathbf{R}_y \triangleq \begin{bmatrix} s\theta_{s_i} & c\theta_{s_i} & 0 & 0 & 0 & 0 \\ s\theta_{s_i} & c\theta_{s_i} & s\theta_{s_{i+1}} & c\theta_{s_{i+1}} & 0 & 0 \\ s\theta_{s_i} & c\theta_{s_i} & s\theta_{s_{i+1}} & c\theta_{s_{i+1}} & s\theta_{s_{i+2}} & c\theta_{s_{i+2}} \end{bmatrix},$$

where $s\theta_{s_i}$ and $c\theta_{s_i}$ stand for $sin\theta_{s_i}$ and $cos\theta_{s_i}$, respectively.

Now, in order to generate automatically the reference trajectory of the CP, the positions (22) can in turn be substituted in (13). Thus, the decision variables to be determined in this case are now the relative footstep positions given by

$$\triangle \boldsymbol{f}_{i+1:3} = \begin{bmatrix} f_{x,s_{i+1}}^{s_i} & f_{y,s_{i+1}}^{s_i} & f_{x,s_{i+2}}^{s_{i+1}} & f_{y,s_{i+2}}^{s_{i+1}} & f_{x,s_{i+3}}^{s_{i+2}} & f_{y,s_{i+3}}^{s_{i+2}} \end{bmatrix}^\top$$

instead of their absolute values $\boldsymbol{f}_{i:i+3}^{w}$ as in [18][20] or [21]. Similar reasoning applies also for $\boldsymbol{\xi}_N$.

Thus, $J_{f_{pose}}$ and J_{ctrl} can now be written as

$$\begin{aligned} J_{f_{Pose}} &= \sum_{h=x,y} \tfrac{\gamma}{2} \left\| \underrightarrow{\boldsymbol{\xi}_h}_k - \underrightarrow{\boldsymbol{\xi}_h^*}_k \right\|^2 + \tfrac{\gamma_\theta}{2} \left\| \underrightarrow{\boldsymbol{\theta}}_k - \underrightarrow{\boldsymbol{\theta}^{*\mathrm{w}}}_k \right\|^2 \\ J_{ctrl} &= \sum_{h=x,y} \tfrac{\kappa}{2} \left\| \underrightarrow{\Delta \boldsymbol{p}_h}_{k-1} \right\|^2 + \tfrac{\kappa_\theta}{2} \left\| \underrightarrow{\ddot{\boldsymbol{\theta}}}_{k-1} \right\|^2 \end{aligned} \tag{23}$$

where $\underrightarrow{\Delta \boldsymbol{p}_h}_{k-1}$ is the variation of $\underrightarrow{\boldsymbol{p}_h}_{k-1}$. γ, γ_θ, κ and κ_θ are weights of the cost function.

F. Global Objective Function

From (7) and (23), the global objective function J to be minimized so as to generate the CoM trajectory and the footstep positions and orientations can now be written as

$$\begin{cases} J \triangleq \sum\limits_{h=x,y} \tfrac{\beta}{2} \left\| \underrightarrow{\dot{\boldsymbol{c}}}_k - \underrightarrow{\dot{\boldsymbol{c}}_h^{ref}}_k \right\|^2 + \tfrac{\kappa}{2} \left\| \underrightarrow{\Delta \boldsymbol{p}_h}_{k-1} \right\|^2 \\ \quad + \tfrac{\gamma}{2} \left\| \underrightarrow{\boldsymbol{\xi}_h}_k - \underrightarrow{\boldsymbol{\xi}_h^*}_k \right\|^2 \\ \quad + \tfrac{\alpha_\theta}{2} \left\| \underrightarrow{\dot{\boldsymbol{\theta}}}_k - \underrightarrow{\dot{\boldsymbol{\theta}}^{ref}}_k \right\|^2 + \tfrac{\kappa_\theta}{2} \left\| \underrightarrow{\ddot{\boldsymbol{\theta}}}_{k-1} \right\|^2 \\ \quad + \tfrac{\gamma_\theta}{2} \left\| \underrightarrow{\boldsymbol{\theta}}_k - \underrightarrow{\boldsymbol{\theta}^{*\mathrm{w}}}_k \right\|^2 \end{cases} \tag{24}$$

