Basic Experiments In Control Systems

Bharat Manvi

Indian Institute of Technology Madras

ee18s035@smail.iitm.ac.in

September 17 2019

Overview

Experiments

1. Find the transfer function of the Lego EV3(large motor).
2. Design a Proportional controller for the motor to attain a desired angular position.
3. Verify the open loop transfer function using closed loop system identification.

DC Motor

- The Dynamics of the DC motor is given by.

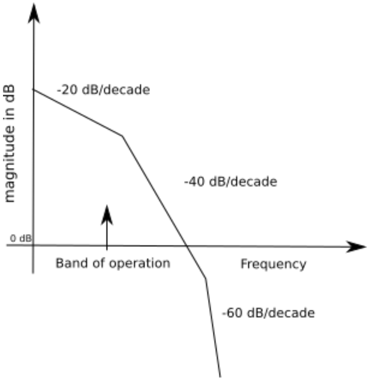
$$\frac{\theta(s)}{Vin(s)} = \frac{\frac{K_t}{R_a B}}{s((1 + s\tau_e)(1 + s\tau_m) + \frac{K_p K_t}{R_a B})} \tag{1}$$

- Where τ_e and τ_m are electrical and mechanical time constants given by, $\tau_e = \frac{L_a}{R_a}$ and $\tau_m = \frac{J}{B}$.
- Where R_a, L_a are armature resistor and inductor respectively. And J and B are Moment of interia and friction force respectively.
- Task is to calculate these parameters.

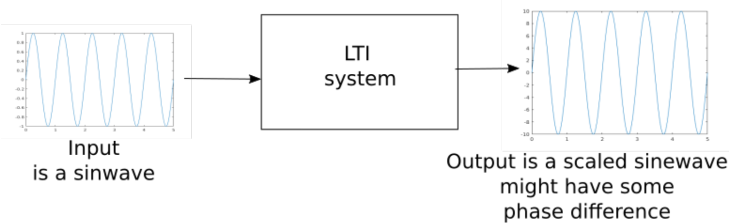
- Since τ_e is negligible compared to τ_m, equation 1 can be written as

$$\frac{\theta(s)}{Vin(s)} = \frac{K}{s(s+\alpha)} \tag{2}$$

- Now problem reduces to finding K and α
- The Magnitude Bode Plot of the motor will look like this,



- Since the band of operation is in low frequency so the approximation given by equation 2 is sensible.



- Send sine waves of different frequencies and collect corresponding output.
- Do this for different frequencies.
- And plot the Magnitude Bode Plot.
- Since the system is minimum phase system magnitude plot is enough to fully characterize the system. (As there are no unstable zeros or zeros on right half plane of S-plane)

What all we need

- An EV3 MicroPython image from its official website[1].
- Visual Studio Code which has required package for Lego EV3.
- Anaconda which has all data science tools(Optional).

Steps to be followed

- Download EV3 MicroPython image and create a bootable media.
- Put SD card into EV3 brick and boot the EV3 brick.
- Download Visual Studio Code and python.
- Create a new project in Visual Studio Code.
- Write a Python Code for the task mentioned.

[1] Getting started with EV3 MicroPython

System Identification

Aim

Find the transfer function of the Lego EV3(large motor),

$$H(s) = \Theta(s)/Vin(s) \tag{3}$$

where $Vin(t)$ is the input voltage and $\theta(t)$ is the angle.

Theory Required

- Let F be set of all frequencies we consider as the input frequencies.
- Since the operation of motor is at low frequencies we choose F to be,

$$F = \{f(Hz) : 0.3, 0.4, 0.5, 0.7, \ldots, 2.0, 2.5, 3.0, 3.5, \ldots, 8.0, 9.0, 10.0\}$$

- Input considered is the sine waves of different frequencies.
  As
  $$Vin(t) = Av \sin(2\pi ft)(Volts). \tag{4}$$
  where f is a signal frequency and $Vmin \leq Av \leq Vmax$
- Equation 4 considers Voltage as an input, but we can only give duty cycle as the input. So we consider $d(t)$ as the input given by, As
  $$d(t) = A\sin(2\pi ft) \tag{5}$$
  where $-100 \leq A \leq 100$ in percentage.

