Xi'an Jiaotong-Liverpool University
西交利物浦大学

Department of Mechatronics and Robotics

MEC208 Instrumentation and Control System

S2, 2024-2025

Computer Lab (Lab 2):

Control System CAD and CAS using MATLAB

Schedule and Deliverable

Lab work release date: **17/18th April 2024, Thursday/Friday of Week 9**

Computer lab session: On-site computer lab session

Lab Group	Date	Time and Venue
1	**28th April 2025, Monday (Week 11)**	**2:00 pm – 5:50 pm: SIP-CBG13**
2		**2:00 pm – 5:50 pm: SIP-IR112**
3		**2:00 pm – 5:50 pm: SIP-CBG15E**
4		**2:00 pm – 5:50 pm: SIP-SD546**
All groups (1, 2, 3, & 4)	**30th April 2025, Wednesday (Week 11)**	**2:00 pm – 4:00 pm: SIP-IR112 (optional session, mainly for those who need more lab hours or on-site consultation)**

You are encouraged to start working on the computer lab tasks/works starting from Week 9. During the on-site lab sessions in Week 11, teaching assistants will be present for lab supervision and Q&A.

Lab report submission deadline: **11th May 2025 11:59pm, Sunday of Week 12**. The LMO submission link in located in MEC208 course page. **No handwritten report, but only computer typeset report will be accepted.**

Facility to prepare

Equipment: Lab computers or own PCs (read more below)

Software: *MATLAB*

Additional notes on installing MATLAB software in your own PCs

Students can install MATLAB software in their personal computers, using XJTLU university license (i.e., students will need register a MATLAB account using XJTLU email). There are two ways of accessing the software:

(i) install *MATLAB* software in your local PC drive. You may follow university IT guideline at https://guide.xjtlu.edu.cn/how-to-register-and-install-matlab-license-for-student.html (reminders: choose “Control System Toolbox” and “Symbolic Math Toolbox”);

(ii) access *MATLAB Online* without installation. Read the above link for account registration, and then visit https://www.mathworks.com/products/matlab-online.html for steps to access MATLAB Online.

Introduction

This laboratory work/assignment carries 15% of MEC208 module mark. The work is about learning the engineering software tool to conduct classical control system design and analysis. Among the several options available, MATLAB is chosen as the main software. To help with your learning, the document first sets out the "Examples" that show some of the very typical commands of the Control System toolbox of MATLAB. You are strongly encouraged to practice them yourself and explore other control design functions mentioned in lecture notes and other resources (e.g., the Help Files of MATLAB).

Once you are familiar with the software environment, you can then start to attempt and solve the control problems. You will need to use the theoretical understanding of Classic Control Theory that you acquired so far, and do appropriate problem analysis, mathematical derivation, and deploy suitable numerical tools/functions. At the start of every question, you are encouraged to briefly explain your approach; for sub-parts that need your analysis, please elaborate them concisely.

You are required to submit a well-formatted Lab Report before the deadline (as above). The university's late submission policy applies to submissions after the deadline. Please refer to LMO module page for the submission steps.

You are reminded about the university's academic integrity policy. If you undertake the preparation with other students, or if you use sources from the internet or books, you must acknowledge these in your report by writing a short statement at the beginning of the report.

About GenAI - According to module design and school policy, students are **not permitted** to use Generative AI (GenAI) to substitute the original work expected from students. **Using the GenAI-generated content and attempting to declare it as own work is strictly prohibited**. **AI-generated solutions (well beyond paraphrasing), as revealed by the index determined by software checker in LMO, will be strictly dealt with according to the University Academic Integrity Policy.** Students may refer to UoL guideline published at *https://www.liverpool.ac.uk/centre-for-innovation-in-education/digital-education/generative-artificial-intelligence/* for more guidance. However, students can, and should, use GenAI to assist their own learning of the learning subject. Taking the control part of this module as an example, students can use GenAI to improve their genuine understanding on control system knowledge, such as "root locus and the fundamental principles", "frequency response and their usage", "stability analysis and their industrial relevance", etc. Students are reminded that GenAI tools are not academic sources; they do not produce fact-checked content [UoL guideline]. In the GenAI era, it is crucial for students to emphasize more on **true conceptual understanding** in order to form a solid knowledge foundation to support his/her own future learning of more advanced topics, and remain competitive in his/her career to provide meaningful and forward-looking engineering solutions to the society.

