1. Generate a transfer function Matlab

- This laboratory session is intended to demonstrate the principles of gain margins in classical control which can be obtained from Bode diagrams
- Consider the transfer function:

$$sys = \frac{10}{s^3 + 10s^2 + 40s + 40}$$

- Use the Matlab tf function to generate this Matlab transfer function for the open loop system and place this in a variable sys.
- If you display the contexts of transfer function variable sys you should get the following output:

```
sys =

             10
  ------------------------
  s^3 + 10 s^2 + 40 s + 40

Continuous-time transfer function.
```

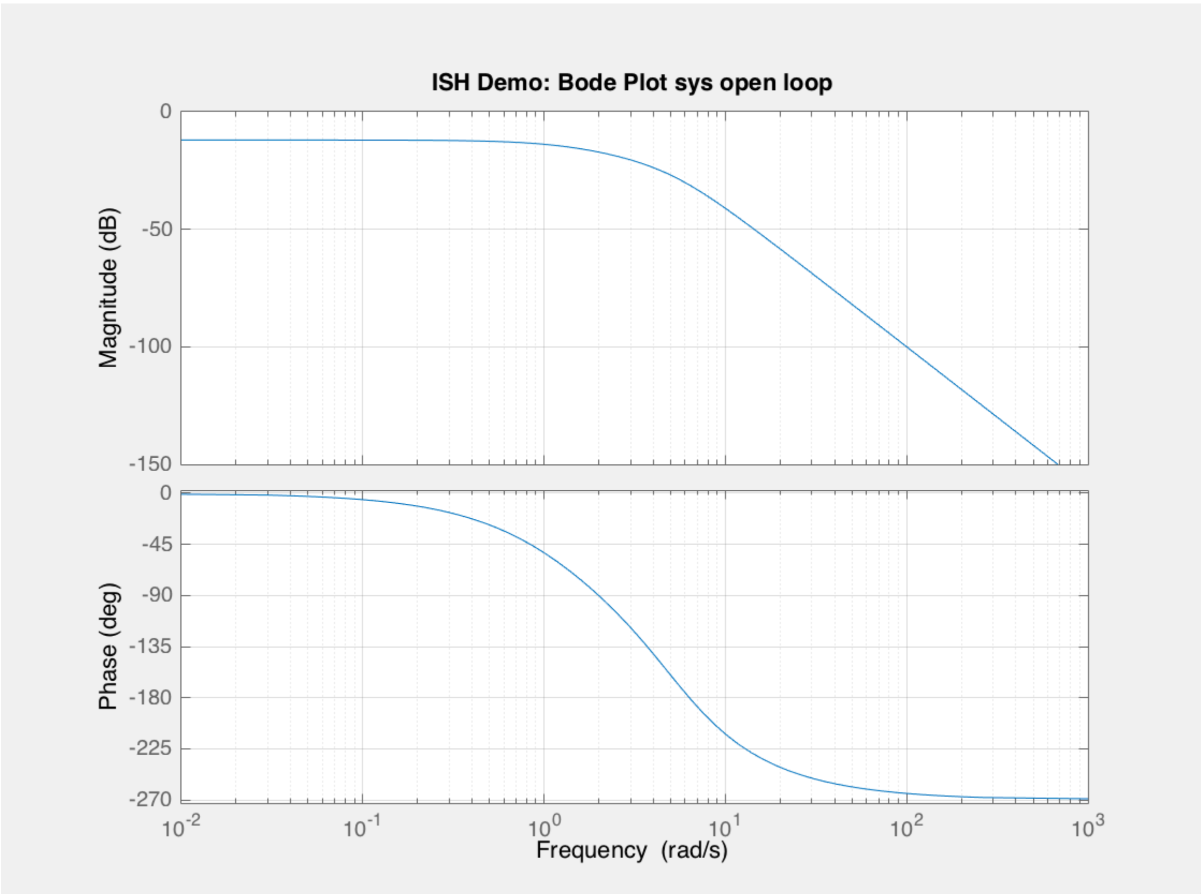
2. Using initial and final value theorem

- Use the Laplace initial and final value theorems to calculate:
  - The initial response to a step input
  - The final response to a step input

- HINT: remember the factor s arising from the theorem and the factor 1/s arising from the step function!
- Show the algebraic steps in your calculation of both of these quantities.

