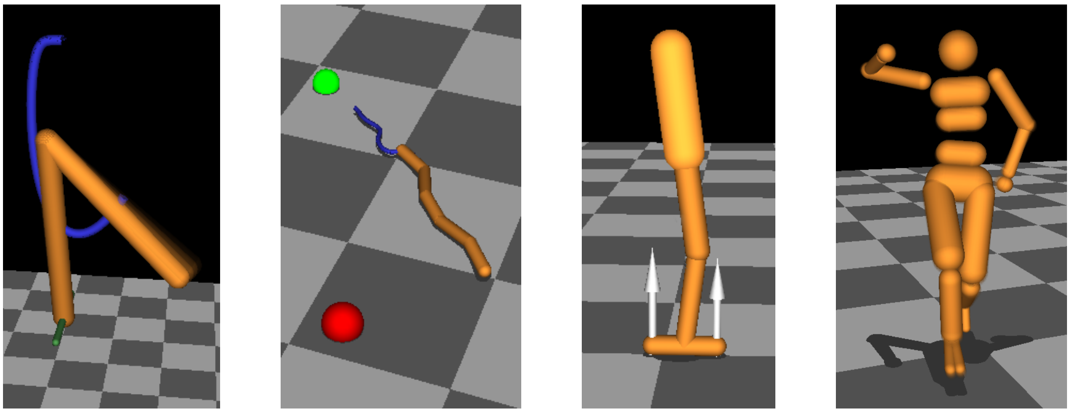
Synthesis of Robust Behaviors through Online Trajectory Optimization

Yuval Tassa, Tom Erez and Emo Todorov
University of Washington, Seattle
{*tassa,etom,todorov*}@*cs.washington.edu*



See movie at dl.dropbox.com/u/56715/humanoid.m4v

1 Summary

We present an online trajectory optimization method and software platform applicable to complex humanoid robots performing challenging tasks such as getting up from an arbitrary pose on the ground and recovering from very large disturbances using adroit acrobatic maneuvers. The resulting behaviors, illustrated in the movie referenced in the figure caption, are computed only 7 times slower than real time, on a single desktop computer. The movie also shows results on the acrobot problem, planar swimming and one-legged hopping. These simpler problems can already be solved in real time, without pre-computing anything and without specifying heuristic approximations to the value function. We show both robustness to state perturbations, a generic feature of the online approach since it always optimizes the trajectory starting at the present (possibly perturbed) state, and robustness to large modeling errors (unknown to the controller).

2 Motivation

The framework of optimal control makes it possible to specify high-level task goals through simple cost functions and synthesize the details of the behavior and control law automatically. In practice however, optimal control is rarely applied to systems with high-dimensional state spaces – due to the curse of dimensionality. The curse is particularly problematic for humanoid robots, whose state space is so large that no control scheme (optimal or not) can explore all of it in advance, and prepare suitable responses for every situation. The most impressive results to date have been achieved by local methods, which side-step the curse of dimensionality at the price of local minima and open-loop control. A further complication in the context of locomotion are contact phenomena – which are inherently discontinuous and thus difficult to handle automatically.

Here we describe our recent efforts to overcome the above limitations. Our goal is to construct intelligent feedback controllers that are not limited to the vicinity of precomputed trajectories, and to do so fully automatically, without need for manual specification of contact phenomena or any other aspects of the behavior. We achieve this through online trajectory optimization, also known as model-predictive control (MPC) or receding-horizon control. The idea is to optimize a trajectory up to some time horizon starting at the current state, apply the initial control signal along this trajectory, and repeat. The previous solution is used to warm-start the optimizer, which often yields convergence after a single step, if the minimum has been well-tracked. Contacts are handled automatically by contact smoothing methods we have developed.

The main difficulty in applying MPC is the need to (re)optimize movement trajectories in real time. This may seem impossible for a 3D humanoid performing a complex task. However advances on multiple fronts have brought us surprisingly close to this goal. Controlling the 23-dof humanoid illustrated in the figure (in the task of getting up from an arbitrary pose) is currently only 7 times slower than real time, on a single desktop machine. The computational bottleneck is in the derivatives of the dynamics, computed via finite differencing, which in turn is “embarrassingly parallel”. Thus if we were to connect 10 such computers in a cluster, we should be able to control this humanoid in real time. We have already implemented the distributed version of our code, and hope to be able to demonstrate such real time control before the workshop.

3 State of the Art

In domains such as chemical process control where the dynamics are sufficiently slow and smooth – and thus online trajectory optimization is already feasible – MPC is the method of choice [1]. In robotics, however, the typical timescales of the dynamics are orders of magnitude smaller. Furthermore, many robotic tasks involve contact phenomena that present a serious challenge to optimization-based approaches. As a result, MPC is rarely used to control complex robots. Autonomous helicopter flight [2] is a recent example of the power of MPC applied to robotics, although that system is lower-dimensional and smooth. Another illustration is our work on bouncing two ping-pong balls on the same paddle [3]. This task involves contacts but the dimensionality is still relatively low.

