Lab 3 – ECE311 Introduction to Linear Control Systems

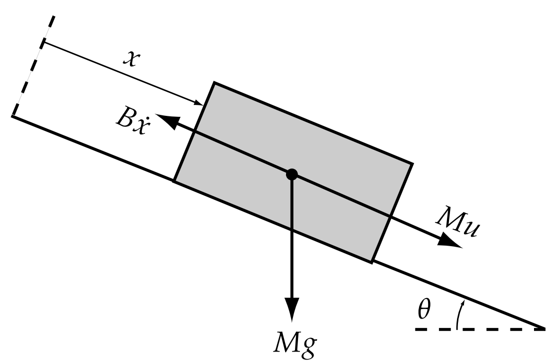
Speed Control of a Simplified Car Model

Main concepts of this lab

- How to apply the Internal Model Principle for speed control of a car model
- How to convert transient performance specifications to desired pole locations

1 Introduction

Imagine a car moving along a straight road with an unknown slope. Neglect the moment of inertia of the wheels, and suppose the friction force is proportional to the speed of the car (i.e. the friction is viscous). Finally, suppose the engine imparts an *acceleration* u (measured in $\mathrm{m/s^2}$) to the car. This is our control input. The schematic representation of the system just described is depicted below.



Its mathematical model is

$$M\ddot{x} = -B\dot{x} + Mu + Mg\sin(\theta).$$

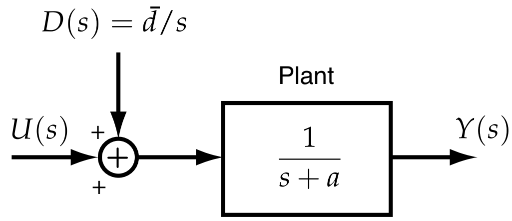
Here, x represents the displacement of the car measured with respect to an inertial frame, and θ is the unknown inclination angle of the road. Since we are interested in controlling the speed $\dot{x}$ of the car, we rewrite the model using $v = \dot{y}$,

$$M\dot{v} = -Bv + Mu + Mg\sin(\theta),$$

and we let $y = v$ denote the output of the plant. Here we are assuming that we have a speed sensor (a tachometer). Denoting $\bar{d} := g\sin(\theta)$, we rewrite the model as

$$\begin{aligned} \dot{v} &= -\frac{B}{M}v + \left(u + \bar{d}\right) \\ y &= v. \end{aligned} \tag{1}$$

Since θ is unknown, so is the constant $\bar{d}$, and therefore we regard it as a disturbance. Letting $a = \frac{B}{M}$, we obtain the plant block diagram depicted below.



In the figure, $D(s)$ represents the disturbance, and we note that it has the expression $D(s) = \bar{d}/s$ because the Laplace transform of a constant $\bar{d}$ is $\bar{d}/s$. The numerical values used in this lab are listed in the table below.

Parameter	Description	Numerical value
M	Car mass	1000 (Kg)
B	Friction coefficient	1 (N $\cdot$ s /m)
g	Acceleration due to gravity	9.81 m/sec^2
a	Transfer function parameter	B/M

The control objective is to make the car speed y converge to a desired constant speed v_{des}, irrespective of the unknown constant disturbance.

Note: This lab was developed using Matlab R2020b. If you use an earlier version of Matlab there is a very small chance that some of the instructions here might not work for your version, in which case you should update your software.

Throughout the lab, you will be guided through a number of steps which will require you to write Matlab code or draw Simulink diagrams. You will write your code in a Matlab script called `labx.m`, where `x` is the lab number. Your Simulink diagram will be saved as `labx.slx`. If there are multiple Simulink files, save them as `labx_1.slx`,`labx_2.slx`, and so on. You will submit this code as a group. Your code should provide certain outputs (figures, numbers, and the like). A request for output is highlighted with a shaded area of text, such as the following.

Output 1. Print the poles of the transfer function.

Parts of the text containing directions for the writing of your Matlab code will be highlighted in a different colour, such as this:

Convert the state space model into a transfer function model, then find the poles.