Writing this objective as a QP problem leads to

$$\boldsymbol{u}_k^* = \underset{\boldsymbol{u}_k}{\operatorname{argmin}} \left(\tfrac{1}{2} \cdot u_k^\top \mathbf{Q_k} u_k + \mathbf{p_k}^\top u_k \right) \tag{25}$$

$$\text{with} \quad \boldsymbol{u}_k \triangleq \begin{bmatrix} \underrightarrow{\boldsymbol{p}_x}_{k-1} & \underrightarrow{\boldsymbol{p}_y}_{k-1} & \triangle \boldsymbol{f}_{i+1:3} & \boldsymbol{\xi}_N^{s_{i+2}} & \underrightarrow{\ddot{\boldsymbol{\theta}}}_{k-1} & \boldsymbol{\theta}_{f,i+1:3}^{\mathrm{w}} \end{bmatrix}^\top$$

$$s.t \qquad \left| \triangle \boldsymbol{f}_{x,i} \right| \leq l_x \tag{26}$$

$$l_{yi} \leq \left| \triangle \boldsymbol{f}_{y,i} \right| \leq l_{yo} \tag{27}$$

$$\left| \theta_{s_{i+1}} - \theta_{s_i} \right| \leq l_\theta, \qquad \text{with } i = 1...3 \tag{28}$$

$$\boldsymbol{N}_k(\underrightarrow{\boldsymbol{\theta}^{*w}}_k) \begin{bmatrix} \underrightarrow{\boldsymbol{p}_x}_{k-1} - \underrightarrow{\boldsymbol{f}_x^{\mathrm{w}}}_{k-1} \\ \underrightarrow{\boldsymbol{p}_y}_{k-1} - \underrightarrow{\boldsymbol{f}_y^{\mathrm{w}}}_{k-1} \end{bmatrix} \leq \boldsymbol{b}_k \tag{29}$$

with $\underrightarrow{\boldsymbol{f}_h^{\mathrm{w}}}_{k-1} = \left(H_{k+1}^c + \boldsymbol{H}_{k+1}^f 1_3 \right) f_{h,i,k}^{\mathrm{w}} + \boldsymbol{H}_{k+1}^f \mathbf{R}_h \triangle \boldsymbol{f}_{i+1:3,k}$ and where l_x and l_θ represent respectively the upper bounds of the relative footstep longitudinal and angular displacements, while l_{yi} and l_{yo} represent respectively the lower and upper bounds of the relative footstep position in the lateral direction. $\boldsymbol{N}_k(\underrightarrow{\boldsymbol{\theta}^{*\mathrm{w}}}_k)$ and $\boldsymbol{b}_k$ represent respectively the matrix gathering the x and y components of the normals to the edges of the support polygon and the bounds of the latter in the direction of the normals. Also, in (25) we have

$$\mathbf{Q_k} = \begin{bmatrix} Q_k^{p_x} & 0 & Q_k^{p_x \triangle \boldsymbol{f}} & Q_k^{p_x \boldsymbol{\xi}_N} & 0 \\ 0 & Q_k^{p_y} & Q_k^{p_y \triangle \boldsymbol{f}} & Q_k^{p_y \boldsymbol{\xi}_N} & 0 \\ Q_k^{\triangle \boldsymbol{f}, p_x} & Q_k^{\triangle \boldsymbol{f}, p_y} & Q_k^{\triangle \boldsymbol{f}} & Q_k^{\triangle \boldsymbol{f} \boldsymbol{\xi}_N} & 0 \\ Q_k^{\boldsymbol{\xi}_N p_x} & Q_k^{\boldsymbol{\xi}_N p_y} & Q_k^{\boldsymbol{\xi}_N \triangle \boldsymbol{f}} & Q_k^{\boldsymbol{\xi}_N} & 0 \\ 0 & 0 & 0 & 0 & Q_k^{\theta} \end{bmatrix} \tag{30}$$

$$\mathbf{p_k} = \begin{bmatrix} p_k^{\boldsymbol{p}_x} & p_k^{\boldsymbol{p}_y} & p_k^{\triangle \boldsymbol{f}} & p_k^{\boldsymbol{\xi}_N} & p_k^{\ddot{\boldsymbol{\theta}}} & p_k^{\boldsymbol{\theta} f} \end{bmatrix}^\top \tag{31}$$

with the $Q_{h,ij}$ and $p_{k,i}$ elements given in Appendix A.