3. Bode diagram for open loop response

- Use Matlab to generate a Bode plot of the open loop transfer function.
- You should get a plot that looks something like this:

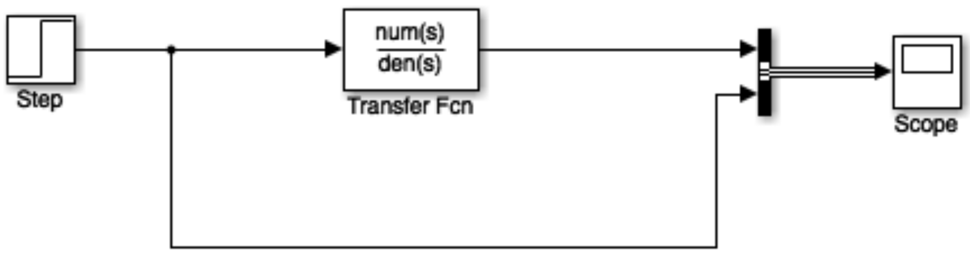


- Estimate the bandwidth of the transfer system from the Bode plot

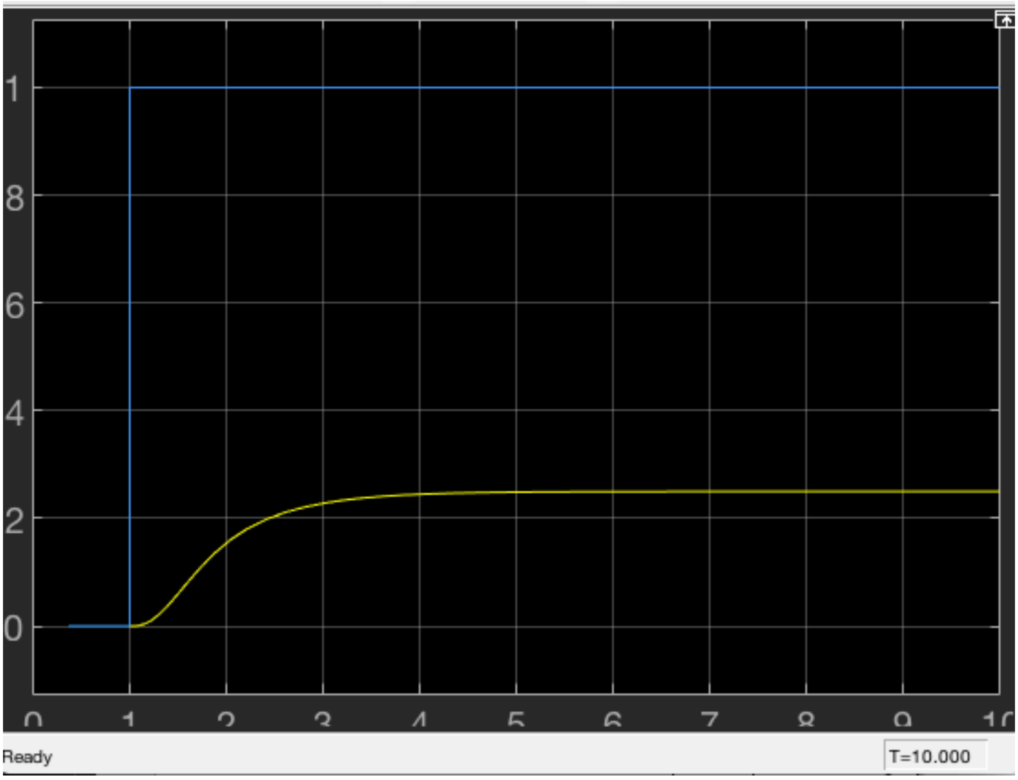
- HINT: Bandwidth frequency is value corresponding to a gain of -3 dB
- Estimate the system rise time for step input
- What is the significance of the gain and phase margins in conjunction with feedback control?
- Identify the open loop system gain and phase margins on the Bode diagram.

