ADVANCED UNDERGRADUATE TOPICS IN CONTROL SYSTEMS DESIGN: HOMEWORK #2

Exercise 3 (Known parameters)**.** The data provided was obtained from a system with one zero and two poles transfer function

$$H(s) = \frac{k(s+.5)}{(s+.3)(s-p)},$$

where the gain k and the pole p are unknown.

1. Estimate the system's transfer function *without making use of the fact that you know the location of the zero and one of the poles.*

2. Estimate the system's transfer function using the fact that you know that the zero is at $s = -.5$ and one of the poles at $s = -.3$.

   Hint: To solve this problem you will need to combine the ideas from Sections 4.5.1 and 4.5.2, by identifying the transfer function from an auxiliar input $\bar{u}$ to an auxiliar output $\bar{y}$. □

If all went well, you should have gotten somewhat similar bode plots for the transfer functions obtained in 1 and 2, but you should have obtained a much better estimate of the poles and zeros in 2.

Exercise 4 (Model order)**.** The data provided was obtained from a continuous-time linear system whose transfer functions has an unknown number of poles and zeros.

Use `tfest` to estimate the system's transfer function for different numbers of poles and zeros ranging from 1 through 5 poles. For the different transfer functions identified,

1. compute the mean square error (MSE) normalized by the sum of squares of the output,

2. compute the largest value of the parameter standard deviation normalized by the corresponding parameter value,

3. plot the location of the transfer functions' poles and zeros in the complex plane.

These two values and plot can be obtained using

```
model=tfest(data,npoles,nzeros);
% compute normalized mean-square error
normalizedMSE=sum(report.Fit.MSE)/(y'*y);
% extract from report the numerator and denominator coefficients
num=report.parameters.ParVector(1:nzeros+1)';
den=report.parameters.ParVector(nzeros+2:nzeros+npoles+1)';
% extract from report corresponding error standard deviations
std_num=sqrt(diag(...
   report.parameters.FreeParCovariance(1:nzeros+1,1:nzeros+1)))';
std_den=sqrt(diag(...
   report.parameters.FreeParCovariance(nzeros+2:nzeros+npoles+1,...
                                       nzeros+2:nzeros+npoles+1)))';
% compute normalized error standard deviations
maxStdDev=max([std_num,std_den]./[num,den]);
% plot root locus
rlocus(model);
```

Use this information to select the best values for the number of poles and zeros and provide the corresponding transfer function. Justify your choices.

Important: Write MATLAB® scripts to automate these procedures. You will need them for the lab.

□

Exercise 5 (Input magnitude)**.** A Simulink block that models a nonlinear spring-mass system is provided. This model expects the following variables to be defined:

```
Ts = 0.1;
tfinal = 100;
```

You will also need to define the magnitude of the step input and the measurement noise variance through the variables

```
step_mag
noise
```

Once all these variables have been set, you can run a simulation using

```
sim('spring',tfinal)
```

after which the variables `t`, `u`, and `y` are created with time, the control input, and the measured output.

1. Use the Simulink block to generate the system's response to step inputs with amplitude 0.1 and 2.0 and no measurement noise.

   For each of these inputs, use `tfest` to estimate the system's transfer function for different numbers of poles and zeros ranging from 1 through 3 poles.

   For the different transfer functions identified,

   (a) compute the mean square error (MSE) normalized by the sum of squares of the output,

   (b) compute the largest value of the parameter standard deviation normalized by the corresponding parameter value,

   (c) plot the transfer functions poles and zeros in the complex plane.

   These values can be obtained using

   ```
   model=tfest(data,npoles,nzeros);
   % compute normalized mean-square error
   normalizedMSE=sum(report.Fit.MSE)/(y'*y);
   % extract from report the numerator and denominator coefficients
   num=report.parameters.ParVector(1:nzeros+1)';
   den=report.parameters.ParVector(nzeros+2:nzeros+npoles+1)';
   % extract from report corresponding error standard deviations
   std_num=sqrt(diag(...
      report.parameters.FreeParCovariance(1:nzeros+1,1:nzeros+1)))';
   std_den=sqrt(diag(...
      report.parameters.FreeParCovariance(nzeros+2:nzeros+npoles+1,...
                                          nzeros+2:nzeros+npoles+1)))';
   % compute normalized error standard deviations
   maxStdDev=max([std_num,std_den]./[num,den]);
   % plot root locus
   rlocus(model);
   ```

   Use this information to select the best values for the number of poles and zeros and provide the corresponding transfer function. Justify your choices.

2. Use the Simulink block to generate the system's response to several step inputs with magnitudes in the range [0.1,2.0] and measurement noise with variance 10^{-3}.

   For the best values for the number of poles and zeros determined above, plot the normalized MSE as a function of the magnitude of the step input. Which magnitude leads to the best model?

Important: Write MATLAB® scripts to automate these procedures. You will need them for the lab.

□