POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

PARAÑAQUE CITY CAMPUS

INSTRUCTIONAL MATERIAL

FOR

CMPE 30133

FEEDBACK AND CONTROL SYSTEMS

COMPILED BY

ENGR. MARVIN DE PEDRO

Course Description and Learning Outcomes

This course includes the control devices, equations of a system, and block diagram of systems. It also introduces essential topics related to control system technologies in building management.

After completing the course, the student must be able to:

- Identify important concepts and terminologies in Control Theory, and describe principles of control systems design and analysis;
- Use block diagrams and signal flow graphs when designing control systems;
- Sketch the system response of a control system both in time and frequency domains;
- Devise models of physical systems in forms suitable for use in the analysis and design of control systems;
- Illustrate transfer function representations as a means of analysing control systems; and
- Develop robotic systems, automation control systems, and other Industry 4.0 systems using control system analysis and design

Course Overview and Outline

Class Rules

Due to the restrictions brought by COVID-19 threats, there will still be no face-to-face class sessions for the 1st Semester of A.Y. 2021-2022, except probably for some hands-on activities. With that, we will use a "virtual classroom" method of distance learning instead. It's a combination of online technology and home schooling type of class sessions wherein:

1. Class lectures will be done asynchronously through recorded videos. Students can get a copy of all lecture videos and other class materials (downloadable from a given Google Drive folder), and from time to time, check the updates on the class' official FB group.
2. Oral assessments and class huddles will be done synchronously through a video conference (Zoom or Google Meet). There will be 2 synchronous class sessions for each chapter (4 modules) where students can also ask questions regarding the lessons, class requirements, and activities.
3. Students can have self-paced learning through the provided class materials. They can feel free to share it with other people at home and may take the lessons as a productive hobby during this season.
4. Those who can't attend the scheduled synchronous class session as well as those who can't participate or comply with certain class requirements must submit a formal letter explaining his/her valid reason. This letter should be submitted as early as possible so that necessary adjustments or arrangements can be done ahead.
5. Some class requirements will be done as team tasks wherein each member is expected to cooperate and contribute to the team's output. **Please report anyone who does not*.

Knowing that some students may not have access to a reliable internet connection, your <u>cooperation</u> and <u>resourcefulness</u> are highly encouraged. Some of you may ask someone to download the materials on your behalf. Some may even ask to have the downloaded files saved in an SD card and send it to them via *Lalamove* or any courier services available. Same methods can be used for submitting the outputs to the leader who will upload the team's final output. These can be done without compromising safety.

Also, each one as an engineering student is encouraged and expected to:

- Practice self-discipline and time management in the individual and team tasks.
- Be responsible enough to cooperate with groupmates without the instructor's supervision and/or coercion.
- Take time to research and read beyond the given lecture materials.
- Be more serious in developing his/her own engineering skills and virtues.

Grading System

Class Standing		50%
• Quizzes / Laboratory Tasks	30%	
• Class Activities	20%	
Exam		30%
Performance Task		20%

		100%

Transmutation Table	
1.00	97 to 100
1.25	94 to 96.99
1.50	91 to 93.99
1.75	88 to 90.99
2.00	85 to 87.99
2.25	82 to 84.99
2.50	79 to 81.99
2.75	76 to 78.99
3.00	75 to 75.99
5.00	Below 75

Grades will be uploaded on the SIS right after the end of the first semester. A period of 3-5 days will be given for any necessary changes before the grades will be finalized.

Course Requirements

Class Activities (20%) + **Exam** (30%) – Team Task

- Assign members of the team to work on the group task for each module, which basically requires watching a series of YouTube tutorial videos intended to supplement the lessons and to give an introduction to necessary operations using the MATLAB software.
- Take notes of the important things learned from the videos and compile them to present a brief summary of it in the form of a simple AVP with a 5-10 minute runtime. Apply these to the design and analysis of the assigned team project.

- All outputs to this requirement must be submitted within the last week of the semester, but it's best to start working on them early to avoid bottlenecking of the tasks at the said deadline period. Feel free to comment your queries on the dedicated FB group post or ask them during one of the synchronous class sessions.

