



FP7-600716

Whole-Body Compliant Dynamical Contacts in Cognitive Humanoids

D3.1

Local solver in rigid-world cases

Editor(s)	Vincent Padois ^{1,2}
Responsible Partner	UPMC
Affiliations	¹ Sorbonne Universités, UPMC Paris 06, UMR 7222, Institut des Systèmes Intelligents et de Robotique (ISIR), F-75005, Paris, France. ² CNRS, UMR 7222, Institut des Systèmes Intelligents et de Robotique (ISIR), F-75005, Paris, France.
Status-Version:	Final-1.0
Date:	Feb. 28, 2015
EC Distribution:	Consortium
Project Number:	600716
Project Title:	Whole-Body Compliant Dynamical Contacts in Cognitive Humanoids

Title of Deliverable:	Local solver in rigid-world cases
Date of delivery to the EC:	28/2/2015

Workpackage responsible for the Deliverable	WP3
Editor(s):	Vincent Padois
Contributor(s):	Aurlien Ibanez, Mingxing Liu, Ryan Lober, Vincent Padois, Nicolas Perrin, Joseph Salini (UPMC), Andrea Del Prete, Francesco Nori, Daniele Pucci, Francesco Romano, Silvio Traversaro (IIT)
Reviewer(s):	
Approved by:	All Partners
Abstract	<p>This deliverable summarizes the contribution of the CoDyCo consortium in the domain of whole-body controllers. First, a review of whole-body control is provided. The analysis of existing research works in that domain clearly positions so-called whole-body controllers as locally optimal, multi-tasks, hierarchical controllers applied to free-floating, underactuated systems, namely humanoids. These local controllers assume the existence of some global controllers providing them with a description of the tasks and contacts to be achieved at each control instant. The contributions of the CoDyCo consortium in the domain of whole-body controllers are then introduced and a concise description of the PhD and postdoc work of Salini, Del Prete and Liu is provided. The report is concluded by a review of the application of these works on the CoDyCo scenarii of Year 1 and Year 2.</p>
Keyword List:	Whole-body controllers, Rigid world, Multi-contacts, Goal-oriented tasks

Document Revision History

Version	Date	Description	Author
v. 0.1	Feb. 20, 2015	Initial draft	Vincent Padois
v. 0.5	Feb. 24, 2015	Intermediate version	Vincent Padois
v. 0.9	Feb. 27, 2015	Final version	Vincent Padois
v. 1.0	Feb. 28, 2015	Proofread version	Francesco Nori

Table of Contents

1	Introduction	5
2	Whole-Body control and the CoDyCo contribution	5
2.1	Whole-body control: global vs local control problem	6
2.2	Hierarchical control	7
2.2.1	Approaches for handling a strict hierarchy	7
2.2.2	Weighting strategy	9
2.2.3	Task transitions	9
2.3	Controllers developed within the framework of CoDyCo	10
3	Results	12
3.1	Year 1 scenario: balancing on multiple rigid contact points	12
3.2	Year 2 scenario: balancing on feet while performing goal directed actions . . .	13
3.3	Other use case of interest	13
4	Conclusion	15
	References	21
A	Generalized Hierarchical Control	22
B	Whole-Body Hierarchical Motion and Force Control for Humanoid Robots	38
C	iCub Whole-body Control through Force Regulation on Rigid Non coplanar Contacts	48
D	Improvement of a Balancing Force and Posture Controller with Torque Minimization	76

Index of Figures

1	Views of iCub balancing on multiple contacts using one the whole-body controller developed within the framework of the CoDyCo project.	12
2	View of the most recent developments in the implementation of the CoDyCo whole-body controller: iCub balancing on one foot while interacting with humans.	13
3	View of some recent developments in the implementation of the CoDyCo whole-body controller in goal oriented actions	14
4	View of a sequence of motion generated using a robotic controller as an attempt to reproduce results obtained in human reaching with additional contact experiments.	14
5	A view of iCub automatically adapting its contact locations based on some perceived balance perturbation.	15

1 Introduction

This deliverable summarizes the contribution of the CoDyCo consortium in the domain of whole-body controllers. First, a review of whole-body control is provided. The analysis of existing research works in that domain clearly positions so-called whole-body controllers as locally optimal, multi-tasks, hierarchical controllers applied to free-floating, underactuated systems, namely humanoids. These local controllers assume the existence of some global controllers providing them with a description of the tasks and contacts to be achieved at each control instant. The contributions of the CoDyCo consortium in the domain of whole-body controllers are then introduced and a concise description of the PhD and postdoc work of Salini, Del Prete and Liu is provided. The report is concluded by a review of the application of these works on the CoDyCo scenarii of Year 1 and Year 2.

2 Whole-Body control and the CoDyCo contribution

With the emergence of humanoid robots and virtual humans, a fairly large research effort is concentrated on the control of these systems. They are indeed complex by nature and their control remains an opened research question.

First, humanoids are free-floating, underactuated systems, *i.e.* systems whose configuration is described by their n joint angles and their 3D pose expressed in some inertial frame¹, while only the joints are directly actuated. Their control is possible thanks to the existence of contacts with the environment². These contacts generate wrenches applied to the robot that provide, if properly used, the missing actuation capabilities of the system. While these contacts are necessary, they are challenging as:

- their existence is conditioned by several conditions:
 - contact points (e.g. feet) can push on the environment (e.g. ground) but not pull: contacts are unilateral;
 - the contact forces have to lie in their associated Coulomb friction cone to avoid slippage;
 - the dynamic wrench applied at the center of mass of the robot has to be controlled in such a way that it can actually be compensated for by the contact forces and thus avoid tipping over.
- they largely constrain the pose of the contact points and thus the overall motion of the system;
- they are discrete in nature: contrarily to wheeled systems, modifying the location of the contact point in a controlled manner³ requires to break contact.

¹This pose in SE(3) can be locally described using a minimal number of 6 coordinates.

²These contacts are generally located at the feet but can potentially occur anywhere at the surface of the robot.

³*i.e.* without slipping.

The second source of complexity comes from the large number of degrees of freedom (DOF) and the tree structure of these systems. These characteristics imply what is generally called "redundancy" and provide the system with the capability to perform several tasks concurrently. While redundancy is appealing, the execution of different tasks may not always be compatible and proper hierachization has to be performed at the control level in order to ensure that safety-critical tasks are not disturbed by less critical ones. This has a fairly large computational complexity.

Finally, the control of such systems requires:

- a fairly good knowledge of their kinematic and dynamic model;
- access to measures or estimation of contact point locations (and potentially measures of forces if they are to be precisely controlled);
- a measure of their 3D pose/velocity/acceleration in some inertial frame;
- a torque controlled robot.

Modeling and identifying the model parameters is thus of major importance. Proper sensing (joint level odometry, force/torque sensing, inertial measurment units) and estimation capabilities are also mandatory.

2.1 Whole-body control: global vs local control problem

Solving the global whole-body control problem as a whole, *i.e.* finding at each control instant the adequate actuation torque given some high level mission description such as "Move object A from place B to C", is beyond reach. As a consequence, what is meant by whole-body control often corresponds to a local control problem, less complex than the global one.

The main assumption is that the operational tasks⁴ to perform at time t in order to achieve a global goal are given. For simple missions, these tasks, their associated goals and relative importances are very often predefined in a very *ad hoc* way. This is hardly possible when considering the specific problem of changing the contact points location and controlling balance. Indeed, although the dynamics of multibody systems are well understood, direct approaches to resolve the motion of all degrees of freedom for locomotion is computationally intractable in real-time due to their large number and to the non-linear behaviour of the system.

In fact, to generate dynamically balanced locomotion trajectories efficiently, current methods are based on simplified models of the robot dynamics [1]. One of the most widespread model is the Inverted Pendulum Model (*IPM*) [2], where the mass of the robot is assumed to be concentrated at its center of mass (*CoM*). Although more accurate models are sometimes required [3], the simple IPM is suitable for many situations where the rotational inertial effects of the robot arms, legs and torso are negligible or can compensate for each other. An example

⁴Operational space tasks are often expressed in 3D space and associated to end-effectors but the concept can be extended to any space that can be expressed as a function of joint space coordinates, including the joint space itself.

of such a scenario includes walking at a moderately fast pace. When all the contacts are coplanar, a point, sometimes called the Zero Moment Point (*ZMP*) [4], that is of particular interest can be defined. Although this name has led to confusion about its physical nature [5], it is often used. A system is considered dynamically balanced if the *ZMP* lies within the support region, i.e. the 2D convex hull of the contact points [6]. In the IPM, the relationship between the CoM and *ZMP* dynamics is defined by nonlinear differential equations. However, if the vertical displacements of the CoM are set in advance, the relationship can be decoupled and expressed as linear differential equations. This substantially simplifies the problem into one that can be more efficiently and easily implemented for locomotion trajectory generation and gait control. The trajectory generation problem has been studied through analytical approaches [7, 8, 9], and more recently, using convex optimization with constraints on the *ZMP* in discrete time [10, 11, 12, 13, 14, 15]. The efficiency of convex optimization solvers allows them to be used within model predictive control (*MPC*) schemes. This is important in the real-time implementation of reactive walking. However, the *ZMP*+*MPC* approach suffers from two main drawbacks: 1) the trajectory of the CoM height (z -direction) must be known or fixed in advance, and 2) the contact points must always be coplanar, making this approach unsuitable for walking in complex unstructured environments or if the arms are to also be used. Due to these two restrictions, the *ZMP*+*MPC* approach can be regarded as a 2D CoM planner that operates in the horizontal xy -plane only.

If real-time is not a constraint, motion planning techniques based on probabilistic searches [16] have been extended to humanoids and can also be used, mostly in an offline fashion, to generate feasible motions for the overall system [17, 18, 19, 20].

Overall, given the assumption that tasks are provided, the local whole-body control problem can be formulated as follows: at each control time, given a set of n_t tasks to be performed and their relative importance, find the (locally) optimal joint torques and contact wrenches such that the task errors are minimized while respecting the task hierarchy, the equation of motion and physical constraints acting on the system (joint limits, joint velocity and joint torque limits, obstacle avoidance). The contact locations are supposed to be known and compatible with the overall desired motion of the system in terms of dynamic equilibrium/balance. The problem to be addressed is thus a very general problem of multi-tasks hierarchical control applied to humanoids.

2.2 Hierarchical control

Approaches to maintain a desired task hierarchy using a multi-objective controller draw a lot of interest. This section reviews some classical types of hierarchical control frameworks, as well as the methods for priority representation and transitions within these frameworks.

2.2.1 Approaches for handling a strict hierarchy

The simultaneous management of multiple objectives is a well known problem in Robotics. The most popular method to deal with a set of tasks is a hierarchical framework [21]: objectives are prioritized and low priority objectives are carried out only if they do not impact

the achievement of top priority objectives. This analytical⁵ method is based on null-space projections and can ensure that lower priority tasks are executed only in the null-space of higher-priority tasks, by means of the appropriate design of a null-space projector. At the velocity kinematics level, this approach is generalized with the notion of *stack of task* [22]. Based on this framework, many kinds of inversion-based control problem resolutions have emerged: potential fields and gradient projection methods [23, 24], weighted inversion techniques [25, 26], clamping [27], etc. They have also been extended to the dynamics level [23, 28, 29]. Despite the fact that most constraints are naturally expressed as inequalities (e.g. maximum actuation torques), these resolution schemes are relying on linear algebra and matrix inversions and thus express them as equalities, *i.e.* in way similar to the way tasks are expressed. These constraints-related tasks aim at getting away from the constraint (active avoidance). The number of constraints being potentially higher than the number of DOF, they mostly fail in highly constrained cases and cannot lead to control solutions that can strictly guarantee the respect of all constraints. One way to circumvent this limitation rely on the appropriate activation of the constraints based on their level of criticity at each instant. This allows the reduction of the number of tasks to be achieved at a given instant but leads to potential discontinuities in the generated control solution. To deal with these discontinuities dedicated developments have been led: adaptative gains for weighting techniques [30], smooth tasks transitions [31], progressive clamping [32] and dedicated inversion operator [33].

These techniques have also been adapted to humanoid systems at the velocity [34, 35] and torque [36] levels. In the case of humanoids, the motion of the overall system should be compatible with the contact constraints and, the constraint being expressed at the velocity level, active avoidance is not an appropriate solution. Instead, projected inverse dynamics schemes have been developed [37, 38, 39, 40, 41] in which tasks are projected into the null-space of the Jacobian of contact constraints.

Deciding when a constraint should be activated remains a problem in these approaches [42] and, as in the seminal work in [43]⁶, control approaches relying on optimization tools such as Linear Quadratic Problem solver have emerged [44, 45]. These solvers are designed to optimally choose the subset of constraints that should be considered when computing the optimal solution of the control problem. The strong mathematical background in convex optimization is such that optimization based methods mostly outperform analytical methods attempting to heuristically activate constraints as in [46, 47].

More recently, to deal with prioritized inequality constraints more easily, hierarchical quadratic programming (HQP) approaches use numerical QP solvers to solve a Hierarchical Quadratic Program [48]. The idea of HQP is to first solve a QP to obtain a solution for a higher priority task objective; and then to solve another QP for a lower priority task, without increasing the obtained minimum of the previous task objective. This prioritization

⁵"Analytical" means that an inverse is explicitly computed contrarily to "optimization-based" approaches where the computation of the inverse is implicit.

⁶In [43], an obstacle avoidance technique includes a control law structure based on a Quadratic Programming (QP). This method limits the velocities toward obstacles using inequalities. Avoidance is thus passive (does not require motion) which is more likely to avoid the collisions whatever the number of obstacles.

process corresponds to solving at best lower-priority tasks in the null-space of higher-priority tasks. The HQP algorithm is applied for solving prioritized inverse dynamics [49] and is also applied to whole-body motion control under unilateral constraints [50]. It requires to solve as many QPs as priority levels, which can be quite time consuming. The computation cost of hierarchical inverse kinematics with inequality constraints is improved by an algorithm developed in [51], which permits real time control of the HRP-4 humanoid robot. Similar work is performed in [52] where a reduction of the equation of motion allows for real time control of the SARCOS humanoid robot.

Generally, for an approach based on strict hierarchy, the relative importance of one task with respect to another one of different priority level is parametrized in a binary way: either strictly higher or strictly lower. However, in many contexts, organizing tasks by assigning them with strict priorities is not generic, *i.e.* can have some limitations. First, a strict priority is just an extreme case of the relations of task importance levels. In fact, a task may not always have a strict priority over another one and it is usually difficult to define a strict hierarchy among a set of tasks. Second, strict priorities can sometimes be too conservative so that they may completely block lower-priority tasks.

2.2.2 Weighting strategy

Non-strict priorities are usually handled by control approaches using weighting strategies [44, 45, 53, 54, 55]. These control frameworks solve all the constraints and task objectives in one QP and provide a trade-off among task objectives with different importance levels. As the performances of higher priority tasks cannot be guaranteed by simply adjusting the weights of task objectives, a prioritized control framework is proposed in [56] to ensure the performance of a higher-priority task within a user defined tolerance margin. However, this approach handles priorities of only two levels. In approaches based on weighting strategies, task priorities can be parametrized continuously. Nonetheless, even though the work in [57] on soft constraints in model predictive control could probably be adapted to provide a way to reach the extreme case of strict priorities, the existing robotic applications of these frameworks do not extend to strict hierarchies.

2.2.3 Task transitions

Earlier versions of analytical methods and HQP approaches can ensure strict priorities among tasks; however, a change in the task set, such as a swap of task priorities, may lead to discontinuity. Recently, different methods have been developed to handle task transition problems. An approach to smooth priority rearrangement between two tasks is proposed in [58, 59]. Approaches for continuous and simultaneous transitions of multiple tasks are developed in [60, 61]. A specific inverse operator is proposed in [60] to ensure continuous inverse in the analytical computation of control laws. The approach presented in [61] is based on intermediate desired values in the task space. When the number of task transitions increases, this approach recommends to use an approximation to reduce the computational cost. Smooth task transitions can be easily achieved within a framework using a weighting strategy by the continuous variation of task weights [54].

2.3 Controllers developped within the framework of CoDyCo

The whole-body controllers developped and implemented by UPMC and IIT within the framework of the CoDyCo projects are mostly related to the work of:

- Joseph Salini at UPMC whose PhD thesis addressed the topic of "Dynamic control for the task/posture coordination of humanoids: toward synthesis of complex activities" [62];
- Andrea Del Prete at IIT whose PhD thesis addressed the topic of "Control of Contact Forces using Whole-Body Force and Tactile Sensors: Theory and Implementation on the iCub Humanoid Robot" [41];
- Mingxing Liu at UPMC whose postdoctoral work has been focused on "Generalized Hierarchical Control" [63]⁷.

On one hand, the work of Salini deals with the general problem of dynamic control of task-oriented under-actuated and redundant systems considering more specifically the humanoid robotic systems or virtual humans. It brings contributions to the problem of motor activities synthesis for constrained systems by their intrinsic capacities and by particular physical interactions with the environment. More specific issues that are addressed relate to:

- the dynamic control of humanoid systems for carrying out basic activities requiring task/posture coordination perturbed by physical interactions;
- the building of sequences of continuous dynamic activities based on a repertoire of motor coordination;
- the planning and the adaptation of activities in the aim of a supervised automatic sequencing for complex non-deterministic tasks.

The major outcome of this work is a generic controller based on the completion of several tasks under constraints. As mentionned in Sec. 2.2.1, these constraints are mainly the representation of physical limitations of the robot, both internal, such as joint limits, and external as frictional contacts during interactions with the environment. The formalism is based on "task functions", the simultaneous resolution of several of them is realized by a quadratic multicriteria optimal control scheme. It is based on convex optimization methods, and the good properties of quadratic programming (QP) can efficiently perform the whole body control (in time, convergence, robustness) for the task/ posture coordination. A strategy based on weights is used to manage conflicting tasks. It allows a great flexibility of control during transitions in the sets of tasks and constraints for the continuity of the control variables. This controller has been implemented in a Arboris-Python, a physics engine also developped by Salini [64], on a virtual model of the iCub robot. These experiments illustrate the achievement of multi-task problems and exhibit the good performance of the controller for managing transitions in task sequences or when state change in the set of constraints, when a contact is broken, for example. In addition, program performance optimization of a temporal point of view are studied for different formalisms. On top of this whole-body controller,

⁷The submitted version of this article is provided in Appendix A

Salini has also developed a high-level controller to plan and adapt sequences of tasks for the realization of more complex activities based on a predetermined repertoire of actions and monitoring their performance, while ensuring the continuity of the control variables .

On the other hand, the objective of the PhD thesis of Del Prete was to improve the abilities of humanoid robots to interact with their surrounding. In particular, the idea of integrating force and touch feedback was investigated, so as to control the robot in presence of external forces acting on any part of its body. To this aim, this work exploited the humanoid robot iCub as test platform, and it tackled three main issues:

- spatial calibration of tactile sensors;
- estimation of contact forces using tactile sensors and force/torque sensors;
- prioritized position and force control.

The major outcome of this work was the synthesis of a new framework for strictly prioritized position and force control of floating-base robots. The framework was compared to other state-of-the-art similar frameworks, both analytically and in simulation, and it proved preferable in terms of optimality and computational efficiency twice as efficient, while preserving the optimality of the solution. Moreover, a method for estimating the 3D positions of tactile sensors was proposed. The method relies on force/torque measurements and it was exploited to calibrate the 1500 tactile sensors mounted on the arms of the iCub robot, with an average error of approximately 7 mm. Another method was introduced, which makes use of the calibrated tactile sensors, together with the distributed force/torque sensors, to estimate an arbitrary number of contact forces acting on any part of the robot's body. The method is based on the Recursive Newton-Euler Algorithm, and it was implemented as part of the open-source C++ library iDyn. Furthermore, a theoretical and empirical analysis investigated how incorrect estimation of contact points may affect the resulting contact forces and induce undesired joint accelerations. Tests on the iCub robot demonstrated a significant improvement in the performance of the force controller when the tactile system was used.

Finally, with the aim of handling both strict and non-strict hierarchies simultaneously, and achieving smooth transitions of priorities, a novel control framework called Generalized Hierarchical Control (GHC) has been developed at UPMC in the work of Liu. The contributions of this work are as follows:

- the development of a generic dynamic control framework, which solves a single quadratic program (QP) to account for an arbitrary number of strict and non-strict task priorities;
- the development of a generalized projector, which ensures desired task priorities, their smooth transitions as well as an elegant way of inserting and deleting tasks among those to be performed. The implementation of such a projector is not restricted to the dynamic control framework. In fact, it can be implemented in many analytical and optimization-based control frameworks. Task priorities can be handled by the modulation of a priority matrix, without the necessity of modifying the control problem formulation each time the priorities change. This projection matrix regulates to what extent a lower-priority

task is projected into the null-space of a higher-priority task. In other words, it allows a task to be completely, partially, or not at all projected into the null-space of some other tasks. The priority levels can be changed by the modulation of the generalized projector.

3 Results

In this section, whole-body control CoDyCo related results obtained both in simulation and on the iCub robot are summarized. The controllers developed by Salini and Del Prete have been ported to C++ during the first months of the CoDyCo project and are now being integrated to the WBI interface developed to provide a common abstraction layer for whole-body controllers and described in Deliverable 1.2. The controller of Liu is coded in C++ within the XDE simulation framework (see Deliverable 1.1) and a quasi-static version has been developed as a C++ Orocos component [65] for the real-time control of the LWR KUKA robot [66].

3.1 Year 1 scenario: balancing on multiple rigid contact points

Year 1 scenario has been validated in simulation using Salini's controller in the XDE simulation framework (see Section 4.7 of Deliverable 5.1). It has also been validated using Liu's controller in simulation using the XDE simulation framework [67]⁸.

On the real robot, Del Petre's controller has been used as a basis for the integration of Year 1 Demo (illustrated on Fig. 1). The proposed balanced controller is described in [68, 69]⁹. Some extension of this work has led to recent developments illustrated in a recently published video¹⁰. A view of the obtained performance is provided on Fig. 2.



Figure 1: Views of iCub balancing on multiple contacts using one of the whole-body controllers developed within the framework of the CoDyCo project.

⁸The article related to the work of Liu *et al.* [67] is provided in Appendix B

⁹The article related to the work of Nori *et al.* [68] is provided in Appendix C. The internship report of Talha Ali Arslan [69] is provided in Appendix D

¹⁰<https://www.youtube.com/watch?v=VrPBSSQEr3A>

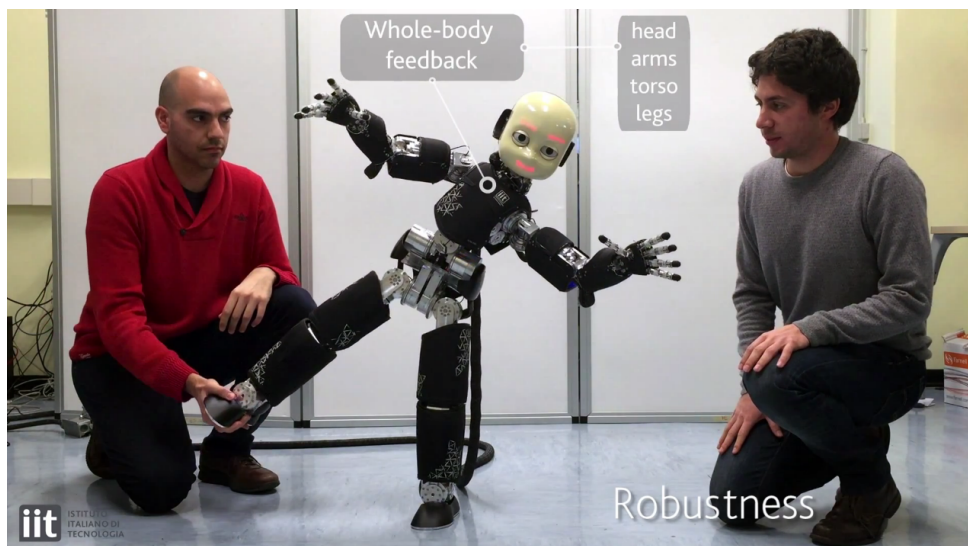


Figure 2: View of the most recent developments in the implementation of the CoDyCo whole-body controller: iCub balancing on one foot while interacting with humans.

3.2 Year 2 scenario: balancing on feet while performing goal directed actions

Year 2 scenario has been validated in simulation using Salini's controller in the XDE simulation framework in several occasions¹¹. More recently, the work of Lober *et al.* aiming at multiple task optimization using dynamical movement primitives for whole-body reactive control builds on Salini's controller [70]. Fig. 3 and a video¹² accompanying the published work illustrate these results.

Fig.4 provides a view of some recent work performed in coordination with Jan Babic from JSI during his period as a visiting professor at UPMC (November 2014). The goal of this on-going cooperation is to reproduce, using a synthetic whole-body controller, results obtained in human reaching with additional contact experiments in Work Package 2.

Year 2 scenario is being currently implemented on the real robot. A description of the scenario and associated controller is provided in Deliverable 5.2.

3.3 Other use case of interest

In the work of Ibanez *et al.* [71] a strategy to automatically combined balance strategies based on continuous postural adjustments and discrete changes in contacts is developed in order to maintain postural stability while considering the engaged walking activity. In order to compute optimal time, duration and position of footsteps along with the center of mass trajectory of a humanoid, a novel mixed-integer model of the system is presented. The introduction of this model in a predictive control problem brings the definition of a Mixed-

¹¹see <http://goo.gl/tsT4Iv> for an overview.

¹²<http://goo.gl/QoSfp7>

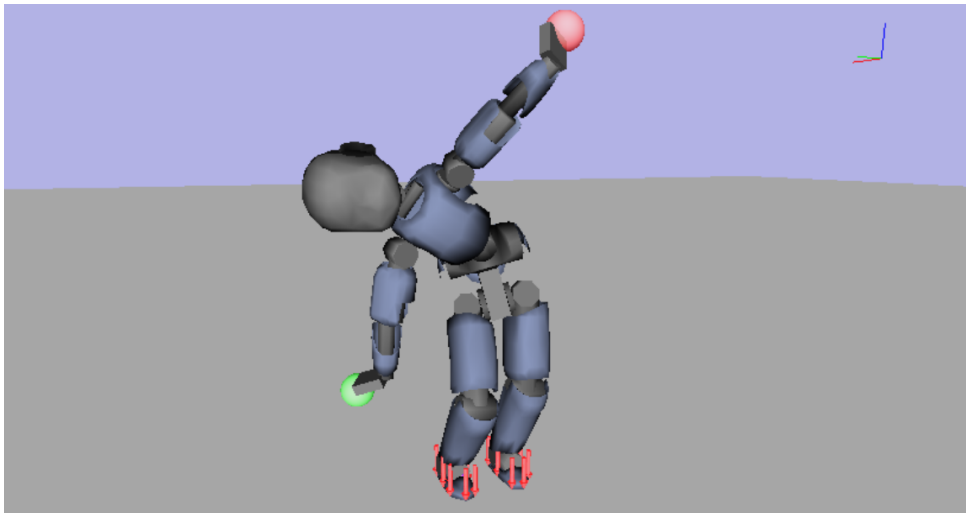


Figure 3: View of some recent developments in the implementation of the CoDyCo whole-body controller in goal oriented actions

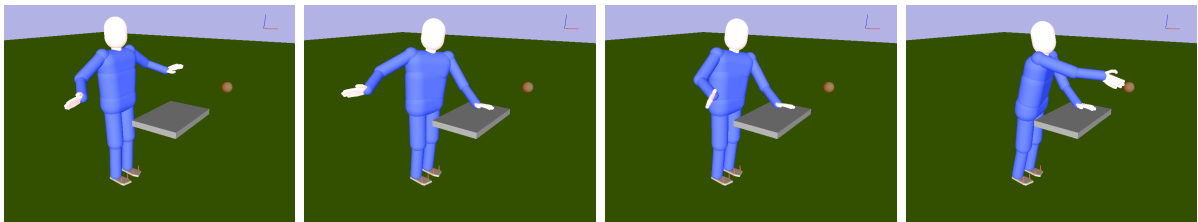


Figure 4: View of a sequence of motion generated using a robotic controller as an attempt to reproduce results obtained in human reaching with additional contact experiments.

Integer Quadratic Program, subject to linear constraints. Simulation results demonstrate the simultaneous adaptation of the gait pattern and posture of the humanoid, in a walking activity under large disturbances, to efficiently compromise between task performance and balance. In addition, a push recovery scenario displays how, using a single balance-performance ratio, distinct behaviors of the humanoid can be specified. Fig. 5 and a video¹³ accompanying the published work illustrate the results. As in [72], this work builds on the local whole-body controller of Salini and provides the global MPC level needed to generate feasible tasks for the robot.

¹³<http://goo.gl/PjcTcS>

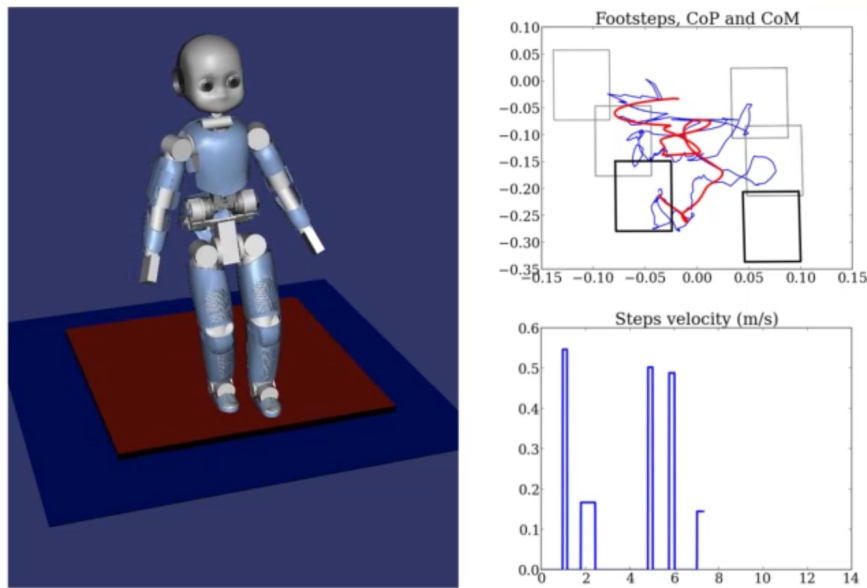


Figure 5: A view of iCub automatically adapting its contact locations based on some perceived balance perturbation.

4 Conclusion

The CoDyCo consortium has a strong expertise in the domain of whole-body control and is well equipped with state-of-the art whole-body controllers. These controllers are the topic of several research results obtained by the consortium and cited in the results Section of this deliverable. Through the WBI software abstraction presented in Deliverable 1.2, they provide appropriate tools based on which the results of all workpackages can be ported to the iCub humanoid robot. This is already the case for the Year 1 and Year 2 demonstration.

References

- [1] S. Kajita and B. Espiau, "Legged robots," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008.
- [2] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3D linear inverted pendulum mode: A simple modeling for a biped walking pattern generation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'01)*, 2001, pp. 239–246.
- [3] S. Lee and A. Goswami, "Reaction mass pendulum (RMP): An explicit model for centroidal angular momentum of humanoid robots," in *IEEE International Conference on Robotics and Automation (ICRA'07)*, 2007, pp. 4667–4672.
- [4] M. Vukobratović and B. Borovac, "Zero-moment point—thirty five years of its life," *International Journal of Humanoid Robotics*, vol. 1, no. 1, pp. 157–173, 2004.
- [5] A. Goswami, "Postural stability of biped robots and the foot-rotation indicator (FRI) point," *The International Journal of Robotics Research*, vol. 18, no. 6, pp. 523–533, 1999.
- [6] P. Wieber, "On the stability of walking systems," in *Proceedings of the International Workshop on Humanoid and Human-Friendly Robotics*, 2002.
- [7] K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Inaba, and H. Inoue, "Online generation of humanoid walking motion based on a fast generation method of motion pattern that follows desired ZMP," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'02)*, 2002, pp. 2684–2689.
- [8] K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa, "An analytical method on real-time gait planning for a humanoid robot," in *IEEE/RAS International Conference on Humanoid Robots (Humanoids'04)*, 2004, pp. 640–655.
- [9] M. Morisawa, K. Harada, S. Kajita, S. Nakaoka, K. Fujiwara, F. Kanehiro, K. Kaneko, and H. Hirukawa, "Experimentation of humanoid walking allowing immediate modification of foot place based on analytical solution," in *IEEE International Conference on Robotics and Automation (ICRA'07)*, 2007, pp. 3989–3994.
- [10] S. Kagami, T. Kitagawa, K. Nishiwaki, T. Sugihara, M. Inaba, and H. Inoue, "A fast dynamically equilibrated walking trajectory generation method of humanoid robot," *Autonomous Robots*, vol. 12, no. 1, pp. 71–82, 2002.
- [11] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *IEEE International Conference on Robotics and Automation (ICRA'03)*, 2003, pp. 1620–1626.

-
- [12] A. Herdt, N. Perrin, and P. Wieber, "LMPC based online generation of more efficient walking motions," in *IEEE/RAS International Conference on Humanoid Robotics (Humanoids'12)*, 2012, pp. 390–395.
- [13] P. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids'06)*, 2006, pp. 137–142.
- [14] —, "Viability and predictive control for safe locomotion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'08)*, 2008, pp. 1103–1108.
- [15] A. Herdt, N. Perrin, and P. Wieber, "Walking without thinking about it," in *IEEE International Conference on Intelligent Robots and Systems (IROS'10)*, 2010, pp. 190–195.
- [16] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.
- [17] K. Hauser, T. Bretl, J.-C. Latombe, K. Harada, and B. Wilcox, "Motion planning for legged robots on varied terrain," *The International Journal of Robotics Research*, vol. 27, no. 11-12, pp. 1325–1349, 2008.
- [18] S. Dalibard, A. El Khoury, F. Lamiriaux, A. Nakhaei, M. Taïx, and J.-P. Laumond, "Dynamic walking and whole-body motion planning for humanoid robots: an integrated approach," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1089–1103, 2013.
- [19] K. Bouyarmane and A. Kheddar, "Humanoid robot locomotion and manipulation step planning," *Advanced Robotics*, vol. 26, no. 10, pp. 1099–1126, 2012.
- [20] S. Lengagne, J. Vaillant, E. Yoshida, and A. Kheddar, "Generation of whole-body optimal dynamic multi-contact motions," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1104–1119, 2013.
- [21] A. Liegeois, "Automatic supervisory control of the configuration and behavior of multi-body mechanisms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 7, no. 12, pp. 868–871, 1977.
- [22] N. Mansard and F. Chaumette, "Task sequencing for high-level sensor-based control," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 60–72, 2007.
- [23] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.
- [24] A. Maciejewski and C. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 109–117, 1985.
- [25] T. Chan and V. D. Dubey, "A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 2, pp. 286–292, 1995.

-
- [26] L. Huo and L. Baron, "The self-adaptation of weights for joint-limits and singularity avoidances of functionally redundant robotic-task," *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 1, pp. 367–376, 2011.
- [27] P. Baerlocher and R. Boulic, "An inverse kinematic architecture enforcing an arbitrary number of strict priority levels," *The Visual Computer: International Journal of Computer Graphics*, vol. 20, no. 6, pp. 402–417, 2004.
- [28] P. Hsu, J. Mauser, and S. Sastry, "Dynamic control of redundant manipulators," *Journal of Robotic Systems*, vol. 6, no. 2, pp. 133–148, 1989. [Online]. Available: <http://dx.doi.org/10.1002/rob.4620060203>
- [29] J. Peters, M. Mistry, F. Udwadia, J. Nakanishi, and S. Schaal, "A unifying framework for robot control with redundant dofs," *Autonomous Robots*, vol. 24, no. 1, pp. 1–12, 2008. [Online]. Available: <http://dx.doi.org/10.1007/s10514-007-9051-x>
- [30] F. Chaumette and É. Marchand, "A redundancy-based iterative approach for avoiding joint limits: Application to visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 5, pp. 719–730, 2001.
- [31] V. Padois, J. Fourquet, and P. Chiron, "Kinematic and dynamic model-based control of wheeled mobile manipulators: A unified framework for reactive approaches," *Robotica*, vol. 25, no. 2, pp. 157–173, 2007.
- [32] D. Raunhardt and R. Boulic, "Progressive clamping," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2007, pp. 4414–4419.
- [33] N. Mansard, O. Khatib, and A. Kheddar, "A unified approach to integrate unilateral constraints in the stack of tasks," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 670–685, 2009.
- [34] M. Mistry, J. Nakanishi, and S. Schaal, "Task space control with prioritization for balance and locomotion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 331–338.
- [35] M. Mistry, J. Nakanishi, G. Cheng, and S. Schaal, "Inverse kinematics with floating base and constraints for full body humanoid robot control," in *8th IEEE-RAS International Conference on Humanoid Robots*, 2008, pp. 22–27.
- [36] L. Sentis and O. Khatib, "Prioritized multi-objective dynamics and control of robots in human environments," in *4th IEEE/RAS International Conference on Humanoid Robots*, vol. 2, 2004, pp. 764–780.
- [37] F. Aghili, "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*, vol. 21, no. 5, pp. 834–849, 2005.

-
- [38] O. Khatib, L. Sentis, and J.-H. Park, "A unified framework for whole-body humanoid robot control with multiple constraints and contacts," in *European Robotics Symposium 2008*, ser. Springer Tracts in Advanced Robotics. Springer Berlin / Heidelberg, 2008, vol. 44, pp. 303–312.
- [39] L. Sentis, J. Park, and O. Khatib, "Compliant control of multi-contact and center of mass behaviors in humanoid robots," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 483–501, june 2010.
- [40] L. Righetti, J. Buchli, M. Mistry, M. Kalakrishnan, and S. Schaal, "Optimal distribution of contact forces with inverse dynamics control," *International Journal of Robotics Research*, no. 3, pp. 280–298, 2013.
- [41] A. Del Prete, "Control of contact forces using whole-body force and tactile sensors: Theory and implementation on the icub humanoid robot," Ph.D. dissertation, University of Genoa and Istituto Italiano di Tecnologia, 2013.
- [42] S. Rubrecht, V. Padois, P. Bidaud, M. de Broissia, and M. Da Silva Simoes, "Motion safety and constraints compatibility for multibody robots," *Autonomous Robots*, vol. 32, no. 3, pp. 333–349, 2012.
- [43] B. Faverjon and P. Tournassoud, "A local based approach for path planning of manipulators with a high number of degrees of freedom," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1987, pp. 1152–1159.
- [44] Y. Abe, M. da Silva, and J. Popović, "Multiobjective control with frictional contacts," in *Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2007, pp. 249–258.
- [45] C. Collette, A. Micaelli, C. Andriot, and P. Lemerle, "Dynamic balance control of humanoids for multiple grasps and non coplanar frictional contacts," in *7th IEEE-RAS International Conference on Humanoid Robots*, 2007, pp. 81–88.
- [46] F. Flacco, A. De Luca, and O. Khatib, "Prioritized multi-task motion control of redundant robots under hard joint constraints," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 3970–3977.
- [47] —, "Motion control of redundant robots under joint constraints: Saturation in the null space," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 2012, pp. 285–292.
- [48] O. Kanoun, F. Lamiraux, P.-B. Wieber, F. Kanehiro, E. Yoshida, and J.-P. Laumond, "Prioritizing linear equality and inequality systems: Application to local motion planning for redundant robots," in *IEEE International Conference on Robotics and Automation*, May 2009, pp. 2939–2944.
- [49] L. Saab, N. Mansard, F. Keith, J.-Y. Fourquet, and P. Soueres, "Generation of dynamic motion for anthropomorphic systems under prioritized equality and inequality constraints,"

- in *IEEE International Conference on Robotics and Automation (ICRA)*, may 2011, pp. 1091–1096.
- [50] L. Saab, O. Ramos, F. Keith, N. Mansard, P. Soueres, and J.-Y. Fourquet, “Dynamic whole-body motion generation under rigid contacts and other unilateral constraints,” *Robotics, IEEE Transactions on*, vol. 29, no. 2, pp. 346–362, 2013.
 - [51] A. Escande, N. Mansard, and P.-B. Wieber, “Hierarchical quadratic programming: Fast online humanoid-robot motion generation,” *International Journal of Robotics Research*, vol. 33, no. 7, pp. 1006–1028, June 2014.
 - [52] A. Herzog, L. Righetti, F. Grimmering, P. Pastor, and S. Schaal, “Balancing experiments on a torque-controlled humanoid with hierarchical inverse dynamics,” in *Proceedings of the IEEE International Conference on Intelligent Robotics Systems*, 2014.
 - [53] M. Liu, A. Micaelli, P. Evrard, A. Escande, and C. Andriot, “Interactive dynamics and balance of a virtual character during manipulation tasks,” in *IEEE International Conference on Robotics and Automation (ICRA)*, may 2011, pp. 1676–1682.
 - [54] J. Salini, V. Padois, and P. Bidaud, “Synthesis of complex humanoid whole-body behavior: A focus on sequencing and tasks transitions,” in *IEEE International Conference on Robotics and Automation*, may 2011, pp. 1283 –1290.
 - [55] K. Bouyarmane and A. Kheddar, “Using a multi-objective controller to synthesize simulated humanoid robot motion with changing contact configurations,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 4414–4419.
 - [56] M. Liu, A. Micaelli, P. Evrard, A. Escande, and C. Andriot, “Interactive virtual humans: A two-level prioritized control framework with wrench bounds,” *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1309–1322, 2012.
 - [57] E. C. Kerrigan and J. M. Maciejowski, “Soft constraints and exact penalty functions in model predictive control,” in *Proceedings of the UKACC International Conference*, Cambridge, UK, September 2000.
 - [58] F. Keith, N. Wieber, P.-B. and Mansard, and A. Kheddar, “Analysis of the discontinuities in prioritized tasks-space control under discreet task scheduling operations,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 3887–3892.
 - [59] T. Petrič and L. Žlajpah, “Smooth continuous transition between tasks on a kinematic control level: Obstacle avoidance as a control problem,” *Robotics and Autonomous Systems*, vol. 61, no. 9, pp. 948–959, 2013.
 - [60] N. Mansard, A. Remazeilles, and F. Chaumette, “Continuity of varying-feature-set control laws,” *Automatic Control, IEEE Transactions on*, vol. 54, no. 11, pp. 2493–2505, 2009.
 - [61] J. Lee, N. Mansard, and J. Park, “Intermediate desired value approach for task transition of robots in kinematic control,” *Robotics, IEEE Transactions on*, vol. 28, no. 6, pp. 1260–1277, 2012.

- [62] J. Salini, "Dynamic control for the task/posture coordination of humanoids: toward synthesis of complex activities," Ph.D. thesis, Université Pierre et Marie Curie, Paris, France, June 2012.
- [63] M. Liu, Y. Tan, and V. Padois, "Generalized hierarchical control," *Submitted to Autonomous Robots*, 2014.
- [64] S. Barthélémy, J. Salini, and A. Micaelli, "Arboris-Python." [Online]. Available: <https://github.com/salini/arboris-python>
- [65] H. Bruyninckx, "Open robot control software: the orocos project," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3, 2001, pp. 2523–2528.
- [66] M. Liu, S. Hak, and V. Padois, "Generalized projector for task priority transitions during hierarchical control," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Seattle, USA, May 2015.
- [67] M. Liu and V. Padois, "Whole-body hierarchical motion and force control for humanoid robots," *Submitted to Autonomous Robots*, 2014.
- [68] F. Nori, S. Traversaro, J. Eljaik, F. Romano, A. Del Prete, and D. Pucci, "icub whole-body control through force egulation on rigid noncoplanar contacts," *Accepted for publication in Frontiers in Robotics and AI*, 2015.
- [69] T. A. Arslan, "Improvement of a balancing force and posture controller with torque minimization," Master's thesis, University of Technology of Eindhoven / Istituto Italiano di Tecnologia, 2014.
- [70] R. Lober, V. Padois, and O. Sigaud, "Multiple task optimization using dynamical movement primitives for whole-body reactive control," in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, Madrid, Spain, Nov. 2014, pp. 193–198.
- [71] A. Ibanez, P. Bidaud, and V. Padois, "Emergence of humanoid walking behaviors from mixed-integer model predictive control," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, USA, Sep. 2014.
- [72] —, "A distributed model predictive control approach for robust postural stability of a humanoid robot," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Hong-Kong, China, Jun. 2014, pp. 202–209.

A Generalized Hierarchical Control

Generalized Hierarchical Control

Mingxing Liu, Yang Tan, and Vincent Padois

Most existing techniques to handle strict task priorities in hierarchical control are based on null-space projectors or a sequence of quadratic programs; whereas non strict task priorities are usually handled by optimization based on a weighting strategy. This paper proposes a novel approach to handle both strict and non-strict priorities of an arbitrary number of tasks, and to achieve multiple priority rearrangements simultaneously. A generalized projector, which makes it possible to completely project a task into the null-space of a set of tasks, while partially projecting it into the null-space of some other tasks, is developed for priority modulation. Priority transitions are achieved by smooth variations of the generalized projector. The control input is computed by solving one quadratic programming problem, where generalized projectors are adopted to maintain a task hierarchy, and constraints can be implemented (e.g. dynamic equilibrium, actuation capabilities, joint limits, obstacle avoidance, contact constraints, etc.). The effectiveness of this approach is demonstrated on a simulated robotic manipulator in a dynamic environment.

Index Terms—Redundant robots, task hierarchy, priority switching, dynamics, torque-based control.

I. INTRODUCTION

Redundant robots, such as humanoids, are nowadays expected to perform complex missions in weakly structured environments (e.g. human environments, construction sites, nuclear dismantling zones, etc.). Even though robot redundancy makes it possible for these robots to perform multiple tasks simultaneously, task conflicts may still occur when all the task objectives cannot be satisfied at the same time. In order to handle conflicts, tasks are usually assigned with different priority levels. Therefore, to control a complex robotic system the controller must be able to handle multiple prioritized tasks simultaneously and to respect various constraints imposed by the robot body (joint limits, actuation capabilities, etc.) and the environment (contacts to maintain, obstacles to avoid, etc.).

A large number of hierarchical control frameworks are presented in the robotics literature for the management of multiple operational task objectives.

- Some of them deal with *strict task hierarchies*, such as analytical methods based on null-space projectors [1]–[5] and hierarchical quadratic programming approaches [6,7]. These approaches can ensure that critical tasks are fulfilled with higher priorities and lower-priority tasks are performed only in the null-space of higher priority tasks.
- Other approaches handle *non-strict task hierarchies*, such as those using weighting strategies [8]–[12]. In a non-strict task hierarchy, a lower priority task is not restricted in the null-space of higher priority tasks, thus it may still affect their performances. The solution of these approaches is a compromise among task objectives of different weights.

In a more general context, the robot may need to deal with both strict and non-strict hierarchies. Moreover, for robots acting in dynamically changing contexts, non-strict priorities between tasks may become strict ones and task priorities may have to be switched in order to cope with changing situations.

With the aim of handling both strict and non-strict hierarchies simultaneously, a novel control framework called

Generalized Hierarchical Control (GHC) is presented in this paper. The contributions of this work are as follows: 1) the development of a generic dynamic control framework, which solves a single quadratic program (QP) to account for an arbitrary number of strict and non-strict task priorities; 2) the development of a generalized projector, which ensures desired task priorities, their transitions as well as an elegant way of inserting and deleting tasks among those to be performed. The implementation of such a projector is not restricted to the dynamic control framework presented in this paper. In fact, it can be implemented in many analytical and optimization-based control frameworks. Task priorities can be handled by the modulation of a priority matrix, without the necessity of modifying the control problem formulation each time the priorities change.

This paper is organized as follows. Related works are described in Section II. The robot model considered in this paper and the tasks and constraints to be handled by the controller are presented in Section III. The GHC framework is developed in Section IV, where detailed explanations of the generalized projector are provided. Some experimental results are presented in Section V to demonstrate the framework capabilities, and comparisons with the results using some other approaches are provided. Several characteristics of this framework and future research directions regarding the computational aspect and the potential application of the proposed approach are presented in Section VII.

