Multi-Contact Interaction with Hierarchical Inverse Dynamics and Momentum Trajectory Generation

Alexander Herzog*, Nicholas Rotella+, Sean Mason+, Felix Grimminger*,
Stefan Schaal*+ and Ludovic Righetti*
alexander.herzog@tuebingen.mpg.de
* Autonomous Motion Dept., Max-Planck Intelligent Systems, Germany
+ CLMC Lab, University of Southern California, Los Angeles, USA

1 Introduction

Legged robots are expected to locomote autonomously in an uncertain and potentially dynamically changing environment. Active interaction with contacts becomes inevitable to move and apply forces in a goal directed way and withstand unpredicted changes in the environment. Therefore, we need to design algorithms that exploit interaction forces and generate desired motions of the robot leading to robust and compliant interaction with the environment. In this context, the choice of a control strategy for legged robots is of primary importance as it can drastically improve performance in face of unexpected disturbances and therefore open the way for agile robots, whether they are locomoting or performing manipulation tasks.

Interactions between a robot and an uncertain environment (e.g. stepping on unsteady ground) require compliant control, which can be achieved well with inverse dynamics on torque controlled robots. Extensions to hierarchical inverse dynamics allow to construct more complex behaviors consisting of force and motion tasks satisfying physical constraints and actuator limitations. This class of algorithms has been shown to create physical motions in simulation [1] where perfect knowledge about the robot model and state is given and real-time requirements can be relaxed. Resulting joint trajectories were replayed with joint position control on a real robot [6]. However, the constraint-aware control capabilities of the inverse dynamics controller are not exploited. In face of unpredicted events like a push or unsteady ground, constraints, such as support limits, may be violated and make the robot tip over. Even with satisfied constraints, joint position control results in very different response to disturbances compared for instance with a hierarchical inverse dynamics controller that runs online. Where the former performs feedback on every single joint independently, the hierarchical inverse dynamics controller realizes several task feedback-loops and is able to absorb a push e.g. at a stance leg, while a swing leg motion remains unaffected as we will show in our experiments.

A prerequisite for such behavior is the realization of hierarchical inverse dynamics controllers that take into account inequality constraints in a fast control cycle. Although, computationally demanding, we implemented a variant of cascades of QPs to run sufficiently fast and solve a hierarchy of tasks in a tight 1kHz control-loop on a torque controlled humanoid [3] leading to constraint-aware multi-task control. In our previous work [3] we used cascades of QPs to make our Sarcos humanoid (see Fig 1) balance. In this work, we simplify the balance control design leading to more robust and dynamical balancing behaviors as we verify in additional experiments.

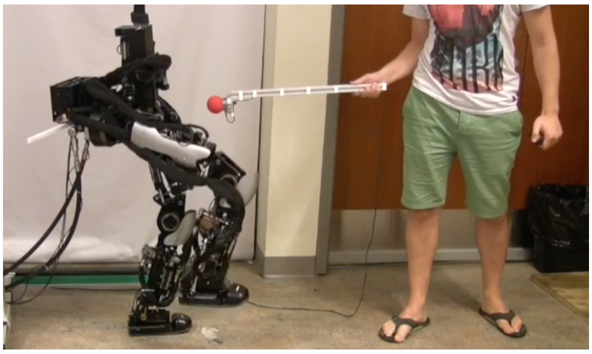

Figure 1: The sarcos humanoid in single support. It swings one leg while being poked with impulsive strong pushes.

Further, we discuss a trajectory generation approach extending this local-time approach with preview capabilities.

