Model-Based Trajectory Planning through Generic Nonlinear Programming

Yuanwen Tian[1], Zehao Xu[2], Haifeng Wang[3], Prof. Tao Gao[4]

[1] Huazhong University of Science and Technology
[2] Zhejiang University
[3] Tsinghua University
[4] Center for Vision, Cognition, Learning and Autonomy, University of California, Los Angeles

University of California, Los Angeles

HUAZHONG UNIVERSITY OF SCIENCE & TECHNOLOGY

Summary

Question: How to build a trajectory optimization program that can be adaptive to various system models?

Task: Our task is to use generic nonlinear programming for trajectory planning while satisfying random-target constraint and realizing contact-free motion.

Results: We test our program in random-target and contact free tasks. We also use different types of models to realize trajectory optimization. The results are shown in Figure 2-4.

Background

Trajectory optimization means finding the best trajectory by selecting the inputs to the system, mostly controls. In our research, we define the optimal trajectory as the one that minimizes the square of controls.

Trajectory optimization plays a significant role in robotic motion planning. A trajectory that contains collisions or violates constraints can be optimized into a high-quality trajectory satisfying all constraints [1].

When solving trajectory optimization problems, traditional simulation methods require the specific system dynamic function. Therefore, the problem lies in that the program needs reconstructing when the observed system varies.

Methods

We use MuJoCo [2] to derive physical dynamic of the system, then we input the parameters to Ipopt [3] to obtain the optimal trajectory.

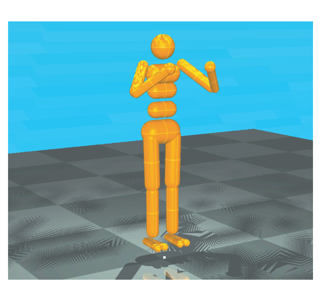


Figure 1. A model example from MuJoCo physics engine. This is a humanoid model with 27 degree of freedom. This model is used to simulate human motion by OpenAI [4].

Workflow:

MuJoCo dynamics -> trajectory optimization problem -> Ipopt solver

$$J = \int_0^1 u^2(\tau)\, d\tau \qquad cost\ function$$

$$\begin{aligned} \min_z \quad & J(z) & \\ s.t. \quad & f(z) = 0, & system\ dynamics \\ & g(z) \le 0, , & constraint \\ & z_{low} \le z \le z_{upp} & path\ bound \end{aligned}$$

Spotlights

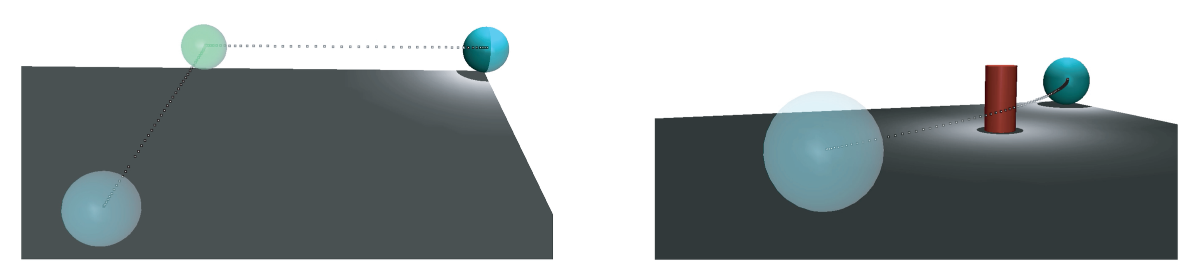
There are many differences in our generic trajectory planning program from traditional ones, we summarize as follows:

- Generic Framework: Our program is independent of any specific system, it can be adaptive to different types of complex models.
- Complex Task: Our program can satisfy random-target constraint and realize contact-free motion.
- Improve Overall Performance: Aside from above, our future goal is to achieve effcient large-scale trajectory optimization.

Results

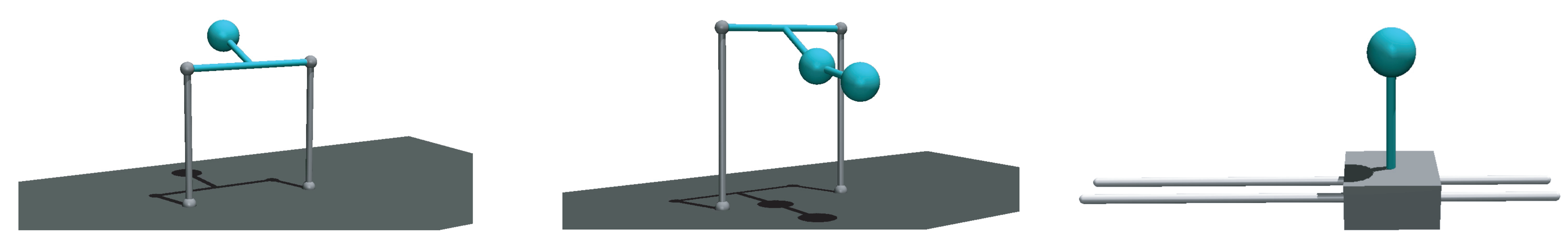
The results can be categorized into two branches: first is the progress in mutiple task scenarios shown in Figure 2; and second is the progress in model extension shown in Figure 3. Figure 4 shows the detailed state and control information of a cart-pole example.