Aims, Learning Outcomes and Outline

This laboratory work aims to enable the student to:

- Understand the concept of open-loop and closed-loop control systems.
- Understand the mathematical modelling of dynamic systems and different formats of system models.
- Learn computer aided system analysis and design.
- Understand the system stability and how to determine if a system is stable.

Having successfully completed the work, you will be able to:

- Use computer software to assist in dynamical control system design.
- Analyze first- and second-order system responses, and transfer function through Laplace transform.
- Determine system performance and stability with Root Locus Analysis.
- Design feedback controller to stabilize overall system and to meet design specification.

Report Format

The format of the lab/assignment report should be of the following:

- The report should be of **single column**, **font size 11**, font style "**Times New Roman**". Please be reminded that handwritten reports will **NOT** be accepted.
- All figures and plots must be clearly labelled.
- In your report, you are expected to explain concisely your approach (i.e., with full sentence and proper grammar, but please avoid unnecessarily long explanation).
- In your answer for each question, do also include all MATLAB code/script that you used to obtain the answers or plots (note: do not separately submit those raw .m files; answer for each question should contain some MATLAB scripts). **These codes/scripts should be copied properly into the report and should remain standalone**. For example, if you include the code for Question 2a, you must make sure that the codes in this part alone (not linked to other questions) can be copied by the marker into MATLAB for a quick verification.
- For analytical questions, whenever necessary, do also include your analysis and derivation.

Marks will be given on the following basis:

- The 6 questions in the "Computer Lab Work" section carry a total of 100 marks.
- Sub-marks will be awarded to technically correct analyses and comments, MATLAB scripts, mathematical derivation (as necessitated by the questions), graphs/plots (and annotations of the plots), and/or numerical answers. All analysis and comments should have proper sentence structure and correct grammar.

Examples

This part is to introduce about how to start using the Control System Toolbox in MATLAB. You may go through the examples (in this section, and in other online resources) before you start working on the assignment tasks.

Example 1: Input a system described by a transfer function

To construct a system model in MATLAB, the command "*tf*" can be used. For example, to obtain the following input transfer function:

$$\frac{s^2+s+1}{s^4+2s^3+5s+2}$$

$$\frac{s+1}{s^4+2s^3+3s^2+3s+10}\cdot\frac{10}{s^2+3}$$

```
» sys1=tf([1 1 1], [1 2 0 5 2])
Transfer function:
s^2 + s + 1
---------------------
s^4 + 2 s^3 + 5 s + 2
```

```
>> sys1=tf([1 1], [1 2 3 3 10])
Transfer function:
s + 1
-----------------------------
s^4 + 2 s^3 + 3 s^2 + 3 s + 10
>> sys2=tf([10], [ 1 0 3])
Transfer function:
10
-------
s^2 + 3
>> sys=sys1*sys2
Transfer function:
10 s + 10
-----------------------------------------------
s^6 + 2 s^5 + 6 s^4 + 9 s^3 + 19 s^2 + 9 s + 30
```

Example 2: Find the system's zeros and poles

The commands used for this task are "pole" and "zero". From the signs of system poles, you can determine if the system is stable. For example, given the system $G(s)$:

$$G(s)=\frac{1}{s^2+s+1}$$

Using MATLAB, the system's poles can be obtained as follows:

```
>> pole(sys)
```

```
ans =
-0.5000 + 0.8660i
-0.5000 - 0.8660i
>>
```

Given another example with system zeros:

$$G(s) = \frac{2s^2 + s + 3}{s^4 + 2s^3 + 5s^2 + 6s + 7}$$

```
>> sys=tf([2 1 3],[1 2 5 6 7])
Transfer function:
2 s^2 + s + 3
-----------------------------
s^4 + 2 s^3 + 5 s^2 + 6 s + 7
>> zero(sys)
ans =
-0.2500 + 1.1990i
-0.2500 - 1.1990i
>>
```

