



**FP7-600716**

**Whole-Body Compliant Dynamical Contacts in Cognitive Humanoids**

**D5.1**

**Validation scenario 1: balancing on multiple rigid contacts**

<b>Editor(s)</b>	Francesco Nori
<b>Responsible Partner</b>	IIT
<b>Affiliations</b>	<sup>1</sup> IIT
<b>Status-Version:</b>	Draft-1.0
<b>Date:</b>	Feb. 28, 2014
<b>EC Distribution:</b>	Consortium
<b>Project Number:</b>	600716
<b>Project Title:</b>	Whole-Body Compliant Dynamical Contacts in Cognitive Humanoids

<b>Title of Deliverable:</b>	Validation scenario 1: balancing on multiple rigid contacts
<b>Date of delivery to the EC:</b>	28/2/2014

<b>Workpackage responsible for the Deliverable</b>	deliv WP5
<b>Editor(s):</b>	Francesco Nori
<b>Contributor(s):</b>	Francesco Nori, Andrea Del Prete, Daniele Pucci, Serena Ivaldi, Vincent Padois
<b>Reviewer(s):</b>	
<b>Approved by:</b>	All Partners
<b>Abstract</b>	<p>In the present deliverable we discuss the technical details and choices for the implementation of on the year-1 validation scenario of the CoDyCo project. This validation scenario consists on balancing with multiple rigid contact points. Contact and trajectory planning is not part of the scenario. The final validation consists in controlling the iCub across the following sequence of control tasks. Multiple (force and position) tasks will be handled with strict priorities by adopting the theoretical framework known as stack of tasks [Del Prete, 2013]. First the robot is standing on two feet and, in order to maintain this posture, a controller takes care of controlling the left and right foot position and forces exchanged with the ground. Second, the robot start moving both hands towards a table with the intend of establishing two additional support contacts: in this phase the robot starts controlling also the hands Cartesian position. As soon as the artificial skin detects the contact between the hands and the table the controllers starts including hands force regulation among the control tasks.</p>
<b>Keyword List:</b>	Multiple, rigid, contacts, control, stability, tracking, forces, torques.

**Document Revision History**

<b>Version</b>	<b>Date</b>	<b>Description</b>	<b>Author</b>
First draft	19 Feb 2014	In this version we simply write down a few considerations on the first year validation scenario as discussed after the mid-year CoDyCo meeting in Paris.	Francesco Nori

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Executive Summary</b>	<b>4</b>
<b>3</b>	<b>First Year Scenario Validation</b>	<b>4</b>
<b>4</b>	<b>First year scenario implementation</b>	<b>5</b>
4.1	Task Space Inverse Dynamics (TSID) . . . . .	5
4.2	Set of admissible tasks . . . . .	5
4.3	Sequencing of tasks . . . . .	7
4.4	Task references . . . . .	8
4.5	Task trajectories . . . . .	9
4.6	Task prioritization . . . . .	12
4.7	Alternative controller formulation . . . . .	13
<b>A</b>	<b>Center of pressure: <i>CoP</i></b>	<b>16</b>
A.1	<i>CoP</i> definition . . . . .	16
A.2	<i>CoP</i> and dynamic equilibrium . . . . .	17

# 1 Introduction

The first year CoDyCo validation scenario deals with iCub balancing on multiple rigid contacts. The theoretical framework for dealing with these kind of situations has been extensively explored in a number of papers [Aghili, 2005, Righetti et al., 2013]. Taking advantage of our experience on the topic [Del Prete, 2013] the first year validation scenario will be implemented in the Task Space Inverse Dynamics (TSID) framework and in particular we will consider its application to floating base kinematic chains. The iCub will be torque controlled and the controller will assume that desired torques are exactly executed by a lower level torque control. Dynamics will be computed with a custom library, iDynTree<sup>1</sup>, built on top of KDL<sup>2</sup>. Desired joint torques will be computed by the TSID algorithm given a set of tasks and their relative priorities. The definition of tasks and priorities is described in details in the following sections.

## 2 Executive Summary

The deliverable is organized as follows. Section 3 gives an high level presentation of the validation scenario to be presented at the first year review meeting. Section 4.1 discusses the numerical technique (TSID) used to implement the validation scenario as a prioritization of concurrent tasks. Section 4.2 discusses the set of control tasks that will be implemented in order to perform the validation scenario. Their sequencing in the form of a finite state machine is discussed in Section 4.3 and issues related to task switching discussed in Section 4.4. Priorities between tasks are summarized in Section 4.6.

## 3 First Year Scenario Validation

The first year scenario aims at validating on the iCub<sup>3</sup> the theoretical results of the consortium in performing the task of balancing on multiple rigid contacts. As clearly stated in the CoDyCo proposal the validation should not involve any planning and therefore the sequence of movements and trajectories will be predefined and kept constant across trials. During the mid-year meeting held in Paris, the sequence of movements was discussed and an agreement was found across the entire consortium. The scenario will begin with the iCub standing on two feet in front of an object that offers the possibility of additional contacts (e.g. a table where to place both hands). After some time the robot will start a movement of both hands towards the additional contact. As soon as the hands detect the additional contacts with the artificial skin, the iCub will start regulating the interaction forces at both hands trying to maintain a comfortable and stable posture.

---

<sup>1</sup>[http://wiki.icub.org/codyco/dox/html/group\\_\\_iDynTree.html](http://wiki.icub.org/codyco/dox/html/group__iDynTree.html)

<sup>2</sup><http://www.orocos.org/kdl>

<sup>3</sup>Implementation is foreseen on the iCub platforms currently available at IIT and UPMC.

## 4 First year scenario implementation

Recently we proposed a numerical technique [Del Prete, 2013] for solving the problem of controlling multiple concurrent tasks on floating base robots. Considered tasks include the control of contact forces and multiple motion tasks. Tasks are ordered according to a priority structure, with force tasks at the highest priority. The proposed solution, named task space inverse dynamics (TSID), is presented in the following section.

### 4.1 Task Space Inverse Dynamics (TSID)

Let's first recall how the force control problem is solved in the TSID framework in the context of floating base manipulators [Del Prete, 2013]. The framework computes the joint torques to match as close as possible a desired vector of forces at the contacts (1a) compatible with the system dynamics (1b) and with the contact constraints (1c):

$$\boldsymbol{\tau}^* = \arg \min_{\boldsymbol{\tau} \in \mathbb{R}^n} \|\mathbf{f} - \mathbf{f}^*\|^2 \quad (1a)$$

$$s.t. \quad M\ddot{\mathbf{q}} + \mathbf{h} - J_c^T \mathbf{f} = S^T \boldsymbol{\tau} \quad (1b)$$

$$J_c \ddot{\mathbf{q}} + \dot{J}_c \dot{\mathbf{q}} = 0 \quad (1c)$$

$$J_i \ddot{\mathbf{q}} + \dot{J}_i \dot{\mathbf{q}} = \ddot{\mathbf{x}}_i^*, \quad i = 1, \dots, N-1 \quad (1d)$$

where  $\mathbf{q}$  is a vector of generalized coordinates describing the robot pose (includes both  $\mathbf{q}_j$  and the floating base position),  $\boldsymbol{\tau}$  is the vector of joint torques,  $M$  is the mass matrix,  $\mathbf{h}$  contains both gravitational and Coriolis terms,  $S$  is the selection matrix which takes into account the fact that we do not have a direct control over the floating base variables,  $\mathbf{f}$  is the contact spatial vector,  $\mathbf{f}^*$  is its desired value and  $J_c$  is the contact Jacobian.