Figure 2. Trajectory optimization of a ball moving in a plane. (light green sphere is the target and red cylinder is the obstacle.)



(a) a random-target task (b) a contact-free task

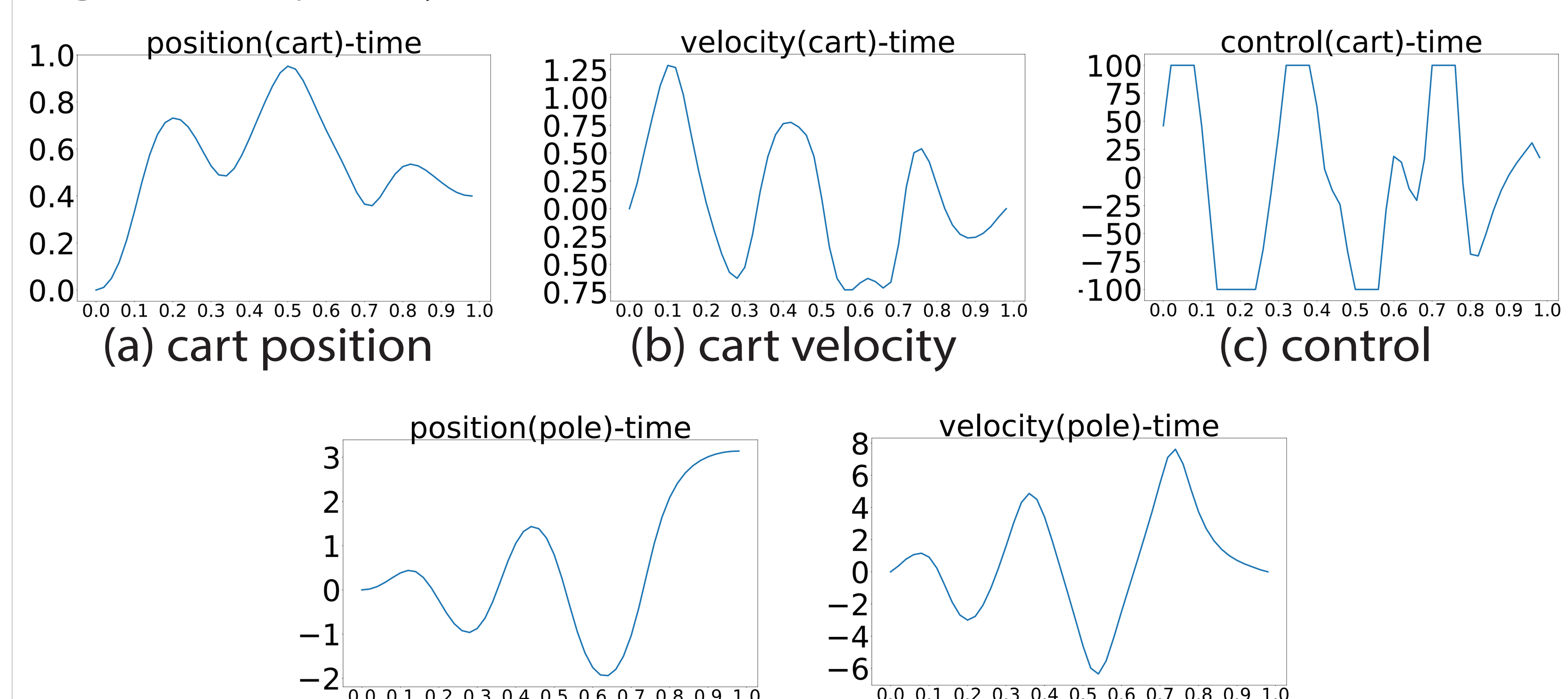
Figure 3. Different models can achieve optimal trajectory.



(a) (b) (c)

(a) inverted pendulum(one rotational joint);
(b) double inverted pendulum(two rotational joint);
(c) cart-pole(one translational joint and one rotational joint).

Figure 4. Trajectory data visualization of the cart-pole example.



(a) cart position (b) cart velocity (c) control
(d) pole position (e)pole velocity

Acknowledgements

This research is collaborated by Yuanwen Tian, Zehao Xu, and Haifeng Wang. We would like to show our gratitude to our mentor, Prof. Tao Gao, for his insight, expertise, and patient instructions. Thanks for UCLA-CSST for providing professional help for me to enjoy this scientific journey.

References

[1] J. Schulman et al., "Finding locally optimal, collision-free trajectories with sequential convex optimization," in RSS, 2013.
[2] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: a physics engine for model-based control," in Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012.
[3] A. Wächter and L. T. Biegler, "On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming," Mathematical Programming 106(1), pp. 25-57, 2006.
[4] G. Brockman et al., "Openai gym," arXiv:1606.01540, 2016.