Note that the use in the proposed MPC formulation of relative instead of absolute footstep positions as decision variables has the advantage to transform a global problem into a local one. It allows us, unlike in [18] [21], to keep the MPC's footstep feasibility constraints linear ((26) and (27)) except the constraints on the CoP (29). However, based on (30) and the consideration that there is no obstacle in the biped's workspace, the angular variable (θ) of the foot is free to reach, independently from the x and y variables, its prescribed value dictated by the desired rotational velocity (provided that the latter is within the robot capabilities). Thus, the orientation can be solved in a separated MPC and the obtained values at each iteration substituted as parameters for the x and y variables. As a result, this MPC problem which is non-linear in nature due to the rotation can be solved sequentially as a linear problem [19], [24].

III. Results

This section presents the results of two kinds of reactive walking experiments carried out on the humanoid iCub (version 2.5). The first experiment is about velocity driven reactive walking task whereas the second is about interaction force based walking tasks.

From input velocity of the robot, the controller generates in real-time the reference trajectories of the CoM, the pose of the next footstep, which are sent to the inverse kinematics module. However, because the controller continuously adapts the pose of the next footstep depending on the input velocities or disturbances, quintic polynomial interpolation using the current and the predicted footstep pose at each iteration was used to ensure smooth 3D trajectories of the swing foot.

The MPC itself was solved with *qpOASES* [25] in an average time of $1\,ms$ on an $Intel^{(R)}\ Core\ i7$, $3.4\,GHz$ and $7.8\,GB$ RAM PC. However, because the inverse kinematics solver took $6-11\,ms$ for each foot, the sampling time was set at $T = 0.040\,s$. The other MPC's parameters were set as follows: step duration $T_{sp} = 0.640\,s$, gains $\beta = 0.20$, $\gamma = 1.50$, $\kappa = 0.80$, $\alpha_\theta = 1.0^{-6}$, $\beta_\theta = 1.00$, $\gamma_\theta = 1.00$, $\kappa_\theta = 0.10$.

A. Velocity Driven Omnidirectional Walking

This experiment validates the ability of the proposed locomotion controller to generate on-line stable and reactive omnidirectional walking trajectories and to stabilize the robot around them. Thus, the robot performs a combination of translations (longitudinal and lateral) and rotations. The desired velocity of the CoM or rather of a frame attached to the CoM is defined relative to the robot by the vector $[v_x\ v_y\ \omega_z]^T$, representing respectively the robot's longitudinal and lateral motion in $[m/s]$ and the rotation motion in $[rad/s]$.

The sequence of desired motion performed by the robot during this experiment is summarized in Table I.

Table I
DESIRED VELOCITIES DURING THE FIRST EXPERIMENT

$v_x\ [m/s]$	$v_y\ [m/s]$	$\omega_z\ [rad/s]$	$time\ [s]$
$+0.06$	0.03	0.00	6
0.00	0.00	-0.10	8
$+0.05$	0.00	-0.05	6

At the beginning, the robot translates in both longitudinal and lateral directions, then performs a pure rotation and finally combines a translation and a rotation. As can be seen in Fig. 4 depicting the trajectories of the CP and the CoM, the robot followed its prescribed velocities and reference trajectories while staying stable. The ZMP, here the control