In the TSID formalism, the idea is to solve the force control task (represented as in (1a)) at the highest priority and to exploit the nullspace of this primary control task to perform  $N-1$  motion tasks at lower priorities (represented as in (1d)). These tasks (indexed with  $i = 1, \dots, N-1$ ) are all represented as the problem of tracking a given reference acceleration  $\ddot{\mathbf{x}}_i^*$  for a variable  $\mathbf{x}_i$  differentially linked to  $\mathbf{q}$  by the Jacobian  $J_i$ , i.e.  $\dot{\mathbf{x}}_i = J_i \dot{\mathbf{q}}$  and  $\ddot{\mathbf{x}}_i = J_i \ddot{\mathbf{q}} + \dot{J}_i \dot{\mathbf{q}}$ . Assuming that the force task has maximum priority the solution is the represented by the following expression:

$$\boldsymbol{\tau}^* = -(J_c \bar{S})^T \mathbf{f}^* + N_j^{-1} \ddot{\mathbf{q}}_1^* + \bar{S}^T \mathbf{h}, \quad (2)$$

where  $\bar{S} = M^{-1} S^T (S M^{-1} S^T)^{-1}$  and the term  $\ddot{\mathbf{q}}_1^*$  is computed solving the following recursion for  $i = N, \dots, 1$ :

$$\begin{aligned} \ddot{\mathbf{q}}_i &= \ddot{\mathbf{q}}_{i+1} + (J_i \bar{S} N_{p(i)})^\dagger + (\ddot{\mathbf{x}}_i^* - \dot{J}_i \dot{\mathbf{q}} + J_i (U^T M_b^{-1} (\mathbf{h}_b - J_{cb}^T \mathbf{f}) - \bar{S} \ddot{\mathbf{q}}_{i+1})) \\ N_{p(i)} &= N_{p(i+1)} - (J_{i+1} \bar{S} N_{p(i+1)})^\dagger J_{i+1} \bar{S} N_{p(i+1)}, \end{aligned} \quad (3)$$

which is initialized setting  $\ddot{\mathbf{q}}_{N+1} = 0$ ,  $N_{p(N)} = I$ ,  $J_N = J_c$  and  $\ddot{\mathbf{x}}_N = 0$ .

### 4.2 Set of admissible tasks

The CoDyCo first year scenario is implemented within the TSID framework which requires the definition of a suitable set of position ( $\ddot{\mathbf{x}}_i^*$ ) and force ( $\mathbf{f}^*$ ) tasks and their relative priority. The

set of admissible tasks is quite flexible also considering the flexibility of the underlying software libraries<sup>4</sup>. Nevertheless we list here a set of possible tasks, which we will use as a reference in the following sections. Quantities are defined as with a notation similar to the one used in [Featherstone, 2008]:  $\mathbf{v}$  indicates a spatial velocity (a single vector for representing linear and angular velocities),  $\mathbf{a}$  indicates a spatial acceleration (a single vector for representing linear and angular accelerations),  $\mathbf{f}$  indicates a spatial force (a single vector for forces and torques), the index  $i = 0, 1, \dots, N_B - 1$  is used to reference the  $N_B$  rigid bodies representing the iCub body chain (0 being defined as the pelvis rigid link), the index  $w$  is used to represent the world reference frame, the superscripts and subscripts  $^{lh}, ^{rh}, ^{lf}, ^{rf}$  indicate reference frames rigidly attached to the left hand, right hand, left foot and right foot respectively, the superscript  $^i$  indicates the reference frame attached to the  $i$ -th rigid body,  $^k \mathbf{X}_i$  represents the rigid roto-translation from the reference frame  $i$  to the reference frame  $k$ ,  $\mathbf{q}_j$  represent the position of the iCub joints.

Tasks will be thrown out of the following set of admissible tasks. For each task  $T_i$  we specify the reference values ( $\ddot{\mathbf{x}}_i$  or  $\mathbf{f}^*$ ) and associated Jacobians ( $J_i$ ).

- $T_X^{rh}$  : right hand linear and angular accelerations.

$$\begin{aligned} \ddot{\mathbf{x}}_i^* &: &^{rh} \ddot{\mathbf{x}}_w &= ^{rh} \ddot{\mathbf{x}}_w^*, & ^{rh} \dot{\boldsymbol{\omega}}_w &= ^{rh} \dot{\boldsymbol{\omega}}_w^*; \\ J_i &: &J_{rh}; \end{aligned}$$

- $T_X^{lh}$  : right hand linear and angular accelerations.

$$\begin{aligned} \ddot{\mathbf{x}}_i^* &: &^{lh} \ddot{\mathbf{x}}_w &= ^{lh} \ddot{\mathbf{x}}_w^*, & ^{lh} \dot{\boldsymbol{\omega}}_w &= ^{lh} \dot{\boldsymbol{\omega}}_w^*; \\ J_i &: &J_{lh}; \end{aligned}$$

- $T_X^{com}$  : center of mass acceleration. Move the center of mass with the following acceleration:

$$\begin{aligned} \ddot{\mathbf{x}}_i^* &: &\ddot{\mathbf{x}}_{com} &= \ddot{\mathbf{x}}_{com}^*; \\ J_i &: &J_{com}; \end{aligned}$$

- $T_f^{rf}$  : right foot force. Regulate the right foot interaction force to a predefined value:

$$\begin{aligned} \ddot{\mathbf{f}}_i^* &: &\mathbf{f}_{rf} &= \mathbf{f}_{rf}^*; \\ J_i &: &J_{rf}; \end{aligned}$$

- $T_f^{lf}$  : left foot force. Regulate the left foot interaction force to a predefined value:

$$\begin{aligned} \ddot{\mathbf{f}}_i^* &: &\mathbf{f}_{lf} &= \mathbf{f}_{lf}^*; \\ J_i &: &J_{lf}; \end{aligned}$$

<sup>4</sup>[http://wiki.icub.org/codyco/dox/html/group\\_\\_iDynTree.html](http://wiki.icub.org/codyco/dox/html/group__iDynTree.html)

- $T_f^{rh}$  : right hand force. Regulate the right hand interaction force to a predefined value:

$$\begin{aligned} \ddot{\mathbf{f}}_i^* &: & \mathbf{f}_{rh} &= \mathbf{f}_{rh}^*; \\ J_i &: & J_{rh}; \end{aligned}$$

- $T_f^{lh}$  : left hand force. Regulate the left hand interaction force to a predefined value:

$$\begin{aligned} \ddot{\mathbf{f}}_i^* &: & \mathbf{f}_{lh} &= \mathbf{f}_{lh}^*; \\ J_i &: & J_{lh}; \end{aligned}$$

- $T^q$  : postural task. Maintain the robot joints  $\mathbf{q}_j$  close to certain reference posture  $\mathbf{q}_j^*$ :

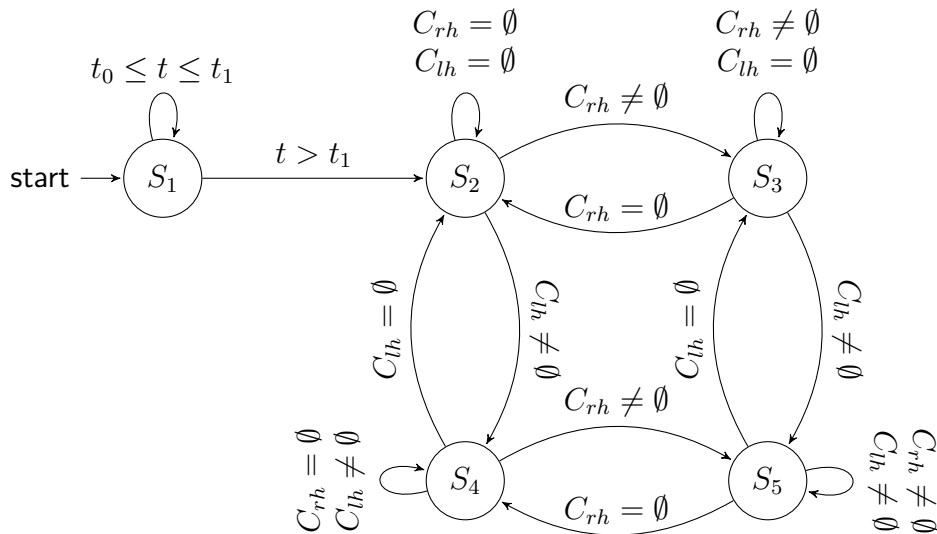
$$\begin{aligned} \ddot{\mathbf{x}}_q^* &: & \ddot{\mathbf{q}}_j &= \ddot{\mathbf{q}}_j^*; \\ J_i &: & I. \end{aligned}$$

### 4.3 Sequencing of tasks

The set of tasks active at a certain instant of time is regulated by a finite state machine. In particular there are five different states  $S_1, \dots, S_5$  each characterized by a different set of active tasks  $S_1, \dots, S_5$ .

