Information Engineering & Technology
Electronics, Communications, & Networks
Systems & Control Course Evaluation Project

Systems and Control Project

Directions:

- A simulation using **MATLAB** is a must for this project.
- Plot the responses got by MATLAB and the code used.
- Soft copies of the solution and hard copies are required.
- Solution must be in groups of **3- 4 students**, not more.

An e-mail to be sent to Eng. Omnia Mahmoud omnia.abdel-hamid@guc.edu.eg or Dr. Phoebe Edward Nashed " phoebe.edward @guc.edu.eg that include the name of each member in the group, ID number and the lab group.

GOOD NEWS: cross-labs is allowed ☺

- This project is of **20% of the total score** of the Systems & control course.
- The deadline of the project will be on the day of the evaluation.
- Evaluation dates for the project are in the last week on 9, 10, and 11 January 2021

(The exact timing for each group to be announced later)

Objective:

The objective of this project is as follows:

- A practical application on the Systems and Control course.
- Analyzing some given Processes.
- Designing the PID controller to get a specific response.
- Tuning the PID controller using different rules.
- Designing Lead-Lag Compensator to get a specific response.
- Tuning the Lead-lag compensator.
- Improving either the transient or the steady state responses or both for any system.

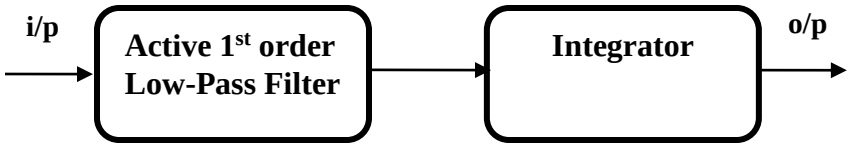
Procedure:

Students are required to do the following:

- Designing the electric circuit of the given block diagram with the shown transfer function **(Using OpAmps)**. Then, draw its detailed block diagram with its hardware Circuit Components in addition to the hand-written analysis.
- Analyzing the system response for a given plant with unity feed-back path for both transient and steady state responses.
- Designing an analog circuit that represents a PID controller with the flexibility of changing its parameters.
- Analyzing the system after inserting the controller in the feed forward path for both transient and steady state response (using MATLAB and Simulink)
- Deducing the improvements made by the controller.
- Designing an analog circuit that represents a Lead-Lag compensator with the flexibility of changing its parameters.
- Analyzing the system after inserting this new controller in the feed forward path for both transient and steady state response (using MATLAB and Simulink)
- Deducing the improvements made by the controller.
- Designing the Hardware circuit (This part is a **Bonus**).

Steps:

For the block diagram shown:



The above block diagram has the shown 2nd order feed-forward transfer function G(s):

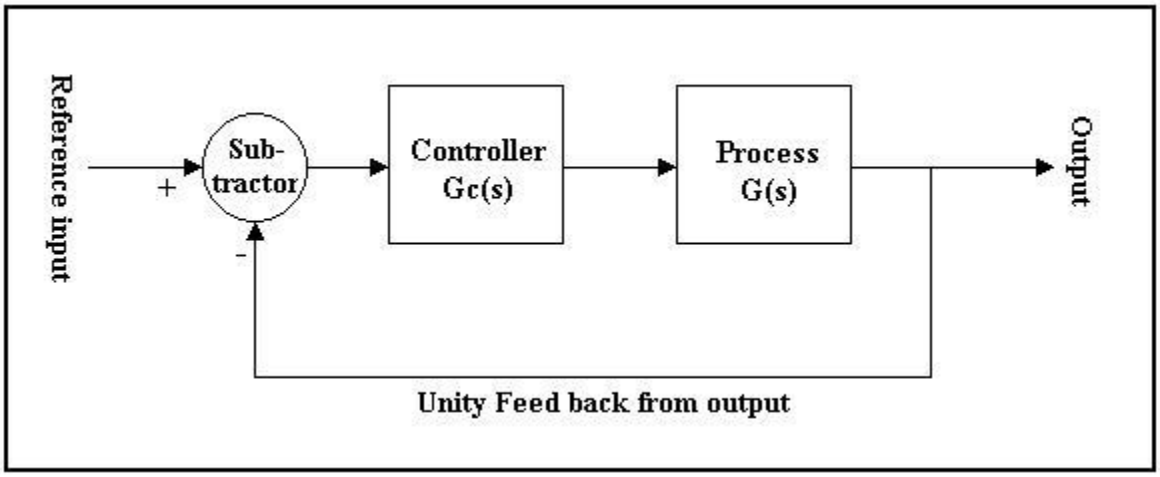
$$G(s) = \frac{\omega n^2}{s(s + 2\zeta\omega n)}$$

Where ω_n is the least last 2 digits in any student ID in the group, and ζ is equal (0.5 if ω_n is multiple of 5, 0.25 if ω_n is multiple of 3, else ζ is equal 0.75).

You are required to do the following (assuming Ideal Opamps):

- Design the equivalent electric circuit (using op-amps) describing your open loop transfer function of this process G(s).
- Derive the closed loop transfer function with the values of components given above assuming unity feedback.
- Use MATLAB to get the Step response, Impulse response, Rise time, Peak time, Settling time, Maximum peak response, Bode plots, and Root loci of this system.

Now, It's required to design a controller Gc(s) to improve the system response as shown:



1. Use MATLAB to design the parameters of that PID controller to get a maximum peak of approximately ***20%*** and steady state error not more than **5%**.
2. Build the analog circuit that represents the PID controller with the flexibility to use only P, PI, PD, or PID controller.
3. Derive the system closed loop transfer function after inserting the controller.
4. Analyze the complete system using MATLAB, and Simulink.
5. Build the analog circuit that represents the Lead- Lag Compensator with the flexibility to use only Lead, Lag or both Lead-Lag compensator.
6. Derive the system closed loop transfer function after inserting this new controller.
7. Analyze the complete system using MATLAB, and Simulink.