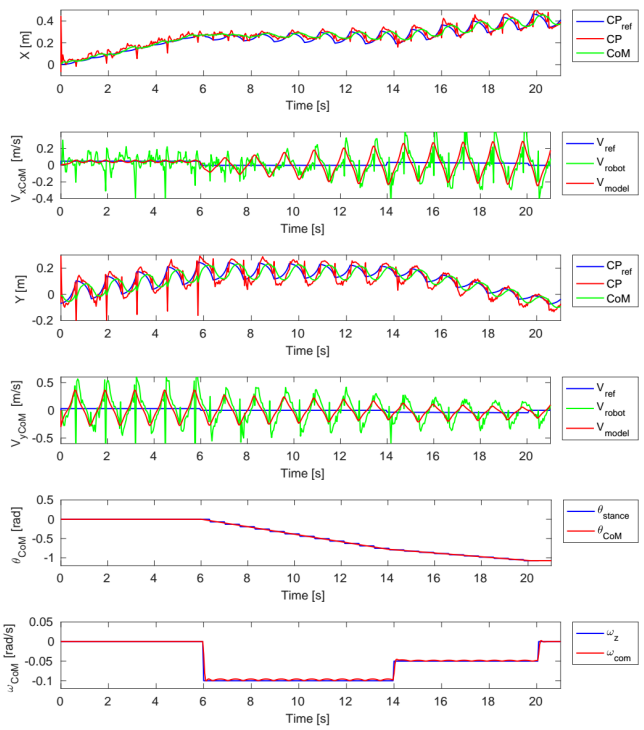

Figure 4. *CP and CoM trajectories and velocities during the tracking task*

variable, varies along its reference values generated by the algorithm. However, it stays within the support polygon as can be seen in Fig. 5, which depicts the footstep poses, the ZMP (reference and actual) and their associated support polygons. If, for instance, the ZMP was kept constant at the 6^{th}second when the CoM state changed abruptly, the CP would have evolved from that new state according to (8). Consequently, it would have required a much bigger step to maintain the balance of the robot. However, because the ZMP is free to move within the support polygon, the proposed controller reacted by computing a minimal action that steered the robot state back towards its desired value.

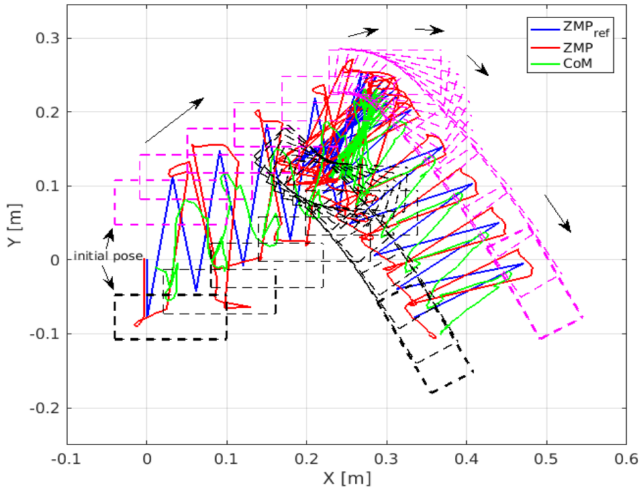

Figure 5. *Footstep positions and orientations during the velocity tracking task. The rectangles represent the support polygon of each foot*

B. Interaction Force based Reactive Walking

In this category two experiments were conducted: a guidance and a transportation tasks. They illustrate hypothetical collaboration tasks between the robot (follower) and a human with whom the robot interacts through forces. To move the robot in a given direction, the human who acts as leader can either push or pull the robot or even force it to rotate. The robot must generate stable walking motions that comply with the intention of the leader. The first uses the ZMP feedback and the second the arm forces/torques sensor information to detect the leader intention.

Fig. 6 and Fig. 7 show the results of the feedback based reactive walking. Throughout this experiment, the desired velocity of the robot was set to zero. At the beginning the velocity loop was open and then closed after $4s$. The robot started by rocking around its initial position before being pulled continuously by its arms in the longitudinal direction and then pushed and pulled in the lateral direction. Furthermore, a torque around the vertical axis was exerted on the robot in both clockwise and anticlockwise direction before stopping the experiment, which lasted $60s$.

The induced changes of the ZMP and the vertical moment with respect to their reference values are interpreted as perturbations to the desired state of the robot. To reduce these perturbations, the compensator appearing in Fig. 2 computed the feedback velocities shown in Fig. 6. The short delay observed between the velocities and the robot's motion is

due to a low-pass filter embedded in the compensator. In Fig. 7, it can be seen that the generated trajectories keep the robot stable as its ZMP stays within the support polygon.

