COMP 417 Assignment 2

Due: Oct 30, 11:59pm

1 Objectives

The main pedagogical objective of this assignment is to provide and demonstrate an understanding of PID control, the most popular yet simple control methodology, as well as the notion of the inverted pendulum in the specific form known as a ”cart pole”.

The inverted pendulum is a benchmark in the context of both waling robots as well as control, where the goal is to balance the pole attached to a cart through an unactuated (i.e. unpowered) joint, meaning that we need to control the pole angle by manipulating the cart displacement. Although PID is indeed a simple controller, it is very widely used due to its ease of implementation, flexibility and also robustness to noise, and uncertainty. Thus, it is imperative to get insight into how it works.

The objective is to move the cart (platform) back and forth to keep the pole balanced and as close to vertical as possible. The controller itself can use the angle of the pole, and the differential properties of this angle, to compute an adjustment to the cart's position.

Note, you are given a base code including the inverted pendulum dynamic model and also visualization to better evaluate the controller performance. Modify the controller script while feeding the dynamic model with proper arguments at each part.

You can run the sample code using the command

```
python inverted_pendulum.py
```

and see some of the options using the invocation

```
python inverted_pendulum.py --help
```

Assuming that we have access to the states of the system i.e. the angle to be controlled:

(A) Design a PID controller to stabilize the angle (keep the pole balanced and as close to vertical as possible). (30%)

(B) Evaluate and attempt to tune the performance of P-control, PD, PI, DI. (Not all of these may work well; explain why.) (30%)

(C) When designing a controller, system model dependency is crucial and reflects controller robustness. Investigate PID controller behavior by varying pendulum mass and gravity force parameters. Observe their impact on controller performance, such as settling time. (20%)

(D) The other critical aspect of a controller is its ability in rejecting disturbances. For this part, you need to modify the code to model the injection of a random input to the system for one or two times in a trial. For instance, you can use something like: (20%)
$if numpy.random.rand() > 0.99 :$
$action = 10$
or you can use the keyboard inputs while controlling the system with the PID controller. The second approach needs a bit of modification on the main script (not suggested), while the former one can be implemented in the controller script.

2 Things to submit

You will need to submit both your code and a report(in pdf format please) zipped together. Please be concise and report interesting things like challenges. At each step report a plot of theta vs time showing your controller performance in a conducted simulation with a brief explanation. Asked to discuss things, give us a summary in two or three lines of what you got.