Towards Dynamic Walking for StarlETH

Christian Gehring*/**, Stelian Coros* and Roland Siegwart**
* DRZ, Disney Research Zurich, Clausiusstrasse 49, 8092 Zurich, Switzerland
stelian.coros@disneyresearch.com, christian.gehring@disneyresearch.com
** ETH Zurich, Tannenstrasse 3, 8092 Zurich, Switzerland
rsiegwart@ethz.ch

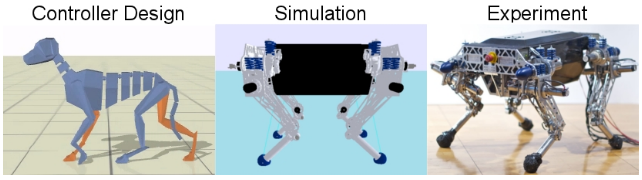

Figure 1: The initial model for the control design (left), Starl*ETH*in SL (middle), and the actual platform (right).

1 Motivation

The agile, graceful and seemingly effortless motions of humans and animals continue to inspire scientists in the field of robotics. Promising locomotion control strategies that work well in simulated environments have been proposed [3]. However, control algorithms tuned in simulation tend to fail when first applied on real-world robotic platforms. The main reason for this is that modeling errors and the various approximations used in numerical simulations (contact and friction models, deformation of structures under load, motor power and bandwidth, sensor noise, time delays, etc.) lead to environments that do not match the real world sufficiently well. However, simple and fast computational models are certainly desirable - for automated parameter tuning, for instance. On the other hand, if simulation results do not guarantee the success of controllers on real-world platforms, then they are of limited use.

To our knowledge, as far as legged robots are concerned, there are currently no research projects aimed at documenting the efforts needed to generalize simulation results to real-life robots. We take a first look at this problem, as we control the compliant quadruped Starl*ETH*, shown in Fig. 1, both in simulation and in real-life. We are interested in determining which sources of simulation error can be safely ignored, and which ones have to be addressed. We are also interested in determining which aspects of the control algorithms need to be adapted in order to increase the robustness of legged robots in real-world environments.

2 State of the Art

Due to its fundamental importance, the problem of locomotion control has been studied extensively. Raibert's hopping robots [8, 6] and Boston Dynamic's BigDog [2] are amongst the most successful examples of legged robots, as they can hop or trot robustly even in the presence of significant unplanned disturbances. A large number of humanoid robots have also been introduced. Honda's ASIMO [9] and Samsung's Mahru III [5] are capable of walking, running, dancing and going up and down stairs, and the Yobotics-IHMC [7] biped can recover from pushes. Despite these accomplishments, the agile, graceful and skilled motions of real humans and animals remain, for now, unmatched.

For this work we perform experiments on a recently developed quadruped robot, Starl*ETH*(Springy Tetrapod with Articulated Robotic Legs) [4]. Starl*ETH* uses an actuation scheme based on highly compliant series elastic actuators, which enables torque control. The robot is currently only capable of static walking, and one of our aims is to increase its repertoire of motions to include faster, more life-like dynamic gaits such as trotting, bounding and galloping.

3 Approach

The goal of our work is to investigate control methods that lead to life-like quadrupedal robots that can operate in every day environments. Such environments, in general, are only partially observable. Appropriate control methods need to therefore be robust to unplanned disturbances such as variations in terrain or external forces. The control strategy discussed in [3] is capable of handling such unplanned interactions (demonstrated in simulations). The controller combines several simple building blocks. An inverted pendulum model computes desired foot fall locations, PD controllers regulate the motions of the legs and virtual forces are used to counteract the effects of gravity and allow for continuous center-of-mass velocity modulation.

We begin by adapting this control strategy to a simulated model of Starl*ETH*. Before applying the controller to the hardware platform we plan on running

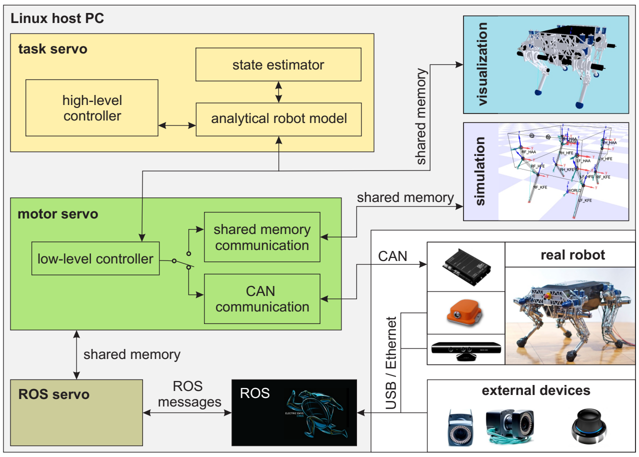

Figure 2: The software architecture.

tests aimed at ensuring that the control method works well under real-world constraints. To this end, we will incrementally increase the complexity of the control task in order to account for various sources of error and hardware limitations.