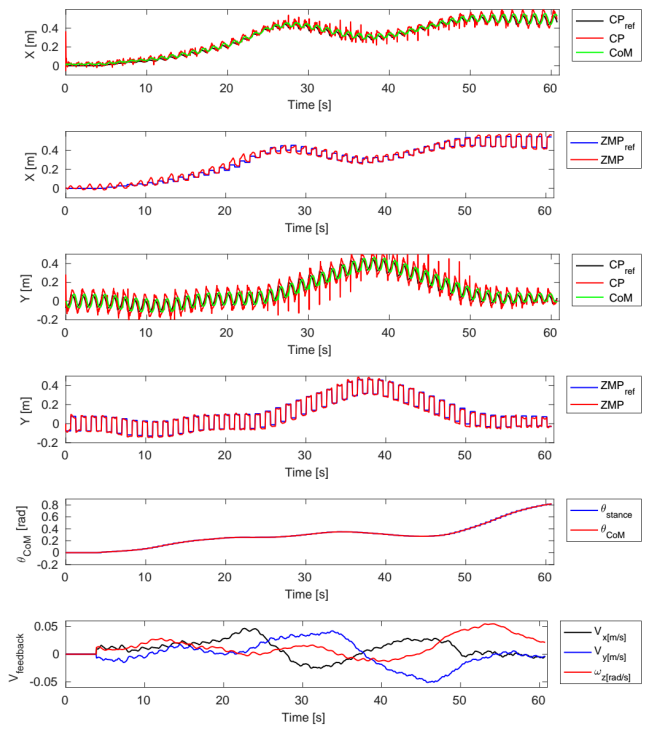

Figure 6. *CP, ZMP and CoM trajectories and feedback velocities generated by the controller in response to forces/torques exerted on the robot. ZMP_{ref} denotes the automatically generated reference footstep position.*

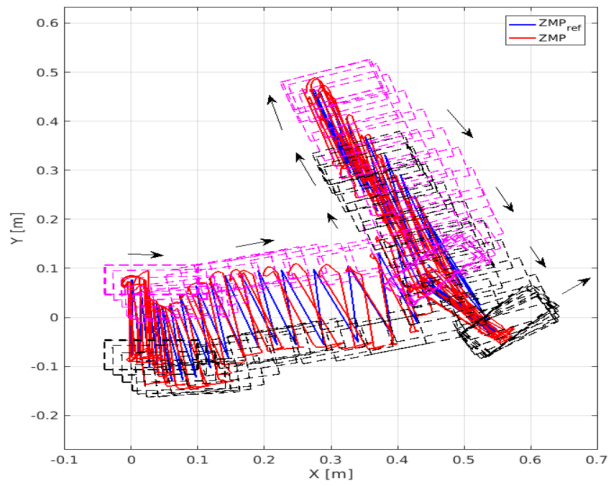

Figure 7. *Footstep poses during the interactive guidance task*

For the transportation task, the forces and torques applied on the robot, through the transported object, are measured by the robot's arms forces sensors and converted by an admittance law into velocities to be tracked by the robot. As this reduces to a tracking problem, only some snapshots are shown in Fig. 8 and a video [26] of all experiments is provided as supplementary material.

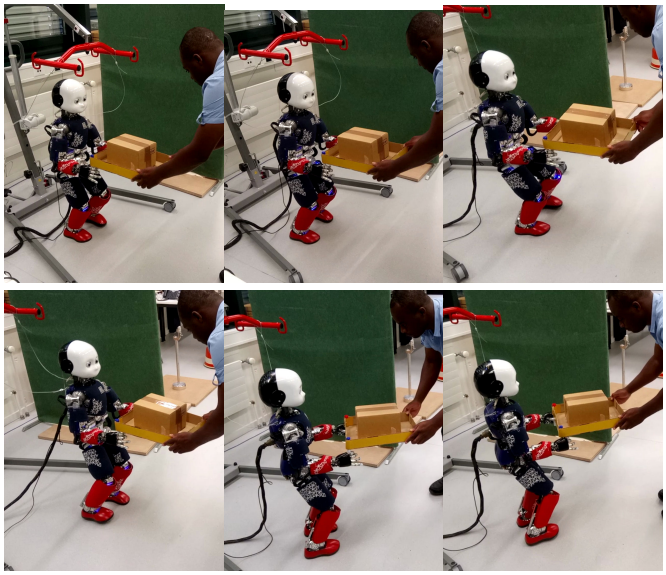

Figure 8. *Reactive walking in collaborative transportation task. The interaction forces sensed by the robot are converted into velocities to be tracked*