- $S_1$  has the following set of active tasks  $S_1 = \{T_{\mathbf{X}}^{com}, T_{\mathbf{f}}^{rf}, T_{\mathbf{f}}^{lf}, T^q\}$ .
- $S_2$  has the following set of active tasks  $S_2 = S_1 \cup \{T_{\mathbf{X}}^{lh}, T_{\mathbf{X}}^{rh}\}$ .
- $S_3$  has the following set of active tasks  $S_3 = S_1 \cup \{T_{\mathbf{X}}^{lh}, T_{\mathbf{f}}^{rh}\}$ .
- $S_4$  has the following set of active tasks  $S_4 = S_1 \cup \{T_{\mathbf{f}}^{lh}, T_{\mathbf{X}}^{rh}\}$ .
- $S_5$  has the following set of active tasks  $S_5 = S_1 \cup \{T_{\mathbf{f}}^{lh}, T_{\mathbf{f}}^{rh}\}$ .

Transition between states is regulated by the following finite state machine where the sets  $C_{rh}$  and  $C_{lh}$  represent the number of taxels (tactile elements) activated at time  $t$ .



In practice, at start ( $t = t_0$ ) the robot is in state  $S_1$  and controls the position and orientation of the pelvis with respect to a world reference frame ( $T_X^{com}$ ). Balance is maintained by controlling also the forces and torques exchanged at the right ( $T_f^{rf}$ ) and left foot ( $T_f^{lf}$ ). A postural tasks ( $T^q$ ) guarantees that the system is not drifting and maintains a posture close to a reference posture. After a predefined amount ( $t = t_1$ ) the system switches to the state  $S_2$  by adding two tasks to the set of active tasks: a control of the position and orientation of the right and left hands ( $T_X^{rh}$  and  $T_X^{lh}$  respectively). Successive transitions are triggered by the tactile sensors. If no contact are detected on the right and left hand ( $C_{rh} = \emptyset$  and  $C_{lh} = \emptyset$  respectively) the system remains in  $S_2$ . A transition from  $S_2$  to  $S_3$  is performed when the system detects a contact on the right hand ( $C_{rh} \neq \emptyset$ ): in this state the active tasks are the same active in  $S_2$  but the right hand position and orientation control ( $T_X^{rh}$ ) is replaced with the force control ( $T_f^{rh}$ ). Similarly, a transition from  $S_2$  to  $S_4$  is performed when the system detects a contact on the left hand ( $C_{lh} \neq \emptyset$ ): in this state the active tasks are the same active in  $S_2$  but the left hand position and orientation control ( $T_X^{lh}$ ) is replaced with the force control ( $T_f^{lh}$ ). Finally a transition from either  $S_3$  or  $S_4$  to  $S_5$  is performed whenever the free moving hand perceives a contact so that in this new situation both hands are in contact ( $C_{rh} \neq \emptyset$  and  $C_{lh} \neq \emptyset$  respectively). In  $S_5$  both hands are used to control the interaction forces by activating the tasks  $T_f^{rh}$  and  $T_f^{lh}$ .

#### 4.4 Task references

In this section we discussed how to compute the task references:

$$\mathbf{f}_{rh}^*, \quad \mathbf{f}_{lh}^*, \quad \mathbf{f}_{rf}^*, \quad \mathbf{f}_{lf}^*, \quad {}^{rh}\ddot{\mathbf{x}}_w^*, \quad {}^{rh}\dot{\boldsymbol{\omega}}_w^*, \quad {}^{lh}\ddot{\mathbf{x}}_w^*, \quad {}^{lh}\dot{\boldsymbol{\omega}}_w^*, \quad \ddot{\mathbf{x}}_{com}^*.$$

The proposed definitions assume that for each reference we have an associated desired trajectory which is pre-specified:

$$\mathbf{f}_{rh}^d(t), \quad \mathbf{f}_{lh}^d(t), \quad \mathbf{f}_{rf}^d(t), \quad \mathbf{f}_{lf}^d(t), \quad {}^{rh}\mathbf{x}_w^d(t), \quad {}^{rh}\mathbf{R}_w^d(t), \quad {}^{lh}\mathbf{x}_w^d(t), \quad {}^{lh}\mathbf{R}_w^d(t), \quad \mathbf{x}_{com}^d(t).$$

The definition of these desired trajectories is discussed in Section 4.5. In this section we discuss how task references are constructed given the task trajectories.

Let's first discuss how the center of mass reference acceleration  $\ddot{\mathbf{x}}_{com}^*$  is generated at each instant of time. Given the desired trajectory  $\mathbf{x}_{com}^d(t)$  trajectory we define:

$$\ddot{\mathbf{x}}_{com}^*(t) = \ddot{\mathbf{x}}_{com}^d(t) + K_d^{com} (\dot{\mathbf{x}}_{com}^d(t) - \dot{\mathbf{x}}_{com}) + K_p^{com} (\mathbf{x}_{com}^d(t) - \mathbf{x}_{com}), \quad (4)$$

where  $K_p^{com}$  and  $K_d^{com}$  are arbitrary positive definite matrices. Similarly, the force reference trajectories  $\mathbf{f}^*(t)$  are obtained from a desired force trajectory  $\mathbf{f}^d(t)$ :

$$\mathbf{f}^*(t) = \mathbf{f}^d(t) + K_i^f \int_0^t (\mathbf{f}^d(\tau) - \mathbf{f}(\tau)) d\tau, \quad (5)$$

where again  $K_i^f$  is an arbitrary positive definite matrix. It is slightly more complicated to define the hands acceleration reference trajectories  $\ddot{\mathbf{x}}_w^*(t)$ ,  $\dot{\boldsymbol{\omega}}_w^*(t)$ . The linear acceleration trajectory  $\ddot{\mathbf{x}}_w^*(t)$  can be handled exactly as it was done for the center of mass position exploiting a suitable desired trajectory  $\mathbf{x}_w^d(t)$ . Angular acceleration trajectories  $\dot{\boldsymbol{\omega}}_w^*(t)$  are slightly more

complicated in considerations of the fact that they are derived from a reference rotation matrix  $R_w^d(t) \in SO(3)$ . Within this context, we follow the definition in [Siciliano et al., 2009] and define an orientation error  $e_O$  between the desired pose  $R_w^d(t) \in SO(3)$  and the current pose  $R_w \in SO(3)$  as:

$$e_O = \sin(\theta)r,$$

being  $\theta$  and  $r$  the axis-angle representation of the error rotation  $R_w^d(t)R_w^\top$ . We have:

$$\dot{e}_O = -L\omega + L^\top\omega_d,$$

with:

$$L = -\frac{1}{2}(S(n_d)S(n) + S(s_d)S(s) + S(a_d)S(a)),$$

where  $n, s, a$  (and correspondingly  $n_d, s_d, a_d$ ) are the columns of  $R_w$ , i.e.  $R_w = [n \ s \ a]$ . Similarly, the second derivative of the angular error is given by:

$$\ddot{e}_O = -\dot{L}\omega - L\dot{\omega} + \dot{L}^\top\omega_d + L^\top\dot{\omega}_d.$$

where the derivative of  $L$  can be expressed as follows (computations omitted):

$$\begin{aligned} \dot{L} = & -\frac{1}{2}(S(\omega \times n_d)S(n) + S(n_d)S(\omega \times n) + S(\omega \times s_d)S(s) + \\ & + S(s_d)S(\omega \times s) + S(\omega \times a_d)S(a) + S(a_d)S(\omega \times a)). \end{aligned}$$