2 Submission guidelines and mark breakdown

Marks are assigned to groups. Members of each group get identical marks, unless special circumstances occur. The group lab mark will be made up of three components.

Matlab code	4 pts
Presentation video	2 pts
Lab report	4 pts
Total	**10 pts**

Matlab code. Your code should be clean and readable, and it should contain abundant commentary, so that the instructors may follow your development and check its correctness. This component of the mark will be based on the correctness of the code and its readability, and it will be assigned as follows:

0 out of 4	the code is absent or largely incomplete
1 out of 4	there are parts missing in the code and the outputs are incorrect
2 out of 4	the code is complete, it produces correct outputs, but there is no commentary in the code, and/or the code is poorly organized
3 out of 4	the code is complete, correct, and contains some but insufficient commentary, and/or its organization is below par
4 out of 4	the code is complete, correct, and the commentary and organization are adequate so that it is easy to read

Presentation video. Once you've completed your lab code, you will submit a 5-6 minute video presentation of the code as a group. Each group member will present a different portion of the code. The objective of the presentation is to show that you understand what you did and why you did it. It's important that each group member contributes equally to the video. In the video, include any observations about the outputs you produced, any insight you derived from the lab steps. You need to display an understanding of what the lab is about, what concepts it covers, and specifically address in the video any request for comments included in the lab document. For example, if the lab document asks you to verify that a transfer function is BIBO stable, you need to show an understanding of the property of BIBO stability.

0 out of 2	group does not submit a presentation, or the presentation is unintelligible
1 out of 2	the presentation does not display adequate understanding of the code and/or the lab document; or the group members give somewhat disconnected presentations; or requests for comments in the lab document are not adequately addressed in the lab video; there does not emerge an adequate understanding of what the lab is about
2 out of 2	the presentation is intelligible, the code is adequately presented, the requests for comments in the lab document are adequately addressed, and there emerges an overall understanding of what the lab is about

Lab report. Write a concise report containing the requested outputs, but don't just print out a bunch of numbers and figures. Add some flesh to the outputs that are requested within the lab document so that one can follow the logical development of the lab. Aim for a style like this:

> **Output 1.** Below is a plot of the output signal that was obtained with this controller.
> (...)
> We observe that the output signal converges to a steady-state value of 25 rad/sec (...)
>
> **Output 2.** Tuning of the controller gain gave the following result, see the figure below illustrating the step response of the system. We observe that there is marked improvement in the transient performance of the control system, and indeed (...)

Do not screenshot Matlab figures. Rather, save them as jpeg files (or other formats) and include them in the report in the appropriate order with clear titles and axes labels.

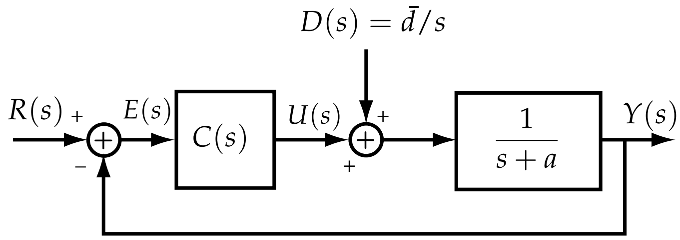
When the lab document asks you to comment on something, strive to provide meaningful, insightful commentary. Make sure you display an understanding of what the lab is about. For example, if you are asked to comment on the experimental observation that a linear controller fails to stabilize a nonlinear system far from the equilibrium, you need to give a convincing explanation of the theoretical phenomenon grounded in the theoretical principles that you've learned in class, starting from the fact that the linearization constitutes an approximation of a nonlinear function. This portion of the mark will be assigned as follows:

0 out of 4	the report is either absent, or a simple printout of the Matlab outputs without commentary
1 out of 4	the report is incomplete and/or the commentary is inadequate
2 out of 4	the report is complete and the commentary reveals a misunderstanding of what the lab is about; or formatting specifications for the outputs are not met
3 out of 4	the report is complete; the formatting specifications for the outputs are met; the commentary is somewhat accurate but there does not emerge a full understanding of what the lab is about
4 out of 4	the report is complete; the formatting specifications for the outputs are met; the commentary is accurate and there emerges an understanding of what the lab is about

Late submissions. *All submissions are due at 8PM* of the day indicated on the course website. We *do not* accept late submissions under any circumstances. For more details, see the *late submission policy* on the course website.