IV. Conclusion and Future Works

This paper presented a Capture-Point based walking controller able to generate reactively omnidirectional walking patterns for a biped robot and to stabilize the robot around them. By implementing the proposed controller on the humanoid robot iCub, its effectiveness was successfully demonstrated on two kinds of tasks where the classical walking approach based on footsteps planning could not apply. The first reactive walking experiment showed that the robot could track omnidirectional velocities and could even rotate around a spot when following a pure rotational velocity. The second experiment demonstrated how this ability to automatically generate stable omnidirectional walking motions could be further exploited in human-robot cooperative tasks. The robot, acting as a follower, successfully adapted its footsteps in order to comply with the intentions of the human, first in a guidance task and then in a cooperative transporting task.

As future works, the proposed algorithm, currently tested in position control mode, will be implemented with torque control in a whole body control framework. Thus, the linear and angular momentum of the robot could be explicitly regulated and the robot's motions made more compliant. This will reduce the high jerky motion observed during the experiments and could further improve the robustness.

Appendix

A. Hessian matrix and gradient vector of the QP

The $Q_{h,ij}$ $(h = x, y)$ and $p_{k,i}$ elements, respectively, of the Hessian matrix and gradient vector are given by

$$Q_k^{p_h} = \beta U_c^\top E^\top E U_c + \gamma U_\xi^\top U_\xi + \kappa \Gamma^\top \Gamma$$
$$Q_k^{p_h \triangle \boldsymbol{f}} = (Q_k^{\triangle \boldsymbol{f} p_h})^\top = -\gamma U_\xi^\top (\underrightarrow{\Xi_{\boldsymbol{f}_m}}_k \mathbf{V}_f \underrightarrow{\Xi_{\boldsymbol{f}_m}}_k \mathbf{R}_h + \underrightarrow{\Xi_{\xi_N}}_k \mathbf{R}_{h(3:)})$$

$$
\begin{aligned}
Q_k^{p_h \boldsymbol{\xi}_N} &= (Q_k^{\boldsymbol{\xi}_N p_x})^\top = -\gamma U_\xi^\top \underrightarrow{\Xi_{\xi_N}}_k R_{h,s_{i+2}}^w \\
Q_{k,h}^{\triangle \boldsymbol{f}} &= \gamma(\underrightarrow{\Xi_{\boldsymbol{f}_m}}_k \mathbf{V}_f \mathbf{R}_h + \underrightarrow{\Xi_{\xi_N}}_k \mathbf{R}_{h(3:)})^\top (\underrightarrow{\Xi_{\boldsymbol{f}_m}}_k \mathbf{V}_f \mathbf{R}_h + \underrightarrow{\Xi_{\xi_N}}_k \mathbf{R}_{h(3:)}) \\
Q_{k,h}^{\triangle \boldsymbol{f} \boldsymbol{\xi}_N} &= (Q_k^{\boldsymbol{\xi}_N \triangle \boldsymbol{f}})^\top = \gamma(\underrightarrow{\Xi_{\boldsymbol{f}_m}}_k \mathbf{V}_f \mathbf{R}_h + \underrightarrow{\Xi_{\xi_N}}_k \mathbf{R}_{h(3:)})^\top \underrightarrow{\Xi_{\xi_N}}_k R_{h,s_{i+2}}^w \\
Q_{k,h}^{\boldsymbol{\xi}_N} &= \gamma(\underrightarrow{\Xi_{\xi_N}}_k + R_{h,s_{i+2}}^w)^\top (\underrightarrow{\Xi_{\xi_N}}_k + R_{h,s_{i+2}}^w)
\end{aligned}
\tag{28}
$$

$$
\begin{cases}
Q_k^{\triangle \boldsymbol{f}} & \triangleq Q_{k,x}^{\triangle \boldsymbol{f}} + Q_{k,y}^{\triangle \boldsymbol{f}} \\
Q_k^{\triangle \boldsymbol{f} \boldsymbol{\xi}_N} & \triangleq Q_{k,x}^{\triangle \boldsymbol{f} \boldsymbol{\xi}_N} + Q_{k,y}^{\triangle \boldsymbol{f} \boldsymbol{\xi}_N} \\
Q_k^{\boldsymbol{\xi}_N} & \triangleq Q_{k,x}^{\boldsymbol{\xi}_N} + Q_{k,y}^{\boldsymbol{\xi}_N}
\end{cases}
$$