The dynamics  $\ddot{e}_O + K_d^O \dot{e}_O + K_p^O e_O = 0$  can be rewritten as:

$$\dot{\omega} = L^{-1} \left( -\dot{L}\omega + \dot{L}^\top\omega_d + L^\top\dot{\omega}_d + K_d^O \dot{e}_O + K_p^O e_O \right),$$

which eventually gives us the following rule for choosing the right and left hand reference trajectories:

$$\begin{aligned} {}^{rh}\dot{\omega}_w^*(t) &= L^{-1} \left( -\dot{L}^{rh}\omega_w + \dot{L}^{\top rh}\omega_w^d + L^{\top rh}\dot{\omega}_w^d + K_d^O \dot{e}_O^{rh} + K_p^O e_O^{rh} \right), \\ {}^{lh}\dot{\omega}_w^*(t) &= L^{-1} \left( -\dot{L}^{lh}\omega_w + \dot{L}^{\top lh}\omega_w^d + L^{\top lh}\dot{\omega}_w^d + K_d^O \dot{e}_O^{lh} + K_p^O e_O^{lh} \right). \end{aligned}$$

## 4.5 Task trajectories

In this section we discuss how to compute the tasks reference trajectories:

$$f_{rh}^d(t), \quad f_{lh}^d(t), \quad f_{rf}^d(t), \quad f_{lf}^d(t), \quad {}^{rh}x_w^d(t), \quad {}^{rh}R_w^d(t), \quad {}^{lh}x_w^d(t), \quad {}^{lh}R_w^d(t), \quad x_{com}^d(t).$$

At this stage of the CoDyCo project, there is no interest in the planning movements but just in controlling them. Therefore, it is out of the scope of the current deliverable to find desired trajectories compatible with the system dynamics, joint limits, motor torque limits and so on. Hands ( $T_X^{rh}$  or  $T_X^{lh}$ ) and center of mass ( $T_X^{rh}$ ) trajectories are predefined and assumed to be chosen according to the specific environment chosen for the validation scenario. In particular, the center of mass trajectory  $x_{com}^d(t)$  is left as a free parameter to be chosen during the

validation. Hand trajectories  ${}^{rh}\mathbf{x}_w^d(t)$ ,  ${}^{rh}R_w^d(t)$ ,  ${}^{lh}\mathbf{x}_w^d(t)$ ,  ${}^{lh}R_w^d(t)$  are constructed by specifying only the final posture to be reached:  ${}^{rh}\bar{\mathbf{x}}_w^d$ ,  ${}^{rh}\bar{R}_w^d$ ,  ${}^{lh}\bar{\mathbf{x}}_w^d$ ,  ${}^{lh}\bar{R}_w^d$ . This posture is chosen so as to reach the hand contacts, assumed to be fixed in the world reference frame<sup>5</sup>. Whenever either  $T_X^{rh}$  or  $T_X^{lh}$  is activated, a new reference trajectory is instantiated corresponding to an interpolation (e.g. a minimum jerk trajectory) from the current hand position to the contact locations; the trajectory duration is left as an open parameter.

The definition of the interaction force tasks  $T_f^{rf}$ ,  $T_f^{lf}$ ,  $T_f^{rh}$ ,  $T_f^{lh}$  and the associated reference force values  $\mathbf{f}_{rh}^*$ ,  $\mathbf{f}_{lh}^*$ ,  $\mathbf{f}_{rf}^*$ ,  $\mathbf{f}_{lf}^*$  requires more care. The motivation resides in the fact that these tasks are intertwined to the center of mass task  $T_X^{com}$  through the system dynamics. At each instant of time external forces determine the center of mass acceleration according to the following:

$$m\ddot{\mathbf{x}}_{com} = \mathbf{f}_{rh} + \mathbf{f}_{lh} + \mathbf{f}_{rf} + \mathbf{f}_{lf}, \quad (6)$$

being  $m$  the total mass of the robot and  $\mathbf{f}_{rh}$ ,  $\mathbf{f}_{lh}$ ,  $\mathbf{f}_{rf}$  and  $\mathbf{f}_{lf}$  the Cartesian (not the spatial!) forces exchanged by the robot at the contact points. In view of this consideration we need to be careful in choosing reference interaction forces  $\mathbf{f}_{rh}^*$ ,  $\mathbf{f}_{lh}^*$ ,  $\mathbf{f}_{rf}^*$ ,  $\mathbf{f}_{lf}^*$  and reference center of mass acceleration  $\ddot{\mathbf{x}}_{com}^*$ . The notation is slightly complicated by the fact that in the different scenario states  $S_1, \dots, S_5$  the meaning of  $\mathbf{f}$  (and consequently  $\mathbf{f}^*$ ) in (1) changes. In particular we have:

$$S_1, S_2 : \mathbf{f} = \begin{bmatrix} \mathbf{f}_{rf} \\ \mathbf{f}_{lf} \end{bmatrix}, \quad S_3 : \mathbf{f} = \begin{bmatrix} \mathbf{f}_{rh} \\ \mathbf{f}_{rf} \\ \mathbf{f}_{lf} \end{bmatrix}, \quad S_4 : \mathbf{f} = \begin{bmatrix} \mathbf{f}_{lh} \\ \mathbf{f}_{rf} \\ \mathbf{f}_{lf} \end{bmatrix}, \quad S_5 : \mathbf{f} = \begin{bmatrix} \mathbf{f}_{rh} \\ \mathbf{f}_{lh} \\ \mathbf{f}_{rf} \\ \mathbf{f}_{lf} \end{bmatrix}, \quad (7)$$

where each spatial force has the usual structure  $\mathbf{f}_i = [\mu_i^\top, f_i^\top]^\top$  being  $f_i$  the Cartesian force and  $\mu_i$  the Cartesian torque always defined in the link reference frame. In the different states, the following constraints on  $\mathbf{f}^*$  hold with the definitions below:

- $S_1$  and  $S_2$ : desired force  $\mathbf{f}^*$  is such that:

$$C_{S_1}\mathbf{f}^* = C_{S_2}\mathbf{f}^* = \mathbf{f}_{rf}^* + \mathbf{f}_{lf}^* = m\ddot{\mathbf{x}}_{com}^*. \quad (8)$$

- $S_3$ : desired force  $\mathbf{f}^*$  is such that:

$$C_{S_3}\mathbf{f}^* = \mathbf{f}_{rh}^* + \mathbf{f}_{rf}^* + \mathbf{f}_{lf}^* = m\ddot{\mathbf{x}}_{com}^*. \quad (9)$$

- $S_4$ : desired force  $\mathbf{f}^*$  is such that:

$$C_{S_4}\mathbf{f}^* = \mathbf{f}_{lh}^* + \mathbf{f}_{rf}^* + \mathbf{f}_{lf}^* = m\ddot{\mathbf{x}}_{com}^*. \quad (10)$$

