

CONTROL SYSTEMS THEORY
University of Florida
Mechanical and Aerospace Engineering

HW 8

You are encouraged to use MATLAB[®] to verify your answers whenever you can. However, unless specified, do not use MATLAB[®] to solve a problem.

Problem 1. A plant is given as a SIMULINK[®] model `plantForExperiment.xls`. An example matlab script to define necessary variables and run a simulation with this plant is provided in `runExperimentOnPlant.m`. Write a MATLAB[®] script to identify the frequency response of the plant using the `etfe` method, by modifying the script I discussed in lecture 40, which is in Files > resources > code > systemIdentification. Tasks:

1. Add a feature to reduce the impact of initial conditions on estimation error (such a feature is absent in the MATLAB[®] script I provided and discussed in the lecture on using `etfe`).
2. Repeat the identification for several types of inputs. Provide two figures: for the best identification results and worst identification results. Keep everything else the same in these two experiments except for the signal type. Also, make sure the input magnitude in all your experiments is within ± 0.6 .
3. In each of the two figures above, show both the estimate and the true frequency response. You can see what the true plant is by looking inside the SIMULINK[®] model. *Beware: As I discussed in the lecture, `etfe` estimates the discrete-time plant whose frequency response is slightly different from its continuous time counterpart, especially in higher frequencies. The simulink model is for a continuous-time system.*

Submit: a brief description of the feature (item 1), not exceeding five sentences. The two figures mentioned above, and a brief description of which inputs you tried and which among them gave you the best and worst results, and why (not exceeding ten sentences).

Hint: it turns out that the last 2 optional parameters in MATLAB[®]'s `etfe` command are important to get good results in this problem. In the example covered in the lecture that was not necessary. Another key to getting good estimation is to drop some initial data to reduce the impact of initial conditions (the answer to part 1 of the question). Here is my code, as an example:

```
.....
uetfe=u(end-N+1:end);
yetfe=y(end-N+1:end);
idDataForThePlant = iddata(yetfe,uetfe,Ts);
smoothingWindowSize = N/10;
Phat_DT = etfe(idDataForThePlant,smoothingWindowSize,numFreqRes);
...
```

Problem 2. Estimate the frequency response of the plant $P(s)$ mentioned in Problem 1 by using the sine sweep method. Use the same simulink model for performing experiments. Make sure

the input magnitude in all your experiments is within ± 0.6 . *Hint: start with the script named `sineSweepExample_withlsim_forstudents.m` which is in Files > resources > code > systemIdentification.*

Submit: (1) a Bode plot showing your estimated frequency response values and the true frequency response of $P(s)$, (2) a brief description of how you decided which frequency values to perform sine sweep on (i.e., why those values) and your choice of various parameters such as number of cycles to average, duration of initial data to drop, etc. (2) Comment on the relative cost vs. benefit of the two methods (etfe vs. sine sweep) you see from your results. (3) Your matlab code, as a single .m file.

Note: Since the solution to the problem is effectively already provided in the script `sineSweepExample_withlsim_forstudents.m`, the matlab script part of the solution to this problem will not be provided. You will need to run sine sweep on a plant provided in a simulink model form in the project. So you should write the MATLAB[®] script for this problem in a clean manner so you can reuse it for the project easily.

Problem 3. [set point tracking, LQR, etc.] Do this using MATLAB[®]. Consider the following plant $P(s)$ with input u and output y :

$$P(s) = \frac{100(s+5)}{(s+0.5)(s+1)(s+10)(s^2+4s+13)} \quad (1)$$

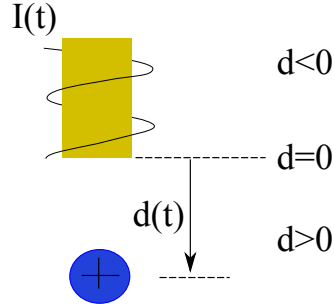
It is almost the same as one the plant in HW7, except this plant is stable. Your job is to design and output feedback controller for this plant by using a state observer and combining it with a state feedback controller to track a constant reference (set point): $r(t) = 3$ for all t , and simulate the closed loop. (use `tf2ss` to get a minimal realization of the plant for design.) You have to design the control system to achieve a certain performance that is specified below. Here are the tasks:

