A Receding Horizon Push Recovery Strategy for Balancing the iCub Humanoid Robot

Stefano Dafarra, Francesco Romano and Francesco Nori

iCub Facility Department, Istituto Italiano di Tecnologia, 16163 Genova, Italy,
name.surname@iit.it

Abstract. Balancing and reacting to strong and unexpected pushes is a critical requirement for humanoid robots. We recently designed a capture point based approach which interfaces with a momentum-based torque controller and we implemented and validated it on the iCub humanoid robot. In this work we implement a Receding Horizon control, also known as Model Predictive Control, to add the possibility to predict the future evolution of the robot, especially the constraints switching given by the hybrid nature of the system. We prove that the proposed MPC extension makes the step-recovery controller more robust and reliable when executing the recovery strategy. Experiments in simulation show the results of the proposed approach.

Keywords: receding horizon, push recovery, iCub humanoid robot

This work was supported by the FP7 EU project CoDyCo (No. 600716 ICT 2011.2.1 Cognitive Systems and Robotics) and Horizon 2020 EU project An.Dy. (No. 731540 Research and Innovation Programme)

1 Introduction

A humanoid robot is generally built with the final intent of having a machine which can help humans performing boring or dangerous tasks. The adoption of a two legged design ideally allows the robot to share the same environment of humans. On the other hand, embedding relatively simple motor skills on such complicated machines is not a straightforward task. A recent trend consists in facing this problem through the adoption of optimal control techniques, like Model Predictive Controllers (MPC). The peculiarity of MPC resides in the possibility of inserting a feedback from the plant into the optimization procedure, through the "Receding Horizon Principle" [1].

Changes in the constraints set, e.g. when walking, result in a different evolution of the constrained dynamical system, making the overall system hybrid [2]. Model Predictive Controllers are appealing for controlling hybrid systems [3]. In fact, while predicting future states, MPC allow to cope with both time- and state-dependent switching, eventually undertaking actions to prepare for impending dynamics variations. Nevertheless, MPC is affected by the same problematics related to hybrid systems integration, which are indeed open research problems.

In literature the complexity of the robot dynamics is usually boiled down to simplified models like the Linear Inverted Pendulum [4], in order to devise simple strategies. In this context, in [5] a model predictive controller has been employed to stabilize walking patterns. It is also worth mentioning the Capture Point framework [6] which allows to elaborate conditions about the possibility of the robot to maintain the upright position. In [7] it has also been applied in conjunction with MPC.

In our approach we decided to take a different direction, taking into account the dynamic evolution of the center of mass (CoM) rather then adopting simple models. In addition, with respect to [8, 9, 10] we define a reactive planner which directly interfaces with the momentum-based whole body torque controller already implemented on iCub [11, 12]. The hybrid nature of the system is taken into account by means of time-varying constraints.

2 Background

Throughout this paper we adopt the following notation.

- $\mathcal{I}$ represents an inertial reference frame with the origin placed on the ground, with the $z-$axis pointing against the gravity and the $x-$axis oriented frontally with respect to the robot.
- $x_{\mathrm{CoM}} \in \mathbb{R}^3$ is the position of the center of mass with respect to $\mathcal{I}$.
- $\|x\|^2_W = x^\top W x$ is the weighted square norm of x.
- 1_n represents a $n \times n$ identity matrix. $0_{n\times m} \in \mathbb{R}^{n\times m}$ is a zero matrix while $0_n = 0_{n\times 1}$ is a zero column vector of size n.
- $x^\wedge \in \mathbb{R}^{3\times 3}$ denotes the skew-symmetric matrix such that $x^\wedge y = x \times y$, where $\times$ denotes the cross product operator in $\mathbb{R}^3$.

2.1 Momentum-based whole-body torque control

The momentum-based balancing controller has a hierarchical structure accomplishing two different control tasks. The most important task is in charge of controlling the robot linear and angular momentum.