Quizzes (30%) – Team Task

- All three teams will work together to perform a system study for a proposed *disaster risk reduction and management system* for Brgy. Sto Niño. Compile the information gathered in this study into a single 5-10 minute AVP and PDF documentation.
- Preferably, each team will assign representatives for each of the following tasks:
    - ✓ Discuss about the general information of the site, including its geography, demography, culture, economy, commercial activities, and governance.
    - ✓ Discuss the common disasters encountered within the site and the existing mitigation for such disasters. Gather information from residents and officials, and relate these with the discussed general information about the site.
    - ✓ Discuss the existing communications services and infrastructures installed within the site, and the possible technologies that could be part of risk mitigation for the said disasters. Apply concepts discussed in this course.
- All outputs to this requirement must be submitted within the last week of November. Feel free to comment your queries on the dedicated FB group post or ask them during one of the synchronous class sessions.

Performance Task (20%) – Team Task

- This grade component will come from the system proposal with feasibility study for the assigned project. It basically requires proper documentation and video presentation.
- See the given content outline document for the specific instructions.
- All "behind-the-scene" activities related to compliance to this requirement must also be documented by taking photos of group members performing the tasks, and by taking screenshots of the accomplished work in any necessary software tools as well as of the conversations, charts, and other project management details of the group.
- All outputs to this requirement must be submitted within the last week of the semester, but it's best to start working on them early to avoid bottlenecking of the tasks at the said

deadline period. Feel free to comment your queries on the dedicated FB group post or ask them during one of the synchronous class sessions.

Rubrics:

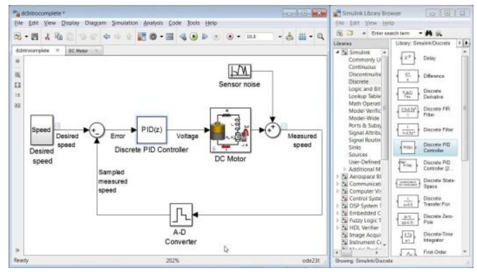
Criteria	Description	Weight Score
Substance	Output includes the expected key points and correct components.	50
Form	Output is presented accurately and systematically with minimum grammatical and formatting errors.	25
Compliance	Output meets the prescribed length, coverage, and other specified instructions.	25
	TOTAL	**100**

Principles of Control Engineering

(CMPE 30133 – Feedback and Control Systems)

1 Chapter

I. OVERVIEW



Control engineers design and oversee the production of many types of complex systems and equipment. They may focus on developing or improving machines or systems that have singular or multiple functions. The mechanism that an engineer may control could be a system, process, or machine. For example, the system may be used in a bank to improve an ATM and decrease the amount of time that a person spends at the machine. The engineer researches, tests, and analyzes the design use in relation to the function of the elements in the system.

II. MODULE OBJECTIVES

After successful completion of modules 1-4 of Chapter 1, you should be able to:

1) Remember important concepts and different types of signals and systems;
2) Understand the working principles and applications of control theory;
3) Evaluate the techniques in designing control systems with block diagrams and signal flow graphs; and
4) Apply the concepts of control theory to design a control system using MATLAB simulation.

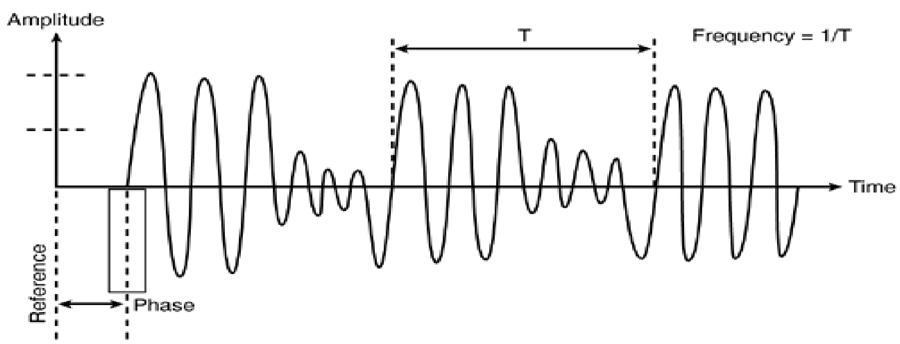
III. COURSE MATERIALS

Suggested Online Resources for Further Learning

- Downloadable Course AVPs: https://bit.ly/3iLs1dM
- Downloadable Course PDFs: https://bit.ly/3mxuSs5

