# Learning robust task priorities of optimization-based whole-body torque-controllers

Marie Charbonneau[1,2], Valerio Modugno[2,3], Francesco Nori[4], Giuseppe Oriolo[3], Daniele Pucci[1] and Serena Ivaldi[2]

***Abstract*— The ability for a humanoid robot to safely evolve within a human environment is currently an important topic of research. Generating robust whole-body movements is still an open challenge, especially in contexts where a robot may physically interact with people and objects. Generating complex whole-body movements for humanoid robots is now most often achieved with the use of multi-task whole-body controllers based on optimization or quadratic programming. To perform on a real robot, however, such controllers often require a human expert to tune or optimize the many parameters of the controller related to the tasks and to the specific robot. This problem can be tackled by automatically optimizing some parameters such as task priorities or task trajectories, while ensuring constraints satisfaction, through simulation. This approach however does not guarantee that the optimized parameters in simulation will be optimal also for the real robot. As a solution to help bridge this reality gap, the present paper focuses on optimizing task priorities in a robust way by looking for solutions which achieve desired tasks under a variety of conditions and perturbations. This approach, which can be referred to as domain randomization, can greatly facilitate the transfer of optimized solutions from simulation to a real robot. The proposed method is demonstrated using the humanoid robot iCub for a whole-body stepping task.**

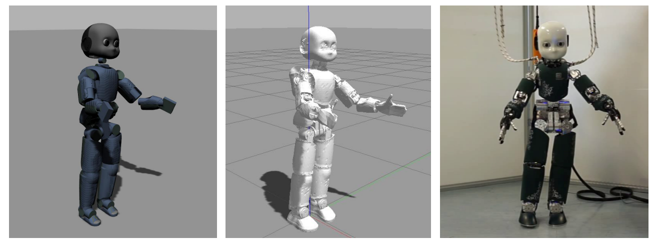

Fig. 1: Different robot models performing a whole-body motion with several tasks. We optimize task priorities for robustness, with the purpose to allow their transfer from the first model to the second, and eventually to the real robot.

## I. Introduction

Applications involving humanoid robots have the potential to bring significant benefits to society. Nevertheless, the design of controllers for humanoid platforms is highly challenging, especially when robots are expected to physically interact with people or the environment.

A promising approach is to use whole-body torque-control methods, which decompose a desired complex behavior into several simple tasks, typically framed as a *stack-of-tasks* [1]. Such a framework requires the tasks to be hierarchized, either in a *strict* or a *soft* way. In strict prioritization strategies, a fixed task hierarchy is ensured by geometrical conditions (e.g. null space task projector) or by the use of optimization constraints [2], [3]. Conversely, soft task prioritization can be achieved by assigning each task a weight defining its relative importance [4]. However, in the case of complex problems, such as whole-body motion of a humanoid robot, the design and proper tuning of task priorities may not always be evident, making it tedious and time consuming.

A recent line of research seeks to tackle the issue of automatically learning whole-body task priorities [5], [6]. Since learning algorithms need a considerable number of iterations and use a random exploration which could harm hardware, they are usually applied in simulation. However, inherent differences between simulated and real robots can render an optimal solution untransferrable from one to the other. Closing this reality gap is the central focus of recent works in robotics [7] and related fields. One approach, Domain Randomization (DR) [8], consists in randomizing some aspects of the simulation to enrich the range of possible environments experienced by the learner. For example in [9], control policies are learned in simulation, given random friction and control delays, and results showed that the learned policies were also effective on the real robot. As a result, it appears that looking for solutions which are *robust*, in opposition to *optimal*, may allow to bridge the reality gap.

This work proposes a method to learn robust task priorities which achieve compliant and stable whole-body motions, while allowing to facilitate the transfer of results from simulation to reality by taking advantage of the DR approach. The effectiveness of the proposed method is demonstrated by optimizing parameters in simulation, and showing that it is possible to overcome issues stemming from large differences between the learning domain and the testing domain.

