Why off-the-shelf physics simulators fail in evaluating feedback controller performance - a case study for quadrupedal robots

Why off-the-shelf physics simulators fail in evaluating feedback controller performance - a case study for quadrupedal robots

M. Neunert, T. Boaventura and J. Buchli

Agile & Dexterous Robotics Lab, ETH Zürich, Switzerland
E-mail: {neunertm, tboaventura buchlij}@ethz.ch
www.adrl.ethz.ch

Simulators are essential to robotics research since they help us to understand physical properties of the robotic system as well as test control and motion planning approaches in a quick and safe manner. Yet most robotics simulators rely entirely on physics engines and do not model actuator dynamics, noise or delay. However, as we demonstrate in this work, modeling these effects is essential when evaluating the performance of closed loop controllers. In fact, neglecting them is one reason why controllers do not transfer well from simulation to hardware. While the presented study applies to most multi-link robots, our parameter choice is inspired by state-of-the-art quadrupedal robots.

Keywords: Simulation, Actuator Dynamics, Noise, Delay, Stability, Sensitivity

1. Introduction

Due to the complexity of modern robots, physics simulations have become an essential tool for robotics researchers. When looking at recent advances in simulators, great care is taken to accurately model the underlying physical behavior of rigid body dynamics.[1] Thus many simulators, both open source[2] and commercial[3,4] rely on mature, accurate physics engines such as ODE[5] or Bullet.[6] Yet, we often experience that controllers are often not directly transferable from simulation to hardware. One reason is that robots are not pure, ideal multi-body systems. Instead their behavior is affected by actuator dynamics, sensor and system noise, delays, disturbances and other effects. While we increasingly see sensor noise models in robotics simulators and/or many provide an API/plugin architecture to include such models, actuator dynamics and delays are rarely included by default and thus often remain unmodeled. As the following study shows, these effects play an important role when assessing the performance, stability, and robustness of a closed-loop control system. While motion planning and trajectory execution usually translates well from simulation to hardware, feedback controllers and their gains usually require separate tuning on hardware. In order to understand this phenomena, we analyze the influence of actuator dynamics, noise and delays with regards to well established control theory performance and stability measures. The analysis is carried out for a conceptual one degree of freedom actuator attached to a single body, where the model parameters are inspired by actuators of state-of-the-art quadrupedal robots.

2. Model Description

In order to make a transparent analysis, we study the properties of a single degree of freedom articulated rigid body model, attached to a linear or cylindrical joint. This single DoF system is assumed to be driven by a force/torque-controlled actuator, which has its own open and closed loop dynamics, and tries to track a desired force/torque, usually generated by an outer control loop. As in many state of the art systems, we use a position controller as an outer control loop. While investigating a single DoF rigid body system seems like a strong simplification, such a model is still able to capture the essential dynamics and stability limitations we consider in this paper. In addition, many state-of-the-art robots rely on low-level single input single output (SISO) joint controllers, which are usually designed for their respective single DoF. In these cases, the coupling with the overall multi-body system is usually seen and treated as a disturbance to the SISO system, and as such its influence on the closed-loop performance and stability can also be analyzed using the tools and methods presented here.

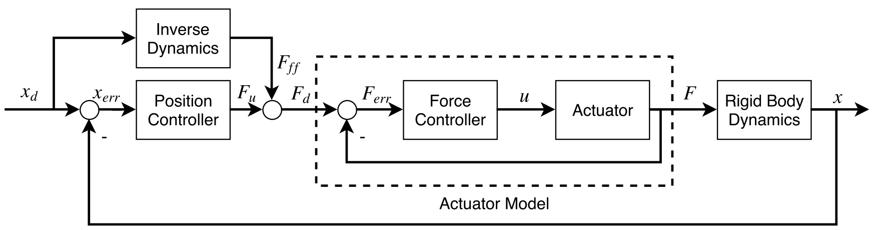

Fig. 1. Model of a typical cascaded control structure of a single articulated rigid body. The actuator itself is driven by an inner force controller which is usually complemented by an outer position or impedance controller.