We denote with $H = [H_{\mathrm{lin}}^\top, H_{\mathrm{ang}}^\top]^\top \in \mathbb{R}^6$ the robot linear and angular momentum (computed around the CoM). Its rate of change is:

$$\dot{H} = \sum_{i=1}^{n_c} {}^{\mathrm{CoM}}X_i f_i + m\bar{g} \tag{1}$$

where $f_i \in \mathbb{R}^6 = [\mathbb{f}_i^\top, \tau_i \top]^\top$ is the i-th of the n_c contact wrenches. They are measured in a frame fixed with the contact surface. Matrices ${}^{\mathrm{CoM}}X_i \in \mathbb{R}^{6\times 6}$ transform these wrenches in a frame oriented as the inertial frame $\mathcal{I}$, with the origin located on the center of mass position. Finally the robot total mass is indicated with m while $\bar{g} \in \mathbb{R}^6$ is the 6D gravity acceleration vector. By controlling the robot momentum through joint torques, the first task enables us to consider $f = [f_1^\top, \cdots, f_{n_c}^\top]^\top$ as virtual control inputs.

The second objective is responsible of tracking joint references, without interfering with the momentum control. In addition, it guarantees the stability of the zero dynamics resulting from the convergence of the first task. For additional details see the results in [11].

2.2 Optimal control problem

A continuous time optimal control problem can be stated as follows:

$$\underset{x(t),u(t)}{\text{minimize}}\ \mathit{\Gamma} = \int_0^{t_f} L(x(t),u(t))\mathrm{d}t + E(x(t_f)) \tag{2a}$$

$$\text{subject to } \dot{x}(t) = \mathrm{f}(x(t),u(t)), \quad x(0) = x_0 \qquad t \in \left[0, t_f\right] \tag{2b}$$

$$g(x(t),u(t)) \leq 0, \qquad t \in \left[0, t_f\right] \tag{2c}$$

where t_f is the "control horizon", $u \in \mathbb{R}^m$ is the control variable, $x \in \mathbb{R}^n$ is the state variable (whose initial value is x_0) and $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ the corresponding state evolution. $L(x(t),u(t))$ is the integral cost, while $E(x(t_f))$ weights x at the end of the horizon. Finally $g : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^{n_g}$ is the constraints function, embracing both equalities and inequalities.

3 Problem Formulation

3.1 The model

In this work we focus on the dynamics of the center of mass. If we consider our case of interest, where no contacts are available except the two applied at the feet, we can rewrite Eq. 1 as:

$$\begin{bmatrix} \ddot{x}_{\mathrm{CoM}} \\ \dot{H}_{\mathrm{ang}} \end{bmatrix} = \begin{bmatrix} m^{-1}1_3 & 0_{3\times3} \\ 0_{3\times3} & 1_3 \end{bmatrix} \begin{bmatrix} {}^{\mathrm{CoM}}X_l & {}^{\mathrm{CoM}}X_r \end{bmatrix} \begin{bmatrix} f_l \\ f_r \end{bmatrix} + \bar{g} \tag{3}$$

where the indexes l and r are relative to the left and right foot respectively, and we used the fact that $H_{\mathrm{lin}} = m\dot{x}_{\mathrm{CoM}}$.

In order to model the step in a model predictive framework, we can assume to know the instant in which the swing foot will touch the ground. In fact, the considered model does not contain any information about the posture of the robot and therefore it is not possible to define a "transition function", e.g. the distance from the foot to the ground is generally used to predict when the step is going to take place. In other words, the controller is aware that the impact will take place in a precise instant in the future, but it does not know whether the quantities involved in the model will affect this instant or not. The most viable choice is then to consider this instant as constant and known in advance, equal to t_{impact}, i.e. to the time needed to perform a step.

For the same reason, the position that the swing foot will have after the step is also assumed to be known and constant. This takes particular importance since ${}^{\mathrm{CoM}}X_s$ (where s refers to the swing foot) directly depends on this position. Summarizing, the peculiar characteristics of the step, i.e. the duration and the target position, are assumed to be known and constant.

3.2 The angular momentum

We focus now on the matrix ${}^{\mathrm{CoM}}X_i$ (here i is a placeholder for subscript l or r), whose structure depends on the choice of the frame to describe the robot momentum. The matrix has the following form:

$$
{}^{\mathrm{CoM}}X_i = \begin{bmatrix} 1_3 & 0_{3\times 3} \\ (x_i - x_{\mathrm{CoM}})^{\wedge} & 1_3 \end{bmatrix}. \tag{4}
$$

