Experiments with Hierarchical Inverse Dynamics Controllers on a Torque Controlled Humanoid

Alexander Herzog\textsuperscript{1}, Ludovic Righetti\textsuperscript{1,2}, Felix Grimminger\textsuperscript{1} and Stefan Schaal\textsuperscript{1,2}
\textsuperscript{1} Autonomous Motion Dept., MPI Intelligent Systems, Tübingen, Germany
\textsuperscript{2} CLMC Lab, University of Southern California, Los Angeles, USA

1 Introduction

We expect autonomous legged robots to perform complex tasks in persistent interaction with an uncertain and changing environment (e.g. in a disaster relief scenario). Therefore, we need to design algorithms that can generate precise but compliant motions while optimizing the interactions with the environment. In this context, torque control algorithms often offer high performance for motion control while guaranteeing a certain level of compliance. In addition they allow for direct control of interaction forces with the environment. Recent contributions have demonstrated the relevance of torque control approaches for humanoid robots, for example for balancing capabilities [5, 6]. Among those we find passivity-based approaches [5] that regulate the position of the Center of Mass (CoM) by applying admissible contact forces under the quasi-static assumption. On the one hand, these approaches do not rely on a precise dynamic model of the robot while naturally guaranteeing robustness due to the passivity property of the controller. On the other hand the quasi-static assumption might be limiting for dynamic motions. A promising way of leveraging this limitation are control algorithms that take the full dynamic model into account [6]. However, the need for a precise dynamic model, sensor noise (particularly in the velocities) and limited torque bandwidth makes them more challenging to implement. Moreover, it is generally required to simplify the optimization process to meet time requirements of fast control loops (typically 1 kHz on modern torque controlled robots). Practical evaluations of both approaches are still rare due to the lack of torque controlled humanoid platforms and the complexity in conducting such robot experiments.

There have been recently very promising approaches for controlling hierarchies of tasks using the full dynamics of the robot [1, 4]. Their problem formulations are more general than approaches based on pseudo-inverses as they naturally allow the inclusion of arbitrary types of tasks including inequalities. The resulting optimization problems are phrased as cascades of quadratic programs to be solved. Evaluation of their applicability was done in simulation and it has been argued that these algorithms can be a) implemented fast enough and b) that they can work on robots with model-uncertainty, sensor noise and limited torque bandwidth. To the best of our knowledge, these controllers have never been used as feedback-controllers on real torque controlled humanoids. In this contribution [2] we evaluate the balancing performance of a humanoid robot running hierarchical inverse dynamics controllers phrased as cascades of QPs [1]. First we propose a simplification of the dynamic constraints that allow us to generally reduce the computational complexity of algorithms using inverse dynamics. It allows us to implement our controller in a 1 kHz real-time control loop. Our main focus is then on presenting various balancing experiments in order to demonstrate the applicability of the approach under real robot conditions, i.e. model uncertainty, sensor noise, state estimation errors and a limited control bandwidth. In one experiment, we implement a momentum-based balance controller [3] to take into account dynamic constraints. This experiment demonstrates the capabilities of such momentum-based balance controllers on a torque controlled robot. The second experiment is a tracking task, demonstrating that tracking accuracy in operational space can still be achieved. It is, to the best of our knowledge, the first demonstration of the applicability of the methods proposed in [1] as feedback controllers on torque controlled humanoids (i.e. without joint space PD control).

2 Problem Fromulation

For our experiments we write hierarchical inverse dynamics controllers and implement the solver proposed by [1] to perform real-time feedback control. In the following we give an example on how tasks can be formulated and revisit the original solver formulation. Finally, a simplification is proposed that is also applicable to other inverse dynamics approaches. Assuming rigid-body dynamics, we can write the equations of motion (EoM) of a robot as

\begin{equation}
M(q) \ddot{q} + N(q, \dot{q}) = \tau + J_c^T \lambda 
\end{equation}