3 Asymptotic tracking with disturbance rejection: theoretical analysis

Consider the basic feedback loop depicted below.



In this section, you will design a controller $C(s)$ meeting the five specifications below.

SPEC 1 The output $y(t)$ of the closed-loop system should asymptotically track reference signals of the form $r(t) = v_{\text{des}}\,\mathbf{1}(t)$ (where v_{des} is the desired speed in m/s), despite the presence of the unknown disturbance $d(t) = \bar{d}\,\mathbf{1}(t)$.

SPEC 2 The closed-loop system (input $R(s)$, output $Y(s)$, and no disturbance, $D(s) = 0$) should be BIBO stable.

SPEC 3 All poles of the above closed-loop system should lie on the real axis, so that the output $y(t)$ does not have oscillatory behaviour.

SPEC 4 When $D(s) = 0$ (i.e., when the road is horizontal), the settling time T_s should be less than 6 seconds.

SPEC 5 When the desired reference speed is $v_{\text{des}} = 14$ m/s (i.e., 50 Km/hour) and $D(s) = 0$, the control input signal $u(t)$ should not exceed the bound $|u(t)| \leq 30\ m/s^2$.

Before proceeding, you need to review your lecture notes on the Internal Model Principle. We will make use of these concepts now.

By the Internal Model Principle, to meet SPEC 1 the controller must have a pole at zero (you will be asked to justify this in Output 1), so we will design a PI controller

$$C(s) = K\frac{T_I s + 1}{T_I s}, \tag{2}$$

where $K, T_I > 0$ are two design parameters that we need to select. In the next box, you will be guided through pen and paper calculations that you'll need to perform in preparation for your Matlab code.

Output 1.
- Appealing to the Internal Model Principle, show that in order to meet SPEC 1, $C(s)$ must have a pole at zero.
- The PI controller (2) has a pole at zero. Give a clear explanation of the Internal Model Principle, and clearly show that it implies that if the gang of four transfer functions (see lecture notes) are BIBO stable, then SPEC 1 and SPEC 2 are met.
- A necessary and sufficient condition for the gang of four transfer functions to be BIBO stable is that

(a) all poles of the transfer function $1/(1 + CG)$ have negative real part, and

(b) the product $C(s)G(s)$ has no pole-zero cancellations in the closed right-half plane.

Show that for any $K, T_I > 0$ the PI controller above satisfies conditions (a) and (b). Present clear and concise arguments.

- Now we need pick $K, T_I > 0$ to meet SPECS 3,4 and 5. We begin with SPEC 3. Find the transfer function $T(s) = Y(s)/R(s)$ (assuming $D(s) = 0$). In order for this transfer function to have negative real poles at $-p_1, -p_2$, its denominator must have the form $s^2 + (p_1 + p_2)s + p_1 p_2$. By equating coefficients, find the unique values of $K, T_I > 0$ such that the poles of $T(s)$ are $-p_1, -p_2$. Present a clear derivation of the relationship between (K, T_I) and (p_1, p_2).

  You've now met SPEC 3, and you can still choose $p_1, p_2 > 0$ to meet SPECS,4 and 5. You will next work on SPEC 5. SPEC 4 will be addressed via tuning in simulation.

- Now we investigate SPEC 5. The car is initialized at zero speed (because we are modelling it as a transfer function), and is asked to accelerate and reach a speed v_{des}. It is natural to assume that the largest acceleration will occur at $t = 0$. Assuming that this is the case (you'll verify this assumption later via simulation), we need to guarantee that $|u(0)| < 30$ m/s^2. For this we use the following result.

  Initial Value Theorem (IVT): Suppose that $F : \mathbb{C} \to \mathbb{C}$ is a rational and strictly proper function, and denote $f(t) = \mathscr{L}^{-1}(F(s))$. Then $f(0) = \lim_{s\to\infty} sF(s)$.