$$
Q_\theta = \begin{bmatrix} \alpha_\theta U_{\dot\theta}^\top U_{\dot\theta} + \gamma_\theta U_\theta^\top U_\theta + \kappa_\theta I & -\gamma_\theta U_\theta^\top H_{k+1}^f \\ -\gamma_\theta (H_{k+1}^f)^\top U_\theta & \gamma_\theta (H_{k+1}^f)^\top H_{k+1}^f \end{bmatrix},
\tag{32}
$$

and the vector $\mathbf{p_k} \triangleq \begin{bmatrix} \mathbf{p}_{\mathbf{k}_{xy}}^\top & \mathbf{p}_{\mathbf{k}\theta}^\top \end{bmatrix}^\top$, with

$$
\begin{aligned}
p_k^{\boldsymbol{p}_h} &= \kappa U_c^\top E^\top (E S_c \boldsymbol{x}_h(k) - \underrightarrow{\dot{\boldsymbol{c}}_h}_k^{ref}) \\
&\quad + \gamma U_\xi^\top (S_\xi \boldsymbol{x}_h(k) - \underrightarrow{\Xi_{\boldsymbol{f}_m}}_k (V_c + \mathbf{V}_f \mathbf{1}_3) f_{i,k}^w) \\
&\quad + \gamma U_\xi^\top \underrightarrow{\Xi_{\xi_N}}_k f_{i,k}^w - \kappa \Gamma^\top \boldsymbol{e}_1 p_{h,k-1} \\
p_k^{\triangle \boldsymbol{f}} &= -\gamma(\underrightarrow{\Xi_{\boldsymbol{f}_m}}_k \mathbf{V}_f \mathbf{R}_h + \underrightarrow{\Xi_{\xi_N}}_k \mathbf{R}_{h(3:)})^\top \\
&\quad (S_\xi \boldsymbol{x}_h(k) - (\underrightarrow{\Xi_{\boldsymbol{f}_m}}_k (V_c + \mathbf{V}_f \mathbf{1}_3) + \underrightarrow{\Xi_{\xi_N}}_k) f_{i,k}^w) \\
p_k^{\boldsymbol{\xi}_N} &= -\gamma(\underrightarrow{\Xi_{\xi_N}}_k R_{h,s_{i+2}}^w)^\top (\underrightarrow{\Xi_{\boldsymbol{f}_m}}_k V_c f_{i,k}^w - S_\xi \boldsymbol{x}_h(k)) \\
&\quad (S_\xi \boldsymbol{x}_h(k) - (\underrightarrow{\Xi_{\boldsymbol{f}_m}}_k (V_c + \mathbf{V}_f \mathbf{1}_3) + \underrightarrow{\Xi_{\xi_N}}_k) f_{i,k}^w)
\end{aligned}
$$

$$
\mathbf{p}_{\mathbf{k}\theta} = \begin{bmatrix} \alpha_\theta U_{\dot\theta}^\top (S_{\dot\theta} \boldsymbol{\theta}(k) - \underrightarrow{\dot{\boldsymbol{\theta}}}_k^{ref}) + \gamma_\theta U_\theta^\top (S_\theta \boldsymbol{\theta}(k) - H_{k+1}^c \theta_{f,i}^w) \\ -\gamma_\theta (H_{k+1}^f)^\top (S_\theta \boldsymbol{\theta}(k) - H_{k+1}^c \theta_{f,i}^w) \end{bmatrix}
\tag{33}
$$

ACKNOWLEDGMENT

This work was supported in part by the EU project COGIMON H2020 ICT. It was performed during a six month internship at the Learning Algorithms and System laboratory (LASA) of EPFL. Thanks go to all LASA's members for their excellent support.

REFERENCES

[1] 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," *2003 IEEE International Conference on Robotics and Automation*, pp. 1620–1626, 2003.

[2] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture Point: A Step toward Humanoid Push Recovery," *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pp. 200–207, Dec. 2006.