The ball-bouncing example relies on a heuristic approximation to the optimal value function, which is used as a final cost applied at the MPC horizon. In general there is a natural tradeoff between how good the value function approximation is, and how much work the MPC machinery has to do [4]. Here we focus on the case when no such approximation is available, and all the work is done by MPC. Thus our results are in some sense worst-case results, and the performance of our method can be improved by using suitable value function approximations. One way to obtain such approximations automatically is to apply machine learning methods to the vast amount of data generated by the MPC controller.

Another approach to generating intelligent feedback (beyond linear) with trajectory optimizers are aggregation methods, like trajectory libraries [5] and LQR trees [6]. While this approach is promising, our guess is that it will eventually run into the curse of dimensionality. Presumably as the volume of state space grows with the dimension, either the number of local controllers will have to grow exponentially, or the region of validity of each local controller will have to grow exponentially, either way susceptible to an explosion in computational complexity.

4 Our Approach

The results presented here are enabled by advances on multiple fronts. Our new physics simulator, called MuJoCo, was used to speed up the computation of dynamics derivatives. MuJoCo is a C-based, platform-independent, multithreaded simulator tailored to control applications. We developed several improvements to the iterative LQG method for trajectory optimization [7] that increase its efficiency and robustness. We also developed several models of contact dynamics [8, 9, 10] which yield different trade-offs between physical realism and speed of simulation/optimization. We introduced cost functions that result in better-behaved energy landscapes and are more amenable to trajectory optimization. Finally, we developed a MATLAB-based environment where the user can modify the dynamics model, cost function or algorithm parameters, while interacting in real time with the controlled system. We have found that the hands-on familiarity with the various strengths and weaknesses of the MPC machinery is invaluable for proper parameter tuning.

[1] M. Diehl, H. Ferreau, and N. Haverbeke. Efficient numerical methods for nonlinear mpc and moving horizon estimation. *Nonlinear Model Predictive Control*, page 391, 2009.

[2] P. Abbeel, A. Coates, M. Quigley, and A. Y Ng. An application of reinforcement learning to aerobatic helicopter flight. In *Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference*, page 1, 2007.

[3] P. Kulchenko and E. Todorov. First-exit model predictive control of fast discontinuous dynamics: Application to ball bouncing. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, page 2144–2151, 2011.

[4] T. Erez, Y. Tassa, and E. Todorov. Infinite horizon model predictive control for nonlinear periodic tasks. *Manuscript under review*, 2011.

[5] Chenggang Liu and C. G Atkeson. Standing balance control using a trajectory library. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009. IROS 2009*, pages 3031–3036. IEEE, October 2009. ISBN 978-1-4244-3803-7.

[6] R. Tedrake. LQR-Trees: feedback motion planning on sparse randomized trees. In *Proceedings of Robotics: Science and Systems (RSS)*, 2009.

[7] E. Todorov and Weiwei Li. A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 300–306, Portland, OR, USA, 2005.

[8] Emanuel Todorov. Implicit nonlinear complementarity: a new approach to contact dynamics. In *International Conference on Robotics and Automation*, 2010.

[9] Y. Tassa and E. Todorov. Stochastic complementarity for local control of discontinuous dynamics. In *Proceedings of Robotics: Science and Systems (RSS)*, 2010.

[10] Emanuel Todorov. A convex, smooth and invertible contact model for trajectory optimization. In *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1071–1076. IEEE, May 2011. ISBN 978-1-61284-386-5.