- $S_5$ : desired force  $\mathbf{f}^*$  is such that:

$$C_{S_5}\mathbf{f}^* = \mathbf{f}_{rh}^* + \mathbf{f}_{lh}^* + \mathbf{f}_{rf}^* + \mathbf{f}_{lf}^* = m\ddot{\mathbf{x}}_{com}^*. \quad (11)$$

<sup>5</sup>In practice final postures are positions on the positions where the hand contact should occur.

where we defined:

$$\begin{aligned} C_{S_1} &= \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix}, \\ C_{S_2} &= \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix}, \\ C_{S_3} &= \begin{bmatrix} C_{S_1} & 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix}, \\ C_{S_4} &= \begin{bmatrix} C_{S_1} & 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix}, \\ C_{S_5} &= \begin{bmatrix} C_{S_3} & 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix}. \end{aligned}$$

In general these constraints (deriving from the Newton equation) can be completed with additional constraints which derive from the angular momentum equilibrium, but we left this complete formulation out to simplify the notation. The interested reader should refer to [Orin et al., 2013]. Similarly to simplify the notation we neglected the effect of gravitational forces, which should be taken into account when balancing forces. Correct computations can be simply obtained by modifying all equations to include a term  $m\mathbf{g}$  being  $\mathbf{g}$  the gravitational acceleration. Otherwise all equations still hold referring to the system “proper acceleration”.

In the TSID formalism, tasks  $T_{\mathbf{f}}^{rf}$ ,  $T_{\mathbf{f}}^{lf}$ ,  $T_{\mathbf{f}}^{rh}$ ,  $T_{\mathbf{f}}^{lh}$  can be solved solving (1a)-(1b)-(1c) at the first stage of the hierarchy (i.e. with maximum priority). However, it is worth observing that the solution of (1a)-(1b)-(1c) might be in conflict with the center of mass task  $T_{\mathbf{x}}^{com}$  if it is specified in (1d) with lower priority with respect to the force task. Therefore, in order to enforce the compatibility of the tasks  $T_{\mathbf{f}}^{rf}$ ,  $T_{\mathbf{f}}^{lf}$ ,  $T_{\mathbf{f}}^{rh}$ ,  $T_{\mathbf{f}}^{lh}$  and  $T_{\mathbf{x}}^{com}$ , we pre-compute a reference force  $\mathbf{f}^*$  compatible with the reference center of mass acceleration  $\mathbf{x}_{com}^*$ . Even if we impose the above constraints on  $\mathbf{f}^*$ , we do not identify a unique solution. Additional constraints or requirements needs to be imposed in order to properly define  $\mathbf{f}^*$ . In order to simplify this ambiguity, the following problem can be solved when at state  $S_i$ :

$$\mathbf{f}^* = \arg \min_{\mathbf{f}} \|\mathbf{f} - \mathbf{f}_0\|_W^2 \text{ s.t. } C_{S_i} \mathbf{f} = m\ddot{\mathbf{x}}_{com}^*, \quad (12)$$

where  $\|\cdot\|_W$  denotes a norm weighted with the matrix  $W = W^\top > 0$ . The solution of this optimization is given by:

$$\mathbf{f}^* = mC_{S_i,W}^\dagger \ddot{\mathbf{x}}_{com}^* + (I - C_{S_i,W}^\dagger C_{S_i}) \mathbf{f}_0, \quad (13)$$

having defined  $C_{S_i,W}^\dagger = (C_{S_i}^\top W C_{S_i})^{-1} C_{S_i}^\top W$ , the weighted pseudo-inverse of  $C_{S_i,W}$ . The solution of this optimization gives a set of desired forces  $\mathbf{f}^*$  which are compatible with the desired center of mass acceleration  $\ddot{\mathbf{x}}_{com}^*$ .

It was worth noticing that in the above optimization constraints involve only the Cartesian forces while the norm to be minimized is referred to the spatial forces, thus including both Cartesian forces and torques. As a consequence, (12) naturally imposes a solution with null Cartesian torques at all contacts. It is worth noting here that all Cartesian forces and torques are defined in a reference frame attached to the rigid link on which the contact is active. Additionally, if these reference frames are properly chosen within the contact support polygon, (12) naturally induces the center of pressure condition (sometimes called also the ZMP condition) for all the contacts which guarantees the stability of the contact itself.

Once a compatible  $\mathbf{f}^*$  is computed solving (12), tasks  $T_{\mathbf{f}}^{rf}$ ,  $T_{\mathbf{f}}^{lf}$ ,  $T_{\mathbf{f}}^{rh}$ ,  $T_{\mathbf{f}}^{lh}$  and  $T_{\mathbf{x}}^{com}$  are simultaneously computed solving the associated optimization (1a)-(1b)-(1c) which has

maximum priority in the hierarchy. Alternatively, a possible solution in a single step consists in solving the following problem:

$$\boldsymbol{\tau}^* = \arg \min_{\boldsymbol{\tau} \in \mathbb{R}^n} \|\boldsymbol{f}\|_W^2 \quad (14a)$$

$$s.t. \quad M\ddot{\boldsymbol{q}} + \boldsymbol{h} - J_c^T \boldsymbol{f} = S^T \boldsymbol{\tau} \quad (14b)$$

$$J_c \ddot{\boldsymbol{q}} + \dot{J}_c \dot{\boldsymbol{q}} = 0 \quad (14c)$$

$$J_{com} \ddot{\boldsymbol{q}} + \dot{J}_{com} \dot{\boldsymbol{q}} = \ddot{\boldsymbol{x}}_{com}^*. \quad (14d)$$

The solution of (14) has strong analogies with the solution of (12) and (1a)-(1b)-(1c) but a complete analysis of the underlying similarities is out of the scope of the present report.

In order to avoid discontinuities in the desired forces to be tracked by the controller we should chose the trajectories  $\boldsymbol{f}_{rh}^*(t)$ ,  $\boldsymbol{f}_{lh}^*(t)$ ,  $\boldsymbol{f}_{rf}^*(t)$ ,  $\boldsymbol{f}_{lf}^*(t)$  so that at the moment of task switching the desired force equal the current one. In order to do so, we simply exploit the fact that at any time, measurements  $\boldsymbol{f}$ ,  $\ddot{\boldsymbol{x}}_{com}$  should satisfy the equilibrium condition  $C_{S_i} \boldsymbol{f} = m\ddot{\boldsymbol{x}}_{com}$ . Therefore, at the switching instant  $t_s$  we can simply solve (12) with  $\boldsymbol{f}_0 = C_{S_i} \boldsymbol{f}(t_s)$  and  $\ddot{\boldsymbol{x}}_{com}^* = \ddot{\boldsymbol{x}}_{com}(t_s)$  to obtain a solution  $\boldsymbol{f}^* = \boldsymbol{f}(t_s)$ . The values of  $\ddot{\boldsymbol{x}}_{com}^*$  and  $\boldsymbol{f}_0$  are then obtained as described in Section 4.4 as the output of a *PID* filter on the difference between desired trajectories  $\ddot{\boldsymbol{x}}_{com}^d(t)$ ,  $\boldsymbol{f}^d(t)$  and current values  $\ddot{\boldsymbol{x}}_{com}$ ,  $\boldsymbol{f}$ . In particular, according the definitions that we gave in (4) and (5) the final reference for forces and accelerations will be:

$$\begin{aligned} \ddot{\boldsymbol{x}}_{com}^*(t) &= \ddot{\boldsymbol{x}}_{com}^d(t) + K_d^{com} (\dot{\boldsymbol{x}}_{com}^d(t) - \dot{\boldsymbol{x}}_{com}) + K_p^{com} (\boldsymbol{x}_{com}^d(t) - \boldsymbol{x}_{com}), \\ \boldsymbol{f}_0(t) &= \boldsymbol{f}^d(t) + K_i^f \int_{t_s}^t (\boldsymbol{f}^d(\tau) - \boldsymbol{f}(\tau)) d\tau, \\ \boldsymbol{f}^* &= mC_{S_i, W}^\dagger \ddot{\boldsymbol{x}}_{com}^* + (I - C_{S_i, W}^\dagger C_{S_i}) \boldsymbol{f}_0. \end{aligned}$$

