Fall 2025 – DE 44 MTS

EE - 379
Linear Control Systems
(LAB - 1)
Introduction to MATLAB

Department of Mechatronics Engineering, National University of Sciences and Technology, College of E&ME

Lab Ethics

- **Just be disciplined and respectful**…

Discipline includes the following:

1. Time management
2. Respecting peers, staff, faculty, and **everyone**
3. Turning off computers at the end of Lab.

- Attendance will be marked at the start of every lab i.e, in first 5 to 10 minutes.
- Eatables are not allowed in lab.
- All the Lab should be done in a group of 3.
- Plagiarism is strictly prohibited.

MATLAB RECAP
ALREADY STUDIED IN MODELING & SIMULATION

Introduction to MATLAB

- MATrix LABoratory
- Purpose is to perform scientific calculations and Analysis

Simple Calculations with MATLAB

- **Scalar Quantities and Variables:**

Consider the following two commands:

```
>> a = 3
>> a = 3;
```

- MATLAB uses the variable **ans** to store the result of our calculation

```
>> 3*4
ans =
12
```

Mathematical Operations

- Arithmetic functions +, -, / and *.
- Trigonometric functions

  sin (sine), cos (cosine) and tan (tangent)
- Exponential functions exp, log, log10 and ˆ.
- Other functions

Vectors in MATLAB

Initializing Vector Objects:

- r = 1:5;
- r = 1:2:5;
- s = linspace(0,1);
- t = linspace(0,1,10);

Dot Arithmetic

To see how dot arithmetic works let's consider a simple example:

```
>> a = [1 2 3];
>> 2*a;
```

- Suppose now we try to multiply a vector by a vector, as in

```
>> a = [1 2 3];
>> b = [4 5 6];
>> a*b
```

- The right way is:

```
>> a.*b
```

- **[a_1b_1, a_2b_2, a_3b_3].**

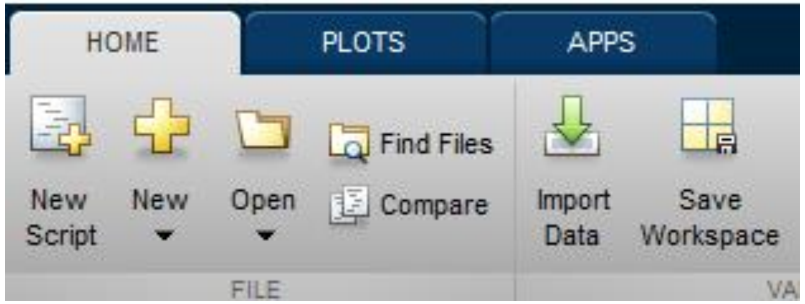
Functions

- A function is a group of statements that together perform a task.
- In MATLAB, functions are defined in separate files.
- Functions operate on variables within their own workspace
- Functions can accept more than one input arguments and may return more than one output arguments.

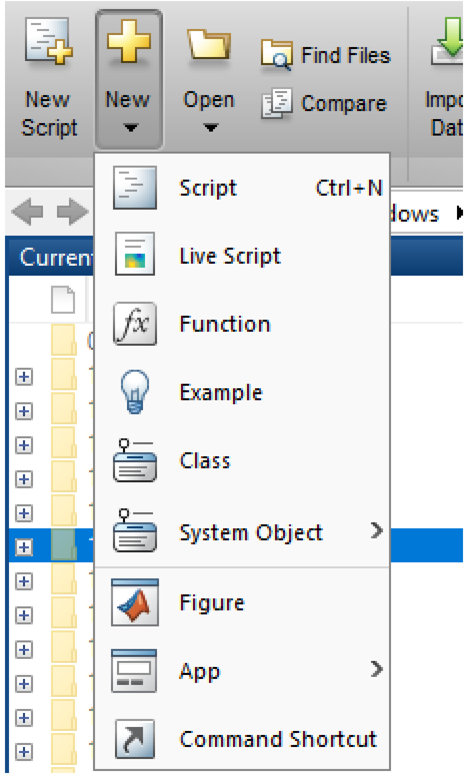
Functions(Continued)…

- How to create a function?

