Control Systems Theory
University of Florida
Mechanical and Aerospace Engineering

Course Project
Issued: April 26, 5 pm, Due: April 30, 9 pm.
No late submission will be allowed.
Please submit your project files *only* through e-learning.
No email or paper submissions will be accepted.

Max score: 100 points

Goal of the project: Determining an approximate model of a physical plant, and ensuring robustness to modeling errors, are crucial part of control design in practice. In this project you will go through the process of system identification from open-loop experiments and control design based on the identified plant model. The task is to identify an LTI-approximation of a plant and design a feedback controller. The open-loop system is shown in Figure 1. The saturation limits of the actuator are known to be $u_{\min} = -4$ and $u_{\max} = 5$.

A "virtual plant" is provided to you through a Simulink model. You will treat this as a real plant, identify a model by experimenting with the plant, and then design and test a feedback controller to control the plant to perform reference tracking. The virtual plant is created to reproduce real experiments as closely as possible, in which every new test will have some random variation.

Files: All files are in e-learning. Download all the files to a folder of your choice. I strongly suggest you "cd" to that folder from MATLAB©'s command promt before you try to run anything. Open all the MATLAB© files and read the comments on the top portion of the scripts.

The Simulink model `openLoopTestBed.slx` (created using R2018a) contains the virtual plant that can be used to perform open loop and closed loop experiments. The simulink model takes an input signal $u[t_i]$ and produces a time-trace of the output $y[t_i]$ at the same time instants. An example MATLAB©script is provided, `conductOpenLoopTestonVirtualPlant.m`, to illustrate how to conduct an open loop test on this virtual plant.

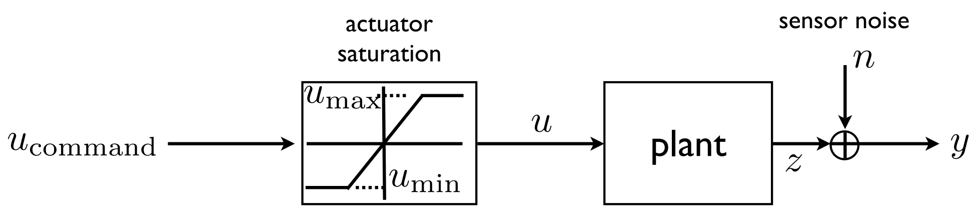

Figure 1: Plant with actuator saturation

1 System identification

1.1 Task I.A

1. Identify a *nominal* frequency response of the plant, $\hat{P}_0(j\omega)$ by performing sine sweep[1].
2. Fit a parametric transfer function $\hat{P}_0(s)$ to the identified nominal frequency response $\hat{P}_0(j\omega)$.

Provide the relevant Bode plots in your report to show how accurately the parametric model fits the frequency response. (More details about the report are at the end.)

2 Control design

2.1 Objectives

The objective is to track a reference signal with small steady state error. The reference signal is sequence of two steps, and is generated by the function `F_ref_at_t.m`, with the option `ref_type = 'step_zero21_at5_back20_at15'`. The script `exampleGenerateReference.m` shows how to generate this reference signal[2].

The target performance criteria to meet are:

1. peak overshoot of less than 15%, and
2. rise time less than 2 seconds.

2.2 Task II.A

Design a controller using state-space design techniques to achieve the performance objectives listed above. Here are the sub-tasks:

1. Describe the structure of your proposed closed loop control system. Show a diagram, not just words. (Hint: to avoid actuator saturation, you will probably need to LQR methodology. also, this is a setpoint tracking problem, not a stabilization problem, don't forget that.)
2. *Design and "design testing":*
   (a) Design the controller parameters for the nominal plant model you identified in the system identification step.
   (b) Test the performance of the closed loop through Simulink simulations using $\hat{P}_0$ for the plant. *Do not use the virtual plant in this phase!* Do not forget to add the actuator saturation block in your Simulink model. These simulations should show that your design

[1] Some have asked me if they can use other methods; methods that have not been covered in class. Due to the constraints of uniformity required for grading, I recommend you do sine sweep. You can use other methods to inform your choice of sine sweep parameters.

[2] The function can produce another reference, and you can modify the function to create all sorts of references, but you do not need to show how your controller perform with those references.

meets the performance requirements, at least in the "design tests" that uses a mathematical model of the plant in place of the real plant[3].

