Assignment 08: Pendulum Swing-Up

Optimal Control for Robotics

Assigned: March 16 — Due: March 28 at 11:55pm

Introduction

In this assignment you will compute the optimal swing-up trajectory for a simple pendulum using multiple shooting. We will keep things simple: use Euler's method on a uniform time grid, with one simulation step per multiple shooting segment.

Deliverables

Implement the function `simplePendulumOptimBvp` using the template provided.

Write-Up:

There is separate write-up for this assignment. Instead, please include the total time you spent working on this assignment in the comments near the top of your implementation of `simplePendulumOptimBvp`. Be sure to clearly organize and document your code.

Comments

In this assignment I provide a simple script to run your optimization, along with an encrypted version of the solution. You should run this script with a variety of different parameters to learn more about how trajectory optimization behaves with different inputs. Once you've done that, then start working on implementing your code.

If you run out of time and your assignment does not work correctly, then add a notes section to the comments in the top of `simplePendulumOptimBvp`. You should clearly state that the function does not work, why you think it does not work, and your best guess about how to go about debugging it if you had time.

The file `simplePendulumOptimBvpSoln.p` provides you with a correct implementation of multiple shooting, against which you can compare your code. This implementation is not guarenteed to obtain the "true" solution to the trajectory optimization problem. If FMINCON converges, then the resulting trajectory is an approximation of a locally optimal trajectory.

One final note: `simplePendulumOptimBvpSoln.p` uses a very simple initialization routine. If you are clever you can probably beat it: either by finding the same solution more quickly or by avoiding a local minima. For example, the optimal solution might require three swings, but `simplePendulumOptimBvpSoln.p` only manages to find a solution with one swing (which has a higher objective function value).