1. Click on “New”.

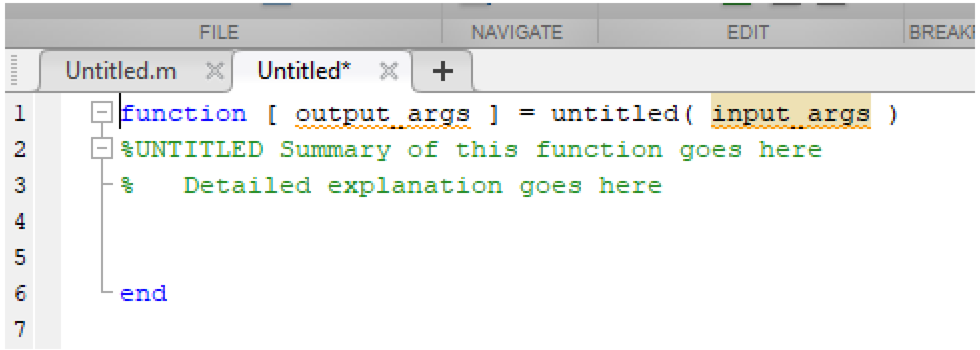


Functions(Continued)…



2. From the drop down menu select “Function”.

3.A new tab like this would open up.



Functions(Continued)…

4. After entering the commands in the function body , save the function.

5. Then you can call the function anywhere in your main program.

Functions(Continued)…

```
function f = fact(n)
f = prod(1:n);
end

x = 5;
y = fact(5)
```

LOOPS

- With loop statements, you can repeatedly execute a block of code.
- There are two types of loops:

1. For statement
2. While statement

FOR LOOP

Syntax of a for statement is

```
for i = 1:N
    commands
end
```

This repeats the commands for each of the values in the vector with i= 1, 2, $\cdots$,N.

For Loop (Example)

```
s=1:10;
for i=s
    a(i)=i^2;
end
disp([1:10; a])
```

```
>> Untitled
     1     2     3     4     5     6     7     8     9    10
     1     4     9    16    25    36    49    64    81   100
```

While Loop

- while statements loop as long as a condition remains true
- Syntax of a while statement is

```
while (condition)
    commands...
end
```

While Loop(Example)

- *Write out the values of x^2 for all positive integer values x such that x^3 < 2000.*

Code:

```
x = 1;
while x^3 < 2000
disp(x^2)
x = x+1;
end
```

Output:

```
>> Untitled
     1

     4

     9

    16

    25

    36

    49

    64

    81

   100

   121

   144
```

Nested Loops

- Loops within a loop are called Nested Loops

```
for ii = 1:3
    for jj = 1:3
        a(ii,jj) = ii+jj;
    end
end
```

Conditional Statements

Conditional statements enable you to select at run time which block of code to execute.

There are two types of conditional statements:

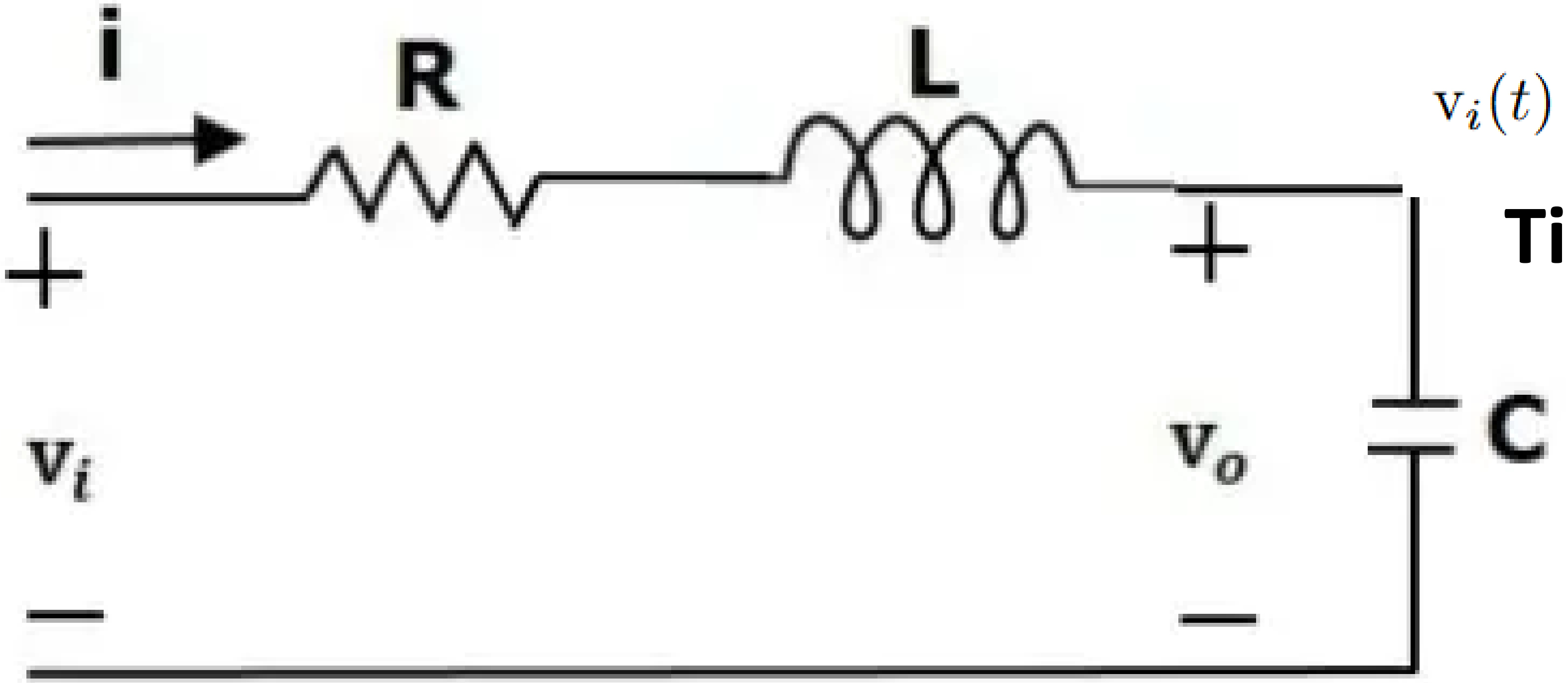
1. If-else statements
2. Switch statements

