SERVO MOTOR SYSTEM IDENTIFICATION TASK

Due Date Friday week 8. Worth 10%

Must use/submit MATLAB script files .m files (not MATLAB live scripts).

In this assessment, we will connect material covered in lectures, tutorials and practicals. You will develop fundamental skills in control system design and synthesis by implementing your first control system. Specifically, you will implement a custom-built motor control system.

1 Background: Servo motors are everywhere.

Many modern machines require small servos or motors to operate. For example, consider a small unmanned aircraft (and its elevator actuator). If we desire a constant angle of deflection in the elevator (to pitch the aircraft upwards) then the elevator's deflection is typically controlled using a servo motor that holds the elevator at a constant angle. However, this is a little complicated to achieve because when a constant voltage is applied to the servo motor, the motor's shaft turns with constant angular velocity (i.e. **does not hold** at a constant angle, as might be desired). Luckily, using feedback control ideas, we can create a position error feedback system for the servo motor that **does hold** the servo motor's shaft angle at a constant value.

The importance of understanding servo motors is not limited to small robotic aircraft. Some other examples of servo motor position control systems include:

- Rudder or wheel position control surface on your remote-control aircraft model, boat, or car.
- Engine throttle in a motor vehicle.
- Just about everywhere in a modern aircraft (fly-by wire).
- Robotics (wherever there is motion of arms etc.).
- DVD or Blue-ray disc players.
- And many others.

In summary, we typically seek utility or autonomy from our servo motors in the sense that we require the motor shaft position to be controlled or regulated to some desired angle.

2 Modelling the Servo motor

This system is based around a servo motor mechanism commonly used in remote control aircraft. Simply speaking, the servo system has a motor, arm, potentiometer and a set of internal electronics. The motor is used to induce rotary (or linear) motion of the arm, which, in turn, alters the displacement of the potentiometer wiper. So, the potentiometer acts like a sensor, used to indicate arm displacement in the form of an output voltage $v_p(t)$.

Transfer function modelling:

The output shaft is connected directly to a potentiometer such that the output voltage indicates the shaft angle position (this provides a signal for position feedback). This means that the potentiometer measures shaft angle position. The simplified transfer function can be approximated by:

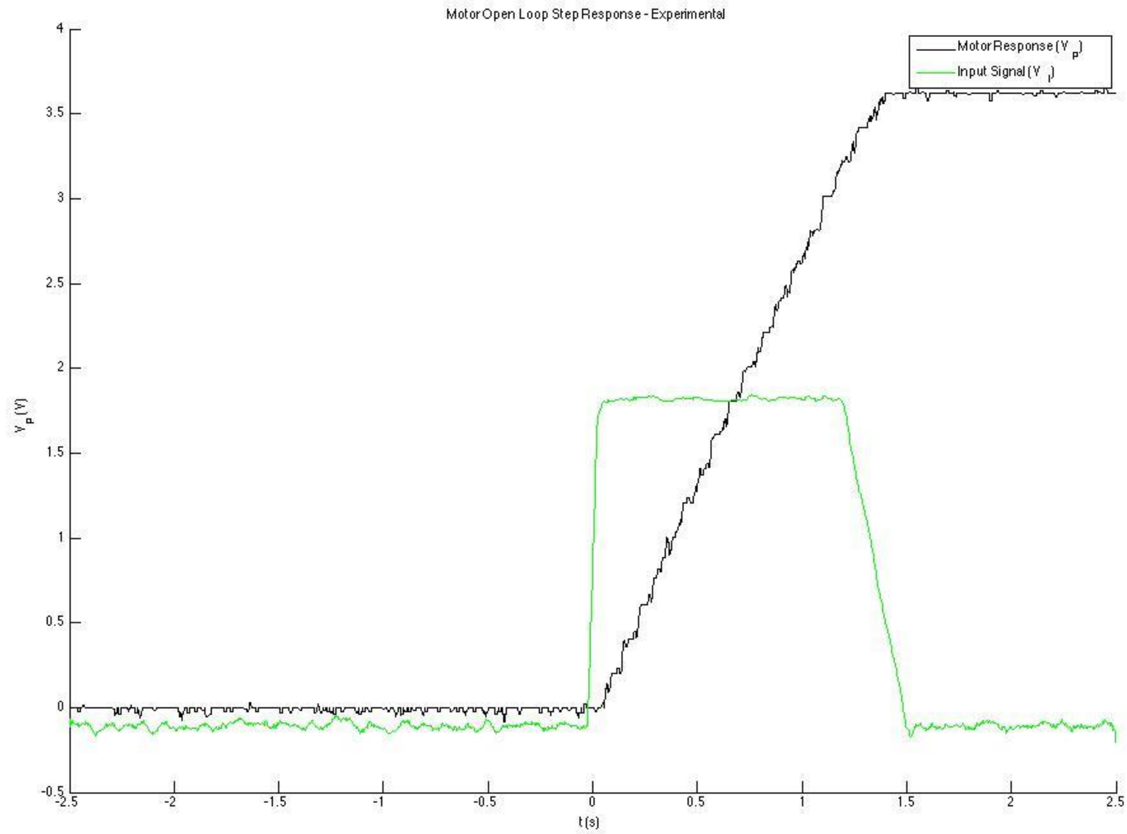
$$G_o(s) = \frac{V_p(s)}{V_m(s)} = \frac{K_m}{s(s + \alpha)}$$