Example 3: Obtain the system model in pole-zero format

It is at times helpful to control designers to express a system into a form of first/second-order components. For example, given *G*(*s*) below, one can express it to pole-zero format as follows:

$$G(s) = \frac{1}{s^4 + s^3 + 3s^2 + 2s + 6}$$

```
 >> sys=tf([1], [1 1 3 2 6])
Transfer function:
1
---------------------------
s^4 + s^3 + 3 s^2 + 2 s + 6
>> zpk(sys)
Zero/pole/gain:
1
----------------------------------------------
(s^2 + 1.969s + 2.609) (s^2 - 0.9693s + 2.3)
>>
```

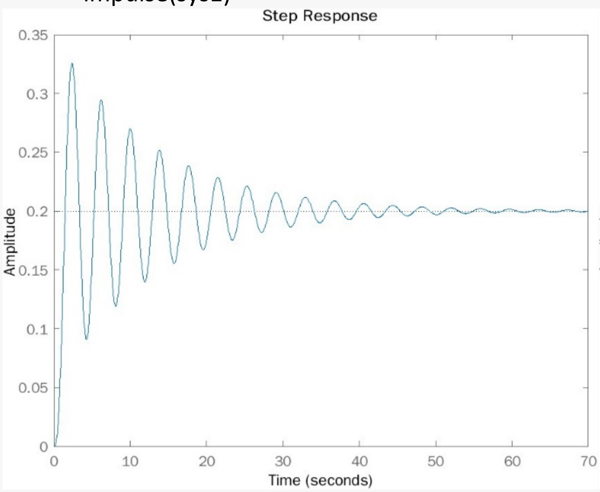
There are many other control-related numerical tools in MATALB. See some of the listed examples in Part II.

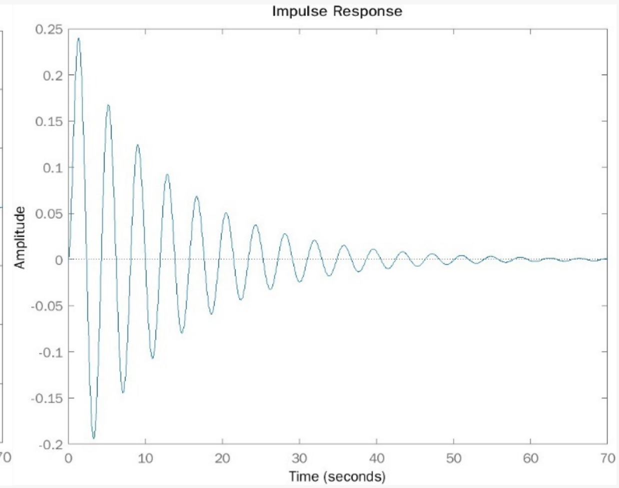
Example 4: Obtain impulse, step, and ramp input responses

The commands used for obtaining the time responses against unit-step and unit-impulse inputs are, respectively, “step()” and “impulse()”. Taking the following system as the example, its step and impulse responses can be obtained as follows:

$$\frac{1}{s^3 + 2s^2 + 3s + 5}$$

```
>> sys1=tf([1], [1 2 3 5])
Transfer function:
1
---------------------
s^3 + 2 s^2 + 3 s + 5
>> step(sys1)
>> impulse(sys1)
```



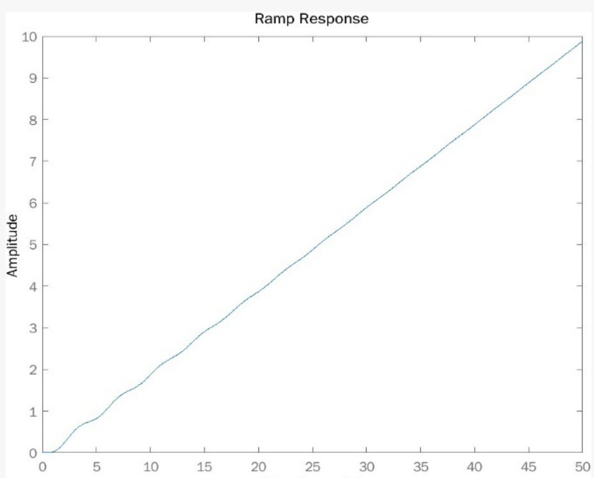


To obtain system ramp input response, the following can be done. Note that the title change is necessary to avoid confusion.

```
>> sys2=tf([1], [1 0])
Transfer function:
```