Learning Objectives

- Understand what linear time-invariant (LTI) control systems are and where they are used.
- Open-loop/close-loop control systems
- Relate standard test inputs (step, ramp, parabolic, sinusoidal) to real engineering scenarios.
- Practice MATLAB basics for signals and polynomials (poly creation, roots, polyval, polyder).

What is a Linear Control System?

Control System Equations for a Series RLC Circuit



$$v_i(t) = LC\frac{d^2v_o(t)}{dt^2} + RC\frac{dv_o(t)}{dt} + v_o(t)$$

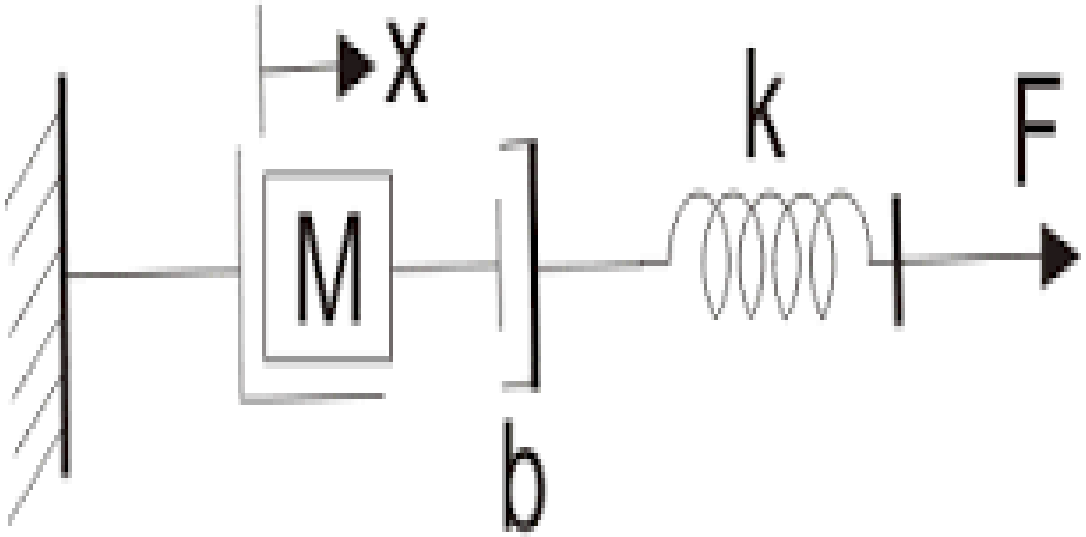
Time-Domain

$$\frac{V_o(s)}{V_i(s)} = \frac{1}{LCs^2 + RCs + 1}$$

Frequency Domain

What is a Linear Control System?