The derivative of the angular momentum $\dot{H}_{\mathrm{ang}}$ may thus be written as $\dot{H}_{\mathrm{ang}} = \sum_i^{l,r} (x_i - x_{\mathrm{CoM}})^{\wedge}\, \mathbb{f}_i + \tau_i$. The product between x_{CoM} and $\mathbb{f}_i$ causes the whole formulation to be non-convex which makes the resolution dramatically more complex. In literature this problem is faced by minimizing an upper-bound of the angular momentum [10], or by approximating it through quadratic constraints [13]. In our application, angular momentum is mainly needed to bound the usage of contact wrenches, rather than be precisely controlled to zero. Thus, we can accept a more coarse approximation, relying on its Taylor expansion (around the last available values of $\mathbb{f}_i$ and x_{CoM}) truncated to the first order, namely:

$$
\begin{aligned}
\dot{H}_{\mathrm{ang}} &\approx \sum_i^{\{l,r\}} \dot{H}_{\mathrm{ang}} + \frac{\partial \dot{H}_{\mathrm{ang}}}{\partial \mathbb{f}_i}\left(\mathbb{f}_i - \mathbb{f}_i^0\right) + \frac{\partial \dot{H}_{\mathrm{ang}}}{\partial x_{\mathrm{CoM}}}\left(x_{\mathrm{CoM}} - x_{\mathrm{CoM}}^0\right) \\
\dot{H}_{\mathrm{ang}} &\approx \sum_i^{\{l,r\}} \tau_i + \left(x_i - x_{\mathrm{CoM}}^0\right)^{\wedge} \mathbb{f}_i^0 + \left(x_i - x_{\mathrm{CoM}}^0\right)^{\wedge} \left(\mathbb{f}_i - \mathbb{f}_i^0\right) + \\
&\quad + \left(\mathbb{f}_i^0\right)^{\wedge} \left(x_{\mathrm{CoM}} - x_{\mathrm{CoM}}^0\right). \\
\Rightarrow \dot{H}_{\mathrm{ang}} &\approx \sum_i^{\{l,r\}} \tau_i + \left(x_i - x_{\mathrm{CoM}}^0\right)^{\wedge} \mathbb{f}_i + \left(\mathbb{f}_i^0\right)^{\wedge} \left(x_{\mathrm{CoM}} - x_{\mathrm{CoM}}^0\right) \qquad (5)
\end{aligned}
$$

where we exploited the anticommutative property of the cross product, i.e. $A^{\wedge}B = -B^{\wedge}A$. The superscript ^{0} refers to the point at which the Taylor expansion is computed, in our case from the last feedback obtained from the robot. The approximation introduced with the truncation to the first order is $o\Big(\left(\mathbb{f}_i - \mathbb{f}_i^0\right)\left(x_{\mathrm{CoM}} - x_{\mathrm{CoM}}^0\right)\Big)$. If we consider a controller horizon t_f not too wide, the term related to the position of the CoM is expected to be small enough, making this linear approximation effective.

Finally, by defining the state variable γ and the control input variable f as $\gamma := \left[\, x_{\mathrm{CoM}}^{\top}\ \dot{x}_{\mathrm{CoM}}^{\top}\ H_{\mathrm{ang}}^{\top} \,\right]^{\top},\ f := \left[\, f_l^{\top}\ f_r^{\top} \,\right]^{\top}$ we can rewrite Eq. (3) as:

$$
\dot{\gamma} = \tilde{E}v_{\gamma}\gamma + \tilde{F}_{\gamma} f + \tilde{G}_{\gamma} + \tilde{S}_{\gamma}^0, \tag{6}
$$

$$
\tilde{E}v_{\gamma} = \begin{bmatrix} 0_{3\times 3} & 1_3 & 0_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} \\ \left(\mathbb{f}_l^0 + \mathbb{f}_r^0\right)^{\wedge} & 0_{3\times 3} & 0_{3\times 3} \end{bmatrix}, \tilde{F}_{\gamma} = \begin{bmatrix} 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} \\ m^{-1}1_3 & 0_{3\times 3} & m^{-1}1_3 & 0_{3\times 3} \\ \left(x_l - x_{\mathrm{CoM}}^0\right)^{\wedge} & 1_3 & \left(x_r - x_{\mathrm{CoM}}^0\right)^{\wedge} & 1_3 \end{bmatrix},
$$

$$
\tilde{G}_{\gamma} = \begin{bmatrix} 0_3 \\ -\bar{g} \end{bmatrix}, \quad \tilde{S}_{\gamma}^0 = \begin{bmatrix} 0_6 \\ -\left(\mathbb{f}_l^0 + \mathbb{f}_r^0\right)^{\wedge} x_{\mathrm{CoM}}^0 \end{bmatrix}.
$$

3.3 Constraints definition

The constraints to be enforced are mainly related to the feasibility of the exerted wrenches, e.g. center of pressure and friction cone among the most common constraints related to unilateral contacts. Furthermore the wrench acting on the swinging foot should be null for every instant before the impact, i.e. t_{impact}.