## II. Methods

The method proposed for learning robust task priorities relies on two main parts: (i) an optimization-based whole-body torque-controller which tracks desired task trajectories

This project has received funding from the European Unions Horizon 2020 framework programme for research and innovation under the Marie Sklodowska-Curie Grant Agreement No.642667 (SECURE) and the ICT Grant Agreement No. 731540 (An.Dy).

[1]iCub Facility, Istituto Italiano di Tecnologia, Genova, Italy. name.surname@iit.it

[2]Inria Nancy - Grand Est, Team LARSEN Villers-lès-Nancy, France. name.surname@inria.fr

[3] Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Universita di Roma, Roma, Italy surname@diag.uniroma1.it

[4]Google DeepMind, London, UK. fnori@google.com

and sends joint torque commands to the robot, and (ii) an optimization method as described in [10], which poses no restrictions on the structure of the learning problem. Task priorities are then optimized at the end of an experiment (i.e. execution of a footstep): the fitness of the obtained trajectories is evaluated, allowing to update the task weights.

The controller assumes the modelling of the robot as described in [3], and the control input $u$ to be composed of joint torques $\tau$ and contact forces $F_C$. A stack of tasks is defined with the objectives to stabilize the center of mass position $X_{CoM}$, stance and swing feet pose $X_{stance}$ and $X_{swing}$, neck orientation $X_{neck}$, joint positions $s$, as well as to minimize joint torques $\tau$. The torque-controller used in this paper was developed in previous works, and described in [11]. Here, the controller is used with the following optimization problem using soft task priorities:

$$u^* = \arg\min_u \ \frac{1}{2} \text{ cost} \tag{1a}$$

$$\text{subject to } Cu \leq b \tag{1b}$$

where the constraint (1b) ensures that the contact forces remain within the associated friction cones. The cost function (1a) is computed as the weighted sum of all task objectives:

$$\text{cost} = \sum_T w_T \left|\ddot{\tilde{X}}_T(u)\right|^2 + w_s \left|\ddot{\tilde{s}}(u)\right|^2 + w_\tau \left|\tau(u)\right|^2 \tag{2}$$

$\ddot{\tilde{X}}_T(u)$ and $w_T$ are acceleration errors and weights associated to each Cartesian task $T$ (CoM, stance, swing and neck), while $w_s, w_\tau$ are the weights of the postural task and joint torque regularization.

## III. Experiments

A series of experiments were performed in order to validate empirically the hypothesis that the method described above is capable of optimizing task priorities, in such a way as to (i) allow the generation of robust whole-body motions, even when contacts due to physical interaction with the environment evolve in time and (ii) be able cope with imperfections in the robot model, disturbances, and noise.

Experiments were conducted in simulation using the open-source robot simulator Gazebo. They were performed with the iCub robot, using 23 DOF on legs, arms and torso, for whole-body torque control. The design of iCub has evolved over the years, which has a significant impact on the inertial properties of the robots. For instance, some models of iCub have tethered power supply, while others have battery packs installed on the back of the torso. This gives us a chance to test our method on different robot models.

The controller described in II was developed in Matlab/Simulink, allowing to control the motion of either a simulated or a real robot. It is applied here to the problem of performing a step, i.e. lifting the foot off the ground and placing it back on the ground.

The experimental procedure was divided into two main parts: (i) training task priorities with a first model of iCub, and (ii) validating the obtained task priorities with a different model of iCub.