Equations for a Mass-Spring-Damper System



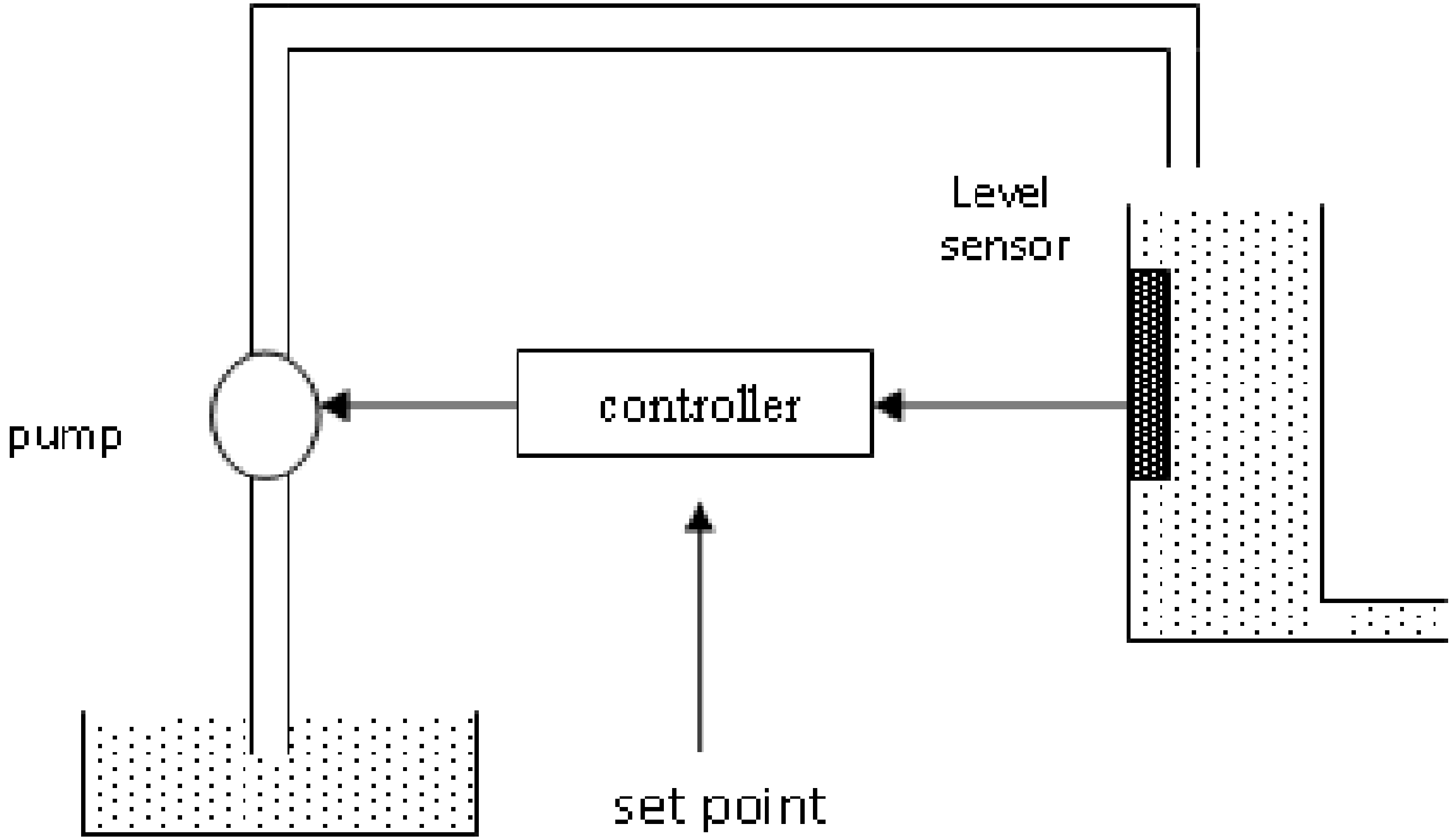
Time-Domain Equation

$$M\ddot{x}(t) + b\dot{x}(t) + kx(t) = F(t)$$

Frequency-Domain Equation (Transfer Function)

$$\frac{X(s)}{F(s)} = \frac{1}{Ms^2 + bs + k}$$

Water Level Controlling System



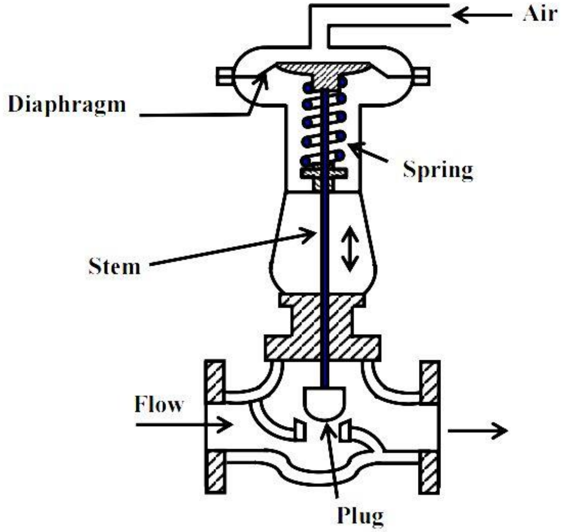


Fig. 1 Control valve

What is a Linear Control System?