1. Draw a sketch of the closed loop control system.
2. Construct a SIMULINK model of the closed loop system. Write a MATLAB[®] script to initialize and define all parameters needed by the Simulink model, and also to run the model using the `sim` command. Choose a sufficiently long time to see transients die out, use the following initial conditions: $x(0) = 10^{-3} \times [5, 5, 5, 5, -5]^T$, $\hat{x}(0) = 0$. Use this initial condition for all subsequent simulations. *Hint: At this stage, don't worry too much about tuning the controller for performance; just ensure that the closed loop is stable by using the pole placement method.*
3. Now refine the design to meet the following objective, or get close to it: “peak overshoot less than 100% and 2% settling time less than 10 seconds, while the control command $u(t)$ less than 0.5 in magnitude for all t . Try the following two design methods for choosing K :
 - (a) Pole placement: pick desired locations of controller poles (that is, eigenvalues of $A - BK$), and then use the `place` command to compute K .
 - (b) LQR.

In both cases, determine L by using pole placement, and choose observer poles to be 5 times to the left of controller poles. The simplest way to do this is to simply multiply controller poles by 5, or add $-5 \times \min(\text{real}(\text{controller_poles}))$ to the controller poles. *Note: I could not achieve the objective using pole placement, but I was able to do it with LQR in less than 2 minutes.*

Submit: (i) plots of the relevant outputs (r and y vs t in one figure, u vs. t in another), (ii) a picture of your simulink block, (iii) a brief description of your design process (not more than 5 sentences for each method, pole placement and LQR).

Problem 4 (open loop electromagnet). Consider the following dynamic model of an electromagnetically suspended mass shown in the figure below

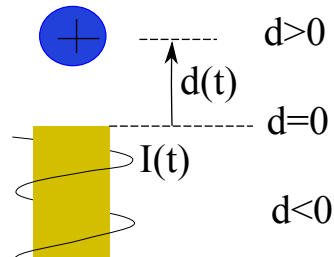


$$m\ddot{d}(t) = mg - F_e(I, d) = mg - \alpha \frac{I(t)^2}{d(t)^2}, \quad d(t) > 0 \quad (2)$$

where $d(t)$ is the position of the mass, $I(t)$ is the magnitude of the current through the electromagnet, m is the mass, g is the acceleration due to gravity, and α is a positive constant. Note that the electromagnet's position is fixed, which determines the origin $d(t) \equiv 0$. The position $d(t)$ is measured with an optical sensor, so it is an output. Finally, it should be emphasized that the model is valid only for positive values of d . The actuation signal or control command is the current $I(t)$.

1. Express the ODE in state space form $\dot{\bar{x}} = f(\bar{x}, \bar{u})$ where $\bar{u}(t) \triangleq I(t)$. Define your state vector \bar{x} explicitly.
2. Determine all the equilibrium points $(\bar{x}_{eq}, \bar{u}_{eq})$ of this system.
3. Suppose the desired position of the mass is d_0 (some positive constant). Determine the equilibrium point $(\bar{x}_{eq}, \bar{u}_{eq})$ corresponding to this desired position.
4. Linearize the system around the equilibrium point corresponding to the desired position d_0 . Is this linearized system internally stable/asymptotically stable? Do you need to know the numerical value of any parameter to answer this question?
5. Is the equilibrium point $(\bar{x}_{eq}, \bar{u}_{eq})$ (corresponding to the desired position d_0) stable, asymptotically stable, or unstable (in the sense of Lyapunov), or is it not possible to say from the linearized system?

Problem 5 (open loop electromagnet - but now flipped up). Consider the following dynamic model of an electromagnetically suspended mass shown in the figure below



$$m\ddot{d}(t) = -mg + \alpha \frac{I(t)^2}{d(t)^2}, \quad d(t) > 0 \quad (3)$$

where $d(t)$ is the position of the mass, $I(t)$ is the magnitude of the current through the electromagnet, m is the mass, g is the acceleration due to gravity, and α is a positive constant. *This magnet exerts a repulsive force on the mass to keep it suspended while the previous one exerted an attractive force.* Note that the electromagnet's position is fixed, which determines the origin $d(t) \equiv 0$. The position $d(t)$ is measured with an optical sensor, so it is an output. Finally, it should be emphasized that the model is valid only for positive values of d . The actuation signal or control command is the current $I(t)$.