4. Build open-loop Simulink simulation

- Implement the transfer function sys in Simulink
- Drive it with a step function
- Display the sys output on a scope
- Also simultaneously display the input step waveform on the scope too
- Your Simulink model should look something like this:



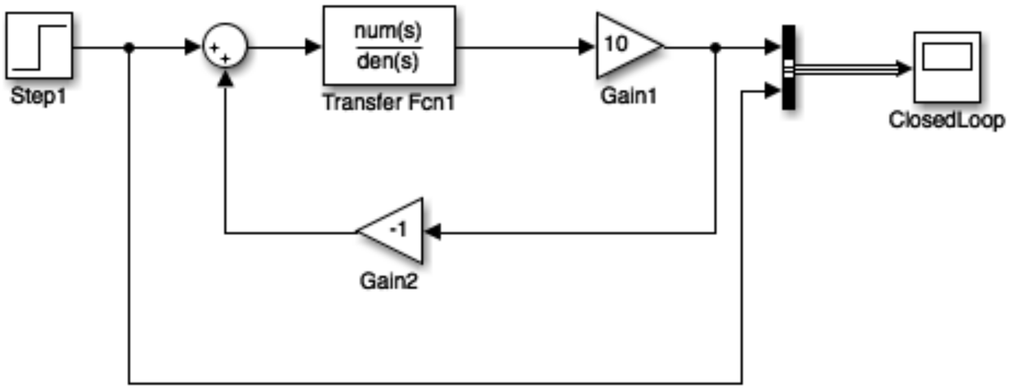
- Put the scope output response into your report.
- Your open-loop Simulink scope output should look something like this:



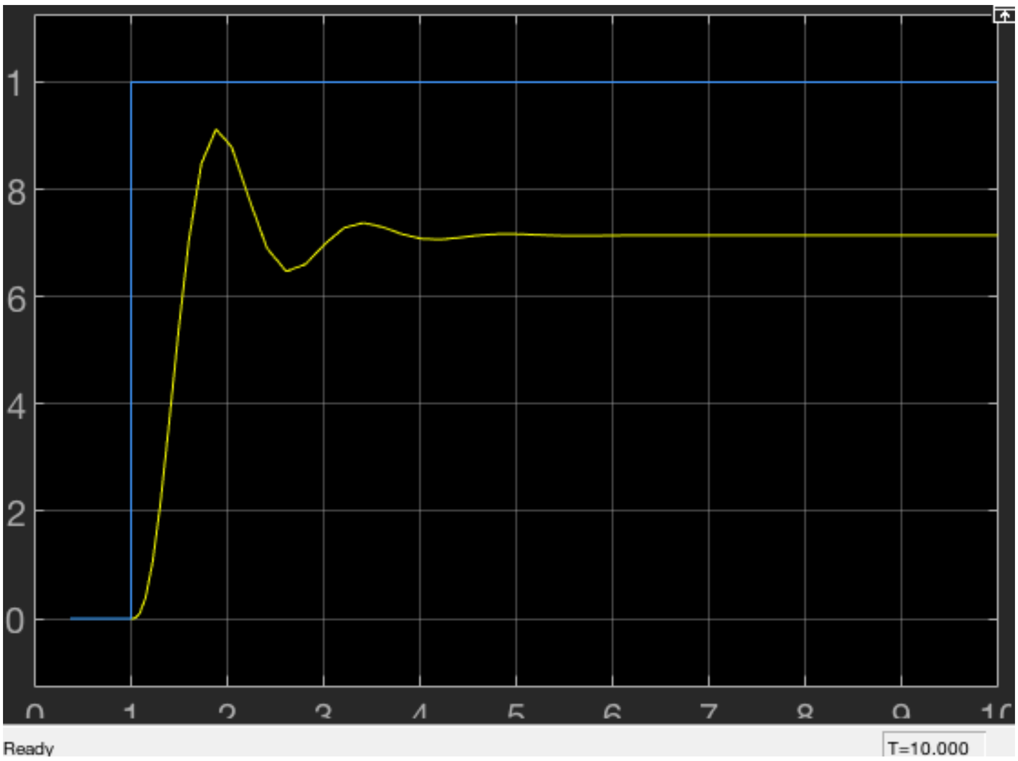
- What characteristics of this response are important?
- Compare your results for the critical measurements of performance with those from the initial and final value theorem predictions and the rise time prediction