$\frac{1}{s}$

```
>> step(sys1*sys2)
>> title('Ramp Response')
```



Example 5: Obtain the equivalent/effective transfer function of a closed-loop system

The relevant command is "feedback". You can obtain the description of this command from MATLAB help, i.e., using command ">>help feedback". For example, given that a system has a forward-path transfer function $G(s)$

$$G(s) = \frac{1}{s^2 + s + 1}$$

and feedback-path transfer function $H(s)$

$$H(s) = \frac{1}{s + 1}$$

The closed-loop transfer function of the feedback system is obtainable through the following commands:

```
>> sys1=tf([1],[1 1 1])
>> sys2=tf([1],[1 1])
>>sys_closed_loop=feedback(sys1,sys2,-1)
Transfer function:
s + 1
--------------------
s^3 + 2 s^2 + 2 s + 2
>>
```

At this stage of learning, you should be able to obtain the closed-loop transfer function through analytical means. You are encouraged to check whether the numerical tools actually give you the correct answer. At times, through cross-checking, you may find that there are some minor mistakes in your own derivation or MATLAB scripts. You may also attempt the simulation questions at the end of each chapter in the course textbook.

Optional

Before attempting the questions, apart from the numerical tools introduced above, it may be helpful to you to spend some time understanding other numerical tools/functions (not in any particular order) available to you in the Control System Toolbox of MATLAB.

tf *step* *impulse* *ramp* *partfrac* *ilaplace* *plot* *feedback* *series*	*pzmap* *pole* *zero* *ss* *tf2ss* *lsim* *expm* *hold on* *hold off*	*Others (check in MATLAB documentation; there are many more…)*

Computer Lab Work (6 questions) typo corrected on 2025-04-20

Problem 1 [10 marks]

For each of the following transfer functions with unity feedback, express it into two equivalent state variable models/representations. For Question 1(a), compare the time responses of the state variable models towards unit step input; for Question 1(b), compare and comment their time responses towards unit ramp input. Clearly state the MATLAB functions/codes.

(a) $G(s) = \frac{s+3}{2s^3+3s+7}$

(b) $G(s) = \frac{1}{(s+2+5j)(s+2-5j)(s+9)}$

Problem 2 [15 marks]

Consider the block diagram in Figure P2.

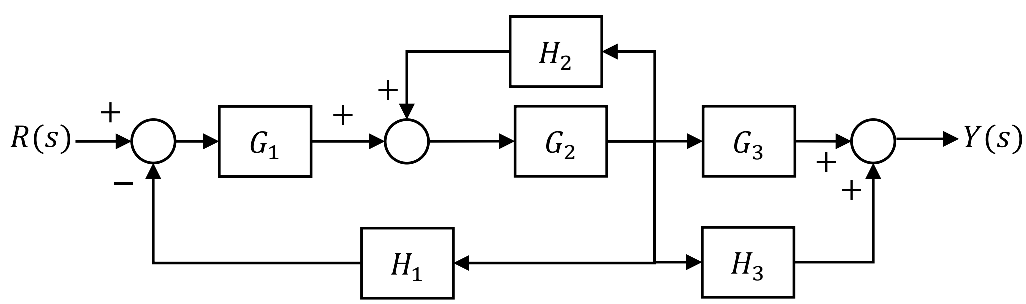


Figure P2: Control block diagram with the following *G* and *H* transfer functions.

$$G_1 = 5 + \frac{3}{s} \qquad G_2 = \frac{s+4}{s^2+2s+10} \qquad G_3 = \frac{1}{s^2+2s+5}$$

$$H_1 = \frac{s+1}{s+5} \qquad H_2 = \frac{ps+2}{s+3} \qquad H_3 = \frac{2}{s+4}$$

(a) Given that p = 5. Use an *m*-file to reduce the block diagram in Figure P2, obtain the equivalent closed-loop transfer function of the above system. Clearly show, using codes or drawings, the steps of block diagram simplification. Generate a pole-zero map in the graphical form and calculate the poles and zeros of the closed-loop transfer function. Comment on the overall system stability.