II. RELATED WORKS

Approaches to maintain a desired task hierarchy using a multi-objective controller draw a lot of interest. This Section reviews some classical types of hierarchical control frameworks, as well as the methods for priority transitions within these frameworks.

A. Approaches for handling a strict hierarchy

Analytical methods based on null-space projections can ensure that lower priority tasks are executed only in the null-space of higher-priority tasks, by means of the appropriate design of a null-space projector [13]. Such an idea is applied

The authors are with Institut des Systèmes Intelligents et de Robotique, CNRS UMR 7222 & Université Pierre et Marie Curie, 4 place Jussieu, 75252 Paris cedex 05, France. e-mail: {liu, tan, padois}@isir.upmc.fr

in prioritized inverse kinematics [2,14], in acceleration based control [3,4], and in joint torque based control [1,5,15]. A generic framework, from which several existing control laws can be derived, is presented in [16]. Projected inverse dynamics schemes are developed for constrained systems in [17,18], where the dynamics equation is projected into the null-space of the Jacobian of constraint equations.

Inequality constraints are usually difficult to be directly dealt with in analytical approaches using pseudo-inverses and projection matrices. A common method is to transform inequality constraints into task objectives by applying artificial potential fields [19], from which repulsive forces are derived to prevent the robot from entering into activation zones of the inequality constraints [19]–[24]. However, performing these tasks cannot guarantee that these inequality constraints are actually met. The approach presented in [25] integrates unilateral constraints at any priority level, albeit time consuming. The algorithm introduced in [4,26] proposes to disable the most critical joint and redistribute joint motion commands to guarantee the satisfaction of some hard bounds of joint variables. However, this algorithm deals with inequality constraints only at the joint level. Furthermore, the optimal solution satisfying the control problem may require the movement of a joint which has unfortunately been disabled.

To deal with prioritized inequality constraints more easily, hierarchical quadratic programming (HQP) approaches use numerical QP solvers to solve a Hierarchical Quadratic Program [6]. The idea of HQP is to first solve a QP to obtain a solution for a higher priority task objective; and then to solve another QP for a lower priority task, without increasing the obtained minimum of the previous task objective. This prioritization process corresponds to solving lower-priority tasks in the null-space of higher-priority tasks while trying to satisfy lower-priority tasks at best. The HQP algorithm is applied for solving prioritized inverse dynamics [7] and is also applied to whole-body motion control under unilateral constraints [27]. It requires to solve as many QPs as priority levels, which can be quite time consuming. The computation cost of hierarchical inverse kinematics with inequality constraints is improved by an algorithm developed in [28], which permits real time control of a humanoid robot.

Generally, for an approach based on strict hierarchy, the relative importance of one task with respect to another one of different priority level is parametrized in a binary way: either strictly higher or strictly lower. However, in many contexts, organizing tasks by assigning them with strict priorities is not generic, *i.e.* can have some limitations. First, a strict priority is just an extreme case of the relations of task importance levels. In fact, a task may not always have a strict priority over another one and it is usually difficult to define a strict hierarchy among a set of tasks. Second, strict priorities can sometimes be too conservative so that they may completely block lower-priority tasks. Unlike a discrete parametrization of task priorities, a continuous parametrization is richer and more informative. Therefore, this work handles task priorities, which can be strict or non-strict, by using a continuous parametrization. Moreover, priorities are defined here by pairs of tasks: this choice extends the classical notion of priority in Robotics while still making it

possible to represent standard lexicographic orders as defined in [27].

B. Approaches for handling a non-strict hierarchy

Non-strict priorities are usually handled by control approaches using weighting strategies [8]–[11,29]. These control frameworks solve all the constraints and task objectives in one QP and provide a trade-off among task objectives with different importance levels. As the performances of higher priority tasks cannot be guaranteed by simply adjusting the weights of task objectives, a prioritized control framework is proposed in [12] to ensure the performance of a higher-priority task within a user defined tolerance margin. However, this approach handles priorities of only two levels. In approaches based on weighting strategies, task priorities can be parametrized continuously. Nonetheless, even though the work in [30] on soft constraints in model predictive control could probably be adapted to provide a way to reach the extreme case of strict priorities, the existing robotic applications of these frameworks do not extend to strict hierarchies.

The control framework proposed in this paper is based on these frameworks: it formulates and solves all tasks and constraints in one QP. It also largely outperforms them by permitting priorities to change continuously from a non-strict case to a strict case.

C. Task transitions

Earlier versions of analytical methods and HQP approaches can ensure strict priorities among tasks; however, a change in the task set, such as a switch of task priorities, may lead to discontinuity. Recently, different methods have been developed to handle task transition problems. An approach to smooth priority rearrangement between two tasks is proposed in [31,32]. Approaches for continuous and simultaneous transitions of multiple tasks are developed in [33,34]. A specific inverse operator is proposed in [33] to ensure continuous inverse in the analytical computation of control laws. The approach presented in [34] is based on intermediate desired values in the task space. When the number of task transitions increases, this approach suggests to apply an approximation to reduce the computational cost. An approach of hierarchical control with continuous null-space projections is presented in [35]. In this approach, an activator associated to directions in the right singular vectors of a task Jacobian matrix is regulated to activate or deactivate these directions. However, the design of such an activator makes this approach difficult to be implemented for the separate handling of different task directions. On the other hand, task transitions can be easily achieved within a non-strict hierarchy by the continuous variation of task weights [29].

The control framework proposed in this paper allows an arbitrary number of task priority transitions. This framework uses continuous priority parametrization, and the extreme case of strict priorities can be achieved. The idea to achieve this goal is based on the construction of a novel generalized projection matrix, which regulates to what extent a lower-priority task is projected into the null-space of a higher-priority

task. In other words, this generalized projector allows a task to be completely, partially, or not at all projected into the null-space of some other tasks. The priority levels can be changed by the simple modulation of the generalized projector. The implementation of this generalized projector in multi-objective control frameworks based on optimization provide them with a mechanism to regulate task priorities more precisely, so that both strict and non-strict priorities can be handled by solving only one optimization problem.

III. MODELING

Consider a robot as an articulated mechanism with n degrees of freedom (DoF) including n_a actuated DoF. The dynamics of the robot in terms of its generalized coordinates $\mathbf{q} \in \mathbb{R}^n$ is written as follows

$$M(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) = J_c(\mathbf{q})^T \boldsymbol{\chi}, \quad (1)$$

where $M(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is the generalized inertia matrix; $\dot{\mathbf{q}} \in \mathbb{R}^n$ and $\ddot{\mathbf{q}} \in \mathbb{R}^n$ are the vector of velocity and the vector of acceleration in generalized coordinates, respectively; $\mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^n$ is the vector of Coriolis, centrifugal and gravity induced joint torques; $\boldsymbol{\chi} = [\mathbf{w}_c^T \ \boldsymbol{\tau}^T]^T$ is the vector of the actuation torques ($\boldsymbol{\tau} \in \mathbb{R}^{n_a}$) and the external contact wrenches applied to the robot ($\mathbf{w}_c = [\mathbf{w}_{c,1}^T \ \dots \ \mathbf{w}_{c,n_c}^T]^T$), with n_c the number of contact points; $J_c(\mathbf{q})^T = [J_{c,1}(\mathbf{q})^T \ \dots \ J_{c,n_c}(\mathbf{q})^T \ S(\mathbf{q}, \dot{\mathbf{q}})^T]^T$ is the transpose of a Jacobian matrix, with $J_{c,n_\beta}(\mathbf{q})$, the Jacobian matrix associated to a contact point β and $S(\mathbf{q}, \dot{\mathbf{q}})^T \in \mathbb{R}^{n \times n_a}$, a selection matrix for the actuated DoF. In the control problem considered in this paper, the vector $\boldsymbol{\chi}$ is called the **action variable**.

A. Task definition

A task in Robotics can be defined as a function of the considered robotic systems [36,37]. This function relates the control level in operational/task space, to the control level in joint space. More specifically, consider a robot controlled by joint torques at the dynamics level¹, a task i can be defined by the following characteristics:

- A physical frame \mathcal{F}_i , *i.e.* a frame attached to a part of the robot body that should be controlled for performing an operational task.
- An associated **task variable** $\boldsymbol{\xi}_i \in \mathbb{R}^{m_i}$ that can be expressed in terms of some high level goals to be achieved by the frame \mathcal{F}_i in the task space, such as a desired position or orientation. m_i is the dimension of a task i .
- A forward model linearly relating the second order derivative of the vector of generalized coordinates to that of the task variable for a given state $(\mathbf{q}, \dot{\mathbf{q}})$

$$\ddot{\boldsymbol{\xi}}_i = J_i(\mathbf{q})\ddot{\mathbf{q}} + \dot{J}_i(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \quad (2)$$

where $J_i(\mathbf{q})$ is the Jacobian matrix, *i.e.* the differential kinematics mapping from joint space to task space, and $\dot{J}_i(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ is the task space drift vector.

¹The velocity kinematics version of this problem can be trivially derived from this more general case.

- A local controller \mathbf{r}_i , the goal of which is to correct task errors and ensure the convergence of the task variable $\boldsymbol{\xi}_i$ towards its desired trajectory $\boldsymbol{\xi}_i^*$

$$\ddot{\boldsymbol{\xi}}_i^d = \mathbf{r}_i \left(\boldsymbol{\xi}_i, \dot{\boldsymbol{\xi}}_i, \boldsymbol{\xi}_i^*, \dot{\boldsymbol{\xi}}_i^*, \ddot{\boldsymbol{\xi}}_i^* \right). \quad (3)$$

For task motion control, the local controller can take the form of a proportional-integral-derivative controller with a feed-forward term

$$\ddot{\boldsymbol{\xi}}_i^d = \ddot{\boldsymbol{\xi}}_i^* + K_p \mathbf{e} + K_d \dot{\mathbf{e}} + K_i \int \mathbf{e} dt, \quad (4)$$

where \mathbf{e} and $\dot{\mathbf{e}}$ are errors of $\boldsymbol{\xi}_i$ and $\dot{\boldsymbol{\xi}}_i$, respectively; and K_p , K_d and K_i are symmetric, positive definite gain matrices.

For task wrench control, the local controller can take the form of a proportional-integral controller with a feed-forward term

$$\mathbf{w}_i^d = \mathbf{w}_i^* + K_{w,p} \mathbf{e}_w + K_{w,i} \int \mathbf{e}_w dt, \quad (5)$$

where \mathbf{w}_i^* is the desired task wrench, \mathbf{e}_w is the error of task wrench, and $K_{w,p}$ and $K_{w,i}$ are symmetric, positive definite gain matrices. The wrench task can be expressed as a motion task using the inverse of the operational space inertia matrix $\Lambda_i(\mathbf{q}) = [J_i(\mathbf{q})M(\mathbf{q})^{-1}J_i(\mathbf{q})^T]^{-1}$ [38,39]

$$\ddot{\boldsymbol{\xi}}_i^d = \Lambda_i(\mathbf{q})^{-1} \mathbf{w}_i^d \quad (6)$$

which maps the desired task wrench \mathbf{w}_i^d to a desired acceleration $\ddot{\boldsymbol{\xi}}_i^d$ at \mathcal{F}_i .

- A set of relative importance levels with respect to n_t tasks, including task i , characterized by a priority matrix α_i

$$\alpha_i = \text{diag}(\alpha_{i1}I_{m_1}, \dots, \alpha_{ij}I_{m_j}, \dots, \alpha_{in_t}I_{m_{n_t}}) \quad (7)$$

where α_i is a diagonal matrix, the main diagonal blocks of which are square matrices: $\alpha_{ij}I_{m_j}$, I_{m_j} is the $m_j \times m_j$ identity matrix, and $\alpha_{ij} \in [0, 1]$. By convention, the coefficient α_{ij} indicates the priority of task j with respect to task i .

- $\alpha_{ij} = 0$ corresponds to the case where task j has strict lower priority with respect to task i .
- $0 < \alpha_{ij} < 1$ corresponds to a soft (non-strict) priority between the two tasks: the greater the value of α_{ij} , the higher the importance level of task j with respect to task i .
- $\alpha_{ij} = 1$ corresponds to the case where task j has a strict higher priority with respect to task i .

B. Constraint definition

Even though the set of task attributes is specific to each task, the control space to joint space forward mapping is task independent and can be written as

$$\ddot{\mathbf{q}} = M(\mathbf{q})^{-1} (J_c(\mathbf{q})^T \boldsymbol{\chi} - \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}})). \quad (8)$$

The equation of motion (8) constitutes an equality constraint, which relates the joint space acceleration to the action variable for a given state $(\mathbf{q}, \dot{\mathbf{q}})$.

The other constraints considered in this work reflect the physical limitations of the system in terms of:

- actuation capabilities (maximum actuators torques and velocities);
- geometrical limits (joint limits, Cartesian space obstacles);
- contact wrenches (contact existence conditions, bounds on the norms of contact wrenches).

Assuming that approximations such as $\dot{\mathbf{q}}_{k+1} = \dot{\mathbf{q}}_k + \Delta t \ddot{\mathbf{q}}_k$ and $\mathbf{q}_{k+1} = \mathbf{q}_k + \Delta t \dot{\mathbf{q}}_k + \frac{\Delta t^2}{2} \ddot{\mathbf{q}}_k$ hold for one control period Δt , these specific constraints can generally be expressed as a linear inequality of the form

$$G(\mathbf{q}, \dot{\mathbf{q}}) \begin{pmatrix} \ddot{\mathbf{q}} \\ \chi \end{pmatrix} \leq \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) \quad (9)$$

where G and \mathbf{h} are the matrix and vector which express these inequality constraints of physical limitations.

IV. CONTROL PROBLEM FORMULATION

In this work, multiple tasks with different priority levels subject to equality and inequality constraints have to be handled. This kind of multi-objective control problem can be formulated as a Linear Quadratic Programming problem (LQP). This is the approach adopted here, where all the task objectives and constraints are solved simultaneously in one LQP.

This Section first briefly reviews the LQP control framework in IV-A, then develops a generalized projector in IV-B, which is implemented in a LQP-based control framework in IV-C for handling both strict and non-strict priorities, as well as for priority transitions.

A. Control framework based on Linear Quadratic Programming

When only non-strict task hierarchies are considered, weighting strategies, such as those proposed in [9]–[11,29], can be applied to handle the relative importance levels of multiple elementary tasks. In this case, the control problem can be formulated as a Linear Quadratic Programming (LQP) problem as

$$\arg \min_{\ddot{\mathbf{q}}, \chi} \sum_{i=1}^{n_t} \left\| \mathbf{f}_i \left(\ddot{\mathbf{q}}, \ddot{\xi}_i^d \right) \right\|_{Q_i}^2 + \left\| \begin{bmatrix} \ddot{\mathbf{q}} \\ \chi \end{bmatrix} \right\|_{Q_r}^2 \quad (10a)$$

$$\text{subject to} \quad \text{constraints (8), (9)} \quad (10b)$$

where $Q_i = \omega_i I_{m_i}$ is a diagonal weighting matrix to regulate the importance level of task i , $Q_r = \omega_r I_{n+n_a+3n_c}$ is the weighting matrix of the regularization term, ω_i is the weight of each task objective i , and $\mathbf{f}_i \left(\ddot{\mathbf{q}}, \ddot{\xi}_i^d \right) = J_i(\mathbf{q})\ddot{\mathbf{q}} + \dot{J}_i(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \ddot{\xi}_i^d$ is the objective function which measures the error of task i . The task objective functions are minimized to achieve a compromise among all the weighted tasks. The regulation term minimizes the norm of accelerations and action variables. For a redundant robot with many solutions satisfying the same task objective, the regulation term is useful for ensuring the uniqueness of the solution [29]. As this regulation term may

increase task error, its weight value ω_r is usually very small compared to task objective weights. In this optimization problem, $\ddot{\mathbf{q}}$ is an overabundant variable, which can be eliminated by using the equality constraint defined by (8).

B. Projectors for hierarchical control

The control framework based on weighting strategy (10) can qualitatively regulate the relative importance levels of tasks by weighting task objectives, but it cannot ensure strict priorities among tasks. This control framework is extended in this paper, the goal of which is to handle both strict and non-strict task priorities. To achieve this goal, a generalized projector, which can precisely regulate how much a task is affected by other tasks, is developed. In other words, this generalized projector can be regulated to *completely, partially, or not* project a task into the null-space of other tasks.

The following part of this subsection first looks at several forms of projectors, then the analysis of these projectors leads to the development of the generalized projector.

1) Review of existing projectors for hierarchical control

Strict priorities can be handled by analytical methods using a null-space projector N_j defined as

$$N_j = I - J_j^\dagger J_j, \quad (11)$$

where J_j^\dagger is the Moore-Penrose pseudo-inverse of the Jacobian J_j ². The projection of a task i into the null-space of another task j can ensure that task i is satisfied only in the null-space of task j . Such projection-based approaches can ensure that a lower-priority task is performed without producing any motion for a higher-priority task. To handle priorities between one task i and a set of other tasks with higher priorities, task i is projected into the null-space of an augmented Jacobian J of all the higher priority tasks [40,41]

$$J = [J_1^T \dots J_j^T \dots J_{n_t}^T]^T \quad (12)$$

where the augmented Jacobian concatenates the Jacobian matrices of all the n_t tasks.

To achieve smooth priority transitions, the null-space projector (11) is replaced by the following matrix in [31,32]

$$N'_j(\alpha_{ij}) = I - \alpha_{ij} J_j^\dagger J_j, \quad (13)$$

where a scalar parameter $\alpha_{ij} \in [0, 1]$ is used to regulate the priority between two tasks i and j . This matrix leads to smooth transitions of task priorities through the smooth change of the scalar parameter α_{ij} :

- when $\alpha_{ij} = 1$, $J_i N'_j = J_i N_j$, task i is completely projected into the null-space of task j ;
- when $0 < \alpha_{ij} < 1$, task i is partially projected into the null-space of task j ;
- when $\alpha_{ij} = 0$, $J_i N'_j = J_i$, task i is not at all projected into the null-space of task j .

This method can handle priority transitions between only two levels of tasks, and it can hardly be extended to the case of simultaneous transitions among multiple levels of task priorities.

²The dependence to \mathbf{q} is omitted for clarity reasons.

Another projection matrix N'' is proposed in [35] for continuous null-space projections

$$N'' = I - VAV^T \quad (14)$$

with $V \in \mathbb{R}^{n \times n}$ the right singular vectors of J and $A \in \mathbb{R}^{n \times n}$ a diagonal activation matrix. The j -th diagonal element of A , a_{jj} , refers to the j -th column vector in V :

- when $a_{jj} = 1$, the j -th direction in V is activated in N'' ;
- when $0 < a_{jj} < 1$, the j -th direction in V is partially deactivated in N'' ;
- when $a_{jj} = 0$, the j -th direction in V is deactivated in N'' .

As mentioned in [35], for any one-dimensional task j ($J_j \in \mathbb{R}^{1 \times n}$), the matrix (14) becomes

$$N_j'' = I - a_{j,j} \frac{J_j^T}{\|J_j\|} \frac{J_j}{\|J_j\|}, \quad (15)$$

which can be applied to achieve smooth activation or deactivation of task j direction in the projection matrix by the smooth variation of the scalar a_{jj} . When extended to tasks of m directions ($J \in \mathbb{R}^{m \times n}$), this method allows us to apply the same transition to all the m directions of J , but its application for achieving the separate regulation of each task direction is not easy. This is because generally, each activator a_{jj} is directly referred to the j -th direction in the right singular vectors V of J , but not directly referred to a specific direction in J .

2) Generalized projector

In order to achieve variations of multiple task priorities simultaneously among an arbitrary number of tasks, and to be able to ensure strict priorities, an approach to the computation of a novel projector is developed here. Similar to the form of the matrix N'' in the case of considering a one-dimensional task (15), the form of this novel projector is obtained without the necessity of the computation of pseudo-inverse matrices. Its computation is based on orthonormal basis computation, and it is simple to implement this novel projector. Moreover, the new projector allows us to regulate the activation of each task directions in a more intuitive way, by regulating the priority matrix α that is more closely related to task directions than the activator A in (14).

Consider a hierarchy of n_t tasks, the joint space acceleration \ddot{q}_i^* for achieving each task i should be modified to account for the hierarchy information contained in α_i . The idea is to achieve this goal by a generalized projector $P_i(\alpha_i) \in \mathbb{R}^{n \times n}$, which projects the joint acceleration ($P_i(\alpha_i)\ddot{q}_i^*$) to satisfy the desired hierarchy.

In order to compute the generalized projector $P_i(\alpha_i)$, a preliminary processing of the matrices J and α_i is carried out according to the priorities of the n_t tasks with respect to task i . As each row of J is associated to α_{ij} , the rows of J can be sorted in descending order with respect to the values of the diagonal elements in α_i . The resulting matrix J_{s_i} is thus constructed so that tasks which should be the least influenced by task i appear in its first rows, while tasks which can be the most influenced by task i appear in its last rows. The values in α_i are sorted accordingly, leading to $\alpha_{i,r}^s$, the diagonal elements

of which are organized in descending order starting from the first row.

Based on J_{s_i} , a projector into the null space of J can be computed. This can be done by first computing a matrix $B_i(J_{s_i}) \in \mathbb{R}^{r \times n}$, where $r = \text{rank}(J_{s_i})$ is the rank of J_{s_i} . The rows of $B_i(J_{s_i})$ form an orthonormal basis of the joint space obtained using elementary row transformations on J_{s_i} . Then this projector can be computed as $P_i' = I_n - B_i^T B_i$. When performing task i by using the projected acceleration $P_i' \ddot{q}_i^*$, the projector P_i' basically cancels any acceleration that impacts all the n_t tasks, including task i itself.

The computation of the projector P_i' can be modified such that tasks having strict priority over task i are perfectly accounted for; tasks over which task i has a strict priority are not considered; and all other tasks with soft priorities are accounted for, according to the value of their respective priority parameters in α_i . Inspired by how the matrix N'' is computed in (15), the generalized projector is given by

$$P_i(\alpha_i) = I_n - B_i(J_{s_i})^T \alpha_{i,r}^s(\alpha_i, \text{origin}) B_i(J_{s_i}), \quad (16)$$

where $\alpha_{i,r}^s$ is a diagonal matrix of degree r . The vector **origin** $\in \mathbb{R}^r$ is a vector of the row indexes of J_{s_i} selected during the construction of the orthonormal basis B_i . Each of these r rows in J_{s_i} is linearly independent to all the previously selected ones. The diagonal elements of $\alpha_{i,r}^s$ are restricted to the r diagonal elements of α_i^s , which correspond to the r rows of J_{s_i} , the row indexes of which belong to **origin**.

Algorithm (1) and (2) summarize the construction of the generalized projector $P_i(\alpha_i)$. As any numerical scheme, tolerances are used for numerical comparison, such as ϵ , which is defined as the smallest value greater than zero in line #11 of Algorithm (2).

Algorithm 1: Generalized projector computation - task i

Data: α_i, J
Result: P_i

```

1 begin
2    $n \leftarrow \text{GetNbCol}(J)$ 
3    $\text{index} \leftarrow \text{GetRowIndexDescOrder}(\alpha_i)$ 
4    $\alpha_i^s \leftarrow \text{SortRows}(\alpha_i, \text{index})$ 
5    $J_{s_i} \leftarrow \text{SortRows}(J, \text{index})$ 
6    $B_i, \text{origin}, r \leftarrow \text{GetOrthBasis}(J_{s_i}) \triangleright \text{Alg. (2)}$ 
7    $\alpha_{i,r}^s \leftarrow \text{GetSubDiagMatrix}(\alpha_i^s, \text{origin})$ 
8    $P_i \leftarrow I_n - B_i^T \alpha_{i,r}^s B_i$ 
9 return  $P_i$ 
```

Note that by varying the value of each α_{ij} in α_i , one can regulate the priority of each task j in the n_t tasks with respect to task i separately.

3) Task insertion and deletion

There is a particular case induced by the proposed formulation and corresponding to the influence of task i on itself. Even though not intuitive, this self-influence has to be interpreted in terms of task existence, modulated by α_{ii} . If $\alpha_{ii} = 1$ then task i is projected into its own null-space, i.e. it is basically canceled out. Decreasing α_{ii} continuously to 0 is a simple and elegant way to introduce the task in the set of tasks. Conversely, increasing α_{ii} continuously from 0 to 1 provides

Algorithm 2: Orthonormal basis computation - *GetOrthBasis(A)*

Data: A, ϵ
Result: $B, origin, r$
1 begin
2 $n \leftarrow \text{GetNbCol}(A)$
3 $m \leftarrow \text{GetNbRow}(A)$
4 $i \leftarrow 0$
5 **for** $k \leftarrow 0$ **to** $m - 1$ **do**
6 **if** $i \geq n$ **then**
7 **break**
8 $B[i, :] \leftarrow A[k, :]$
9 **for** $j \leftarrow 0$ **to** $i - 1$ **do**
10 $B[i, :] \leftarrow B[i, :] - (B[i, :]B[j, :]^T) B[j, :]$
11 **if** $\text{norm}(B[i, :]) > \epsilon$ **then**
12 $B[i, :] \leftarrow B[i, :] / \text{norm}(B[i, :])$
13 $origin[i] \leftarrow k$
14 $i \leftarrow i + 1$
15 $r \leftarrow i$
16 return $B, origin, r$

with a proper task deletion procedure. When being added or suppressed, the influence of task i with respect to other tasks also has to be defined but here again this can be done in a continuous manner.

C. Generalized hierarchical control framework

The control problem that solves one task i , while taking into account the constraints as well as the influence of a set of other tasks over it, can be written as follows

$$\arg \min_{\ddot{\mathbf{q}}, \chi} \left\| \mathbf{f}_i(\ddot{\mathbf{q}}, \ddot{\xi}_i^d) \right\|^2 + \left\| \begin{bmatrix} \ddot{\mathbf{q}} \\ \chi \end{bmatrix} \right\|_{Q_r}^2 \quad (17a)$$

subject to

$$J_c(\mathbf{q})^T \chi = M(\mathbf{q})P_i(\alpha_i)\ddot{\mathbf{q}} + \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) \quad (17b)$$

$$G(\mathbf{q}, \dot{\mathbf{q}}) \begin{pmatrix} P_i(\alpha_i)\ddot{\mathbf{q}} \\ \chi \end{pmatrix} \leq \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) \quad (17c)$$

where the generalized projector defined by (16) is applied in the constraints to handle task priorities. Here, the task objective weighting matrix Q_i is omitted, as it is set to the identity matrix; and the matrix $Q_r = \omega_r I_{n+n_a+3n_c}$ is set to a diagonal matrix with the weight value ω_r being very small compared to 1.

Now consider the control problem for solving n_t tasks. A joint acceleration variable $\ddot{\mathbf{q}}'_i$ is associated to each task i , such that the overall joint space acceleration accounting for the sets of relative importance parameters $(\alpha_1, \dots, \alpha_{n_t})$ is given by

$$\ddot{\mathbf{q}} = \sum_{i=1}^{n_t} P_i(\alpha_i)\ddot{\mathbf{q}}'_i. \quad (18)$$

The GHC framework solves the LQP problem formulated

as

$$\arg \min_{\ddot{\mathbf{q}}', \chi} \sum_{i=1}^{n_t} \left\| \mathbf{f}_i(\ddot{\mathbf{q}}'_i, \ddot{\xi}_i^d) \right\|^2 + \left\| \begin{bmatrix} \ddot{\mathbf{q}}' \\ \chi \end{bmatrix} \right\|_{Q_r}^2 \quad (19a)$$

subject to

$$J_c(\mathbf{q})^T \chi = M(\mathbf{q})P\ddot{\mathbf{q}}' + \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) \quad (19b)$$

$$G(\mathbf{q}, \dot{\mathbf{q}}) \begin{pmatrix} P\ddot{\mathbf{q}}' \\ \chi \end{pmatrix} \leq \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}), \quad (19c)$$

with $\ddot{\mathbf{q}}' = \begin{bmatrix} \ddot{\mathbf{q}}'_1 \\ \vdots \\ \ddot{\mathbf{q}}'_{n_t} \end{bmatrix}$ and $P = [P_1(\alpha_1) \dots P_{n_t}(\alpha_{n_t})]$.

This optimization problem minimizes the objective function of each task as well as the magnitude of the control input, subject to a set of linear constraints. By solving this optimization problem, the solution of joint accelerations and the action variable χ can be obtained. The solution of joint torques is extracted from the optimal value of χ . The overall joint space acceleration $\ddot{\mathbf{q}}$ is optimized to achieve all the tasks according to the set of their relative importance $(\alpha_1, \dots, \alpha_{n_t})$. Especially, it is proved in Appendix A that this GHC framework can handle strict task hierarchies represented by standard lexicographic orders.

This control approach is robust to both kinematic and algorithmic singularities. In the GHC framework based on LQP formulation, tasks are expressed in a forward way and most LQP solvers do not require the explicit inversion of task Jacobian matrices. Therefore, the GHC framework does not have problems of numerical singularities due to kinematic singularities. Moreover, unlike approaches using the pseudo-inverse of projected Jacobians $(J_i N_j)$, which requires special treatment for handling algorithmic singularities when the projected Jacobian drops rank [42], the GHC framework does not necessitate the inversion of projected Jacobians. Therefore, the framework does not have to handle such kind of algorithmic singularities.

V. RESULTS

The proposed GHC framework (19) is applied to the control of a 7-DoF Kuka LWR robot. The experiments are conducted in the Arboris-Python simulator [43], which is a rigid multi-body dynamics and contacts simulator written in Python. The LQP problem is solved by a QP solver included in CasADi-Python [44], which is a symbolic framework for dynamic optimization.

In the experiments, three tasks are defined: task 1 for the control of the three dimensional position of the end-effector, task 2 for the three dimensional position of the elbow, and task 3 for the posture. Any wrench task is transformed into a motion task by applying (6). Targets of the three tasks are not compatible with one another. The elbow task target is a fixed target position and the posture task target is a fixed posture during all the experiments.

The GHC framework (19) is applied, with $n_t = 3$. For each task i , an optimization variable $\ddot{\mathbf{q}}'_i \in \mathbb{R}^7$ is defined. A local controller (4) is used to ensure the convergence of each task variable towards its target. More precisely, a

proportional-derivative controller $\ddot{\xi}_i^d$ is applied for each task. When a task target is static, $\ddot{\xi}_i^d = K_p e_i + K_d \dot{e}_i$ with $K_p = 30s^{-2}$ and $K_d = 20s^{-1}$. When tracking a desired trajectory $\ddot{\xi}_i^*$, $\ddot{\xi}_i^d = \ddot{\xi}_i^* + K_p e_i + K_d \dot{e}_i$ with $K_p = 100s^{-2}$ and $K_d = 20s^{-1}$. The priority parameter matrices associated with the three tasks are: $\alpha_1 = \text{diag}(\alpha_{11}I_3, \alpha_{12}I_3, \alpha_{13}I_7)$, $\alpha_2 = \text{diag}(\alpha_{21}I_3, \alpha_{22}I_3, \alpha_{23}I_7)$, $\alpha_3 = \text{diag}(\alpha_{31}I_3, \alpha_{32}I_3, \alpha_{33}I_7)$. The regularization weight Q_r is chosen as 0.01.

In the rest of the paper, the notation $i \triangleright j$ indicates that task i has a strict higher priority over task j , and the notation \Rightarrow stands for a transition of hierarchy setting. The following function is used for the smooth variation of an α_{ij} (conversely α_{ji}) from 0 to 1 during the transition time period $([t_1, t_2])$

$$\alpha_{ij}(t) = 0.5 - 0.5 \cos\left(\frac{t - t_1}{t_2 - t_1} \pi\right), \text{ with } t \in [t_1, t_2], \quad (20)$$

$$\alpha_{ji}(t) = 1 - \alpha_{ij}(t).$$

A. Priority switching subject to constraints

This experiment is carried out to demonstrate that GHC allows handling task priorities subject to a variety of constraints. All the task targets are static. An obstacle plane is inserted between the initial position of the end-effector and its target position (see Fig. 1). The robot should avoid penetration into the obstacle while performing tasks. A threshold value of $0.02m$ is chosen as the minimum authorized distance between the end-effector and the obstacle plane.

The optimization variables are \dot{q}'_1 , \dot{q}'_2 , \dot{q}'_3 , and τ . The inequality constraints (19c) are:

- bounds on joint velocities (bjv), with $G_{bjv} = \begin{bmatrix} I_7 & 0_7 \\ -I_7 & 0_7 \end{bmatrix}$ and $h_{bjv} = \begin{bmatrix} (\bar{q} - \dot{q})/\Delta t \\ (-\underline{q} + \dot{q})/\Delta t \end{bmatrix}$, where $(\bar{\cdot})$ and $(\underline{\cdot})$ denote the upper and lower bounds, respectively;
- bounds on joint torques (bjt), with $G_{bjt} = \begin{bmatrix} 0_7 & I_7 \\ 0_7 & -I_7 \end{bmatrix}$ and $h_{bjt} = \begin{bmatrix} \bar{\tau} \\ -\underline{\tau} \end{bmatrix}$;
- and obstacle avoidance (obs), with $G_{obs} = [n_{obs}^T J_1 \ 0_7]$ and $h_{obs} = (\bar{v}_1(d_{obs}) - n_{obs}^T J_1 \dot{q})/\Delta t$, where n_{obs} is the unit normal vector pointing from the end-effector to the obstacle plane, J_1 is the Jacobian of the end-effector task, and \bar{v}_1 is the bound on the end-effector velocity towards the plane, which depends on the distance (d_{obs}).

For the sake of clarity, joint limits and obstacle avoidance between the bodies of the robot (other than the end-effector) and the environment are not considered in this example.

At the beginning, the tasks, in a priority level decreasing order, are the elbow task, the end-effector task, and the posture task. Then the end-effector task priority increases and becomes the task with the highest priority. Afterwards, the priorities of the posture task and the elbow task are switched. Then the priorities of the posture task and the end-effector task are switched. At the end, the posture task becomes the task with the highest priority. The evolution of the task hierarchy is defined as: $2 \triangleright 1 \triangleright 3 \Rightarrow 1 \triangleright 2 \triangleright 3 \Rightarrow 1 \triangleright 3 \triangleright 2 \Rightarrow 3 \triangleright 1 \triangleright 2$.

The task errors are presented in Fig. 2. The desired priority switches are successfully performed and desired task priorities

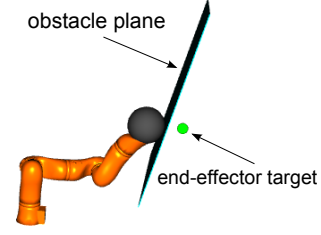


Fig. 1: The end-effector moves towards its target position while avoiding penetration into the obstacle plane.

are well maintained. The resulting end-effector trajectory and the measured distance between the end-effector and the obstacle are presented in Fig. 3. The resulting joint velocities and joint torques are shown in Figures 4 and 5, respectively. It can be seen from these figures that joint velocity bounds and joint torques bounds are respected. Also the end-effector does not penetrate into the obstacle while trying to move towards its target position. The results of this experiment illustrate the fact that GHC can maintain desired task priorities while satisfying all these constraints. A video of this experiment is attached to this paper.

B. Contact force control

In this experiment, the end-effector is expected to move towards a plane, and then to apply a desired contact force against the plane in the vertical direction (see Fig. 6). Before the establishment of the contact with the plane, the end-effector task is a motion task with its task target located on the surface of the plane. Once the end-effector contacts the plane, the end-effector task is a composition of a position task in the horizontal plane and a force task in the vertical direction. The end-effector starts from an initial position, which is above its target position and pointing upwards, then it moves towards the target and starts to apply a contact force to the plane.

The evolution of task hierarchy is $2 \triangleright 1 \triangleright 3 \Rightarrow 1 \triangleright 2 \triangleright 3$. At the beginning of this experiment, the elbow task has the highest priority, then the priorities between the elbow task and the end-effector task switches. The change of α , the positions errors, and the actual and desired contact forces are shown in Fig. 7.

When the end-effector task becomes the task with the highest priority, the end-effector position error is small and the generated contact force follows the references of the contact force, except for when the contact is established between the two rigid bodies. This result illustrates the fact that the highest priority task of the end-effector is maintained after the application of the contact force.

C. Empirical comparison with other approaches

In this Section, the GHC approach is compared with other approaches dedicated to hierarchical control subject to inequality constraints.

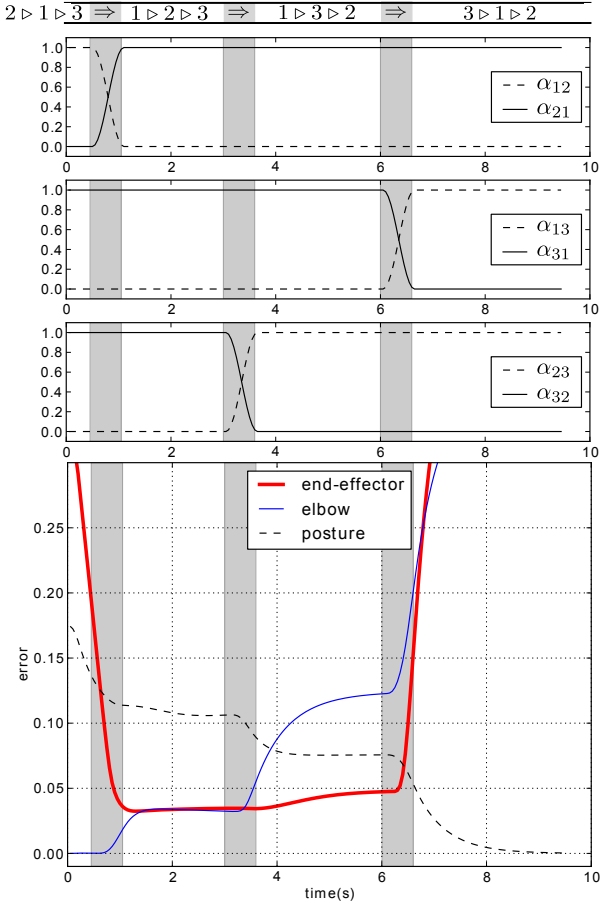


Fig. 2: Evolution of α s (top) and task errors (bottom) during priority switching subject to constraints. Task priorities are switched continuously with the continuous change of α s. The end-effector task error is not decreased to 0 when its task priority is the highest: the obstacle avoidance constraint is respected, and the end-effector cannot arrive at its target position behind the obstacle plane.

1) Comparison with HQP

In this experiment, GHC is compared with the HQP approach [45]. Task hierarchy is changed four times (see Fig. 8) and joint velocity and joint torques bounds are imposed. The evolution of the task hierarchy is $3 > 2 > 1 \Rightarrow 1 > 2 > 3 \Rightarrow 2 > 1 > 3 \Rightarrow 1 > 3 \Rightarrow 1 > 2 > 3$. In the beginning, the tasks, in the priority level decreasing order, are the posture task, the elbow task, and the end-effector task. Then the end-effector task priority increases and the posture task priority decreases simultaneously. During the second priority switching, the priorities of the end-effector task and the elbow task are switched. Then the elbow task is removed. Finally, the elbow task is inserted between the end-effector task and the posture task.

The experiment is carried out first using fixed task targets, then using a desired end-effector trajectory with a lemniscate shape.

The results corresponding to the use of fixed task targets are presented in Fig. 9 to 11. Task errors by using GHC and HQP are shown in Figures 9 and 10, respectively. Fig. 11 shows

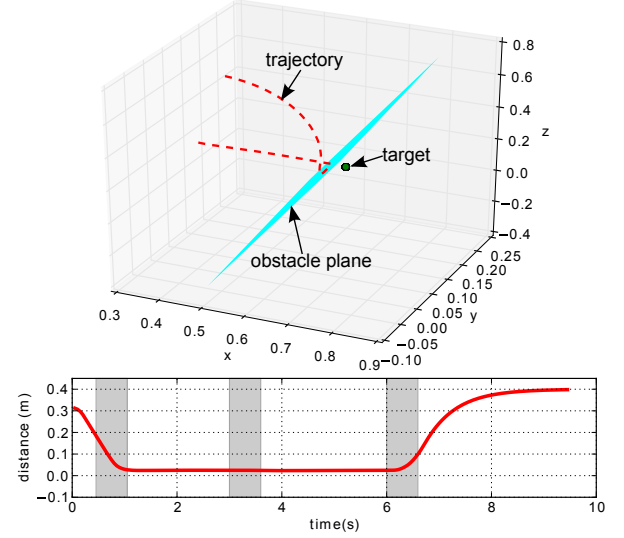


Fig. 3: The resulting end-effector trajectory (top) and the distance between the end-effector and the obstacle (bottom). The end-effector stops moving toward the obstacle plane when its distance to the obstacle decreases to the threshold value of $0.02m$.

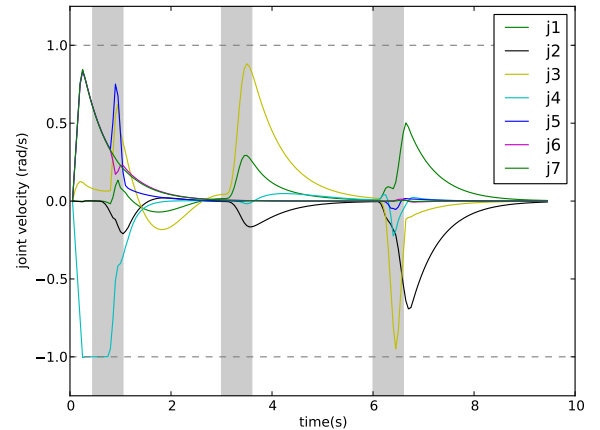


Fig. 4: Evolution of the joint velocities. The upper and lower bounds of \dot{q} are 1 rad/s and -1 rad/s , respectively. These bounds are voluntarily set low in order to easily illustrate the fact that they are respected.

the integration of the absolute value of each resulting joint jerk $\left(\int_0^t \left| \frac{d^3 q}{dt^3} \right| dt \right)$ using these two approaches. Steady state task errors for each task hierarchy configuration are shown in Table I, where the results using GHC and HQP are included.

When a lemniscate-shaped end-effector trajectory is used, the end-effector task is to move along this lemniscate orbit periodically, with an orbital period of $2\pi s$. The desired and the resulting end-effector trajectory is shown in Fig. 12. In this case, the results of task errors and the integration of the absolute values of joint jerks are presented in Fig. 13 and 14, respectively. A video of this experiment that presents the main

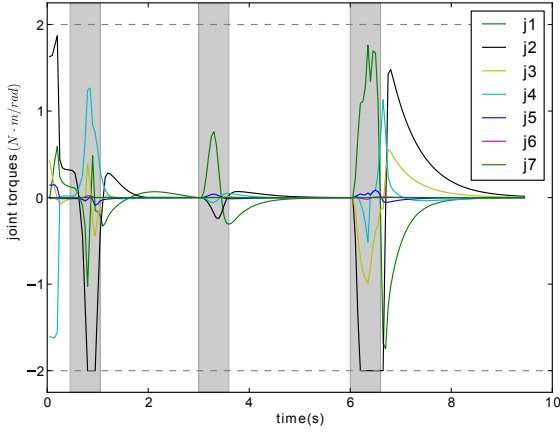


Fig. 5: The resulting joint torques. The upper and lower bounds of τ are $2 \text{ N} \cdot \text{m/rad}$ and $-2 \text{ N} \cdot \text{m/rad}$, respectively. These bounds are voluntarily set low in order to easily illustrate the fact that they are respected.

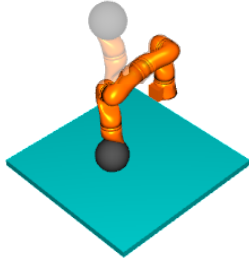


Fig. 6: The target position for the end-effector is on the plane. The end-effector starts from an initial position, which is above the target position and pointing upwards, then it moves towards the target and starts to apply a contact force to the plane.

TABLE I: Steady state task errors for each task hierarchy configuration

priority	3 ▷ 2 ▷ 1		
task	1	2	3
GHC	0.46	0.40	2.2e-30
HQP	0.46	0.40	2.8e-10
priority	1 ▷ 2 ▷ 3		
task	1	2	3
GHC	1.0e-6	0.46	1.8
HQP	4.5e-7	0.46	1.8
priority	2 ▷ 1 ▷ 3		
task	1	2	3
GHC	0.42	2.6e-6	3.0
HQP	0.42	2.7e-6	3.1
priority	1 ▷ 3		
task	1	2	3
GHC	3.9e-6	0.55	0.79
HQP	4.5e-6	0.55	0.79

features of GHC (priority transitions, the insertion and deletion of tasks) is attached to this paper.

GHC provides similar results in terms of task errors compared with HQP, as can be observed in Fig. 9, 10, and 13. The

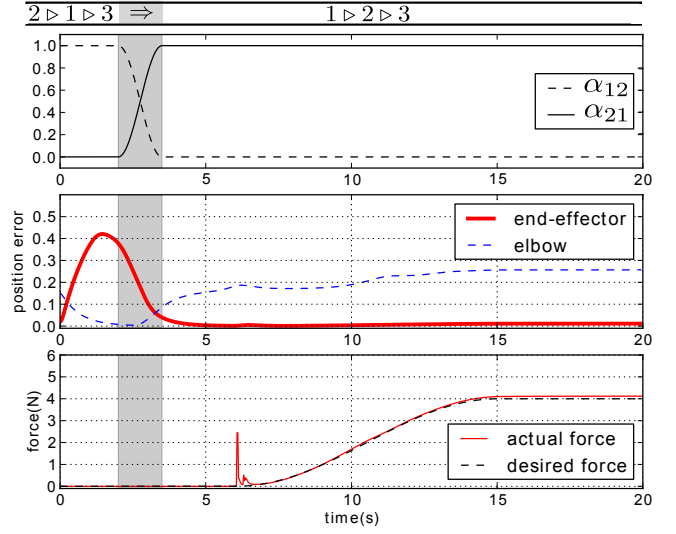


Fig. 7: Results of contact force control. The top figure shows the change of α s. Task 1 is the end-effector task and task 2 is the elbow task. The figure in the middle shows the end-effector position error in the horizontal plane as well as the elbow position error in 3-d. The end-effector starts from an initial position above the target position, then it moves towards the target and starts to apply a contact force to the plane. The bottom figure represents the actual and desired contact forces between the end-effector and the plane.

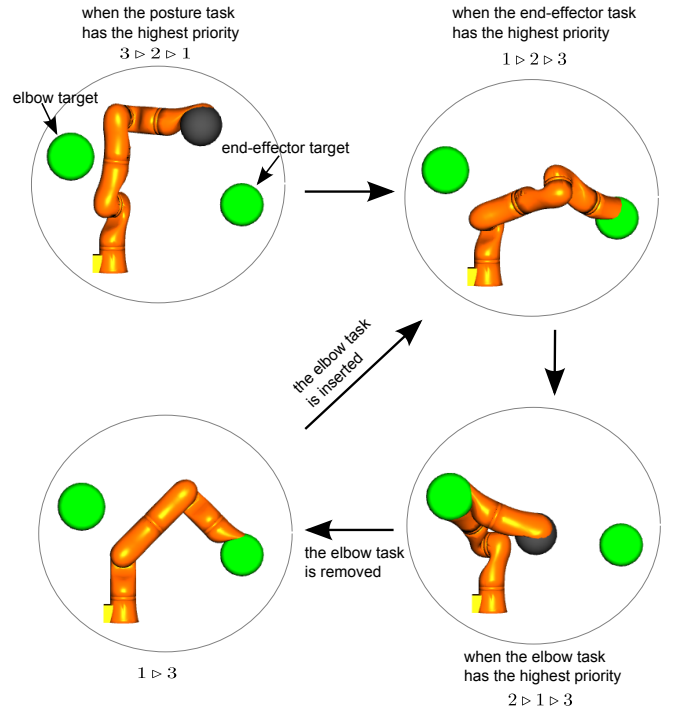


Fig. 8: Experiment of priority switching for the comparison of the HQP and GHC approaches.

results of task errors in Table I show that both GHC and HQP can ensure strict priority. When controlled by GHC and HQP, errors of the task with the highest priority are very small.