5. Build closed-loop Simulink simulation

- Now build a closed-loop control system
- Set the feedback path gain to a value of -1 to give negative feedback
- Put an additional gain of 10 in the forward path
- Drive with step function and display output on scope
- Also simultaneously display the input step waveform on the scope too
- Your Simulink model should look something like this:

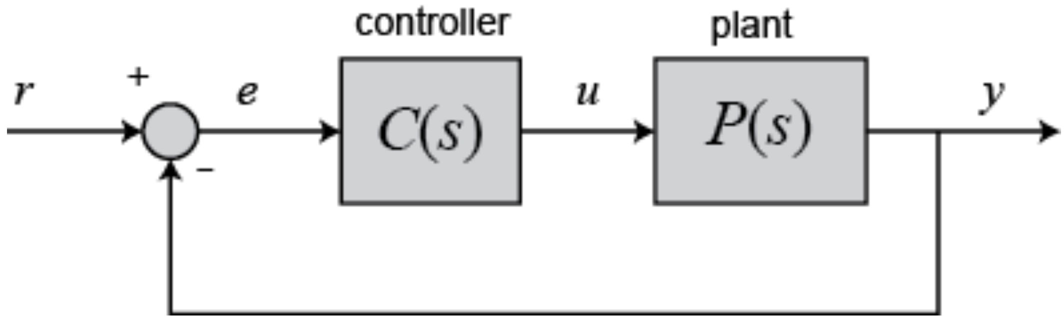


- Put the scope output response into your report.
- Your closed-loop Simulink scope output should look something like this:



6. Build closed-loop transfer function

- Feedback gives rise to a modified transfer function
- HINT: remember for following system:



$$\frac{Y(s)}{R(s)} = \frac{C(s)P(s)}{1+C(s)P(s)}$$

- HINT: In this case C(s) = 10

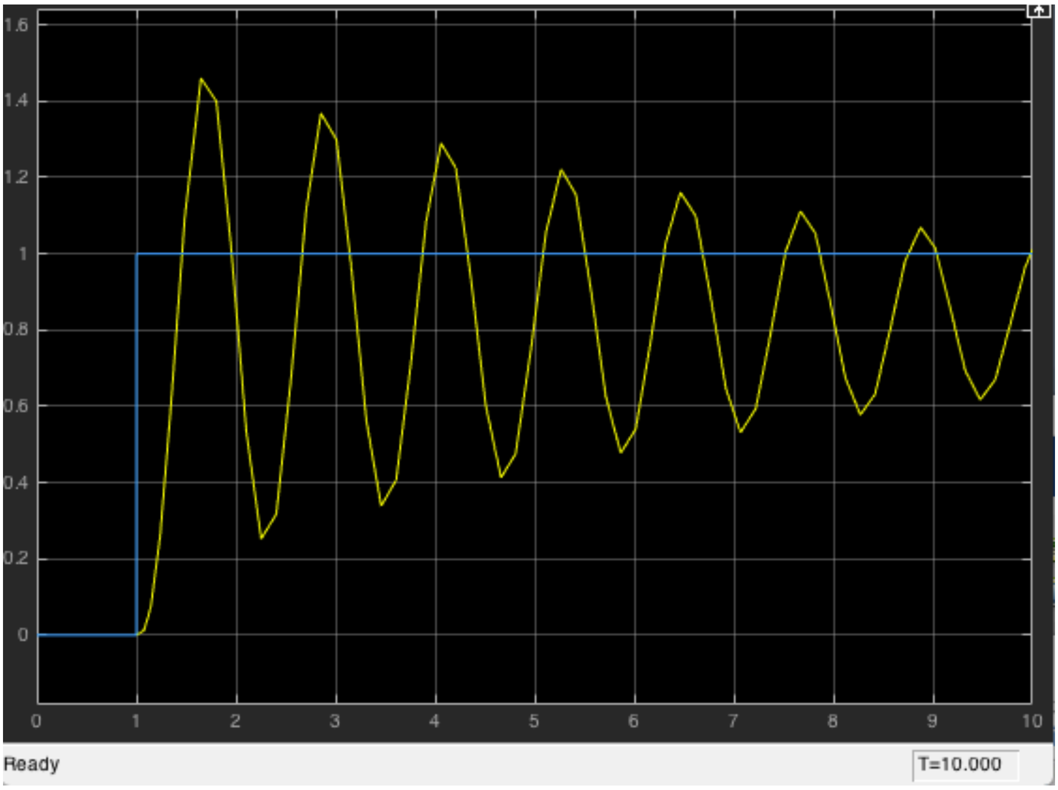
- Calculate the closed-loop transfer function of your system with the additional gain of 10 in the forward path.
- Directly implement this transfer function in an open-loop simulation using Simulink
- Again drive it by a step waveform and record the scope output
- Show that you get the same response as you did with the feedback system in part 4.