At the switching instant  $t_s$  we choose  $\boldsymbol{f}^d = \boldsymbol{f}$ ,  $\ddot{\boldsymbol{x}}_{com}^d = \ddot{\boldsymbol{x}}_{com}$ ,  $\dot{\boldsymbol{x}}_{com}^d = \dot{\boldsymbol{x}}_{com}$ ,  $\boldsymbol{x}_{com}^d = \boldsymbol{x}_{com}$  so as to obtain  $\boldsymbol{f}^* = \boldsymbol{f}$  (remeber that  $\boldsymbol{f}$ ,  $\ddot{\boldsymbol{x}}_{com}$  are always compatible). Then we can choose  $\boldsymbol{f}^d(t)$  and  $\ddot{\boldsymbol{x}}_{com}^d(t)$  arbitrarily with the guarantee that the resulting  $\boldsymbol{f}^*$  and  $\ddot{\boldsymbol{x}}_{com}^*$  will always be dynamically consistent.

## 4.6 Task prioritization

In the TSID framework, control tasks are executed with a lexicographic priority, i.e. tasks with higher priority should be satisfied at the expenses of the tasks with lower priority. In order to represent  $T_i$  with a lower lexicographic priority with respect to  $T_j$  we use the notation  $T_i \prec T_j$ . In consideration of what discussed above, tasks force tasks  $T_{\boldsymbol{f}}^{rh}$ ,  $T_{\boldsymbol{f}}^{lf}$ ,  $T_{\boldsymbol{f}}^{rh}$ ,  $T_{\boldsymbol{f}}^{lh}$  and center of mass tasks  $T_{\boldsymbol{X}}^{com}$  are grouped in a unique optimization which we denote  $T_{\boldsymbol{f}}^{com}$ . In the different states, priorities are handled as follows:

- $S_1$  and  $S_2$ :  $T^q \prec T_{\boldsymbol{f}}^{com}$ .
- $S_2$ :  $T^q \prec T_{\boldsymbol{X}}^{lh} \prec T_{\boldsymbol{X}}^{rh} \prec T_{\boldsymbol{f}}^{com}$ .
- $S_3$ :  $T^q \prec T_{\boldsymbol{X}}^{lh} \prec T_{\boldsymbol{f}}^{com}$ .

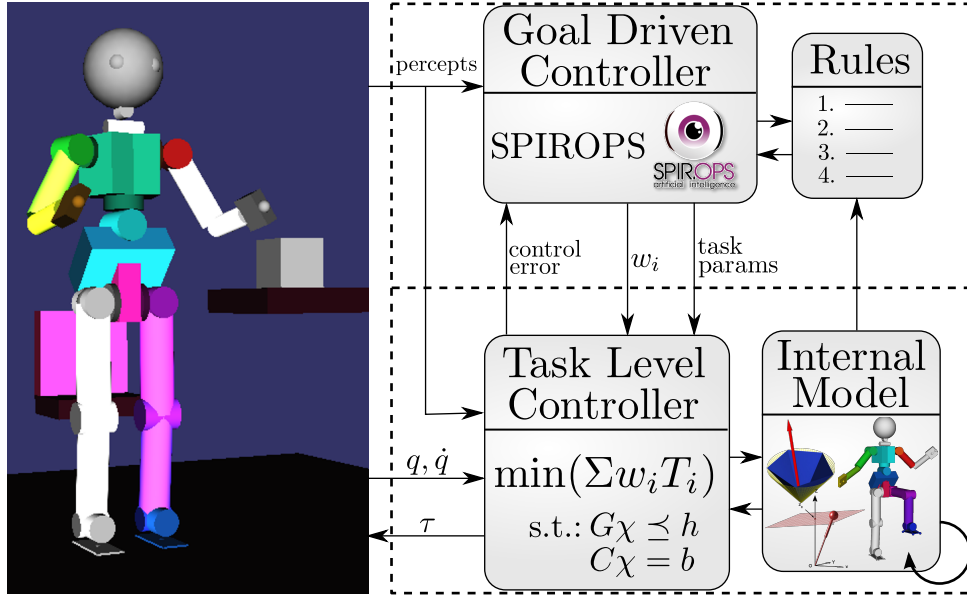


Figure 1: Global architecture of the soft hierarchy Task Space Inverse Dynamics control scheme developed at UPMC.

- $S_4: T^q \prec T_X^{rh} \prec T_f^{com}$ .
- $S_5: T^q \prec T_f^{com}$ .

#### 4.7 Alternative controller formulation

The Task Space Inverse Dynamics (TSID) can be formulated in a fashion where the ordering of the tasks priorities can be based on soft priorities rather than on a strict lexicographic order. The control problem can still be written as an optimization problem under constraints and its resolution only requires one optimization level, independently from the number of tasks to be achieved. The work of Joseph Salini [Salini, 2012] at UPMC lays the foundation of this alternative formulation paradigm. Figure 1 illustrates the principles of this formulation where a weight is associated to each task accordingly to its importance and can be modulated using a rule-based high level decision making process.

While appealing, the work of J. Salini can not guarantee strict priorities between tasks. On the other hand, control approaches based on strict hierarchies induce discontinuities in the control input when modifying the ordering of the hierarchy and require to solve as many optimization problems as the number of tasks. In order to solve these issues and as a contribution to WP3, UPMC recently developed a controller named Generalized Smooth Hierarchical Control (GSHC) bridging the gap between strict and soft hierarchy Task Space Inverse Dynamics controllers. The GHSC algorithm, submitted to the IEEE Transaction on Robotics journal [Liu et al., 2013], provides with a control framework such that:

- it can handle multiple tasks with strict and soft priorities;

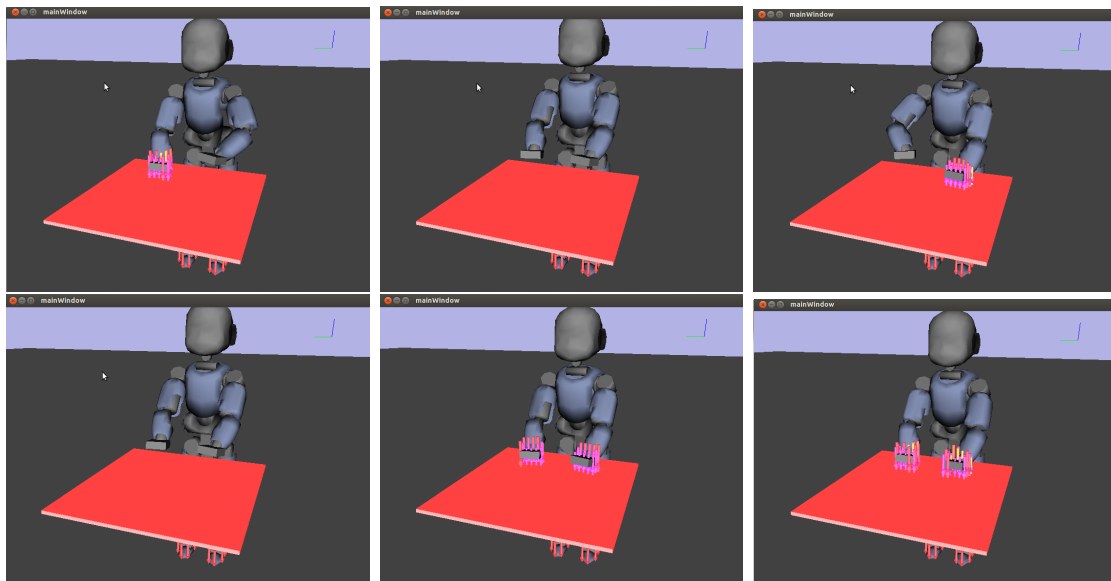


Figure 2: Screenshots of the validation scenario simulated in XDE.