2.1. *Actuator Model*

In this work we consider the actuator model, depicted by a dashed box in Figure 1, as the closed-loop transfer function between the desired and the actual actuator force command. In the case of an electric actuator, the model would include the controller of the motor electronics as well as the dynamics of the rotor and the gearbox. For a hydraulic actuator this would include, besides the force controller, the valve and cylinder dynamics. While not considered here, friction and other non-linear effects could be included in this model as well. For our examples, we study the behavior of actuators as found in the state-of-the-art quadrupeds HyQ[7] as well as StarlETH.[8] Based on the data presented by the respective developers, we can see that both clsed-loop dynamics can be coarsely approximated by a second order system with delay defined as $P(s) = e^{-t_d s}\frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$. More precisely, based on Figures 6.11 and 7.3 in Ref. 9 as well as Figure 4.3 in Ref. 8, we approximate the parameters of the second order system for intermediate performance points as $\omega_n = 800, \zeta = 0.45$ for HyQ and $\omega_n = 110, \zeta = 0.9$ for StarlETH. Figure 4.3 in Ref. 8 shows a worst case delay of 5 ms. We decided to consider a more ideal time delay of 1 ms. Naturally, our fitted models are oversimplified and will not capture all the effects of the closed-loop force controlled system. Especially, it does not model potential internal instabilities of the force controller. However, this simple model is sufficient to identify important dynamical effects concerning control stability. In general, it is application-dependent how detailed the chosen models should be. Thus, we consider this discussion beyond the scope of this paper. The actuator model output directly acts on the rigid body dynamics. Here, we model a single DoF body with mass $m = 0.7$ kg. Based on its properties it can be imagined being a typical endeffector of a quadruped or humanoid robot. While we assume a linear joint, the findings below also qualitatively translate to rotary joints. We further assume that a position controller closes the loop around the actuator model and rigid body dynamics. It is represented as a Proportional-Derivative (PD) controller with fixed gains at $k_p = 400$ and $k_d = 20$ respectively.

2.2. *Noise, Delays and other Effects*

Sensor models that partially include noise, biases and delays can help to evaluate state estimation approaches. However, from a control perspective, noise can impair the control performance and depending on the sensitivity of the system it can be even amplified. Another important influence on control stability of robotic systems are delays. These delays occur due to communication between components, sampling times, computational time required by algorithms, asynchronous computation of discrete controllers or internal delays of components. In this work we assume all occurring delays are pure time delays and can be attributed to the actuator model. Based on

Figure 4.3 in Ref. 8 we assume that the worst-case total internal delay accumulates to 5 ms. Apart from delay and noise, there are additional effects that can affect control stability and performance, which we intentionally neglect in this study. Amongst others, these are internal (non-linear) effects of the actuation model such as saturation, measurement quantization, discretization and sampling, aliasing, static friction, gear backlash, etc.

3. Sensor-actuator effects on a robotic joint controller

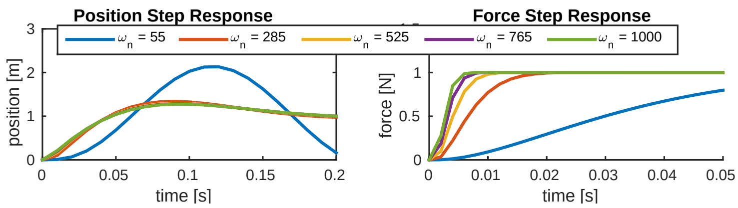

Fig. 2. Step responses of the position control loop and the actuator dynamics for different resonance frequencies ω_n of the actuator model. Unless very slow actuator dynamics are considered, the difference in position step responses are very small, suggesting that for evaluating kinematic plans, actuator dynamics do not play a significant role.

In the following analysis we compare neglecting actuator dynamics and approximating them with a simple linear second order model as described above. To get a first impression on how actuator dynamics influence the accuracy of a simulator, we look at step responses both of our second-order actuator model with delay as well as the closed-loop position controlled system. During this analysis we vary the resonance frequency ω_n of the second-order actuator model. When looking at these step responses in Figure 2, we can observe that the force response significantly varies in terms of rise times for different resonance frequencies w_n. Yet, these differences appear on a relatively small time scale. When looking at the step response of our position controller, we see that for time scales comparable to the dynamics of the rigid body system, the differences between various actuator dynamics become small unless they are very slow. So if we are only interested in obtaining a good approximation of the rigid body motion as e.g. in evaluating kinematic plans, neglecting the actuator dynamics seems acceptable. Although the position step response is not deeply affected by the low-level actuation system dynamics, the stability and robustness of the closed-loop position controlled system may be significantly affected. Therefore, we will investigate how controller stability metrics are affected by the actuation dynamics. We focus on three well established control performance criteria: Gain margin, phase (delay) margin, and sensitivity.