$$RC\frac{dv_c}{dt} + v_c(t) = v_{in}(t)$$

This is linear (no squares, no sinusoids multiplied, etc.) and time-invariant (R and C don’t change with time).

Why perfect? ☺

If you double the input voltage, capacitor voltage doubles. (linearity)

If you apply the same input tomorrow, it behaves the same. (time invariance)

Open Loop System

No feedback
output not measured
Example: basic toaster timer.

Closed Loop System

Measure output, compare with reference (error = r − y), adjust input.

Toaster with feedback Human

Air Conditioner (Thermostat):

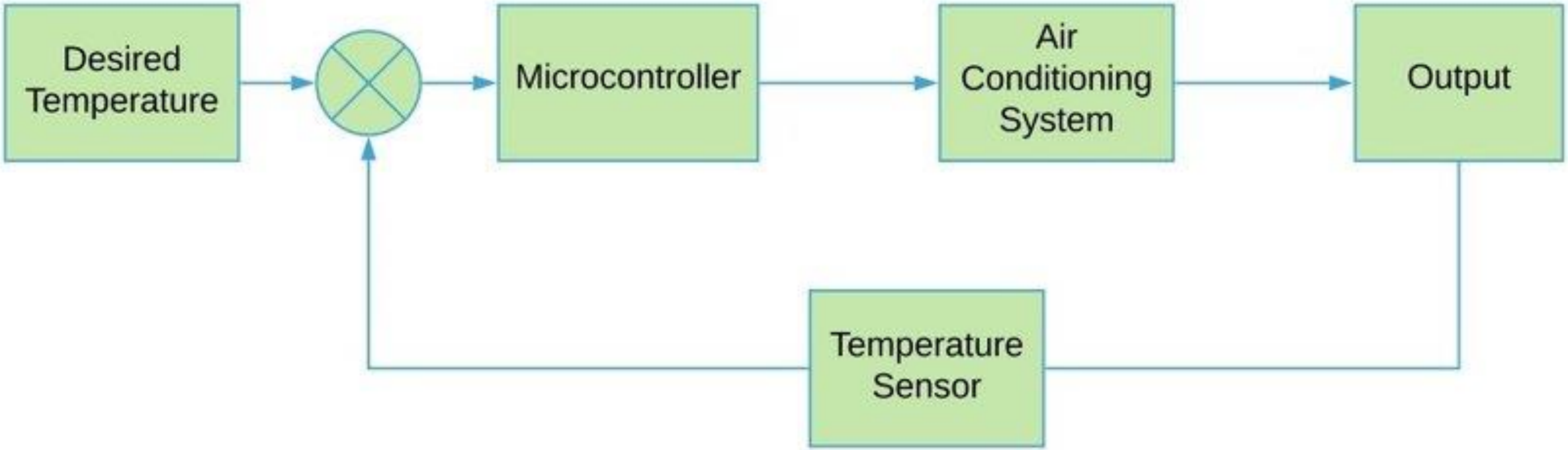
Compares room temperature with setpoint, turns compressor ON/OFF accordingly.

Always compute $e=r-y$

$e = r - y$

- *Sign of error tells direction:*
- *$e<0 \rightarrow$ output is above reference → apply cooling.*
- *$e>0 \rightarrow$ output is below reference → apply heating (if available).*
- *$e=0 \rightarrow$ perfect tracking → do nothing.*

Cont…


Desired Temperature
Microcontroller
Air Conditioning System
Output
Temperature Sensor

Line Follow Robot

- **Input (reference):** Stay on line center.
- **Feedback (sensor):** IR array / camera reads actual line position.
- **Error:** Difference between desired and actual position.
- **Controller (PID):** Calculates correction signal.
- **Actuation:** Motor driver adjusts left/right wheel speeds.
- **Output:** Robot moves smoothly along the line.

