Parameter tuning as a challenge in the development of optimization-based whole-body torque-controllers*

Marie Charbonneau[1,2], Serena Ivaldi[2], Jean-Baptiste Mouret[2], Francesco Nori[3] and Daniele Pucci[1]

I. INTRODUCTION

The field of humanoid robotics has seen impressive developments recently. However, it remains a challenge for a robot to function autonomously in the real world and safely interact with people and environment. At the same time, this ability becomes crucial for envisioned applications such as service robots for care and companionship. Physical interactions with objects and people significantly influence the behavior of a robot, in particular its balance and stability. Therefore, there is a need to investigate methods that allow the coordination of whole-body motion tasks with external contacts, whether planned or unexpected.

When compliance and physical interactions (e.g. safe human-robot physical interaction) matter, torque control, in opposition to position control, is usually preferred. A reactive torque-based whole-body controller, typically using quadratic programming (QP) optimization, can be developed for this purpose. It allows for motion control, while ensuring not only that the robot keeps balance when external forces are applied, but also that it softly yields under forces, allowing for safer physical interaction. Such a scheme has shown to produce impressive results, for example with the famous iCub yoga demo [1].

II. APPROACH

The scope of our study is to design algorithms and methods for a torque-based whole-body controller, with the objective to enable a humanoid robot to safely move around a human environment. In this, the alliance of stack of tasks approaches and QP solvers have shown to be effective tools, simplifying the development of a controller.

Our approach, described in [2], [3], can be summarized as follows. The robot is modeled as a floating-base manipulator, with acceleration constraints at the contact of the feet with the ground. A stack of tasks is defined with the center of mass position, feet pose, torso orientation and joint positions, allowing to track their acceleration. Reference accelerations, for their part, are computed using a proportional-derivative (PD) control strategy, given desired task positions. The controller then relies on quadratic programming optimization to compute the joint torques and contact forces required for achieving the desired motions. The QP is formulated with soft tasks, minimizing a weighted sum of task errors and a regularization term on the joint torques, given constraints on the feet/ground contact and support polygon.

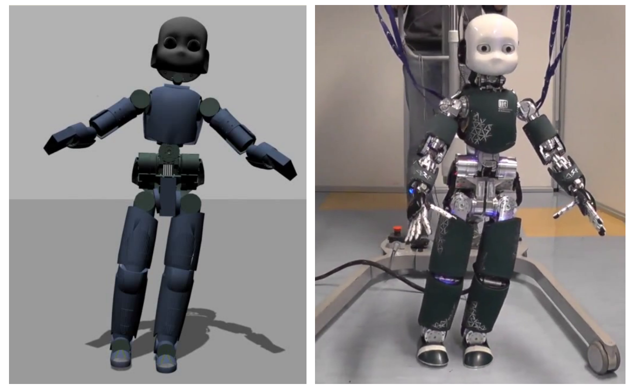

Fig. 1: Motion achieved with the whole-body torque-controller: lifting the right foot in simulation (left) and with the robot (right).

In [2], a validation scenario in which the robot walks in place was defined with a state machine determining task trajectories; in [3], model predictive control was used to provide trajectories for walking. Apart from the desired trajectory of each task, variables of the controller included the weights associated to each task, as well as proportional-derivative gains used for computing reference accelerations (for each state of the state machine).

III. RESULTS

In [2], simulation results showed the robot could follow trajectories with enough precision to be able to walk in place practically indefinitely. When running experiments on the robot however, although the robot succeeded in performing a walking in place motion, it was noticed that trajectories were followed with increasing errors, and the balance of the robot was not always ensured.

Similarly in [3], the robot could walk in simulation experiments, but during experiments on the robot, feet positioning errors had to be coped with and the walking speed was slowed down.

The discrepancy between simulation and real world experiment results was reported to be mostly related to the estimation of joint torques and floating base, which contributed to limiting the performance of the torque controller.

