Control of Mechanical engineering systems

University of Florida
Mechanical and Aerospace Engineering

Mini project 1
Issued: March 5, 2009
Due: March 23
Max score (with extra credit) = 125
(a score of 100 contributes 5% to your total course grade).

Instructions and guidelines: You are allowed to discuss the project with others, but the work you turn in must be your own. Points will be awarded for correctness, completeness and clarity. If you make a mistake at one stage that makes your subsequent answers wrong, you will not get any credit for getting the "steps right". So check and re-check your work carefully. When you provide plots, describe clearly what you are plotting and label your axes.

Please submit your derivations, MATLAB code, and plots in paper format. Submit all your simulink blocks (along with any MATLAB code needed to run them) in electronic format as a .zip archive by email to pbarooah@ufl.edu. Name your .zip file as your LastNameUFID.zip. For example, if your last name is Smith and your UFID is 1234-5678, then name it as Smith12345678.zip

Problem 1. The following ODE describes the dependence on the speed v (in m/s) of a vehicle in a highway to the throttle angle θ (in Volt):

$$m\dot{v}(t) = b\theta(t) - av(t)^2. \tag{1}$$

where m is the mass of the vehicle (in Kg), b is the gain from the throttle angle to force (in $N/Volt$), and a can be thought of as a drag coefficient (in Ns^2/m^2). Note that the sensor that is used to measure the throttle angle produces and output in Volt, which is why in this model the unit of measuring throttle angle is Volt, not degree or radians. A 1 Volt output of the sensor corresponds to idling position of the throttle and a 4 Volt output of the sensor corresponds to the throttle being fully open. The values of the parameters are $m = 1300kg$, $a = 150.91Ns^2/m^2$, and $b = 42926.2\ N/Volt$.

A simulink model `car_nonlinear.mdl` is provided in the course website that you can use to simulate this non-linear system (1) with step commands for the throttle angle. The MATLAB script `init_model.m` (also in the course website) defines the variables needed to run the simulink model. By default, with variables defined in `init_model.m`, the model `car_nonlinear.mdl` simulates the response of the car (from time $t = 0$ to $t = 20$ sec, starting from zero speed: $v(0) = 0$) to a step input of 1.5 Volt at time $t = 0$ and then another of 2 Volt at time $t = 10$.

1. Determine the throttle angle θ_0 required to keep the vehicle moving at a constant speed of 60 mph. After assigning appropriate values of the parameters θ_1 and θ_2 in MATLAB workspace, run the simulink model `car_nonlinear.mdl` to determine the response of the non-linear system (1) starting from $v = 0$ at time $t = 0$, when the throttle command provided by the driver is the following:

$$\theta(t) = \begin{cases} \theta_0 & 0 \le t < 10 \\ \theta_0 + 0.3 & 10 \le t \end{cases} \tag{2}$$

simulate the response of the *plant* (i.e., the car) for the first 20 seconds, and plot both $\theta(t)$ and $v(t)$ in a MATLAB figure (do not provide printouts of simulink scopes; save the variables to MATLAB workspace and plot in MATLAB).

2. Linearize the ODE (1) around the operating point of $v_0 = 60$ mph (don't forget to convert it to m/s) and derive a linear model of the form $\dot{y}(t) + py(t) = ku(t)$, where $y = v(t) - v_0$ is the deviation of the velocity from the nominal and $u(t) = \theta(t) - \theta_0$ is the change in the throttle angle from the nominal value θ_0. For this linear model, $u(t)$ will server as the *control signal* and $y(t)$ as the *output signal*. Provide the numerical values of p and k (note that they depend only on m, b, a, and v_0).

   For future reference, let us decide here that the linearized model will be considered "accurate enough" as long as $|\frac{y}{v_0}| \leq 0.1$.

3. Write a MATLAB script to determine the response of the linear plant model to a step input of size 0.3 Volt (i.e., $u(t) = 0.3 \times 1(t)$) at time $t = 10$ sec. Compare the velocity predictions from the linearized and the true non-linear model for the same step input, by plotting them against time in the same figure. (important: remember that the output of the linear model, $y(t)$ is not the speed but its deviation from the nominal speed v_0.)

4. We want to develop a closed-loop cruise control system so that the output $y(t)$ is maintained (as closely as possible) to the reference command $r(t)$ provided by the driver. Assume there are no disturbances or sensor noise. Design a PI control law $C(s) = k_p + k_i/s$ by choosing the gains k_p and k_i so that (i) steady state error to step reference commands is 0, and (ii) the closed loop transfer function from $R(s)$ to $Y(s)$ has a rise time of 3 seconds.

5. Write a MATLAB script to do the following: (i) construct the closed loop system (the `series` and `feedback` commands are useful), (ii) plot the step responses (for 10 seconds) of the closed loop transfer function to steps with the following sizes: $0.2, 0.5, 0.8$ and 1 (in a single figure), (ii) plot the control signals $u(t)$ (in one figure) and the corresponding throttle angles $\theta(t)$ (in another figure).

6. Use the `step` command in MATLAB By varying the size of the steps and examining the resulting responses, determine how large a step command in the reference can the closed loop handle without the system getting out of the linear regime? Show your plots in a figure to justify your answer.

7. **Extra credit:** Now implement your linear control law in the non-linear system. To do that, modify the simulink model provided to compute the control signal $u(t) = k_p e(t) + k_i \int_0^t e(\tau)d\tau$, where $e(t) = r(t) - y(t)$. Note that to generate the throttle command $\theta(t)$ for the non-linear system, you have to add $u(t)$ to θ_0. Similarly, pay attention that the reference command is for the signal $y(t)$, not for the speed $v(t)$.

   Simulate the response of the true non-linear system with PI control when subjected to a *total speed reference* command $r'(t)$ (where $r'(t) = r(t) + v_0$):

$$r'(t) = \begin{cases} 60\ mph & 0 \leq t < 10 \\ 65\ mph & 10 \leq t \end{cases}, \tag{3}$$

   and plot it in a MATLAB figure.