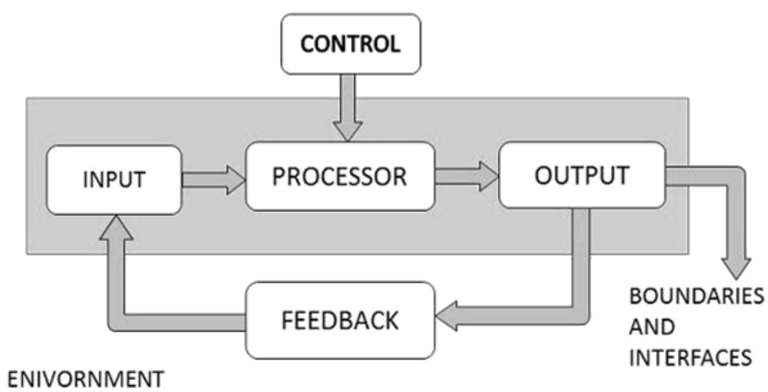
Module 1: **Signals and Systems**

Electronic Signal is a time varying voltage, current or EM wave that potentially provides information on the status of a physical system, or conveys a message between observers, among other possibilities.



System may be referred to any set of components, which function in interrelated manner for a common cause or objective. It has the following characteristics:

- ✓ Organization
- ✓ Interaction
- ✓ Interdependence
- ✓ Integration
- ✓ Central Objective



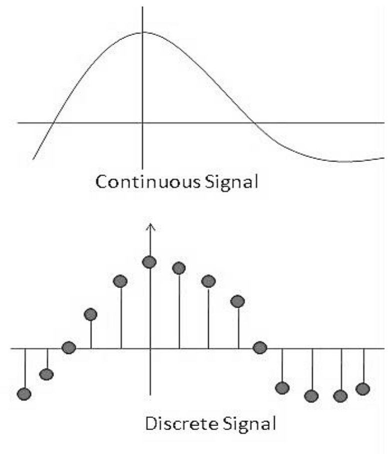
Components of a Basic Feedback System

Continuous-Time Signals

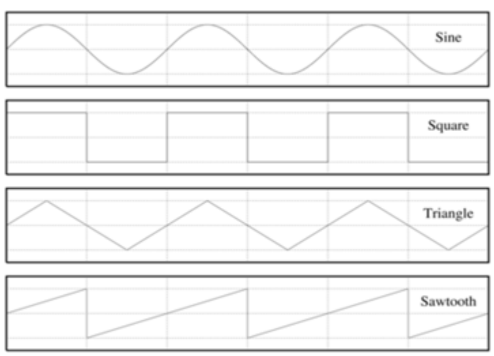
- defined along a continuum of time and are thus, represented by a continuous independent variable
- often referred to as analog signals, show continuity both in amplitude and time (these will have values at each instant of time)
- sine and cosine functions are the best example

Discrete Signals

- defined at discrete times where every independent variable has distinct value, and thus represented as sequence of numbers

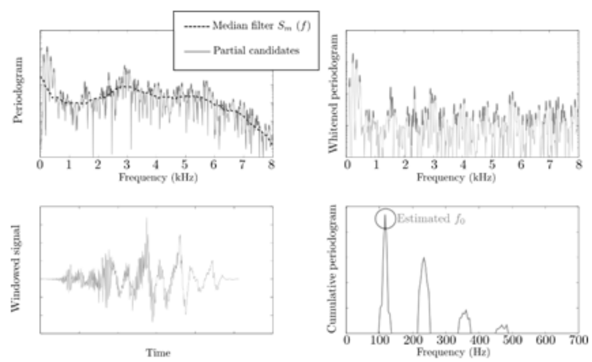


Periodic and Aperiodic Signals



$$x(n+N) = x(n) \qquad N = \frac{K}{f_0}$$

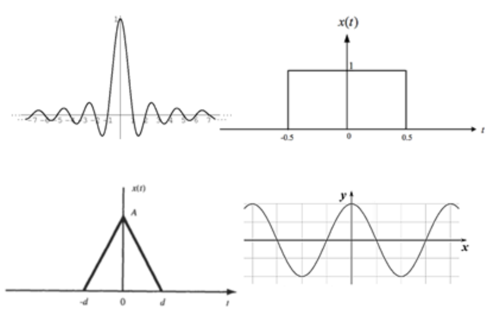
PERIODIC SIGNALS



(does not satisfy the condition for periodic signal)