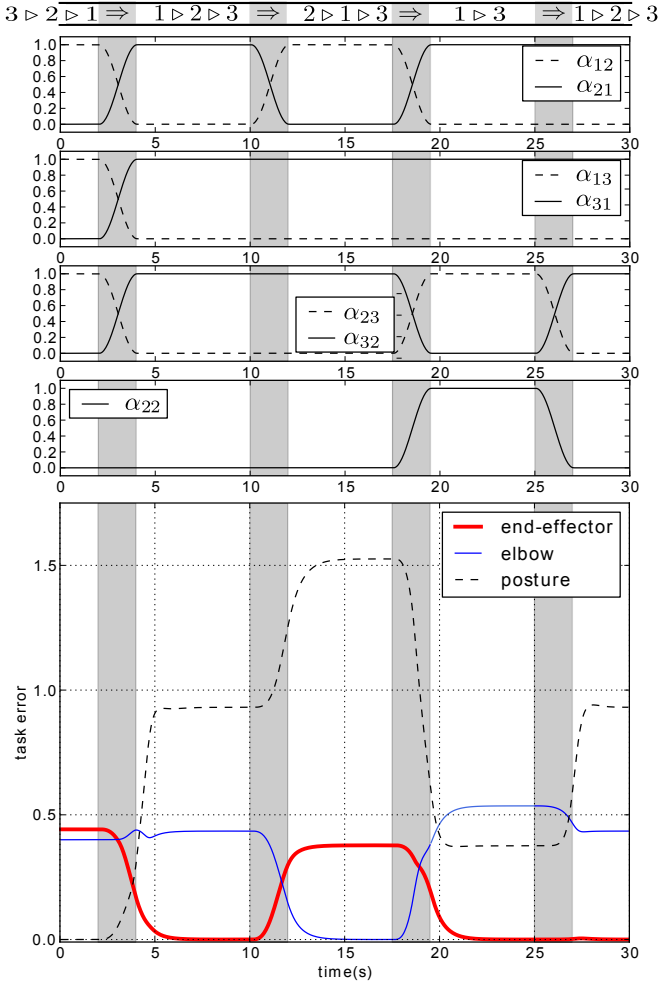


Fig. 9: Evolution of α s (top) and task errors (bottom) using GHC, with fixed task targets. Priority transitions as well as the insertion and deletion of the elbow task are performed. Strict priorities are well respected and the error of the highest priority task is maintained at 0 during steady states.

2) Comparison with a non-strict hierarchy strategy

The evolution of the task hierarchy in this experiment is $1 \triangleright 2 \triangleright 3 \Rightarrow 2 \triangleright 1 \triangleright 3$. The priorities of the end-effector task and the elbow task are switched once. During priority switching, the task objective weights of the end-effector task (w_1) and the elbow task (w_2) are changed smoothly when the weighting strategy is applied, and the priority parameters α_{12} and α_{21} are changed smoothly when GHC is applied. When the weighting strategy is used, two pairs of the weights ($w_1 = 1, w_2 = 0.1$) and ($w_1 = 1, w_2 = 0.001$) are applied.

Fig. 15 presents how task errors change with the priority parameters using GHC, as well as how they change with different task objective weights using the weighting strategy described by (10). It can be seen in this figure that a continuous change of corresponding values of α s can generate similar variations of task errors as a continuous change of task weights does. The priority of the end-effector task decreases gradually with respect to the elbow task, either by the continuous decrease of the weight of the end-effector (w_1) and the

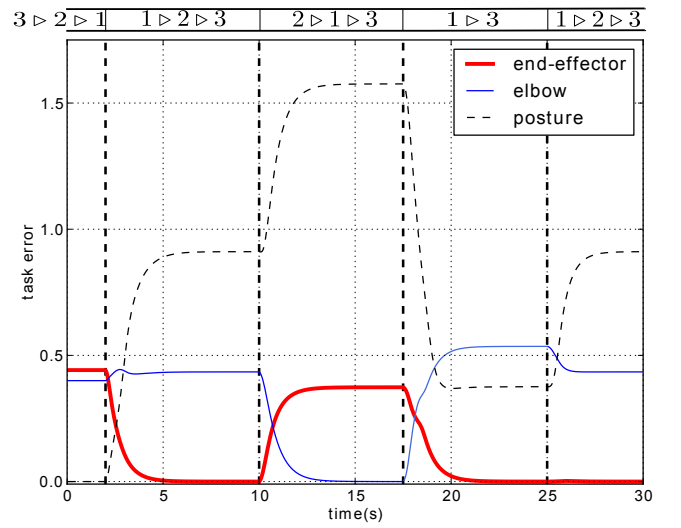


Fig. 10: Task errors using HQP, with fixed task targets. HQP provides similar results in terms of task errors as GHC (Fig. 9).

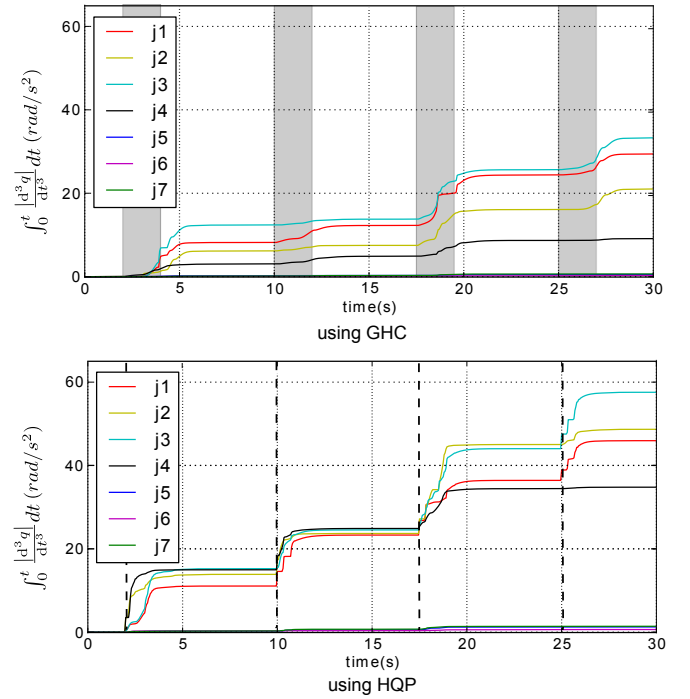


Fig. 11: Integration of the absolute values of joint jerks using GHC (top) and HQP (bottom), with fixed task targets. GHC generates smaller joint jerks than HQP does, while the latter one provides larger jerks each time task hierarchy is changed.

increase of the weight of the elbow task (w_2), or by the continuous increase of α_{21} and decrease of α_{12} . Moreover, the larger the difference between the maximum and the minimum values of task weights are, the closer the task performances are to those generated by GHC. This is because an increase of the difference between task weights makes non-strict task hierarchies evolve towards the extreme case of strict task

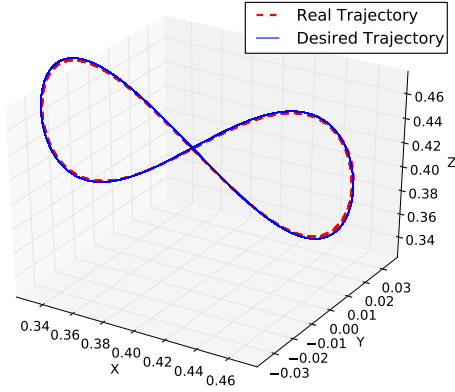


Fig. 12: The desired and the resulting end-effector trajectory provided by GHC, when the end-effector task has the highest priority. The end-effector moves along the lemniscate-shaped trajectory with an orbital period of $2\pi s$.

hierarchies. However, if a large number of importance levels has to be handled, then a huge difference between the weight of the highest priority task and the one of the lowest priority task has to be used. On the contrary, when GHC is applied, strict priorities can be easily achieved by setting the relevant α_{ij} to its limit values 0 or 1.

VI. DISCUSSION

In this section, the computation cost and the continuity aspects of this approach are discussed.

A. Computation time

For a robot of n DoFs performing k priority levels of tasks with a total dimension of m , the computation cost by using the HQP solver [46] is dominated by the hierarchical complete orthogonal decomposition, whose cost is equivalent to $n^2m + nm^2 + \sum_{i=1}^k (m_i - r_i)m_i^2$, with m_i and r_i being respectively the dimension of tasks and the rank of task jacobian in the i -th hierarchy. By using the GHC strategy, the magnitude order of optimization variables is kn , since a joint acceleration variable $\ddot{\mathbf{q}}'_i \in \mathbb{R}^n$ is associated to each task i . In this case, one level of QP (19) needs to be solved, so the computation cost is in $O((kn)^2m + knm^2 + (m - r)m^2)$, with r being the rank of the augmented task jacobian.

The computational cost of the current GHC strategy is sensitive to the number of DoFs of the robot and the number of tasks. For a fixed-based KUKA robot with 7 DoFs performing n_1 motion tasks of different priority levels, a set of joint acceleration variables $\ddot{\mathbf{q}}' \in \mathbb{R}^{7n_1}$ and the joint torques $\tau \in \mathbb{R}^7$ needs to be solved for. For a fixed-based humanoid robot iCub with 32 DoFs performing n_2 tasks, the number of variables would be $32(n_2 + 1)$. Fig. 16 shows the computation time of using GHC to solve randomly selected hierarchical control problems for the KUKA robot and the iCub robot performing

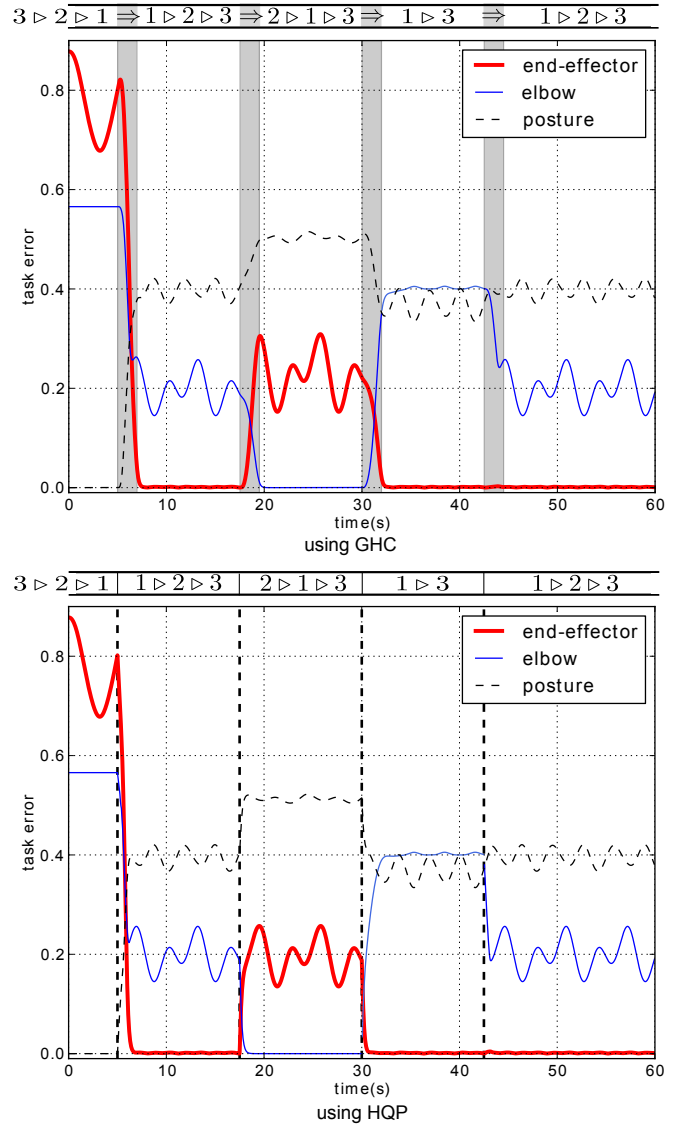


Fig. 13: Task errors using GHC (top) and HQP (bottom), with the end-effector tracking a lemniscate-shaped trajectory. Both approaches achieve desired priority transitions as well as the insertion and deletion of the elbow task, and both of them can maintain strict priorities.

different numbers of tasks. Each control problem consists of the constraint (19b), a posture task with random joint goal positions, and a set of 3-dimensional Cartesian motion tasks with random goal positions. For the KUKA robot performing totally 5 tasks, the mean computation time per iteration is 2.7 ms; for the iCub robot performing the same number of tasks, the mean computation time is 88ms. These results correspond to a C++ implementation of the controller on a standard Linux PC.

B. Continuity

It can be seen in Fig. 11 and 14 that GHC generates smaller joint jerks than HQP does, which implies that GHC provides smoother priority transitions. Basically, the solution

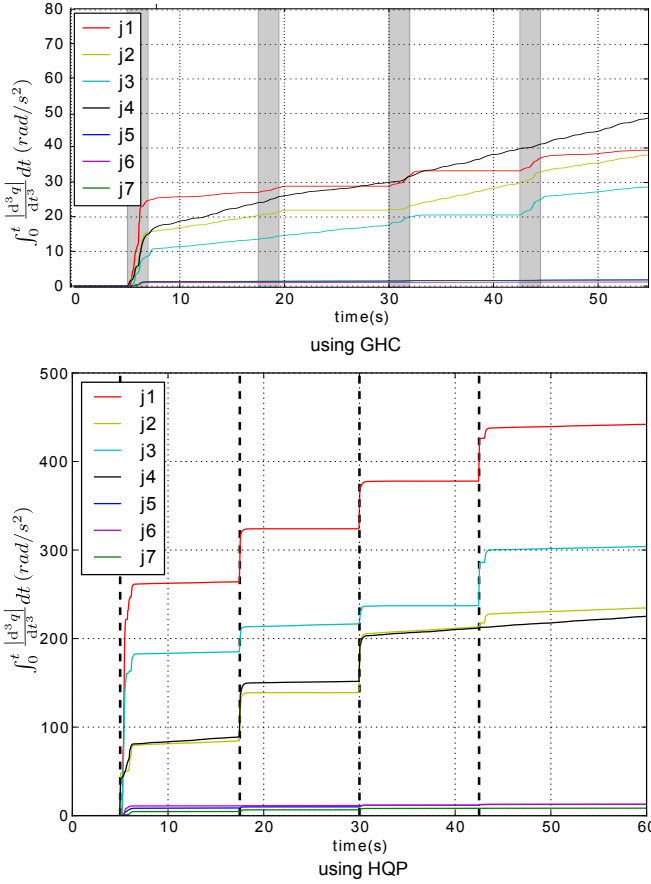


Fig. 14: Integration of the absolute values of joint jerks using GHC (top) and HQP (bottom), with the end-effector tracking a lemniscate-shaped trajectory. GHC generates smaller joint jerks than HQP does, while the latter one provides larger jerks each time task hierarchy is changed.

of GHC is continuous, even during hierarchy rearrangements, if the vector *origin* in (16) remains the same before and after the rearrangements. Indeed, in this case, the basis B_i used to compute the generalized projector varies continuously with J_i , and the generalized projector varies continuously with B_i and α_i . However, similarly to the HQP algorithm, discontinuity may still occur during the switch of priorities or during the insertion and deletion of tasks. In GHC, such a discontinuity is due to the change of the basis B_i during hierarchy rearrangements.

VII. CONCLUSIONS AND FUTURE WORKS

This paper proposes a novel and unifying generalized hierarchical control approach for handling multiple tasks with strict and soft priorities. A generalized projector is developed. It can precisely regulate how much a task can influence or be influenced by other tasks through the modulation of a priority matrix: a task can be completely, partially, or not at all projected into the null-space of other tasks. Multiple simultaneous changes of task priorities can be achieved by using this generalized projector and, using the same mechanism, tasks can be easily inserted or deleted. Moreover,

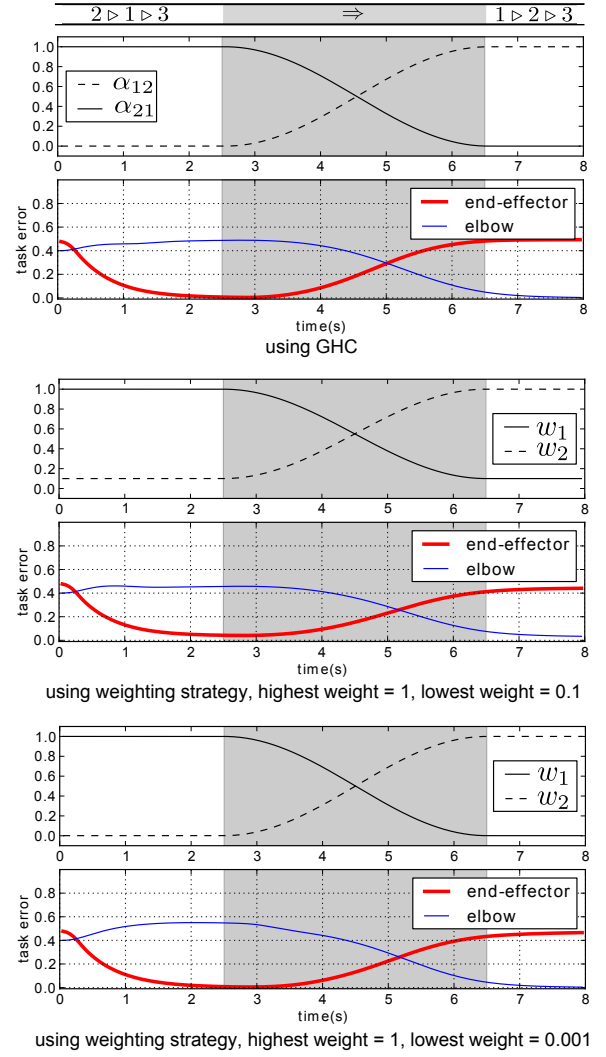


Fig. 15: Evolution of task errors with respect to the evolution of α s using GHC (top) and with respect to different weights by using the weighting strategy (middle and below).

the GHC approach can maintain and switch task priorities while respecting a set of equality and inequality constraints.

Several experiments are conducted to demonstrate that GHC allows task insertion and deletion, as well as the handling of task priorities subject to constraints. Both motion and contact force tasks can be handled by GHC. These experiments emphasize several characteristics of this approach:

- 1) Priorities among tasks can be maintained by applying the generalized projectors. Through the modulation of the priority matrices $\alpha_1, \dots, \alpha_{n_t}$ (and consequently of the associated generalized projectors), GHC can behave as a controller that takes into account a strict hierarchy (by setting some $\alpha_{ij} = 0$ or 1) and as a controller that uses a weighting strategy (by setting some $\alpha_{ij} \in]0, 1[$). In other words, the controller can be configured to control simultaneously tasks assigned with strict priorities, as well as tasks with different weights (non-strict priorities).

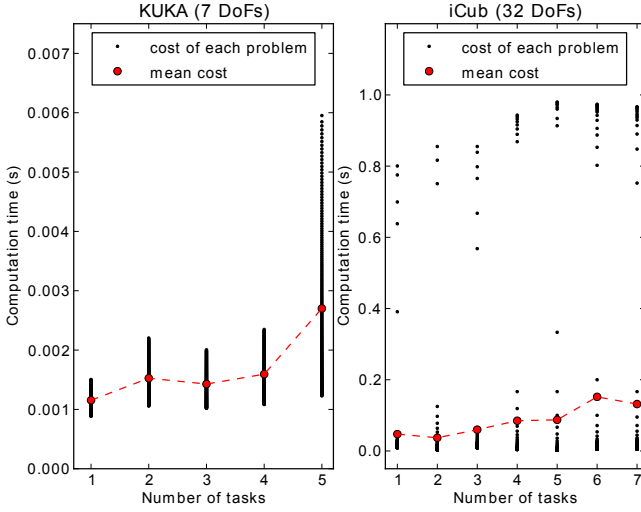


Fig. 16: Computation time per iteration when using GHC to solve randomly selected hierarchical control problems for a fixed-based KUKA robot and a fixed-based iCub robot. Each control problem consists of a posture task and a set of 3D Cartesian motion tasks (0 to 4 motion tasks for KUKA and 0 to 6 motion tasks for iCub), subject to the whole-body equilibrium constraint (19b). The computation time tends to increase with the number of DoFs of the robot and the number of tasks.

- 2) Simultaneous rearrangements of multiple task priorities can be achieved easily by the variations of relevant entries in the generalized projectors associated to these tasks.

In this work, the GHC approach is illustrated at the dynamic level; however, the generalized projector introduced here is not restricted to this case. In fact, it can also be used in other types of controllers, such as a velocity kinematics controller. The basic idea is to associate each task with a task variable in joint space (\dot{q}'_i , \ddot{q}'_i , τ'_i , etc.), then to apply generalized projectors to these task variables, and finally the global joint space variable is the sum of each projected task variables ($P_i(\alpha_i)\dot{q}'_i$, $P_i(\alpha_i)\ddot{q}'_i$, $P_i(\alpha_i)\tau'_i$, etc.).

Immediate future work includes the reduction of the computational cost of GHC to achieve real-time control of complex robots with a high number of DoF. The continuity problem also clearly remains an open problem to tackle in future work. Finally, the use of robot learning techniques to incrementally learn and improve the tuning of the relative influence of each task with respect to others is also of great interest. Finally,

APPENDIX A

PROOF OF THE MAINTENANCE OF STRICT HIERARCHIES REPRESENTED BY STANDARD LEXICOGRAPHIC ORDERS SUBJECT TO CONSTRAINTS

This section proves that the proposed GHC approach (19) can maintain strict task hierarchies represented by standard lexicographic orders while accounting for linear constraints.

Suppose there are n_t tasks that should be organized in a way such that each task i has a strict lower priority than task $i - 1$ with $i = 2, \dots, n_t$. In this case, the generalized projector P_i of a task i is in fact a null-space projector, which projects a task Jacobian into the null-space of all the previous $i - 1$ tasks, and each α_i is an identity matrix. The dependence of P_i to α_i is omitted in this proof for clarity reasons. Let each task objective function be $f_i = J_i x'_i - x_i^d$, with x'_i being a joint space task variable. Moreover, the global variable $x = \sum_i P_i x'_i$ should satisfy linear equality or inequality constraints $Ax \leq b$.

At the first stage, the regulation term is neglected, and the optimization problem can be written as follows

$$\begin{aligned} \arg \min_{x'_{(n_t)}} & \sum_{i=1}^{n_t} \|J_i x'_i - x_i^d\|^2 \\ \text{subject to } & A \sum_{i=1}^{n_t} P_i x'_i \leq b \end{aligned} \quad (21)$$

where $x'_{(n_t)} = \{x'_1, x'_2, \dots, x'_{n_t}\}$, and the solution to (21) is denoted as $x^*_{(n_t)} = \{x^*_1, x^*_2, \dots, x^*_{n_t}\}$.

Consider the case of $n_t = 1$, then the optimization problem can be written as

$$\begin{aligned} \arg \min_{x'_{(1)}} & \|J_1 x'_{(1)} - x_1^d\|^2 \\ \text{subject to } & Ax'_{(1)} \leq b. \end{aligned} \quad (22)$$

The solution to this problem $x^*_{(1)}$ is the same as the one to the problem formulated by HQP.

When $n_t = k$, then the optimization problem is formulated as

$$\begin{aligned} \arg \min_{x'_{(k)}} & \sum_{i=1}^k \|J_i x'_i - x_i^d\|^2 \\ \text{subject to } & A \sum_{i=1}^k P_i x'_i \leq b. \end{aligned} \quad (23)$$

Suppose the solution $x^*_{(k)}$ can maintain the strict task hierarchy: if a task $k + 1$ is inserted with lowest priority with respect to the set of k tasks, then the optimization problem with the $k + 1$ tasks can be written as

$$\begin{aligned} \arg \min_{x'_{(k+1)}} & \sum_{i=1}^k \|J_i x'_i - x_i^d\|^2 + \|J_{k+1} x'_{k+1} - x_{k+1}^d\|^2 \\ \text{subject to } & A \left(\sum_{i=1}^k P_i x'_i + P_{k+1} x'_{k+1} \right) \leq b. \end{aligned} \quad (24)$$

As $P_k P_{k+1} = P_{k+1}$, the term $\sum_{i=1}^k P_i x'_i + P_{k+1} x'_{k+1}$ in the constraint in (24) is equivalent to $\sum_{i=1}^{k-1} P_i x'_i + P_k x'_k$, with

$$\varsigma_k = x'_k + P_{k+1} x'_{k+1}. \quad (25)$$

Then problem (24) can be written as

$$\begin{aligned} \arg \min_{\mathbf{x}'_{(k)}, \varsigma_k, \mathbf{x}_{k+1}} & \sum_{i=1}^{k-1} \|J_i \mathbf{x}'_i - \mathbf{x}_i^d\|^2 + \|J_k \varsigma_k - \mathbf{x}_k^d\|^2 + \\ & \|J_{k+1} \mathbf{x}'_{k+1} - \mathbf{x}_{k+1}^d\|^2 \\ \text{subject to } & A \left(\sum_{i=1}^{k-1} P_i \mathbf{x}'_i + P_k \varsigma_k \right) \leq \mathbf{b} \\ & \varsigma_k = \mathbf{x}'_k + P_{k+1} \mathbf{x}'_{k+1}. \end{aligned} \quad (26)$$

\mathbf{x}'_k in (26) is a free variable, and this problem can be separated into two sub-problems. The first sub-problem is

$$\begin{aligned} \arg \min_{\mathbf{x}'_{(k-1)}, \varsigma_{k-1}} & \sum_{i=1}^{k-1} \|J_i \mathbf{x}'_i - \mathbf{x}_i^d\|^2 + \|J_k \varsigma_k - \mathbf{x}_k^d\|^2 \\ \text{subject to } & A \left(\sum_{i=1}^{k-1} P_i \mathbf{x}'_i + P_k \varsigma_k \right) \leq \mathbf{b}. \end{aligned} \quad (27)$$

The optimal solution $\sum_{i=1}^{k-1} P_i \mathbf{x}_i^{*,\prime} + P_k \varsigma_k^*$ to this problem is equivalent to the one of (23). Indeed, these two solutions have the same effect on task k

$$J_k \sum_{i=1}^k P_i \mathbf{x}_i^{*,\prime} = J_k \left(\sum_{i=1}^{k-1} P_i \mathbf{x}_i^{*,\prime} + P_k \varsigma_k^* \right). \quad (28)$$

To prove (28), one needs to notice that $J_i P_j = \mathbf{0}$ with $j \geq i$. The second sub-problem is given by

$$\arg \min_{\mathbf{x}_{k+1}} \|J_{k+1} \mathbf{x}'_{k+1} - \mathbf{x}_{k+1}^d\|^2. \quad (29)$$

Therefore, the insertion of a lower priority task $k+1$ does not change the optima of the k previous task objectives. In other words, the strict task hierarchy of an arbitrary number of tasks subject to linear constraints can be maintained.

We have proved that each lower priority task will not increase the obtained optima of all the previous tasks. The rest of this proof explains the roles of the regulation term. As mentioned in Section IV-A, the use of a regulation term, which minimizes the norm of each task variable, helps to ensure the uniqueness of the solution. As each task objective i is assigned with the weight $\omega_i = 1$, which is much greater than the weight of the regulation term ($\omega_r \ll 1$), the task variables are optimized to mainly satisfy task objectives. Moreover, in GHC, this regulation term also helps to improve the performance of lower priority tasks. Consider $k+1$ levels of tasks to handle, as $J_i P_j = \mathbf{0}$ with $j \geq i$, the final solution is $\sum_{i=1}^k P_i \mathbf{x}_i^* + P_{k+1} \mathbf{x}_{k+1}^*$. Denoting the elements required by task i as $\mathbf{x}_i^{i,*}$ and the rest elements that are not effectively handled by task objective i as $\mathbf{x}_i^{f,*}$, the final solution can be rewritten as $S = \sum_{i=1}^k P_i^i \mathbf{x}_i^{i,*} + \sum_{i=1}^k P_i^f \mathbf{x}_i^{f,*} + P_{k+1} \mathbf{x}_{k+1}^*$, with P_i^i and P_i^f the columns in P_i that correspond to $\mathbf{x}_i^{i,*}$ and $\mathbf{x}_i^{f,*}$ respectively. The term $\sum_{i=1}^k P_i^f \mathbf{x}_i^{f,*}$ that is not required by the k previous tasks may contribute to task $k+1$ and affect its task performance. The minimization of the norm of \mathbf{x}_i^f in the

regulation term improves the performance of task $k+1$ by making S closer to $\sum_{i=1}^k P_i^i \mathbf{x}_i^{i,*} + P_{k+1} \mathbf{x}_{k+1}^*$, where $P_i^i \mathbf{x}_i^{i,*}$ are used to perform the k previous tasks and $P_{k+1} \mathbf{x}_{k+1}^*$ is used to perform the $(k+1)$ -th task in the null-space of all the higher priority tasks.

ACKNOWLEDGMENT

This work was partially supported by the European Commission, within the CoDyCo project (FP7-ICT-2011-9, No. 600716) and by the RTE company through the RTE/UPMC chair Robotics Systems for field intervention in constrained environments held by Vincent Padois.

REFERENCES

- [1] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *Robotics and Automation, IEEE Journal of*, vol. 3, no. 1, pp. 43–53, 1987.
- [2] M. Mistry, J. Nakanishi, and S. Schaal, "Task space control with prioritization for balance and locomotion," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, 2007, pp. 331–338.
- [3] P. Hsu, J. Mauser, and S. Sastry, "Dynamic control of redundant manipulators," *Journal of Robotic Systems*, vol. 6, no. 2, pp. 133–148, 1989. [Online]. Available: <http://dx.doi.org/10.1002/rob.4620060203>
- [4] F. Flacco, A. De Luca, and O. Khatib, "Prioritized multi-task motion control of redundant robots under hard joint constraints," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 3970–3977.
- [5] L. Sentis and O. Khatib, "Prioritized multi-objective dynamics and control of robots in human environments," in *4th IEEE/RAS International Conference on Humanoid Robots*, vol. 2, 2004, pp. 764–780 Vol. 2.
- [6] O. Kanoun, F. Lamiroux, P.-B. Wieber, F. Kanehiro, E. Yoshida, and J.-P. Laumond, "Prioritizing linear equality and inequality systems: Application to local motion planning for redundant robots," in *IEEE International Conference on Robotics and Automation (2009)*, may 2009, pp. 2939–2944.
- [7] L. Saab, N. Mansard, F. Keith, J.-Y. Fourquet, and P. Soueres, "Generation of dynamic motion for anthropomorphic systems under prioritized equality and inequality constraints," in *IEEE International Conference on Robotics and Automation (ICRA)*, may 2011, pp. 1091–1096.
- [8] Y. Abe, M. da Silva, and J. Popović, "Multiobjective control with frictional contacts," in *Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2007, pp. 249–258.
- [9] C. Collette, A. Micaelli, C. Andriot, and P. Lemerle, "Dynamic balance control of humanoid for multiple grasps and non coplanar frictional contacts," in *7th IEEE-RAS International Conference on Humanoid Robots*, 2007, pp. 81–88.
- [10] M. Liu, A. Micaelli, P. Evrard, A. Escande, and C. Andriot, "Interactive dynamics and balance of a virtual character during manipulation tasks," in *IEEE International Conference on Robotics and Automation (ICRA)*, may 2011, pp. 1676–1682.
- [11] K. Bouyarmane and A. Kheddar, "Using a multi-objective controller to synthesize simulated humanoid robot motion with changing contact configurations," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, 2011, pp. 4414–4419.
- [12] M. Liu, A. Micaelli, P. Evrard, A. Escande, and C. Andriot, "Interactive virtual humans: A two-level prioritized control framework with wrench bounds," *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1309–1322, 2012.
- [13] A. Liégeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 7, no. 12, pp. 868–871, dec. 1977.
- [14] M. Mistry, J. Nakanishi, G. Cheng, and S. Schaal, "Inverse kinematics with floating base and constraints for full body humanoid robot control," in *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on*, 2008, pp. 22–27.
- [15] L. Sentis, J. Park, and O. Khatib, "Compliant control of multi-contact and center of mass behaviors in humanoid robots," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 483–501, june 2010.

- [16] J. Peters, M. Mistry, F. Udwadia, J. Nakanishi, and S. Schaal, "A unifying framework for robot control with redundant dofs," *Autonomous Robots*, vol. 24, no. 1, pp. 1–12, 2008. [Online]. Available: <http://dx.doi.org/10.1007/s10514-007-9051-x>
- [17] F. Aghili, "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*, vol. 21, no. 5, pp. 834–849, 2005.
- [18] O. Khatib, L. Sentis, and J.-H. Park, "A unified framework for whole-body humanoid robot control with multiple constraints and contacts," in *European Robotics Symposium 2008*, ser. Springer Tracts in Advanced Robotics. Springer Berlin / Heidelberg, 2008, vol. 44, pp. 303–312.
- [19] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The international journal of robotics research*, vol. 5, no. 1, pp. 90–98, 1986.
- [20] L. Sentis and O. Khatib, "Task-oriented control of humanoid robots through prioritization," in *IEEE RAS/RSJ International Conference on Humanoid Robots*, 2004.
- [21] —, "Synthesis of whole-body behaviors through hierarchical control of behavioral primitives," *International Journal of Humanoid Robotics*, vol. 02, no. 04, pp. 505–518, 2005.
- [22] O. Stasse, A. Escande, N. Mansard, S. Miossec, P. Evrard, and A. Kheddar, "Real-time (self)-collision avoidance task on a hrp-2 humanoid robot," in *IEEE International Conference on Robotics and Automation (ICRA)*, may 2008, pp. 3200–3205.
- [23] V. Padois, J.-Y. Fourquet, P. Chiron *et al.*, "Kinematic and dynamic model-based control of wheeled mobile manipulators: A unified framework for reactive approaches," *Robotica*, vol. 25, no. 2, p. 157, 2007.
- [24] L. Saab, P. Soueres, and J.-Y. Fourquet, "Coupling manipulation and locomotion tasks for a humanoid robot," in *International Conference on Advances in Computational Tools for Engineering Applications (ACTEA)*, july 2009, pp. 84–89.
- [25] N. Mansard, O. Khatib, and A. Kheddar, "A unified approach to integrate unilateral constraints in the stack of tasks," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 670–685, june 2009.
- [26] F. Flacco, A. De Luca, and O. Khatib, "Motion control of redundant robots under joint constraints: Saturation in the null space," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 2012, pp. 285–292.
- [27] L. Saab, O. Ramos, F. Keith, N. Mansard, P. Soueres, and J.-Y. Fourquet, "Dynamic whole-body motion generation under rigid contacts and other unilateral constraints," *Robotics, IEEE Transactions on*, vol. 29, no. 2, pp. 346–362, 2013.
- [28] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *The International Journal of Robotics Research*, p. 0278364914521306, 2014.
- [29] J. Salini, V. Padois, and P. Bidaud, "Synthesis of complex humanoid whole-body behavior: A focus on sequencing and tasks transitions," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, may 2011, pp. 1283–1290.
- [30] K. E. C. and M. J. M., "Soft constraints and exact penalty functions in model predictive control," in *Proceedings of the UKACC International Conference*, Cambridge, UK, September 2000.
- [31] F. Keith, N. Wieber, P.-B. and Mansard, and A. Kheddar, "Analysis of the discontinuities in prioritized tasks-space control under discrete task scheduling operations," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 3887–3892.
- [32] T. Petrič and L. Žlajpah, "Smooth continuous transition between tasks on a kinematic control level: Obstacle avoidance as a control problem," *Robotics and Autonomous Systems*, vol. 61, no. 9, pp. 948–959, 2013.
- [33] N. Mansard, A. Remazeilles, and F. Chaumette, "Continuity of varying-feature-set control laws," *Automatic Control, IEEE Transactions on*, vol. 54, no. 11, pp. 2493–2505, 2009.
- [34] J. Lee, N. Mansard, and J. Park, "Intermediate desired value approach for task transition of robots in kinematic control," *Robotics, IEEE Transactions on*, vol. 28, no. 6, pp. 1260–1277, 2012.
- [35] A. Dietrich, A. Albu-Schaffer, and G. Hirzinger, "On continuous null space projections for torque-based, hierarchical, multi-objective manipulation," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, May 2012, pp. 2978–2985.
- [36] O. Khatib, L. Sentis, J. Park, and J. Warren, "Whole-body dynamic behavior and control of human-like robots," *International Journal of Humanoid Robotics*, vol. 1, no. 1, pp. 29–43, 2004.
- [37] C. Samson, M. L. Borgne, and B. Espiau, *Robot control: the Task Function Approach*. Oxford, United Kingdom: Addison-Wesley Longman Publishing Co., Inc., 1991.
- [38] O. Khatib, "Inertial properties in robotic manipulation: An object-level framework," *The International Journal of Robotics Research*, vol. 14, no. 1, pp. 19–36, 1995.
- [39] K.-S. Chang and O. Khatib, "Efficient algorithm for extended operational space inertia matrix," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1. IEEE, 1999, pp. 350–355.
- [40] B. Siciliano and J.-J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, june 1991, pp. 1211–1216 vol.2.
- [41] P. Baerlocher and R. Boulic, "Task-priority formulations for the kinematic control of highly redundant articulated structures," in *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, vol. 1, 1998, pp. 323–329 vol.1.
- [42] H. Sadeghian, L. Villani, M. Keshmiri, and B. Siciliano, "Dynamic multi-priority control in redundant robotic systems," *Robotica*, vol. 31, pp. 1155–1167, 10 2013.
- [43] J. Salini, "Dynamic control for the task/posture coordination of humanoids: toward synthesis of complex activities," Ph.D. thesis, Université Pierre et Marie Curie, Paris, France, June 2012.
- [44] J. Andersson, J. Kesson, and M. Diehl, "Casadi - a symbolic package for automatic differentiation and optimal control," *Recent Advances in Algorithmic Differentiation*, 2012.
- [45] O. Kanoun, F. Lamiroux, and P.-B. Wieber, "Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 785–792, aug. 2011.
- [46] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Companion report," Tech. Rep. [Online]. Available: <http://hal.archives-ouvertes.fr/hal-00970816>

B Whole-Body Hierarchical Motion and Force Control for Humanoid Robots

Whole-Body Hierarchical Motion and Force Control for Humanoid Robots

Mingxing Liu, Vincent Padois

Received: date / Accepted: date

Abstract Robots acting in human environments usually need to perform multiple motion and force tasks while respecting a set of constraints. When a physical contact with the environment establishes, the newly activated force task or contact constraint may interfere with other tasks. This paper aims at handling a dynamically changing hierarchy of motion and force tasks of different priority levels. A torque-based control framework is proposed, which solves a quadratic programming problem to maintain desired task hierarchies, subject to constraints. This approach can achieve simultaneous priority transitions as well as activation or deactivation of tasks. A novel generalized projector is used to regulate quantitatively how much a task can influence or be influenced by other tasks through the modulation of a priority matrix. By the smooth variations of the priority matrix, sudden hierarchy rearrangements can be avoided to reduce the risk of instability. The effectiveness of this approach is demonstrated on a free-floating humanoid robot in simulation.

Keywords Whole-body control · Physical contact · Torque-based control · Humanoid robots

1 INTRODUCTION

Humanoids are expected to perform complex tasks, including physical interactions with environments (see Figure 1) through the control of their whole-body motion. When both motion and force tasks are involved, three problems should be handled. First, when the freedoms of a motion and a force task are not orthogonal to each other, for example when both motion and force tasks defined at the end-effector frame require the same degrees of freedom (DoF), then the priorities

between these two tasks should be handled, since both of them may not be satisfied all the time. Second, as motion and contact forces applied at different body frames can interfere with each other through robot dynamics, a control of each task with respect to their importance level is desired to achieve an appropriate whole-body performance. Third, if non-sliding contact constraints need to be satisfied, for example when foot contact forces need to be maintained within friction cones to avoid foot slippage, then the hierarchy of tasks should be consistent with such constraints. This paper develops a whole-body control framework for handling prioritized motion and force tasks during physical interactions. It focuses on the generalized framework formulation that allows for the regulation of priorities between motion and force tasks, as well as the activation or deactivation of tasks.

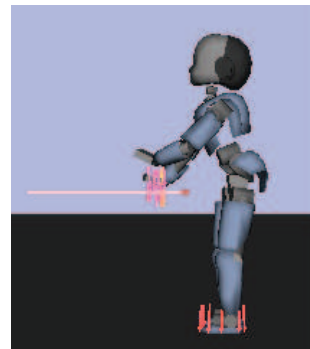


Fig. 1: Example of a humanoid robot in physical interactions with its environment.

This approach uses task specification matrices to restrict operational space positional freedom in the subspace orthogonal to the directions of force that is to be applied by the end-effector. With the development of humanoid robots, several whole-body motion and force control approaches have been proposed. A dynamic balance force controller [2] is developed for the control of center of mass (CoM) motion and contact forces of humanoid robots, where an additional task force is computed based on a CoM dynamics model and external forces to ensure balance. In these approaches, the control of an arbitrary number of prioritized tasks is not dealt with.

Recently, some hierarchical control frameworks have been proposed for the control of redundant robots. These control frameworks can be divided into three categories. The first category refers to analytical methods based on null-space projectors [3–6]. These approaches ensure *strict task hierarchies*, which means that lower-priority tasks are performed only in the null-space of higher priority tasks. Equality constraints can be implemented by the projection into the null-space of the Jacobian of constraint equations. However, it is difficult to impose inequality constraints with these approaches since they use pseudo-inverses and projection matrices. Therefore, non-sliding constraints that restrict contact forces inside friction cones can not be properly implemented.

The second category of approaches that can handle inequality constraints is hierarchical quadratic programming (HQP) [7–9], which is applied to whole-body motion control under unilateral constraints [10]. The idea of HQP is to first solve a QP to obtain a solution for a higher priority task objective; and then to solve another QP for a lower priority task, without increasing the obtained minimum of the previous task objective. This prioritization process corresponds to solving lower-priority tasks in the null-space of higher-priority tasks. Therefore, HQP also deals with *strict task hierarchies*.

In many contexts, organizing tasks by assigning them with strict priorities is not generic, *i.e.* can have some limitations. A task may not always have a strict priority over another one, and strict priorities can sometimes be too conservative so that they may completely block lower-priority tasks. Unlike the first two categories, the third one handles *non-strict task hierarchies* using weighting strategies [11–15]. These control frameworks solve all the constraints and task objectives in one QP and provide a trade-off among task objectives with different weights. As the performances of higher priority tasks cannot be guaranteed by simply adjusting the weights of task objectives, a prioritized control framework is proposed in [16] to ensure the performance of a higher-priority task within a user defined tolerance margin. However, this approach handles priorities of only two levels.

Another important difference between strict and non-strict hierarchies is how efficiently they achieve hierarchy rearrangements. For robots acting in dynamically changing contexts, task priorities may have to be switched, and certain tasks may have to be activated or deactivated to cope with changing situations, for example, frequent establishment and break of contacts. In this case, a sudden rearrangement of task hierarchy may lead to a great discontinuity in control laws and the increase of system instability. Recently, different methods based on strict hierarchies have been developed to handle priority transition problems. An approach to smooth priority rearrangement between two tasks is proposed in [17,18]. Approaches for continuous and simultaneous transitions of multiple tasks are developed in [19,20]. In [19], a specific inverse operator ensures continuous inverse in the analytical computation of control laws. The approach presented in [20] is based on intermediate desired values in the task space. When the number of task transitions increases, this approach suggests to apply an approximation to reduce the computational cost. An approach of hierarchical control with continuous null-space projections is presented in [21]. In this approach, an activator associated to directions in the right singular vectors of a task Jacobian matrix is regulated to activate or deactivate these directions. However, the design of such an activator makes this approach difficult to be implemented for the separate handling of different task directions. On the other hand, priority transitions can be easily achieved within a non-strict hierarchy by the continuous variation of task weights [14].

This paper addresses such smooth hierarchy rearrangement problems. The contribution of this paper is a whole-body hierarchical control framework, which can handle both strict and non-strict hierarchies, and can perform task hierarchy rearrangements gradually so that system instability due to such rearrangements can be reduced. This control framework formulates and solves all tasks and constraints in one QP, where linear inequality constraints, such as non-sliding contacts, can be implemented. Hierarchy rearrangements within this framework are based on the use of a novel generalized projection matrix, which has been developed in our previous work [22]. This projection matrix regulates to what extent a lower-priority task is projected into the null-space of a higher-priority task. In other words, it allows a task to be completely, partially, or not at all projected into the null-space of some other tasks. The priority levels can be changed by the modulation of the generalized projector. In [22] this projector is used in a dynamic control framework on a robotic manipulator. In this paper, it is the first time that such a projector is implemented in the control of a free-floating robot.

This paper is organized as follows. The robot model, as well as task definitions and task priority parametrization used in this paper are presented in Section 2. The control

framework is developed in Section 3. Some experimental results on a humanoid robot are presented in Section 4 to demonstrate the framework capabilities. Future research directions regarding the potential application of the proposed approach are presented in Section 5.

2 Modeling

Consider a robot as an articulated mechanism with n degrees of freedom (DoF) including n_a actuated DoF. The dynamics of the robot in terms of its generalized coordinates $q \in \mathbb{R}^n$ is written as follows

$$M(q)\ddot{q} + n(q, \dot{q}) + g(q) = S(q, \dot{q})^T \tau + J_c(q)^T w_c, \quad (1)$$

where $M(q) \in \mathbb{R}^{n \times n}$ is the generalized inertia matrix; $\dot{q} \in \mathbb{R}^n$ and $\ddot{q} \in \mathbb{R}^n$ are the vector of velocity and the vector of acceleration in generalized coordinates, respectively; $n(q, \dot{q}) \in \mathbb{R}^n$ is the vector of Coriolis and centrifugal induced joint torques; $g(q) \in \mathbb{R}^n$ is the vector of gravity induced joint torques; $S(q, \dot{q})^T \in \mathbb{R}^{n \times n_a}$ is a selection matrix for the actuated DoF; $\tau \in \mathbb{R}^{n_a}$ is the vector of the actuation torques; $J_c(q)^T = [J_{c,1}(q)^T \dots J_{c,n_c}(q)^T]^T$ is the transpose of a Jacobian matrix, with $J_{c,n_b}(q)$, the Jacobian matrix associated to a contact point β ; $w_c = [w_{c,1}^T \dots w_{c,n_c}^T]^T$ is the external contact wrenches applied to the robot, with n_c the number of contact points.

2.1 Motion and force task

Consider a robot performing motion and force tasks. Each task is associated with its task wrench. For a motion task i , the desired task wrench is the output of a task space proportional-derivative (PD) controller

$$w_{t,i}^d = K_{P,i}e_i + K_{D,i}\dot{e}_i, \quad (2)$$

where e_i and \dot{e}_i are task position and velocity errors, respectively; and $K_{P,i}$ and $K_{D,i}$ are symmetric, positive definite gain matrices.

For a force task, the desired task wrench is the output of a proportional-integral controller with a feed-forward term

$$w_{t,i}^d = w_{t,i}^* + K_{P,i}e_w + K_{I,i} \int e_w dt, \quad (3)$$

where $w_{t,i}^*$ is the desired task wrench, e_w is the error of task wrench, and $K_{P,i}$ and $K_{I,i}$ are symmetric, positive definite gain matrices.

2.2 Priority parametrization

The relative importance levels of each task i with respect to a set of n_t tasks, including task i , is characterized by a priority matrix α_i

$$\alpha_i = \text{diag}(\alpha_{i1}I_{m_1}, \dots, \alpha_{ij}I_{m_j}, \dots, \alpha_{in_t}I_{m_{n_t}}) \quad (4)$$

where m_j is the dimension of task j , α_i is a diagonal matrix, the main diagonal blocks of which are square matrices: $\alpha_{ij}I_{m_j}$. I_{m_j} is the $m_j \times m_j$ identity matrix, and $\alpha_{ij} \in [0, 1]$. In this paper, the notation $i \triangleright j$ indicates that task i has a strict higher priority over task j . By convention, the coefficient α_{ij} indicates the priority of task j with respect to task i .