- The relation between $d(t)$ and $Vin(t)$ is given by,

$$Vin(t) = Vc \times 0.01 \times d(t) \tag{6}$$

- Denote the voltage stored in EV3 brick's battery as Vc, which is measured in mV. At full charge $Vc = 8$ V.
- Select sampling frequency f_s s.t it holds Nyquist criteria

$$f_s \geq 2f \qquad \forall f \in F \tag{7}$$

Task 1.1

- Measure the angle $\theta(t)$ for different input frequencies $f \in F$.
- Write a Python code to print the encoder and timer values on to terminal.

Steps

- To get angle information see the syntax in document[1].
- Similarly store time (Search from same source to syntax for time).
- Store the angle and time as lists and print it on Terminal.
- Also print the battery percentage on terminal.
- Then copy the content of the terminal.

[1] Getting started with EV3 MicroPython

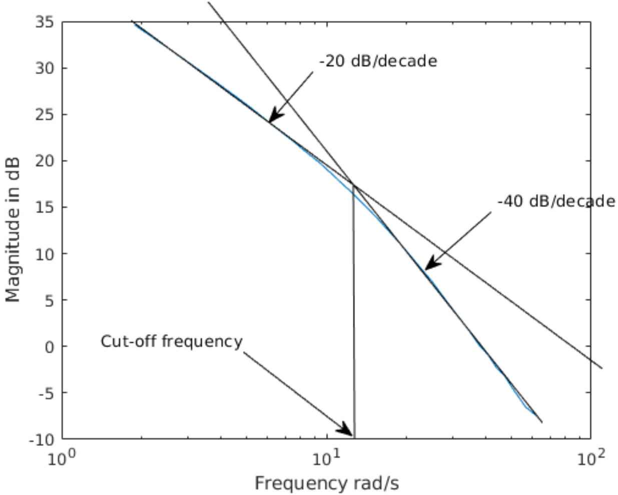
Task 1.2

- Write a Python code to calculate the magnitude ratio of input to output. i.e $\theta(t)/d(t)$ (wheel encoder values to duty cycle) for every frequency f.
- Write a Python code to plot the magnitude vs frequency graph (magnitude plot).

Steps continued ...

- After Copying the contents of the terminal paste it in python and store it as time-t(vector), battery voltage -Vc(scalar) and Angle-$\theta(t)$(vector).
- Duty cycle can be calculated in python using equation 5.
- Do the same steps for all frequencies, plot the magnitude plot in log scale and calculate the cutoff frequency as shown in figure.

Magnitude Plot



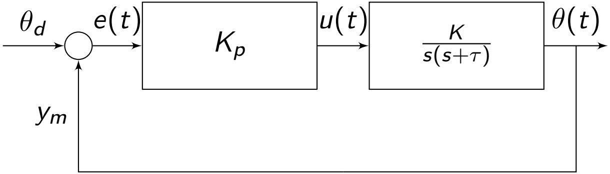
- The transfer function should look like,

$$G(s) = \frac{1257}{s(s+12.57)} \tag{8}$$

Proportional Controller

Aim

- The error $e(t) = \theta_d - \theta(t)$ should go to zero as t tends to infinity. Where θ_d is desired angle and $e(t)$ is the error.
- To design a Proportional Controller K_p for damping ratio ζ =0.68.
- To verify the simulation results with the practical model.



Theory Required

The closed-loop Transfer function of the system with K_p is shown in the previous figure is,

$$G_{cls} = \frac{\theta(s)}{\theta_d(s)} = \frac{K_p K}{s^2 + s\sigma + K_p K} \tag{9}$$