Practical Applications of Linear Control

- DC motor speed control in robotics actuators.
- Temperature regulation in ovens/HVAC.
- Drone/quadcopter altitude stabilization.
- Industrial process control: level/flow/pressure loops.
- Servo position control in CNC/3D printers.

MATLAB Polynomial Commands

- Create: p = [1 3 2]; % p(s) = s^2 + 3s + 2
- Evaluate: y = polyval(p, x);
- Roots: r = roots(p);
- From roots: p = poly(r);
- Derivative: dp = polyder(p);
- Product: pq = conv(p, q);
- Division: [q,r] = deconv(p, d);

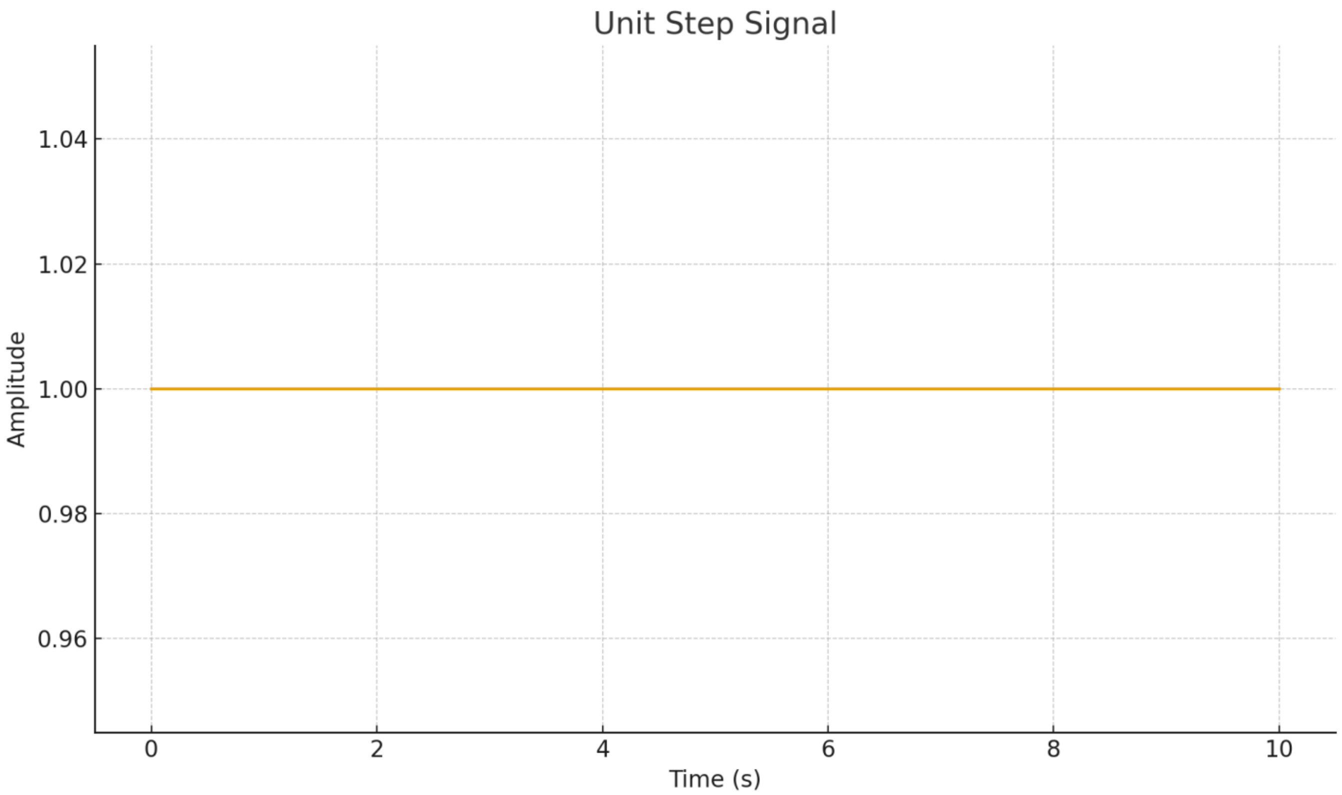
Unit Step Input u(t)

Definition:

$u(t) = 1$ for $t \geq 0$; 0 otherwise

MATLAB snippet:

```
t = 0:0.01:10;
u = ones(size(t));
plot(t,u);
```



Practical examples:

- Switching a light or heater ON
- Applying a voltage step to a DC motor
- Sudden command change in a servo