Experiment-based system modelling:

The above simplified transfer function means that the servo motor system can be characterised by two numbers, K_m and α (this is typical of many simple electromechanical systems). However, we do not know any of the constants associated with these parameters such as motor constants, inertia of the load and the gear ratio. We will need to treat the system as a black box from which we expect a particular response, and we will estimate the unknown parameters using **experiment-based system modelling**.

To do such modelling, we will provide you with the system response to a step input. This is referred to as the **open-loop time response** of the motor system and is measured in terms of how the voltage applied to the motor affects the voltage seen at the potentiometer (constant voltage input should result in constant shaft speed). Then, to match with our model for $G_o(s)$, you will derive a good approximation to the servo motor system transfer function by selecting the best values that you can find for K_m and α.

An example of an open-loop time response is shown in the figure below. The green signal is a measure of the input step response, you can see that a 2V step response has been applied. The black signal corresponds to the voltage seen at the potentiometer. Just after the step response has been applied, the motor starts to turn and this voltage begins to increase.



3 Assessed Questions – System Identification

In this section, we will provide you with the open-loop response. You will use this to estimate the parameters from this model.

In preparation for these questions, download the functions `GenerateCSVRandom.p` and `UnknownData.p` from Blackboard and ensure they are in the appropriate MATLAB path.

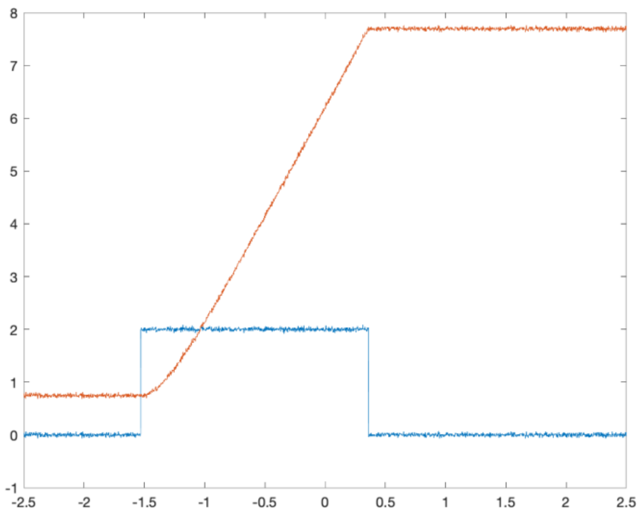
3.1 Question 1:

Create your randomly assigned data using the following MATLAB command:

```
>> [km, alpha] = GenerateCSVRandom('EGB345RandomData.csv');
```

This function will generate a CSV file ('EGB345RandomData.csv') that contains the time vector, the step input, and the step response of your servo system. Your true values of K_m and α will also be stored in the variables 'km' and 'alpha', respectively. Please note these values down for submission.

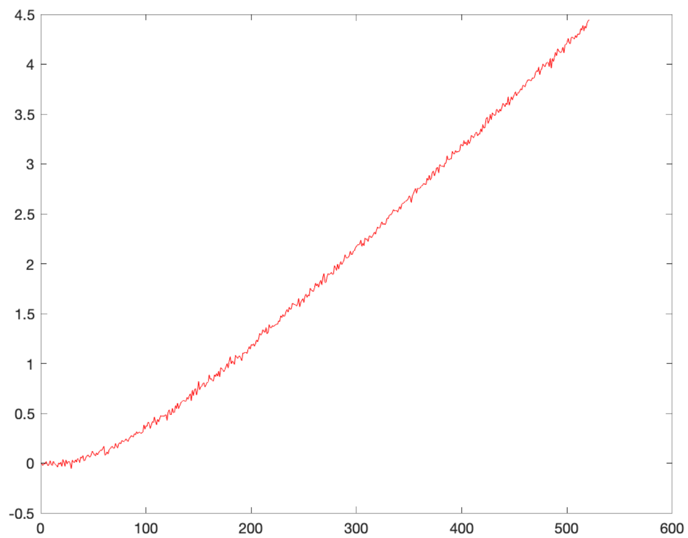
Read in the CSV file into your MATLAB workspace. The columns are labelled with text within the CSV file. Note that channel 1 is reading the 2V step input (not the usual 1V step) and channel 2 is reading the step response. This is to match the expected configuration as if you were taking these values from a real oscilloscope. The step response data should be named `'yn_random'`. Your data will look something like the following when plotted.