- $\alpha_{ij} = 0$ corresponds to the case where task j has strict lower priority with respect to task i ($i \triangleright j$).
- $0 < \alpha_{ij} < 1$ corresponds to a soft (non-strict) priority between the two tasks: the greater the value of α_{ij} , the higher the importance level of task j with respect to task i .
- $\alpha_{ij} = 1$ corresponds to the case where task j has a strict higher priority with respect to task i ($j \triangleright i$).

3 Control problem formulation

In this work, multiple tasks with different priority levels subject to equality and inequality constraints have to be handled. This kind of multi-objective control problem can be formulated as a Linear Quadratic Programming problem (LQP), where all the task objectives and constraints are solved simultaneously in one LQP. Moreover, Jacobian-transpose method is adopted here to compute joint torques that are equivalent to task wrenches w_i applied at task frames.

This Section first briefly illustrate the control framework based on LQP and Jacobian-transpose method in 3.1, then develops a generalized projector in 3.2, which is implemented in this LQP-based control framework in 3.3 for handling a user-defined task hierarchy.

3.1 Control framework based on Linear Quadratic Programming and Jacobian-transpose method

The hierarchical control framework proposed by this paper extends the multi-objective control framework introduced in [16], which implements a weighting strategy to handle the importance levels of multiple elementary tasks. This Jacobian-transpose based quasi-static control framework is adopted in this paper because it is fast enough to achieve real-time control of robots with a high number of degrees of freedom.

This control framework proposes to solve a constrained multi-objective control problem in two steps. The first step

solves the LQP problem (3.1) to obtain optimal task forces and contact forces.

$$\arg \min_{w_{t,i}, w_{c,j}} \sum_i \|w_{t,i}^d - w_{t,i}\|_{\mathbf{Q}_{t,i}}^2 + \sum_j \|w_{c,j}\|_{\mathbf{Q}_{c,j}}^2 \quad (5a)$$

$$\text{s.t. } \sum_i \mathbf{J}_{t,i}^{rtT} w_{t,i} + \sum_j \mathbf{J}_{c,j}^{rtT} w_{c,j} + g^{rt} = 0 \quad (5b)$$

$$\mathbf{G} \begin{bmatrix} w_{t,i} \\ w_{c,j} \end{bmatrix} \leq \mathbf{h}. \quad (5c)$$

where the matrices $\mathbf{Q}_{t,i}$, $\mathbf{Q}_{c,j}$ are diagonal weighting matrices with $\mathbf{Q}_{t,i} = \omega_{t,i} \mathbf{I}_{m_i}$ and $\mathbf{Q}_{c,j} = \omega_{c,j} \mathbf{I}_3$, with ω , the scalar parameter of a task weight, \mathbf{I}_a , the $a \times a$ identity matrix, and m_i , the dimension of task i . If a task i is more important than another task j , then $\omega_{t,i} > \omega_{t,j}$. The norms of the task wrench errors are minimized to achieve a compromise among all the weighted tasks. The equality constraint (5b) is the static equilibrium of the root body under $w_{t,i}$, $w_{c,j}$, and g , where the superscript rt stands for the root (free-floating base) DoFs. The inequality constraints (5c) may include non-sliding contact constraints and bounds on wrench variables or on joint torques

$$\underline{\tau} \leq \sum_i \mathbf{J}_{t,i}^{acT} w_{t,i} + \sum_j \mathbf{J}_{c,j}^{acT} w_{c,j} + g^{ac} \leq \bar{\tau}, \quad (6)$$

where $\underline{\tau}$ and $\bar{\tau}$ are the lower and upper bounds of τ .

The second step is the computation of joint torques by using the solution of the first step. Let $w_{t,i}^*$ and $w_{c,j}^*$ denote the solution of (5). Joint torques are computed as follows

$$\tau = \sum_i \mathbf{J}_{t,i}^{acT} w_{t,i}^* + \sum_j \mathbf{J}_{c,j}^{acT} w_{c,j}^* + g^{ac} \quad (7)$$

where the superscript ac denotes the actuated DoFs.

3.2 Projectors for hierarchical control

Strict priorities can be handled by analytical methods using a null-space projector $N_j = I - J_j^\dagger J_j$ as defined in [23], where J_j^\dagger is the Moore-Penrose pseudo-inverse of the Jacobian J_j . The projection of a task i in the null-space of another task j can ensure that task i is performed without producing any motion for a task j . The idea of the use of null space projections to handle strict task priorities can be generalized to handle either strict or non-strict task priorities by either a complete or a partial projection of a task in the null-space of other tasks. This generalization leads to the development of the generalized projector $P_i(\alpha_i) \in \mathbb{R}^{n \times n}$ [22], which can handle both strict and non-strict priorities in a generalized way by the precise regulation of how much a task is affected by other tasks. For a torque controlled robot, the projector $P_i(\alpha_i)$ should be able to modify task torques τ_i by an appropriate projection ($P_i(\alpha_i)\tau_i$) to account for the hierarchy

information contained in α_i . This section provides a short outline of the development of the generalized projector as needed in this paper. For more details refer to [22].

In order to compute the generalized projector $P_i(\alpha_i)$, a preliminary processing of α_i and the augmented Jacobian J , which concatenates the Jacobian matrices of all the n_t tasks in a hierarchy ($J = [J_1^T \dots J_j^T \dots J_{n_t}^T]^T$), is carried out according to the priorities of all the tasks with respect to task i . As each row of J is associated to the same row in α_i , the rows of J can be sorted in descending order with respect to the values of the diagonal elements in α_i . The resulting matrix J_{s_i} is thus constructed so that tasks which should be the least influenced by task i appear in its first rows, while tasks which can be the most influenced by task i appear in its last rows. The values in α_i are sorted accordingly, leading to α_i^s , the diagonal elements of which are organized in descending order starting from the first row.

Based on J_{s_i} , a projector into the null space of J can be computed. This can be done by first computing a matrix $B_i(J_{s_i}) \in \mathbb{R}^{r \times n}$, where $r = \text{rank}(J_{s_i})$ is the rank of J_{s_i} . The rows of $B_i(J_{s_i})$ form an orthonormal basis of the joint space obtained using elementary row transformations on J_{s_i} . Then this projector can be computed as $P_i' = I_n - B_i^T B_i$, which is a symmetric matrix. When performing task i by using the projected joint torques $P_i' \tau_i = (J_i P_i')^T w_i$, the projector P_i' basically cancels any joint torque that impacts all the n_t tasks, including task i itself.

The computation of the projector P_i' can be modified such that tasks having strict priority over task i are perfectly accounted for; tasks over which task i has a strict priority are not considered; and all other tasks with soft priorities are accounted for, according to the value of their respective priority parameters in α_i . The generalized projector taking account of all these requirements is given by

$$P_i(\alpha_i) = I_n - B_i(J_{s_i})^T \alpha_{i,r}^s(\alpha_i, \text{origin}) B_i(J_{s_i}), \quad (8)$$

where $\alpha_{i,r}^s$ is a diagonal matrix of degree r . The vector $\text{origin} \in \mathbb{R}^r$ is a vector of the row indexes of J_{s_i} selected during the construction of the orthonormal basis B_i . Each of these r rows in J_{s_i} is linearly independent to all the previously selected ones. The diagonal elements of $\alpha_{i,r}^s$ are restricted to the r diagonal elements of α_i^s , which correspond to the r rows of J_{s_i} , the row indexes of which belong to origin .

Algorithm (1) and (2) summarize the construction of the generalized projector $P_i(\alpha_i)$. As any numerical scheme, tolerances are used for numerical comparison, such as ϵ , which is defined as the smallest value greater than zero in line #11 of Algorithm (2).

Note that by varying the value of each α_{ij} in α_i , one can regulate the priority of each task j in the n_t tasks with respect to task i separately.

¹ The dependence to q is omitted for clarity reasons.

Algorithm 1: Generalized projector computation - task i

Data: α_i, J
Result: P_i

```

1 begin
2    $n \leftarrow \text{GetNbCol}(J)$ 
3    $\text{index} \leftarrow \text{GetRowsIndexDescOrder}(\alpha_i)$ 
4    $\alpha_i^s \leftarrow \text{SortRows}(\alpha_i, \text{index})$ 
5    $J_{s_i} \leftarrow \text{SortRows}(J, \text{index})$ 
6    $B_i, \text{origin}, r \leftarrow \text{GetOrthBasis}(J_{s_i}) \triangleright \text{Alg. (2)}$ 
7    $\alpha_{i,r}^s \leftarrow \text{GetSubDiagMatrix}(\alpha_i^s, \text{origin})$ 
8    $P_i \leftarrow I_n - B_i^T \alpha_{i,r}^s B_i$ 
9   return  $P_i$ 
10 end
```

Algorithm 2: Orthonormal basis computation - $\text{GetOrthBasis}(A)$

Data: A, ϵ
Result: B, origin, r

```

1 begin
2    $n \leftarrow \text{GetNbCol}(A)$ 
3    $m \leftarrow \text{GetNbRow}(A)$ 
4    $i \leftarrow 0$ 
5   for  $k \leftarrow 0$  to  $m-1$  do
6     if  $i \geq n$  then
7       break
8      $B[i, :] \leftarrow A[k, :]$ 
9     for  $j \leftarrow 0$  to  $i-1$  do
10       $B[i, :] \leftarrow B[i, :] - (B[i, :] B[j, :]^T) B[j, :]$ 
11     if  $\text{norm}(B[i, :]) > \epsilon$  then
12        $B[i, :] \leftarrow B[i, :] / \text{norm}(B[i, :])$ 
13        $\text{origin}[i] \leftarrow k$ 
14        $i \leftarrow i + 1$ 
15 end
16  $r \leftarrow i$ 
17 return  $B, \text{origin}, r$ 
```

3.2.1 Task insertion and deletion

There is a particular case induced by the proposed formulation and corresponding to the influence of task i on itself. Even though not intuitive, this self-influence has to be interpreted in terms of task existence, modulated by α_{ii} . If $\alpha_{ii} = 1$ then task i is projected into its own null-space, *i.e.* it is basically canceled out. Decreasing α_{ii} continuously to 0 is a simple and elegant way to introduce the task in the set of tasks. Conversely, increasing α_{ii} continuously from 0 to 1 provides with a proper task deletion procedure. When being added or suppressed, the influence of task i with respect to other tasks also has to be defined and here again this can be done by the regulation of α_{ij} .

3.3 Hierarchical Motion and Force Control

The control framework presented in Section 3.1 can qualitatively regulate the relative importance levels of tasks through task objective weights, but it cannot precisely ensure strict priorities among tasks. This control framework is extended in this paper to account for both strict and non strict task priorities. Moreover, an advantage of this approach is that a priority rearrangement can be performed between any two tasks.

The major difference between our hierarchical control framework and the control framework reviewed in Section 3.1 is that each task Jacobian \mathbf{J}_i is modulated by the generalized projector to account for the desired priority levels. Therefore, the control framework presented in (3.1) is modified here by the application of modulated Jacobians. The LQP problem to be solved is

$$\arg \min_{w_i, w_{c_j}} \sum_i \|w_i^d - w_i\|_{\mathbf{I}}^2 + \sum_j \|w_{c_j}\|_{\mathbf{Q}_{c_j}}^2 \quad (9a)$$

$$\text{s.t. } \sum_i \mathbf{P}_i \mathbf{J}_i^{rt} w_i + \sum_j \mathbf{J}_{c_j}^{rt} w_{c_j} + g^{rt} = 0 \quad (9b)$$

$$\mathbf{G}(\mathbf{P}_i) \begin{bmatrix} w_i \\ w_{c_j} \end{bmatrix} \leq \mathbf{h}(\mathbf{P}_i). \quad (9c)$$

The control input τ is computed from modulated task wrenches (7).

$$\tau = \sum_i \mathbf{P}_i \mathbf{J}_i^{acT} w_i^* + \sum_j \mathbf{J}_{c_j}^{acT} w_{c_j}^* + g^{ac} \quad (10)$$

Here, the task objective weighting matrix \mathbf{Q}_i is set to the identity matrix, and the task hierarchy is handled by generalized projectors. The weighting matrix $\mathbf{Q}_r = \omega_r \mathbf{I}_{n+n_a+3n_c}$ of the contact force objective is set to a diagonal matrix with the weight value ω_r being very small compared to 1. The contact force objective here is not a target tracking objective. It is used to ensure the uniqueness of the optimization solution. The norm of the contact wrenches w_{c_j} is minimized, as their desired values are unknown *a priori*. Appropriate value of w_{c_j} , which satisfies the static equilibrium and non-sliding contact constraints, is computed by solving the LQP.

Non-sliding contact constraints are implemented as inequality constraints, where contact forces are constrained inside linearized Coulomb friction cones. Bounds of joint torques (11) can be implemented as inequality constraints within this framework using modulated task Jacobians.

$$\underline{\tau} \leq \sum_i \mathbf{P}_i \mathbf{J}_i^{acT} w_i + \sum_j \mathbf{J}_{c_j}^{acT} w_{c_j} + g^{ac} \leq \bar{\tau}. \quad (11)$$

It should be noticed that, in this framework, the real task wrenches applying on the system are not w_i , but those modulated by generalized projectors. As generalized projectors are always directly applied to task Jacobians instead of task

wrenches, it is difficult to apply bounds or non-sliding contact constraints directly on modulated task wrenches. Therefore, crucial contacts, such as foot contacts that should not slip, are set as constraints instead of tasks.

4 RESULTS

The proposed control approach has been implemented on a free-floating humanoid robot iCub and validated by experiments. The experiments are carried out on the simulator XDE [24], which is a software environment that manages physics simulation in real time. The robot has 38 DoFs, including 6 DoFs of its root body, and 32 DoFs of its joints. It is required to stand on the ground and switch its hands to apply a contact force of $30N$ on a table periodically (see Figure 2). The table surface is connected with the ground through a spring with a stiffness of $2000 N/m$ and a damping of $89 Ns/m$.

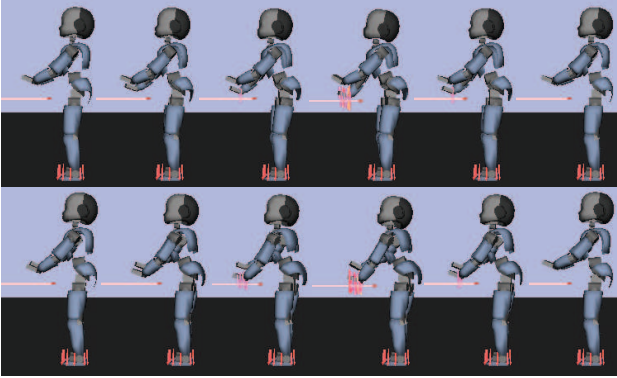


Fig. 2: Snapshots of the robot switch its hands to apply a contact force on a table periodically by using the control framework proposed in this paper.

Four tasks are considered, namely the 2-D center of mass (CoM) task, the 3-D right hand (rh) and left hand (lh) position tasks, the 3-D right hand and left hand orientation tasks, the 1-D right hand force (rhf) and left hand force (lhf) tasks, the 1-D head orientation task, the 32-D posture task, and four 3-D contact force tasks on each feet. The static equilibrium constraint (9b) is applied to the free-floating base. Non-sliding contact constraints are applied to contact points on the feet.

During the experiment, the CoM task has the highest priority over all the other tasks to ensure the balance of the robot. The posture task, which is used for redundancy resolution, is always assigned with the lowest priority.

A finite state machine (FSM) is applied. The states are: *idle*, *rh-reaching*, *rh-contact*, *rh-release*, *lh-reaching*, *lh-contact*, *lh-release*. As the table is connected with the ground by a

spring, the table surface will move downward when the hand pushes it strongly. Hand task targets during contact states are fixed on the surface of the initial table position; while the actual hand position during this state should be lower than this target position to be able to increase the contact force to $30N$. This means that during the periodical behavior of contact establishment and break between the hands and the table, priorities between hand force tasks and hand position tasks should be modified. Task priorities with respect to different FSM states are illustrated in Figure 3.

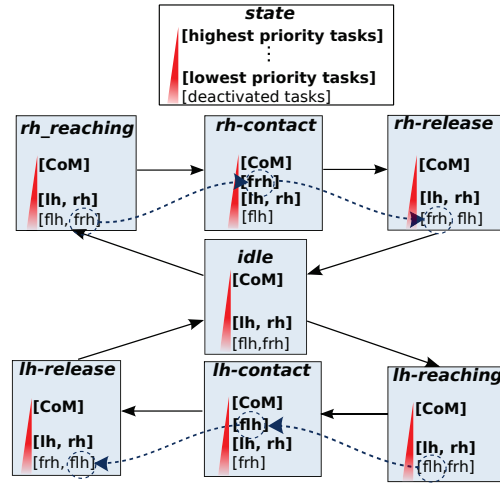


Fig. 3: Task priorities with respect to different states of the finite state machine. Priorities of the CoM task, hand position tasks, and hand force tasks are shown, and those of the other tasks are omitted for clarity.

- At the beginning, the robot is in *idle* state. During this state, its hands are not in contact with the table. The hand force tasks are deactivated, and they have a strict lower priority than hand position tasks by default.
- In *rh/lh-reaching* state, the hand moves toward the table.
- When a contact is established with the table, the FSM enters *rh/lh-contact* state. When entering this state, the hand force task is gradually activated and its priority increases gradually over hand position task to enhance the control of hand contact force.
- When *rh/lh-release* state starts, the hand should move away from the table to a target position above it. When entering this state, the hand force task is gradually deactivated and its priority with respect to hand position task decreases to enhance hand position control.

The following function is used for the smooth variation of an α_{ij} (conversely α_{ji}) from 0 to 1 during the transition

time period $([t_1, t_2])$

$$\alpha_{ij}(t) = 0.5 - 0.5 \cos\left(\frac{t-t_1}{t_2-t_1}\pi\right), \quad t \in [t_1, t_2], \quad (12)$$

$$\alpha_{ji}(t) = 1 - \alpha_{ij}(t).$$

The experiment is first conducted with the hierarchy rearrangement period $(t_2 - t_1)$ being set to 0.08s. The result of α , hand contact forces, as well as the errors of the CoM and the hand position tasks are shown in Figure 4. At the beginning, $\alpha_{lhf,lhf} = 1$ and $\alpha_{rhf,rhf} = 1$, which means that the force tasks are deactivated since they are projected in their own null-spaces. When the hand touches the table, $\alpha_{lhf,lhf}$ (or $\alpha_{rhf,rhf}$) decreases to zero smoothly to activate the force task gradually. During the contact phase, $\alpha_{rhf,rh}$ (or $\alpha_{lhf,lh}$) decreases to zero and $\alpha_{rh,rhf}$ (or $\alpha_{lh,lhf}$) increases smoothly so that the priority of hand force task increases gradually over hand position task. During this hierarchy rearrangement, as can be observed in Figure 4, the hand task error increases while the force task tracks better its reference.

Moreover, during the experiment, the equilibrium of the robot is maintained and no foot slippage is observed, which demonstrates that this approach can handle a task hierarchy subject to both equality constraint (static equilibrium) and inequality constraint (non-sliding contacts).

An advantage of this approach is that the rearrangement of task hierarchy can be carried out gradually and more smoothly to avoid abrupt hierarchy changes and thus reduce system instability. To demonstrate this, the same experiment is carried out with a sudden change of relevant α s (during 0.03s which is much faster than in the previous experiment). The resulting hand contact forces are shown in Figure 5, and hand force task errors with both gradual and sudden hierarchy rearrangements are shown in Figure 6.

Figure 5 and 6 show that greater force task errors with larger peaks can be observed when hierarchies are rearranged suddenly, compared with the previous experiment where hierarchies are changed more slowly (by the smoother variations of α). In fact, this hierarchical force and motion control approach allows for the precious adjustment of hierarchy rearrangement speed. As this approach parametrizes task priorities in a continuous way and can encode priorities between each pair of tasks, it is richer and more informative compared with a discrete parametrization used in approaches that handle strict-priorities only (such as analytical approaches based on null-space projection and HQP). A potential application of this framework could be a combination with robot learning techniques to incrementally learn and improve the tuning of priority parameters α for different scenarios of interactions with the environment.

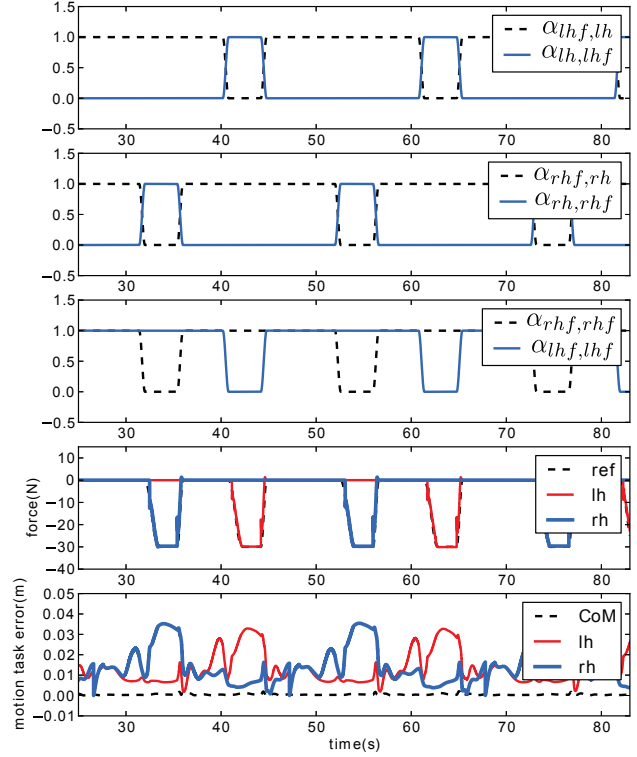


Fig. 4: Change of α (top), desired and real hand contact forces (middle), and the errors of the CoM and the hand position tasks (bottom). Hierarchy rearrangement period lasts 0.08s.

5 Conclusions and Future Works

This paper proposes a novel hierarchical control approach to handling multiple motion and force tasks for a free-floating humanoid robot. A novel generalized projector is used to precisely regulate how much a task can influence or be influenced by other tasks through the modulation of a priority matrix. By using the same mechanism, tasks can be easily activated or deactivated. Experiments demonstrate that both motion and contact force tasks of different priorities can be handled by this approach. Task priorities can be maintained and switched while respecting both equality and inequality constraints.

The control framework presented here is quasi-static; however, the hierarchical control based on generalized projectors is not restricted in such a quasi-static case. In fact, it can also be used in other types of controllers, such as a dynamic controller. The basic idea is to associate each task with a task variable in joint space (\dot{q}_i' , \ddot{q}_i' , or τ_i'), then to apply generalized projectors to modulate these task variables, and finally the global joint space variable is the sum of each projected task variables ($P_i(\alpha_i)\dot{q}_i'$, $P_i(\alpha_i)\ddot{q}_i'$, or $P_i(\alpha_i)\tau_i'$).

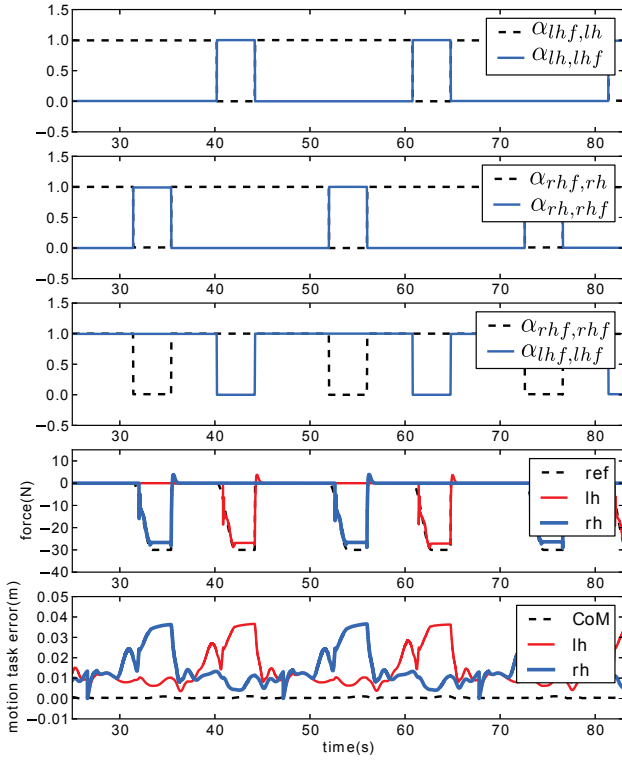


Fig. 5: Change of α (top), desired and real hand contact forces (middle), and the errors of the CoM and the hand position tasks (bottom). Hierarchy rearrangement period lasts 0.03s.

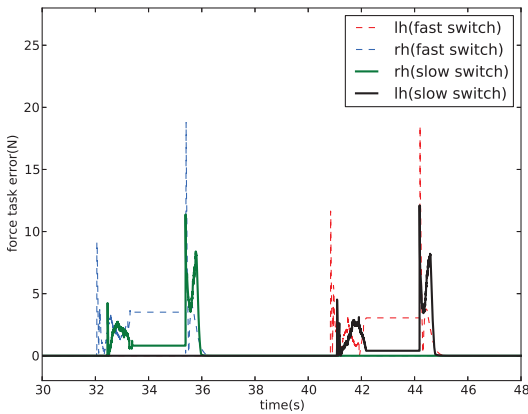


Fig. 6: Hand contact force errors with a slower hierarchy rearrangement of 0.08s (solid lines) and the faster rearrangement of 0.03s (dotted lines).

A future research direction is to extend this approach for the dynamic control of humanoid robots and to analyze its computational cost. As in the dynamic case, the relation between joint accelerations, joint torques, and contact forces has to be considered, so the optimization problem should handle more variables. Finally, as this approach uses a continuous parametrization of task priorities, the use of robot learning techniques to incrementally learn and improve the tuning of these priority parameters is also of great interest.

Acknowledgements This work was partially supported by the European Commission, within the CoDyCo project (FP7-ICT-2011-9, No. 600716) and by the RTE company through the RTE/UPMC chair “Robotics Systems for field intervention in constrained environment” held by Vincent Padois.

References

1. O. Khatib, Robotics and Automation, IEEE Journal of **3**(1), 43 (1987)
2. B. Stephens, C. Atkeson, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2010), pp. 1248–1255
3. L. Sentis, O. Khatib, in *4th IEEE/RAS International Conference on Humanoid Robots*, vol. 2 (2004), vol. 2, pp. 764–780 Vol. 2. DOI 10.1109/ICHR.2004.1442684
4. M. Mistry, J. Nakanishi, S. Schaal, in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on* (2007), pp. 331–338. DOI 10.1109/IROS.2007.4399595
5. P. Hsu, J. Mauser, S. Sastry, Journal of Robotic Systems **6**(2), 133 (1989). DOI 10.1002/rob.4620060203. URL <http://dx.doi.org/10.1002/rob.4620060203>
6. F. Flacco, A. De Luca, O. Khatib, in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on* (2012), pp. 3970–3977. DOI 10.1109/IROS.2012.6385619
7. O. Kanoun, F. Lamiraux, P.B. Wieber, F. Kanehiro, E. Yoshida, J.P. Laumond, in *IEEE International Conference on Robotics and Automation (ICRA)* (2009) (2009), pp. 2939–2944
8. L. Saab, N. Mansard, F. Keith, J.Y. Fourquet, P. Soueres, in *IEEE International Conference on Robotics and Automation (ICRA)* (2011), pp. 1091–1096
9. A. Escande, N. Mansard, P.B. Wieber, The International Journal of Robotics Research p. 0278364914521306 (2014)
10. L. Saab, O. Ramos, F. Keith, N. Mansard, P. Soueres, J.Y. Fourquet, Robotics, IEEE Transactions on **29**(2), 346 (2013). DOI 10.1109/TRO.2012.2234351
11. Y. Abe, M. da Silva, J. Popović, in *Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation* (2007), pp. 249–258
12. C. Collette, A. Micaelli, C. Andriot, P. Lemerle, in *7th IEEE-RAS International Conference on Humanoid Robots* (2007), pp. 81–88
13. M. Liu, A. Micaelli, P. Evrard, A. Escande, C. Andriot, in *IEEE International Conference on Robotics and Automation (ICRA)* (2011), pp. 1676–1682
14. J. Salini, V. Padois, P. Bidaud, in *Robotics and Automation (ICRA), 2011 IEEE International Conference on* (2011), pp. 1283–1290. DOI 10.1109/ICRA.2011.5980202
15. K. Bouyarmane, A. Kheddar, in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on* (2011), pp. 4414–4419. DOI 10.1109/IROS.2011.6094483
16. M. Liu, A. Micaelli, P. Evrard, A. Escande, C. Andriot, IEEE Transactions on Robotics **28**(6), 1309 (2012). DOI 10.1109/TRO.2012.2208829

17. F. Keith, N. Wieber, P.-B. and Mansard, A. Kheddar, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2011), pp. 3887–3892. DOI 10.1109/IROS.2011.6094706
18. T. Petrič, L. Žlajpah, *Robotics and Autonomous Systems* **61**(9), 948 (2013). DOI <http://dx.doi.org/10.1016/j.robot.2013.04.019>
19. N. Mansard, A. Remazeilles, F. Chaumette, *Automatic Control, IEEE Transactions on* **54**(11), 2493 (2009). DOI 10.1109/TAC.2009.2031202
20. J. Lee, N. Mansard, J. Park, *Robotics, IEEE Transactions on* **28**(6), 1260 (2012). DOI 10.1109/TRO.2012.2210293
21. A. Dietrich, A. Albu-Schaffer, G. Hirzinger, in *Robotics and Automation (ICRA), 2012 IEEE International Conference on* (2012), pp. 2978–2985. DOI 10.1109/ICRA.2012.6224571
22. M. Liu, Y. Tan, V. Padois, *Generalized Hierarchical Control* (2014). URL <http://hal.archives-ouvertes.fr/hal-01068404>
23. A. Liégeois, *IEEE Transactions on Systems, Man and Cybernetics* **7**(12), 868 (1977)
24. CEA-List. XDE eXtended Dynamic Engine. <http://www.kalisteo.fr/lisi/en/aucune/a-propos-de-xde>

mainly focused on the automatic design, the modeling and the control of redundant and complex systems such as wheeled mobile manipulators, humanoid robots as well as standard manipulators evolving under constraints in complex environments. He is also involved in research activities that aim at bridging the gap between adaptation and decision making techniques provided by Artificial Intelligence and low-level, reactive control. Since 2011, he holds the “Intervention Robotics” RTE/UPMC chair position.



Mingxing Liu is a post-doctoral fellow at Institut des Systèmes Intelligents et de Robotique (ISIR, UMR CNRS 7222) at Université Pierre et Marie Curie (UPMC), Paris, France. In 2009, she received both her engineering degree from Ecole Centrale Paris, France and her master’s degree in Automatic Control from Shanghai Jiao Tong University, China. From 2009 to 2012, she is a PhD student at the Systems and Technologies Integration Laboratory of the French Atomic Energy Commis-

sion (CEA-LIST). She received the Ph.D. degree in Robotics in 2013 from UPMC. Her research interests include automatic control, whole-body control for redundant robots such as industrial manipulators and humanoid robots, as well as human-robot interaction.



Vincent Padois is an associated professor of Robotics and Computer Science and a member of the Institut des Systèmes Intelligents et de Robotique (ISIR, UMR CNRS 7222) at Université Pierre et Marie Curie in Paris, France. In 2001, he receives both an engineering degree from the Ecole Nationale d’Ingénieurs de Tarbes (ENIT), France and his master’s degree in Automatic Control from the Institut National Polytechnique de Toulouse (INPT), France. From 2001 to 2005, he is a PhD stu-

dent in Robotics of the ENIT/INPT Laboratoire Genie de Production.

In 2006 and 2007, he is a post-doctoral fellow in the Stanford Artificial Intelligence Laboratory and more specifically in the group of Professor O. Khatib. Since 2007, his research activities at ISIR are

C iCub Whole-body Control through Force Regulation on Rigid Non coplanar Contacts



iCub Whole-body Control through Force Regulation on Rigid Noncoplanar Contacts

Francesco Nori¹, Silvio Traversaro¹, Jorhabib Eljaik¹, Francesco Romano¹,
Andrea Del Prete² and Daniele Pucci¹

¹*Cognitive Humanoids Laboratory, Fondazione Istituto Italiano di Tecnologia, Robotics Brain and Cognitive Sciences Department, Genoa, Italy.*

²*CNRS, LAAS, 7 avenue du colonel Roche, Univ de Toulouse, LAAS, F-31400 Toulouse, France.*

Correspondence*:

Nori Francesco

Cognitive Humanoids Laboratory, Fondazione Istituto Italiano di Tecnologia,
Robotics Brain and Cognitive Sciences Department, Via Morego 30, 16163
Genova, Italy, francesco.nori@iit.it

ABSTRACT

This paper details the implementation of state-of-the-art whole-body control algorithms on the humanoid robot iCub. We regulate the forces between the robot and its surrounding environment to stabilize a desired posture. We assume that the forces and torques are exerted on rigid contacts. The validity of this assumption is guaranteed by constraining the contact forces and torques, e.g., the contact forces must belong to the associated friction cones. The implementation of this control strategy requires the estimation of both joint torques and external forces acting on the robot. We then detail algorithms to obtain these estimates when using a robot with an iCub-like sensor set, i.e. distributed six-axis force-torque sensors and whole-body tactile sensors. A general theory for identifying the robot inertial parameters is also presented. From an actuation standpoint, we show how to implement a joint torque control in the case of DC brushless motors. In addition, the coupling mechanism of the iCub torso is investigated. The soundness of the entire control architecture is validated in a real scenario involving the robot iCub balancing and making contact with both arms.

16

17 **Keywords:** whole-body control, floating base robots, rigid contacts, noncoplanar contact, tactile sensors, force sensors.

1 INTRODUCTION

Classical industrial applications employ robots with limited mobility. Consequently, assuming that the robot is firmly attached to the ground, interaction control (e.g. manipulation) is usually achieved separately from whole-body posture control (e.g. balancing). Foreseen applications involve robots with augmented autonomy and physical mobility. Within this novel context, physical interaction influences stability and balance. To allow robots to overcome barriers between interaction and posture control, forthcoming robotics research needs to investigate the principles governing *whole-body coordination with contact dynamics*, as these represent important challenges towards achieving robot physical autonomy and will therefore be the focus of the present paper.

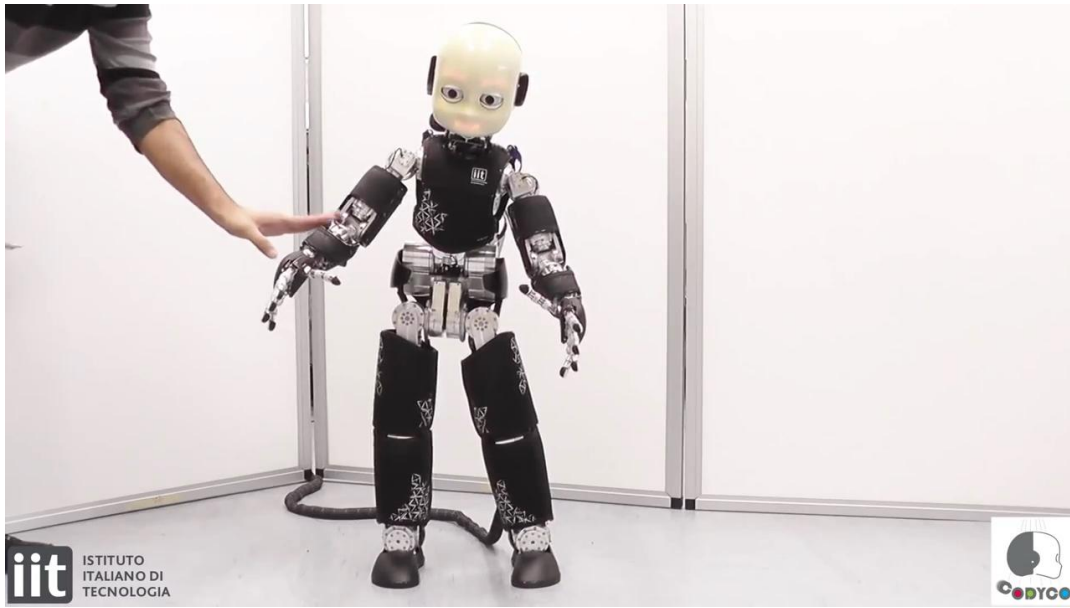


Figure 1. See the video showing the control performances of the control architecture https://www.youtube.com/watch?v=jaTEbCsFp_M

It is worth recalling that the aforementioned industrial robots have been extensively studied since the early seventies. Robot physical autonomy asks for switching from conventional fixed-base to free-floating robots, whose control has been addressed only during recent years. Free-floating mechanical systems are under actuated and therefore cannot be fully feedback linearized **Spong** (1994). The problem becomes even more complex when these systems are constrained, that is their dynamics are subject to a set of (possibly time-varying) nonlinear constraints. This is the typical case for legged robots, for which motion is constrained by rigid contacts with the ground.

The major contribution of this work is the implementation and integration of all the building blocks composing a system for balance and motion control of a humanoid robot. The system includes the low-level joint-torque control, the task-space inverse-dynamics control, the task planner and the estimation of contact forces and joint torques. Even though in recent years other similar systems have been presented **Ott et al.** (2011); **Herzog et al.** (2013), the originality of our contribution lies i) in the specificities of our test platform and ii) in a number of design choices that traded off simplicity of implementation for performances of the control system. In particular:

- differently from the other robots, iCub can localize and estimate contact forces on its whole body thanks to its distributed tactile sensors
- similarly to the *DLR-Biped* **Ott et al.** (2011), iCub is actuated with DC motors and harmonic drives, but we chose to neglect the gear-box flexibility, which simplified the motor-identification procedure and the low-level torque controller
- differently from the above-mentioned platforms, iCub is not equipped with joint-torque sensors, but we designed a method that exploits its internal 6-axis force/torque sensors to estimate the joint torques
- all our control loops run at 100 Hz, which is (at least) 10 times slower with respect to **Ott et al.** (2011); **Herzog et al.** (2013)

We believe that further investigation will be necessary to thoroughly understand all the consequences of our hardware/software design choices. Nonetheless, these peculiarities make the presented system unique, and for this reason we think it is important to share our results with the robotics community.

The paper is organized as follows: Section 2 reviews the state of the art and motivates our specific choices, with a focus on why we defined postural stability by means of the center of pressure at individual contacts. A counterexample discussed in Section 2.3.3 shows that the commonly-used global center of pressure is not suitable for the scope of our application. Section 3.1 describes the whole-body distributed force and tactile sensors on iCub. These sensors are used to estimate contact forces (Section 3.2.1), internal torques (Section 3.2.2) and to improve the accuracy of the robot's inertial parameters (Section 3.3), while Section 3.4 presents the prioritized contact-force controller. Section 4 discusses the implementation scenario, consisting in controlling the iCub posture and contact forces at both arms and feet. Remarkably, iCub can establish and break contacts at the arms using tactile sensing for both contact detection and localization. Finally, Section 5 draws the conclusions.

2 BACKGROUND

This section reviews previous literature on rigid contacts and their role in whole-body stability. Then we conclude that for the scope of the current paper we need to consider local contact stability as opposed to global stability criteria proposed in previous literature. Section 2.1 makes some general considerations about contacts and Section 2.1.1 gives a characterization of contacts by means of the center of pressure, a point in space that summarizes the effects of distributed forces acting on a rigid body. Section 2.2 focuses on planar unilateral contacts and their stability (Section 2.2.1). This specific type of contacts is associated with a center of pressure that lies on the contact plane (Section 2.2.2). This property is exploited to give necessary and sufficient conditions for the stability of a planar unilateral contact (Section 2.2.3). Section 2.3 reviews previous literature on multiple contacts. In particular, Section 2.3.1 considers the coplanar case, whereas Section 2.3.2 the noncoplanar one. Section 2.3.3 discusses a counterexample to justify our choice of addressing the multiple-contact case without resorting to a global stability criteria as proposed in most of previous approaches. Finally Section 2.4 briefly reviews the state of the art of prioritized task-space inverse dynamics.

2.1 CONTACTS

We consider articulated rigid-body systems under the effects of multiple rigid contacts. In general, a contact can be seen as a continuum of infinitesimal forces acting on the surface of a rigid body. The effect of contact forces will be represented with an equivalent wrench $w_c = (f_c, \mu_c)$, composed by a three-dimensional force and a three-dimensional torque, denoted f_c and μ_c respectively. Considering that a contact exerts infinitesimal forces distributed over a surface, f_c is computed as the integral of infinitesimal forces over the surface. Similarly, μ_c is computed as the integral of the infinitesimal torques due to infinitesimal forces over the surface. The effect of other (noncontact) forces and torques acting on the rigid body, will be denoted $w_o = (f_o, \mu_o)$, being f_o and μ_o the equivalent force and torque (respectively) resulting from all noncontact forces.

2.1.1 Definition of center of pressure (CoP). Given a rigid body subject to contact forces, we associate a field of pressure to the contact itself. For each contact point, the pressure is defined as the amount of normal force acting per unit area. The center of pressure (CoP) is defined as an application point where the force obtained by the integration of the field of pressures causes an effect that is equivalent to that of the field of pressures itself. *Remark 1:* by definition, pressure accounts only for the normal component of the contact forces acting on the surface of a rigid body. Therefore the CoP comes handy especially when the tangential forces (shear stress) are negligible or do not play a role in the rigid body dynamics (e.g. the effect of tangential forces is compensated by the static friction). Otherwise, the CoP does not represent per-se a full characterization of the forces acting on the system and the effect of tangential forces should be also taken into account. *Remark 2:* at any point in space the effect of a field of pressures can be represented by an equivalent force and an equivalent torque (the integral of infinitesimal forces and

torques respectively). Given the above definition, the CoP is an application point where the equivalent torque is null.

2.1.2 Existence of the center of pressure (CoP). Given the above definition, we can infer that the CoP sometimes might not exist. As a trivial example, the CoP does not exist when a field of pressure generates a zero net force but a nonzero net torque. Excluding this trivial case, by resorting to the Poinot theorem it can be shown (see Appendix A) that the CoP is defined if and only if the resulting net torque is orthogonal to the resulting net force. Two relevant cases that satisfy this condition can be identified: field of pressures due to forces lying on a plane (torques orthogonal to the plane) and field of pressures due to forces orthogonal to a plane (torques lying on the plane). The first is a typical example used in aerodynamics (profile of a wing) and the latter is a typical example in the field of humanoid walking (contact with planar surfaces). In any case, the CoP is never uniquely defined and the set of valid CoPs corresponds to the Poinot axis (see Appendix A.2). In the case of planar contacts, we will uniquely identify the CoP with the intersection between the axis and the planar contact surface as proposed by **Sardain and Bessonnet** (2004).

2.2 PLANAR UNILATERAL CONTACT

A particular type of contacts, nominally planar unilateral contacts, has been widely studied to characterize the stability of an articulated rigid-body system while walking on flat terrain. The typical case-study considers a single link (foot) in contact with a flat surface (ground). Proposed stability criteria take into account the fact that while the foot has to be constantly in contact with the ground, the rest of the body is moving and therefore transfers inertial and gravitational forces to the foot. The foot is therefore subject to two sets of wrenches: those due to the contact with the ground (w_c) and those due to the movements of the rest of the body (w_o). For the contact to be stable these forces should balance (see Section 2.2.1). Force balance might not always hold since planar unilateral contacts exert a limited range of forces and torques. Original stability properties were proposed by **Vukobratovic and Juricic** (1969), who introduced the *zero moment point* (ZMP) concept. The ZMP coincides with the unique point on the ground where f_c , μ_c produce zero tangential moments (see Section 2.2.2). As it was pointed out by **Sardain and Bessonnet** (2004), in the case of planar contacts the ZMP coincides with the intersection of the Poinot axis with the contact plane as defined in Section 2.1.1; the ZMP is therefore a valid CoP. Other stability criteria for planar unilateral contacts have been proposed by **Goswami** (1999) and reviewed in Section 2.2.3.

2.2.1 Definition of stable planar unilateral contact. So far we have only discussed about forces generated by contacts. In general, contacts also introduce motion constraints and there is always a duality between contact forces and constrained motion. In the case of rigid contacts, the directions in which motion is constrained are precisely those in which (contact) forces can be exerted as observed by **Murray et al.** (1994). In other words, contact forces and possible motions are always orthogonal. From a control point of view it is often desirable that the set of motion constraints does not change over time, since if it does, the control problem becomes harder to solve (see for example literature on hybrid and switching systems). This is the reason why *we say that a contact is stable when the motion constraints induced by it do not change over time*. Interestingly, motion constraints are effective on the system only if certain conditions are satisfied. If these conditions are not met (e.g. contact forces violate unilateral constraints or exceed friction cones) contacts are broken and motion constraints are no longer active on the system. It is therefore important to find a set of necessary and sufficient conditions for a contact to constrain always the same motion.

In the case of planar unilateral contacts these conditions assume an elegant form that will be presented in Section 2.2.3. The analysis is simplified by observing that planar unilateral contacts impose constraints

on all linear and angular motions¹. Therefore a planar unilateral contact imposes null linear and angular accelerations. By means of the Newton-Euler equations on the contact link, motion constraints are therefore guaranteed if and only if $f_c = -f_o$ and $\mu_c = -\mu_o$. In a sense, deciding whether or not a planar unilateral contact is stable (with the above terminology) corresponds to understanding if a given wrench f_o, μ_o can be compensated by the forces f_c and torques μ_c generated by the given type of contact. If $-f_o, -\mu_o$ lie outside the space of wrenches that a planar unilateral contact can generate, then nonzero accelerations are generated and the contact is broken. The following section shows how to characterize the set of wrenches generated by a planar unilateral contact.

2.2.2 Characterization of the CoP for planar unilateral contacts. Given a planar unilateral contact, the set of forces induced by the contact are such that it is always possible to find a point on the plane where the equivalent moment has null tangential components. This point has been named zero moment point (ZMP) by **Vukobratovic and Juricic** (1969). The name is sometimes considered misleading (see for example **Sardain and Bessonnet** (2004)) since at the ZMP the “tipping” (or tangential) moment and not the “total” moment is zeroed. The computation of the zero tipping moment point is relatively straightforward and reformulated in Appendix B.1. In the case of unilateral contacts, the ZMP coincides with the CoP (see **Sardain and Bessonnet** (2004)) and always lies in the convex hull of the contact points as shown by **Wieber** (2002) and in Appendix B.1. It is worth noticing that (in the ZMP context) restricting to tangential moments corresponds (in the CoP context) to neglecting tangential forces (see the first remark in Section 2.1.1).

2.2.3 Characterization of stable planar unilateral contact. **Goswami** (1999) pointed out that the ZMP lying within the contact convex hull is not a proper stability measure. He therefore formulated some different statements for the characterization of the stability of planar unilateral contacts. These statements make use of the foot rotation indicator (FRI) which corresponds to the unique zero tipping moment point associated to f_o, μ_o and belonging to the contact plane. The name FRI is misleading since this physical quantity can be associated to any rigid body in contact with a planar surface, regardless of the fact that the body itself is a foot or not. Remarkably, the FRI (differently from the CoP) is not constrained in the contact convex hull because f_o and μ_o are not the result of unilateral contact forces. Other names used in literature for the FRI are fictitious ZMP (FZMP) and computed ZMP (CZMP) used by **Vukobratovic and Borovac** (2004) and **Kajita and Espiau** (2008) respectively. If the FRI is not within the contact convex hull, **Goswami** (1999) has shown that a rotation of the rigid link is occurring (i.e. angular acceleration is not identically zero). Viceversa, if the FRI is within the contact convex hull and if frictional constraints are satisfied², then the unilateral contact is stable (i.e. the contact link has null accelerations). The complete proof of the latter statement requires some additional considerations which are outside the scope of the present paper. The interested reader should refer to chapter eleven of **Featherstone** (2008).

