

MATLAB Control Toolbox Tutorial

Using MATLAB to Support Chapters 4–6

Feedback Characteristics, Performance, and Stability

Control Practice Session

October 10, 2026

1 Exercises (5 Problems)

Each problem is designed to be non-trivial. Many have multiple parts. Total suggested time: about 4 hours.

Problem 1: Basic Closed Loop and Error Signal

For the plant $G(s) = 1/(s + 1)$ and proportional controller $G_c(s) = K$:

1. Implement the closed-loop system in MATLAB using `tf` and `feedback` for $K = 2, 5, 10$.
2. For each K , compute and plot the step response and error signal $e(t) = r(t) - y(t)$.
3. Use `stepinfo` to tabulate T_r , T_p , T_s , and %OS as K increases.
4. Comment on how feedback gain affects transient performance and error.

Problem 2: Sensitivity to Gain Variations

Using the same system as Problem 1:

1. Write a MATLAB script that sweeps K from 0 to 50 and computes the closed-loop DC gain for each K .
2. Numerically estimate the normalized sensitivity S_K^T at $K = 10$ using a small perturbation (e.g., $K \pm 5\%$).
3. Plot S_K^T versus K and interpret your results (where is the sensitivity low/high?).

Problem 3: Disturbance Rejection at Plant Input and Output

Consider $G(s) = 1/(s + 1)$, $G_c(s) = K$.

1. Derive expressions for the transfer functions from an input disturbance T_d (at plant input) and an output disturbance T_{do} (at plant output) to $Y(s)$.
2. Implement both disturbance paths in MATLAB using transfer function algebra.

3. For $K = 2$ and $K = 10$, simulate step disturbances of amplitude 0.2 at both locations and compare outputs.
4. Explain the difference in disturbance rejection performance for the two disturbance locations.

Problem 4: Measurement Noise and Frequency Shaping

Use unity feedback with $G_c(s) = K$, $G(s) = 1/(s(s+1))$ (a type-1 plant).

1. Construct the loop gain $L(s)$ and complementary sensitivity $C(s)$.
2. Plot the Bode magnitude of $C(s)$ for $K = 1, 10, 50$ on the same figure.
3. Assume measurement noise is concentrated at high frequency. Discuss which value of K is most robust to noise and why.

Problem 5: Second-Order Approximation and Time-Domain Specs

You are given a closed-loop transfer function:

$$T(s) = \frac{16}{s^2 + 4s + 16}.$$

1. Implement $T(s)$ in MATLAB and plot its step response.
2. Use `damp` and `stepinfo` to extract ζ , ω_n , T_p , %OS, and T_s .
3. Compare the measured values of T_p , %OS, and T_s with the standard second-order formulas.

2 Mini Project (3–4 Hours): DC Motor Positioning System

Design and analyze a DC motor position control system in MATLAB, integrating concepts from Chapters 4–6.

System Description

Consider the simplified motor model:

$$G(s) = \frac{K_m}{(Js + b)s},$$

where K_m is the torque constant, J the inertia, and b the viscous friction. Use nominal values, e.g., $K_m = 1$, $J = 0.01$, $b = 0.1$ (you may choose reasonable alternatives).

Requirements

- Position control with unity feedback.
- Closed-loop step response for a unit step in position:
 - Settling time $T_s \leq 1.5$ s (2% criterion).

- Overshoot $\%OS \leq 10\%$.
- Steady-state position error $e_{ss} = 0$ for step inputs (type ≥ 1).
- Acceptable disturbance rejection to a step load torque disturbance at the plant input.
- Reasonable robustness to measurement noise at high frequency.

Tasks

1. Modeling in MATLAB

- Define $G(s)$ using `tf`.
- Choose a controller structure, e.g. PI or PID: $G_c(s) = K_p + K_i/s + K_d s$ (you may set $K_d = 0$ initially).

2. Initial Design

- Start with a PI controller. Select initial (K_p, K_i) values by trial-and-error or simple pole placement ideas.
- Use `feedback` to form the closed-loop transfer function.

3. Time-Domain Performance

- Use `step` and `stepinfo` to measure T_s and $\%OS$.
- Adjust (K_p, K_i) to meet the time-domain requirements.

4. Steady-State Error and Type

- Compute the open-loop $L(s) = G_c(s)G(s)$ and determine its type by checking the number of integrators.
- Verify $e_{ss} = 0$ for step inputs numerically via simulation.