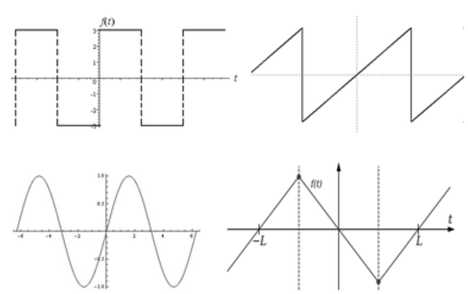
APERIODIC SIGNALS

Odd and Even Signals



$$x(-n) = x(n)$$

EVEN SIGNALS



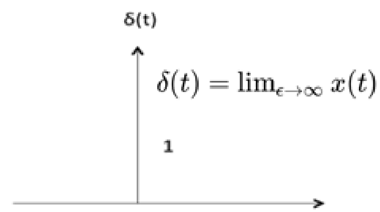
$$x(-n) = -x(n)$$

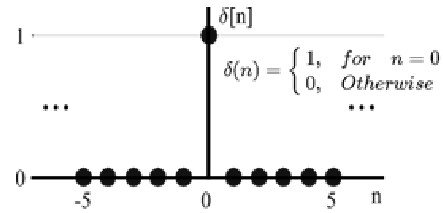
ODD SIGNALS

Common Basic Electronic Signals (Test Signals)

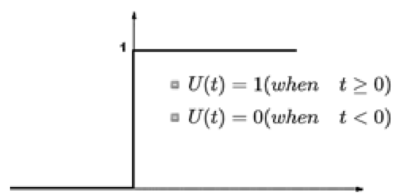
	Continuous-Time	Discrete-Time

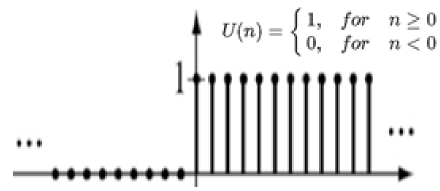
Unit Impulse Signal (Dirac Delta Function)



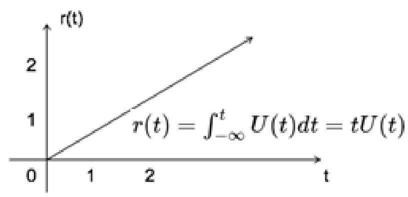


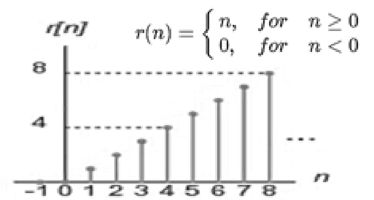
Unit Step Signal



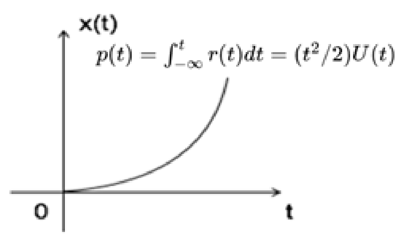


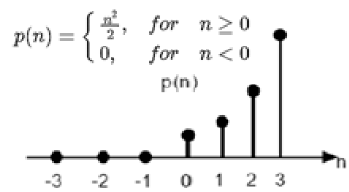
Ramp Signal





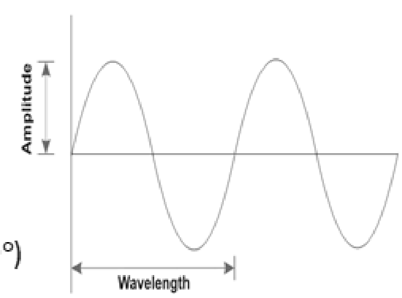
Parabolic Signal

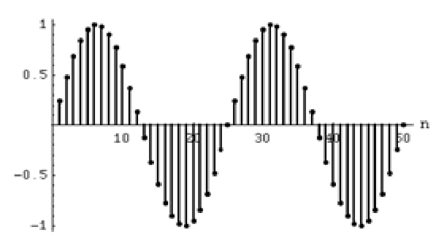




Sinusoidal Signal

- periodic signal
- either
  - sine (odd at $\emptyset = 0°$)
  - cosine (even at $\emptyset = 0°$)





