# A Whole-Body Stack-of-Tasks compliant control for the Humanoid Robot COMAN

Alessio Rocchi* Enrico Mingo Hoffman* Edoardo Farnioli*† Nikos G. Tsagarakis*

*Abstract*—**A fundamental aspect of controlling humanoid robots is the capability to use the entire body to perform tasks. In this paper we present an ongoing work to add this capability to the compliant humanoid robot COMAN, designed at the Italian Institute of Technology. Our control architecture is composed by a high level, whole-body inverse kinematic solver and a decentralized, low level, joint impedance control. Such architecture allows to regulate impedance using different strategies maintaining a high level of robustness and it has been developed to perform rescue operations in disaster scenarios.**

***Keywords*—*Whole-Body Control, Humanoid Bipedal Robot, Stack of Tasks, Compliance***

## INTRODUCTION

The recent DARPA Robotics Challenge showed all the difficulties that arise when advanced research topics need to be connected to practical tasks. In particular these tasks have to be performed in unstructured environments where classical stiff robots and controllers are not well suited. The main problem in real world scenarios are the uncertainties associated with the robot state and the environment state. These uncertainties make the interaction between the robot and the environment potentially dangerous. On the other hand, stiff controllers such as the classical PID position control are robust and can be implemented in a decentralized way. We think decentralized control schemes are fundamental in robots that have to operate continuously, without the possibility to be turned off. Let's take for example a semi-autonomous and tele-operated humanoid bipedal robot: it has always to keep balance also in the case in which a recoverable hardware or software failure occurs on its on-board computer, or the communication with the robot is temporarily severed. Such scenario would require a restart of the control modules: if the robot is in a stable configuration the decentralized joint impedance control scheme can maintain the standing posture in the robot up to the end of the restart procedure, where a centralized control scheme could not be restarted without first bringing the robot to a safe rest configuration.

With these considerations in mind, pure torque controllers such as in the case of operational space control schemes [8] are not well suited to systems where robustness is critical, but at the same time stiff position controllers are not suited for the interaction as well. A trade-off between the two approaches can be represented by a combination of a centralized pure kinematic control together with joint impedance control. In fact, the latter can be implemented in a decentralized way and it adds compliance to the system allowing the possibility to regulate the impedance both at the joint level and at the Cartesian level through conservative congruence transformation [9].

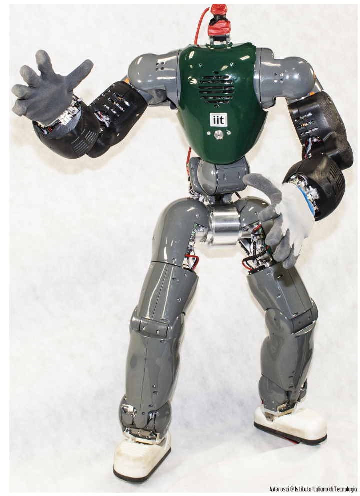


Fig. 1. COMAN is a humanoid bipedal robot equipped with series elastic actuators (SEA) and torque controlled. It has 29 DOFs, 4 Force/torque sensors (two in the ankles and two in the arms) and one IMU placed in the waist.

In this paper we present the control architecture that we are developing based on a Cartesian whole-body trajectory generator and a joint impedance controller for the humanoid bipedal robot COMAN [13] (Figure 1), developed at the Italian Institute of Technology. First we introduce some of the work done in the Whole-Body Inverse Kinematics/Dynamics compliant control then we introduce our methodology, finally we show some experimental results obtained on the humanoid robot COMAN.

## RELATED WORKS

Kinematic and dynamic inversion are well known problems in robotics. In general, given some tasks specified in Cartesian position, velocity, accelerations or forces, we want to find the joints position, velocity, accelerations or torque that realize those tasks. Many solutions to this problem have been presented for single and multiple kinematic chains [12] [14].

An interesting subset of these algorithms are the ones based on numerical optimization. If on one side numerical optimization could be less efficient than other algorithms at solving the kinematic inversion problem, on the other side they allow for the explicit introduction of unilateral/bilateral constraints in the inverse kinematic/dynamic problem, which are fundamental when working on the real hardware.