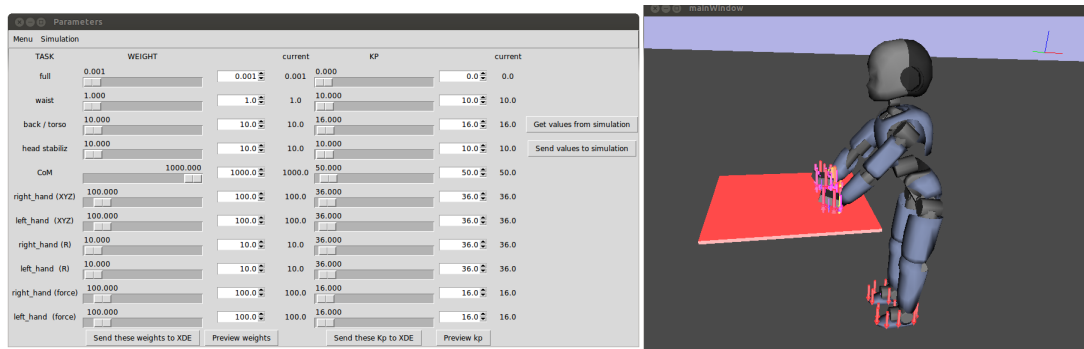


Figure 3: Tuning of the task priority parameters in XDE.

- smooth and simultaneous rearrangements of multiple task priorities can be achieved easily (by the smooth variations of relevant bounded scalar values);
- smooth insertion and deletion of tasks can be performed;
- various constraints imposed by the robot body (joint limits, actuation capabilities...) and the environment (contacts to maintain, obstacles to avoid...) can be respected.

The first year demo described in this deliverable thus also benefits from this alternative controller. It is being intensively tested on a simulated iCub in the XDE simulator at UPMC before being transferred to IIT for an implementation on the real robot. The validation scenario is also being investigated (see Figure 2). For example, UPMC is testing how to smoothly sequence force and position tasks when contact is detected. UPMC also has developed a set of tools to rapidly prototype and test different behaviors of the robot by acting on tasks priority parameters (see Figure 3).

## References

- [Aghili, 2005] Aghili, F. (2005). A unified approach for inverse and direct dynamics of constrained multibody systems based on linear projection operator: applications to control and simulation. *Robotics, IEEE Transactions on*, 21(5):834–849.
- [Del Prete, 2013] Del Prete, A. (2013). *Control of Contact Forces using Whole-Body Force and Tactile Sensors: Theory and Implementation on the iCub Humanoid Robot*. PhD thesis, Istituto Italiano di Tecnologia.
- [Featherstone, 2008] Featherstone, R. (2008). *Rigid Body Dynamics Algorithms*. Springer.
- [Liu et al., 2013] Liu, M., Tan, Y., and Padois, V. (2013). Generalized smooth hierarchical control. *Submitted to IEEE Transactions on Robotics*.
- [Orin et al., 2013] Orin, D. E., Goswami, A., and Lee, S.-H. (2013). Centroidal dynamics of a humanoid robot. *Auton. Robots*, 35(2-3):161–176.
- [Righetti et al., 2013] Righetti, L., Buchli, J., Mistry, M., Kalakrishnan, M., and Schaal, S. (2013). Optimal distribution of contact forces with inverse dynamics control. *The International Journal of Robotics Research*.
- [Salini, 2012] Salini, J. (2012). *Dynamic control for the task/posture coordination of humanoid: toward synthesis of complex activities*. PhD thesis, Université Pierre et Marie Curie, Paris, France.
- [Sardain and Bessonnet, 2004] Sardain, P. and Bessonnet, G. (2004). Forces acting on a biped robot. center of pressure-zero moment point. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 34(5):630–637.
- [Siciliano and Khatib, 2008] Siciliano, B. and Khatib, O. (2008). *Springer Handbook of Robotics*. Gale virtual reference library. Springer.
- [Siciliano et al., 2009] Siciliano, B., Sciavicco, L., and Villani, L. (2009). *Robotics: Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing. Springer.

## A Center of pressure: *CoP*

In this Appendix we recall some concepts that have been extensively discussed in the CoDyCo deliverable D2.1, where the main focus is on human motion analysis. Remarkably, concepts like center of pressure and (fictitious) zero moment point apply to both humans and robots thus representing a strong modeling analogy between the two fields of research. In general, all dynamic principles (Newton-Euler equation, energy, work, forces, torques, etc.) represent an abstraction layer that allows the CoDyCo consortium to have a common set of tools which can be independently applied to both humans and robots.

### A.1 *CoP* definition

In order to define the center of pressure it is first necessary to define the pressure. Pressure is a scalar quantity, defined as the amount of force acting perpendicularly per unit area. Let's assume that a set of distributed forces acts on a surface  $\mathcal{S}$  and let  $\mathbf{s}$  be a point on the surface, i.e.  $\mathbf{s} \in \mathcal{S}$ . Let's denote with  $f_n(\mathbf{s})$  the component of  $\mathbf{f}(\mathbf{s})$  orthogonal to  $\mathcal{S}$  in  $\mathbf{s}$ . Given  $\mathbf{s} \in \mathcal{S}$ , assuming an element  $\Delta S(\mathbf{s})$  of infinitesimal area centered in  $\mathbf{s}$ , let  $f_n(\Delta S, \mathbf{s})$  be the orthogonal force acting on  $\Delta S(\mathbf{s})$ . The pressure  $p$  in  $\mathbf{s}$  is defined as:

$$p(\mathbf{s}) = \lim_{\Delta S \rightarrow 0} \frac{f_n(\Delta S, \mathbf{s})}{\Delta S} \quad \mathbf{s} \in \Delta S.$$

Under this definition, the center of pressure *CoP* with respect to the reference frame  $\Sigma_1$  is defined as:

$$CoP_1 = \frac{\int_{\mathcal{S}} \mathbf{s}_1 \cdot p(\mathbf{s}) dS}{\int_{\mathcal{S}} p(\mathbf{s}) dS},$$

where  $\mathbf{s}_1$  indicates the position of  $\mathbf{s}$  in  $\Sigma_1$ . If  $\mathcal{S}$  is a three dimensional surface, the *CoP* is a three dimensional point. It is totally equivalent to the concept of the center of mass, but instead of being related to the first order moment of the mass it involves the first order moment of the pressure. A number of interesting properties follow and in particular if  $\mathcal{S}$  is convex then  $CoP \in \mathcal{S}$ . Another important property is  $CoP_2 = {}^2T_1 \cdot CoP_1$  if  ${}^2T_1$  transforms from  $\Sigma_1$  to  $\Sigma_2$ . The total orthogonal force  $f_n$  acting on the surface is given by:

$$f_n = \int_{\mathcal{S}} p(\mathbf{s}) dS,$$

and the global force  $\mathbf{f}$  is the sum of the global normal force ( $\mathbf{f}_n$ ) and the global tangential force ( $\mathbf{f}_t$ ) due to the tangential elementary forces  $\mathbf{t}(\mathbf{s})$ :

$$\mathbf{f} = \mathbf{f}_n + \mathbf{f}_t = f_n \cdot \mathbf{n} + \int_{\mathcal{S}} \mathbf{t}(\mathbf{s}) dS = \int_{\mathcal{S}} p(\mathbf{s}) \cdot \mathbf{n} + \mathbf{t}(\mathbf{s}) dS,$$

being  $\mathbf{n}$  the surface normal. Clearly we have  $\langle \mathbf{t}(\mathbf{s}), \mathbf{n} \rangle = 0$ , for all  $\mathbf{s} \in \mathcal{S}$ . Also the global moment  $\boldsymbol{\mu}$  can be decomposed in two different contributions, one due to orthogonal forces  $\boldsymbol{\mu}_n$  and one due to tangential force  $\boldsymbol{\mu}_t$ . On  $\Sigma_1$  we have:

$$\boldsymbol{\mu}_{n,1} = \left( \int_{\mathcal{S}} p(\mathbf{s}) \mathbf{s}_1 dS \right) \times \mathbf{n} = CoP_1 \times (f_n \cdot \mathbf{n}), \quad \boldsymbol{\mu}_{t,1} = \int_{\mathcal{S}} \mathbf{s}_1 \times \mathbf{t}(\mathbf{s}) dS.$$