2 Momentum Control Design

A robust balancing behavior is achieved with control of the momentum dynamics

$$\begin{aligned} M\dot{\mathbf{r}} &= \mathbf{l} & (1)\\ \dot{\mathbf{l}} &= M\mathbf{g} + \sum \mathbf{f}_i & (2)\\ \dot{\boldsymbol{\kappa}} &= \sum \boldsymbol{\tau}_i + \sum (\mathbf{p}_i - \mathbf{r}) \times \mathbf{f}_i, & (3)\end{aligned}$$

where M is the mass of the robot, $\mathbf{r}$ is the Center of Mass (CoM) and $\mathbf{l}, \boldsymbol{\kappa}$ the overall linear and angular momenta. A change of momentum is generated through contact forces $\mathbf{f}_i$ and torques $\boldsymbol{\tau}_i$ acting at the endeffectors $\mathbf{p}_i$. Assuming that $\mathbf{p}_i - \mathbf{r}$ remains constant over one control cycle, a desired closed-loop behavior on the change of momentum is typically written [5] in form of a spring-damper

$$\dot{\mathbf{h}}_{ref} := \begin{bmatrix} \dot{\mathbf{l}}_{ref} \\ \dot{\boldsymbol{\kappa}}_{ref} \end{bmatrix} = \mathbf{P} \begin{bmatrix} m(\mathbf{x}_{ref} - \mathbf{x}_{cog}) \\ \mathbf{0} \end{bmatrix} + \mathbf{D}(\mathbf{h}_{ref} - \mathbf{h}) + \dot{\mathbf{h}}_{ref} \quad (4)$$

There are, however, several issues with such an approach. First, the tuning of the PD controller can be problematic. In our experience, on the real robot it is necessary to have different gains for different contact configurations to ensure proper tracking which leads to a time consuming tuning process with many open parameters. Second, such a controller does not exploit the coupling between linear and angular momentum rate of change through interaction forces that is expressed in Eq (3). We propose to use the model of Eqs (1), (2), (3) to compute optimal feedback gains. We linearize the dynamics and compute a LQR controller by selecting a desired performance cost. We find a control law of the form

$$\begin{bmatrix} \mathbf{f} \\ \boldsymbol{\tau} \end{bmatrix} = -\mathbf{K} \begin{bmatrix} \mathbf{x}_{cog} \\ \mathbf{h} \end{bmatrix} + \mathbf{k}(\mathbf{x}_{ref}, \mathbf{h}_{ref}) \quad (5)$$

that contains both feedback and feedforward terms. A desired closed-loop behavior for the momentum that appropriately takes into account the momentum couplings is then computed. In our experiments we demonstrate that such a design leads to better performance than a naive PD control design. Moreover, gains are computed automatically for any contact configuration, significantly simplifying the implementation of the controller.

3 Experiments

We implement the momentum control task in a QP cascade variant [4] and run the balance controller in a 1kHz control loop on the lower part of our Sarcos humanoid[1] (see Fig 1). Physical constraints such as consistency with the robot dynamics and contact force limitations are put in higher priority whereas posture control and force regularization is put in lower priority. Various kinds of disturbances were applied to the robot. It was put on a rolling platform and tilting platform and balanced out rapid displacement of the support as can be seen in the video. We excerpt a sequence of systematic pushes of up to 9Ns at different points on the robot structure and compare the disturbance rejection of the two control designs in Eqs (4), (5). For both momentum control tasks, the robot was able to withstand impacts with high peak forces and strong impulses without falling. For every push, the change in momentum was damped out quickly and the CoG was tracked after an initial disturbance. With the LQR gains we see a significant improvement in recovering the CoG even though the robot was pushed harder than with the controller using diagonal gain matrices.

In an additional experiment, we made the robot move on one foot, while it was performing a swing motion with the other leg. The swing leg task was put in the same priority as the momentum control task. Not only was the robot able to track the swing leg motion, but it was also able to balance out strong shocks of peak forces up to 290N. Fig. 2 shows the swing leg tracking and momentum of the robot at the moment of impact. Here, we can see the advantage of realizing several task feedback loops simultaneously. Due to the momentum control the injected disturbance momentum remains bounded and is damped out eventually while at the same time tracking of the swing leg is not visibly affected. The reaction forces and CoPs stay inside their bounds at all times and are never traded-off due to the higher priority.