*1) Training with a first iCub model:* First, task priorities were optimized on a simulated tethered iCub model, as shown in the left part of fig. 1, performing a whole-body movement (one step). The fitness function $\phi_{pr}$ was evaluated in 10 separate learning experiments, in order to optimize task priorities.

$$\phi_{pr} = \frac{1}{2}(\phi_p + \phi_r) \tag{3a}$$

$$\phi_p = -\frac{1}{P_{ZMP_{max}}} \sum_{t=0}^{t_{end}} \left|P_{ZMP} - O_{SP}\right|^2 \tag{3b}$$

$$\phi_r = -\frac{1}{X_{T_{max}}} \sum_{t=0}^{t_{end}} \sum_T \left|\tilde{X}_T\right|^2 - \frac{0.0001}{\tau_{max}} \sum_{t=0}^{t_{end}} |\tau|^2 \tag{3c}$$

This particular fitness function, $\phi_{pr}$, favors robust solutions with $\phi_p$ by encouraging smaller excursions of the ZMP position $P_{ZMP}$ with respect to the center of the support polygon $O_{SP}$. On the other hand, the term $\phi_r$ seeks to maximize performance on the Cartesian tasks with a minimal effort. In these equations, $X_{T_{max}}$, $\tau_{max}$ and $P_{ZMP_{max}}$ are normalization factors. In case the robot was unable to accomplish a full step, a penalty of $-1.5$ is added to $\phi_{pr}$.

In addition, the robot was subjected to random sets of conditions during training, in order to achieve robustness through DR. For each learning iteration, the following conditions were randomized: Gaussian noise on input F/T sensor signals, swing foot, motion of the swing foot, displacement of the CoM, and a random number of random external wrenches applied to the chest. The external wrenches not only served to increase the robustness of the controller, but also to promote the soft behavior of the robot in case of physical interaction with people, while still keeping balance.

Having been verified to allow the first iCub model to successfully perform the desired stepping motion, the following hand-tuned task priorities were used as a starting point for the optmization:

$$w_{CoM} = 1 \tag{4a}$$

$$w_{stance} = 1 \tag{4b}$$

$$w_{swing} = 1 \tag{4c}$$

$$w_{neck} = 0.1 \tag{4d}$$

$$w_s = 0.001 \tag{4e}$$

$$w_\tau = 0.0001 \tag{4f}$$

Then, optimized task priorities were obtained by performing 200 learning iterations with applied to the control framework, with an exploration rate of $0.1$. The optimization procedure was repeated for 10 separate trainings, allowing to verify the consistency of the method.

*2) Testing with a second iCub model:* In order to validate the robustness achieved with the optimized task priorities, while attempting to replicate conditions similar to performing experiments on the real robot, each one of the resulting 10 sets of optimized task priorities was tested on an iCub model with a battery pack on the back, as shown in the middle part of fig. 1. The robot was made to perform a sequence

TABLE I: Optimized task priorities: mean and standard deviation obtained from 10 different training experiments

| weight | mean | std deviation |
|---|---|---|
| $w_{CoM}$ | 1 | 0 |
| $w_{stance}$ | 0.9 | 1.3 |
| $w_{swing}$ | 2.4 | 1.1 |
| $w_{neck}$ | 0.6 | 1.2 |
| $w_s$ | 1e−6 | 0 |
| $w_\tau$ | 1e−10 | 0 |

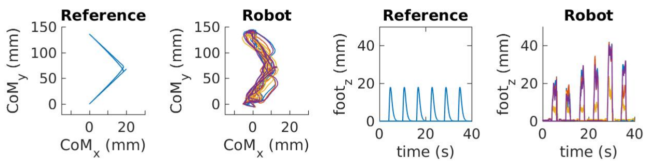


Fig. 2: Typical CoM and feet trajectories for 6 strides performed with the second iCub model. Each color denotes the use of a different set of optimized weights. The $x$, $y$ and $z$ axes correspond to the sagittal, frontal and vertical axes.

of whole-body movements (6 steps), under different noise conditions as those used for training. It was subjected to external wrenches on the chest, as well as Gaussian noise on the F/T sensor and joint velocity measurements.

## IV. Results

The mean and standard deviation of the optimized task priorities, as obtained with the experiments explained above, are shown in table I.