Both the ZMP and the FRI concept have been used by several authors to define a suitable stability margin for balancing an articulated rigid body system. **Hirai et al.** (1998) used the ZMP concept to balance one of the earliest versions of the Honda walking humanoids. **Huang et al.** (2001) adopted a similar concept to define a stability margin tunable by modifying the robot hip motion. **Li et al.** (1993) used the error between a desired and the computed ZMP to learn stable walking. A good reason to prefer the FRI has been pointed out by **Goswami** (1999): given that “the ZMP cannot distinguish between the marginal state of static equilibrium and a complete loss of equilibrium of the foot (in both cases the ZMP is situated at the support boundary), its utility in gait planning is limited. FRI point, on the other hand, may exit the physical boundary of the support polygon and it does so whenever the foot is subjected to a net rotational moment.”.

¹ Normal linear motion and tangential angular motions are constrained by the unilateral contact forces; tangential linear motions and normal angular motion are constrained by friction.

² The FRI and ZMP (as they have been defined) do not depend neither on the tangential forces nor on the normal moments. However, frictional constraints depend on these quantities. Therefore, no stability criteria can be formulated using only the FRI and the ZMP quantities. Within this context, it comes with no surprise that the sufficient condition for contact stability requires tangential forces and normal moments to be within the friction cones of the contact itself.

2.3 MULTIPLE CONTACTS

So far we discussed the stability of an articulated rigid body system subject to a single planar unidirectional contact. The stability characterization given by **Goswami** (1999) guarantees contact stability by requiring the FRI to stay inside the support convex hull. In this section we review extensions of this criterium to the case of multiple contacts. Proposed extensions search for some global stability criteria to condense the local stability criteria on individual contacts. Section 2.3.1 considers the simple case of multiple coplanar contacts and the associated global stability criteria, known as global CoP. Section 2.3.2 reviews previous literature on global criteria with multiple noncoplanar contacts. The present section concludes by observing that, for the scope of the present paper, it is mandatory to abandon global criteria and stick to local ones. The conclusion follows from a counterexample, provided in Appendix A.2 and discussed in Section 2.3.3.

2.3.1 Multiple coplanar contacts. In this section we consider the case of articulated rigid body systems in contact with a flat surface (typically the ground). Differently from the previous sections, we assume that more than one single rigid body is in contact with the flat ground and therefore we consider a multiple coplanar contacts scenario. Within this context, we distinguish between local CoP (one per each rigid body in contact with the ground) and global CoP (the center of pressure resulting from all rigid bodies in contact). The local CoP of a rigid body has been defined and characterized in Section 2.1.1, 2.2.2 and 2.2.3. These definitions and characterizations refer to a single rigid body (e.g. the foot) but can be extended to any rigid body of the articulated system. Computations in this case account only for the contact forces acting on the rigid body itself. Global CoP (GCoP) is instead a quantity associated with the whole articulated system and corresponds to the center of pressure obtained by integrating all contact forces acting on the articulated system. Most of the previous literature does not distinguish between local and global center of pressure but often refer to the latter when characterizing stability during the double support phase of flat terrain walking. Remarkably, the property of the GCoP lying inside the contacts convex hull still holds. This stability criterion has been used by several authors (**Huang et al.** (2001); **Wieber** (2002); **Stonier and Kim** (2006)) to infer stability in flat terrain walking. In particular, **Wieber** (2002) defined a motion to be realizable if and only if the GCoP lies inside the convex hull of contact points. Even though **Popovic and Herr** (2005) questioned the use of the GCoP as a way to guarantee postural stability, associated criteria are at present the most adopted for planning walking trajectories. In Section 2.3.3 we further question the GCoP as a stability criteria, focusing in particular on the scope of the current paper.

2.3.2 Mutiple noncoplanar contacts. **Harada et al.** (2003) defined a generalized ZMP (GZMP) and a projected convex hull to formulate stability conditions for a limited class of arm/leg coordination tasks. **Sardain and Bessonnet** (2004) proposed a concept of virtual surface and virtual CoP-ZMP limited to the case of two noncoplanar contacts. In spite of the adopted simplification, authors themselves admit their failure in finding an associated pseudo support-polygon onto which the pseudo-ZMP stays. **Hyon et al.** (2007) presented a framework for computing joint torques that optimally distributes forces across multiple contacts; conditions for the CoP to lie within the supporting convex hull are formulated but stability conditions are not formulated with sufficient level of details.

2.3.3 Global versus local CoP. In the present paper, we formulate a whole-body postural control which assumes stable contacts. Stability, as defined in Section 2.2.1 guarantees time invariance of motion constraints and avoids the complications of controlling hybrid systems. Necessary and sufficient conditions for stability of individual contacts can be obtained by resorting to the FRI of each contact. Most of previous literature on flat terrain walking postulates the GCoP to lie in the contacts convex hull as a stability criterium. This criterium is a necessary and sufficient condition for a whole body motion to be realizable as pointed out by **Wieber** (2002) (see in particular section 3.2 of his paper). However, it is not a sufficient condition to guarantee stability of all contacts. Appendix B provides a counterexample in the simple case of two coplanar contacts: the GCoP is shown to lie in the contacts convex hull but individual

contacts are proven to be unstable. We therefore decide in this paper to stick to local contact stability criteria since no previous global criteria guarantee the stability of all local contacts.

2.4 TASK-SPACE INVERSE DYNAMICS

We now briefly review the vast literature on prioritized task-space inverse-dynamics control and we motivate our choices in this regard. **Sentis** (2007) and **Park** (2006) have been pioneers in the control of articulated free-floating rigid bodies exploiting the operational-space framework **Khatib** (1987). More recent approaches have explored the idea of simplifying the system dynamic equations by performing suitable projections onto the null space of the contact forces as proposed by **Righetti et al.** (2011); **Aghili** (2005). While being computationally efficient (i.e. total computation time below 1 ms), all these approaches share a common drawback: contact forces cannot be controlled. As a consequence, stability of the contacts cannot be guaranteed, which may lead the robot to tip over and fall.

As opposed to these analytical solutions to the control problem, an alternative numerical approach proposed by **de Lasa et al.** (2010) is to use a Quadratic Programming solver. This allows to include inequality constraints into the problem formulation, which can model control tasks and physical constraints (e.g. joint limits, motor-torque bounds, force friction cones). Even if this technique can guarantee contact stability, solving a cascade of Quadratic Programs with inequality constraints can be critical from a computational standpoint.

We decided to take an in-between approach: our framework of choice (see Section 3.4 or **Del Prete et al.** (2014) for details) allows to control the contact forces, but with a computational complexity of the same order of inverse-dynamics-based methods. Compared to optimization-based methods, our implementation does not allow for inequality constraints. To the best of our knowledge, the only real-time implementation of a cascade of Quadratic Programs with inequalities has been tested with a 14-DoF robot on a fast 3.8 GHz CPU **Herzog et al.** (2013). We cannot be sure that this method will be fast enough for 26 DoFs and/or a slower CPU. For this reason, while we think that using inequalities could be useful, we postponed it to the (near) future because we know that it demands for an efficient and careful software implementation.

3 MATERIALS & METHODS

In this section we present our approach to solve the problem of controlling whole-body posture on multiple rigid planar contacts. We suppose each contact to be planar, but contact planes to be in general noncoplanar. Within this context, the considerations presented in the previous sections justify our choice to abandon the idea of defining a global stability criterion (such as the GCoP). In case of noncoplanar contacts, a global CoP is not even properly defined given that the resulting force and torque might not in general be orthogonal (see Section 2.1.2 and Appendix A.2). In any case, the counterexample in Appendix B.1 suggests to consider multiple local stability criteria instead of a single global one. Local contact stability has been defined in Section 2.2.1 and it has been characterized as a condition for motion constraints to be time invariant. At present, necessary and sufficient conditions for contact stability have been formulated only in the case of planar rigid unilateral contacts (see Section 2.2.3). This is the reason why the scope of the current paper is limited to multiple planar contacts on rigid noncoplanar surfaces. Future extensions of the present work are foreseen in the direction of characterizing contact stability in more general situations.

In the rigid and planar contact case, contact stability has been characterized by means of the contact FRI and CoP. Both quantities depend on the wrench (i.e. both force and torque) at the contact point. Assuming that contacts might occur at any point on the robot body, an estimate of the contact wrench might be difficult to obtain if not impossible. Conventional manipulators measure wrenches at the end-effector, where force and torque sensors are placed. Joint torque sensing gives only an incomplete characterization of contact wrenches. The problem can be solved adopting whole-body distributed force/torque (F/T) and tactile sensors as those integrated in the iCub humanoid (Section 3.1). This specific design choice calls for

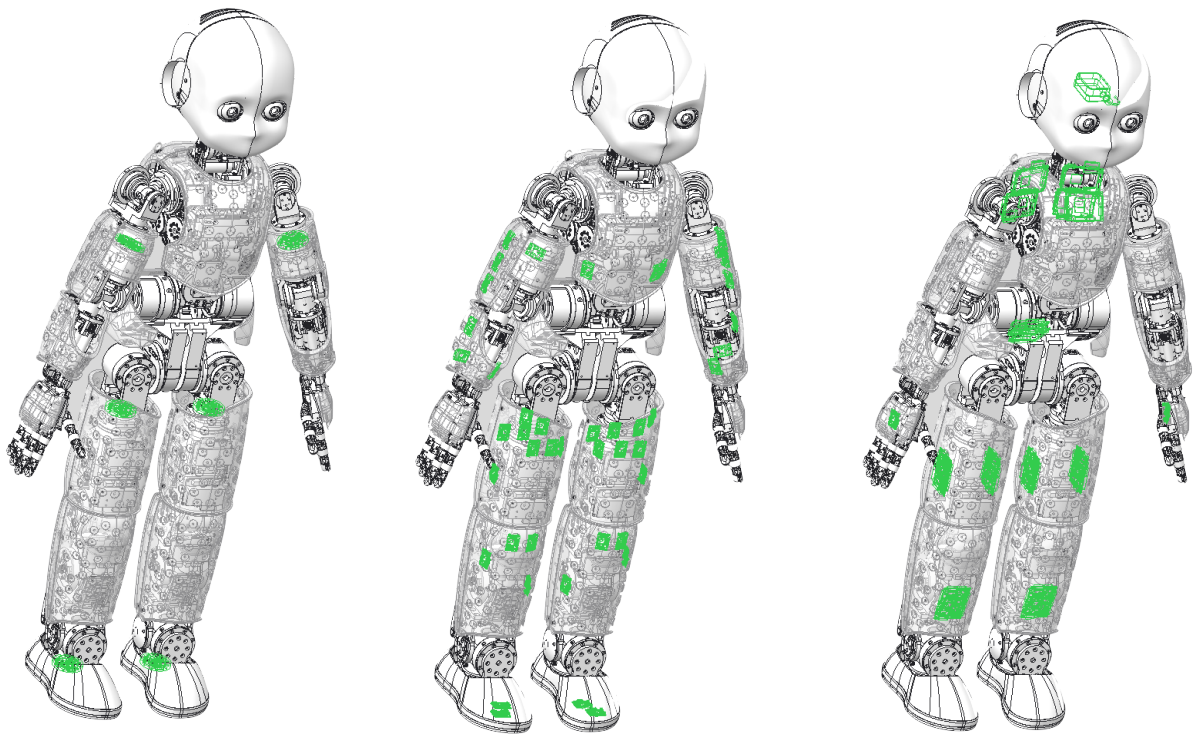


Figure 2. Mechanical schemes of the humanoid robot iCub with force/torque sensors, gyroscopes and accelerometers highlighted in green. **Left:** locations of the six proximal six-axis F/T sensors (legs and arms). **Center:** locations of the skin microcontrollers, which have a 3D accelerometer embedded. **Right:** locations of the motor microcontrollers and the commercial inertial sensor, with the latter having a 3D gyroscope and a 3D accelerometer embedded.

273 custom algorithms for: contact wrench estimation (Section 3.2.1), internal torques measurement (Section
 274 3.2.2) and dynamic model identification (Section 3.3). Whole-body control with multiple noncoplanar
 275 contacts is discussed in Section 3.4 and requires all the above custom components for its implementation
 276 (Section 4).

3.1 WHOLE-BODY DISTRIBUTED WRENCH AND CONTACT SENSING

277 The platform used to perform experimental tests is the iCub humanoid robot which is extensively
 278 described in **Metta et al.** (2010). One of the main features of this system is represented by the large variety
 279 of sensors, which include whole-body distributed F/T sensors, accelerometers, gyroscopes (see Fig. 2) and
 280 pressure sensitive skin. Furthermore the robot possesses two digital cameras and two microphones. From
 281 a mechanical standpoint iCub is 104 cm tall and has 53 degrees of freedom: 6 in the head, 16 in each
 282 arm, 3 in the torso and 6 in each leg. All joints but the hands and head are controlled by brushless electric
 283 motors coupled with harmonic drive gears. During experimental tests we mainly exploit two kinds of
 284 sensors: the F/T sensors and the distributed sensorized skin. The F/T sensor described in **Fumagalli et al.**
 285 (2012) is a 6-axis custom-made sensor that is mounted in both iCub's arms between the shoulders and
 286 elbows and in both legs between the knees and hip and between the ankles and feet. This solution allows
 287 to measure internal reaction forces, which in turn can be exploited to estimate both the internal dynamics
 288 and external forces exerted on its limbs. The robot skin **Cannata et al.** (2008) **Maiolino et al.** (2013) is
 289 a compliant distributed pressure sensor composed by a flexible printed circuit board (PCB) covered by
 290 a layer of three dimensionally structured elastic fabric further enveloped by a thin conductive layer. The
 291 PCB is composed by triangular modules of 10 taxels which act as capacitance gauges plus two temperature
 292 sensors for drift compensation. In our experiments, iCub's upper body was wrapped with approximately
 293 2000 sensors, each foot sole is covered with 250 taxels, while 1080 further sensors are at the last design

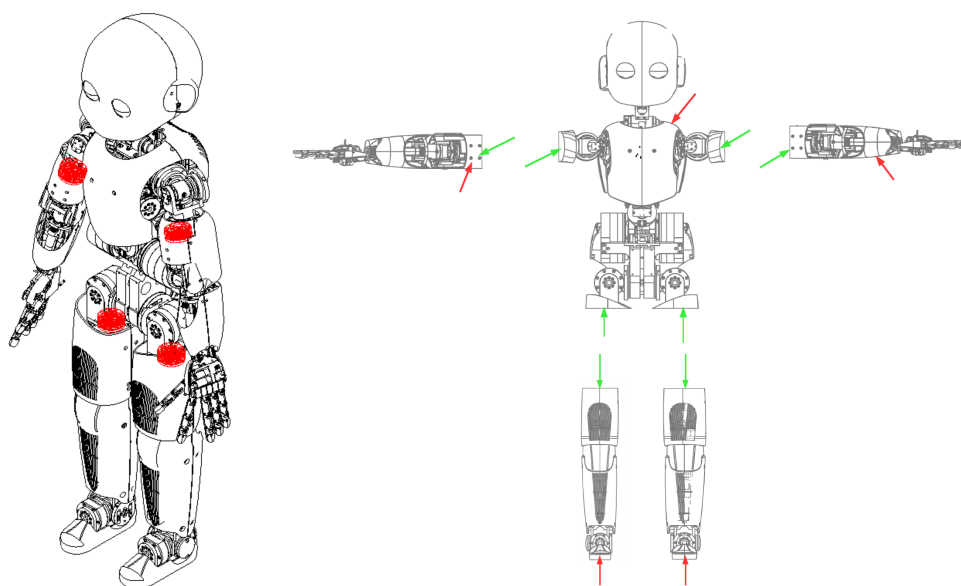


Figure 3. The left picture shows the location of four (out of six) F/T sensors on the iCub humanoid (sensors at the feet are omitted in this picture). The right picture shows the induced iCub kinematic tree partitioning. Each obtained subpart can be considered an independent floating-base structure subject to an external wrench which coincides with the one measured by the F/T sensor (green arrow). Red arrows represent possible location for the unknown external wrenches.

and integration stage on the lower body. Each single taxel has 8 bits of resolution, and measurements can be provided as raw data or as thermal drift compensated.

3.2 INTERNAL AND EXTERNAL (CONTACT) WRENCH ESTIMATION

Fumagalli et al. (2012) proposed a theoretical framework that exploits embedded F/T sensors to estimate internal/external forces acting on floating-base kinematic trees with multiple-branches. From a theoretical point of view, the proposed framework allows to virtually relocate the available F/T sensors anywhere along the kinematic tree. The algorithm consists in performing classical Recursive Newton-Euler algorithm (RNEA) steps with modified boundary conditions, determined by the contact and F/T sensor location. It can be shown that relocation relies solely on inertial parameters, velocities and accelerations of the rigid links in between the real and virtual sensors (see in particular the experimental analysis conducted by **Randazzo et al. (2011)**). The proposed algorithm consists in cutting the floating-base tree at the level of the (embedded) F/T sensors obtaining multiple subtrees as in Fig. 3. Then, each subtree is an independent articulated floating-base structure governed by the Newton-Euler dynamic equations. The F/T sensor, gives a direct measurement of one specific external wrench acting on the structure (green arrows in Fig. 3). Other external wrenches (red arrows in Fig. 3 and Fig. 4) can be estimated with the procedure hereafter described.

3.2.1 Method for estimating external wrenches. We now describe a method for the estimation of contact wrenches; more details can be found in **Del Prete et al. (2012)**. Let us consider a kinematic chain composed by N links, having a F/T sensor at the base (see Fig. 4), where w_i is the wrench (i.e. force and moment) exerted from link i to link $i + 1$, \ddot{p}_{c_i} is the acceleration of the center of mass of link i and m_i is the mass of link i . We know w_0 (i.e. the F/T sensor measurement), the K contact locations $r_{0,ei}, \dots, r_{0,eK}$ (i.e. the locations where the skin senses contacts), and we want to estimate the K contact wrenches w_{e1}, \dots, w_{eK} . Writing Newton's and Euler's equations for each rigid link and summing up all the N resulting equations we obtain:

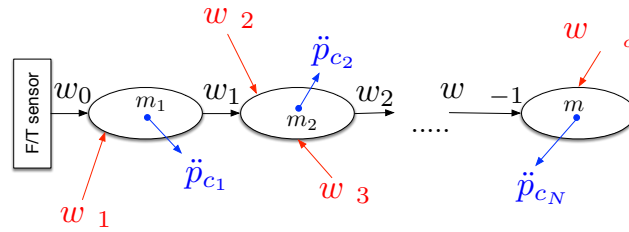


Figure 4. Sketch of a kinematic chain with an embedded F/T sensor. Although the sketch refers to a serial chain, the theoretical framework holds also in the case of multiple branches articulated chains (see for example the torso subchain in Fig. 3).

$$f_0 + \sum_{i=1}^K f_{ei} = \sum_{i=1}^N m_i \ddot{p}_{c_i}, \quad (1)$$

$$\mu_0 + \sum_{i=1}^K (\mu_{ei} + r_{0,ei} \times f_{ei}) = \sum_{i=1}^N (r_{0,ci} \times m_i \ddot{p}_{c_i} + I_i^i \dot{\omega}_i + \omega_i \times I_i^i \omega_i), \quad (2)$$

where I_i^i is the inertia of link i , ω_i and $\dot{\omega}_i$ are the angular velocity and acceleration of link i respectively, and $r_{0,ci}$ is the vector connecting the chain base to the center of mass of link i . Noting that in (1) and (2) the only unknowns are the contact wrenches, the estimation problem may be solved rewriting these equations in matrix form $Ax = b$, where $x \in \mathbb{R}^u$ contains all the u contact unknowns, whereas $A \in \mathbb{R}^{6 \times u}$ and $b \in \mathbb{R}^6$ are completely determined. The equations are constructed taking into account the type of possible contacts among the following three: pure wrench (w_e , 6-dimensional vector corresponding to force and torque); pure force (f_e , 3-dimensional vector corresponding to a pure force and no torque); force norm ($\|f_e\|$, one-dimensional unknown assuming the force to be orthogonal to the contact surface). In the simplest case, only a single contact acts on the sub-chain and the associated pure wrench can be uniquely determined (system of six equations and six unknowns). In other cases, a solution can be obtained with the following least squares procedure. The matrix A is built by attaching columns for each contact according to its type. The columns associated to pure wrenches (A_w), pure forces (A_f) and force norm (A_n) are the following:

$$A_w = \begin{bmatrix} I & 0 \\ S(r_{0,en}) & I \end{bmatrix}, \quad A_f = \begin{bmatrix} I \\ S(r_{0,en}) \end{bmatrix}, \quad A_n = \begin{bmatrix} \hat{u}_n \\ r_{0,en} \times \hat{u}_n \end{bmatrix}.$$

where $S(v) \in \mathbb{R}^{3 \times 3}$ is the skew-symmetric matrix such that $S(v)z = v \times z$, with \times denoting the cross product operator, and \hat{u}_n is the versor of the contact force f_{en} . The matrix A mainly depends on the skin spatial calibration, which can be obtained and refined with the procedure described by **Del Prete et al.** (2011). The 6-dimensional vector b is defined as:

$$b = \begin{bmatrix} f_b \\ \mu_b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^N m_i \ddot{p}_{c_i} - f_0 \\ \sum_{i=1}^N (r_{0,ci} \times m_i \ddot{p}_{c_i} + I_i^i \dot{\omega}_i + \omega_i \times I_i^i \omega_i) - \mu_0 \end{bmatrix}$$

The vector b depends on kinematic quantities which can be derived for the whole-body distributed gyros, accelerometers and encoders. Details on how to estimate these quantities have been detailed by **Fumagalli et al.** (2012). Once A and b have been computed, we can use the equation $Ax = b$ for estimating

external wrenches. The equation defines a unique solution if there is exactly one unknown wrench on the considered subchain. In the case of interest for the present paper, an exact characterization of the interaction wrenches can be obtained if there exists exactly one contact wrench per each of the subchains obtained by the body structure partition induced by the F/T sensor positions (see Fig. 3). In all other situations, an exact estimate cannot be obtained but from a procedural point of view it is preferable to give a reasonable estimate of all the contact wrenches. The solution we adopted consists in computing the minimum norm x^* that minimizes the square error residual:

$$x^* = A^\dagger b$$

where A^\dagger is the Moore-Penrose pseudo-inverse of A . The method has been implemented as an extension of the iDyn library³ and it has been integrated with other software modules to create an efficient software system able to estimate internal and external wrenches of the whole iCub robot.

3.2.2 Method for estimating internal torques. Once an estimate of external forces is obtained with the method described in Section 3.2.1, internal wrenches can also be estimated with a standard Newton-Euler force propagation recursion. Projection of the internal wrenches on the joint axes provides an estimate of the joint torques τ . A torque controller with joint friction compensation guarantees that each motor provides the desired amount of torque to the joints. In order to improve the torque tracking performance a suitable identification procedure was adopted to estimate the voltage to torque transfer function for each motor. Details are given in the following section together with some details on the dynamic model identification.

3.3 DYNAMIC MODEL ESTIMATION

As it was previously pointed out, the technology available in the iCub (nominally, whole-body distributed tactile and F/T sensing) and the estimation algorithm presented in the previous sections allows to simultaneously estimate internal (i.e. joint torques) and external (i.e. contact) forces. The accuracy of the estimates is decisive for the efficacy of the control algorithm that will be described in Section 3.4. A key element to improve the estimation and control accuracy is the availability of a reliable dynamic model (masses, inertias and center of mass positions in equations (1) and (2)). Standard identification procedures do not apply directly and a customization to iCub specific sensor modalities and distributions is necessary. In the following two subsections we describe the solution that we implemented in order to improve dynamic model accuracy (Section 3.3.1) and torque tracking performances (Section 3.3.2).

3.3.1 Dynamic model identification. The accuracy of the system dynamics (7) is crucial in the proposed control framework since it affects the controller (8) and the internal/external torque estimation procedure described in Section 3.2. Individual dynamic parameters (mass, inertia and center of mass position) of the rigid bodies constituting an articulated chain can be directly obtained from CAD drawings. These parameters are often not sufficiently accurate and standard identification procedures such as those proposed in the handbook of robotics by **Hollerbach et al.** (2008) can be applied in order to improve modeling accuracy. Remarkably, these procedures do not give an estimate of individual parameters but some linear combination of them, known in literature as the *base parameters*. It remains therefore to be clarified if the base parameters suffice to implement the procedure described in Section 3.2.1 to estimate external wrenches. This procedure, written as it is, requires the knowledge of individual dynamic parameters as evident from equations (1) and (2). Additionally, it needs to be verified that also the procedure for estimating internal torques presented in Section 3.2.2 can be reformulated in terms of the base parameters only. Interestingly, it can be shown, resorting to the work by **Ayusawa et al.** (2014) that the base parameters are a subset of those used for both the estimation procedures in Section 3.2.

³ See the software library documentation http://wiki.icub.org/iCub_documentation/idyn_introduction.html.

364 **3.3.2 Modeling and representation of the motor transfer function.** Another important component in
 365 implementing the control strategy detailed in Section 3.4 is torque control. For the scope of the present
 366 paper, it was necessary to implement a model-based controller tuned for each motor. The model assumes
 367 that the i -th joint's torque τ_i is proportional (k_t) to the voltage V_i applied to the associated motor, with the
 368 additional contribution of some viscous (k_v) and Coulomb (k_c) friction :

$$V_i = k_t \tau_i + (k_{vp}s(\dot{\theta}_i) + k_{vn}s(-\dot{\theta}_i))\dot{\theta}_i + (k_{cp}s(\dot{\theta}_i) + k_{cn}s(-\dot{\theta}_i))\text{sign}(\dot{\theta}_i), \quad (3)$$

369 where $\dot{\theta}_i$ is the motor velocity, $s(x)$ is the step function (1 for $x > 0$, 0 otherwise) and $\text{sign}(x)$ is
 370 the sign function (1 for $x > 0$, -1 for $x < 0$, 0 for $x = 0$). Operationally, it was observed quite
 371 relevant to distinguish between positive and negative rotations as represented in the model above. We
 372 identified the coefficients $k_t, k_{vp}, k_{vn}, k_{cp}, k_{cn}$ for each joint with an automatic procedure implemented in
 373 an open-source software module⁴. The motor controller exploits the transmission model and implements
 374 the following control strategy:

$$V_i = k_t \left(\tau_i^d - k_p \tilde{\tau}_i - k_i \int \tilde{\tau}_i dt \right) + [k_{vp}s(\dot{\theta}_i) + k_{vn}s(-\dot{\theta}_i)]\dot{\theta}_i + [k_{cp}s(\dot{\theta}_i) + k_{cn}s(-\dot{\theta}_i)] \tanh(k_s \dot{\theta}_i), \quad (4)$$

375 where $\tanh(k_s \dot{\theta}_i)$ is used to smooth out the sign function, k_s is a user-specified parameter that regulates
 376 the smoothing action, and $\tilde{\tau}_i = \tau_i - \tau_i^d$ is the i -th torque tracking error, and $k_p, k_i > 0$ are the low-level
 377 control gains. The control objective for the torque controller consists in obtaining $\tau \simeq \tau^d$ and therefore
 378 the controller will make the assumption that the commanded value τ_i^d is perfectly tracked by the torque
 379 controller. Also, observe that there is no derivative term in the parenthesis on the right hand side of (4). In
 380 fact, the measurement of $\dot{\tau}_i$ is noisy and unreliable at the current state of the iCub's measuring devices.

381 **3.3.3 Differential joint torque and motion coupling.** Specific care was posed in controlling the torques
 382 at joints actuated with a differential mechanism. As an example, we consider here the torso roll, pitch and
 383 yaw joint represented in Fig. 5. In particular, define $q \in \mathbb{R}^3$ the vector of the joint angles Corresponding to
 384 the torso yaw, the torso roll, and torso pitch degree of freedom. Also, by abusing notation, define $\theta \in \mathbb{R}^3$
 385 as the angles between the stator and the rotor of the motors 0B4M0, 0B3M0, 0B3M1. Then, a simple
 386 analysis leads to the following relationship :

$$\dot{q} = T \dot{\theta}, \quad T =: \begin{bmatrix} \frac{r}{R} & \frac{r}{2R} & \frac{r}{2R} \\ 0 & 0.5 & 0.5 \\ 0 & -0.5 & 0.5 \end{bmatrix}, \quad (5)$$

387 where r and R are the radius of the pulleys sketched in Fig. 5. The above matrix T is obtained
 388 by combining a classical differential coupling between pitch and yaw (first two rows) with a more
 389 complicated coupling with the roll motion (third row, see Fig. 6). Defining τ_q to be the link torques
 390 and τ_θ the motor torques, the coupling induced on torques can be easily obtained by imposing the equality
 391 between link and motor powers:

$$\dot{q}^\top \tau_q = \dot{\theta}^\top \tau_\theta \quad \forall \dot{q}, \dot{\theta} \quad \Rightarrow \quad \tau_q = T^{-\top} \tau_\theta \quad \text{with} \quad T^{-\top} := \begin{bmatrix} \frac{R}{r} & 0 & 0 \\ -1 & 1 & 1 \\ 0 & -1 & 1 \end{bmatrix}. \quad (6)$$

392 In the case of coupled joints, the transformation matrix T is used in both (3) and (4) which hold
 393 at the motor level. Since position (\dot{q}) and torque (τ) feedback is available at the joint level, suitable

⁴ <https://github.com/robotology/codyco-modules/tree/master/src/modules/motorFrictionIdentification>

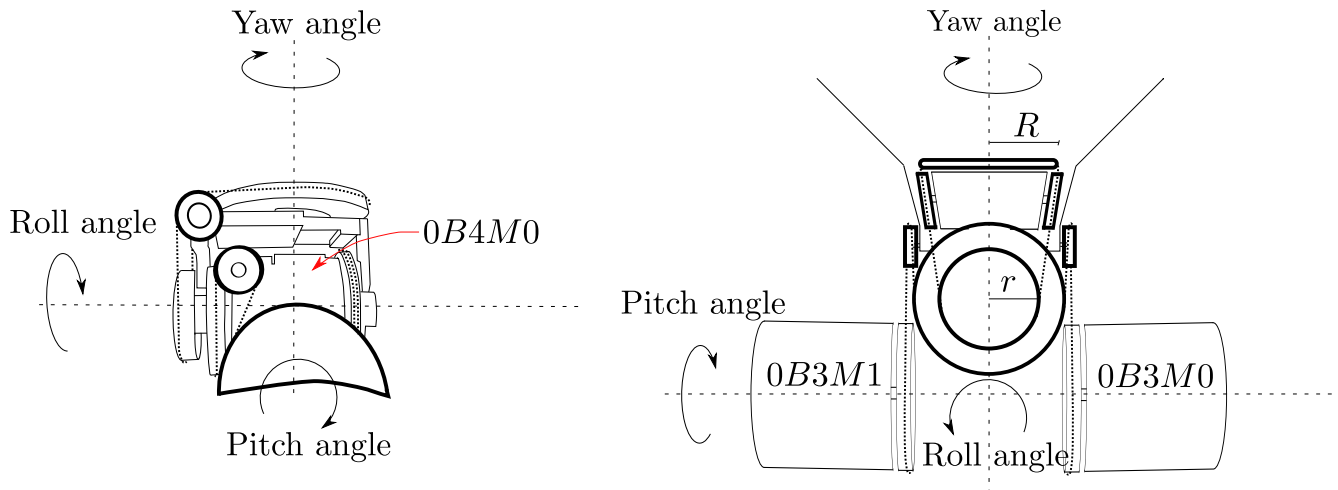


Figure 5. The sketch represents the differential joint at the iCub torso. The pitch, yaw and roll joints are actuated with three motors 0B4M0, 0B3M0, 0B3M1 in differential configuration. Motor ($\theta \in \mathbb{R}^3$) and joint ($q \in \mathbb{R}^3$) positions are coupled as described in Fig. 6.

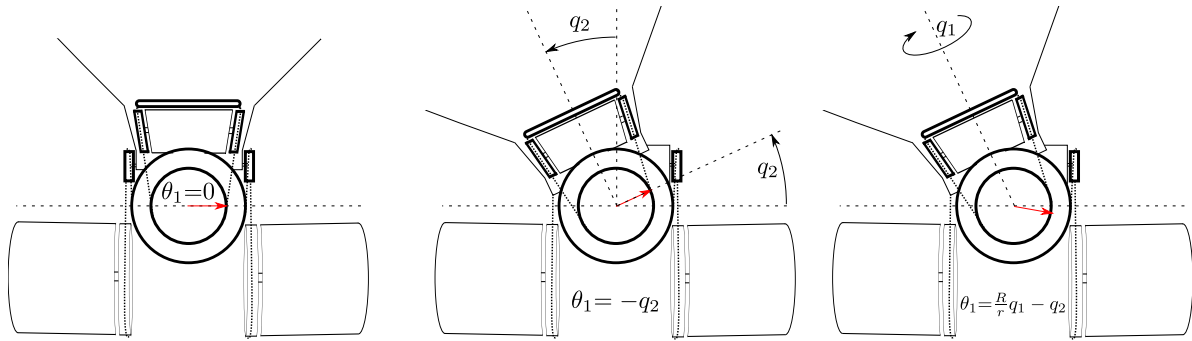


Figure 6. The sketch is used to represent the kinematic coupling between the yaw and roll movements. A roll movement of an angle q_2 implies an equal and opposite movement of the rotor of the motor 0B4M0. This rotor moves, modulo the transmission ratio, also when a yaw movement of an angle q_1 occurs.

transformations need to be applied. In the case of coupled joints, our current software⁵ implements joint per joint equations (3) and (4) with substitutions $\tau \leftrightarrow \tau_\theta$ and $\dot{q} \leftrightarrow \dot{\theta}$ where motor velocity ($\dot{\theta}$) and motor torques τ_θ are obtained from joint velocity (\dot{q}) and joint torques (τ_q) as follows:

$$\dot{\theta} = T^{-1}\dot{q}, \quad \tau_\theta = T^\top \tau_q.$$

3.4 COMPLETE FORCE CONTROL

In this section we describe the control algorithm for controlling an articulated rigid body subject to multiple rigid constraints. The system dynamics are described by the following constrained differential equations:

⁵ Both the identification (3) and control (4) are available with an open source license. See the documentation in http://wiki.icub.org/codyco/dox/html/group__motorFrictionIdentification.html and http://wiki.icub.org/codyco/dox/html/group__codyco__jointTorqueControl.html respectively.

$$\underbrace{\begin{bmatrix} M_b & M_{bj} \\ M_{bj}^\top & M_j \end{bmatrix}}_{M(q)} \underbrace{\begin{bmatrix} \dot{v}_b \\ \ddot{q}_j \end{bmatrix}}_{\dot{v}} + \underbrace{\begin{bmatrix} h_b \\ h_j \end{bmatrix}}_{h(q,v)} - \underbrace{\begin{bmatrix} J_{cb}^\top \\ J_{cj}^\top \end{bmatrix}}_{J_c(q)^\top} f = \underbrace{\begin{bmatrix} 0_{6 \times n} \\ I_{n \times n} \end{bmatrix}}_{S^\top} \tau \quad (7a)$$

$$J_c(q)\dot{v} + \dot{J}_c(q,v)v = 0, \quad (7b)$$

where $q \in SE(3) \times \mathbb{R}^n$, with $SE(3)$ the special euclidian group, represents the configuration of the floating base system, which is given by the pose of a base-frame (belonging to $SE(3)$) and n generalized coordinates (q_j) characterizing the joint angles. Then, $v \in \mathbb{R}^{n+6}$ represents the robot velocity (it includes both $\dot{q}_j \in \mathbb{R}^n$ and the floating-base linear and angular velocity $v_b \in \mathbb{R}^6$), \dot{v} is the derivative of v , the control input $\tau \in \mathbb{R}^n$ is the vector of joint torques, $M \in \mathbb{R}^{(n+6) \times (n+6)}$ is the mass matrix, $h \in \mathbb{R}^{n+6}$ contains both gravitational and Coriolis terms, $S \in \mathbb{R}^{n \times (n+6)}$ is the matrix selecting the actuated degrees of freedom, $f \in \mathbb{R}^k$ is the vector obtained by stacking all contact wrenches, which implies that $k = 6N_c$ and N_c the number of (rigid) contacts, $J_c \in \mathbb{R}^{k \times (n+6)}$ is the contact Jacobian. Let us first recall how the force-control problem is solved in the Task Space Inverse Dynamics (TSID) framework proposed by **Del Prete** (2013) in the context of floating-base robots. The framework computes the joint torques to match as close as possible a desired vector of forces at the contacts (8a) while being compatible with the system dynamics (8b) and contact constraints (8c):

$$\tau^* = \arg \min_{\tau \in \mathbb{R}^n} \|f - f^*\|^2 \quad (8a)$$

$$s.t. \quad M\dot{v} + h - J_c^\top f = S^\top \tau \quad (8b)$$

$$J_c\dot{v} + \dot{J}_c v = 0 \quad (8c)$$

where $f^* \in \mathbb{R}^k$ is the desired value for the contact forces. Then we can exploit the null space of the force task to perform $N - 1$ motion tasks at lower priorities. These tasks (indexed with $i = 1, \dots, N - 1$) are all represented as the problem of tracking a given reference acceleration \ddot{x}_i^* for a variable x_i differentially linked to q by the Jacobian J_i as follows:

$$\dot{x}_i = J_i v, \quad \ddot{x}_i = J_i \dot{v} + \dot{J}_i v. \quad (9)$$

Assuming that the force task has maximum priority the solution is:

$$\tau^* = -(J_c \bar{S})^\top f^* + N_j^{-1} \dot{v}_1^* + \bar{S}^\top n, \quad (10)$$

where $N_j^{-1} = M_j - M_{bj} M_j^{-1} M_{bj}^\top$, $\bar{S} = \begin{bmatrix} -M_{bj}^\top M_b^{-1} & I \end{bmatrix}^\top$ and the term \dot{v}_1^* is computed solving the following recursion for $i = N, \dots, 1$:

$$\begin{aligned} \dot{v}_i &= \dot{v}_{i+1} + (J_i \bar{S} N_{p(i)})^\dagger (\ddot{x}_i^* - \dot{J}_i v + J_i (U^\top M_b^{-1} (h_b - J_{cb}^\top f) - \bar{S} \dot{v}_{i+1})) \\ N_{p(i)} &= N_{p(i+1)} - (J_{i+1} \bar{S} N_{p(i+1)})^\dagger J_{i+1} \bar{S} N_{p(i+1)}, \end{aligned} \quad (11)$$

where $U \in \mathbb{R}^{6 \times (n+6)}$ is the matrix selecting the floating-base variables, and the algorithm is initialized setting $\dot{v}_{N+1} = 0$, $N_{p(N)} = I$, $J_N = J_c$ and $\ddot{x}_N = 0$. The implementation of this controller exploits the fact that we can compute (10) with an efficient hybrid-dynamics algorithm.

4 RESULTS

4.1 SET OF ADMISSIBLE TASKS

The final validation of the proposed control framework requires the definition of a suitable set of position (\ddot{x}_i^*) and wrenches (w^*) tasks and their relative priority. The set of admissible tasks is quite flexible also considering the flexibility of the underlying software libraries⁶. Nevertheless we list here a set of possible tasks, which we will use as a reference in the following sections. Quantities are defined with a notation similar to the one used by **Featherstone** (2008): H denotes the total spatial momentum of an articulated rigid body (including linear and angular), w indicates a wrench (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 la, ra, lf and rf indicate reference frames rigidly attached to the left arm, right arm, left foot and right foot respectively, the superscript i indicates the reference frame attached to the i -th rigid body, kX_i represents the rigid motion vector transformation from the reference frame i to the reference frame j , ${}^jX_i^*$ represents the force vector transformation from the reference frame i to the reference frame j , q_j represents the angular position of 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{x}_i or w^*) and associated Jacobians (J_i).

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

$$\begin{aligned} w_i^* & : & w_{rf} &= w_{rf}^*; \\ J_i & : & J_{rf} &; \end{aligned}$$

- T_f^{lf} : left foot wrench task. Regulates the left foot interaction wrench to a predefined value:

$$\begin{aligned} w_i^* & : & w_{lf} &= w_{lf}^*; \\ J_i & : & J_{lf} &; \end{aligned}$$

- T_f^{ra} : right arm wrench task. Regulate the right arm interaction wrench to a predefined value:

$$\begin{aligned} w_i^* & : & w_{ra} &= w_{ra}^*; \\ J_i & : & J_{ra} &; \end{aligned}$$

- T_f^{la} : left arm wrench task. Regulates the left arm interaction wrench to a predefined value:

$$\begin{aligned} w_i^* & : & w_{la} &= w_{la}^*; \\ J_i & : & J_{la} &; \end{aligned}$$

- T^q : postural task. Maintains the robot joints q_j close to certain reference posture q_j^* :

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

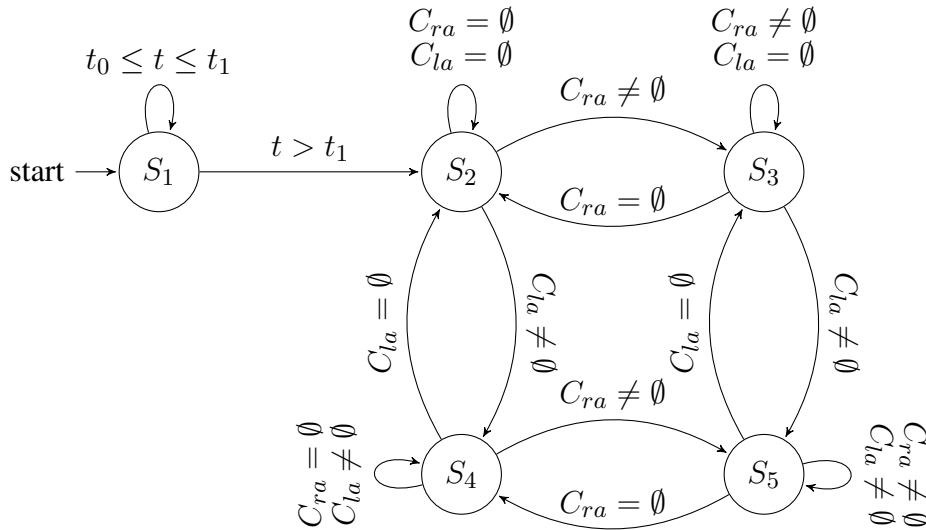
⁶ http://wiki.icub.org/codyco/dox/html/group__iDynTree.html

4.2 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 $\mathcal{S}_1, \dots, \mathcal{S}_5$.

- S_1 has the following set of active tasks $\mathcal{S}_1 = \{T^q\}$.
- S_2 has the following set of active tasks $\mathcal{S}_2 = \{T_f^{lf}, T_f^{rf}\} \cup \mathcal{S}_1$.
- S_3 has the following set of active tasks $\mathcal{S}_3 = \{T_f^{ra}\} \cup \mathcal{S}_2$.
- S_4 has the following set of active tasks $\mathcal{S}_4 = \{T_f^{la}\} \cup \mathcal{S}_2$.
- S_5 has the following set of active tasks $\mathcal{S}_5 = \{T_f^{la}, T_f^{ra}\} \cup \mathcal{S}_2$.

Transition between states is regulated by the following finite state machine where the sets C_{ra} and C_{la} contain the taxels (tactile elements) activated at time t .



In practice, at start ($t = t_0$) the robot is in state S_1 in order to maintain a configuration which is as close as possible to the initial configuration with a postural task (T^q) that guarantees that the system is not drifting. After a predefined amount of time ($t = t_1$) the system switches to state S_2 by adding two tasks to the set of active tasks: a control of the forces exchanged by the left and right foot (T_f^{rf} , T_f^{lf} respectively). The control of these forces allows for a direct control over the rate of change of the momentum as will be explained in Section 4.3. Successive transitions are triggered by the tactile sensors. If no contact is detected on the right and left arm ($C_{ra} = \emptyset$ and $C_{la} = \emptyset$ 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 arm ($C_{ra} \neq \emptyset$): in this state the active tasks are the same active in S_2 with the addition of the task responsible for controlling the force at the right arm contact location (T_f^{ra}). Similarly, a transition from S_2 to S_4 is performed when the system detects a contact on the left arm ($C_{la} \neq \emptyset$): in this state the active tasks are the same active in S_2 with the addition of a task responsible for controlling the force at the left arm contact location (T_f^{la}). Finally a transition from either S_3 or S_4 to S_5 is performed whenever the robot perceives a contact so that

in this new situation both arms are in contact ($C_{ra} \neq \emptyset$ and $C_{la} \neq \emptyset$ respectively). In S_5 both arms are used to control the interaction forces by activating the tasks T_f^{ra} and T_f^{la} .

4.3 TASK REFERENCES

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

$$w_{ra}^*, \quad w_{la}^*, \quad w_{rf}^*, \quad w_{lf}^*, \quad \ddot{q}_j^*.$$

to be used in the controller (8). Instantaneous values for forces are computed so as to follow a desired trajectory of the center of mass (x_{com}^d) and to reduce the system's angular momentum. Instantaneous values for \ddot{q}_j^* are chosen so as to follow a desired reference posture q_j^d . The latter is obtained by choosing:

$$\ddot{q}_j^*(t) = \ddot{q}_j^d(t) - K_d^q \left(\dot{q}_j - \dot{q}_j^d(t) \right) - K_p^q \left(q_j - q_j^d(t) \right), \quad (12)$$

where K_p^q and K_d^q are arbitrary positive-definite matrices that take into account that in the presence of modeling errors, the acceleration imposed on the system \ddot{q}_j might differ from the ideal one \ddot{q}_j^* . Instantaneous values for interaction forces are instead computed to follow a prescribed center-of-mass trajectory (x_{com}^d) and to reduce angular momentum. In order to do so, a reference value \dot{H}_{com}^* for the total rate of change of spatial momentum (expressed at the center of mass) is computed with a strategy similar to (12).

$$\dot{H}_{com}^*(t) = \dot{H}_{com}^d(t) - K_d^h \left(H_{com} - H_{com}^d(t) \right) - \begin{bmatrix} K_p^{com} (x_{com} - x_{com}^d(t)) \\ 0_{3 \times 1} \end{bmatrix}, \quad H_{com}^d = \begin{bmatrix} m \dot{x}_{com}^d \\ 0_{3 \times 1} \end{bmatrix} \quad (13)$$

where K_d^h , K_p^{com} are suitably defined gain matrices, H_{com} is the spatial momentum around the center of mass, x_{com} is the center-of-mass cartesian position, x_{com}^d its desired value and m is the total mass of the robot. Finally, values for f_{ra}^* , f_{la}^* , f_{rf}^* , f_{lf}^* can be computed from \dot{H}_{com}^* considering that the time derivative of the momentum equals the resultant of forces and torques if all quantities are computed with respect to the center of mass. The notation is slightly complicated due to the fact that in the different scenario states S_1, \dots, S_5 the meaning of f (and consequently f^*) in (8) changes. In particular we have:

$$S_2 : f = \begin{bmatrix} w_{rf} \\ w_{lf} \end{bmatrix}, \quad S_3 : f = \begin{bmatrix} w_{ra} \\ w_{rf} \\ w_{lf} \end{bmatrix}, \quad S_4 : f = \begin{bmatrix} w_{la} \\ w_{rf} \\ w_{lf} \end{bmatrix}, \quad S_5 : f = \begin{bmatrix} w_{ra} \\ w_{la} \\ w_{rf} \\ w_{lf} \end{bmatrix}, \quad (14)$$

In the different states, the following equations on f and \dot{H}_{com} always hold:

$$S_i : C_{S_i} f + f_g = \dot{H}_{com}, \quad (15)$$

where f_g is the gravitational force and where we defined C_{S_i} to be the matrix that expresses the spatial forces with respect to the center of mass:

$$\begin{aligned} C_{S_2} &= \begin{bmatrix} {}^{com}X_{rf}^* & {}^{com}X_{lf}^* \end{bmatrix}, \\ C_{S_3} &= \begin{bmatrix} {}^{com}X_{ra}^* & {}^{com}X_{rf}^* & {}^{com}X_{lf}^* \end{bmatrix}, \\ C_{S_4} &= \begin{bmatrix} {}^{com}X_{la}^* & {}^{com}X_{rf}^* & {}^{com}X_{lf}^* \end{bmatrix}, \\ C_{S_5} &= \begin{bmatrix} {}^{com}X_{ra}^* & {}^{com}X_{la}^* & {}^{com}X_{rf}^* & {}^{com}X_{lf}^* \end{bmatrix}. \end{aligned}$$

In literature these equations (derived from the Newton-Euler equations) have been presented in detail by **Orin et al.** (2013) under the name of *centroidal dynamics*. The constraints (15) on f given \dot{H}_{com} are not sufficient to identify a unique solution. Additional constraints or requirements need to be imposed in order to properly define f^* to achieve the desired momentum derivative. In order to get rid of this ambiguity, the following problem can be solved when at state S_i :

$$f^* = \arg \min_f \|f - f_0\|_W^2 \quad \text{s.t.} \quad C_{S_i} f + f_g = \dot{H}_{com}^*, \quad (16)$$

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

$$f^* = C_{S_i}^{\dagger W} \left(\dot{H}_{com}^* - f_g \right) + (I - C_{S_i}^{\dagger W} C_{S_i}) f_0, \quad C_{S_i}^{\dagger W} = W^{-1} C_{S_i}^\top \left(C_{S_i} W^{-1} C_{S_i}^\top \right)^{-1}. \quad (17)$$

This solution gives a set of desired forces f^* , which generate the desired momentum derivative \dot{H}_{com}^* . It is worth noting here that additional constraints should be imposed on the contact forces to guarantee contact stability. In particular, planar unilateral contacts should have an associated FRI lying in the contact support polygon (see Section 2.2.3) and forces should be maintained within the contact friction cones. Considering that these constraints can be approximated with a set of linear inequalities, adding them into (16) corresponds to transforming the problem into a quadratic program, as proposed by **de Lasa et al.** (2010). In the present implementation we follow a different strategy where stability constraints are enforced by solving (16) with a suitable choice for the reference value f_0 and weight matrix W . To ensure that reference contact forces are stable in the sense of Section 2.2.3 we express f in a reference frame whose origin coincides with the center of the maximum circle inscribed in the contact polygon. Choosing f_0 to have a null torque component penalizes solutions whose FRI is closer to the polygon borders and this penalty monotonically increases with the distance from the center of the maximum circle. Similarly, friction cones constraints are enforced by choosing the components of f_0 to be sufficiently far from the cone borders. Assuming contact plane normals to coincide with the z -axis, cone borders distance is maximized with null x and y components. If a good choice for the z -axis force components is available, it can be used in f_0 . Otherwise a viable choice is also to choose $f_0 = 0$ since in most realistic situations the solution f^* is dominated by the component f_g , which is always nonzero.