Remove the offsets and delays of the step response (generated by the given code above) so that the step response starts at $t = 0$s. You may assume the motor was initially in zero angle position, before there is a step input that causes the motor to start moving. You may use the step input as a reference to determine when the step response begins and ends.

You are asked to remove the voltage offset and shift the data's time index so that the motor's step response starts at the start of the data vector. You should also remove any data after the step input is switched off. The correctly shifted data should be named `'yn_random_fixed'`.

After shifting, you should have something like the following.




```
>> save('prelabdata_yn_random.txt','yn_random','-ascii')
>> save('prelabdata_yn_random_fixed.txt','yn_random_fixed','-ascii')
```

<u>Submit the following:</u>

1. Your true K_m and α values generated from `GenerateCSVRandom` (in the provided report template).
2. Plot `yn_random` (plotted in red) and `yn_random_fixed` (plotted in blue) **on the same plot** and submit as 'png' or 'jpg' files (in the provided report template).
3. Generated step response `yn_random` data as a .txt file (submitted to Blackboard).
4. Shifted `yn_random_fixed` data as a .txt file (submitted to Blackboard).

3.2 Question 2:

Using an approach similar to Practical 3, compose some MATLAB code able to simulate the motor's shaft angle for various values for K_m and α.

This will include creating the transfer function $G_o(s)$. You can then simulate the time response data using the *step* function in MATLAB. You should ensure you are observing the transient and steady state response.

Change values K_m and α in your MATLAB code until your simulated data matches the test data. You may use trial and error, or exhaustive search using *for loops*, to compare based on RMS error. We recommend you create a function to do this (see template for estmotor.m on Blackboard). The K_m and α values of closest match are called your estimated values of these parameters.

Use this estimation technique on the data `yn_random_fixed` and confirm that you can recover your true values of K_m and α (used to generate the data `yn_random`).

HINT: You may reuse the time vector from EGB345RandomData.csv above. You may need to shift or truncate the time vector to match your shifted step response data `yn_random_fixed`, if you haven't already.

Submit the following:

1. Your estimated K_m and α values (in the provided report template).
2. Plot `yn_random_fixed` (plotted in red) and the estimated model data using your estimated values of K_m and α (plotted in blue) **on the same plot** as 'png' or 'jpg' files (in the provided report template).
3. Paste your MATLAB code for estimating parameters (in the provided report template).

3.3 Question 3:

Create a second set of randomly assigned data using the following MATLAB command:

```
>> UnknownData('EGB345UnknownData.csv',yourstudentnumber)
```

where `yourstudentnumber` is replaced by your student number.

Please ensure you use the UnknownData function here, not GenerateCSVRandom from above. The generated data will be unique to your student number and will have different (hidden) model parameters of K_m and α.

You are required to conduct data analysis and estimate the model parameters of K_m and α. This will test whether your estimation process in Question 2 generalises to other sets of data.

As before, first remove the offsets and delays from the generated step response data. This fixed step response data should be named `'y1'`.

```
>> save('prelabdata_y1.txt','y1','-ascii')
```

Then, apply your estimation code from Question 2:

```
>> [alpha_est,K_est]=estmotor(t,y1);
```

Plot both the fixed step response data `'y1'` from EGB345UnknownData (plotted in red) and your estimated model data (plotted in blue) **on the same plot** to highlight how well they match.

Submit the following:

1. Your estimates of the K_m and α parameters corresponding to the EGB345UnknownData (in the provided report template).
2. The requested plot as 'png' or 'jpg' files (in the provided report template).
3. Based on the accuracy of your estimation in Question 2 (with ground truth parameters known), how accurate (use percentages) do you think your estimation of EGB345TestData is in Question 3?

4 Submission aspects (Do not submit as a zip file):

This multiple part assessment submission is very short, but each of the assessment submission parts must be scannable by SafeAssign and 10MB or less (attempts to subvert SafeAssign will be considered academic misconduct). Data and text must be ascii and editable.

4.1 Submission part i) [Document]

You must use the provided Word document template: Includes text and plots. Replace highlighted text with your submissions. Submit the Word document.

4.2 Submission part ii) [File attached submissions, 3 parts]

1. [File submission/attachment] From Question 1 above, provide the self-generated data. That is, the `'prelabdata_yn_random.txt'` data file (must be in the .txt format and editable).

2. [File submissions/attachments] From Question 1 above, provide the self-generated data with offsets and delays removed. That is, the `'prelabdata_yn_random_fixed.txt'` data file (must be in the .txt format and editable).

3. [File submission/attachment] From Question 3 above, provide the self-generated data. That is, the `'prelabdata_y1.txt'` data file (must be in the .txt format and editable).