5. Disturbance Rejection

- Derive and implement in MATLAB the transfer function from a step torque disturbance at plant input to position output.
- Compare the disturbance response for your chosen controller with a lower-gain controller. Discuss trade-offs.

6. Stability and Relative Stability

- Compute the poles of the closed-loop system and verify all are in the LHP.
- Estimate a minimum decay rate σ_0 from the real parts of poles and compare with the observed settling time.

7. Noise Sensitivity

- Construct the complementary sensitivity $C(s)$ and plot its Bode magnitude.
- Discuss whether your design excessively amplifies high-frequency noise. If so, adjust controller parameters or consider adding a low-pass filter.

8. Short Report

- Prepare a report summarizing:
 - Your model and controller design.

- Key MATLAB plots (step, disturbance response, Bode of $C(s)$).
- How the design meets the specifications.
- Trade-offs between performance, disturbance rejection, noise sensitivity, and stability margins.

Closing Remarks

By completing this tutorial, you will have:

- Translated the theoretical notions of error, sensitivity, performance indices, and stability into MATLAB workflows.
- Practiced using key Control System Toolbox functions: `tf`, `ss`, `feedback`, `step`, `impulse`, `lsim`, `pzmap`, `damp`, `isstable`, and eigenvalue analysis.
- Gained experience with a small implementation of the Routh–Hurwitz test.
- Integrated these tools in a mini project that mimics the design reasoning of the main course.

Hand-in Requirements for Problems and Mini Project

Each student must submit a written report and the corresponding MATLAB files. The submission should be clear, reproducible, and well organized. This section specifies in detail what must be included.

1. General Format

- The main report must be submitted as a single **PDF** document.
- MATLAB code must be submitted as zip files, which separate .m files are included.
- Use a clear and consistent structure with numbered sections and subsections.
- Figures should be readable (axes labeled, legends included, units indicated).
- All plots should be generated by MATLAB and referenced in the text.
- The files should be sent to homeworkscut2026@126.com before Dec. 25 2025.
- Create one subsection per problem (e.g., “Problem 1: First-order Thermal System”).
- For each problem, include:
 1. *Problem statement (short)*: Restate or summarize the task in your own words.
 2. *Modeling steps*: Equations, assumptions, and derivations (e.g., ODE, transfer function, state-space).
 3. *MATLAB code snippets*:
 - Show the key code used to solve the problem (not necessarily every line, but enough to understand the approach).
 - Code should be commented to explain important commands and parameters.

4. *Simulation results:*

- Plots of key responses (step response, impulse response, time-domain outputs, pole-zero maps, etc. as required).
- Each figure must have a figure number, title, axis labels, and legend if needed.

5. *Interpretation and discussion:*

- Briefly explain what the results mean: stability, damping, steady-state error, effect of parameter changes, etc.
- Connect the results to concepts from the lecture (e.g., type of system, time-domain specifications, controllability).

Appendix: MATLAB Code

- Include complete MATLAB scripts/functions in an appendix or refer to attached files.
- Each script should have a header comment explaining its purpose (e.g., “Script for Problem 3: Nonlinear Spring Linearization”).
- Ensure the code is well commented and can be run without modification (set random seeds or parameter values inside the script).

3. MATLAB File Submission

In addition to the PDF report, submit:

- One folder named `Name_ID_ControlPractice` (replace with your name and ID).
- Inside the folder, include:
 - `Problem1.m`, `Problem2.m`, ..., `Problem5.m` (or clearly named scripts/functions per problem).
 - A main script for the mini project (e.g., `MiniProject_DCMotor.m`) and any helper functions.
 - Optional: a `README.txt` briefly explaining how to run the scripts.
- Ensure all paths are relative (no hard-coded local paths) so that the instructor can run the files directly.

4. Grading Criteria (Indicative)

The assignment will be evaluated according to the following aspects:

- **Correctness of modeling and analysis:**
 - Equations and MATLAB models correctly reflect the physical systems.
 - Controller designs are consistent with specifications.
- **Use of MATLAB and Control System Toolbox:**
 - Appropriate use of functions such as `tf`, `ss`, `step`, `lsim`, `bode`, `ctrb`, `obsv`, `pid`, `pidtune`, `place`, `lqr`, etc.

- Code is clean, commented, and reproducible.
- **Interpretation and discussion:**
 - Results are clearly interpreted in terms of control concepts (stability, transient response, steady-state error, robustness).
 - Parameter choices and adjustments are justified logically (not just trial-and-error).
- **Clarity and presentation:**
 - Report is well organized, with clear figures and explanations.
 - All requirements of this section are satisfied.