1. Express the ODE in state space form $\dot{\bar{x}} = f(\bar{x}, \bar{u})$ where $\bar{u}(t) \triangleq I(t)$. Define your state vector \bar{x} explicitly.
2. Determine all the equilibrium points $(\bar{x}_{eq}, \bar{u}_{eq})$ of this system.
3. Suppose the desired position of the mass is d_0 (some positive constant). Determine the equilibrium point $(\bar{x}_{eq}, \bar{u}_{eq})$ corresponding to this desired position.
4. Linearize the system around the equilibrium point corresponding to the desired position d_0 . Is this linearized system internally stable/asymptotically stable? Do you need to know the numerical value of any parameter to answer this question?
5. Is the equilibrium point $(\bar{x}_{eq}, \bar{u}_{eq})$ (corresponding to the desired position d_0) stable, asymptotically stable, or unstable (in the sense of Lyapunov), or is it not possible to say from the linearized system?

Hint: the linearized system in the previous problem (attractive electromagnet) is unstable, while this one (repulsive) is stable.

Problem 6 (A more realistic electromagnet model). In practice, the current cannot be directly commanded, it is generated by using a voltage source. The voltage of the voltage source, $V_s(t)$, can be commanded to take any desired value within a range¹. The coil in the electromagnet acts as an inductor and resistor connected in series, forming a so-called R-L (resistor inductor) circuit. The equation for the electrical circuit is

$$V_s(t) = L \frac{dI(t)}{dt} + I(t)R \quad (4)$$

where R is the electrical resistance and L is the inductance of the coil. So, the actuation or control command is V_s , the current I is in fact an internal state. Consider such a current source powering a “vertically up” electromagnet of Problem 5, whose position is governed by

$$m\ddot{d}(t) = -mg + \alpha \frac{I(t)^2}{d(t)^2} - \beta \dot{d}(t) \quad (5)$$

where $\beta > 0$ is a drag coefficient.

¹There are many ways of doing it. One can connect a battery with a constant voltage to a voltage divider. A better method is to use a solid state voltage regulator in place of the voltage divider, which uses pulse width modulation to deliver an effective voltage that is equal to the commanded voltage $V_s(t)$.

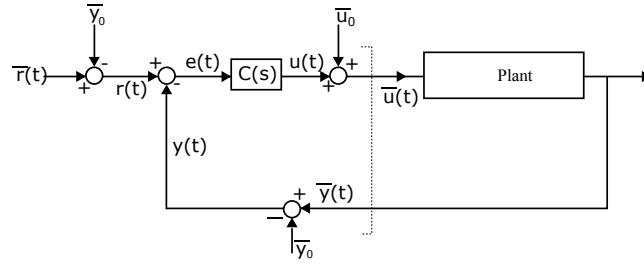


Figure 1: Control system: left of the dashed line is the “controller”.

1. Derive a state-space model of the system (that is, equations (4)-(5)) the form $\dot{\bar{x}} = f(\bar{x}, \bar{u})$, $\bar{y} = h(\bar{x}, \bar{u})$, where $\bar{u} \triangleq V_s$ and $\bar{y} = d$. Use the following state: $\bar{x} \triangleq \begin{bmatrix} d \\ \dot{d} \\ I \end{bmatrix}$.
2. Compute all the equilibrium points of the system corresponding to a constant position d_0 . (*both the equilibrium position and the equilibrium current have to be positive.*)
3. Derive an LTI approximation around this equilibrium point. Is this equilibrium point stable, unstable, or “not possible to say from the linearization”?

Problem 7 (Closed loop electromagnet). Consider the “vertically up” electromagnet model described in the previous problem. Suppose we wish to use a dynamic compensator to control the mass position. The reference for the mass position is denoted by $\bar{r}(t)$. Define the deviation control input $u(t) \triangleq \bar{u} - \bar{u}_{eq}$, where \bar{u}_{eq} was the input part of the equilibrium point you derived in the previous problem (corresponding to the desired position being d_0). The control command $u(t)$ is to be computed by a linear dynamic compensator $C(s)$:

$$U(s) = C(s)E(s), \text{ where } e \triangleq r(t) - y(t), y(t) \triangleq d(t) - d_0, r(t) \triangleq \bar{r}(t) - d_0$$

Figure 1 shows the control architecture (discussed in class) for implementing the controller $C(s)$ to control this nonlinear plant