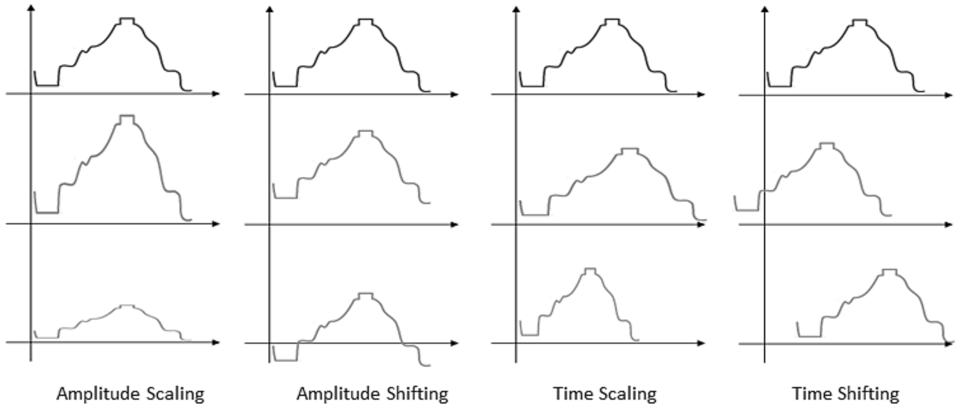
NOTE: sine and cosine functions are 90^{0} apart

$$x(t) = A\sin(\omega t + \phi)$$

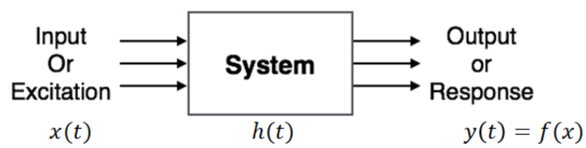
$$x(n) = A\sin(\omega n + \phi)$$

$$N = \frac{2\pi m}{\omega}$$

Basic Operations of Signal Processing



In control engineering, a *system* is a basic, complete and functional machine, including all the hardware and software required to make it functional for a user. It should have the ability to receive user input, process data, and with the processed data, create a response for output and/or storage. The input to a system is called as ***excitation*** and output from it is called as ***response***. For one or more inputs, the system can have one or more outputs.



Properties of a System

1. Memory
   - <u>Static Systems</u> – do not have feedback, and thus do not have any memory; output depends only upon the present values of the input where past and future values of the data are not present.

     Examples: $y(t) = x(2t)$

     $y(t) = cos[x(t)]$
   - <u>Dynamic Systems</u> – have feedback and can store past and future values, and thus require some memory; output depends upon the past and future value of the signal at any instant of the time; those with signals that have cases of *amplitude shifting*, *time shifting*, *time scaling*, *integration*, and *differentiation*.

     Examples: $y(t) = x(t) + x(t-1)$

     $$y(t) = \frac{[x(t) + x(t-1)]}{2}$$

2. Causality
    - Causal Systems – independent from the future values only; practically or physically realizable because impulse response must be 0 for all t < 0

      Examples: $y(t) = x(t-1)$
      $y(t) = x(t) + 10$
    - Non-Causal Systems – include image processing systems (independent variable is space) and post-processing audio systems

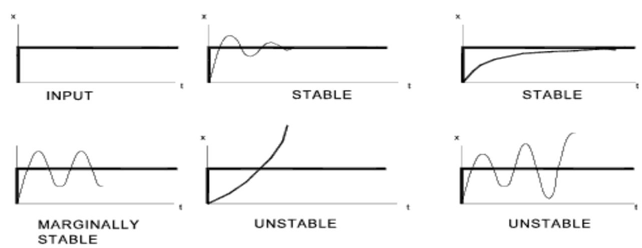
      Example: $y(t) = x(t) + x(t+1)$
    - Anti-Causal Systems – depend upon the future values of the input only