4.4 EXPERIMENTAL RESULTS

We implemented the proposed control strategy on the iCub humanoid. In a first phase, the iCub was balancing with both feet on the ground plane (coplanar flat contacts). The desired center of mass position

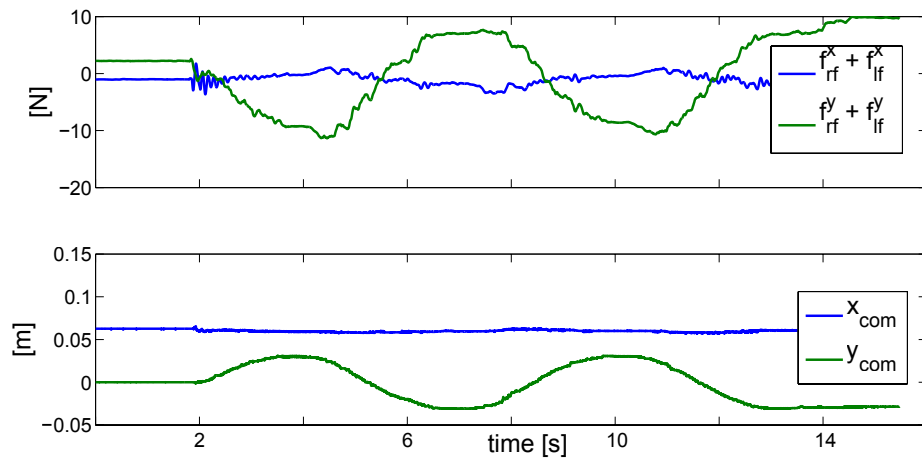


Figure 7. Results of the double support experiment on planar contacts (left and right feet). The picture shows the time behavior of forces (top) and center of mass position (bottom) on the sagittal (blue) and transverse (green) axis. It is worth noting that forces should be proportional to center of mass accelerations and this is visible in the plot considering that accelerations are sinusoidal in counter phase with positions. Rapid variations of the contact forces at the time $t \approx 2$, i.e. starting time, are due to the activation of the torque control.

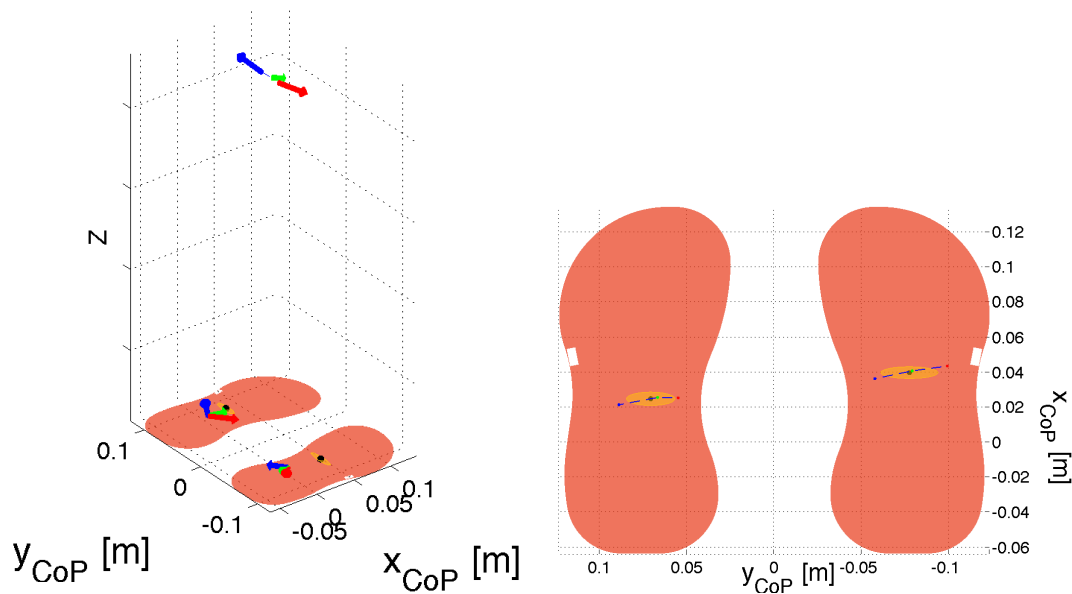


Figure 8. Results of the double support experiment on planar contacts (left and right feet). The left picture shows in three dimensions the feet contacts, the feet center of pressures, the forces at the feet and at the the center of mass during three instants: at two extrema of the sinusoid (red and blue) and in the middle of the sinusoid (green). Remarkably forces are maximum at the extrema when also accelerations are maximal. The right picture shows a close-up of of the feet with the trajectory of the center of pressure, an ellipse representing a Gaussian fit of the data points and three points corresponding to the position of the centers of pressure when at the two extrema of the sinusoid (red and blue) and in the middle of the sinusoid (green).

was moved left to right with a sinusoidal overimposed on its initial position $x_{com}(t_0)$ along the robot transverse axis (n):

$$x_{com}^d(t) = x_{com}(t_0) + n \cdot A \sin(2\pi f_r t) \quad (18)$$

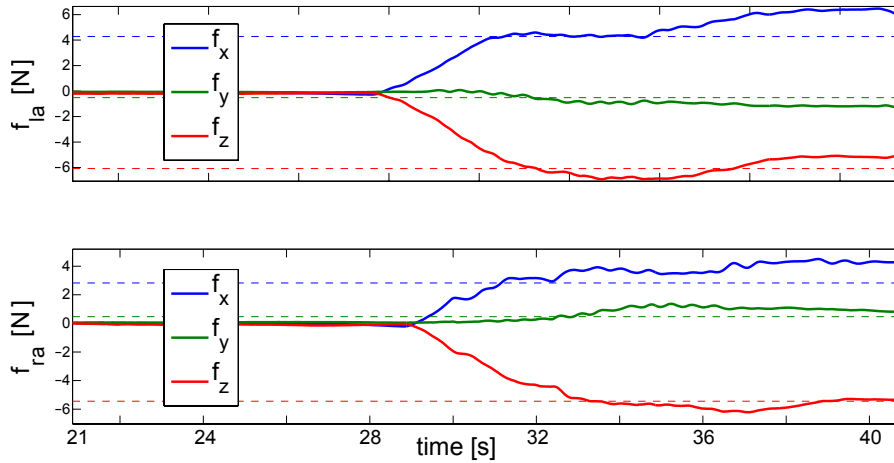


Figure 9. Results of the double support experiment on four noncoplanar contacts (both feet and arms). The plots represent the evolution of the contact forces at the left (top) and right (bottom) arms when contact is established.

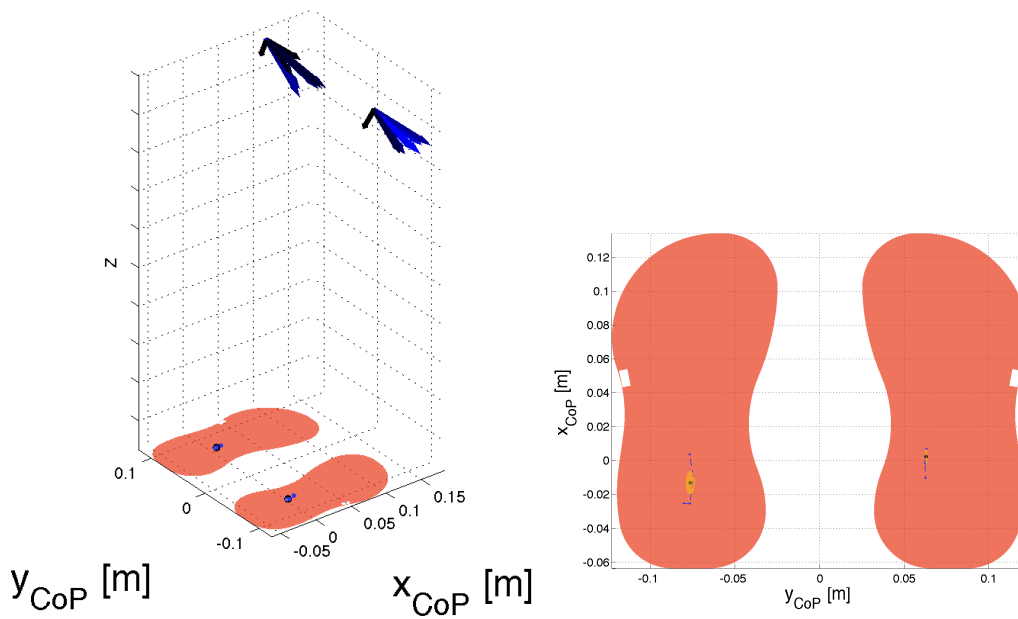


Figure 10. Results of the double-support experiment on four noncoplanar contacts (both feet and arms). The left picture gives a three-dimensional view of the foot center-of-pressure positions together with the arm contact forces. Forces are represented in a color scale that goes from black (contact establishment) to blue (steady state). The right picture gives a closeup on the foot center-of-pressure positions with an ellipse that represents the Gaussian approximation of its distribution.

513 with $f_r = 0.15Hz$ and $A = 0.02m$ (see Fig. 7). The reference posture $q_j^d(t)$ was maintained at its initial
 514 configuration $q_j^d(t_0)$. As previously described the desired center-of-mass acceleration was obtained by
 515 suitably choosing the forces at the contact points (in this case at the feet) as represented in Fig. 8. A video
 516 of the first phase of the experiment ⁷ is available for the interested reader).

⁷ https://www.youtube.com/watch?v=jaTEbCsFp_M.

In a second phase, the iCub was maintaining its center of mass at its initial position $x_{com}^d(t) \equiv x_{com}(t_0)$ and the joint reference posture $q_j^d(t)$ was chosen so as to move the arms toward a table in front of the robot. The controller (8) was regulated by the finite state machine described in Section 4.2. At the occurrence of contacts, forces at the arms were regulated to a predefined value f_d , which was obtained by imposing two additional constraints in solving (16): $w_{ra} = w_d$ and $w_{la} = w_d$. The force-regulation task at the arms is shown in Fig. 9 and Fig. 10, which also shows that the generation of forces at the arms does not affect the center of pressure at the feet.

5 CONCLUSIONS

The present paper addressed the problem of whole-body motion control in the presence of multiple non-coplanar and rigid contacts. The proposed solution defines a stability criterion based on the FRI of individual contacts as opposed to global stability criteria. It is argued that the FRI lying on the contact support polygon is a necessary and sufficient condition for the contacts to impose always the same motion constraints on the whole-body dynamics. These stability conditions are therefore the ones adopted in this paper in order to avoid the complications of hybrid and switching systems control. The chosen stability conditions require the capability of simultaneously measuring forces and torques (i.e. wrenches) at any possible contact location. This is not possible with conventional torque-controlled manipulators and requires whole-body distributed force and tactile sensing. These sensing capabilities are available in the iCub humanoid exploiting its whole-body distributed artificial skin and force/torque sensors. In consideration of this specific hardware, in the present paper we discussed our approach to obtain: contact-wrench estimates, internal-torque measurement and dynamic model identification. All these components are functional to the implementation of the proposed whole-body controller with multiple noncoplanar contacts.

ACKNOWLEDGEMENT

The authors acknowledge Ali Paikan (Istituto Italiano di Tecnologia, iCub Facility) for his contribution on the Lua bindings, Daniele Domenichelli (Istituto Italiano di Tecnologia, iCub Facility) for the codyco-superbuild support, Lorenzo Natale (Istituto Italiano di Tecnologia, iCub Facility) for the support to the YARP framework, Marco Randazzo (Istituto Italiano di Tecnologia, iCub Facility) for the implementation of torque control on the iCub humanoid, Julien Jenvrin (Istituto Italiano di Tecnologia, iCub Facility) for the technical hardware support and Giorgio Metta for the coordination of the iCub Facility activities.

Funding: This paper was supported by the FP7 EU projects CoDyCo (No. 600716 ICT 2011.2.1 Cognitive Systems and Robotics), and Koroibot (No. 611909 ICT-2013.2.1 Cognitive Systems and Robotics).

A COP EXISTENCE CONDITIONS

The Poinso theorem (see pag. 65 in Murray et al. (1994)) states that every wrench applied to a rigid body is equivalent to a force applied along a fixed axis plus a torque about the same axis. In particular let us define a wrench (f_c^A, μ_c^A) applied at the point A . This wrench is equivalent to another wrench (f_c^L, μ_c^L) applied at any point L on an axis $l = \{q' : q' = q + \lambda\omega, \quad \forall \lambda \in \mathbb{R}\}$ and whose components f_c^L and μ_c^L are parallel to l , i.e. $f_c^L = \lambda_f \omega$ and $\mu_c^L = \lambda_\mu \omega$. Assuming $\|f_c^A\| \neq 0$, quantities are defined as follows:

$$\omega = \frac{f_c^A}{\|f_c^A\|}, \quad q = \frac{f_c^A \times \mu_c^A}{\|f_c^A\|^2}, \quad \lambda_f = \|f_c^A\|, \quad \lambda_\mu = \frac{f_c^{A\top} \mu_c^A}{\|f_c^A\|}. \quad (19)$$

The proof of the equivalence between (f_c^A, μ_c^A) and (f_c^L, μ_c^L) is reported by **Murray et al.** (1994) and therefore here omitted. In the next subsection, we prove instead that all equivalent wrenches on the Poinot axis have minimum norm torque.

A.1 POINSOT AXIS AS THE GEOMETRIC LOCUS OF MINIMUM TORQUES

Besides being an equivalent wrench, the Poinot wrench (f_c^L, μ_c^L) has an associated minimum norm torque μ_c^L among all equivalent wrenches. In order to prove this optimality principle, let us consider the equivalent torque μ_c^Q at an arbitrary point Q :

$$\mu_c^Q = \mu_c^L + r_{QL} \times f_c^L = \lambda_\mu \omega + r_{QL} \times \lambda_f \omega, \quad (20)$$

being r_{QL} the vector connecting L to Q . Applying the norm to the above equation and observing that the sum is an orthogonal decomposition, we obtain:

$$\|\mu_c^Q\| = \|\lambda_\mu \omega\| + \|r_{QL} \times \lambda_f \omega\| \geq \|\lambda_\mu \omega\| = \|\mu_c^L\|. \quad (21)$$

Therefore $\|\mu_c^Q\| \geq \|\mu_c^L\|$ and the equality holds if and only if r_{QL} is parallel to ω , i.e. when Q lies on the Poinot axis defined as:

$$\left\{ q' : q' = \frac{f_c^A \times \mu_c^A}{\|f_c^A\|^2} + \lambda \frac{f_c^A}{\|f_c^A\|}, \quad \forall \lambda \in \mathbb{R} \right\}. \quad (22)$$

On the Poinot axis, the torque norm is minimal and equals:

$$\|\mu_c^{min}\| = \frac{f_c^{A\top} \mu_c^A}{\|f_c^A\|}. \quad (23)$$

A.2 COP EXISTENCE AND POINSOT AXIS

Given a rigid body subject to a field of pressures, the center of pressure (CoP) is an application point where the equivalent torque (due to the field of pressure) is null. Being the Poinot axis the geometric locus of minimum torques, it is evident that a CoP can be defined if and only if $\|\mu_c^{min}\| = 0$, or equivalently if and only if f_c and μ_c are orthogonal. When this is the case, the CoP is not uniquely defined and the geometric locus of valid CoP corresponds to the Poinot axis (22).

B COUNTEREXAMPLE ON THE GLOBAL COP AS A CONTACT STABILITY CRITERIA

In this section we provide a counterexample to show that the condition on GCoP (global center of pressure) to lie in the contacts convex hull is not sufficient to guarantee the stability of individual contacts. Consider the simple system represented in Figure 11. For certain values of the torques at the joints the global stability criteria are met (i.e. the GCoP lies in the convex hull of contacts) but individual contacts are unstable (i.e. the FRI of each contact lies outside the contact area). Given the unilateral nature of the contacts acting on the system, its solution is non-trivial and requires to make hypotheses on whether or contacts break or persist as clearly explained by **Featherstone** (2008) in chapter eleven of his book. In the specific case of Figure 11 however, it would be sufficient to just provide that particular situation in which torques at the joints force contacts to break while maintaining the GCoP within the contact support polygon. This specific situation is presented in B.3. For sake of clarity, we start with discussing the case in which contacts persist. This case just helps in understanding the proposed counterexample.

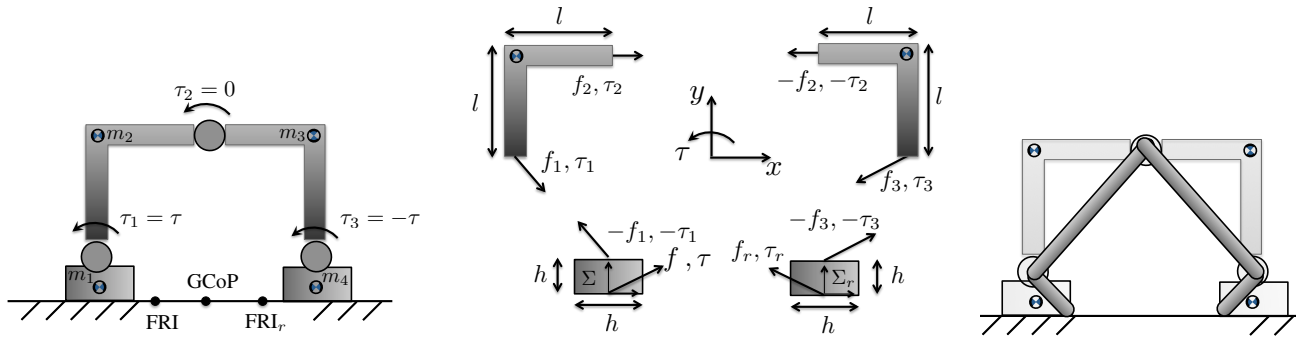


Figure 11. The image shows the example of an articulated rigid body with two links in contact with the ground. **Left figure.** The system is composed of four rigid bodies. The centers of mass associated to the rigid bodies are indicated with a check board circle. Three rotational joints have associated torques τ_1 , τ_2 and τ_3 . **Central figure.** The sketch shows the convention and the geometric dimension used in the computations. The contact interaction forces are indicated with $f_r, f_l \in \mathbb{R}^2$, $\tau_r, \tau_l \in \mathbb{R}$; the internal constraint forces are indicated with $f_1, f_2, f_3 \in \mathbb{R}^2$. **Right figure.** The linkage that describes the system kinematic constraints when assuming that the left foot is pivoting around its right edge and the right foot around its left edge.

B.1 ZMP COMPUTATION

In this section we provide explicit computation for the zero tipping moment point associated to a wrench $w_c = (f_c, \mu_c)$ applied at a generic point P . It is worth stressing once more that the zero tipping moment point is by definition the ZMP, the latter name begin misleading. For sake of simpler notation, let the contact plane coincide with the x - y plane. Given the contact wrench $w_c = (f_c, \mu_c)$ at a generic point $P = [P_x, P_y, 0]$ of the contact plane, the equivalent torque μ'_c at $P' = [P'_x, P'_y, 0]$ is given by:

$$\mu'_c = -r_{PP'} \times f_c + \mu_c = \begin{bmatrix} \mu_c^x - r_{PP'}^y f_c^z \\ \mu_c^y + r_{PP'}^x f_c^z \\ \mu_c^z - r_{PP'}^x f_c^y + r_{PP'}^y f_c^x \end{bmatrix} \quad (24)$$

where $r_{PP'} = P' - P$. At a particular point P' the tipping moments along the x and y axes equal zero. This point corresponds to the zero tipping moment associated to $w_c = (f_c, \mu_c)$ and equals:

$$r_{\text{ZMP}}^x = -\frac{\mu_c^y}{f_c^z}, \quad r_{\text{ZMP}}^y = \frac{\mu_c^x}{f_c^z}. \quad (25)$$

Assuming that the contact wrench $w_c = (f_c, \mu_c)$ is the resultant of a field of pressures p_c on the surface S , we have:

$$f_c^z = \int_S p_c dS, \quad \mu_c^x = \int_S y p_c dS, \quad \mu_c^y = \int_S -x p_c dS. \quad (26)$$

Therefore we have:

$$r_{\text{ZMP}}^x = \int_S x \alpha dS, \quad r_{\text{ZMP}}^y = \int_S y \alpha dS, \quad \alpha = \frac{p_c}{\int_S p_c dS}, \quad (27)$$

where in case of unilateral contacts ($p_c \geq 0$) it results evident that the ZMP is the convex combination of points in S and as such it belongs to the convex hull of S .

B.2 EQUILIBRIUM CONFIGURATIONS

In the following paragraphs we study equilibrium configurations for the system in Fig. 11. Let us first consider the case in which both contacts are active. The idea is to find conditions on the applied torques

to guarantee that the system is in dynamic equilibrium (i.e. null accelerations) and contacts persist. The planar Newton-Euler equilibrium equations for each of the four rigid bodies composing the system give twelve equations. Interaction ($f_r, f_l \in \mathbb{R}^2, \tau_r, \tau_l \in \mathbb{R}$) and internal ($f_1, f_2, f_3 \in \mathbb{R}^2$) forces and torques give twelve unknowns that can be uniquely solved for any choice of the joint torques τ_1, τ_2, τ_3 . To simplify the notation and to obtain a symmetric solution we assume $\tau_1 = \tau, \tau_2 = 0, \tau_3 = -\tau$. Solving the associated equations gives the following solution for the contact forces:

$$f_r = \begin{bmatrix} \tau/l \\ (m_3 + m_4)g \end{bmatrix}, \quad f_l = \begin{bmatrix} -\tau/l \\ (m_1 + m_2)g \end{bmatrix}, \quad (28a)$$

$$\tau_r = -\tau \left(1 + \frac{h}{l}\right), \quad \tau_l = \tau \left(1 + \frac{h}{l}\right), \quad (28b)$$

and internal forces:

$$f_1 = \begin{bmatrix} \tau/l \\ m_2g \end{bmatrix}, \quad f_2 = \begin{bmatrix} \tau/l \\ 0 \end{bmatrix}, \quad f_3 = \begin{bmatrix} -\tau/l \\ m_3g \end{bmatrix}. \quad (29)$$

Assuming for the sake of simpler notation $m_1 + m_2 = m_3 + m_4 = m_{tot}/2$, (28) gives the following expressions for the local right and left foot FRI⁸ (expressed in the associated reference frames Σ_r and Σ_l respectively):

$$\text{FRI}_r = \begin{bmatrix} -\tau \frac{h+l}{m_l g/2} \\ 0 \end{bmatrix}, \quad \text{FRI}_l = \begin{bmatrix} \tau \frac{h+l}{m_l g/2} \\ 0 \end{bmatrix}.$$

The global center of pressure (GCoP) can be computed by representing f_r, f_l, τ_r and τ_l in a common reference frame to obtain the total force and torque f_{tot}, τ_{tot} . Using a reference frame in the middle of the two contacts we have:

$$\tau_{tot}^l = \tau_r + l f_r^y + \tau_l - l f_l^y = 0, \quad f_{tot}^l = \begin{bmatrix} 0 \\ mg \end{bmatrix},$$

and therefore $\text{GCoP} = [0, 0]^T$. As expected by the system symmetry, the global center of pressure is always in the middle of the two contacts regardless of the value given to τ . Instead, the local contacts rotation indicators FRI_r and FRI_l linearly depend on τ and, for a given contact geometry, it is always possible to find a τ which brings them outside the contact areas. As an example, we can assume that the surfaces in contact have width $2h$ (twice the foot height) and, for sake of simpler notation $h = l/2$. With this simplification the FRI is within the support polygon of each contact if and only if:

$$-\frac{mgl}{6} \leq \tau \leq \frac{mgl}{6}.$$

If $\tau \geq mgl/6$, the left foot rotation indicator FRI_l is on the right of the left foot support polygon. Similarly, FRI_r is on the left of the right foot support polygon. In practice, recalling the results presented in Section 2.2, this fact implies that the computed f_r, τ_r, f_l, τ_l for the system equilibrium cannot be generated by a unilateral contact of the given geometry. In a sense, the equilibrium assumption is wrong and we need to redo computations with a different assumption. In the following we assume that the left foot is rotating with respect to its right edge and that the right foot is rotating with respect to its left edge. We then check that the solution found is feasible in terms of contact forces.

⁸ Using the FRI definition and assuming the contact plane to be $y = 0$, a force f and a torque τ have an associated FRI with x -coordinate given by $\text{FRI}^x = \tau/f^y$.

B.3 INWARD FEET ROTATION CONFIGURATIONS

In this section we make the hypothesis that contact constraints force the left foot to rotate around its right edge and the right foot around its left edge. In this specific case the torques at the (point wise) contacts are identically null. With respect to the previous situation we therefore have two unknowns less. Additional unknowns come from the fact that we are no longer assuming accelerations to be identically null since the system is no longer assumed at equilibrium. These additional unknowns can be expressed as a function of only two unknowns (e.g. the left and right foot tipping accelerations). This results by taking into consideration the kinematic constraints in the system, as represented in the left hand side of Figure 11. Therefore, with respect to the equilibrium case we removed two unknowns (the torques at the contacts) and inserted other two (the foot angular acceleration at the tipping point). As a result, the Newton-Euler system of equations is still solvable. Deriving the solution is out of the scope of the present paper and therefore omitted. The interested reader can have a look at the computations which are available in MATLAB⁹. We report here the values of some important variables such as the right and left foot acceleration (denoted \ddot{q}_r and \ddot{q}_l):

$$\ddot{q}_r = k(6\tau - mgl), \quad \ddot{q}_l = -k(6\tau - mgl).$$

for some positive scalar $k > 0$ which depends only on the system geometric and dynamic parameters. By convention positive accelerations correspond to counter clock wise rotations. As soon as the right foot FRI starts moving away from its left edge ($\tau > mgl/6$) the right foot starts counter clock wise rotating. Similarly, when the left foot FRI moves away from its right edge (again, $\tau > mgl/6$) the left foot starts clock wise rotating. The left foot FRI can be computed as well. Given that the foot is not in equilibrium, FRI_l is computed using f_1 and τ_1 reprojected on the planar contact surface. Similar considerations hold for FRI_r . The position of the left and right feet rotation indicators with respect to the pivoting point is given by:

$$FRI_l = \frac{6\tau - mgl}{k_1 + k_2\tau}, \quad FRI_r = -\frac{6\tau - mgl}{k_1 + k_2\tau}$$

where k_1 and k_2 are positive scalars which again depend only on the system geometric and dynamic parameters. As expected FRI_r and FRI_l coincide with the pivoting point (the edge of the support polygon) when $\tau = mgl/6$ and move away from the support polygon when $\tau > mgl/6$. The vertical forces at the contact have the following expressions:

$$f_l^y = k_3\tau + k_4, \quad f_r^y = k_3\tau + k_4$$

for positive constants k_3 and k_4 . Contact forces are therefore positive as expected given the unilaterally of the contact. Finally, given the symmetry of the problem, the GCoP is constantly at the center of symmetry of the system and therefore within the contact support polygon. Therefore, when $\tau > mgl/6$ the system starts rotating feet at their edges (contacts are broken) even if the global center of pressure is within the contact convex hull. This is therefore the counterexample we were looking for.

REFERENCES

- 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

⁹ <https://github.com/iron76/wholeBodyCounterExample>

- 651 Ayusawa, K., Venture, G., and Nakamura, Y. (2014), Identifiability and identification of inertial
 652 parameters using the underactuated base-link dynamics for legged multibody systems, *The*
 653 *International Journal of Robotics Research*, 33, 3, 446–468
- 654 Cannata, G., Maggiali, M., Metta, G., and Sandini, G. (2008), An embedded artificial skin for humanoid
 655 robots, *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent*
 656 *Systems*, 434–438
- 657 de Lasa, M., Mordatch, I., and Hertzmann, A. (2010), Feature-Based Locomotion Controllers, *ACM*
 658 *Transactions on Graphics*, 29, 3
- 659 Del Prete, A. (2013), Control of Contact Forces using Whole-Body Force and Tactile Sensors: Theory
 660 and Implementation on the iCub Humanoid Robot, Ph.D. thesis, Istituto Italiano di Tecnologia
- 661 Del Prete, A., Denei, S., Natale, L., Mastrogiovanni, F., Nori, F., Cannata, G., et al. (2011), Skin spatial
 662 calibration using force/torque measurements., in *IROS (IEEE)*, 3694–3700
- 663 Del Prete, A., Mansard, N., Nori, F., Metta, G., and Natale, L. (2014), Partial force control of constrained
 664 floating-base robots, in *Intelligent Robots and Systems, 2014. (IROS 2003). Proceedings. 2003*
 665 *IEEE/RSJ International Conference on*, –
- 666 Del Prete, A., Natale, L., Nori, F., and Metta, G. (2012), Contact Force Estimations Using Tactile Sensors
 667 and Force / Torque Sensors, in *Human-Robot Interaction (HRI), workshop on Advances in Tactile*
 668 *Sensing and Touch based Human-Robot Interaction, Boston, USA, ACM/IEEE*, 0–2
- 669 Featherstone, R. (2008), *Rigid Body Dynamics Algorithms* (Springer)
- 670 Fumagalli, M., Ivaldi, S., Randazzo, M., Natale, L., Metta, G., Sandini, G., et al. (2012), Force feedback
 671 exploiting tactile and proximal force/torque sensing. Theory and implementation on the humanoid robot
 672 iCub, *Autonomous Robots*, 33, 4, 381–398
- 673 Goswami, A. (1999), Postural stability of biped robots and the foot-rotation indicator (fri) point, *The*
 674 *International Journal of Robotics Research*, 18, 6, 523–533
- 675 Harada, K., Kajita, S., Kaneko, K., and Hirukawa, H. (2003), Zmp analysis for arm/leg coordination,
 676 in *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International*
 677 *Conference on*, volume 1, volume 1, 75–81 vol.1
- 678 Herzog, A., Righetti, L., and Grimminger, F. (2013), Experiments with a hierarchical inverse dynamics
 679 controller on a torque-controlled humanoid, *arXiv preprint arXiv:1305.2042*
- 680 Hirai, K., Hirose, M., Haikawa, Y., and Takenaka, T. (1998), The development of honda humanoid robot,
 681 in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 2,
 682 volume 2, 1321–1326 vol.2
- 683 Hollerbach, J., Khalil, W., and Gautier, M. (2008), Model Identification, in B. Siciliano and O. Khatib,
 684 eds., *Springer Handbook of Robotics* (Springer Berlin Heidelberg), 321–344
- 685 Huang, Q., Yokoi, K., Kajita, S., Kaneko, K., Arai, H., Koyachi, N., et al. (2001), Planning walking
 686 patterns for a biped robot, *IEEE Transactions on Robotics and Automation*, 17, 280–289
- 687 Hyon, S.-H., Hale, J., and Cheng, G. (2007), Full-body compliant human-humanoid interaction:
 688 Balancing in the presence of unknown external forces, *Robotics, IEEE Transactions on*, 23, 5, 884
 689 –898
- 690 Kajita, S. and Espiau, B. (2008), Legged robots, in B. Siciliano and O. Khatib, eds., *Springer Handbook*
 691 *of Robotics*, volume C (Springer Berlin Heidelberg), chapter 16, 361–389
- 692 Khatib, O. (1987), A unified approach for motion and force control of robot manipulators: The operational
 693 space formulation, *IEEE Journal on Robotics and Automation*, 3, 1, 43–53
- 694 Li, Q., Takanishi, A., and Kato, I. (1993), Learning control for a biped walking robot with a trunk,
 695 in *Intelligent Robots and Systems '93, IROS '93. Proceedings of the 1993 IEEE/RSJ International*
 696 *Conference on*, volume 3, volume 3, 1771–1777 vol.3
- 697 Maiolino, P., Maggiali, M., Cannata, G., Metta, G., and Natale, L. (2013), A Flexible and Robust Large
 698 Scale Capacitive Tactile Sensor for Robots, *IEEE Sensors Journal*
- 699 Metta, G., Natale, L., Nori, F., Sandini, G., Vernon, D., Fadiga, L., et al. (2010), The iCub humanoid
 700 robot: an open-systems platform for research in cognitive development., *Neural Networks*, 23, 8-9,
 701 1125–1134
- 702 Murray, R. M., Sastry, S. S., and Zexiang, L. (1994), *A Mathematical Introduction to Robotic*
 703 *Manipulation* (CRC Press, Inc., Boca Raton, FL, USA), 1st edition

- Orin, D. E., Goswami, A., and Lee, S.-H. (2013), Centroidal dynamics of a humanoid robot., *Auton. Robots*, 35, 2-3, 161–176
- Ott, C., Roa, M. a., and Hirzinger, G. (2011), Posture and balance control for biped robots based on contact force optimization, *2011 11th IEEE-RAS International Conference on Humanoid Robots*, 26–33, doi:10.1109/Humanoids.2011.6100882
- Park, J. (2006), Control strategies for robots in contact, Ph.D. thesis, Stanford
- Popovic, M. B. and Herr, H. (2005), Ground reference points in legged locomotion: Definitions, biological trajectories and control implications, *Int. J. Robot. Res*, 24, 2005
- Randazzo, M., Fumagalli, M., Nori, F., Natale, L., Metta, G., and Sandini, G. (2011), A comparison between joint level torque sensing and proximal F/T sensor torque estimation: implementation on the iCub, in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on (IEEE)*, 4161–4167
- Righetti, L., Buchli, J., Mistry, M., and Schaal, S. (2011), Control of legged robots with optimal distribution of contact forces, *2011 11th IEEE-RAS International Conference on Humanoid Robots*, 318–324
- Sardain, P. and Bessonnet, G. (2004), Forces Acting on a Biped Robot. Center of Pressure—Zero Moment Point, *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 34, 5, 630–637, doi:10.1109/TSMCA.2004.832811
- Sentis, L. (2007), Synthesis and control of whole-body behaviors in humanoid systems, Ph.D. thesis, Stanford University
- Spong, M. W. (1994), The control of underactuated mechanical systems, *First international conference on mechatronics*
- Stonier, D. and Kim, J.-H. (2006), Zmp analysis for realisation of humanoid motion on complex topologies, in *Systems, Man and Cybernetics, 2006. SMC '06. IEEE International Conference on*, volume 1, volume 1, 247–252, doi:10.1109/ICSMC.2006.384390
- Vukobratovic, M. and Borovac, B. (2004), Zero-moment point - thirty five years of its life., *I. J. Humanoid Robotics*, 1, 1, 157–173
- Vukobratovic, M. and Juricic, D. (1969), Contribution to the synthesis of biped gait, *Biomedical Engineering, IEEE Transactions on*, BME-16, 1, 1–6
- Wieber, P.-B. (2002), On the stability of walking systems, in *Proceedings of the International Workshop on Humanoid and Human Friendly Robotics*, –

FIGURES

- Figure 1.** Enter the caption for your figure here. Repeat as necessary for each of your figures.

D Improvement of a Balancing Force and Posture Controller with Torque Minimization

Improvement of a Balancing Force and Posture Controller with Torque Minimization

DC 2015.006 - Internship Project Report

Talha Ali Arslan

Supervisors:
prof.dr. Henk Nijmeijer
dr. Alessandro Saccon
dr. Francesco Nori (IIT)
dr. Daniele Pucci (IIT)

Eindhoven, December 2014

Internship report

Talha Ali Arslan

December 12, 2014

Preface

This report is the result of an internship project that took place in the Robotics, Brain and Cognitive Sciences (RBCS) group in the Italian Institute of Tecnology in Genoa, Italy between September 1st, to November 14th in 2014 for 11 weeks, as part of the Mechanical Engineering MSc study program in TU/e (Eindhoven University of Technology) in Eindhoven, The Netherlands.

On the IIT side, I would like to thank the RBCS group members, dr. Naveen Kuppuswamy, Silvio Traversaro, Jorhabib Eljaik, Francesco Romano and Luca Fiorio for assisting and helping me whenever I needed and for creating such a friendly atmosphere to work in. I would also like to thank dr. Daniele Pucci, for his supervision and guidance throughout this project and all the discussions we had which were greatly useful. Finally, I would like to thank dr. Francesco Nori, who is also the team leader, for his supervision and for hosting me in a group of such great spirited people.

On the TU/e side, I would like to thank dr. Alessandro Saccon for his support and encouragement and providing me with this opportunity. Finally, I would like to thank prof.dr. Henk Nijmeijer for supporting and providing me with opportunities since the beginning of my study.

Talha Ali Arslan
December, 2014

Abstract

To improve the balancing of the humanoid robot iCub, the minimization torques and conditions on contact wrenches for balancing stability are implemented to the force and posture control of iCub which is made possible thanks to its wholebody distributed force&torque sensors. The new formulation of the control problem optimizes the feet contact forces in order to minimize the joint torques which helps in reducing any unnecessary internal torques, and during the optimization, conditions on the feet contact forces that ensure stable balancing such as friction cones and zero moment point are imposed as linear inequalities onto the formulized quadratic program (QP). The new controller is also tested by simulations and experiments on the real robot.

Contents

Preface	i
Abstract	iii
1 Introduction	1
1.1 Motivation	1
1.2 Objectives	2
1.3 Outline	2
2 Dynamics of Humanoid Robots	3
2.1 Fully Actuated Systems	3
2.2 Underactuated Systems	3
2.3 Floating-base Systems	4
2.4 Humanoid Robots as Constrained Floating-base Systems	4
3 The Humanoid iCub	7
3.1 Mechanics	8
3.2 Sensors	9
3.3 iCub Software for Simulations	11
4 Balancing with Force and Posture Control	13
4.1 Concepts and Conditions Related to Balancing Stability	13
4.1.1 Friction Cone	13
4.1.2 Support Polygon	13
4.1.3 Zero Moment Point (ZMP)	13
4.2 Formulation of a Force and Posture Controller	14
4.2.1 Centroidal Momentum Derivative	14
4.2.2 Desired Contact Forces	15
4.2.3 Desired Joint Torques	16
4.2.4 The Control Law	17
5 Balancing Controller with Torque Minimization	19
5.1 Problem Definition	19
5.1.1 Balancing Task	19
5.1.2 Postural Task	20
5.1.3 Torque Minimization and constraints	20
5.2 Controller Design	20
5.2.1 Desired Contact Forces	20
5.2.2 Desired Joint Torques	21
5.3 Simulation Environment	23
5.3.1 Forward Dynamics	23
5.3.2 Matlab Visualizer	25

5.4	Results	26
5.4.1	Simulations	26
5.4.2	Experiments on the Robot (iCubGenova01)	30
5.5	Discussion	35
	Conclusion	37

Chapter 1

Introduction

1.1 Motivation

Robots today have more interaction with their environment and users than ever before and it seems that this trend will continue as they become more common in the industry and the daily life. To ensure the safety for all of the elements of the environment that are subject to possible damage in case of an interaction (including the robot itself), the robot must be able to anticipate and/or reciprocate any interactions in a safe manner.

Various control methods are used to ensure that the robots follow and complete their tasks as planned before or during their operation. However, the sensitivity of the robot to its environment has limits and to ensure safety, the question of how the robot will take action in the case of an unexpected change in the environment, must be answered profoundly. Especially in the field of humanoid robotics, safety of interaction is a major issue since they are designed and built to operate and interact with their environment in various ways such as care-giving for the elderly, servicing and assisting in different scenarios and working side by side with humans. Since the diversity of possible end-users of such robots is growing continuously, it is fair to state that the possible end-user is not always meant to be a person who has broad knowledge on how such a robotic platform operates and who can make sure of meeting the safety rules at all times. Hence, it is the engineers' duty to make sure that the robot can operate safely in the cases of unpredicted events in the robot's environment.

If a robot, which is controlled by the positions of its degrees of freedom that can be actuated to complete given tasks, is allowed to interact with its environment without being able to sense what is going on in its surroundings and update its tasks accordingly; any disturbance, unmodeled behavior or unexpected interaction can have a catastrophic result. Nevertheless, the safety of interaction with a position controlled robot can be improved in many ways. Any one or combination of the items below can contribute to safety of interactions:

- Addition of compliance to the system, which can highly reduce damage due to impacts, which may be achieved by using compliant mechanisms and/or control strategies to mimic compliant behavior.
- Incorporation of more sensory data (via additional sensors for various purposes) into the planning and updating of tasks during the operation for better estimation of states, and to anticipate or sense interactions with the environment.
- Addition of more restrictions on the environment to operate in.

Another method is the force control in which the torques on the joints, rather than their position or velocity, are controlled to generate desired forces at contact points. This can be

achieved by using force&torque sensors to estimate joint torques, contact forces and the dynamical model of the robot. At the cost of the estimation of forces and dynamical model using sensors, safety of interactions with the environment is achieved. In any case of interaction or unexpected impacts, compliant behavior is achieved at the joints for they are controlled to follow desired torques rather than position or velocity which otherwise would result in stiff behavior. However, force control strategy differs also in terms of given tasks and it is not as popular as position control, for now.

The humanoid robot iCub of the Italian Institute of Technology (IIT) in Genova, Italy, is a suitable platform for both position and force control with its various whole body distributed sensors such as force&torque and tactile skin sensors. In the Cognitive Humanoids Laboratory of the Robotics, Brain and Cognitive Sciences Department, a force and posture controller is used to keep the robot balanced by generating the desired contact forces at both feet. It is desired to use the possibilities of force control, to have a humanoid robot that can safely interact with its environment as humans do in the daily life. Walking is one of the necessary activities and it seems that the iCub is approaching that goal as it is already able to balance on its feet and move its center of mass with respect to given desired trajectories. This is the stage this internship project is aimed to contribute to, with the implementation of joint torque minimization while respecting the conditions on stable balancing, which will improve the balancing of the iCub to a point where it will hopefully decide to take one foot of the ground soon.

1.2 Objectives

The main objective of this project is to improve the balancing control of the humanoid robot iCub, by means of implementing new features such as the minimization of the joint torques and the consideration of the conditions related to stability during balancing. Objectives alongside the main one are the familiarization with the state of the art software and hardware that is used for designing, building and experimenting with humanoid robots capable of whole body force, torque and position control and to explore more of the theoretical and practical knowledge related to the dynamics and control of humanoid robots.

1.3 Outline

The outline of this report is as follows:

- In Chapter 2, a brief introduction to the dynamics and modeling of humanoid robots as floating-base systems, is given,
- In Chapter 3, the main actor of this and many other projects, the humanoid robot iCub of IIT¹ is introduced within the context of whole body control and this project.
- In Chapter 4, several concepts and conditions on the balancing task are explained and a balancing force and posture controller that is the starting point of this project is explained.
- In Chapter 5, the new controller is introduced along with the new formulation to achieve minimization of torques and to satisfy necessary conditions for balancing stability, followed by the results from simulations and experiments on the robot and discussions.
- In Conclusion, final remarks are made.

¹Istituto Italiano di Tecnologia (Italian Institute of Technology), Genova, Italy

Chapter 2

Dynamics of Humanoid Robots

In this chapter, a brief introduction to the dynamics and equations of motion of humanoid robots is given by short descriptions of the classes of systems humanoids are related or belong to; to familiarize ourselves with the descriptions of dynamics of humanoid robots as constrained floating-base systems.

2.1 Fully Actuated Systems

Definition: Fully actuated systems have exactly the same number of control inputs and total number of degrees of freedom.

The equations of motion of a fully-actuated system can be defined as;

$$M(q)\ddot{q} + h(q, \dot{q}) = \tau, \quad (2.1)$$

where $q \in \mathbb{R}^n$ is the vector of generalized coordinates, $M(q) \in \mathbb{R}^{n \times n}$ is the inertia matrix, $\tau \in \mathbb{R}^n$ is the vector of generalized input torques and $h(q, \dot{q}) \in \mathbb{R}^n$ is the vector of bias forces containing Coriolis, centrifugal and gravity forces.

With fixed-base and fully actuated systems at initial configuration q_0 and $\dot{q}(0) = 0$, a desired trajectory q^* which can be defined as;

$$q^* = q_0 + \int_0^T \ddot{q}^* dt \quad (2.2)$$

and \ddot{q}^* can directly be used to calculate input torque trajectory that is necessary to follow q^* by;

$$\tau^* = M(q)\ddot{q}^* + h(q, \dot{q}). \quad (2.3)$$

which is not the case for systems that are not fully actuated.

2.2 Underactuated Systems

Definition: Underactuated systems are characterized by having fewer number of control inputs than the total number of degrees of freedom.

For the underactuated systems, the equations of motion can be written as;

$$M(q)\ddot{q} + h(q, \dot{q}) = S^T \tau. \quad (2.4)$$

with the only difference, from the equations of motion of fully actuated systems (2.1), being $\tau \in \mathbb{R}^a$ where a is the number of actuated degrees of freedom with $a < n$ and $S \in \mathbb{R}^{a \times n}$ being the selector matrix which selects the control inputs for the a active joints rather than $(n - a)$ passive joints.

Due to the difference between the total number of degrees of freedom (n) and the number of actuated degrees of freedom ($a < n$), the underactuated systems are not fully feedback linearizable and the problem of finding an acceleration trajectory \ddot{q}^* that brings the system from an initial configuration q_0 to a desired configuration q^* is not trivial at all, because the acceleration of the system that can be achieved at an instant of time is not available in all directions and it is also dependent on the states due to the fact that,

$$\ddot{q} = M^{-1}(q)(-h(q, \dot{q}) + S^T \tau^*) \quad (2.5)$$

defines $\ddot{q} \in \mathbb{R}^n$ at an instant of time with input torques $\tau^* \in \mathbb{R}^a$ where $a < n$ meaning that the n -dimensional vector \ddot{q} lies on a manifold of dimension a , because of the underactuation. Although acceleration at an instance is only available in a subset of all directions, it is still possible that the system can be controllable, meaning that there can be found a trajectory for input torques τ^* that can achieve an acceleration trajectory \ddot{q}^* that will bring the system from any initial configuration q_0 to a any final desired configuration q^* in a finite time.