## DECENTRALIZED JOINT IMPEDANCE CONTROL

Joint Impedance Control is a well known way to control a robot ensuring a compliant behaviour. The torque sent to $n$ actuators is computed as

$$\tau = K_q(q_d - q) - K_d\dot{q} + g(q) \quad (1)$$

where $q \in \mathbb{R}^n$ and $q_d \in \mathbb{R}^n$ are respectively vectors of actual and desired joint positions, $\dot{q} \in \mathbb{R}^n$ is the vector of actual joint velocities, $K_q \in \mathbb{R}^{n\times n}$ is a positive definite joint stiffness matrix, $K_d \in \mathbb{R}^{n\times n}$ is a positive definite joint damping matrix and $g(q) \in \mathbb{R}^n$ is a vector of gravity compensation torques.

If we compute (1) in a centralized way then $K_q$ and $K_d$ are symmetric [10]. If we compute (1) in a decentralized way then $K_q$ and $K_d$ are diagonal since the law (1) is implemented locally at each actuator, which implies that is not possible to obtain any Cartesian stiffness at the end effector, rather we need to set an optimization problem to achieve a "close" Cartesian stiffness behavior according to some metric [1]. Notice that in both the cases, the $g(q)$ term as to be computed in a centralized way since it depends on the robot whole posture.

## WHOLE-BODY INVERSE KINEMATICS

Without loss of generality, we assume that for the $i^{\text{th}}$ task we have a proper function $e(q)$ describing the task error. In this hypothesis, the time derivative of the error can be computed as

$$\dot{e}_i = \frac{\partial e_i}{\partial q}\dot{q} = J_i\dot{q}.$$

The convergence of the method can be assured by imposing an exponential dynamic for the task error, that is

$$\dot{e}_i = -\lambda e_i.$$

If the robot is redundant with respect to a task, secondary tasks can be also executed without affecting the performances of the first one. Supposing we have a set of tasks, each one described by a couple $T_i = (e_i, J_i)$, we would like the robot to execute them at the best of its capabilities. In order to obtain this result, we define two kinds of relationships between tasks: hard priority and soft priority. While looking for a solution, the performance in executing a task should never be deteriorated by executing a task with a hard, lower priority, while tasks put in different, soft priorities can influence the performance of each other if the error in executing each task becomes too large. The idea of executing the set of tasks has a well-known solution in the stack-of-tasks, where hard priorities are enforced by the order of the task in the stack. To take into account also soft priorities, we then need to use the augmented Jacobian formulation, and define $J$ as the augmented Jacobian of our humanoid robot as in [2]. It must be noted how the augmented Jacobian formulation alone can not enforce hard priorities since putting many tasks together can generate an augmented Jacobian matrix which is not well conditioned. In our case to generate whole-body motions for our humanoid robot a series of QP (Quadratic Programming) problems in cascade is solved [7]. This is a well known method to derive motions by executing tasks adding bilateral constrains to the inverse kinematics problem [4].

The problem can be formulated as a stack of quadratic programs. For the first task, we have:

$$\begin{aligned} \dot{q}_1 = \quad & \underset{\dot{q}}{\text{argmin}} \quad \|J_1\dot{q} + \lambda e_1\| \\ & s.t. \quad \mathrm{l}_1 \leq A_1\dot{q} \leq \mathrm{u}_1. \end{aligned} \quad (2)$$

The constraints form used in (2) for the constraints can be profitably used to express lower and upper bounds for the variable value as well as equality constraints, corresponding to using $l_i = b_i$, or unilateral constraint, corresponding to the case $l_i = -\infty$ or $b_i = \infty$). According to the classical stack-of-task method based on null-space projection [11] the tasks which needs to be projected in the null-space of the previous task (hard priority), for the simple case of two hard priorities, can be written as

$$\dot{q} = -\lambda_1 J_1^+ e_1 + \left(I - J_1^+ J_1\right)(\lambda_2 J_2^+ e_2). \quad (3)$$

