ECEN 4413
Automatic Control Systems
Spring 2005
Computer Project- Part B

Objective:

In the second part of project your goal is to design a controller for a nonlinear system. To accomplish this, you will first linearize the system model and then use the tools you have used in Part A of this project to design a linear controller. Finally, you will analyze the applicability of your controller to the actual nonlinear system using Simulink.

Requirement:

Show all your steps, plots (responses) and clearly state your comments and explanations if required. Include all the MATLAB code lines and programs (with documentation).

Stage 1: Linearization and Simulation

Below is the dynamic equation of a nonlinear system.

$$\ddot{y} + 0.75\ddot{y} + 0.27\dot{y}\cos(y) + (0.027 + y\sin(y))y = 0.1x$$

a. Use Simulink to create a simulation model of the <u>nonlinear</u> system. Generate a print out of the model (use Edit → Copy Model to Clipboard).

b. In this hypothetical application we are interested only in small variations around the origin for the output y. Knowing this, your next step is to generate a linear approximation for the system's dynamic equation. Recall that for small values of y:
$$\cos(y) \cong 1 \quad \text{and} \quad \sin(y) \cong 0$$

c. Create a model for the <u>linearized</u> system in Simulink in the same .mdl file you did the nonlinear system. Generate a print out of the model.

d. Plot and compare with words the outputs y of the <u>nonlinear</u> and <u>linearized</u> systems for the following step inputs:
$$x(t) = 0.001u(t)$$
$$x(t) = 0.01u(t)$$
$$x(t) = 0.1u(t)$$
$$x(t) = 0.3u(t)$$

Attention: Adjust all your simulation for a 100s duration. $u(t)$ is the unit step function. For the first input, the outputs of the nonlinear and linearized system should be almost identical. Make sure you get this result before proceeding to the next stage.

Stage 2: Controller Design

e. Using **ltiview**, obtain for the <u>linearized</u> system its settling time, overshoot percentage and steady state error in response to a step input.

f. Using **sisotool**, design for the <u>linearized</u> system a controller capable of achieving:
- Zero steady state error;
- Settling time less or equal to 30s;
- Overshoot less or equal to 15%.

Attention: Follow the same procedure you used in Part A of this project. You should not require more than two zeros and two poles for this controller. Make sure your controller does not have more zeros than poles.

Stage 3: Implementation

g. Return to the Simulink .mdl file that contains both your <u>nonlinear</u> and <u>linearized</u> systems and add two copies of the controller you designed, one for each system. Don't forget to close the **negative** feedback loop and add a *source block* for your reference $\hat{y}$.

h. Plot and compare with words the outputs y of the controlled <u>nonlinear</u> and <u>linearized</u> systems for the following reference signals:

$$\hat{y}(t) = 0.001u(t)$$

$$\hat{y}(t) = 0.01u(t)$$

$$\hat{y}(t) = 0.1u(t)$$

$$\hat{y}(t) = 0.3u(t)$$

i. If this was a real world situation, what limit would you place in the reference $\hat{y}$ that the user can adjust in order to guarantee that the design parameters in item **f** would not be violated when the <u>nonlinear</u> system is used?