Initial tests show that a trotting gait can be successfully controlled in our simulated environment which is based on the software package (SL) [10]. This software package allows us to use exactly the same control code either for simulations or directly on Starl*ETH*. This makes it ideally suited for the goals of this project, as we aim to study the difference between simulated and real-world robotic platforms.

Figure 2 illustrates the software architecture we use on a Linux host PC. The software is divided into various processes that run in parallel. The aforementioned control algorithm is executed on the process called *task servo*, and is represented by the *high-level controller* block in the diagram. The controller has access to an *analytical robot model* that keeps track of the state of the robot using a *state estimator*. A *low-level controller* runs on a different process, the *motor servo*. It receives the computed commands from the high-level controller and sends them to either the *simulation* process or the the motor controller boards (shown in black via CAN). A fourth process, the *ROS servo*, receives and processes additional sensory signals and command inputs, e.g. IMU and Kinect measurements, VICON ground truth measurements, joystick commands, etc.

The main difficulty we face, as we transition from simulation to the real-world, is due to the fact that the high-level controller outputs torques. Torque control is easy to implement in simulation, but on robotic platforms it presents some challenges, as the electrical motors we use can only directly control angular velocities. Since the actuation used is highly compliant, the control bandwidth is strongly reduced. This limitation, as well as limits on the net torques that the actuators can output, must be taken account. It is not yet clear if it is sufficient to clamp the output joint torques as a post-process, or if the bandwidth and actuator limitations have to be considered explicitly by the controller. The control strategy additionally needs to be augmented by a position controller, as the low bandwidth nature of the actuation model does not suffice for fast leg positioning during swing. This hybrid (position/torque) control structure increases the influence of the control update rate on the performance of the system, since the precise moment when the control mode is switched is very important. A further physical limitation that needs to be considered at the control level is the effective range of motion of the quadruped. What should happen if a joint reaches its limit angle? One option would be to conclude that the controller failed, or we could actively counteract the torques by a virtual wall. Such decisions can greatly influence the behavior of the controller.

When dealing with real-world hardware, the model of the robot the controller operates on (see the *analytical robot model* block) usually differs from the real robot to some extent. In simulation, the exact model can be used both for control and for dynamics. Separating the model used by the controller from the model used in simulation allows us to evaluate such modeling errors and their effects on the controller robustness. Relatedly, when dealing with numerical simulations, we usually have access to the full state of the robot at any time. In real-life, state estimation relies on sensor signals that are often noisy. Filter consistency problems and time delays add further complications [1]. We aim to study the effects of noisy state estimations and robot modeling errors in simulations first, before applying our control method to Starl*ETH*.

4 Discussion outline

The discussion we propose will touch on two separate aspects related to control of legged robots. First, we will present the experiments used to test the significance of the various sources of error described above, and their effect on controller performance. Second, we plan to discuss the changes that are necessary in order to adapt the control algorithm to Starl*ETH*, given these real-world constraints. As an example, the control approach was initially desined for fast gaits. As we moved from simulation to the real platform, we wanted to start with slow motions for safety reasons. This led to some adaptations of the control strategy since some aspects of static walking (accurate control over of the center of mass position, force distribution to the stance legs) are more important than in dynamic gaits.

A question that will eventually be answered in this study is: at which level of simulation accuracy will the simulated results be indicative of what would happen in real-life? Consequently, how significant are simulation results for a control algorithm that is evaluated independent of a robotic platform?

5 Keywords

locomotion control

References

[1] M. Bloesch, M. Hutter, M. H. Hoepflinger, C. D. Remy, C. Gehring, and R. Siegwart. State estimation for legged robots - consistent fusion of leg kinematics and imu. *submitted to RSS Robotics Science and Systems*, 2012.

[2] P. R. Buehler, M. and M. Raibert. Robots step outside. *Int. Symp. Adaptive Motion of Animals and Machines (AMAM)*, 2005.

[3] S. Coros, A. Karpathy, B. Jones, L. Reveret, and M. van de Panne. Locomotion skills for simulated quadrupeds. *ACM Transactions on Graphics*, 30(4):Article TBD, 2011.

[4] M. Hutter, C. Gehring, M. Bloesch, M. H. Hoepflinger, C. D. Remy, , and R. Siegwart. Starleth: A compliant quadruped for fast, efficient, and versatile locomotion (under review). 2012.

[5] K. H. K. P. J. K. R. C. H. L. J. P. J. k. K. W. Kwon, W. and K. Roh. Biped humanoid robot mahru iii. *7th IEEE-RAS International Conference on Humanoid Robots*, 2007.

[6] S. S. Murthy and M. H. Raibert. 3d balance in legged locomotion: modeling and simulation for the one-legged case. *Inter-Disciplinary Workshop on Motion: Representation and Perception*, 1983.

[7] J. Pratt and B. Krupp. Design of a bipedal walking robot. *Proceedings of the 2008 SPIE*, 2008.

[8] M. H. Raibert. Four-legged running with one-legged algorithms. *Second International Symposium on Robotics Research*, 1985.

[9] W. R. A. C. M. S. H. N. Sakagami, Y. and K. Fujimura. The intelligent asimo: system overview and integration. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002.

[10] S. Schaal. The sl simulation and real-time control software package. 2006.