An equivalent formulation can also be obtained as a stack of QP programs, which also allows for the introduction of unilateral and bilateral bounds, solving then a constrained inverse kinematics (CIK) problem. In general, the $n^{\text{th}}$ task will then be defined as

$$\begin{aligned} \dot{q}_d = \quad & \underset{\dot{q}}{\text{argmin}} \quad \|J_n\dot{q} + \lambda e_n\| \\ & s.t. \quad A_1\dot{q} = A_1\dot{q}_1 \\ & \qquad \vdots \\ & \qquad A_{n-1}\dot{q} = A_{n-1}\dot{q}_{n-1} \\ & \qquad \mathrm{l}_1 \leq A_1\dot{q} \leq \mathrm{u}_1 \\ & \qquad \vdots \\ & \qquad \mathrm{l}_{n-1} \leq A_{n-1}\dot{q} \leq \mathrm{u}_{n-1} \\ & \qquad \mathrm{l}_n \leq A_n\dot{q} \leq \mathrm{u}_n \end{aligned} \quad (4)$$

where $\dot{q_d}$ is the desired velocity (control variable). In (4) the previous solutions $\dot{q}_i, i < n$ are taken into account with constraints of the type $A_i\dot{q} = A_i\dot{q}_i \ \forall i < n$, so that the optimality of all higher priority tasks is not changed by the current solution.

While in (4) the first task has a relationship of hard priority with respect to the second, and so on, for each *level* of priority, a soft priority relationship between tasks can be imposed using the augmented Jacobian technique. To this aim, introducing the relative weights $\beta_i$, the augmented Jacobian and the error vector can be rewritten as

$$\begin{aligned} J_{\text{aug}} &= \begin{bmatrix} \beta_1 J_1^T & \beta_2 J_2^T & \dots & \beta_n J_n^T \end{bmatrix}^T \\ e_{\text{aug}} &= \begin{bmatrix} \beta_1 e_1^T & \beta_2 e_2^T & \dots & \beta_n e_n^T \end{bmatrix}^T, \end{aligned}$$

where the soft priority between tasks can then be modified by changing the relative weights $\beta_i$, and in particular having higher priority tasks for bigger $\beta_i$. Building stacks of augmented tasks allows to mix hard and soft priorities seamlessly. Tasks at a lower level in the stack can not influence optimality of tasks at higher levels in the stack, while tasks at the same level in the stack can influence each other performance. The solution obtained can then be sent directly to a velocity controlled robot or integrated in a position controlled robot as

$$q_d = q + \dot{q}\delta t \quad (5)$$

where $\delta t$ is the control loop period.

## EXPERIMENTS

### *The Solver*

qpOASES [6] is an open-source C++ implementation of an online active set strategy [5] QP solver which is part of the ACADO suite. It implements several automatic regularization techniques, implements warm-start and allows to specify initial guesses to speed up the optimization (e.g., usually used with the previous solution)

### *Tasks description*

In our case, the first task is, according to the augmented Jacobian formulation [3], equal to

$$\begin{aligned} J_1 &= \begin{bmatrix}\beta_{11} J_{\text{l\_wr}}^T & \beta_{12} J_{\text{r\_wr}}^T & \beta_{13} J_{\text{CoM}}^T & \beta_{14} J_{\text{sw\_ft}}^T\end{bmatrix}^T \\ e_1 &= \begin{bmatrix}\beta_{11}\lambda_{11} e_{\text{l\_wr}}^T & \beta_{12}\lambda_{12} e_{\text{r\_wr}}^T & \cdots \\ & \beta_{13}\lambda_{13} e_{\text{CoM}}^T & \beta_{14}\lambda_{14} e_{\text{sw\_ft}}^T\end{bmatrix}^T \end{aligned} \tag{6}$$

where the pedices `l_wr` and `r_wr` represent the left and right wrist, `sw_ft` represents the swing foot. The Cartesian errors $e_{\text{l\_wr}}$, $e_{\text{r\_wr}}$, $e_{\text{sw\_ft}}$ are in general defined in terms of a position and orientation error as