\begin{equation}
M(q) \dot{q} + N(q, q) = J_{c}^T \lambda 
\end{equation}

where \( q \) denotes the generalized configuration of the robot, \( \tau \) is the commanded joint torques and \( \lambda \) are the generalized contact forces. \( M(q) \) is the inertia matrix, \( N(q, \dot{q}) \) is the vector of all forces (Coriolis, centrifugal, gravity, friction, etc...) and \( J_c \) is the Jacobian of the contact constraints. We separate the matrices into the actuated part consisting of the first \( n \) rows (indexed by \( a \)) and the unactuated part consisting of the last \( 6 \) rows (indexed by \( u \)), where \( n \) is the number of Degrees of Freedom (DoF) of the robot. We are interested in executing position and force controllers expressed as affine functions of the variables \( q, \lambda, \tau \) while satisfying consistency with the EoM (1), (2). E.g. we can express a desired closed-loop behavior of the momentum

\begin{equation}
\dot{h} = H_{G} \dot{q} + H_{C} \dot{q} = PD(x_{CoM}, h),
\end{equation}

where \( h = H_{G}(q) \dot{q} \) is the linear and angular momentum and the right-hand side of Eq (3) is a PD control on the momentum. As this is not always realizable together with the
EoM (1), (2) and possibly additional hardware constraints (e.g. torque saturation limits $\tau_{min} \leq \tau \leq \tau_{max}$), we express a cascade of quadratic programs that find admissible $q^*$, which satisfy Eq (3) as good as possible in a least-squares sense. In general, two subsequent cascades can be written as

$$\begin{align*}
\text{min.} & \quad \|v_1\|^2 + \|w_1\|^2 \\
\text{s.t.} & \quad A_1 y + a_1 \leq v_1, \\
& \quad B_1 y + b_1 = w_1, \\
\end{align*}$$

and substitute $v_1, w_1$ into Eq (1) for $\tau$ and substitute for all occurrences of $\tau$ inside of all QPs. This way we reduce computation time by 40%. This is crucial for us to run our controller in a 1 kHz feedback-control loop.

3 Experiments

We formulated balancing and motion tracking tasks in the framework discussed in the previous section and evaluated them on the lower part of the Sarcos humanoid as shown in Fig 1 and in the video summarizing the experiments\(^1\). Our goal was to evaluate such control formulations in face of model-uncertainty, sensor noise, imperfect estimation and a limited torque bandwidth. The performance of the robot was tested in three different scenarios: pushing experiments in single and double support and a tracking task. In our tasks we gave highest priority to the EoM and torque saturation constraints, followed by joint acceleration limits and dynamic constraints that emerge from the finite foot support. In the next lower priorities we wrote a controller on the momentum as in Eq (3) and expressed a motion task on the feet to either keep them stationary or move one foot up. Redundancy resolution on motion and reaction forces is achieved on the lowest priorities. We examined the balance performance of the robot in several scenarios as illustrated in Fig 1, where we applied disturbances of various kind. When standing on two legs, the humanoid was put on a balancing board and was able to keep standing despite rapid changes of the slope. It showed a compliant behavior and adapted the posture when we pushed it at several points on its structure. When we put the robot on a rolling platform and displaced it rapidly, the humanoid was able to keep balance and damped out the disturbance after a short time. In an additional experiment we constructed a more complicated task with contact switches, where we made the robot move on one leg and lift the unloaded leg. We pushed it with a stick that has a force sensor attached to acquire measurements of the push force. Again, the robot was able to balance out pushes comparably high relative to experiments in related work [5].

4 Conclusion

In this contribution we implemented hierarchical inverse dynamics controllers on the lower part of a torque controlled humanoid and evaluated them in several balancing and tracking experiments. We propose a simplification that is exploited to realize a 1 kHz feedback-control loop. From our experiments we conclude that hierarchical inverse dynamics controllers can perform well even under model-uncertainty, sensor noise, imperfect estimation and a limited torque bandwidth.

5 Outlook

The robust balancing capabilities of the Sarcos humanoid raised our interest in investigating the locomotion performance when the robot is controlled with the discussed algorithm. In simulation, we extended our single support balancing task to a gait allowing the robot to walk while it resists disturbances. Our focus is now to implement the same on the real robot.

References


\[7\] The video is available on http://youtu.be/RlVOaW4vPU8

Figure 1: Snapshots from the balancing experiments we conducted on the lower part of the Sarcos Humanoid.