Walking on the Sarcos Hydraulic Humanoid

Eric C. Whitman
Carnegie Mellon University
ewhitman@cmu.edu

1 Motivation

For walking robots to be practically useful, they must be able to cope with significant unexpected disturbances. It is inevitable that they will bump into things and that things will bump into them. Outside of a laboratory setting, it is very difficult to accurately model the environment.

We would also like our robots to function well in the face of modeling error. It is very difficult and time-consuming to accurately model a complex robot. Even if it can be modeled accurately, we would like our controllers to continue to function well when the model changes, either due to normal wear or minor modifications. It is common to get a controller working well in simulation only to have it fail when implemented on physical hardware.

We present preliminary walking results for our hydraulic force-controlled humanoid robot as well as the steps we had to take when moving from simulation to hardware in order to cope with modeling error.

2 State of the Art

Many walking controllers focus on CoM motion. A standard method of control is to first generate a CoM trajectory and then track that trajectory with inverse kinematics [1]. Unfortunately, even when tracking an optimal trajectory, the resulting controller is only optimal when near the desired trajectory, which is not the case following a significant unexpected disturbance. Due to constraints on reaction forces, linear independent joint controllers often can stabilize only a small region of state space. It is possible to frequently recalculate the CoM trajectory, taking into account the current robot state [2]. Model Predictive Control (MPC) and receding horizon control offer methods of generating trajectories online that continuously start from the current robot state [3].

For the system to recover from large disturbances, it is necessary to modify the reaction force constraints by adjusting the footstep placement or timing. One possible approach to this is trajectory libraries, where multiple trajectories are generated in advance and an appropriate one is used depending on the current robot state. Examples of trajectory libraries are given for standing balance in [4] and for walking in [5]. It is also possible to modify MPC so that it determines foot placement online [6]. In [7], the footstep timing is modified online in response to manually changed footstep locations.

Because of many walkings systems' high-dimensionality, which makes control difficult, it is common to model parts of a walking system as decoupled so that the lower-dimensional subsystems can be controlled separately [8], [9]. PD servos on individual joints is a very basic form of such decoupling. Unless coordination is handled carefully, the combined controller will be sub-optimal because the subsystem controllers lack the information necessary to make optimal decisions. We present a method of coordination that produces an optimal combined controller.

3 Own Approach

3.1 Simulation Controller

In [10], we describe a walking controller that performs well in simulation. We avoid the generation and tracking of trajectories. Instead, we focus on creating a state space controller that maps directly from state to action. Dynamic programming can generate state space controllers for complex non-linear systems and cost functions, but is only computationally feasible for low-dimensional state spaces. To cope with this limitation, we break our walking model into several subsystems. These subsystems (Sagittal center of mass, Coronal center of mass, Swing foot X, Swing foot Y, and Swing foot Z) are only coupled at touchdown and liftoff, and then only by the time and location of touchdown/liftoff. Except at these instants, the subsystems are decoupled and evolve independently. We refer to models with this structure as Instantaneously Coupled Systems.

If we knew ahead of time the time and location of touchdown and liftoff, we could optimally solve for each of the subsystems individually because they have no further interaction with each other. However, the optimal time and location of touchdown depends on the state of all of the subsystems. If we only consider the next transition event and assume all future transitions will have nominal values, we can solve for optimal controllers for each subsystem and then combine them. We augment each subsystem with coordination variables (time and location of touchdown) and then use dynamic programming to produce optimal controllers.

In addition to policies, dynamic programming produces value functions (a measure of cost-to-go as a function of state). Given the system state, we can get a value function for each subsystem as only a function of the added coordination variables. Adding these together gives a value function for the combined system as a function of the coordination variables. Minimizing this gives the optimal value for the coordination variables (time and location of touchdown). Once these have been selected, each subsystem can be controlled independently.