$$e_{\text{cart}} = \begin{bmatrix} e_p^T & \lambda_o e_o^T \end{bmatrix}^T$$

where the weight $\lambda_o$ normalizes orientation and position errors. For the first and the second task, we set also a joint limit and a velocity limit such that

$$\begin{aligned} \mathrm{u}_{\text{j\_lims}} &= (q_{\max} - q)\,\Delta t & (7)\\ \mathrm{l}_{\text{j\_lims}} &= (q_{\min} - q)\,\Delta t & (8)\\ \mathrm{u}_{\text{v\_lims}} &= \alpha_i \dot{q}_{\max}\Delta t & (9)\\ \mathrm{l}_{\text{v\_lims}} &= -\mathrm{u}_{\text{v\_lims}} & (10)\\ \mathrm{l}_1 &= \max\left(\mathrm{l}_{\text{j\_lims}}, \mathrm{l}_{\text{v\_lims}}\right) & (11)\\ \mathrm{u}_1 &= \min\left(\mathrm{u}_{\text{j\_lims}}, \mathrm{u}_{\text{v\_lims}}\right) & (12) \end{aligned}$$

with $\mathrm{u}_{\text{j\_lims}}$ and $\mathrm{l}_{\text{j\_lims}}$ the $\dot{q}$ where $\alpha_1 < \alpha_2$ scales the bounds in order to implement a simple velocity allocation scheme between the primary and secondary task.

The second task is a postural (joint-space) task which again follows the formulation of the augmented Jacobian

$$\begin{aligned} J_2 &= \begin{bmatrix}\beta_2 W_1 I_{\text{nJ}}^T & (1-\beta_2)\, I_{\text{nJ}}^T\end{bmatrix}^T \\ e_2 &= \left[\beta_2 W_1 \lambda_{11} (q - q_d)^T \quad + (1-\beta_2)\,\lambda_{12} \nabla_q C_{g(q)}^T\right]^T \end{aligned}$$

where $C_{g(q)}$ is a cost function defined in terms of the gravity torque vector $\tau_{g(q)}$ as

$$C_{g(q)} = \tau_{g(q)}^T \tau_{g(q)} \tag{13}$$

and $W_1$ is a weight matrix, which in the experiment is equal to the joint-space inertia matrix $M(q)$ This is equivalent as having two tasks where the second is in the null-space of the previous one, minimizing

$$\left\| W \left(I - \lambda_{11} (q - q_d)\right)\right\|_2 \tag{14}$$

which is equivalent to minimizing

$$= \left(I - \lambda_{21} (q - q_d)\right)^T W_2^2 \left(I - \lambda_{21} (q - q_d)\right), \tag{15}$$

and the minimization of the minimum effort cost function

$$\left\| I + \lambda_{22} \nabla_q C_{g(q)} \right\|_2. \tag{16}$$

where $\nabla_q C_{g(q)}$ is computed numerically by means of two-point estimation.

By using $J_{\text{aug}}$ we can control the end-effectors and the Center of Mass of our humanoid robot, considering also shared kinematic chains (for instance, the torso for the arms). Notice that the second QP problem is needed to stabilize auto-motions due to redundancy. Therefore is a solution in the null-space of the first QP problem that does not compromise the previous solution. In the last QP we set the postural tasks as well as minimum effort tasks.

The weights used in the experiment are and

| Weight | Value | meaning |
|---|---|---|
| $\lambda_{11}$ | 1.0 | convergence speed for left wrist cartesian error |
| $\lambda_{12}$ | 1.0 | convergence speed for right wrist cartesian error |
| $\lambda_{13}$ | 1.0 | convergence speed for CoM position error |
| $\lambda_{14}$ | 1.0 | convergence speed for swing foot cartesian error |
| $\lambda_{21}$ | 1.0 | convergence speed for postural joint error |
| $\lambda_{22}$ | 1.0 | step size along minimum effort gradient |
| $\alpha_1$ | 0.5 | velocity allocation weight |
| $W_2$ | $M(q)$ | postural weight matrix |
| $\beta_{11}$ | 1.0 | relative importance of left wrist cartesian error in first stack |
| $\beta_{12}$ | 1.0 | relative importance of right wrist cartesian error in first stack |
| $\beta_{13}$ | 1.0 | relative importance of CoM position error in first stack |
| $\beta_{14}$ | 1.0 | relative importance of swing foot position error in first stack |