We start by examining the constraints acting on the stance foot (here assumed to be the left):

$$A_{cl} f_l \leq b_{cl} \quad \forall t : t \leq t_f. \tag{7}$$

Eq. (7) encompasses all the considered inequalities constraints, namely: i) friction cone with linear approximation, ii) center of pressure, iii) positivity of the normal component of the contact force.

The constraints on the swing foot instead, catches the hybrid nature of the system. In particular they represent the fact that the wrench must be null before the impact and feasible after it. Formally:

$$\begin{cases} f_r = 0_6 & \forall t : \ t < t_{impact} \\ A_{cr} f_r \leq b_{cr} & \forall t : \ t_{impact} \leq t \leq t_f \end{cases} \tag{8}$$

where A_{cr} and b_{cr} describe the same feasibility constraints as A_{cl} and b_{cl}. The above equation assumes that $t_{impact} \leq t_f$. If the impact does not occur inside the control horizon, then the wrench exerted by the right foot is forced to be null throughout the whole horizon.

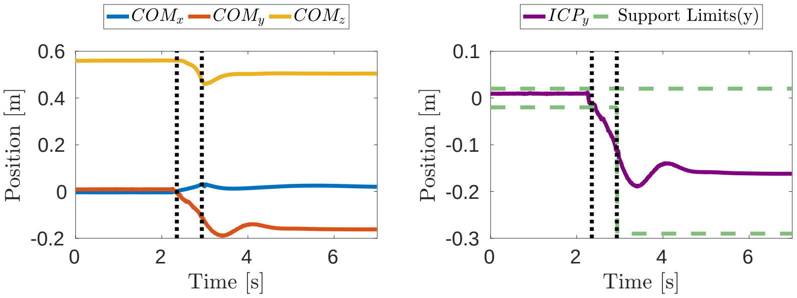
We also included an additional constraint to enforce that balancing is kept after the step. In particular, after this instant, we can constraint the instantaneous capture point to be inside the convex hull of the two feet, which can be predicted by knowing the future position of the right foot. Thus, we can define a set of linear inequalities such that if $A_{ch} x_{icp} \leq b_{ch} \quad \forall t : t \geq t_{impact}$ is satisfied, than x_{icp}, i.e. the instantaneous capture point location, is in the convex hull. By imposing this constraint, we are forcing the instantaneous capture point to be in a *capturable* state after the step is performed. In fact, the convex hull represents the region in which it is possible to stabilize the instantaneous capture point dynamics without taking additional steps.

3.4 Cost function definition

Following the optimal control formulation of Eq. (2) we are left to define the cost function applied within the MPC controller, called $\varGamma$. Note that different terms of the cost function act only after the step is taken. Formally, it has the following expression:

$$\varGamma = \frac{1}{2}\left(\int_0^{t_f} \left\| \gamma(\tau) - \gamma^d(\tau) \right\|^2_{K_\gamma} \mathrm{d}\tau + \int_{\bar{t}_{imp}}^{t_f} \left\| \gamma(\tau) - \gamma^d(\tau) \right\|^2_{K_\gamma^{imp}} \mathrm{d}\tau + \right. \tag{9}$$

$$\left. + \int_0^{t_f} \left\| f(\tau) \right\|^2_{K_f} \mathrm{d}\tau + \int_{\bar{t}_{imp}}^{t_f} \left\| x_{icp}(\tau) - x^d_{icp}(\tau) \right\|^2_{K_{icp}} \mathrm{d}\tau + \left\| \gamma(t_f) - \gamma^d(t_f) \right\|^2_{K_\gamma^{imp}} \right)$$



(a) (b)

Fig. 1: (a) represents the center of mass evolution while performing the step. The first dotted line denotes the time of the push which occurred at $t = 2.4$s. The second one at $t = 3.0$s indicates when the right foot hits the ground. (b) presents the evolution of the y-component of the instantaneous capture point. The horizontal green dotted lines represent the approximation of the convex hull.

where the superscript d indicates the reference values. $K_{(\cdot)}^{(\cdot)}$ is a real positive semi-definite matrix of gains with suitable dimensions, accounting also for unit of measurement mismatches inside Γ. $\bar{t}_{imp}$ is the minimum between t_{impact} and t_f while, for the sake of simplicity, the initial time instant is set to zero. Thus, it is possible to vary the cost applied to the state γ after the impact. This is done through the use of the matrix K_γ^{imp} which, after the impact, adds up to K_γ. Finally, a terminal cost term is inserted using the same matrix K_γ^{imp}, to model the finiteness of the control horizon.