7. Add gain to close-loop Simulink simulation

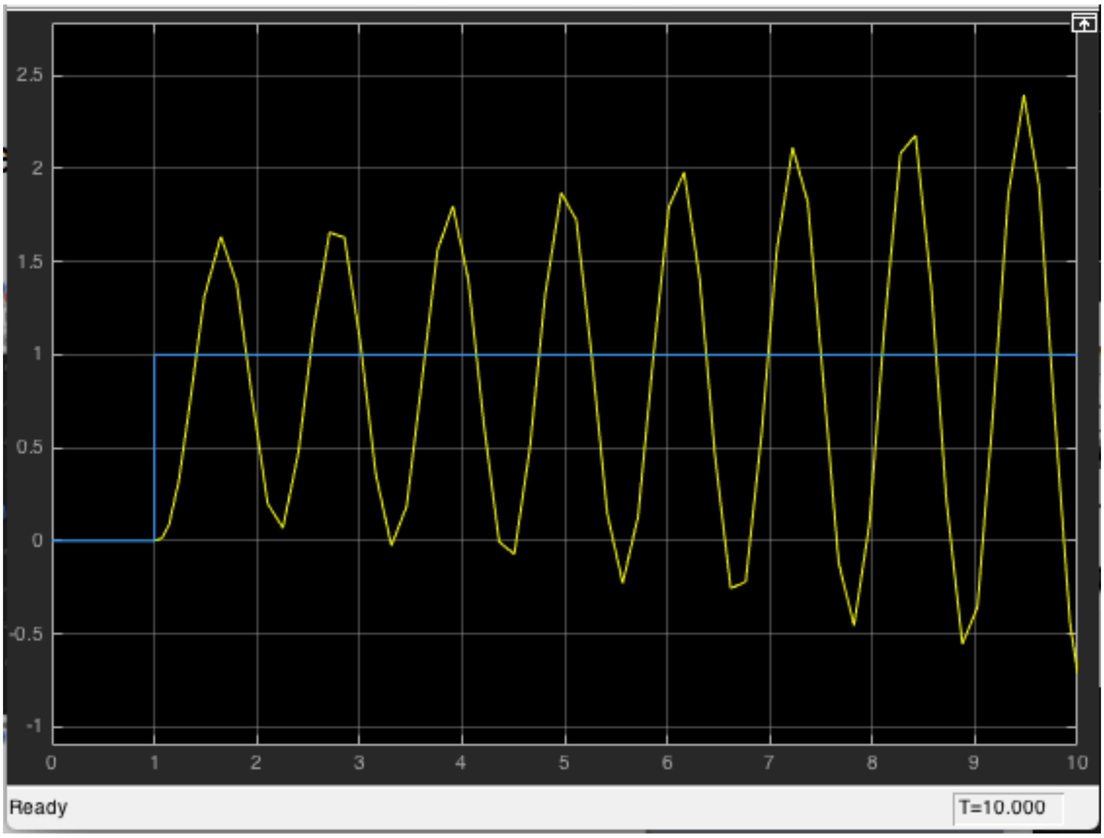
- Using the gain margin you computed earlier, adjust gain of your feedback control system in Matlab so that results in a system that has gain that is:

  - Just below the gain margin
  - Just above the gain margin
- What gain values did you use?
- Run the step response simulations in Simulink and record the output in both cases.
- You should get results similar to these:

  - Just below gain margin:



  - Just above gain margin:



- What can you say about the system outputs in these two gain conditions?
- What characteristics of system output are important if a system is to remain stable?