In the case of flat surfaces  $\mathcal{S}$ , it can be shown [Sardain and Bessonnet, 2004] that the *CoP* is the unique point  $C \in \mathcal{S}$  on which  $\boldsymbol{\mu}_{n,C} = 0^6$ . Equivalently, we can say that the *CoP* is the unique point  $C \in \mathcal{S}$  on which  $\boldsymbol{\mu}_C = \boldsymbol{\mu}_{t,C}$ , i.e. the total torque  $\boldsymbol{\mu}_C$  is orthogonal to  $\mathcal{S}$ . It is here worth noticing that on flat surfaces the torque due to normal forces  $\boldsymbol{\mu}_t$  is parallel to the plane normal  $\mathbf{n}$  while the torque due to tangential forces  $\boldsymbol{\mu}_n$  lies on the plane.

## A.2 *CoP* and dynamic equilibrium

The concept of center of pressure is often related to the concept of dynamic equilibrium for a free floating articulated body. Within the multiple point of views proposed in literature we primarily follow the one in [Sardain and Bessonnet, 2004]. Therefore, we do not make any formal distinction between the concept of *CoP* (center of pressure) and the concept of *ZMP* (zero moment point) because, as it is clearly stated in [Sardain and Bessonnet, 2004], “*CoP* and the *ZMP* are the same point. [...] This fact does not admit any discussion, it is true whatever the stability of the balance is, and in particular it is true when the walker is falling down, as long as a contact exists with the ground”. What might generate confusion is that Vukobratović, the author that introduced the concept of *ZMP*, also introduced the concept of fictitious *ZMP* lately renamed computed zero moment point (*cZMP*, see chapter sixteen of the handbook of robotics [Siciliano and Khatib, 2008]).

Similarly to the *CoP* the *cZMP* is a point associated with a force/torque couple  $\mathbf{f}^b, \boldsymbol{\mu}^b$  represented in a given reference frame  $\Sigma_b$ . Similarly to the *CoP* its definition passes through an associated plane  $\mathcal{S}$  (the planar support). In particular the *cZMP* is defined as the unique point  $C \in \mathcal{S}$  where  $\boldsymbol{\mu}_C^b$  is orthogonal to  $\mathcal{S}$ . The difference between the *CoP* and the *cZMP* relies on the fact that the *CoP* is defined on the force/torque  $(\mathbf{f}, \boldsymbol{\mu})$  that the foot exchanges with the ground, while the *cZMP* is defined on all the other forces/torques  $(\mathbf{f}^b, \boldsymbol{\mu}^b)$  that the foot exchanges with the articulated body. At the foot center of mass the usual Newton-Euler equations should hold:

$$m_f \mathbf{a}_f = \mathbf{f}^b + \mathbf{f}, \quad \mathbf{I}_f \dot{\boldsymbol{\omega}}_f = \boldsymbol{\mu}^b + \boldsymbol{\mu},$$

where all quantities are referred to a reference frame in the foot center of mass,  $m_f$  is the total mass of the foot,  $\mathbf{a}_f$  is the foot linear acceleration,  $\mathbf{I}_f$  is the foot inertia tensor and  $\dot{\boldsymbol{\omega}}_f$  the foot angular acceleration.

If the foot is at equilibrium ( $\dot{\boldsymbol{\omega}}_f = \mathbf{a}_f = 0$ ) then the *CoP* and the *cZMP* are defined on equal and opposite force/torque couples since we have  $(\mathbf{f}^b, \boldsymbol{\mu}^b) = (-\mathbf{f}, -\boldsymbol{\mu})$ . It can be shown<sup>7</sup> that in this case the *CoP* and the *cZMP* coincide. As a consequence the *cZMP* lies in the foot support polygon.

<sup>6</sup>We can prove that indeed  $\boldsymbol{\mu}_{n,CoP} = 0$ . First of all, let's observe that:

$$\boldsymbol{\mu}_{n,2} = \boldsymbol{\mu}_{n,1} + \mathbf{f}_n \cdot \mathbf{n} \times {}^2\mathbf{s}_1,$$

where  ${}^2\mathbf{s}_1$  is the vector from  $\Sigma_1$  to  $\Sigma_2$ . We have:

$$\boldsymbol{\mu}_{n,CoP} = \left( \int_{\mathcal{S}} p(\mathbf{s}) \mathbf{s}_1 dS \right) \times \mathbf{n} + \int_{\mathcal{S}} p(\mathbf{s}) \cdot \mathbf{n} dS \times \frac{\int_{\mathcal{S}} \mathbf{s}_1 \cdot p(\mathbf{s}) dS}{\int_{\mathcal{S}} p(\mathbf{s}) dS} = 0.$$

<sup>7</sup>Given a force/torque couple  $\mathbf{f}, \boldsymbol{\mu}$  and a plane  $\mathcal{S}$  the associated *CoP* is the same associated with the opposite force/torque couple  $-\mathbf{f}, -\boldsymbol{\mu}$  on the same plane  $\mathcal{S}$ .

Vice-versa, let's prove that if  $cZMP$  is outside the support polygon then necessarily either  $\dot{\omega}_f \neq 0$  or  $\mathbf{a}_f \neq 0$  and therefore the foot is not in equilibrium. Suppose by contradiction that  $\dot{\omega}_f = \mathbf{a}_f = 0$ . As previously pointed out  $CoP$  and  $cZMP$  coincide and since the  $CoP$  is necessarily within the support polygon we conclude that the  $cZMP$  is also within the support polygon thus leading to the contradiction.

Finally, we would like to give sufficient conditions for the foot stability. In particular, can we conclude that if the  $cZMP$  is within the support polygon then the foot is at equilibrium? Unfortunately this conclusion cannot be drawn stated as it is. The main motivation resides in the fact that the  $cZMP$  might be associated with a force/torque couple  $\mathbf{f}^b, \boldsymbol{\mu}^b$  whose tangential force components ( $\mathbf{f}_t$ ) are outside the friction cones. Remarkably, this might always be the case since the zero moment point condition does not involve the tangential forces. Similarly, the  $cZMP$  might be associated with a force/torque couple  $\mathbf{f}^b, \boldsymbol{\mu}^b$  whose normal torque component ( $\boldsymbol{\mu}_t$ ) is outside the friction cones. Again, this might always be the case since the zero moment point condition does not involve the torque component normal to the contact plane. What it can be proven (but the complete proof is outside the scope of the current deliverable) is that if the  $cZMP$  is inside the support polygon and if the the associated force/torque couple  $\mathbf{f}^b, \boldsymbol{\mu}^b$  have tangential force components ( $\mathbf{f}_t^b$ ) and normal torque components ( $\boldsymbol{\mu}_t^b$ ) compatible with the friction cone, then the foot is at equilibrium.