2.3 Floating-base Systems

Floating-base systems, being a class of underactuated systems, have their equations of motion defined similar to (2.4) by adding a passive joint with 6 degrees of freedom from the base of the robot to the inertial frame resulting in a model in the same way as a fixed-base underactuated system;

$$M(q)\ddot{q} + h(q, \dot{q}) = S^T \tau, \quad (2.6)$$

with,

$$q = \begin{bmatrix} x_b \\ q_j \end{bmatrix} \in \mathbb{R}^{n+6}, \quad \begin{matrix} x_b \in \mathbb{R}^6 \\ q_j \in \mathbb{R}^n \end{matrix}, \quad M(q) \in \mathbb{R}^{(n+6) \times (n+6)}, \quad h(q) \in \mathbb{R}^{n+6}, \quad S^T = \begin{bmatrix} 0_{6 \times n} \\ I_{n \times n} \end{bmatrix} \quad (2.7)$$

where the vector of generalized coordinates $q \in \mathbb{R}^{n+6}$ is formed by combination of the 6 degrees of freedom (position and orientation) that come from the generalized coordinates of the base $x_b \in \mathbb{R}^6$ and generalized coordinates of the actuated joints $q_j \in \mathbb{R}^n$, which is typical for a robot with all of its joints actuated and free floating in 3-dimensional space.

2.4 Humanoid Robots as Constrained Floating-base Systems

As they are not fixed to any reference frame and can be found floating freely, humanoids are a class of floating base systems. However, humanoid robots usually have contacts -almost always with the ground-. Hence, they can be described as constrained floating-base systems.

Making contacts with environment results in constraints on the motion at the contact points which restricts movements in constrained directions where the constraint forces act, Assuming that the contacts are rigid and holonomic, the constraints on movements can be represented by imposing zero velocity at the contact points as;

$$J_c(q)\dot{q} = 0, \quad (2.8)$$

and also at the acceleration level as;

$$J_c(q)\ddot{q} + \dot{J}_c(q)\dot{q} = 0, \quad (2.9)$$

where $J_c(q) \in \mathbb{R}^{k \times (n+6)}$ is the jacobian of the contact points and k is the number of constrained directions of motion.

With the addition of constraints, forces at the rigid contact points act on the body and equation of motion of constrained floating-base systems becomes;

$$\begin{aligned} M(q)\ddot{q} + h(q, \dot{q}) - J_c^T(q)f_c &= S^T\tau, \\ \text{s.t. } J_c(q)\ddot{q} + \dot{J}_c(q)\dot{q} &= 0. \end{aligned} \quad (2.10)$$

where $q \in \mathbb{R}^{(n+6)}$ is the vector of generalized coordinates containing the position and orientation of the base $[x_b \ w_b]^T \in \mathbb{R}^6$ and the joint positions $q_j \in \mathbb{R}^n$, $M(q) \in \mathbb{R}^{n \times n}$ is the inertia matrix, $\tau \in \mathbb{R}^n$ is the vector of generalized input torques, $f_c \in \mathbb{R}^k$ is the constraint forces that maintain the constraints, $J_c(q) \in \mathbb{R}^{k \times n}$ is the constraint jacobian and $h(q, \dot{q}) \in \mathbb{R}^n$ is the vector of bias forces containing Coriolis, centrifugal and gravity forces.

Chapter 3

The Humanoid iCub

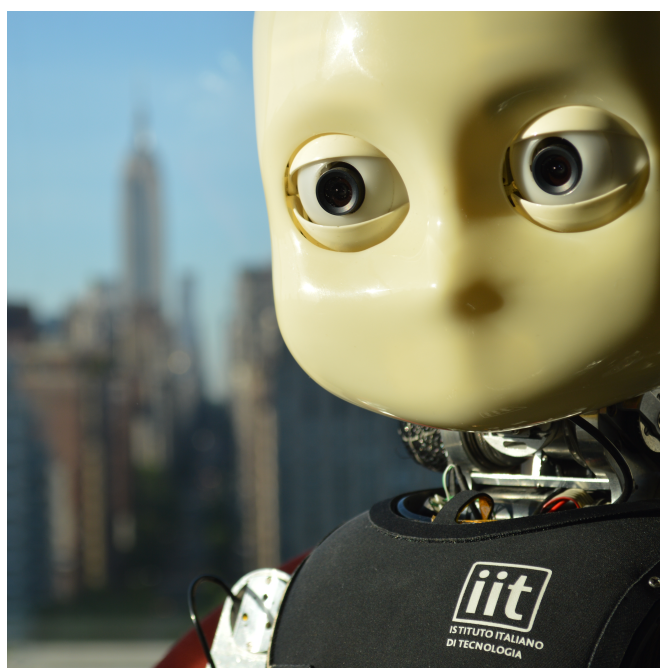


Figure 3.1: iCub¹

The humanoid robot iCub is an open-source platform for research in humanoid robotics, brain and cognitive sciences. Being a platform that is used in such broad fields of research, it has various sensors and mechanical features that allow testing for various tasks. At the size of a human child, iCub has four limbs consisting of two legs and two arms including hands with five fingers and various sensors such as force&torque, tactile skin, gyroscopes, accelerometers and cameras distributed over its body to be able to sense its environment. Basically, it is a humanoid that is built with the intention to allow it to interact with people and the environment. In this chapter, we introduce the humanoid iCub in the context of whole body force and posture control for balancing; by describing the mechanical characteristics, whole body distributed sensors and software to be used for the purposes of this project. It should be noted that these are only a subset of all the features of the humanoid iCub and more details on the platform can be found in [1].

¹Photo Credit: Istituto Italiano di Tecnologia - <http://www.iit.it/en/social/photo-gallery.html>

3.1 Mechanics

iCub has 53 degrees of freedom that are actuated, distributed as; 6 for the head (3 for the neck and 3 for the eyes), 3 for the torso, 16 for each arm (9 of which are for each hand) and 6 for each leg. Except for the hands and the head where brushed DC motors are used, all of the remaining joints are actuated by brushless DC motors with harmonic drive gears.

The material used for the body parts is mostly Ergal (an aluminum alloy). Steel and plastic is also used and the total mass of the robot is around 25 kg. It does not have a certain mass since it is a research platform that is subject to be customized by researchers by removal, addition and testing of new sensors and parts.

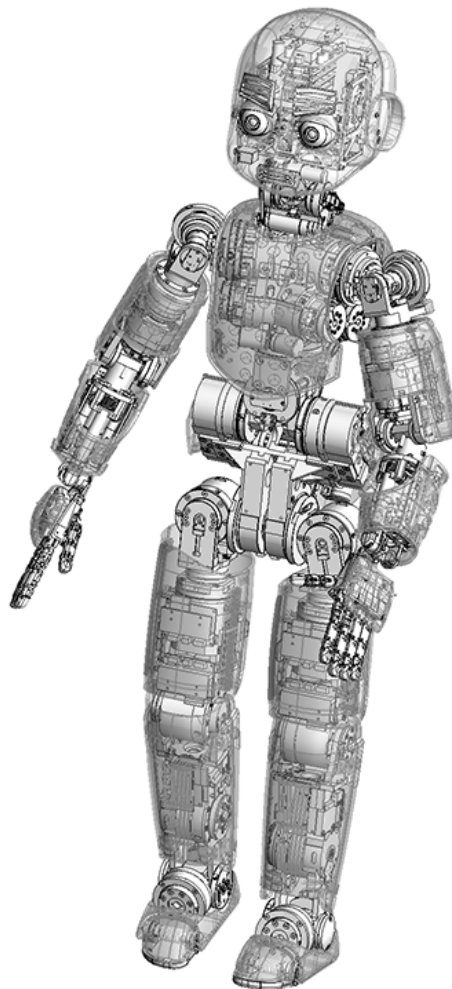


Figure 3.2: iCub Mechanics²

For the purpose of the project this report is on, 25 degrees of freedom from the arms (3 from each shoulder, 2 from each elbow), the legs (3 from each hip, 1 from each knee and 2 from each ankle) and the torso are exploited and others such as the wrists, the hands and the head are kept at an initial configuration with position control.

iCub is not fully autonomous, yet, because power is supplied to the robot via a cord which also communicates with a cluster machine where the higher-level control takes place. For the

²Image Credit: Istituto Italiano di Tecnologia - <http://www.iit.it/en/products/catalog.html>

low-level motor control, there are microcontroller boards on the robot that can be used. During the course of this project, for joint torque control, high and low level control were both carried out at 100 Hz by the off-board cluster machine. To implement the estimation of joint torques to on-board means of computation is an ongoing process which can allow low-level joint torque control at a higher frequency with less delays.

3.2 Sensors

iCub is equipped with sensors such as; two digital cameras in the head, microphones, force&torque sensors, tactile skin, gyroscopes and accelerometers. In addition, there are absolute position encoders in every actuated joint.

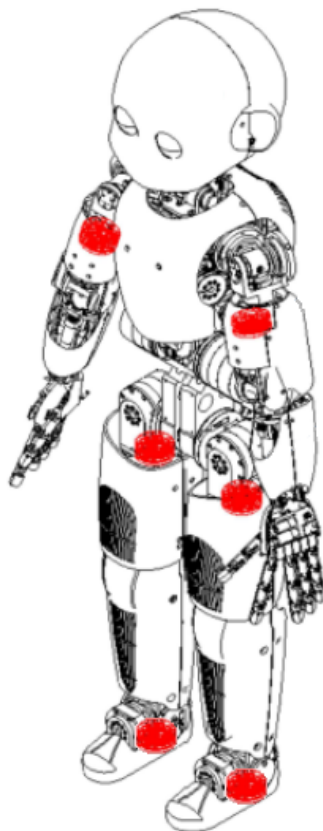


Figure 3.3: Force/torque sensors on iCub

Whole-body distributed force&torque sensors play an important role in the estimation of joint torques and the control the torques at each joint. There are six force&torque(F/T) sensors; one between the shoulder and the elbow on each arm, one between the hip and the knee and one between the ankle and the foot on each leg, as depicted in Figure 3.3. By using these sensors, internal forces can be estimated at the joints which in turn can be used to estimate internal dynamics and external forces.

The framework proposed by [2] for the estimation of internal/external forces on the floating-base system consists of separating the whole body kinematic tree of the floating-base into subtrees by the location of the F/T sensors. In return, independent kinematic subtrees are obtained which are also governed by Newton-Euler dynamic equations. Each sensor gives a direct mea-

surement of the external wrench acting on that body consisting of the subtree. Then by using these external wrenches in a Newton-Euler force propagation recursion, the joint torques can be estimated. Hence, a torque controller with joint-torque friction compensation is able to provide desired torques by using the estimation from the measurement of internal forces and torques. Tactile skin sensors can also be incorporated into this process to estimate external wrenches with their accurate locations and this is an ongoing work too.

Specifications of the humanoid iCub can also be summarized as in the table below[3].

Height	104cm
Weight	23-25 kg
Sensors	Stereo Cameras, microphones, encoders, force/torque sensors, tactile sensors (capacitive) fingertips and skin on upper body and arms, gyroscopes, accelerometers
Actuators	Large joints as e.g. the shoulder (brushless motors, 150W), small joints as e.g. hands (DC motors). 54 motors in total
Power	220/110V AC, tethered via 48-12V power supply
Computing (On-board)	20 microcontroller boards for movement, 16 boards for sensors and a Pentium duo for data acquisition and synchronization.
Computing (Off-board)	a cluster with 30-40 cores and GPU processing and more if needed
Software (On-board)	Debian Linux.
Software (Off-board)	any of Windows, Linux, MacOS in any combination depending on the configuration/user needs.
	Software middleware controlling the cluster and the robot called YARP
Degrees of Freedom	53 motors controlling 76 joints
Structure and Materials	Mostly Ergal (aluminum alloy), steel and plastic
Cost	Latest configuration about 250.000 Euros
Year	Started 2004, first release 2008
Location	Genoa, Rome, Lyon, Paris, Barcelona, Munich, Bielefeld, London, Plymouth, Aberystwyth, Lisbon, Urbana-Champaign and Ankara

3.3 iCub Software for Simulations

Being an open-source research platform that is used in more than 20 different locations around the world, iCub project is maintained by a large research community. This way, it is easy to access and even contribute to any materials on research related to iCub.

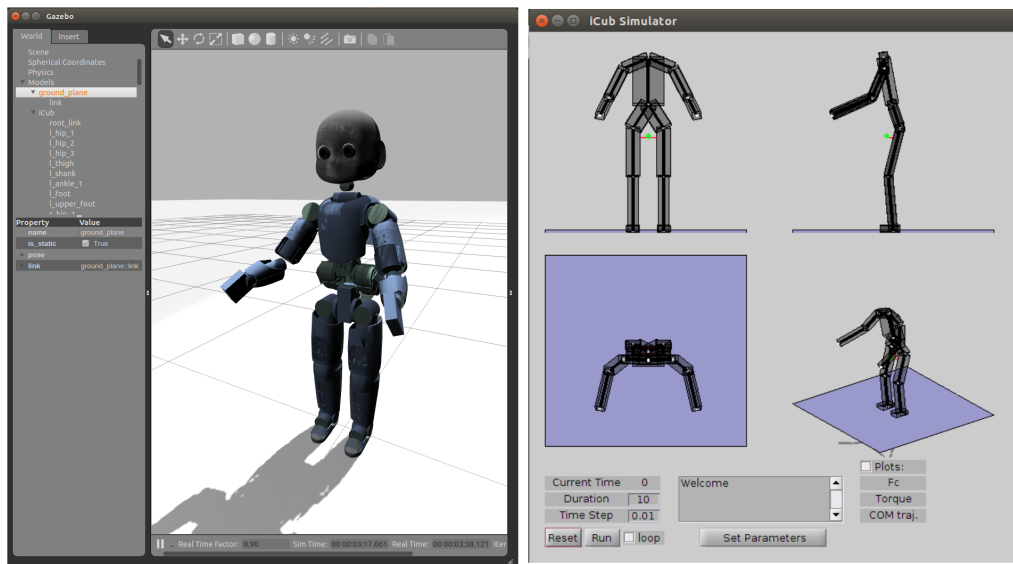


Figure 3.4: Simulating iCub in Gazebo (left) & Matlab (right)

Besides the real experimental platform, the humanoid iCub can also be simulated and switching between simulations and the real platform does not require a lot of effort. Simulations could previously be done by using Gazebo. However because of the complexity of the robot model and a lack of freedom on customization of the Gazebo simulator, force&torque control simulations could not be done properly, even if they were very simple tasks and they were also previously validated on the real robot. For this reason, a software package using Matlab is developed, to which this internship project also contributed. It is still under development but it was extensively used for prototyping purposes and it will be explained in Section 5.3: Simulation Environment.

To be able work on iCub either through simulations or on one of the real platforms, certain packages of software needs to be installed and this can be done by following the instructions in Wiki page for the iCub [4] and packages' related repositories on GitHub³. Any one of the Linux, Mac OS X or Windows operating systems can be used. During this project, Linux-based Ubuntu 14.04 LTS was used.

One of the core component of the related software is the software library iDynTree which is used to obtain the estimated joint torques and external wrenches in an efficient, generic and easy way. The library uses the kinematic information and the data from force&torque sensors to do the estimations with methods from [5, 6]. Another core software is YARP (Yet Another Robot Platform)[Metta et.al.2006] which is a middleware used for decoupling the all the software modules and devices cleanly. By using different connection types, it allows creating a collection of programs that communicate in a peer-to-peer way. It is also free and open-source. All the other related software necessary to be able to work with iCub is available with instructions to

³<https://github.com/>

install at the iCub Wiki[4].

Currently, the balancing controller for the iCub is tested on the real robot using Matlab Simulink. The WBI Toolbox in Simulink that comes along with the iCub software consists of Simulink blocks that provide the robot dynamics, kinematics, measurements etc. to be able to compute desired joint torques within a controller. Being able to design the controller in Simulink makes it easier for the researchers to understand, debug and modify them. However, it should be noted that using Simulink creates extra computational effort and this process can be done a lot faster using C/C++ implementations of the controllers. Considering that the current control loop in Matlab is processed at 10 ms (i.e. at a frequency of 100 Hz), which suggests that the speed of the control loop can not be counted on for highly dynamical tasks. However, for the balancing task, this frequency proves to be sufficient for now. Also there is an ongoing work on the implementaion of the controllers into C/C++.

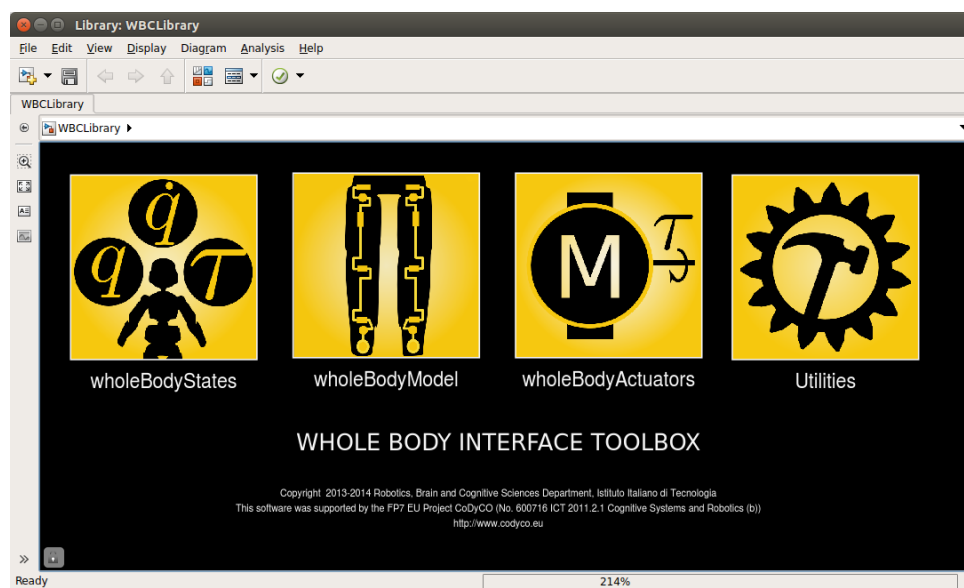


Figure 3.5: WBI Toolbox in Simulink

Chapter 4

Balancing with Force and Posture Control

In this chapter, the balancing task for a biped humanoid will be explained by related concepts and conditions. Then, a method which is based on the TSID (Task Space Inverse Dynamics) framework by [7] to balance a biped humanoid robot is explained. This method also forms the basis for the new controller with torque minimization which is the result of this project and will be explained in the next chapter.

4.1 Concepts and Conditions Related to Balancing Stability

To ensure stability during the balancing task, there are some conditions related to the feet contact forces (contact wrenches) that can be imposed as necessary while computing desired feet contact forces.

4.1.1 Friction Cone

Due to the friction coefficient between the surfaces of a planar contact, there is limit tangential force that can be supported by the contact until slipping occurs. If this happens with a humanoid robot, it would mean that the contact position changes and while the contact is not constrained in the directions it was previously, the state of the controller must change in order to operate successfully. For the balancing task, we want to keep the feet at their initial configurations at all times and since we will control the contact wrenches at the feet (from now on 'feet contact forces'), we must choose the desired feet contact forces in accordance with conditions of none slipping.

Friction cones can be used for this purpose, where the net force acting on the contact is desired to be kept inside a cone of friction, the cone angle of which is defined by the friction coefficient.

4.1.2 Support Polygon

The support polygon(SP) of a robot describes the minimal convex hull that can be formed around and thus including contact points of the robot.

4.1.3 Zero Moment Point (ZMP)

Zero moment point (ZMP) can be described as the tipping point where a planar contact's resultant wrench (force and moment) can be described with a force and moment couple where the net tangential moment is zero. If this point is outside the support polygon, meaning that the resultant contact wrench can be replaced by the same force and zero moment at a location

where it is outside the edges of a contact or the area supported in the case of multiple contacts, then there would be tipping over the edge closest to the ZMP. If zero moment point is on the edge of the support polygon, then

4.2 Formulation of a Force and Posture Controller

In this section, the force and posture controller based on the TSID framework proposed by [7] for balancing of the iCub is explained. It will be referred to as the previous controller in the next chapters.

4.2.1 Centroidal Momentum Derivative

When a floating-base system has external wrenches due to gravitation or contacts, according to D'Alembert's principle, the rate of change in angular and linear momenta of the system's center of mass are equal to the sum of external moments and forces, respectively. The rate in linear and angular momenta can be written as;

$$\begin{aligned} \dot{l}_{com} &= \vec{F}_c + m\vec{g} \\ \dot{k}_{com} &= (\vec{p}_c - \vec{p}_{com}) \times \vec{F}_c + \vec{M}_c \end{aligned} \quad (4.1)$$

where m is the total mass of the system, g is the gravitational acceleration, \dot{l}_{com} and \dot{k}_{com} denote the linear and angular momenta of the center mass respectively, \vec{F}_c and \vec{M}_c denote the force and moment components of the external wrenches at the contact points respectively, \vec{p}_c is the position of the contact and \vec{p}_{com} is the position of the center of mass.

From $\tau = \vec{r} \times \vec{f} = S(\vec{r})\vec{f}$, the equation 4.1 can be written in a more compact form as;

$$\begin{bmatrix} \dot{l}_{com} \\ \dot{k}_{com} \end{bmatrix} = \begin{bmatrix} I & 0 \\ S(\vec{r}_c) & I \end{bmatrix} \begin{bmatrix} \vec{F}_c \\ \vec{M}_c \end{bmatrix} + \begin{bmatrix} m\vec{g} \\ 0 \end{bmatrix} \quad (4.2)$$

where $S(\cdot)$ denotes the skew symmetric matrix operator and $\vec{r}_c = \vec{p}_c - \vec{p}_{com}$ denote the relative position of the contact point with respect to the center of mass.

For the two contacts case, similarly;

$$\begin{bmatrix} \dot{l}_{com} \\ \dot{k}_{com} \end{bmatrix} = \begin{bmatrix} I & 0 & I & 0 \\ S(\vec{r}_{c1}) & I & S(\vec{r}_{c2}) & I \end{bmatrix} \begin{bmatrix} \vec{F}_{c1} \\ \vec{M}_{c1} \\ \vec{F}_{c2} \\ \vec{M}_{c2} \end{bmatrix} + \begin{bmatrix} m\vec{g} \\ 0 \end{bmatrix} \quad (4.3)$$

Then the rate of change in the spatial centroidal momentum of the system can be written as;

$$\dot{H}_{com} = Af_c + f_{grav} \quad (4.4)$$

where we defined,

- The vector containing the external wrenches on multiple contact points: $\begin{bmatrix} \vec{F}_{c1} \\ \vec{M}_{c1} \\ \vec{F}_{c2} \\ \vec{M}_{c2} \end{bmatrix} \in \mathbb{R}^{12}$ as

the spatial feet contact forces $f_c = \begin{bmatrix} f_{left} \\ f_{right} \end{bmatrix}$ (we omit the vector notation for simplicity),

- The summation matrix for the forces and moments: $\begin{bmatrix} I & 0 & I & 0 \\ S(r_{c_1}) & I & S(r_{c_2}) & I \end{bmatrix} \in \mathbb{R}^{6 \times 12}$ as A matrix,
- The spatial force acting the system due to gravity: $\begin{bmatrix} m\vec{g} \\ 0 \end{bmatrix} \in \mathbb{R}^6$ as f_{grav} .

We know that for the center of mass, the linear momentum derivative can also be written as,

$$\dot{l}_{com} = m\ddot{p}_{com} \quad (4.5)$$

Hence if we can control the rate of change in the linear momentum of the center of mass, we can control the acceleration and therefore velocity and position of the center of mass. Additionally, since this is a humanoid, not a free-floating contactless robot, we are interested in minimum angular momentum. Hence we can define the desired rate of change in the spatial centroidal momentum to achieve desired accelerations of the center of mass and minimize the angular part as,

$$\dot{H}^* = \begin{bmatrix} \dot{l}_{com}^* \\ \dot{k}_{com}^* \end{bmatrix} = \begin{bmatrix} m\ddot{p}_{com}^* \\ -K_h k_{com} \end{bmatrix} \quad (4.6)$$

To follow a reference trajectory $p_{com}^d(t)$ with continuous first and second derivatives $\dot{p}_{com}^d(t)$, $\ddot{p}_{com}^d(t)$, the desired acceleration can be chosen as,

$$\ddot{p}_{com}^* = \ddot{p}_{com}^d - K_p(p_{com} - p_{com}^d) - K_d(\dot{p}_{com} - \dot{p}_{com}^d) \quad (4.7)$$

where p_{com} and \dot{p}_{com} denote the position and the linear velocity of the center of mass in Cartesian space respectively, K_p and K_d are matrices with positive terms on the diagonal which denote the proportional and derivative gains on the position and velocity errors in three directions. The position of the center of mass is available via forward kinematics and the linear velocity of the center of mass can be acquired by, $\dot{p}_{com}^d = [I_{3 \times 3} \quad 0_{3 \times 3}] J_{com}(q)^T v$, where J_{com} is the jacobian for the center of mass and v is the derivative of the generalized coordinates.

Finally, we can define the \dot{H}^* which will help in achieving desired center of mass trajectory and minimizing angular momentum, from Equations 4.6 and 4.7 as,

$$\dot{H}^* = \begin{bmatrix} m(\ddot{p}_{com}^d - K_p(p_{com} - p_{com}^d) - K_d(\dot{p}_{com} - \dot{p}_{com}^d)) \\ -K_h k_{com} \end{bmatrix} \quad (4.8)$$

4.2.2 Desired Contact Forces

Recall Equation 4.4 from the previous subsection defining the rate of change in spatial momentum of the center of mass of the robot,

$$\dot{H} = A f_c + f_{grav} \quad (4.9)$$

where f_c denotes the vector containing the contact forces f_{left} and f_{right} on left and right feet respectively; f_{grav} denotes the gravitational forces acting on the robot due to its mass; A denotes the matrix that relates the spatial feet contact forces to the rate of change in spatial centroidal momentum which is denoted as \dot{H} .

A desired trajectory of the center of mass can be achieved by setting a desired time derivative of the centroidal momentum and computing the desired contact forces accordingly. Hence, 4.4 needs to be solved with respect to f_c .

$$Af_c^* = \dot{H}^* - f_{grav} \quad (4.10)$$

For the case of both feet in contact with the ground, A matrix is a rectangular matrix ($m = 6$ rows by $n = 12$ columns with $n < m$) which does not have a full column rank and the system is indeterminate, meaning that there are infinite number of solutions to the problem above (4.10). We can formulate it as an optimization problem,

$$f_c^* = \arg \min_f \|\dot{H}^* - \dot{H}\|^2 \quad (4.11)$$

which is subject to Equation 4.9 and the solution of this optimization is given by,

$$f_c^* = A^+(\dot{H}^* - f_{grav}) + (I - A^+A)f_0 \quad (4.12)$$

where f_0 is an arbitrary vector that is projected into the nullspace of f_c^* and A^+ is the Moore-Penrose pseudo inverse of A . This solution gives a set of desired feet contact forces f_c^* which would result in the desired rate of change in the centroidal momentum \dot{H}^* . If the arbitrary vector f_0 is taken as a zero vector, we get,

$$f_c^* = A^+(\dot{H}^* - f_{grav}) \quad (4.13)$$

which gives the minimum norm solution for f_c .

4.2.3 Desired Joint Torques

Recall the equations of motion of the humanoid robot from Section 2.4 as,

$$\begin{aligned} M(q)\dot{v} + h(q, v) &= S^T\tau + J_c^T(q)f_c \\ J_c(q)\dot{v} + \dot{J}_cv &= 0 \end{aligned} \quad (4.14)$$

where $M(q)$ is the inertial mass matrix, $h(q, v)$ is the vector containing the bias forces, S is the selector matrix $[0, I_{25 \times 25}]$, J_c is the jacobian matrix for the contact points and f_c is the vector containing contact forces.

While omitting the dependencies on q and v for simplicity, \dot{v} from Equation 4.14 can be obtained as,

$$\dot{v} = M^{-1}(S^T\tau + J_c^T f_c - h) \quad (4.15)$$

then substituting above expression into the constraint part of Equation 4.14 and leaving τ on one side,

$$J_c M^{-1} S^T \tau = J_c M^{-1} (h - J_c^T f_c) - \dot{J}_c v \quad (4.16)$$

Due to the fact that $J_c M^{-1} S^T$ does not have full column rank, the above expression does not have a unique but a set of solutions for τ , similar to Equation 4.10. Hence, it can be solved for τ as,

$$\tau^* = \Lambda^+ \left(J_c M^{-1} (h - J_c^T f_c^*) - \dot{J}_c v \right) + N_\tau \tau_0 \quad (4.17)$$

$$\text{with } \Lambda = J_c M^{-1} S^T, \quad N_\tau = (I - \Lambda^+ \Lambda)$$

where Λ^+ is the Moore-Penrose pseudo inverse of Λ , N_τ is the null-space projector matrix, τ_0 is an arbitrary vector.

In the TSID framework by [7], the task of controlling the forces at the contacts is done with the highest priority and any secondary tasks such as motion and posture can be done by exploiting the null-space of the primary control task. The above expression for the desired joint torques is derived such that the forces at the contacts match closely to the desired values and the null-space of this solution can be exploited by setting the arbitrary vector τ_0 for any secondary task, which in this case will be the postural task.

With the task of controlling the contact forces at the feet, we ensure the controlled movement and position of the center of mass of the robot. However, this task can be achieved by infinitely many joint configurations. Hence, to keep joint position at acceptable regions, we have to impose a postural task to make each joint withstand the external forces due to the contacts and gravity and stay close to a desired position. At the joint level, this can be done by considering the generalized forces and forces due to the contacts acting on the joints along with a feedback on position to impose impedance for the joints as they move away from their desired values which form the desired posture for the robot. Hence the arbitrary vector τ_0 , that will be projected onto the null-space of the force control task, can be defined as;

$$\tau_0 = S (g(q) - J_c(q)^T f_c^*) - K_{imp}(q_j - q_0) \quad (4.18)$$

where S is transpose of the selector matrix as in Equation 4.14 to choose only joint related terms, $g(q)$ is the vector of generalized forces, $J_c(q)$ is the contact Jacobian, f_c is the feet contact forces, K_{imp} is a diagonal matrix containing the gains for the impedance behavior for each joint and q_0 is the desired joint configuration which is fixed as the initial configuration for now.

4.2.4 The Control Law

From the derivation in the last three subsections, the force and posture control law can be concisely written as (From Equations 4.8, 4.13, 4.17, 4.18),

$$\begin{aligned} \tau^* &= \Lambda^+ \left(J_c M^{-1} (h - J_c^T f_c^*) - \dot{J}_c v \right) + N_\tau \tau_0 \\ \text{with,} \\ \Lambda &= J_c M^{-1} S^T \\ N_\tau &= (I - \Lambda^+ \Lambda) \\ f_c^* &= A^+ (\dot{H}^* - f_{grav}) \end{aligned} \quad (4.19)$$

$$\dot{H}^* = \begin{bmatrix} m (\ddot{p}_{com}^d - K_p(p_{com} - p_{com}^d) - K_d(\dot{p}_{com} - \dot{p}_{com}^d)) \\ -K_h k_{com} \end{bmatrix}$$

$$\tau_0 = S (g(q) - J_c(q)^T f_c^*) - K_{imp}(q_j - q_0)$$

This controller computes the joint torques necessary in achieving the desired contact forces in order to control the position of the center of mass for a given trajectory while as a secondary task, it keeps the posture of the robot close to a desired configuration. However, it is not yet complete in the sense that it does not include conditions of the contact forces to ensure stability of the balancing task. This will be addressed in the formulation for a controller which is the outcome of this project. From now on, the controller described in 4.19 will be referred to as the previous controller. Simulations and experiments done with this controller will also be presented in the next chapter to help the analysis and comparison with the new controller which comes next.

Chapter 5

Balancing Controller with Torque Minimization

In this chapter, we present the main outcome of this project which is a balancing force and posture controller with torque minimization achieved by a new formulation as an optimization problem where additional constraints can be imposed. Firstly, we start with the problem definition in Section 5.1 where we describe the tasks we want to address with this controller. Following that is the derivation for the new controller in Section 5.2, which has the controller described in the previous chapter as its starting point (from now on 'the previous controller'). After the formulation of the new controller, we describe the implementation of the controller to simulations and the real robot in Section 5.3. Finally, we present and discuss the results from simulations and experiments in Section 5.4 and Section 5.5, respectively.

5.1 Problem Definition

The previous balancing controller exploits the feet contact forces to change the centroidal momentum of the robot which results in a desired acceleration of the center of mass as it is presented in the previous chapter in Section 4.2. However, due to the fact that the problem of choosing the feet contact forces that result in the desired centroidal momentum derivative does not have a unique but a set of solutions. Exploiting it can allow us to take into account any desired constraints at the force and torque level and it can help towards the minimization of the desired joint torques.

The objective of this project is to improve upon the previous state of the balancing controller by implementing optimization to minimize joint torques and along the process adding constraints on contact forces, joint torques and other additional means if necessary, to ensure that the given motion tasks on the center of mass (CoM) of the robot can be achieved. In the following subsections, these tasks are defined and how they are going to be addressed in the derivation of the new formulation for the controller will be discussed briefly. Afterwards, in Section 5.2, the new controller will be explained in detail.

5.1.1 Balancing Task

The balancing task for the robot will be for the case of both feet touching the ground while the contacts are assumed to be rigid. In order to keep its balance, the robot feet where the contacts are present should not slip, lift or rotate in any direction. The conditions that ensure such stability for the balancing task can be included as constraints on the feet contacts forces staying within a friction cone which ensure no slipping, and constraints on the zero moment

point (ZMP)[Vukobratovic and Jorjic 1969] resulting from the feet contact forces being inside the support polygon of the robot which ensure no tipping over. In order to ensure that the balancing is achieved these constraints must be included in the computation for the desired contact forces.

5.1.2 Postural Task

Keeping the center of mass of the robot (CoM) at a desired position can be achieved in an infinite number of configurations with a floating-base system which is a subclass of underactuated systems. Hence to ensure internal stability of the joints and keeping a meaningful and useful whole body configuration at all times, a postural task needs to be implemented. This is implemented in the new controller in the same way as it is done in the previous version of the controller (Equation 4.19), by exploiting the nullspace of the task of generating the desired feet contact forces as described in Section 4.2.3.

5.1.3 Torque Minimization and constraints

Since there is no unique solution for the feet contact forces that result in achieving the balancing and postural tasks, this free choice can be exploited to formulate the desired joint torques as the objective function of an optimization problem. The solution to such a problem will result in desired feet contact forces that minimize the desired joint torques. Additionally, the constraints on the feet contact forces and the joint torques will be added to the optimization problem to ensure a desired execution for the balancing task.

5.2 Controller Design

5.2.1 Desired Contact Forces

Recall the derivation of the desired contact forces in 4.2.2 for the previous controller (4.19), where the set of solutions to the problem of computing the contact forces to achieve the desired centroidal momentum derivative,

$$\dot{H} = A f_c + f_{grav} \quad (5.1)$$

is given as,

$$f_c^* = A^+(\dot{H}^* - f_{grav}) + (I - A^+A)f_0. \quad (5.2)$$

where f_c^* denote the vector containing the external wrenches at the contact points (*feet contact forces*), \dot{H}^* denotes the spatial centroidal momentum derivative containing the linear and angular momenta derivative of the center of mass, A is the matrix relating the forces and moments at the contact points to the momenta derivative, A^+ is the Moore-Penrose pseudo inverse of A , f_{grav} is the gravitation force acting on the center of mass, $(I - A^+A)$ is the nullspace projector matrix and f_0 is an arbitrary vector.

The above equation defines the set of possible choices for the contact forces that solves to problem of achieving a desired rate of change in the spatial centroidal momentum to control the center of mass for a given trajectory. Although there seems to be many solutions, there are conditions, as described in Section 4.1, that must be considered in the process, to ensure balancing stability and for this, we can exploit the choice of the the arbitrary vector f_0 in Equation 5.2 to make sure that the contact forces are defined with respect to necessary conditions.

5.2.2 Desired Joint Torques

Recalling the derivation of the desired joint torques in 4.2.3 for the previous controller (4.19), where the set of solutions to the problem of computing the joint torques that results in achieving desired contact forces,

$$J_c M^{-1} S^T \tau = J_c M^{-1} (h - J_c^T f_c) - \dot{J}_c v \quad (5.3)$$

is given as,

$$\tau^* = \Lambda^+ \left(J_c M^{-1} (h - J_c^T f_c^*) - \dot{J}_c v \right) + N_\tau \tau_0 \quad (5.4)$$

$$\text{with } \Lambda = J_c M^{-1} S^T, \quad N_\tau = (I - \Lambda^+ \Lambda)$$

where Λ^+ is the Moore-Penrose pseudo inverse of Λ , N_τ is the null-space projector matrix, τ_0 is an arbitrary vector.

We will exploit the null-space projection to impose the secondary task of keeping the robot posture close to a desired configuration, with the same choice of the arbitrary vector as τ_0 as in the previous controller (4.19),

$$\tau_0 = S \left(g(q) - J_c(q)^T f_c^* \right) - K_{imp} (q_j - q_0) \quad (5.5)$$

For simplicity in the derivation which will follow, we group the terms in τ_0 as,

$$\tau_0 = m_{\tau_0} f_c^* + n_{\tau_0} \quad (5.6)$$

where,

$$m_{\tau_0} = -S J_c(q)^T \quad \text{and} \quad n_{\tau_0} = S^T g(q) - K_{imp} (q_j - q_0)$$

Substituting the expression for the desired feet contact forces from Equation 5.2 into the Equation 5.4 with Equation 5.6,

$$\begin{aligned} \tau^* = & \Lambda^+ \left(J_c M^{-1} \left(h - J_c^T \left(A^+ (\dot{H}^* - f_{grav}) + (I - A^+ A) f_0 \right) \right) - \dot{J}_c v \right) \\ & + N_\tau \left(m_{\tau_0} \left(A^+ (\dot{H}^* - f_{grav}) + (I - A^+ A) f_0 \right) + n_{\tau_0} \right) \end{aligned} \quad (5.7)$$

By grouping the multipliers of f_0 in τ^* ,

$$\begin{aligned} \tau^* = & \left[(N_\tau m_{\tau_0} - \Lambda^+ J_c M^{-1} J_c^T) (I - A^+ A) \right] f_0 \\ & + \left[(N_\tau m_{\tau_0} - \Lambda^+ J_c M^{-1} J_c^T) \left(A^+ (\dot{H}^* - f_{grav}) \right) \right. \\ & \quad \left. + N_\tau n_{\tau_0} + \Lambda^+ \left(J_c M^{-1} h - \dot{J}_c v \right) \right] \\ \tau^* = & X_\tau f_0 + Y_\tau \end{aligned} \quad (5.8)$$

with,

$$\begin{aligned} X_\tau &= (N_\tau m_{\tau_0} - \Lambda^+ J_c M^{-1} J_c^T) (I - A^+ A) \\ Y_\tau &= (N_\tau m_{\tau_0} - \Lambda^+ J_c M^{-1} J_c^T) \left(A^+ \left(\dot{H}^* - f_{grav} \right) \right) + N_\tau n_{\tau_0} + \Lambda^+ \left(J_c M^{-1} h - \dot{J}_c v \right) \end{aligned}$$

Let V be a function related to the norm of the joint torques that is to be minimized with respect to the arbitrary vector f_0 ,

$$V = \frac{1}{2} \|\tau\|^2 \quad (5.9)$$

Substituting 5.8 into 5.9,

$$\begin{aligned} V &= \frac{1}{2} \|X_\tau f_0 + Y_\tau\|^2 \\ &= \frac{1}{2} (f_0^T X_\tau^T + Y_\tau^T) (X_\tau f_0 + Y_\tau) \\ V &= \frac{1}{2} (f_0^T X_\tau^T X_\tau f_0 + 2Y_\tau^T X_\tau f_0 + Y_\tau^T Y_\tau) \end{aligned} \quad (5.10)$$

The square of the joint torque norm is to be minimized with respect to f_0 , so by extracting the terms from the expression 5.10 which are related to the arbitrary vector f_0 ,

$$V_{f_0} = \frac{1}{2} f_0^T (X_\tau^T X_\tau) f_0 + (X_\tau^T Y_\tau)^T f_0 \quad (5.11)$$

can be recognized as a candidate objective function of an optimization problem where we minimize the joint torques, with respect to the arbitrary vector f_0 , which will be used to compute the desired contact forces and related joint torques. Additionally, we want to impose conditions on the desired contact forces such that the balancing stability is ensured. Since these conditions are in terms of the feet contact forces, they can be imposed on the optimization problem as linear inequality constraints for the solution of f_0 . Hence the optimization problem can be formulated as a quadratic program the quadratic and linear terms of which are $X_\tau^T X_\tau$ and the $X_\tau^T Y_\tau$, respectively, as follows,

$$\begin{aligned} \min_{f_0} \quad & V(f_0) = \frac{1}{2} f_0^T (X_\tau^T X_\tau) f_0 + (X_\tau^T Y_\tau)^T f_0 \\ \text{subject to} \quad & A_{ineq} f_0 \leq b_{ineq} \end{aligned} \quad (5.12)$$

where the inequality constraints,

$$A_{ineq} = \begin{bmatrix} A_{fcone} \\ A_{zmp} \end{bmatrix}$$

will be formulated next.

Inequality Constraints of Feet Contact Forces

Now the conditions of feet contact forces, discussed in Section 4.1 are going to be defined in terms of f_0 to be included in the quadratic program as described by (5.12).

Friction cone can be considered in the optimization as linear inequalities obtained by using a numerical approximation of the cone via a set of segments and using the angular offsets and coefficients we can get the linear inequalities on the forces as;

$$A_{ineq_f} f \leq b_{ineq_f},$$

where,

$$A_{ineq_f} = \begin{bmatrix} -\alpha_i & 1 & -o_i c_f & 0 & 0 & 0 \\ \dots & \dots & \dots & 0 & 0 & 0 \\ \alpha_i & -1 & o_i c_f & 0 & 0 & 0 \\ \dots & \dots & \dots & 0 & 0 & 0 \end{bmatrix} \quad (5.13)$$

$$b_{ineq_f} = 0$$

where i denote the subscript for each approximated point, α_i denotes the angular coefficient of the line between i th and $(i+1)$ th approximation points and c_f denotes the friction coefficient. A_{ineq_f} is at each contact where and for two contacts,

$$A_{ineq_{fc}} = \begin{bmatrix} A_{ineq_f} & 0 \\ 0 & A_{ineq_f} \end{bmatrix} \quad b_{ineq_{fc}} = \begin{bmatrix} b_{ineq_f} \\ b_{ineq_f} \end{bmatrix} \quad (5.14)$$

can be used.

For the ZMP conditions we use a simple set of constraints related to the rectangular approximation of the feet and tipping moments necessary for those dimension.

5.3 Simulation Environment

Matlab¹ is used to simulate the balancing task. For this, we used mex-wholebodymodel project[8] which is a mex-C/C++ interface to the WBI (Whole Body Interface) components and which allows obtaining forward dynamics from Matlab. The forward dynamics is integrated over the duration of the simulation by the Matlab function 'ode15s'. The progress of the states and the contact forces are recorded and these results are then used in the visualizer part to animate the hole body motion. In the following two subsections, this process is explained in more detail.

5.3.1 Forward Dynamics

From Chapter 2, recalling the equations of motion of a constrained floating base robot (2.10) with an additional term,

$$M(q, v)\dot{v} + h(q, v) - J_c(q)^T f_c - Q_f = S^T \tau^* \quad (5.15)$$

$$s.t. \quad J_c(q)\dot{v} + \dot{J}_c(q, v)v = 0$$

where Q_f is added to add frictional effects, disturbances etc. at the joint space to evaluate the performance of the controller more realistically, if desired.

Taking the inverse of the mass matrix in the first part of 5.15 to obtain \dot{v} ,

$$\dot{v} = M^{-1} (-h + J_c^T f_c + S^T \tau^* + Q_f) \quad (5.16)$$

¹Matlab R2014a with student licence under TU/e

And substituting the above expression for \dot{v} into the constraint part of 5.15,

$$J_c M^{-1} (-h + J_c^T f_c + S^T \tau^* + Q_f) + \dot{J}_c v = 0 \quad (5.17)$$

Then we obtain the contact forces due to the contacts, which are assumed to be rigid, as,

$$f_c = (J_c M^{-1} J_c^T)^{-1} (J_c M^{-1} (h - S^T \tau^* + Q_f) - \dot{J}_c v) \quad (5.18)$$

For the integration of the forward dynamics, we form the vector,

$$x = \begin{bmatrix} q \\ v \end{bmatrix} \quad (5.19)$$

with,

$$q = \begin{bmatrix} p_{base} \\ Q_{base} \\ q_j \end{bmatrix} \in \mathbb{R}^{3+4+25} \quad v = \begin{bmatrix} \dot{p}_{base} \\ w_{base} \\ \dot{q}_j \end{bmatrix} \in \mathbb{R}^{3+3+25}$$

where p_{base} is the position of the floating base, Q_{base} is the orientation of the floating base in terms of quaternions, q_j is the joint angular positions, \dot{q}_j is the joint angular velocities and \dot{p}_{base} and w_{base} are the linear and the angular velocities of the floating-base respectively.

By forming the derivative of x as below, note that $\dot{q} \neq v$,

$$\dot{x} = \begin{bmatrix} \dot{q} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} \dot{p}_{base} \\ \dot{Q}_{base} \\ \dot{q}_j \\ \dot{v} \end{bmatrix} \quad (5.20)$$

where \dot{Q}_{base} is obtained by calculating the quaternion derivative from w_{base} and Q_{base} as,

$$\dot{Q}_{base} = \frac{1}{2} \begin{bmatrix} 0 & -w_{base}^T \\ w & -S(w_{base}) \end{bmatrix} Q_{base} + (1 - |Q_{base}|) Q_{base}$$

where also a numerical trick is used to make the norm of the Q_{base} equal to 1.

Substituting Equation 5.18 and Equation 5.16 into Equation 5.20,

$$\dot{x} = \begin{bmatrix} \dot{p}_{base} \\ \dot{Q}_{base} \\ \dot{q}_j \\ M^{-1} \left(-h + J_c^T (J_c M^{-1} J_c^T)^{-1} (J_c M^{-1} (h - S^T \tau^* + Q_f) - \dot{J}_c v) + S^T \tau^* + Q_f \right) \end{bmatrix} \quad (5.21)$$

To create the forward dynamics, all the other necessary terms such as M , h , J_c and $\dot{J}_c v$ in the forward dynamics part and g , H , J_{com} , p_{com} , p_{right} required specially for the controller to compute desired joint torques, are obtained by using mex-wholebodymodel interface.

We can then integrate \dot{x} over time to evaluate the states of the floating base robot with rigid contacts as described in 5.15 and desired torques obtained from the control law described by 5.8 where the arbitrary vector f_0 is obtained by the optimization problem 5.12. The integration

function 'ode15s', which solves stiff differential equations and differential algebraic equations with variable order method, seems to be the most efficient one to use compared to fixed order and non-stiff solvers, for our purpose.

After the integration, the states at each time step is stored and the movement of the robot along with other desired results can be visualized and plotted.

