EGB345 Control and Dynamic Systems JJF/20

**Servo Motor System Identification Report**

Instructions: replace the yellow highlighted text with your own words and the requested plots (that is, delete the yellow text). Note the submitted must be saved from MATLAB (not screen captures).

Authorship details

|  |  |
| --- | --- |
| Author | **Chiran Gayanga Walisundara****(N10454012)** |
| List others providing assistance | *-* |

Text and plot answers

|  |  |
| --- | --- |
| Question 1 [Text] |  |
| Question 1 [Plot] | A close up of a map  Description automatically generated |
| Question 2 [Text] |  |
| Question 2 [Plot] | A close up of a map  Description automatically generated |
| Question 2  [MATLAB] | function [K\_est, alpha\_est] = estmotor(t, ydata)    % simulating the motor's shaft angle  % 't' contains the shifted and truncated time vector  % 'ydata' contains the step response data to be compared against  % i have used 3 - double for loops each searching for a different significant figure  % approach based on exhaustive search using for loops and comparing root mean squared error  % by narrowing down for loop ranges through interpolation we arrive at our estimated km and alpha values    % starting root mean squared value of error  old\_RMSE = 1e100;    % 0th decimal place  % estimation for 1st significant figure  % iterating over 2 for loops with step size of 1 to estimate km and alpha  for km\_1sigfig = 0 : 1 : 10  for alpha\_1sigfig = 0 : 1 : 10    % setting up test motor transfer function  % getting numerator and denominator and denominator  % calculating transfer function using 'tf' command  numeratorTest = [ km\_1sigfig ];  denominatorTest = [ 1 alpha\_1sigfig 0 ];  motor\_TF = tf(numeratorTest, denominatorTest);    % test motor system response to step input to get step response  motor\_SR = step(motor\_TF, t);    % error of current motor data  data\_error = ydata - motor\_SR;    % calculating the root mean squared value of the error  RMSE = rms(data\_error);    % if the rmse of the current data  % is less than the rmse of the previous data  % storing the current data in variables  % as numerator and denominator  if RMSE < old\_RMSE    num\_proper = [ km\_1sigfig ]; % numerator  den\_proper = [ 1 alpha\_1sigfig 0 ]; % denominator  old\_RMSE = RMSE;    end  end  end    % 1st decimal place  % estimation for 2nd significant figure  % interpolating data with a new range and decreased step size  % iterating over 2 for loops with step size of 0.1 to estimate km and alpha  for km\_2sigfigs = num\_proper(:) - 2 : 0.1 : num\_proper(:) + 2  for alpha\_2sigfigs = den\_proper(:,2) - 2 : 0.1 : den\_proper(:,2) + 2    % setting up motor test transfer function  numeratorTest = [ km\_2sigfigs ];  denominatorTest = [ 1 alpha\_2sigfigs 0 ];  motor\_TF = tf(numeratorTest, denominatorTest);    % test motor system response to step input to get step response  motor\_SR = step(motor\_TF, t);    % error of current motor data  data\_error = ydata - motor\_SR;    % calculating the root mean squared value of the error  RMSE = rms(data\_error);    % if the rmse of the current data  % is less than the rmse of the previous data  % storing the current data in variables  % as numerator and denominator  if RMSE < old\_RMSE    num\_proper = [ km\_2sigfigs ]; % numerator  den\_proper = [ 1 alpha\_2sigfigs 0 ]; % denominator  old\_RMSE = RMSE;    end  end  end    % 2nd decimal place  % estimation for 3rd significant figure  % interpolating data with a new range and decreased step size  % iterating over 2 for loops with step size of 0.01 to estimate km and alpha  for km\_3sigfigs = num\_proper(:) - 1 : 0.01 : num\_proper(:) + 1  for alpha\_3sigfigs = den\_proper(:,2) - 1 : 0.01 : den\_proper(:,2) + 1    % setting up motor test transfer function  numeratorTest = [ km\_3sigfigs ];  denominatorTest = [ 1 alpha\_3sigfigs 0 ];  motor\_TF = tf(numeratorTest, denominatorTest);    % test motor system response to step input to get step response  motor\_SR = step(motor\_TF, t);    % error of current motor data  data\_error = ydata - motor\_SR;    % calculating the root mean squared value of the error  RMSE = rms(data\_error);    % if the rmse of the current data  % is less than the rmse of the previous data  % storing the current data in variables  % as numerator and denominator  if RMSE < old\_RMSE    num\_proper = [ km\_3sigfigs ]; % numerator  den\_proper = [ 1 alpha\_3sigfigs 0 ]; % denominator  old\_RMSE = RMSE;    end  end  end    % final root mean squared error of the estimation  disp('final rmse is');  disp(old\_RMSE);    % numerator and denominator now contain  % the final estimates with 3 significant figures for variables km and alpha  % estimated km and alpha values for the motor's shaft angle as outputs are stored in K\_est and alpha\_est  K\_est = num\_proper; % final km value  alpha\_est = den\_proper(:,2); % final alpha value    end |
| Question 3 [Text] |  |
| Question 3 [Plot] | A map of a person  Description automatically generated |
| Question 3 [Text] | I think this estimation has the very decent accuracy as the estimation for the true values as both the known simulation and the unknown simulation produced extremely close root mean squared error (RMSE) values of 0.0174 and 0.0182 respectively. Therefore, I think my km value has an error of approximately and my alpha value has an error of approximately for the unknown model parameters that generated the data. |