(b) If p increases gradually and it eventually reaches 10. Based on the new pole-zero map, comment on the changes to the zeros and poles, as well as the system stability.

Problem 3 [15 marks] Typo corrected on 2025-04-24. Change "-5" to "5".

Consider the system

$$\begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\\ \dot{x}_3(t)\end{bmatrix} = \begin{bmatrix}-1 & -1 & 0\\ 0 & -1 & 2\\ 5 & 0 & -3\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\\ x_3(t)\end{bmatrix} + \begin{bmatrix}0\\ 1\\ 1\end{bmatrix}u(t)$$

$$\mathrm{y}(t) = \begin{bmatrix}3 & 2 & 1\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\\ x_3(t)\end{bmatrix}$$

(a) Using the *tf* function, determine the transfer function $Y(s)/U(s)$.

(b) Assume that $u(t)$ behaves in this manner: start with zero magnitude/value at time $t = 0$ s, then step increase to 10-magnitude at $t = 5$ s, then decrease linearly to zero in 3 seconds starting from $t = 7$ s. Plot the response of the system with initial condition $x(0) = \begin{bmatrix}1 & -1 & 3\end{bmatrix}^T$, for $0 \leq t \leq 20$.

(c) Now, if the input $u(t)$ from part (b) remains the same except that it becomes zero for time t = 7 s to 10 s (means the decreasing slope disappears). Plot the response of the system towards the new $u(t)$ for the same initial condition $x(0) = \begin{bmatrix}1 & -1 & 3\end{bmatrix}^T$, for $0 \leq t \leq 20$. Comment on the most obvious difference between the output $y(t)$ responses from part (b) and part (c) and deduce the best possible explanation.

Problem 4 [20 marks]

Consider the following transfer function:

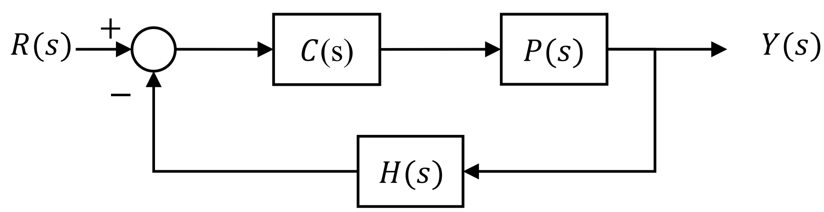


Figure P5

where $P(s) = \frac{2s+1}{s^3+4s^2+2s+7}$ and $H(s) = \frac{1}{s+3}$.

(a) A senior engineer has proposed that $C(s)$ is designed as continuous-time PI controller with proportional and integral gains. He/she chose the integral gain as 2, but is unsure of the value of the proportional gain K_p. Please analyze the poles trajectories for different K_p within 0 to 20, propose a suitable value of proportional gain that can deliver good response in practice.

(b) For the PI controller with the proposed K_p, identify the dominant complex poles of the system (if any), and calculate the dominant complex poles' natural frequency, damping factor, characteristic rise time (5 to 95%), peak time, percent overshoot and 2% settling time. Then, plot and clearly label the unit-step response of $Y(s)$.

(c) If you are given an opportunity to re-propose the type of the controllers (note: please limit the choices to P, PI, or PID] including the gain values in order to improve the system response further. What would be your proposal? Justify your answers with correct technical basis.

Problem 5 [20 marks]

Figure P5 shows a closed-loop system with a tunable feedback parameter k, where k is a positive number.