5.3.2 Matlab Visualizer

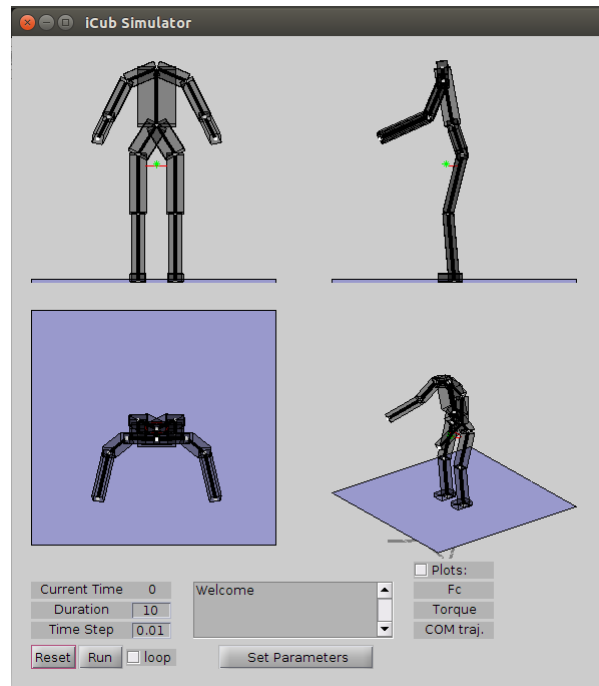


Figure 5.1: Matlab Visualizer for simulating iCub

The visualizer made in Matlab consists of a graphical user interface (GUI) which can be used to,

- Run simulations,
- Visualize the results on a robot figure,
- Change simulation parameters such as duration, time step etc. ,
- Edit the m-file where other parameters such as trajectory, controller gains, initial configuration etc. are defined and can be edited.

The visualizer along with the forward dynamics and controller implementation are all included in mex-wholebodymodel project² which is open-source.

²<https://github.com/robotology-playground/mex-wholebodymodel>

5.4 Results

In this section, several results from simulations and experiments will be presented. Completion of the primary task of moving the center of mass of the robot with a certain trajectory will be investigated through the error between desired and achieved trajectories. Task of minimization of the torques will be investigated by the norm of the joint torques. In addition, to see the influence of torque minimization on the feet contact forces, they will be investigated and the effect on internal torques will be discussed.

5.4.1 Simulations

To observe the minimization of joint torques with the new formulation, the results of three simulations with the same tasks but different controllers will be presented in this section.

The task for the center of mass of the robot to follow is a sinusoidal trajectory with an amplitude of 5 cm to the sides (y-direction) and a frequency of 1 rad/s (duration of approximately 6.28 seconds per cycle). Firstly, the previous controller without minimization using f_0 ; secondly, the current controller with no constraints on optimization and lastly the current controller with constraints on the contact forces is tested and results are presented as below.

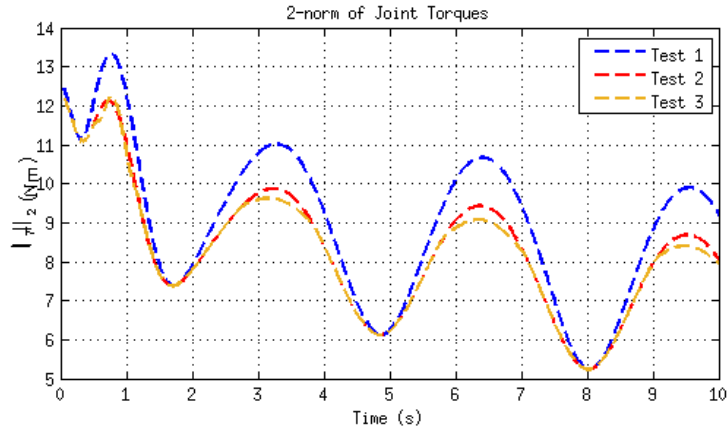


Figure 5.2

Simulation with the previous controller (without optimization)

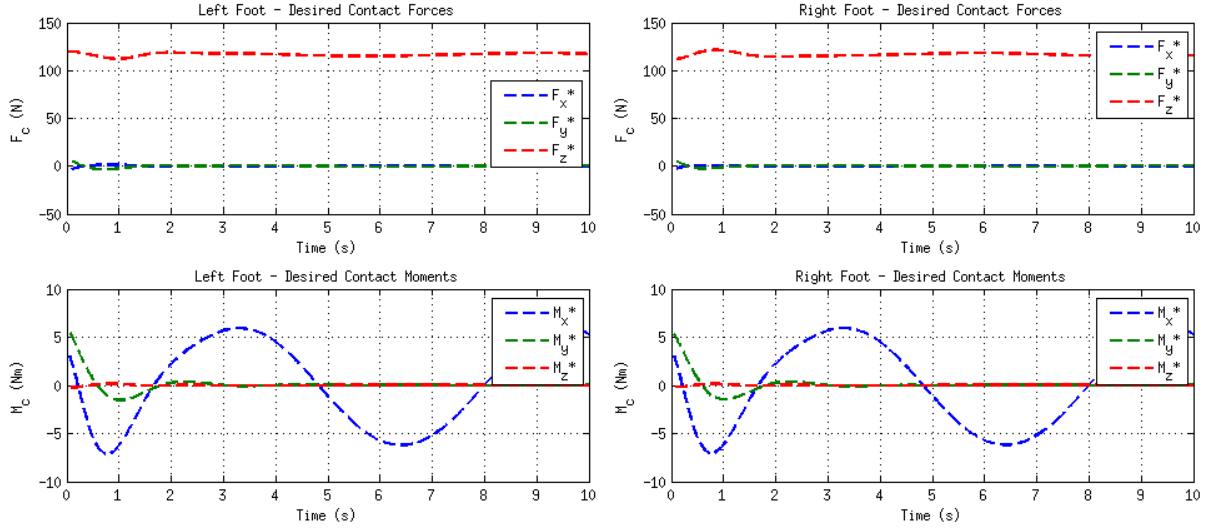


Figure 5.3: **Simulation with no QP** - Feet Contact Forces - left foot forces(upper left) & left foot moments(lower left) & right foot forces(upper right) & right foot forces(lower right)

From Figure 5.3, it can be seen that during the motion of the center of mass which is periodic from side to side, there is no significant difference between left and right foot contact forces. Although the vertical projection of the center of mass moves towards the feet while inside the support polygon, the desired contact forces are such that similar vertical forces are generated and it suggest that the controller results in extra internal joint torques to generate such contact forces even if it is close to being able to support itself on one foot. This can clearly be seen in the right part of Figure 5.4, where the robot is seen leaning to the sides as a result of the given task, but the desired resultant linear forces on the feet do not differ based on the projection of CoM. Additionally, it should be noted that the robot is able to perform the task of CoM trajectory with minor errors as seen in the lower left part of Figure 5.4 and the impedance task was also achieved which kept the posture of the robot close to its initial configuration. The norm of the joint torques of this test is presented in Figure 5.2 by the blue dashed line.

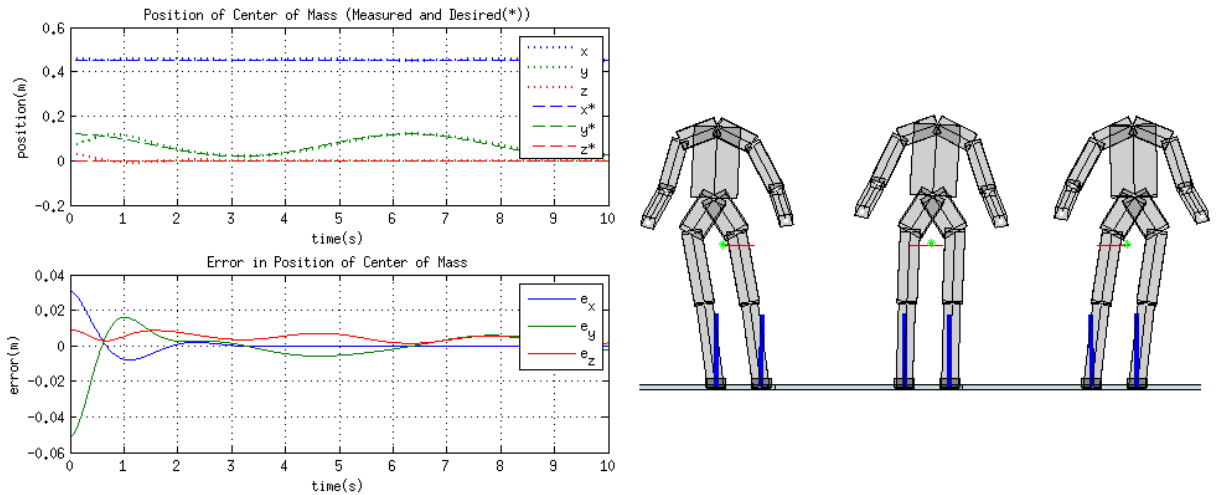


Figure 5.4: **Simulation with no QP** - Desired (dashed) and Measured (dotted) Center of Mass trajectories(upper left) & Error in CoM trajectory(lower left) & Snapshots of the robot from the test with contact forces as blue arrows(right)

Simulation with the new controller (with optimization)

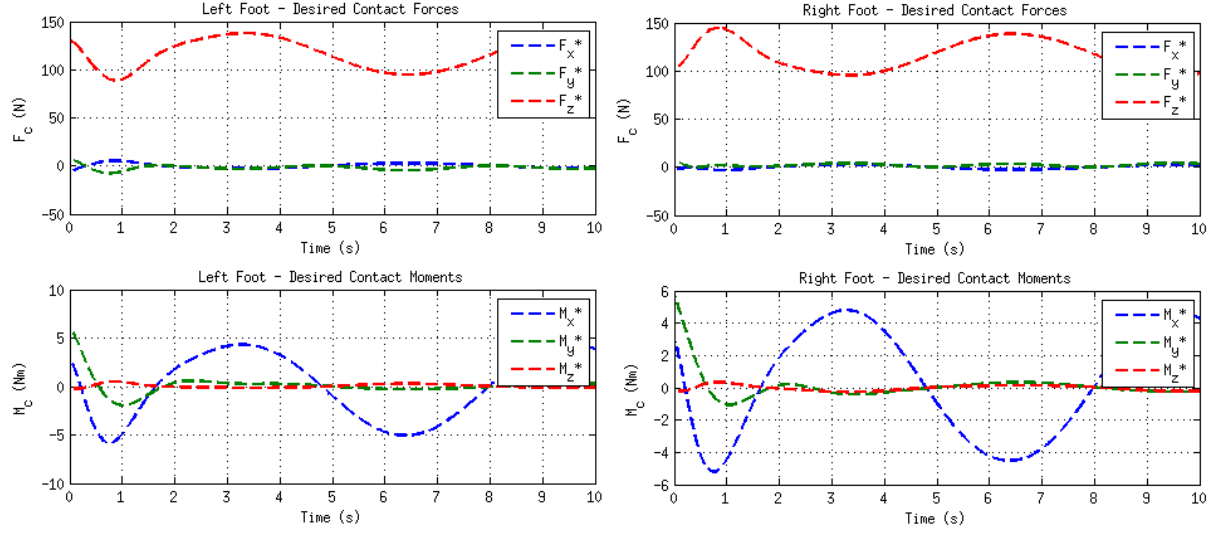


Figure 5.5: **Simulation with QP** - Feet Contact Forces - left foot forces(upper left) & left foot moments(lower left) & right foot forces(upper right) & right foot forces(lower right)

With the new controller, as explained in Section 5.2, desired feet contact forces are based on the minimization of joint torques. As a result of the new formulation, difference between right and left feet contact forces is observed as seen in Figure 5.5, as opposed to the previous test's results in Figure 5.3. Comparing also with the trajectory plot in the Figure 5.6, the desired vertical forces on both feet are oppositely harmonious with the periodic motion of CoM. In other words, as the robot leans to one side, the new controller results in a desired vertical force on that corresponding foot greater than the other one, as it can also be seen by the blue arrows depicting net feet contact forces in Figure 5.6's right side. In Figure 5.2, the joint torque norm of this test can be found (dashed red) and it can clearly be seen that the minimization of the joint torques is achieved. It is also interesting to note that the instances when both norms get close to each other are when the robot's CoM position is crossing the center of both feet and the minimization factor is maximum when the CoM is at the extremes.

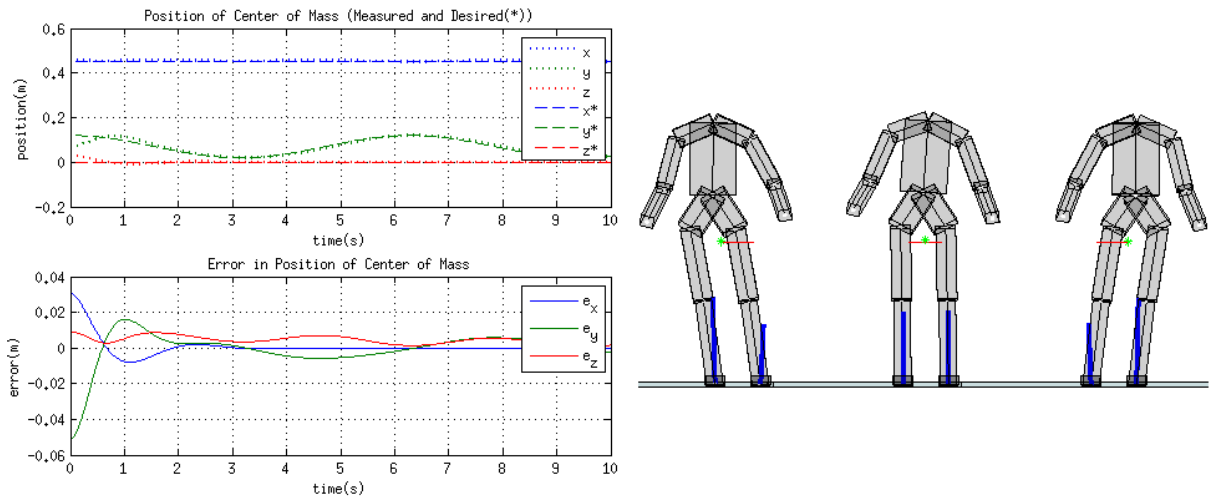


Figure 5.6: **Simulation with QP** - Desired (dashed) and Measured (dotted) Center of Mass trajectories(upper left) & Error in CoM trajectory(lower left) & Snapshots of the robot from the test with contact forces as blue arrows(right)

Simulation with the new (with optimization and a strict constraint)

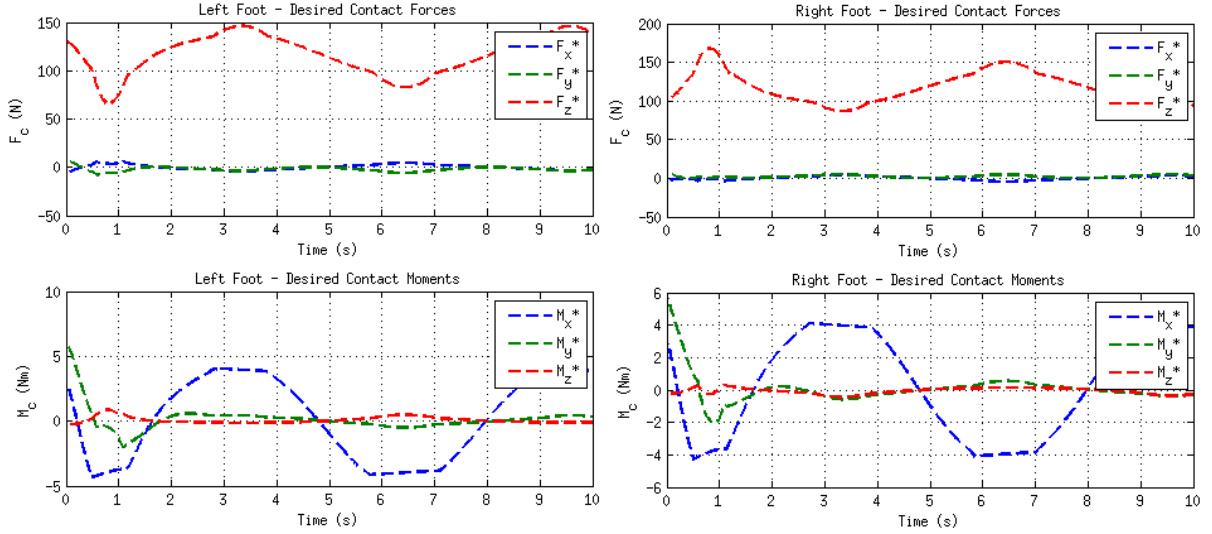


Figure 5.7: **Simulation with QP and a strict constraint** - Feet Contact Forces - left foot forces(upper left) & left foot moments(lower left) & right foot forces(upper right) & right foot forces(lower right)

To clearly observe how imposing a constraint will affect the completion of the task when it becomes active, a strict constraint on the desired contact moment in x-direction (M_x^*) along with other constraints formulated in Section 5.2.2 is introduced. With the use of quadratic programming, minimal joint torques and feet contact forces that satisfy these constraints can be found if the constraints allow such solution. As seen in Figure 5.7, a limit of 4 Nm on $|M_x^*|$ is achieved and the motion task along with impedance task (Figure 5.8) is achieved in a similar way to the previous tests. However, relying on QP for a solution under such a strict constraint is not desirable for it increases the computational load in an unpredictable way for each control loop. Still this test is important, because regarding the side stepping and walking tasks in the future, an attempt to find a solution to achieve the task and which is also satisfied by the constraints may allow to anticipate when a recovery action such as a side step will be necessary. Additionally, in Figure 5.2, a strict constraint becoming active seems to minimize the torques although it is actually due to the change in achieved CoM trajectory which is slightly different.

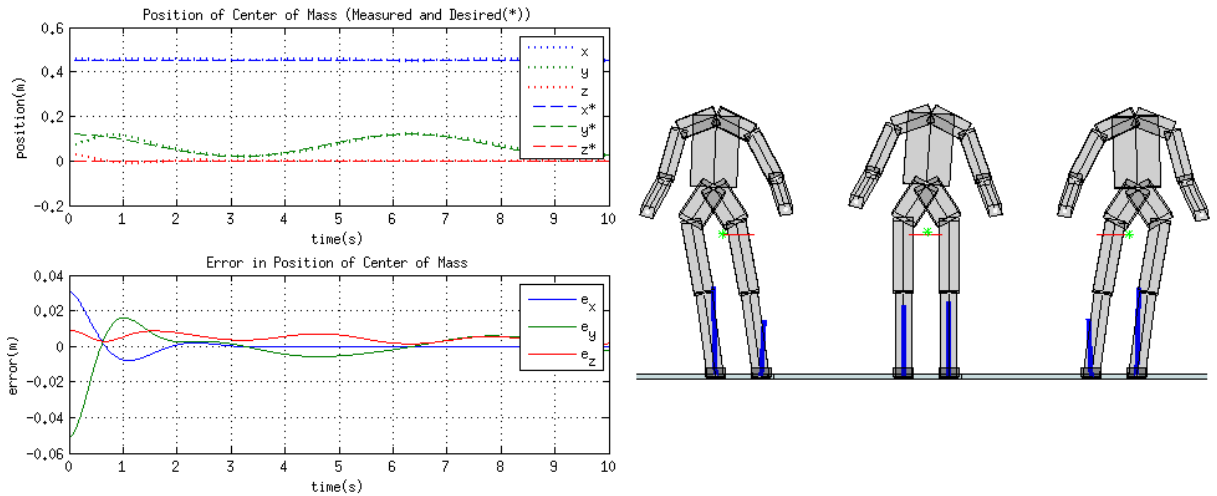


Figure 5.8: **Simulation with QP and a strict constraint** - Desired (dashed) and Measured (dotted) Center of Mass trajectories(upper left) & Error in CoM trajectory(lower left) & Snapshots of the robot from the test with contact forces as blue arrows(right)

5.4.2 Experiments on the Robot (iCubGenova01)

Experiments are done on the robot iCubGenova01 ('black iCub') which is currently in IIT, Genova, Italy. Here, the results from tests with two kinds of trajectory tasks for the CoM carried out with the previous and the new controller are presented.

The first task is keeping the CoM at the initial configuration (constant trajectory). This task is tested with the previous and the new controller. Feet contact forces and CoM trajectories are presented separately whereas the joint torque norms are presented together. It should be noted that for about 10 seconds initially, the robot is not disturbed and after about 10 seconds from the start, external disturbances are introduced in the order of a downward push on the left arm, a downward push on the right arm, a downward push on both arms and a backwards push on the torso. The measured forces and torques are included in the results to compare them with the desired values even though there is no feedback law relating them in the controller. This is done to test the robustness of such a force and posture controller which does not use the error between desired and measured forces and torques and this also allows to evaluate motor joint torque control and the whole body estimation of joint torques and contact forces.

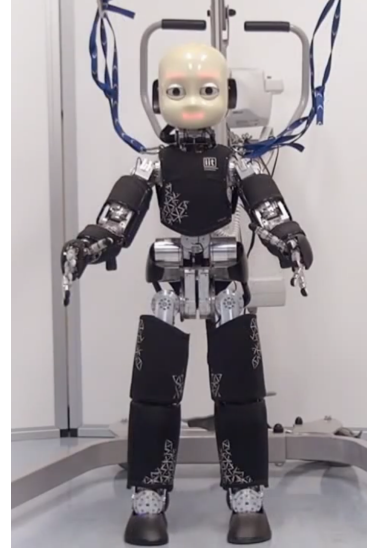


Figure 5.9: The Black iCub - *iCubGenova01*

The second task is a periodic trajectory given to the CoM, but the parameters defining the task for the new controller is slightly different. During tests, it was seen that with the previous controller, the robot is not able to achieve a movement of 3 cm for its CoM to the sides, even if it is a minimum-jerk trajectory generated using the results of [Pattacini 2010]. Hence we were able to test periodic motions with amplitudes only smaller than 3 cm and we tested the previous controller with 2.5 cm of amplitude and with a frequency of 0.3 Hz (duration of approximately 3.3 seconds per cycle). Then for the new controller we present the results of a task with an amplitude of 4 cm with a frequency of 0.25 Hz.

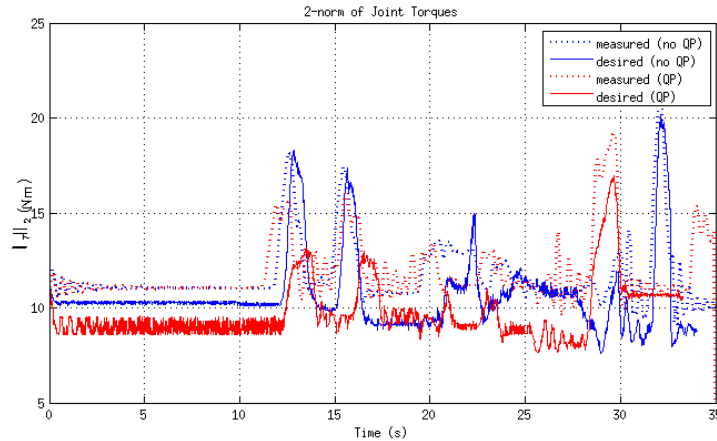


Figure 5.10

Experiment with the previous controller (no optimization) for constant trajectory

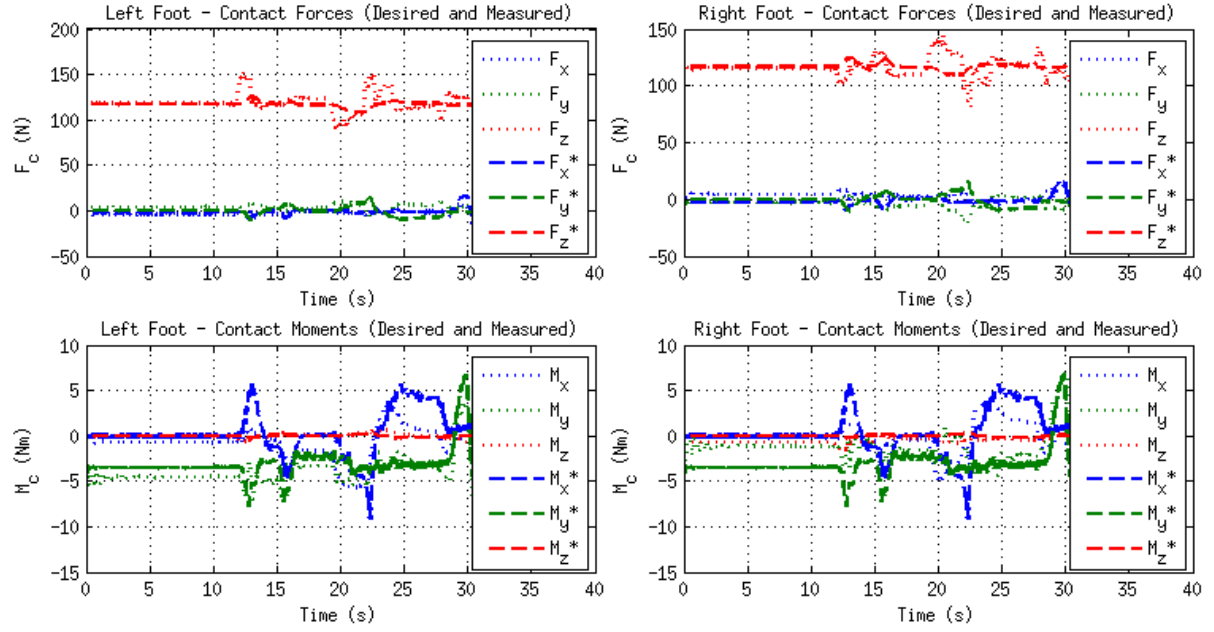


Figure 5.11: **Experiment with no QP** - Desired (dashed) and Measured (dotted) Feet Contact Forces - left foot forces(upper left) & left foot moments(lower left) & right foot forces(upper right) & right foot moments(lower right)

First 10 seconds in Figures 5.11 and 5.12, the robot keeps its CoM at its initial position with minor errors, with the previous controller. To test if it is also robust to external disturbances, after 10 seconds, the robot is disturbed from its arms and torso. After each disturbance, the measured contact forces are observed to converge to their desired values and the error CoM trajectory is minimized after the external disturbances. Eventhough the controller does not have any information about these newly introduced contacts, it reacts in a safe manner just to achieve desired contact forces and joint torques rather than positions, which is an advantage of using such force and postural control framework. Additionally in Figure 5.15, the norm of the desired and the estimated joint torques from this test is depicted with the blue line and the blue dotted line, respectively. Mismatch is seen between the desired and estimated torques and currently the estimation process is still under development.

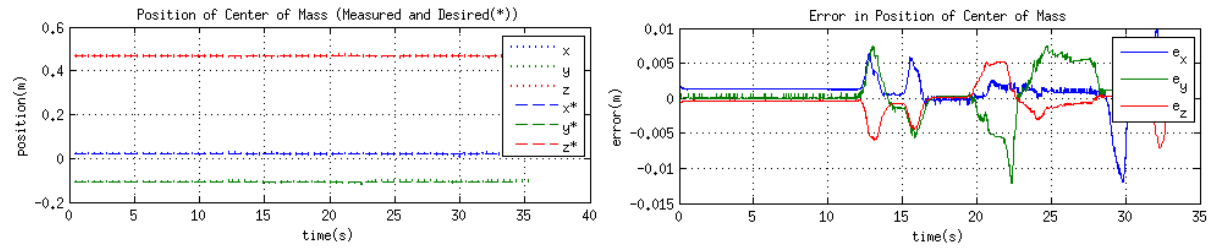


Figure 5.12: **Experiment with no QP** - Desired (dashed) and Measured (dotted) Center of Mass trajectories(upper left) & Error in CoM trajectory(lower left) & Snapshots of the robot from the test with contact forces as blue arrows(right)

Experiment with the new controller (with optimization) for constant trajectory

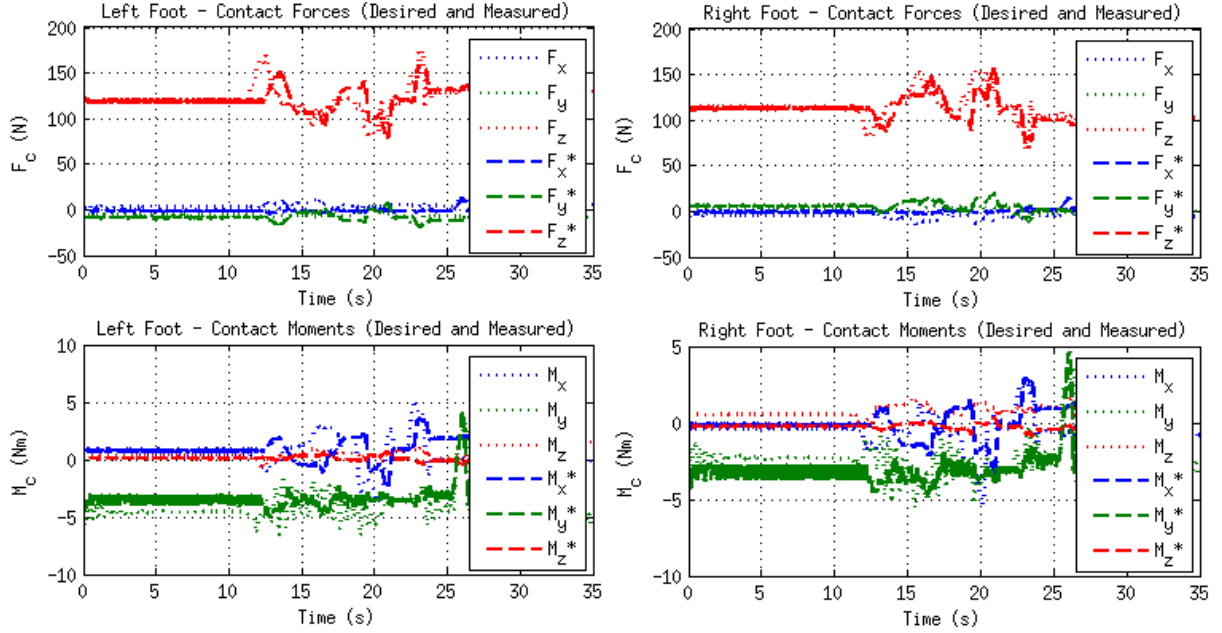


Figure 5.13: **Experiment with QP** - Desired (dashed) and Measured (dotted) Feet Contact Forces - left foot forces(upper left) & left foot moments(lower left) & right foot forces(upper right) & right foot moments(lower right)

In this test with the new controller with torque minimization, the task is the same as the previous test (constant trajectory of CoM in the center). The aim is to observe the motion and contact forces in the cases of without and with external disturbances. The feet contact forces in Figure 5.13 and the CoM trajectory and its error in Figure 5.14 show that during the no disturbance phase in the initial 10 seconds, the robot does not move and it keeps its CoM at the desired position with a minor error. Also when the robot is disturbed, as the robot posture changes, desired contact forces changes and the deflection of the CoM can also be seen in the error plot. However, errors are not kept constant and as the contacts are released, error CoM position gets smaller again. As in the previous test, we see the advantage of using a force and posture controller as it allows the robot have safe interactions with its environment even if it is not aware of these interactions as external disturbances. Also the minimization of torques can be seen in Figure 5.10, where the norm of desired joint torques from this test is depicted in red which is below the norm of the torques from the previous test most of the time even under external disturbances.

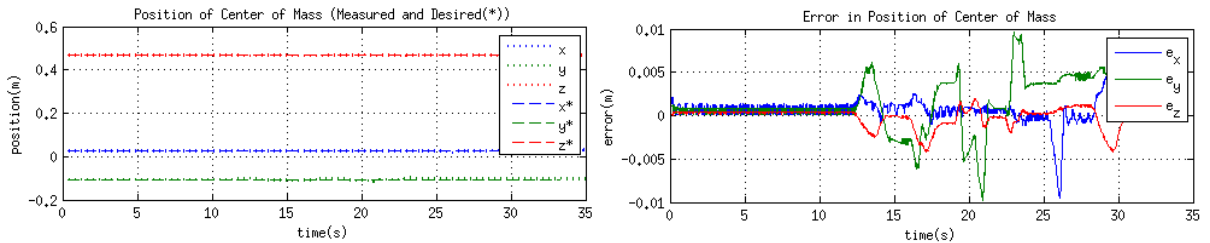


Figure 5.14: **Experiment with QP** - Desired (dashed) and Measured (dotted) Center of Mass trajectories(upper left) & Error in CoM trajectory(lower left) & Snapshots of the robot from the test with contact forces as blue arrows(right)

Experiment with the previous controller (no optimization) for periodic trajectory

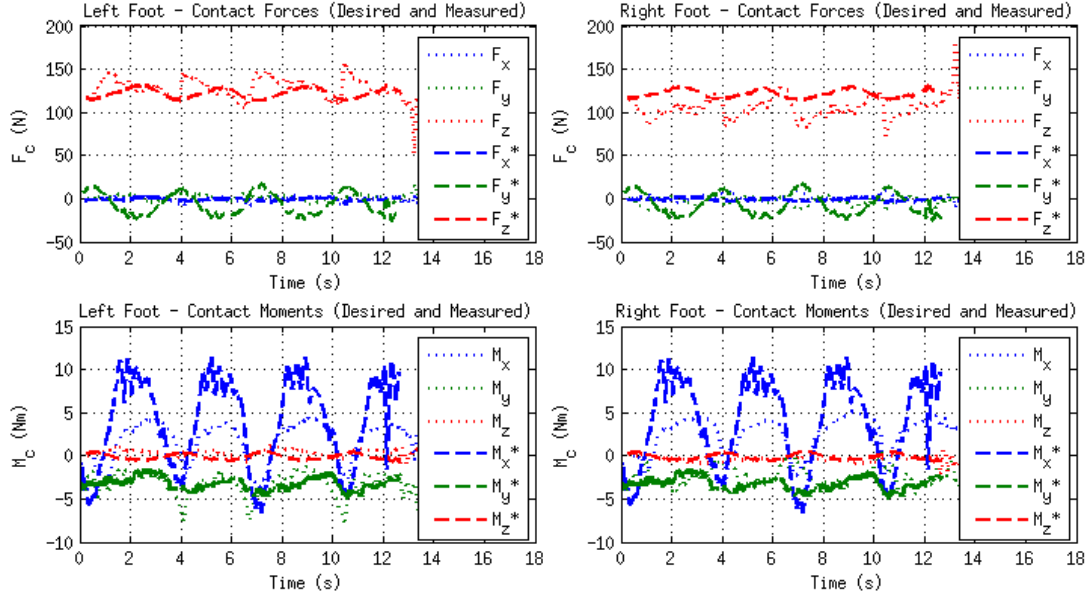


Figure 5.15: **Experiment with no QP** - Desired (dashed) and Measured (dotted) Feet Contact Forces - left foot forces(upper left) & left foot moments(lower left) & right foot forces(upper right) & right foot moments(lower right)

In this test, a task of periodic motion of CoM with an amplitude of 2.5 cm and 0.3 Hz is given to the robot using the old controller. Resulting desired feet contact forces in Figure 5.15 are in a harmony with the task and with each other too, meaning that regardless of the direction causing a symmetry in the posture of the robot, moving CoM sideways results on desired feet contact forces that are similar to each other. However measured feet contact forces show trends that are opposite to each other in the sense that when vertical force on one foot increases as the CoM moves to that side, the vertical force on the other foot decreases. Computation of desired joint torques from such desired contact forces that require more effort and that may or may not be achievable is suspected to result in unnecessary internal torques. We also suspected that the reason the robot is not able to move its CoM 3 cm and more to the sides with the previous control is the generation of internal torques to achieve such contact forces.

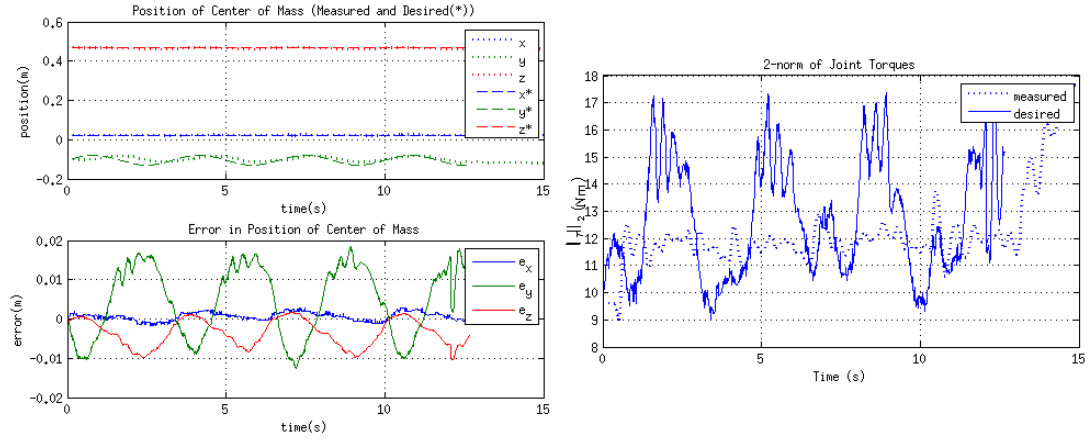


Figure 5.16: **Experiment with no QP** - Desired (dashed) and Measured (dotted) Center of Mass trajectories(upper left) & Error in CoM trajectory(lower left) & Snapshots of the robot from the test with contact forces as blue arrows(right)

Experiment with the new controller (with optimization)

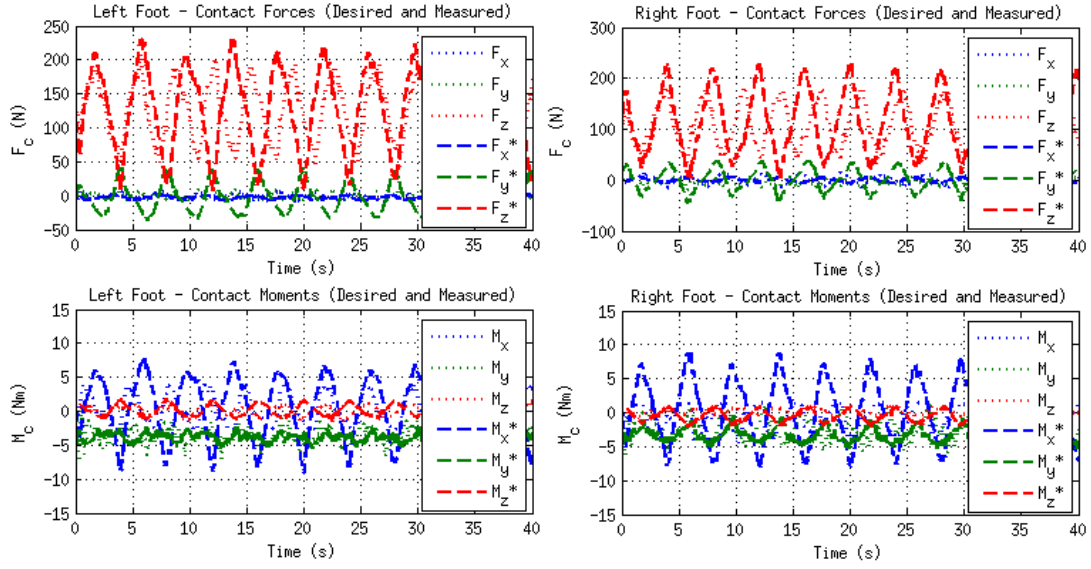


Figure 5.17: **Experiment with QP** - Desired (dashed) and Measured (dotted) Feet Contact Forces - left foot forces(upper left) & left foot moments(lower left) & right foot forces(upper right) & right foot moments(lower right)

In this test³, a task for the motion of CoM with an amplitude of 4 cm and a frequency of 0.25 Hz is given to the robot with the new controller implemented. With the new controller, the robot is seen to be able to achieve motions that are highly dynamic relative to the previous tasks. As seen in the Figure 5.17, the desired contact forces are now related to the CoM position and the asymmetry of the robot during the motion. In fact the desired vertical forces seem to oscillate as the CoM motion progresses but with a different phase from each other. Also from Figure 5.18's right part, the norm of joint torques seem to oscillate between similar maximum and minimum values with the previous test (Figure 5.16) even though the amplitude of motion is increased from 2.5 to 4 cm.

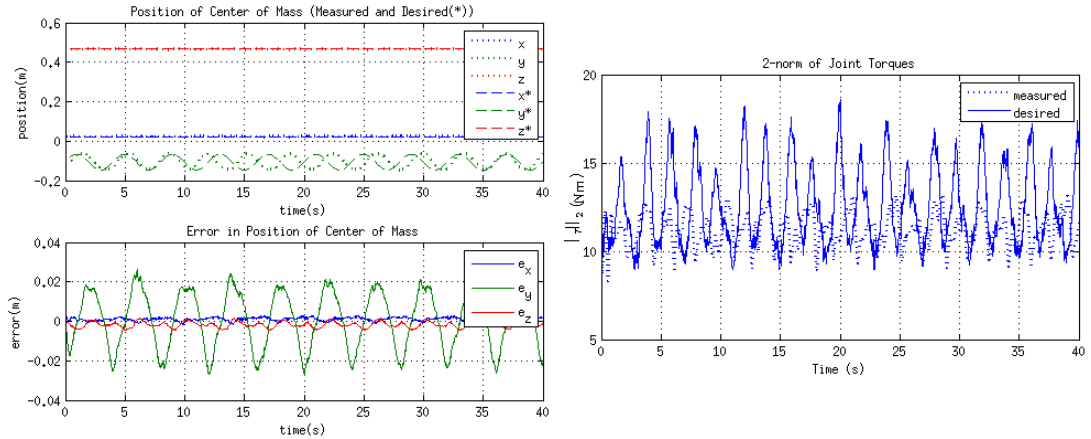


Figure 5.18: **Experiment with QP** - Desired (dashed) and Measured (dotted) Center of Mass trajectories(upper left) & Error in CoM trajectory(lower left) & Snapshots of the robot from the test with contact forces as blue arrows(right)

³The measured data (dotted) should be ignored for this experiment. It is due to non-real time operation of two computers collecting data and it will be explained in the next section.

5.5 Discussion

Experimental results showed that with the new formulation of the desired contact forces that relate to the minimization of desired joint torques allows the robot to complete CoM tasks which it was not able to. As we examined for both controllers the desired feet contact forces and resultant joint torque norms, we saw that with the new formulation, the desired contact forces are computed differently which results in a better joint torque distribution among the robot, helping in minimizing the torques as well as moving in a more relaxed and flexible way.

Although the new controller helped in exposing the mechanical capabilities of the robot, there are some issues about the implementation on the experimental setup. Firstly we have noted that neither in the old nor the new controller, there is no feedback from measured contact forces inside the controller. In a way, it relies on the fact that the modeling of the robot and the estimation of the internal wrenches and joint torques are accurate and thus it is possible to accurately compute joint torques that will result in desired contact forces. During the experiments and analysis, we mostly refer to the desired values given by the controller, because it is observed in several cases that the desired and the estimated joint torques differ depending on the current states of the robot. This is an ongoing work carried out by the group and for now it proves to be sufficient for balancing task. Since there is no feedback law for the contact forces, differences between measured and desired contact forces only helps in our analysis to show that with such a framework, despite the uncertainties in the estimation of forces and torques, balancing on both feet and motion tasks for CoM can be achieved even under external disturbances.

The frequency of the high and low level control loops is an important issue that can have a significant effect on the whole body force and postural control. The frequency that is used during testing is 100 Hz (10 ms per loop). With the implementation of the optimization under constraints, unless it is given a challenging task that require a faster motion and activate the constraints, the robot could be controlled at the same frequency of 100 Hz with the optimization too. However this is still slow for controlling a humanoid robot. During tests, we have seen that when some of the constraints on feet contact forces became active (e.g. when the robot is about to slip its foot), the quadratic solver required more time for the solution and if the motion task is challenging, the controller was not able to keep up with the setup in real-time. Additionally, the joint torque control is done at a high-level where it also has a frequency of 100 Hz. There is an ongoing work too, to implement this joint torque control along with estimation of joint torques back into the firmware on the robot which can result in a low-level control with a frequency higher than that of the high-level control. Also currently the controllers are tested on the robot by implementing them into Matlab Simulink models. This makes the process easier but also it creates additional delays in communication and computational effort. That is why during the tests that are presented in this report, two computers were used to collect data to ease the load on one. However due to operating in non-real time and different computational loads, there were differences in the sampling frequencies for both computers. A promising alternative such as using C/C++ for programming controllers for the robot in an efficient way is an ongoing work and moving the control part into a more efficient framework can improve in completing more challenging tasks.

Due to the reasons above, we have gone as far as being able to minimize joint torques based on the new formulation with quadratic programming, but challenging tasks which would result in activating some constraints could not be thoroughly tested, because in such cases, the control loop takes longer than 10 ms, the robot generates jerky motions and it becomes risky as the robot or the things it can interact can get damaged. For less challenging and slower tasks, this was no issue. Despite the fact that the challenging tasks were pushing the computational

effort to the limits, in those cases, the quadratic solver indicates (and therefore the controller knows) when the constraints are becoming active and the motion is becoming more challenging (maybe impossible) and decisions on the control side can be made to take an action to recover from that situation such as a step. Hence, instead of relying on the solution of the optimization problem that finds the feet contact forces while respecting inequality conditions and minimizing the torques, it can be used as an indicator to change the control strategy on the go, which can be useful for the walking task.

Conclusion

With the implementation of torque minimization via a new formulation as a quadratic optimization problem for finding the desired contact wrenches and joint torques, it is seen that the movements of the robot has become more relaxed and easy, such that it can achieve tasks on moving its center of mass with extents and speeds it was not able to achieve before. Along with torque minimiation, we implemented in the optimization problem the conditions on feet contact forces. It was observed that for tasks that we can call less challenging, the controller was able to follow given tasks that required the quadratic program solver to find proper set of solutions when the constraints became active. However, for faster motions, which were more challenging in terms of computational load too, it was not observed. Nevertheless, it is fair to state that with certain conditions on the operation of the optimization problem, the state of the control can be monitored and the cases of instabilities due to contact states changing can be anticipated beforehand and necessary control action can be taken.

Bibliography

- [1] G. Metta, L. Natale, F. Nori, G. Sandini, D. Vernon, L. Fadiga, C. von Hofsten, K. Rosander, M. Lopes, J. Santos-Victor, A. Bernardino, and L. Montesano, “The icub humanoid robot: an open-systems platform for research in cognitive development,” *Neural Networks*, 2010.
- [2] M. Fumagalli, S. Ivaldi, M. Randazzo, L. Natale, G. Metta, and G. Sandini, “Force feedback exploiting tactile and proximal force/torque sensing. theory and implementation on the humanoid robot icub,” *Autonomous Robots*, 2010.
- [3] iCub Specifications, “<http://www.icub.org/images/brochures/icubflyer.pdf>,”
- [4] the iCub Wiki, “http://wiki.icub.org/wiki/icub_software_installation,”
- [5] A. D. Prete, L. Natale, F. Nori, and G. Metta, “Contact force estimations using tactile sensors and force / torque sensors,” *Human Robot Interaction*, 2012.
- [6] S. Ivaldi, M. Fumagalli, M. Randazzo, F. Nori, G. Metta, and G. Sandini, “Computing robot internal/external wrenches by means of inertial, tactile and f/t sensors: theory and implementation on the icub,” *Proc. of the 11th IEEE-RAS International Conference on Humanoid Robots, Bled, Slovenia*, 2011.
- [7] A. Del Prete, *Control of Contact Forces using Whole-Body Force and Tactile Sensors: Theory and Implementation on the iCub Humanoid Robot*. PhD thesis, Universita di Genova, Istituto Italiano di Tecnologia, Genoa, February 2013.
- [8] mex-wholebodymodel project GitHub Repository, “<https://github.com/robotology-playground/mex-wholebodymodel>,” *Robotology Playground*.
- [9] D. E. Orin, A. Goswami, and S.-H. Lee, “Centroidal dynamics of a humanoid robot,” *Autonomous Robots*, 2013.
- [10] M. W. Spong, “The control of underactuated mechanical systems,” *Plenary Address at the First International Conference on Mechatronics*, 1994.
- [11] A. Del Prete, N. Mansard, F. Nori, G. Metta, and L. Natale, “Partial force control of constrained floating-base robots,” *Intelligent Robots and Systems (IROS)*, 2014.
- [12] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 1999.