Model Predictive Control of Acrobot in Simulated Environment

Qingmu Deng

qingmu@stanford.edu

May 4, 2024

1 Proposal

Model predictive control, also known as online trajectory optimization, has been a powerful method in Controls and increasing gaining popularity in real-time application with the ever faster computing capabilities in robotic systems. For the final project for AA203 Optimal and Learning Based Control at Stanford, I propose to implement model predictive control for a simulated acrobot in a simulated environment. An example of the said acrobot and control can be seen in the first minute of [1].

Through the final project, my goal is to develop solid theoretical understanding and gain practical implementation experience in Python for model predictive control. I would also love to learn to use some of the simulated environments for developing and evaluating control policies as sometimes it may be expensive or impractical for an individual learner such as myself to acquire all the physical hardware for a humanoid robot or a drone and repair any damages incurred during the control experiments.

The project idea came from a video named "Model Predictive Control with iLQG and MuJoCo" by Tassa, Erez, and Todorov.[1] It is accompanied by their paper that detailed the theoretical underpinning of model predictive control and the practical considerations of the dynamic simulation and control particularly in MuJuCo.[2] MuJuCo is "a general purpose physics engine that aims to facilitate research and development in robotics, biomechanics, graphics and animation, machine learning, and other areas which demand fast and accurate simulation of articulated structures interacting with their environment." [3] Another popular simulation framework that I am considering for model building and control in this project is PyBullet. [4]

I am particularly looking for feedbacks on

- the choice of the modeling environment between PyBullet and MujuCo. Since I don't have experience in either, I appreciate some feedbacks on the usefulness and ease of entry of either framework. Since we are heavily focused on controls, it is natural that not too much time should be spent on building the model as opposed to controlling the system.
- the scope of the project. I have decided that working by myself lends me the most flexibility both in time management as a full-time professional and a part-time student and in pursuing my intellectual interests given my limited time for study. My goals are outlined as above, and I hope my project will at a minimum help me achieve those goals. However, if your experience tells you that what I've proposed can be easily achieved, I am all ears for hearing from you in terms of possible extensions to the basic proposal.

2 Related Work

3 Problem Statement

4 Approach

5 Experiments

6 Conclusions and Future Work

References

[1] Y. Tassa, T. Erez, and E. Todorov, "Model predictive control with ilqg and mujoco," YouTube, May 2019.

[2] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4906–4913, 2012.

[3] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, IEEE, 2012.

[4] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning." `http://pybullet.org`, 2016–2019.