*This work was supported the FP7 European project CoDyCo under Grant 600716 ICT 2011.2.1 Cognitive Systems and Robotics and by the Innovative Training Network SECURE, funded by the Horizon 2020 Marie Skłodowska Curie Actions (MCSA) of the European Commission (H2020-MSCA-ITN-2014- 642667)

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

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

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

IV. DISCUSSION

It is known that achieving precise motions with torque control can be more challenging than with position control for example. By extension, so is the problem of achieving whole-body motions which are robust and compliant to interaction with the environment. Moreover, with a stack-of-tasks/QP-based controller, a certain number of parameters are needed to define desired trajectories, controller gains, as well as weights of tasks. In other words, the number of tunable parameters can rapidly increase with the number of tasks. As a result, deployment on a real robot is not ensured to be straightforward. Indeed, when passing from a working simulation to a robot, the success or failure of an experiment may highly depend on proper tuning of parameters, and not only on precise estimation of the state of the robot.

Notably, the controller of the yoga++ demo [1], [4] was also based on a stack-of-tasks approach with QP solver, and tuning was a significant part of its success. It is thus possible that given more time, more tuning, the controllers described above would have been improved to achieve more impressive results.

Nonetheless, a number which generally does not appear on paper is the time spent on tuning parameters of the controller. Indeed, it shows to be a significantly non-trivial task on its own, contributing to making the passage from simulation to experiments on the robot a challenge. From anecdotal evidence, it does not seem so rare to find that, in the development of a new experiment with whole-body control, tuning of parameters took several months and was done by hand. A tuning which, additionally, may need to be corrected for any change in conditions. As a result, parameter tuning can be considered a straining and thankless task, if one does not have the intuition for it. And when parameter tuning becomes tedious, effective tools would come in handy.

V. CONCLUSIONS AND FUTURE WORK

Whole-body control methods combining stack of tasks approaches and QP solvers can yield very interesting results, allowing for balance and motion control, as well as safe physical interaction between people and robot. However, it appears to be a general matter that they rely on the proper tuning of a non negligible number of parameters.

Therefore, it seems worth opening a discussion on the subject of parameter tuning, and investigating solutions to alleviate the difficulty of this task. At the moment, only few papers have been published on the subject of parameter tuning for a humanoid robot [5], [6], [7]. Notably, a tool for automatic gain tuning of a whole-body torque-controller has been investigated in [5], but the solution called for relatively large computations, which may not always be affordable for real-time applications.

Further solutions to ease tuning need to be investigated, which is the focus of our ongoing work. Taking inspiration from [6], we are currently developing a reinforcement learning method for tuning the weights w associated to task priorities, in order to achieve robust whole-body motions. The optimization process can be summarily formulated as

$$w^* = \arg\max_{w} \ \text{fitness}(w) \tag{1a}$$

$$\text{subject to QP success} \tag{1b}$$

where a learning algorithm (e.g. bayesian optimization or (1+1)-CMA-ES) is used to optimize the weights, the fitness is computed from results of an experiment given a set of weights provided by the learning algorithm, and a constraint is added on the successful solution of the QP over the experiment.

Consequently, future works will be investigating methods to make the deployment of whole-body controllers easier, aiming to define methods general enough to be applied to a wide range of optimization-based controllers.

REFERENCES

[1] D. Pucci, F. Romano, S. Traversaro, and F. Nori, "Highly dynamic balancing via force control," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, Nov 2016, pp. 141–141.

[2] M. Charbonneau, G. Nava, F. Nori, and D. Pucci, "An optimization based control framework for balancing and walking: Implementation on the icub robot," 2017, manuscript submitted for publication. [Online]. Available: https://arxiv.org/abs/1707.08359

[3] S. D. 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, manuscript submitted for publication.

[4] 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.

[5] D. Pucci, G. Nava, and F. Nori, "Automatic gain tuning of a momentum based balancing controller for humanoid robots," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, Nov 2016, pp. 158–164.

[6] V. Modugno, G. Neumann, E. Rueckert, G. Oriolo, J. Peters, and S. Ivaldi, "Learning soft task priorities for control of redundant robots," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 221–226.

[7] 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.