3.1. *Influence of actuator dynamics on the gain margin*

The gain margin GM gives us a measure of how much we can increase our feedback gains before the system becomes unstable. It is defined as the offset of the open loop magnitude to 0 at the frequency ω_{pc} where the open loop phase crosses -180°: $GM = 1/|G(j\omega_{pc})|$. We compute the gain margin for both the closed loop system with and without actuator dynamics. The results are shown in Figure 3. The system without actuator dynamics is not plotted since it has infinite gain margin. This means, we could poten-

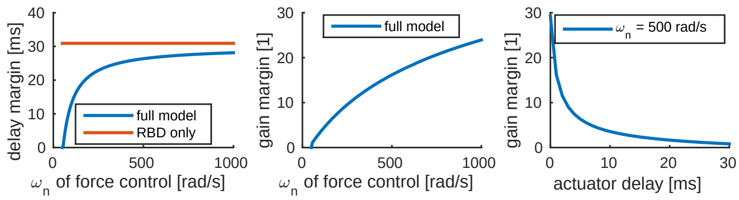

Fig. 3. Gain and delay margin (blue) for the position loop gain for different resonance frequencies ω_n of the actuator dynamics. The reference system without actuator dynamics (red) has infinite gain margin.

tially have infinitely high gains without the controller becoming unstable. Of course, this fact does not hold in simulation since we observe instabilities with finite gains. This results from the discrete implementation of our controllers and system dynamics. However, this also means that we can mitigate this limitation by increasing our sampling/integration rates[10] to prevent instability of the controller. Although we can also increase sampling on real hardware or simulate the system with actuator dynamics, this will not help indefinitely in the latter cases. For instance, Figure 3 shows that our gain margin monotonically decreases with the dynamics of our actuator model. While the absolute numbers might not be representative, it is important to know that gain margin can drop by a factor of 3 to 5 for the chosen range. Additionally, as the right graph in Figure 3 shows, the gain margin is heavily influenced by delays of the actuator dynamics. Even delays in the millisecond range influence the gain margin significantly.

3.2. *Influence of actuator dynamics on the delay margin*

The phase margin is a measure of how much extra phase (or pure time delay) our controller can cope with. It corresponds to the loop gain phase at the frequency that the magnitude crosses 0 db. While there might be more than one such crossing, the delay margin represents the phase margin that is closest to instability when adding delay. The delay margin is thus defined as $DM = min(180^\circ + \angle G(j\omega_{gc}))$. In Figure 3 we plot the delay margin for different values of our second order actuator model resonance frequency. We can observe that the delay margin asymptotically approaches the one of the model without actuator dynamics. For dynamics below 200 rad/s which we e.g. see in StarlETH, the gain margin drops significantly. Any additional delay in the system will shift the delay margin further down. This analysis reveals two interesting aspects. First, if for a given system the delay margin is already small, it is important to model delays to make sure that delays in an outer control loop (here the position controller) do not make the system unstable. Furthermore, the results also suggest that for systems with parameters in comparable range to the example, small delays in the millisecond range might lead to an unstable system. Considering that common position control loop rates often lie between 100 to 1000 Hz and occasionally run asynchronous with sensing, we can expect delays of up to 5 ms[8] and above, which can become critical for actuators with slow dynamics ($\omega_n < 200 rad/s$). Delays can become even worse when controllers are not closed on joint-level but on a whole-body-level, where complex state estimators can introduce extra delays.

3.3. *Influence of actuator dynamics on the sensitivity*

The sensitivity analysis tells us how variations in the process influence the behavior of the system. Using the standard definition of the sensitivity function, we can compute its peak, the nominal sensitivity peak as $M_s = \underset{0 \leq \omega < \infty}{max} |S(j\omega)| = \underset{0 \leq \omega < \infty}{max} \left| \frac{1}{1+G(jw)C(jw)} \right|$. Since the sensitivity describes how measurement output influences the systems feedback behavior, it also tells us the influence of noise and disturbances on our system. A sensitivity greater than 1 tells us that disturbances or noise at the corresponding frequency get amplified by the system. Therefore, we focus on two aspects in the sensitivity analysis: what the nominal sensitivity peak is and at what frequency it occurs. By looking at the bode plot of the sensitivity function in Figure 4, we can see that the system nicely attenuates disturbances and noise in the lower extreme of the frequency spectrum. This attenuation results from the outer control loop and is not influenced by the actuator model. At the higher extreme, any disturbance is passed through by the feedback, also with no dependency on the model. Since we expect sensor noise at higher frequencies, we assume that noise rejection behaves similar with or without an actuator model. Depending on the rigid body mass and the position control parameters, the frequency at which we observe the sensitivity peak varies between around 1 to 100 rad/s. For