  Find the transfer function $U(s)/R(s)$, and setting $R(s) = v_{\text{des}}/s = 14/s$, compute $U(s)$. Then, using the IVT, compute $u(0)$ as a function of K and T_I. In turn, K and T_I depend on p_1 and p_2. Find the constraint that p_1 and p_2 must satisfy in order that $|u(0)| < 30$. Write out this constraint neatly. It should be an upper bound on the sum $p_1 + p_2$. This upper bound should depend on the constant a.

We end this section with a remark about SPEC 4. You might be wondering why in the above theoretical analysis we did not attempt to meet this specification using the formula for settling time $T_s \approx 4/(\zeta\omega_n)$. The reason is that this formula is only valid for complex conjugate poles, whereas in SPEC 3 we require the poles to be real.

4 Asymptotic tracking with disturbance rejection: numerical simulation

In this section you will use your theoretical analysis of Section 3 to simulate a PI controller solving the constant reference tracking problem with disturbance rejection.

Matlab commands

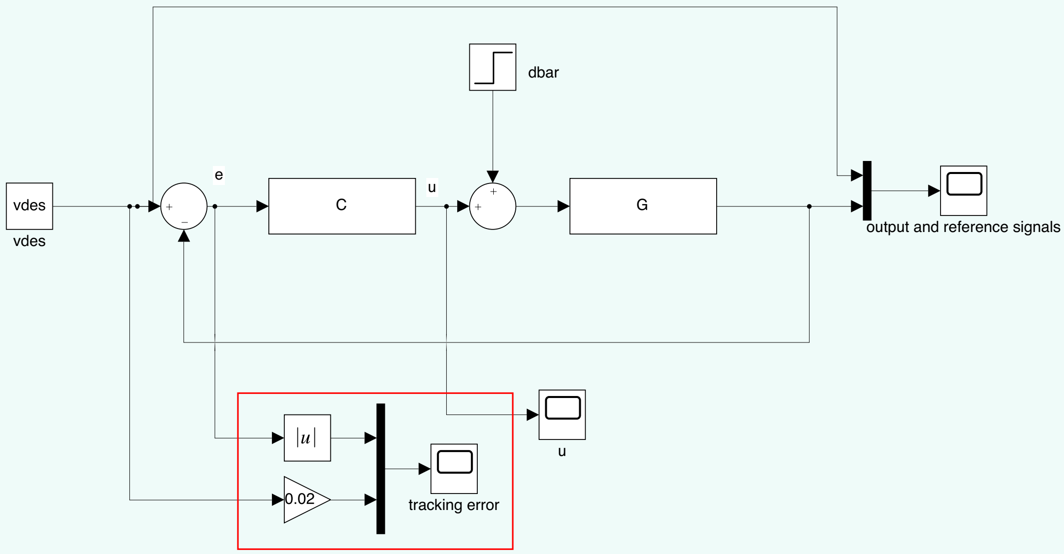
`h=stepinfo(G)`	Determines the transient performance information from the step response of the LTI object `G`. The variable `h` is a structure containing various performance parameters
`minreal(G)`	Performs any pole-zero cancellations in a transfer function LTI object

Simulink blocks (inside the Library Browser)

`Sources` → `Step`	Step function
`Math operations` → `Abs`	Absolute value
`Math operations` → `Gain`	Constant gain

Create a script named `lab3.m`.

- Define variables `M,B,g,a` with numerical values given in the table in Section 1. Define a variable `vdes` with numerical value 14, and a variable `dbar = g*sin(theta)`, where `theta` is assigned a numerical value of $-\pi/6$ (i.e., the road will be inclined uphill).
- Define an LTI object named `G` with the plant transfer function $G(s) = 1/(s+a)$.
- Define variables `p1,p2` containing initial values for the poles of the closed-loop system meeting specification `SPEC 5` as per your development in Section 3. You will tune `p1,p2` later to meet `SPEC 4`. Define PI control parameters named `K,TI` in terms of `p1` and `p2`. Define an LTI object named `C` with the controller transfer function $C(s)$ in (2).
- Using the LTI objects `C` and `G`, define an LTI object named `T` containing the transfer function $Y(s)/R(s)$ (assuming $D(s) = 0$). Using the command `minreal`, perform any pole-zero cancellations in `T`. Then, extract the poles of `T` (see Lab 1 for how to do this) and verify that they coincide with $-p_1, -p_2$.