[3] P.-b. Wieber, "Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations," *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pp. 137–142, Dec. 2006.

[4] S. Kajita, M. Morisawa, K. Harada, K. Kaneko, F. Kanehiro, K. Fujiwara, and H. Hirukawa, "Biped Walking Pattern Generator allowing Auxiliary ZMP Control," *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, pp. 2993–2999, Oct. 2006.

[5] 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," *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1121–1126, Sep. 2008.

[6] 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," no. April, pp. 10–14, 2007.

[7] B. J. Stephens and C. G. Atkeson, "Push recovery by stepping for humanoid robots with force controlled joints," in *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*. IEEE, 2010, pp. 52–59.

[8] J. Urata, K. Nshiwaki, Y. Nakanishi, K. Okada, S. Kagami, and M. Inaba, "Online decision of foot placement using singular lq preview regulation," in *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*. IEEE, 2011, pp. 13–18.

[9] A. L. Hof, "The extrapolated center of mass concept suggests a simple control of balance in walking," *Human movement science*, vol. 27, no. 1, pp. 112–125, 2008.

[10] 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.

[11] J. Pratt, T. Koolen, T. De Boer, J. Rebula, S. Cotton, J. Carff, M. Johnson, and P. Neuhaus, "Capturability-based analysis and control of legged locomotion, part 2: Application to m2v2, a lower body humanoid," *The International Journal of Robotics Research*, p. 0278364912452762, 2012.

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

[13] J. Englsberger and C. Ott, "Integration of vertical com motion and angular momentum in an extended capture point tracking controller for bipedal walking," in *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*. IEEE, 2012, pp. 183–189.

[14] M. Krause, J. Englsberger, P.-B. Wieber, and C. Ott, "Stabilization of the capture point dynamics for bipedal walking based on model predictive control," in *Robot Control*, vol. 10, no. 1, 2012, pp. 165–171.

[15] R. J. Griffin and A. Leonessa, "Model predictive control for dynamic footstep adjustment using the divergent component of motion," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1763–1768.

[16] M. Shafiee-Ashtiani, A. Yousefi-Koma, M. Shariat-Panahi, and M. Khadiv, "Push recovery of a humanoid robot based on model predictive control and capture point," in *Robotics and Mechatronics (ICROM), 2016 4th International Conference on*. IEEE, 2016, pp. 433–438.

[17] M. Shafiee-Ashtiani, A. Yousefi-Koma, and M. Shariat-Panahi, "Robust bipedal locomotion control based on model predictive control and divergent component of motion," in *Robotics and Automation (ICRA), 2017 International Conference on*. IEEE, 2017, pp. 3505–3510.

[18] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online walking motion generation with automatic footstep placement," *Advanced Robotics*, vol. 24, no. 5-6, pp. 719–737, 2010.

[19] M. Bombile, "Visual servoing based positioning and object tracking on humanoid robot," in *New Trends in Networking, Computing, E-learning, Systems Sciences, and Engineering*. Springer, 2015, pp. 19–27.

[20] A. Herdt, N. Perrin, and P.-B. Wieber, "Walking without thinking about it," *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 190–195, Oct. 2010.

[21] M. Naveau, M. Kudruss, O. Stasse, C. Kirches, K. Mombaur, and P. Souères, "A reactive walking pattern generator based on nonlinear model predictive control," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 10–17, 2017.

[22] S. Kajita, F. Kanehiro, K. Kando, K. Yokoi, and H. Hirukawa, "The 3D Linear Inverted Pendulum Mode: A simple modeling for a biped walking pattern generation," pp. 239–246, 2001.

[23] J. Englsberger and C. Ott, "Walking stabilization for humanoid robots based on control of the capture point," *AT-AUTOMATISIERUNGSTECHNIK*, vol. 60, no. 11, pp. 692–703, 2012.

[24] M. Bombile, "Visual servo control on a humanoid robot," Master's thesis, University of Cape Town, 2015.

[25] H. Ferreau, C. Kirches, A. Potschka, H. Bock, and M. Diehl, "qpOASES: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.

[26] M. Bombile. Video attachement to paper: Capture-point based reactive omnidirectional walking controller. [Online]. Available: https://youtu.be/evk8U6MEQ2o