1. If a compensator $C(s)$ is provided to you, is it possible to implement the controller (in hardware, not simulation) without knowing the the parameters m, α, d_0, L, β ?
2. Suppose $\alpha = 10^{-4} \text{ Nm}^2/\text{A}^2$, $m = 0.5 \text{ Kg}$, $g = 9.81 \text{ m/s}^2$, $R = 50\Omega$, $L = 0.015 \text{ H}$ (Henry), $d_0 = 0.1 \text{ m}$, $\beta = 0.5 \text{ kg/s}^3$. Design a compensator $C(s)$ so that the desired equilibrium point of the *closed loop* nonlinear system is asymptotically stable. (The “desired equilibrium point” is the one that corresponds to the position being d_0 . *Hint: use MATLAB[®]’s sisotool to design a controller to make the linear closed loop $CP/(1+CP)$ stable, where P is the linearized plant. Then argue why that also makes the equilibrium point of the nonlinear closed loop stable. Remember that the plant $P(s)$ only exists in your mind, the actual plant is nonlinear.*)
3. Perform a closed-loop simulation with the nonlinear plant and your controller. A *SIMULINK[®]* model of the plant is provided in *plant_ElectromagnetUpward.slx*. You have to add to it to make it look like Figure 1. The blocks inside the plant expects you to define the parameters m, α

etc., so be consistent. Use the reference to be $\bar{r}(t) = d_0$, and start with the initial condition $\bar{x}(0) = \bar{x}_{eq}$. Then vary the initial condition to see how well the closed loop behaves as the initial condition varies away from the equilibrium point. Once it varies sufficiently, the non-linear closed loop system may become unstable. If that happens, comment on how different $\bar{x}(0)$ could be from the equilibrium point before you observed instability.

What to submit For part 1, submit a brief explanation. For part 2, specify the controller $C(s)$ and how you obtained it (a sentence summary will suffice, such as “PID tuning method in sisotools”), the closed loop simulation with the linear plant P , and the Bode plot of the closed loop transfer function, a Bode plot of the loop gain L with gain and phase margins marked, and for Part 3, provide simulations plots with $\bar{r}(t)$ and $\bar{y}(t)$ vs. t , and $\bar{u}(t)$ vs. t . Don't forget to add the discussion about the initial condition mentioned in part 3. The discussion of the comparison between the actual nonlinear system and should be based on a figure that shows (in the same figure) $\bar{y}(t)$ from the simulation of the closed loop nonlinear system and $\bar{y}(t)$ that is predicted by the closed loop with the linearized plant (i.e., compute $y(t)$ from the closed loop system with plant $P(s)$ and controller $C(s)$, and add d_0).

OPTIONAL (NOT GRADED)

Problem 8. Consider the following non-linear ODE model for the spread of Coronavirus in a population:

$$\frac{d}{dt}I(t) = \frac{\theta}{N}(N - I(t) - R(t))I(t) - (\mu + \sigma)I(t) \quad (6)$$

$$\frac{d}{dt}R(t) = \sigma I(t) \quad (7)$$

Read the document `CoronaVirusModel.pdf` in e-learning Files > Resources > coronaVirusNotes to understand the model, and then use a first order Euler method to discretize the ODEs and numerically compute the solution². Note that if the ODE is $\dot{x} = f(x)$, the first order forward Euler is $x_{k+1} = x_k + \Delta t f(x_k)$.

Tasks:

1. Look at the original paper by Gotz where the model came from to obtain the values of the parameters θ, μ, σ, N . For θ , use the value 0.3056. Use the initial condition $I(0) = 800, R(0) = 0$.
2. Determine the unit of θ .
3. (“Extra credit”) Repeat the first exercise for the time-varying model of Gotz (see eq. 3 in the `CoronaVirusModel.pdf`) to produce the solution that Gotz obtained, that is shown in Figure 2 of the document (Figure 7 of the paper by Gotz). You will have to estimate θ_1 from the data he provides, or make an educated guess. The value of θ_2 can be obtained from the last equation of Gotz’s paper.

²The notes were written in March/April 2020; I have not updated them. But the points I am making in the notes remain valid today. You will also notice that the model described above is slightly different from that reported by Gotz in his paper. That is because when I simulated the model myself, my plot did not look anything like what Gotz reported in his paper. So I emailed him asking what the heck is going in. And he replied, saying, “sorry, there was a typo: θ in 2(a) should be θ/N ” !!

Plot your solution and compare with the data on “Cumulated cases” from China that Gotz uses. (see again Gotz’s paper for the definition of “Cumulated cases”.) If your plot looks quite different from Gotz’s, try tuning the the value of $R(0)$ to a small positive number (that is consistent with the definition of $R(t)$ and $I(t)$) to see if you can get results that are close to Gotz’s. *Beware of the unit of t : some automatically assume t is in seconds in performing simulations, which is not correct. Here that mistake will cost you. The units of the parameters and the x -axis of the plots in Gotz’ paper will help you in figuring out the time unit.*