EE 554 Homework Assignment 4
October 6, 2009

Build yourself a control system simulator based upon a numerical ordinary differential equation solver (e.g., *ode23()* in Matlab) following the approach discussed in class. You'll want the option of performing dynamic control in continuous-time with the controller added to the system's differential equations or as a discrete-time controller that updates at specified intervals of time while the system continuously evolves.

1. Build a simulation for the pendulum using the nonlinear model and parameters used in the m-files handed out (also find them on the class web page), and test that it behaves as expected for a constant control input (zero and less than gravitational torque) and variety of initial conditions. Plot the angular position and velocity versus time, and turn in an example.

2. Build a simulation for the magnetic-ball system using the nonlinear model and parameters from the first homework, and confirm it will levitate at 0.5m by starting it there with exactly the right (constant) input. Test that it will fall if the input is less than that necessary for levitation and move up if the input is greater than that necessary for levitation. Plot the position, velocity and current versus time, and turn in an example.

3. Download the simple visualization/animation m-files *drawpendulum()* and *drawmagneticball()* from the course web page and use them to view the systems' responses.

4. Implement continuous- and discrete-time PID controllers (your choice of numerical differentiation and integration) for both the pendulum and magnetic-ball.

   (a) Choose position as the single output of both.

   (b) Add the input u_0 needed for starting at equilibrium to the PID controllers (they will have four terms now - $P - I - D - u_0$).

   (c) Set the reference point to somewhere near the starting equilibrium, say something around 0.25rad away for the pendulum and 0.02m away for the magnetic ball.

   (d) Experimentally tune PID controller by setting $k_P = k_D = k_I = 0$ (should not move due to u_0); increase k_P until a small amount of overshoot ($\approx 10\%$); increase k_D until overshoot goes away; and increase k_I until there is once again a small amount of overshoot.

   (e) Compare the continuous- and discrete-time implementations. What is the steady-state error, settling time and overshoot? What is the effect of sampling time in the discrete-time case? What sampling time is a good balance between efficiency and performance? How does noise affect your performance?

   (f) Vary the reference point with a sinusoid of amplitude the same as your change in the reference point above. How well does your PID controller perform tracking?