Comparing coefficients with Second Order System's characteristic equation we get,

$$\sigma = 2 \times \zeta\omega_n \tag{10}$$

$$\omega_n^2 = K_p K \tag{11}$$

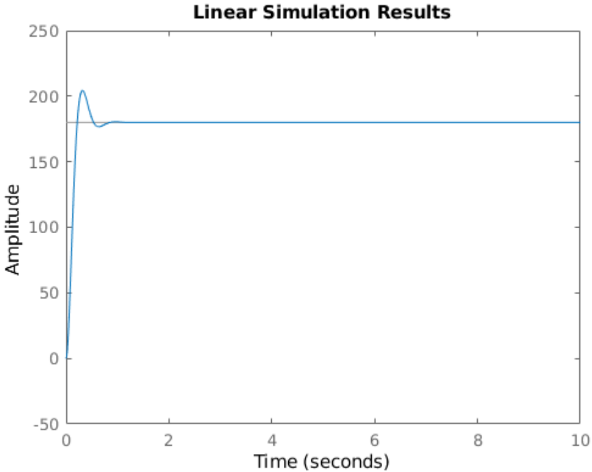
Here $\theta_d(s) = c/s$ where c is the desired angle in degrees. This problem is called setpoint problem.

Steps

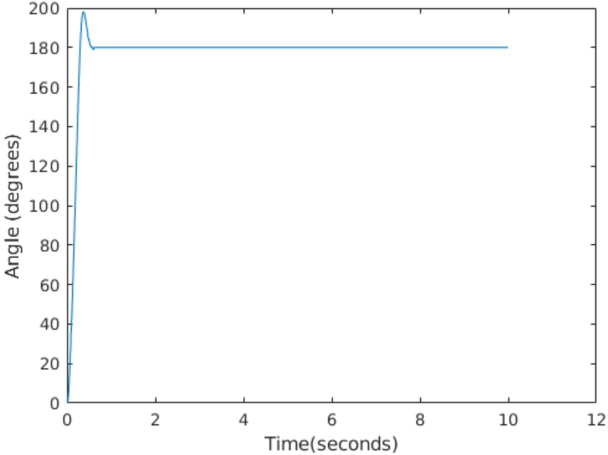
- Now create a system model in python control package using 'control.matlab.sys()' command.[2]
- Simulate the closed loop model using 'control.matlab.feedback()' command.
- Plot the step response of the system using 'control.matlab.lsim()' with desired angle as the step input.
- Write a Python code for a proportional controller for the motor model.
- Print the results on terminal as a list in python and use them in Python to get the Step response.
- Compare the results with simulation results.

[2] See python control package documents for the syntax

Expected Results-Simulation



Expected Results-Practical



Closed Loop System Identification

Aim

Verify the Open loop transfer function using the bode plot of closed loop system.

Theory Required

The closed-loop Transfer function of the system with K_p as earlier is given by,

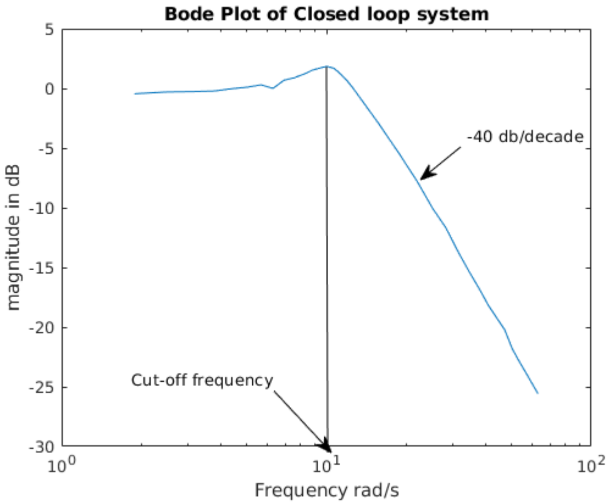
$$G_{cls} = \frac{\theta(s)}{\theta_d(s)} = \frac{K_p K}{s^2 + s\sigma + K_p K} \tag{12}$$

Now, $\theta_d = c(t)$. This problem is called as tracking problem.

Steps

- Give $\theta_d(t)$ as the sine wave different frequencies and measure the corresponding outputs.
- Use previous Python code to find magnitude ratio of the signals.
- Plot frequency vs magnitude graph to calculate cut-off frequency.
- Find the corresponding open loop transfer function.
- Compare the cut-off frequencies of transfer functions obtained by two methods.

Expected Results



Things to Ponder

One will be able to answer these questions after the completion of experiments

- An exact comparison of practical and theoretical results is not possible, Why?
- Why the Proportional controller can reach the final value?

References

LEGO education
Getting started with EV3 MicroPython
Version 1.0.0

Thank You