This method produces a controller that is valid for a large region of state space, so can cope with large disturbances that push it far from the nominal walking cycle. It can react at each control cycle (about 1 kHz) to a disturbance by adjusting the planned timing and location of the next touchdown or liftoff event as well as the motion of the individual subsystems.

3.2 Hardware Controller

When we tried to implement this on our hydraulic force-controlled humanoid robot, we found that the system had significant unmodeled dynamics. Unmodeled higher-order modes caused the robot to react very poorly (oscillations, instability, etc.) to large jerks. We solved this by adding acceleration as an additional state to our swing leg policies and controlling jerk. We continued to use the acceleration in the inverse dynamics to generate torques, but this ensured that the acceleration varied smoothly and did not excite the unmodeled higher-order modes.

Achieving desired accelerations through inverse dynamics requires an accurate inertial model. This is especially true for the swing leg, which has a small mass and high accelerations. In order to hit touchdown target locations despite an inaccurate inertia model, we had to add individual joint PD gains. We found desired CoM and foot positions by integrating the desired acceleration, then used inverse kinematics to find desired joint angles. These gains have very little effect on the stance leg, but significantly impact the dynamics of the swing leg.

3.3 Results

We present simulation results demonstrating the robustness of our controller to unexpected disturbances such as pushes (both continuous and impulsive), trips, ground elevation changes, and slopes. We also present preliminary walking data on our Sarcos humanoid force-controlled hydraulic robot.

4 Discussion Outline

1. Should we, as researchers, be putting our resources into more accurately modeling our robots or into generating controllers than can cope with less accurate models?
2. Do animals have an accurate dynamic model of their own bodies?
3. How can we reduce the sensitivity of our controllers to inaccuracy in our dynamics models?

5 Format

Either a talk or a poster would be fine. No Preference.

References

[1] A. Takanishi, T. Takeya, H. Karaki, and I. Kato, “A control method for dynamic biped walking under unknown external force”, in *IEEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications*. 1990, vol. 29, pp. 795–801, IEEE.

[2] Koichi Nishiwaki and Satoshi Kagami, “High frequency walking pattern generation based on preview control of ZMP”, in *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2006, pp. 2667–2672.

[3] Pierre-Brice Wieber, “Trajectory free linear model predictive control for stable walking in the presence of strong perturbations”, in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, 2006.

[4] Chenggang Liu and Christopher G. Atkeson, “Standing balance control using a trajectory library”, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

[5] Pierre-Brice Wieber and Christine Chevallereau, “Online adaptation of reference trajectories for the control of walking systems”, *Robotics and Autonomous Systems*, vol. 54, no. 7, pp. 559–566, 2006.

[6] Holger Diedam, Dimitar Dimitrov, Pierre-Brice Wieber, Katja Mombaur, and Moritz Diehl, “Online walking gait generation with adaptive foot positioning through linear model predictive control”, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008.

[7] Mitsuharu Morisawa, Kensuke Harada, Shuuji Kajita, Shinichiro Nakaoka, Kiyoshi Fujiwara, Fumio Kanehiro, Kenji Kaneko, and Hirohisa Hirukawa, “Experimentation of humanoid walking allowing immediate modification of foot place based on analytical solution”, in *Proceedings of IEEE International Conference on Robotics and Automation*, October 2007, pp. 3989–3994.

[8] Michiel van de Panne, Eugene Fiume, and Zvonko G. Vranesic, “A controller for the dynamic walk of a biped across variable terrain”, in *Proceedings of the 31st Conference on Decision and Control*, December 1992, pp. pp. 2668–2673.

[9] KangKang Yin, Kevin Loken, and Michiel van de Panne, “Simbicon: simple biped locomotion control”, in *SIGGRAPH ’07: ACM SIGGRAPH 2007 papers*, New York, NY, USA, 2007, p. 105, ACM.

[10] Eric C. Whitman and Christopher G. Atkeson, “Control of instantaneously coupled systems applied to humanoid walking”, in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, December 2010.