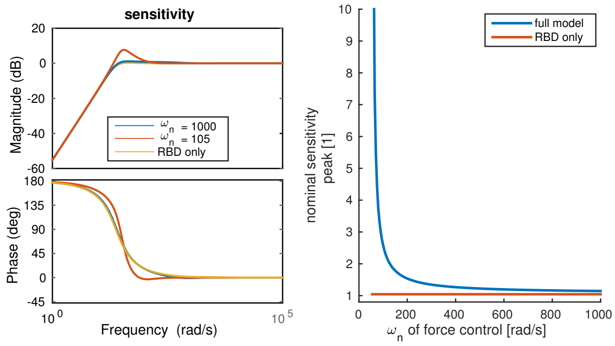

Fig. 4. The left plot shows the sensitivity between output and feedback of the closed loop system for different values of the actuator dynamics resonance frequency ω_n as well as the actuator dynamics free simulation. The right plot shows the nominal sensitivity peak as a function of the actuator dynamics resonance frequency ω_n.

the given parameters we find a sensitivity peak at a frequency around 40 rad/s. As Figure 4 shows, the magnitude of the peak changes with different actuation models. Slower actuator dynamics lead to a higher peak sensitivity, making the system more vulnerable to disturbances with corresponding frequency. The sensitivity analysis tells us that actuator models are less relevant to noise rejection characteristics than to disturbances. Thus, modeling actuator dynamics becomes important where we expect significant disturbances and want to verify control stability under these conditions.

3.4. *Summary of effects on the control stability*

We have seen that actuator dynamics and delays have a significant influence on control loop stability and sensitivity. Without modeling actuator dynamics, gain margin is infinite and thus, we might unintentionally overtune our controllers in simulation. Also, if delays are unmodeled we might oversee a lack in phase margin also rendering the controller unstable on the real hardware. While we have analyzed the influence of parameters independently, there is cross coupling between the resulting effects.

4. Simulating a low-level actuation system

As the analysis above shows, we should include a model of our actuators in our simulator for control performance and robustness analysis. Hence, we implement a Gazebo plugin that simulates second-order actuator dynamics. Furthermore, we add setpoint, measurement, and output delays. Last but not least we model A/D and encoder quantization as well as sensor noise and biases. Our model of the low-level input-output and control system is evaluated at the same frequency as on hardware, ensuring consistent sampling time. Due to the simplicity of the actuator model, the simulator can still run in real-time, despite integrating 18 actuator systems at kilohertz rates. To test the improved simulator, we increase our hardware tuned, stable PD gains by a factor of 3. We then run our simulation with delays and the actuator model and once without. As shown in the video (`https://youtu.be/GNCFuhsuTkE`)), the bare simulation suggests that these gains are stable. Enabling the low-level actuation model, reveals the instability due to the over-tuning. The unaltered hardware tuned gains are stable both in the bare and in the improved simulator.

5. Conclusion and Outlook

In this work, we have analyzed the importance of modeling the low-level actuation system of a robot to be able to better evaluate control performance and stability in simulation. An actuator dynamic free, noise-free simulation does not reveal gain margins, encouraging over-tuning of feedback gains. Also an actuator model can help to understand how noise and disturbances get rejected by the system. While we performed a first analysis of our improved simulator with added actuator dynamics, noise and delays, we will perform an in-depth comparison to the real system and also study the trade-off between actuator model simplicity and accuracy.

Acknowledgments

This research has been funded through a Swiss National Science Foundation Professorship award to Jonas Buchli, by the Swiss National Centre of Competence in Research Robotics, and the EU Project BALANCE (Grant 601003 of the EU FP7 program).

References

1. S. Peters and J. Hsu, Comparison of rigid body dynamic simulators for robotic simulation in gazebo, in *ROSCon*, 2014.
2. Gazebo `http://gazebosim.org/`, 2016-02-20.
3. E. Rohmer, S. P. Singh and M. Freese, V-REP: A versatile and scalable robot simulation framework, in *Intelligent Robots and Systems (IROS)*, 2013.
4. O. Michel, WebotsTM: Professional mobile robot simulation (2004).
5. R. Smith *et al.*, Open dynamics engine (2005).
6. E. Coumans *et al.*, Bullet physics library (2013).
7. C. Semini, N. G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella and D. G. Caldwell, Design of HyQ – a hydraulically and electrically actuated quadruped robot, *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* 2011.
8. M. Hutter, StarlETH & Co – design and control of legged robots with compliant actuation, PhD thesis, ETH Zürich, 2013.
9. T. Boaventura, Hydraulic compliance control of the quadruped robot HyQ, PhD thesis, Advanced Robotics Department, University of Genova,2013.
10. J. Tan, K. Liu and G. Turk, Stable proportional-derivative controllers, in *Computer Graphics and Applications*, 2011.