EE5327, Fall 2016
Homework 6: Fuzzy Systems
Due 11/28/2017

As you did for Homework 5, use the same model and data for the first problem. Recall you implemented the following system in Simulink and assumed a damping ratio (zeta) of 0.7, a fixed step integration with a time step of 0.01 sec, and a unit step input occurring at 1 second, with a simulation stop time of 15 seconds. Generate a training dataset of (time, input, omega) as inputs and y as the output for values of omega 0.5, 1.0, 1.5, 2.0, 2.5, and 3.0 rad/sec.

$$y(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} u(s)$$

Problem 1) Fuzzy Inference System Estimator (60 points)

a) Use the neuroFuzzyDesigner tool to fit a few fuzzy inference systems. First, let's see which one has the best fit to the training data. Try the following making note of the final error after 3 epochs of training (to save time):
    1) 3-3-3 with *triangular* membership functions and *linear* output.
    2) 5-5-5 with *triangular* membership functions and *linear* output
    3) 3-3-3 with *generalized bell* membership functions and *linear* output.
    4) 5-5-5 with *generalized bell* membership functions and *linear* output.

b) Export each FIS model to the workspace and include in the Simulink simulation using a Fuzzy Logic Controller block. Generate comparisons of the FIS model with the training data. Make any observations about the type of FIS and the quality of fit it produces.

c) Now test the FIS by comparing its output with the true system output for omega values 0.75, 1.25, 1.75, 2.25, and 2.75 rad/sec. Make any observations about the type of FIS and the quality of fit it produces for this test data.

d) Finally, comment on the overall ability of the FIS to work or not work for this application.

Problem 2) Fuzzy Inference System as a Predictor (40 points)

MathWorks packs the Mackey-Glass time series data as an example in their Fuzzy Logic Toolbox. They provide an excellent example of using the ANFIS as a step-ahead predictor using MATLAB commands. This problem requires you use their data and approach, but using the neuroFuzzyDesigner tool to design and test the ANFIS, then export it back to the workspace for completing the analysis.

Load and preprocess the data as they did in the example:

```
load mgdata.dat
a = mgdata;
time = a(:, 1);
x_t = a(:, 2);
plot(time, x_t);
xlabel('Time (sec)','fontsize',10); ylabel('x(t)','fontsize',10);
```

```
title('Mackey-Glass Chaotic Time Series','fontsize',10);
trn_data = zeros(500, 5);
chk_data = zeros(500, 5);

% prepare training data
trn_data(:, 1) = x_t(101:600);
trn_data(:, 2) = x_t(107:606);
trn_data(:, 3) = x_t(113:612);
trn_data(:, 4) = x_t(119:618);
trn_data(:, 5) = x_t(125:624);

% prepare checking data
chk_data(:, 1) = x_t(601:1100);
chk_data(:, 2) = x_t(607:1106);
chk_data(:, 3) = x_t(613:1112);
chk_data(:, 4) = x_t(619:1118);
chk_data(:, 5) = x_t(625:1124);
index = 119:1118; % ts starts with t = 0
plot(time(index), x_t(index));
xlabel('Time (sec)','fontsize',10); ylabel('x(t)','fontsize',10);
title('Mackey-Glass Chaotic Time Series','fontsize',10);
```

But instead of using their command-line FIS generation as they used in the help example, load the training and test (checking) data into the neuroFuzzyDesigner tool.

a) Train the first FIS with 2 *generalized bell* functions per input and the *constant* output membership function for three epochs (the default). Send the trained 2-2-2-2 FIS back to MATLAB and compute the sum squared error between the test (chk) data and the FIS output (see the example in help).
b) To see if it matters, repeat part a) but use the *linear* output membership function. Take this FIS back to MATLAB and compute the sum squared error of the test (chk) data and compare to part a). Which is better?
c) To see if the input membership functions matter, train another FIS with 2 *triangular* functions per input for three epochs with a *linear* output membership function. Compare the sum squared error of the test (chk) data and compare to part b). Which FIS is better?