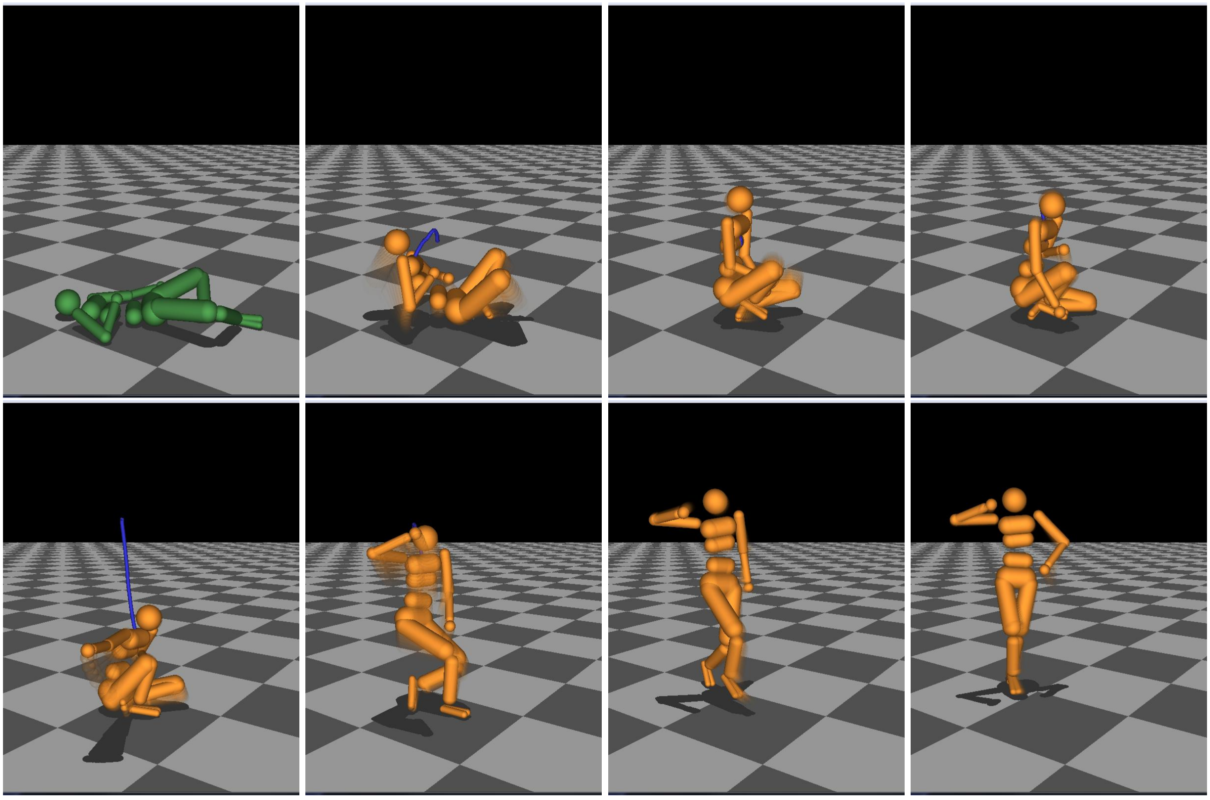


Figure 1: Getting up. The blue line shows the planned position of the torso. This sequence is at 3m50s in the movie.

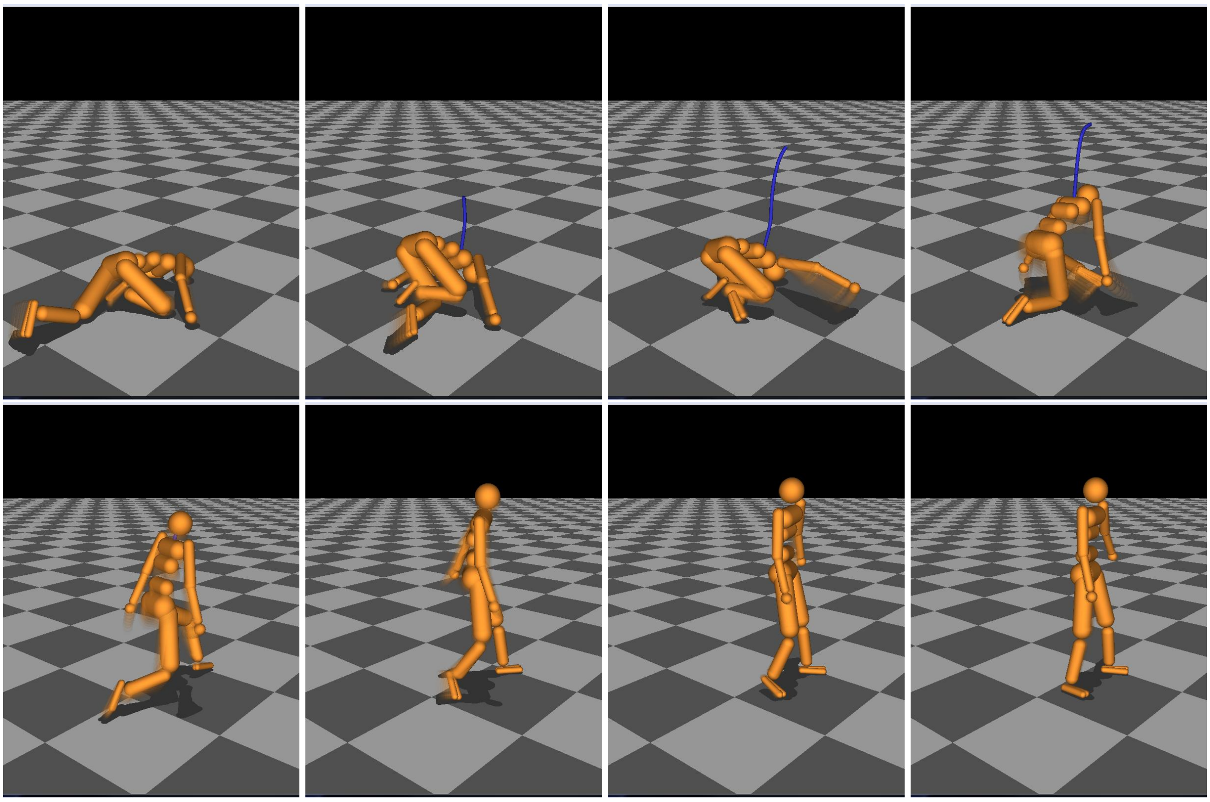


Figure 2: Getting up from a different initial pose. This sequence is shown at 3m19s in the movie.