4 Trajectory Generation

Our experiments show that the proposed whole-body control approach is able to generate desired closed loop behaviors sufficiently robust to perform balancing in single and double support. In our tasks we used simple design techniques to obtain desired trajectories for the (linear and angular) momentum and swing leg motions. If we want to perform more complex tasks, such as walking, we need to generate momentum trajectories that are consistent with the interaction forces and lead to stable gaits. Although, there exists an established set of tools to generate CoM trajectories under the LIPM assumption (Linear Inverted Pendulum Model) [2], it remains unclear how physically valid momentum trajectories can be generated once the LIPM assumption does not hold anymore. For instance, if the robot is to step on footholds that are significantly above the ground, it might be restricting to assume that the CoM remains at constant height. If in addition, footholds are tilted relative to each other, a common ZMP is not even defined.

[1] A video summarizing the experiments can be found on http://www.youtube.com/watch?v=jMj3Uv2Q8Xg

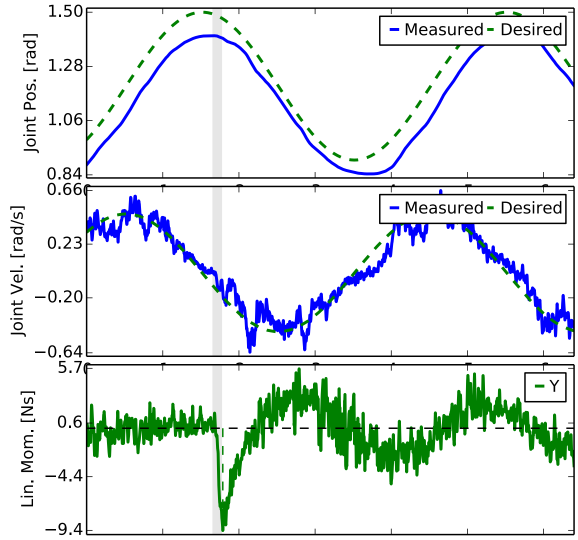

Figure 2: The robot state, when it was balancing in single support and performing a swing leg motion. The grey bar marks the moment of push that was applied at the hip. Position and velocity tracking (top two plots) of the swinging hip joint remains unaffected, while the momentum (3rd plot from top) is kept bounded and damped out eventually.

It is desirable to exploit both, the robust behavior of simple models but also generate sufficiently complex tasks, which require less restricting assumptions. In our current research we use the momentum dynamics in Eqs (1), (2), (3) to plan momentum trajectories. They are executed together with other tasks using the presented whole-body control approach. Without the restriction of a zero change of angular momentum or co-planar footholds, a broader class of tasks can be generated. In a preliminary simulation result the robot performs steps using automatically generated momentum trajectories that are realized with the presented whole-body control approach. We are now working on realizing more dynamical motions with the discussed approach. Although negligible in simulation, estimation of the momentum becomes a prerequisite for good performance on the real system [7].

References

[1] M. de Lasa, I. Mordatch, and A. Hertzmann. Feature-Based Locomotion Controllers. *ACM Transactions on Graphics*, 29(3), 2010.

[2] J. Englsberger, C. Ott, M. A. Roa, A. Albu-Schaffer, and G. Hirzinger. Bipedal walking control based on Capture Point dynamics. In *IROS*, pages 4420–4427, 2011.

[3] A. Herzog, L. Righetti, F. Grimminger, P. Pastor, and S. Schaal. Balancing experiments on a torque-controlled humanoid with hierarchical inverse dynamics. In *IROS*, 2014.

[4] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, and L. Righetti. Momentum Control with Hierarchical Inverse Dynamics on a Torque-Controlled Humanoid. *arXiv.org*, Oct. 2014.

[5] S.-H. Lee and A. Goswami. A momentum-based balance controller for humanoid robots on non-level and non-stationary ground. *Autonomous Robots*, 33:399–414, 2012.

[6] N. Mansard. A dedicated solver for fast operational-space inverse dynamics. In *ICRA*, 2012.

[7] N. Rotella, A. Herzog, S. Schaal, and L. Righetti. Momentum Estimation, Planning and Control for Force-Centric Bipedal Locomotion. In *Dynamic Walking (submission)*, 2015.