Consider now the final objective of having the center of mass over the centroid of the support polygon once the step is made. We decide to weight only the $z-$component of the CoM error throughout the whole horizon, while weighting the traverse components (i.e. x and y) only in the terminal cost (last term of Eq.(9)) and after the step is made (second part of Eq.(9)). The same rationale could be applied to the reference for the instantaneous capture point. During the step, its dynamics is exponentially diverging and, as a consequence, this contribution takes a role only after t_{impact}, where its distance from the centroid of the convex hull is weighted.

Finally, the requested reaction forces are weighted too, mostly for avoiding impulsive responses which are dangerous for the mechanical structure of the robot, without any warranty that they will be actually attained through the underneath controller. Intuitively, the tracking error for the desired wrenches will be lower with a smooth and limited reference.

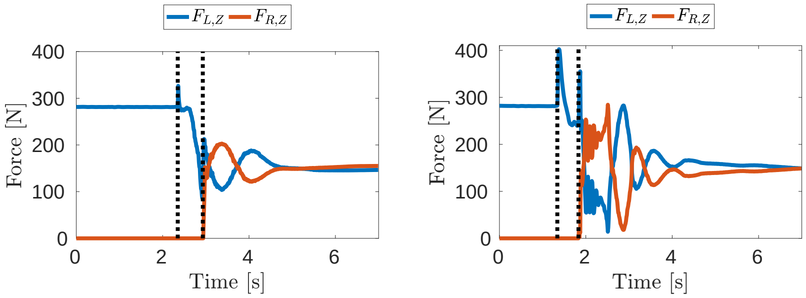


Fig. 2: Representation of the z-component of the demanded forces.

4 Simulation Results

The proposed MPC approach has been tested on an iCub model in the Gazebo simulator [14], employing YARP Plugins [15].

The iCub humanoid robot [16] possesses 53 actuated joints, but nearly half of them are concentrated on the eyes and the hands. For this reason only 23 degrees of freedom (DoF) are torque controlled and used for balancing purposes.

The proposed controller has been implemented in the MATLAB®/Simulink® environment, using the WBToolbox library[17]. The experimental scenario consists of the robot balancing on its left foot. We have chosen this simple scenario to test the performance of the presented strategy with a single contact activation.

When a strong push occurs the step recovery strategy starts. The time step $\mathrm{d}t$ is set to 10ms, while the controller horizon N is chosen to be 15. It has been observed that for smaller control horizons, the robot is not able to sustain the disturbance. Figure 1 depicts the result of the push experiment performed adopting the MPC approach. The robot was able to fully recover after the push, undergoing just a slight oscillation after the impact of the swing foot, whose occurrence is highlighted by the second dashed line. The capability of recovering by performing the step can be noticed also from Figure 1, where the behavior of the y-component of the Instantaneous Capture Point is depicted (the x component has undergone just a slight perturbation). In particular the Capture Point, after exiting the support polygon (corresponding to the left foot), is fully contained in the new convex hull once the step is performed.

Finally, it is worth considering the desired wrenches. Figures 2 compare the desired vertical force obtained using the presented control strategy and the one proposed in [18] (on the right). While oscillating in the first stage of the double support, the steady state is reached much faster than the previous approach.

5 Conclusions and Future Work

The presented approach adopts, as model, the dynamics of the center of mass with only a small approximation necessary to remove non-convexities. It also considers the discontinuous contribution of the swing foot through the adoption of time varying constraints. With respect to [18], this approach does not need the definition of a trajectory for the CoM, which indeed is part of the optimization process. As a consequence, the robustness of this technique relies more in the intrinsic knowledge of the system rather than in the expertise of the designer.