      Examples: $y(t) = x(t+1)$

3. Stability
    - Stable Systems – satisfy the BIBO (bounded input for bounded output) condition; bounded means finite in amplitude examples of bounded inputs are functions of sine, cosine, DC, signum and unit step

      Examples: $y(t) = x(t) + 10$
      $y(t) = \sin[x(t)]$
    - Unstable Systems – do not satisfy the BIBO conditions (ex. exponentials)

      Examples: $y(t) = tx(t)$
      $y(t) = \dfrac{x(t)}{\sin t}$



4. Invertibility
    - Invertible Systems – distinct inputs lead to distinct outputs and an inverse transformation (inverse system) exists denoted by $T^{-1}[\,]$, which maps the outputs of original systems to the inputs applied: $TT^{-1} = T^{-1}T = I$ ($I = 1$ for SISO Systems)

      Examples: $y(t) = x(t+2)$
    - Non-Invertible Systems – distinct inputs leads to same outputs and an inverse system will not exist

      Examples: $y(t) = \sin[2\pi x(t)]$

$$y(t) = \int_{-\infty}^{t} x(\tau)d\tau$$

$$y(t) = \frac{dx(t)}{dt}$$

5. Time-Invariance
   - Time-Invariant Systems – any delay provided in the input must be reflected in the output

     Examples: $y(T) = x(2T)$

     $y(T) \ = \ \sin[x(T)]$
   - Time-Invariant Systems – output and input should be delayed by some time constant but the delay at the input should not reflect at the output; coefficient in the system relationship is a function of time (examples include all time scaling cases)

     Examples: $y(T) = x[\cos T]$

     $y(T) = \cos(T) \cdot x(T)$
6. Linearity
   - Linear Systems – follows the *Law of Superposition*, *Law of additivity* and *Law of Homogeneity*; the output should be zero for zero input and there should not be any non-linear operator present in the system

     Examples: $y(t) = \sin[tx(t)]$

     $y(t) = x(t+1) + x(t-1)$
   - Non-Linear Systems – output is not zero when input applied is zero; can be applied on the either input or on the output to make the system non-linear; common non-linear operators (both for x and y):
     - Trigonometric operators- Sin, Cos, Tan, Cot, Sec, Cosec etc.
     - Exponential, logarithmic, modulus, square, Cube etc.
     - Sa(i/p) , Sinc (i/p) , Sqn (i/p) etc.

     Examples: $y(t) = sin[tx(t)]$

     $y(t) = x(t+1) + x(t-1)$

ASSESSMENT

Group Task

Watch the following YouTube tutorials about MATLAB and perform the tasks they feature:

- ✓ What is MATLAB
- ✓ How to Install MATLAB
- ✓ Getting Started with MATLAB
- ✓ What is Simulink?
- ✓ How to Get Started with Control Systems in MATLAB

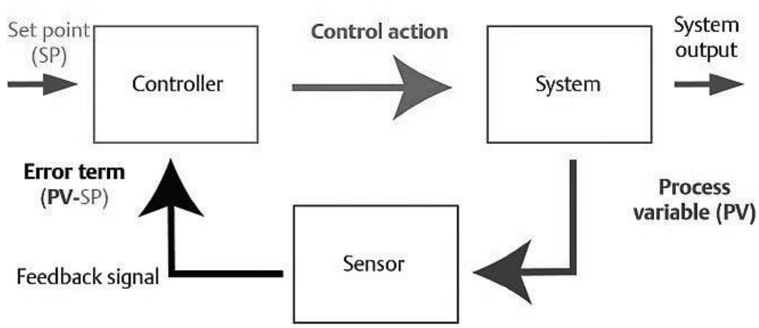
See the discussion proper on the Introduction to MATLAB (Module 4),

Module 2: **Control Theory**

Control System is basically a system that controls – manages, commands, directs or regulates – the behavior of other devices or systems to achieve desired results. If either the output or some part of the output is returned to the input side and utilized as part of the system input, then it is known as ***feedback***, which plays an important role in order to improve the performance of the control systems.