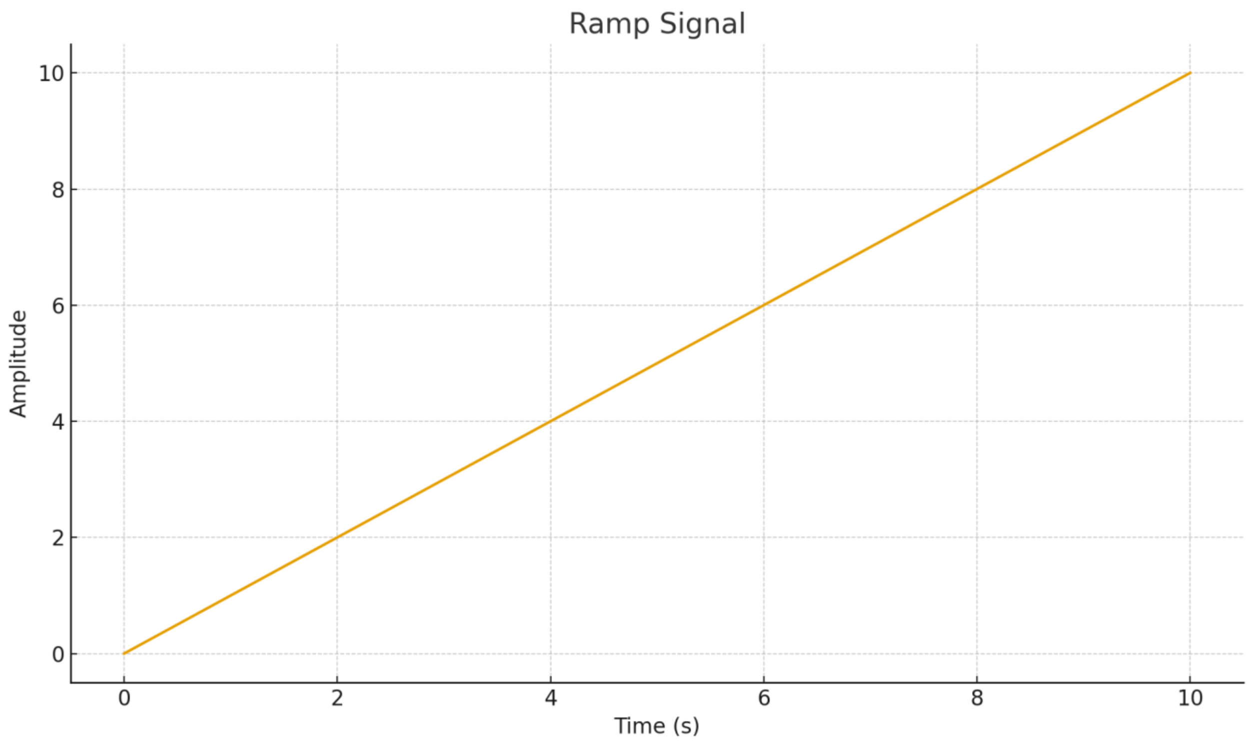
Unit RAMP Input r(t)

Definition:

$r(t) = t \ (t \geq 0)$

MATLAB snippet:

```
t = 0:0.01:10;
r = t;
plot(t,r);
```



Practical examples:

- Linearly increasing throttle on a car
- Conveyor belt speed ramp-up
- Reference position moving at constant velocity

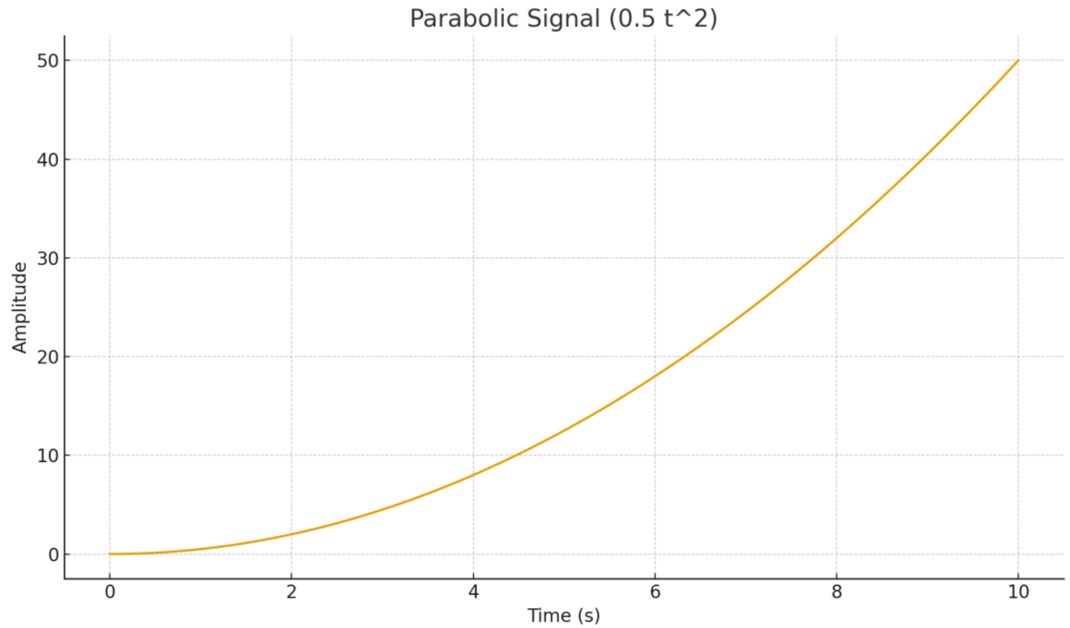
Parabolic Input P(t)