These task priorities, when used with the controller described in Sec. II, allowed the first robot to perform one step, under the conditions used for training. They also successfully allowed the second robot model to perform 6 steps, under the noise conditions mentioned previously, with a success rate of 100%. In comparison, the starting task weights defined in 4 did not prove to be successful, showing that the optimized weights did improve the effectiveness of the controller.

The CoM and feet trajectories achieved with the optimized task priorities on the second robot model, illustrated in Fig. 2, show convergence of the robot motion. These results demonstrate that the optimized weights allow for a higher robustness of the controller.

## V. Discussion and Conclusions

In summary, the proposed method can be used to generate robust task priorities for whole-body torque-control of humanoids. It was demonstrated by performing training on a first robot, then testing on a second model with different physical properties and working conditions.

A fitness function combining robustness and performance has shown to allow the obtention of sensible task priorities. In the achieved results, swing foot placement, crucial for stability at touchdown, is given high importance, while the neck orientation task a lesser one, allowing compliance to external perturbations (i.e. physical interactions with the environment, such as the impact of the foot on the ground). As for the postural task, its low priority allows it to be used as regularization (just as joint torques), instead of competing with Cartesian tasks.

Such a solution is interesting, as it may not have been *a priori* self-evident to an expert defining task priorities. Furthermore, the ranges over which sets of optimized weights were obtained show that although task priorities require proper tuning, the controller is not highly sensitive to a precise adjustment of task weights.

Finally, the proposed method has shown to achieve compliant and stable behaviors with a robot model different than the one used for learning, and subjected to diverse working conditions. The robustness achieved in this way is promising and could allow higher success when passing from simulation to real-world experiments. Upcoming work shall provide a more extensive analysis of the method, comparing results obtained with different fitness functions, as well as with and without domain randomization, in order to asses the contribution of fitness parameters and DR to the success of the method. Our approach shall also be tested with experiments on the real robot.

## References


[1] M. A. Hopkins, D. W. Hong, and A. Leonessa, "Compliant locomotion using whole-body control and divergent component of motion tracking," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 5726–5733.

[2] L. Saab, O. E. Ramos, F. Keith, N. Mansard, P. Souères, and J. Y. Fourquet, "Dynamic whole-body motion generation under rigid contacts and other unilateral constraints," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 346–362, April 2013.

[3] G. Nava, F. Romano, F. Nori, and D. Pucci, "Stability analysis and design of momentum-based controllers for humanoid robots," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 680–687.

[4] J. Salini, V. Padois, and P. Bidaud, "Synthesis of complex humanoid whole-body behavior: A focus on sequencing and tasks transitions," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 1283–1290.

[5] N. Dehio, R. F. Reinhart, and J. J. Steil, "Multiple task optimization with a mixture of controllers for motion generation," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2015, pp. 6416–6421.

[6] S. Ha and C. Liu, "Evolutionary optimization for parameterized whole-body dynamic motor skills," in *ICRA*, 2016.

[7] D. Clever, M. Harant, K. D. Mombaur, M. Naveau, O. Stasse, and D. Endres, "Cocomopl: A novel approach for humanoid walking generation combining optimal control, movement primitives and learning and its transfer to the real robot HRP-2," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 977–984, 2017.

[8] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2017, pp. 23–30.

[9] R. Antonova, S. Cruciani, C. Smith, and D. Kragic, "Reinforcement learning for pivoting task," *CoRR*, vol. abs/1703.00472, 2017.

[10] V. Modugno, G. Nava, D. Pucci, F. Nori, G. Oriolo, and S. Ivaldi, "Safe trajectory optimization for whole-body motion of humanoids," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, November 2017, pp. 763–770.

[11] S. Dafarra, G. Nava, M. Charbonneau, N. Guedelha, F. Andrade, S. Traversaro, L. Fiorio, F. Romano, F. Nori, G. Metta, and D. Pucci, "An online predictive kinematic planner for position and torque controlled walking of humanoid robots," 2018, submitted for publication.