Linear Control Systems
(EE-379)
DE-44 Mechatronics
Syndicate– C

Lab Report 1

Name of members:

NC M. Ahmed Hussain Babar (**427781)**

PC Abdul Haseeb Zahid (**432219)**

PC Muhammad Rizwan Aslam (432069)

NC Muhammad Saqib (**411829)**

Submitted to: LE M.Shahzad Alam

Abstract

This lab is focused on an introduction to MATLAB. The exercise reviewed fundamental MATLAB concepts, including scalar operations, vectors, functions, loops, and conditional statements. It also introduced principles of linear time-invariant (LTI) control systems, distinctions between open-loop and closed-loop configurations, and standard test inputs such as step, ramp, parabolic, and sinusoidal signals. Practical engineering applications were examined, alongside MATLAB commands for polynomial manipulation and signal visualization. The exercise reinforced ethical guidelines and collaborative practices, establishing a basis for subsequent control system analyses.

Introduction

The objective of this laboratory exercise was to provide familiarity with MATLAB as a computational tool for scientific analysis in linear control systems. MATLAB, an acronym for MATrix LABoratory, facilitates matrix-oriented operations, simulations, and data visualization in engineering contexts. This exercise built upon prior concepts from modeling and simulation, while introducing control system applications, including RLC circuit modeling, mass-spring-damper systems, and scenarios such as water level regulation and line-following robotics.

Emphasis was placed on LTI systems, characterized by linearity (proportional response to input scaling) and time-invariance (consistent behavior across time). Distinctions were drawn between open-loop systems (lacking feedback) and closed-loop systems (incorporating feedback for error minimization). Standard test inputs were analyzed to mimic engineering conditions, preparing for system response evaluations.

Objectives

- Comprehension of LTI control systems and their engineering applications.
- Differentiation between open-loop and closed-loop control architectures.
- Association of standard test inputs (step, ramp, parabolic, sinusoidal) with practical engineering contexts.
- Application of MATLAB fundamentals, encompassing scalar and vector operations, functions, loops, conditional statements, and polynomial commands.
- Adherence to laboratory ethics, including discipline, collaboration in groups of three, and prohibition of plagiarism.

Methodology

1. **MATLAB Review:**
    - Basic calculations with scalars and variables were examined (e.g., a = 3; suppresses display, whereas 3*4 assigns the result to ans).
    - Mathematical operations were covered: arithmetic (+, -, /, *), trigonometric (sin, cos, tan), exponential (exp, log, log10, ^), and additional functions.
    - Vector initialization was demonstrated (e.g., r = 1:5;, r = 1:2:5;, s = linspace(0,1);, t = linspace(0,1,10);).
    - Element-wise dot arithmetic was illustrated (e.g., a = [1 2 3]; b = [4 5 6]; a.*b produces [4 10 18]).
2. **Functions in MATLAB:**
    - Functions consist of statement groups defined in separate files, operating within isolated workspaces.
    - They accommodate multiple inputs and outputs.
    - Creation involved selecting "New" > "Function," composing the body, saving, and invoking in primary scripts.
    - An example factorial function was presented: function f = fact(n) f = prod(1:n); end, invoked as y = fact(5); yielding 120.
3. **Loops and Conditional Statements:**
    - **For Loop:** Structured as for i = 1:N commands end, iterating over vector elements.
    - **While Loop:** Formatted as while (condition) commands end, continuing while the condition holds. An example computed x^2 for integers x where $x^3 < 2000$.
    - Nested loops enabled iterative complexity.
    - **If-Else and Switch Statements:** Facilitated runtime code selection.
4. **Control Systems Overview:**
    - LTI systems were explored via examples like RLC circuits (time-domain: $RC\, dv_c/dt + v_c = v_{in}$; frequency-domain transfer functions) and mass-spring-damper systems.
    - Open-loop (no feedback) versus closed-loop (feedback with error $e = r - y$) systems were contrasted.
    - Applications included water level control, thermostat-regulated air conditioning,

and PID-based line-follow robots.

5. **MATLAB Polynomial Commands and Test Inputs:**
   - Polynomial functions: Creation (p = [1 3 2];), evaluation (polyval(p, x)), root finding (roots(p)), root-to-polynomial conversion (poly(r)), differentiation (polyder(p)), multiplication (conv(p, q)), division ([q,r] = deconv(p, d)).
   - Test inputs:
     - Unit Step: $u(t) = 1$ for $t \geq 0$; Plotted via t = 0:0.01:10; u = ones(size(t)); plot(t,u); (e.g., voltage application to motors).
     - Unit Ramp: $r(t) = t$ for $t \geq 0$; Plotted via t = 0:0.01:10; r = t; plot(t,r); (e.g., velocity ramp in conveyors).
     - Parabolic: p(t) = 0.5 t^2; Plotted via t = 0:0.01:10; p = 0.5*t.^2; plot(t,p); (e.g., accelerated motion in elevators).
     - Sinusoidal: $s(t) = \sin(2\pi f t)$; Plotted via t = 0:0.01:10; f=1; s=sin(2*pi*f*t); plot(t,s); (e.g., periodic forces in machinery).

Key Concepts

- **MATLAB Utility in Control Systems:** MATLAB supports matrix and vector computations, ideal for LTI system simulations and transfer function analysis.
- **LTI Properties:** Exemplified in RC circuits, where linearity ensures proportional outputs and time-invariance guarantees temporal consistency.
- **Closed-Loop Feedback:** Error computation ($e = r - y$) enables corrective actions, as observed in thermostats and robotic systems.
- **Engineering Applications:** Encompass DC motor regulation, thermal control in HVAC, drone stabilization, industrial process management, and servo positioning in manufacturing.
- **Programming Practices:** Semicolons suppress outputs, dot operators enable element-wise operations, and modular functions promote code reusability.
- **Test Input Significance:** Step signals represent abrupt changes, ramps indicate constant velocities, parabolas denote accelerations, and sinusoids model oscillations.

Results

MATLAB commands yielded anticipated outcomes:

- Vector operations generated correct arrays (e.g., linspace(0,1,10) produced 10 equidistant values).
- Function executions, such as factorial, delivered accurate results.
- Loops performed iterations as specified, e.g., the while loop enumerated squares up to x=12.
- Test input plots confirmed theoretical definitions.

The exercise emphasized conceptual understanding over quantitative data collection.

Discussion

The exercise integrated MATLAB proficiency with control theory principles. Notable aspects included mastering dot arithmetic to prevent matrix errors and optimizing nested loops. Illustrations, such as robotic line-following, demonstrated feedback's role in performance enhancement. Potential extensions involve simulating input responses using transfer function tools like Limitations arose from the review-oriented content, which reinforced foundational knowledge. The exercise laid groundwork for advanced control system modeling and design.

Conclusion

This introductory laboratory exercise established proficiency in MATLAB for control system applications. Through review of programming elements and exploration of LTI concepts, open/closed-loop systems, and test inputs, essential analytical skills were developed. Ethical adherence supported effective collaboration. The acquired knowledge supports progression to complex system simulations and controller implementations.