3. Analyze the gain and phase margin of the closed loop and comment on how robust your controller is to error in the plant model $\hat{P}_0$.

4. *"Production test":* Once you are satisfied with your design (based on the simulations above), test your design on the real plant by wrapping a feedback loop around the "plant" block in the Simulink model provided. (Goes without saying that you will have to create a separate Simulink model: copy the plant block from the open loop example provided into that model's window and add the controller.) If the performance in the production tests are different from that seen in the design tests, comment on where you think the difference is coming from.

3 Report

3.1 Task III

Write a clear report.

The grading for this project will be based mostly on the report. I will consider several aspects.

1. Your report must be clear and concise. The major results must be clearly visible and not hidden within unimportant details.

2. For the system identification part, you must clearly describe what you have done to minimize the effect of transients and measurement noise. The more precise and well-reasoned your choices are, the better. For example, if the first X seconds of data was ignored to minimize the effect of initial conditions, more points will be awarded if the choice of X is clearly explained.

3. Your report must describe the derivation of the controller. Merely specifying the control parameters without an explanation of how they were arrived at will fetch you a 0. Describe your methodology, for example, what steps you have taken to ensure that the chances of reaching saturation limits are minimized? What steps have you taken to guard against instability or poor performance due to uncertainty in the plant model?

4. All plots have to have clear x-axis and y-axis labels and plot legends if multiple curves are plotted in the same figure. The font size of the labels, captions, and legends have to sufficiently large to the legible.

In terms of apportioning your time to these tasks, a rough guideline is to assume that Task I (system identification) has 20 (=15+5) points, Tasks II (controller design) has 40 (=5+15+10+10) points, and Task III (report) has 40 points.

This weight on the report may appear artificial: it is not possible to separate the report from Tasks I and II. After all, if you do not do Tasks I and II, there is nothing to write a report on. In that case, no matter how beautiful a report you submit, you will not get 40/100, instead you will get

[3] If you cannot meet the requirements, that is ok, as long as you get close. Meaning, if your design achieves a peak overshoot of 20% instead of 15%, I recommend you move on, and perhaps come back to refine the design later if you have time; you would not lose too many points for being able to achieve 20% instead of the target 15%. Read the grading rubric at the end before you spend too much time fine tuning your controller.

0/100! But I mention "40 points for the report" so that you appreciate the role the report plays in your grade. This relates to one of the footnote comments: as long as you get close to meeting the performance requirements, do not spend much time refining the design to get performance. I will not penalize you for missing the performance requirements by a little bit. But if it is not clear to me from the report how you proceeded with the design, what design choices you made, and why, you will lose a big fraction of the 40 points, even if your design meets all the performance requirements!

4 What to submit

1. A report in .pdf format (typed, not hand-written) that is at most 10 pages, including figures. There should not be any MATLAB© code in the report. The report should named `FirstnameLastname_ProjectReport.pdf`. For example, if I were a student in this class, my report will be named `PrabirBarooah_ProjectReport.pdf`

2. A MATLAB©script named `sineSweepOnVirtualPlant.m` that uses the Simulink model openLoopTestBed.slx to perform sine sweep. The script should produce a plot of the identified frequency response.

3. Two Simulink models, one for the "design test" and one for the "production test". These models should be named `closedLoop_Design.slx` and `closedLoop_Production.slx`. And, a MATLAB© script named `testClosedLoop.m` that calls these Simulink models to perform the design tests and production tests. The script should generate the plots you used in your report to show closed loop performance. (It does not have to generate all the report plots, such as those for margin analysis.)

4. All the MATLAB© and Simulink files I provided as part of the project.

Noe comes the really important part: *If you submit any other files other than those listed above, you will automatically get a 0 in the project*[4]. I will download the files you submit into a folder and run your MATLAB©scripts. If they do not run (for instance, they throw an error), you will get a 0 in the project.

[4]You can use other MATLAB© scripts during the design stage, that is fine. But then, consolidate your design and testing into just the files I mentioned. I recommend that when you are ready to submit, create a folder, and move *only* the files you need to submit to that folder. Go through the checklist carefully, a silly error can cost you the whole project! Next, test that your scripts run without an error from that folder. You can then submit all files in that folder to e learning.