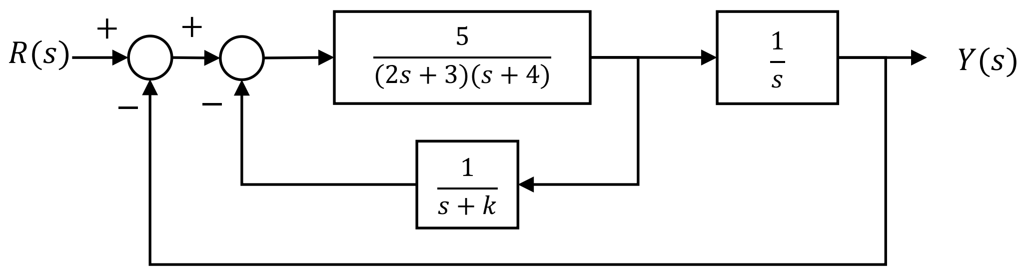


Figure P5

(a) Draw a root locus diagram to reflect the effect of changing k. Then, determine the range of possible k values that can produce a pair of dominant oscillatory/complex poles with the characteristic damping factor more than 0.7 and the characteristic 2% settling time less than 3 s.

(b) Analyze the plot in part (a) and propose the final value of k that can produce the quickest possible response. Plot the unit-step response of the output $y(t)$. Verify whether the final system response meets to the above requirement. Explain the discrepancy, if any.

Problem 6 [20 marks]

As the control engineer in Xiaomi's electric vehicle (EV) company, you are assigned tasks to analyze and propose a control solution to the upcoming new model YU7's drive-by-wire technology. The front and rear wheels are powered by two closed-loop controlled interior permanent magnet motors. The design task focuses on the new vehicle's turning dynamics. The overall vehicle's turning speed dynamics (simplified version) is depicted graphically in Figure P6. $\theta^{ref}(s)$ is the angular position reference input, and $\theta(s)$ is the actual angular position. You are expected to justify, in the form of textual description, **every relevant design step** to reach the final answer, using the control system understanding from the module (especially the system modelling and control part), mathematical derivation, simulation result, or any other means deemed appropriate.

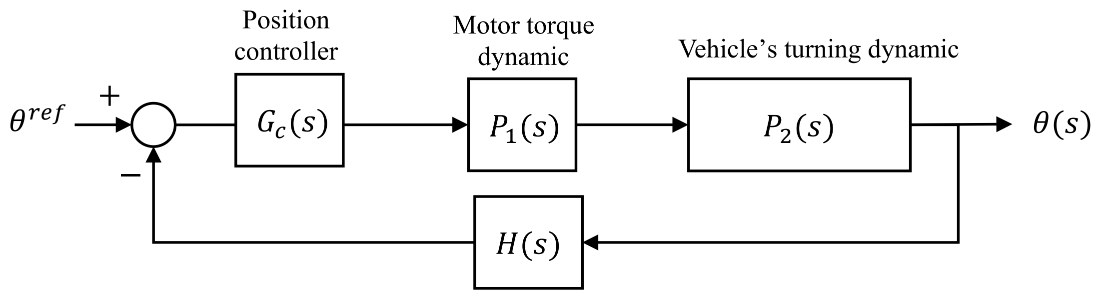


Figure P6

The plant models are as follow: the motor torque dynamics is simplified as $P_1(s) = \frac{A}{s+A}$, where $A = 10$; the vehicle's turning dynamic is $P_2(s) = \frac{s+D}{s^2+Bs+C}$, where $B = 1$, $C = 2$, and $D = 0.5$, which are estimated by analyzing the experimental data. The position feedback sensor has a dynamic of H (s), which is estimated be $H(s) = \frac{1}{0.5s+1}$. Your task is to propose a suitable form of controller $G_c(s)$ that can eliminate the steady-state angular position error due to step input and achieve a fast and comfortable turning so that the vehicle user experience can be improved. Explain your answer with sufficient evidence and suggest suitable $G_c(s)$ gain/parameter value(s) to ensure a fast closed-loop response with damping coefficient being in between 0.2 to 0.4 values. Plot the unit-step input response of the closed-loop system.

~~ The End. All the best! ~~