This approach has been meant not to be a simple planner, but instead to provide reaction to external disturbances in real-time. However, at the present time it takes almost 0.1 seconds to get a solution on a machine running Ubuntu 14.04 on a quad-core Intel® Core i5@2.30GHz with 16GB of RAM, by using as solver the MATLAB® interface of MOSEK®. As a future work, this strategy will be implemented on an optimized C++ code and applied on the real platform.

Finally, as an additional improvement we plan to supply the controller with more informations about the kinematics of the leg, thereby modeling the step movement directly as part of the problem formulation and relaxing the hypothesis of knowing a priori the time of the impact.

References

[1] Mayne D, Michalska H (1990) Receding horizon control of nonlinear systems. IEEE Transactions on Automatic Control 35(7):814–824

[2] Lygeros J, Tomlin C, Sastry S (1999) Hybrid systems: modeling, analysis and control. preprint

[3] Lazar M, Heemels W, Weiland S, Bemporad A (2006) Stabilizing model predictive control of hybrid systems. IEEE Transactions on Automatic Control

[4] Kajita S, Kanehiro F, Kaneko K, Yokoi K, Hirukawa H (2001) The 3D linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. In: Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on, vol 1, pp 239–246 vol.1

[5] Wieber PB (2006) Trajectory free linear model predictive control for stable walking in the presence of strong perturbations. In: 2006 6th IEEE-RAS International Conference on Humanoid Robots, IEEE, pp 137–142

[6] Pratt J, Carff J, Drakunov S, Goswami A (2006) Capture point: A step toward humanoid push recovery. In: Humanoid Robots, 2006 6th IEEE-RAS International Conference on, pp 200–207

[7] Krause M, Englsberger J, Wieber PB, Ott C (2012) Stabilization of the capture point dynamics for bipedal walking based on model predictive control. IFAC Proceedings Volumes 45(22):165–171

[8] Dai H, Valenzuela A, Tedrake R (2014) Whole-body motion planning with centroidal dynamics and full kinematics. In: 2014 IEEE-RAS International Conference on Humanoid Robots, IEEE, pp 295–302

[9] Herzog A, Rotella N, Schaal S, Righetti L (2015) Trajectory generation for multi-contact momentum control. In: Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on, IEEE, pp 874–880

[10] Dai H, Tedrake R (2016) Planning robust walking motion on uneven terrain via convex optimization. In: Humanoid Robots (Humanoids), 2016 IEEE-RAS International Conference on

[11] Nava G, Romano F, Nori F, Pucci D (2016) Stability analysis and design of momentum-based controllers for humanoid robots. Intelligent Robots and Systems (IROS) 2016 IEEE International Conference on

[12] Nori F, Traversaro S, Eljaik J, Romano F, Del Prete A, Pucci D (2015) iCub whole-body control through force regulation on rigid noncoplanar contacts. Frontiers in Robotics and AI 2(6)

[13] Ponton B, Herzog A, Schaal S, Righetti L (2016) A convex model of momentum dynamics for multi-contact motion generation. In: Humanoid Robots (Humanoids), 2016 IEEE-RAS International Conference on

[14] Koenig N, Howard A (2004) Design and use paradigms for gazebo, an open-source multi-robot simulator. Intelligent Robots and Systems, 2004 (IROS 2004) Proceedings 2004 IEEE/RSJ International Conference on

[15] Mingo E, Traversaro S, Rocchi A, Ferrati M, Settimi A, Romano F, Natale L, Bicchi A, Nori F, Tsagarakis NG (2014) Yarp based plugins for gazebo simulator. In: 2014 Modelling and Simulation for Autonomous Systems Workshop (MESAS), Roma, Italy, 5 -6 May 2014

[16] Metta G, Sandini G, Vernon D, Caldwell D, Tsagarakis N, Beira R, Santos-Victor J, Ijspeert A, Righetti L, Cappiello G, et al (2005) The RobotCub project-an open framework for research in embodied cognition. Humanoids Workshop, IEEE–RAS International Conference on Humanoid Robots

[17] Romano F, Traversaro S, Pucci D, Prete AD, Eljaik J, Nori F (2017) A whole-body software abstraction layer for control design of free-floating mechanical systems. In: 2017 IEEE 1st International Conference on Robotic Computing

[18] Dafarra S, Romano F, Nori F (2016) Torque-controlled stepping-strategy push recovery: Design and implementation on the iCub Humanoid robot. In: Humanoid Robots (Humanoids), 2016 IEEE-RAS International Conference on