Draw the Simulink diagram displayed above, and name it `lab3_1.slx`.

- In the block diagram, there are two `LTI System` blocks pointing to the objects `G` and `C` that your script `lab3.m` has defined in the workspace. There is a `step function` block to generate the disturbance signal, and three scopes to measure the output and reference signals, the tracking error, and the control input signal.

- Observe the area of the block diagram delimited by a red box. Here we are comparing two signals: the absolute value of the tracking error $e(t)$ and a the constant signal $0.02 \cdot v_{\mathrm{des}}$. The settling time is the first time when the former signal decreases and stays below the latter one.

- Open the `Model settings` and under the `Solver` menu, set both relative and absolute tolerances to 10^{-10}. Make sure that the solver is `variable-step`, and set the `stop time` to 30 seconds.

- Open the `Step function` block, set `step time` to 15 seconds, `initial value` to 0, and final value to `dbar` (this variable was defined in your script). In words, the road will be flat for 15 seconds, after which the inclination will change to $-\pi/6$ (i.e., the car will drive uphill; in practice the road inclination would change gradually). The car speed will clearly be affected by the sudden change in road inclination, but your controller should be able to recover from it.

- Run the simulation and save three figures generated by the three scopes in the block diagram. Make sure that the tracking error scope is configured as in the block diagram. From this latter scope, deduce the settling time over the time interval 0 to 15 seconds, and check whether it is below 6 seconds. From the control signal scope, check whether our assumption in Section 3 that $u(t)$ would achieve its maximum value at $t = 0$ is correct. Verify that $|u(t)| < 30$. It should be, if your theoretical analysis of Section 3 is correct, and if $p_1 + p_2$ satisfies the bound you found in that section.

- Tune the poles p_1 and p_2 (and correspondingly the PI control parameters K and T_I) in your `lab3.m` script, rerun the code, and then rerun the Simulink diagram. The objective of your tuning is to get the *smallest* settling time over the time interval 0 to 15 seconds that does not violate the control bound of `SPEC 5` over that interval.

- Now return to your Matlab script `lab3.m`, and recall that you've defined an LTI object `T` containing the transfer function $Y(s)/R(s)$ assuming that $D(s) = 0$ (which is the case over the time interval 0 to 15 seconds). Using the command `stepinfo`, find the settling time of $y(t)$, and compare it with the one you found above (over the interval 0 to 15 seconds) for the optimal values of p_1 and p_2. You should get the same settling time. If not, you have an error, find it and fix it.

Output 2.

- Print your initial choices of p_1, p_2, and the corresponding values of K and T_I. Produce three plots corresponding to the three scopes in the Simulink diagram, and estimate the settling time based on the system response over the first 15 seconds. Then, estimate the settling time after the disturbance kicks in at $t = 15$ s. Your plots should have titles describing their content.

- Comment on your initial findings before tuning. Are all the five specs met? Which ones aren't? Guide the reader of your report through the verification of which specs are met and which ones aren't. Explain the tuning you plan to do to improve the performance, and show your reasoning behind it.

- After tuning, print the values of p_1, p_2 minimizing the settling time while respecting the bound on $u(t)$. Repeat again your analysis verifying whether all the specs are met. What is the best settling time you could get over the time interval 0 to 15 seconds? What is the settling time over the time interval 15 to 30 seconds? Print the settling time you found using `stepinfo` and verify that it's the same as the one over the time interval 0 to 15 seconds that you deduced from the plots.

- You will notice that, after the disturbance signal is enabled, the control signal $u(t)$ will settle to a nonzero value. Find this value, and explain how it conforms with the physics of the problem (a mass on an inclined plane subject to gravity).