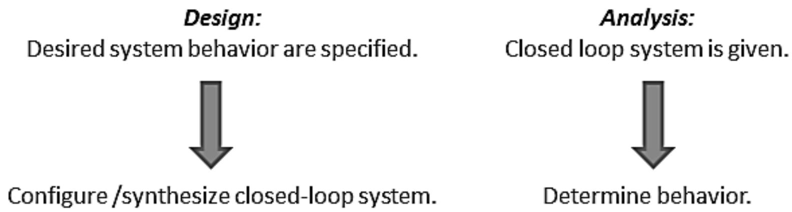
Key Terms in Control Systems

- Controller – monitors the controlled *PV*, and compares it with the *reference* or *SP*
- Plant – physical object, system or process to be controlled
- Controlling Variable (*Control Action* or *Actuating Signal*) – input or an effect (generated outside the plant) that causes the plant to behave in a certain way
- Controlled Variable (*Processed Value* or *PV*) – actual value of the output or the quantity or condition that is of interest and thus is controlled
- Reference/Command Input (*Set Point* or *SP*) – a signal supplied to the control system which represents the desired value (or variation) of the controlled output
- Error Signal (or *PV-SP Error*) – difference between *actual* and *desired value* of the PV applied as *feedback signal* to generate a control action
- Disturbance Input – unwanted input that tends to adversely affect the system output

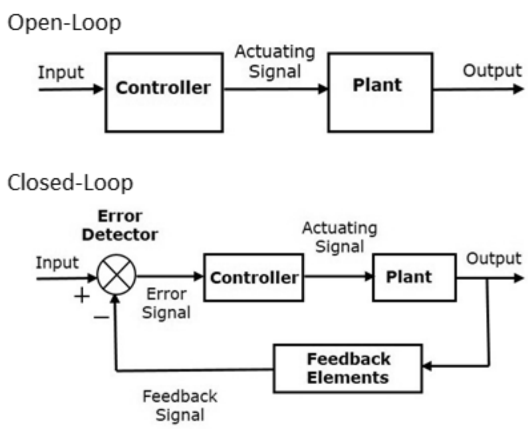


Control Theory is the field of engineering and applied mathematics that deals with the behavior of dynamical systems. The goal of control engineers is to develop a control model for controlling such systems using a control action in an optimum manner without *delay or overshoot* and ensuring control stability.

Control Theory helps us gain insight on how and why feedback control systems work and how to systematically deal with various design and analysis issues.



Mathematical models are most often used to predict future behavior, and control system design methodologies are based on such models. Understanding control theory requires engineers to be well versed in basic mathematical concepts such as *Differential Equations* and using *Laplace Transform*.



Open-Loop	Closed-Loop
No feedback element	Has feedback element
No error detector	Has error detector
Stable	May become unstable
Easy to construct	Difficult to construct
Economical	Costly
Small bandwidth	Large bandwidth
Less maintenance	More maintenance
Unreliable	Reliable
Examples: Hair Dryer, tea maker	Examples: Servo voltage stabilizer, car control engine

Control Systems can be classified based on the type of the signal used.

- <u>Continuous-Time</u> – all the signals are continuous in time.
- <u>Discrete-Time</u> – there exists one or more discrete time signals.