TABLE I. WEIGHTS USED IN THE STACK OF TASKS

the optimization algorithm uses the ”reliable” default settings of qpOASES, with regularisation enabled `enableRegularisation=BT_TRUE` with regularisation $\epsilon$ set with respect to the default one as `epsRegularisation=2E2`, the maximum number of working set calculations `nWSR=32`.

### *Experimental setup*

We impose a desired position for the postural task $q_d$ and for the end effectors $x_{d_{\text{l\_wrist}}}$, $x_{d_{\text{r\_wrist}}}$, $x_{d_{\text{swing\_foot}}}$, $x_{d_{\text{CoM}}}$ as the joint configuration, end effector position and CoM pose at the moment at the instant $t_0$ in which the control is started. We then tune the gain $\beta$ online in order to switch from a pure postural task to a mix of postural and minimum effort for the second task

## REFERENCES


[1] Alin Albu-Schäffer, Max Fischer, Günter Schreiber, Florian Schoeppe, and Gerd Hirzinger. Soft robotics: what cartesian stiffness can obtain with passively compliant, uncoupled joints? In *IROS*, pages 3295–3301. IEEE, 2004.

[2] Pasquale Chiacchio, Stefano Chiaverini, Lorenzo Sciavicco, and Bruno Siciliano. Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy. *The International Journal of Robotics Research*, 10(4):410–425, 1991.

[3] Agostino De Santis, Giuseppe Di Gironimo, Luigi Pelliccia, Bruno Siciliano, and Andrea Tarallo. Multiple-point kinematic control of a humanoid robot. In *Advances in Robot Kinematics: Motion in Man and Machine*, pages 157–168. Springer, 2010.

[4] Adrien Escande, Nicolas Mansard, Pierre-Brice Wieber, et al. Hierarchical quadratic programming. 2012.

[5] Hans Joachim Ferreau, Hans Georg Bock, and Moritz Diehl. An online active set strategy to overcome the limitations of explicit mpc. *International Journal of Robust and Nonlinear Control*, 18(8):816–830, 2008.

[6] Hans Joachim Ferreau, Christian Kirches, Andreas Potschka, Hans Georg Bock, and Moritz Diehl. qpoases: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, pages 1–37, 2013.

[7] Oussama Kanoun, Florent Lamiraux, and P-B Wieber. Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task. *Robotics, IEEE Transactions on*, 27(4):785–792, 2011.

[8] Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE J. Rob. Autom.*, 3(1):43–53, 1987.

[9] Christian Ott. *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*. Springer Publishing Company, Incorporated, 1 edition, 2008.

[10] J. K. Salisbury. Active stiffness control of a manipulator in Cartesian coordinates. In *Proceedings of the 19th IEEE Conference on Decision and Control*, 1980. Reprinted in: A tutorial on Robotics 2nd Edition 1986 IEEE.

[11] Lorenzo Sciavicco and Bruno Siciliano. A solution algorithm to the inverse kinematic problem for redundant manipulators. *Robotics and Automation, IEEE Journal of*, 4(4):403–410, 1988.

[12] Bruno Siciliano. Kinematic control of redundant robot manipulators: A tutorial. *Journal of Intelligent and Robotic Systems*, 3(3):201–212, 1990.

[13] Nikos G. Tsagarakis, G. M. Cerda, Zhibin Li, and Darwin G. Caldwell. Compliant humanoid coman: Optimal joint stiffness tuning for modal frequency control. In *ICRA*, pages 665–670, 2013.

[14] Patrick M Wensing and David E Orin. Generation of dynamic humanoid behaviors through task-space control with conic optimization. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3103–3109. IEEE, 2013.