Definition (control context):

p(t) = 0.5 t^2 (constant acceleration)

MATLAB snippet:

```
t = 0:0.01:10;
p = 0.5*t.^2;
plot(t,p);
```



Practical examples:

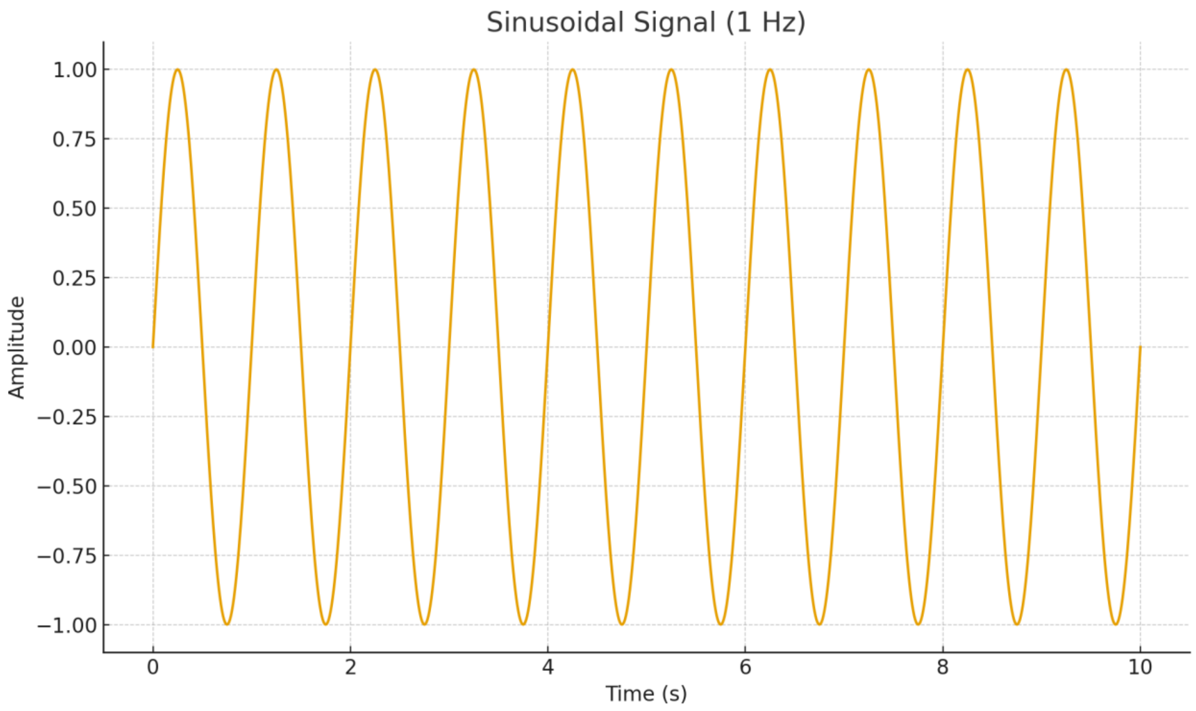
- Elevator starting with constant acceleration

Sinusoidal Input s(t)

$s(t) = \sin(2\pi f t)$

MATLAB snippet:

```
t = 0:0.01:10; f = 1;
s = sin(2*pi*f*t);
plot(t,s);
```



Practical examples:

- Vibration testing with a shaker
- AC voltage/current signals
- Rotating machinery imbalance (periodic forcing)