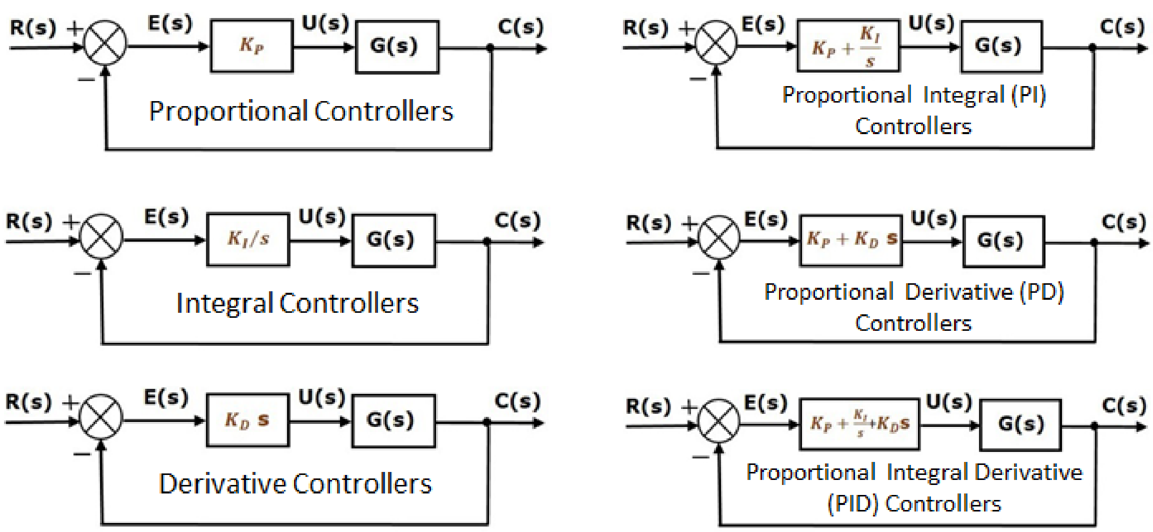
Control Systems can also be classified based on the number of inputs and outputs present.

- <u>SISO</u> – Single Input and Single Output
- <u>SIMO</u> – Single Input and Multiple Outputs
- <u>MISO</u> – Multiple Inputs and Single Output
- <u>MIMO</u> – Multiple Inputs and Multiple Outputs

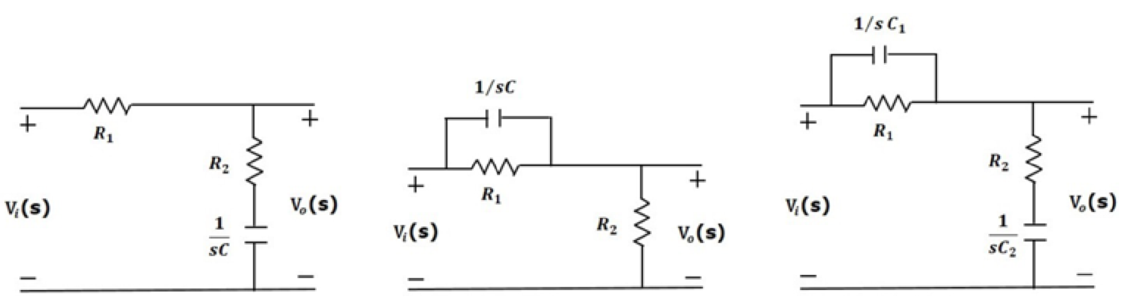
There are two types of control systems according to the objective of control:

- <u>Regulator</u> – system functions as a *compensator* that keeps the controlled output at a constant value at all times in the presence of disturbances (the command input is either *zero or constant*)
- <u>Servomechanism</u> – system functions as a *controller* that keeps the controlled output following a time varying command input (originally with mechanical position, velocity or acceleration as the controlled output)

Types of Controller



Types of Compensator



Lag Compensator

$$\frac{V_o(s)}{V_i(s)} = \frac{1}{\alpha}\left(\frac{s+\frac{1}{\tau}}{s+\frac{1}{\alpha\tau}}\right)$$

$$\tau = R_2 C \qquad \alpha = \frac{R_1+R_2}{R_2}$$

Lead Compensator

$$\frac{V_o(s)}{V_i(s)} = \beta\left(\frac{s\tau+1}{\beta s\tau+1}\right)$$

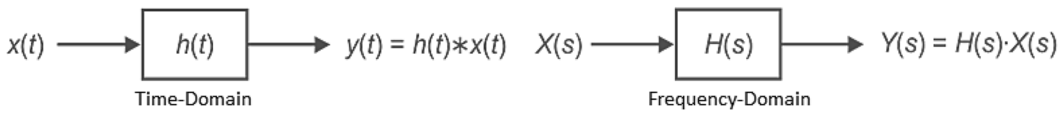
$$\tau = R_1 C \qquad \beta = \frac{R_2}{R_1+R_2}$$

Lag-Lead Compensator

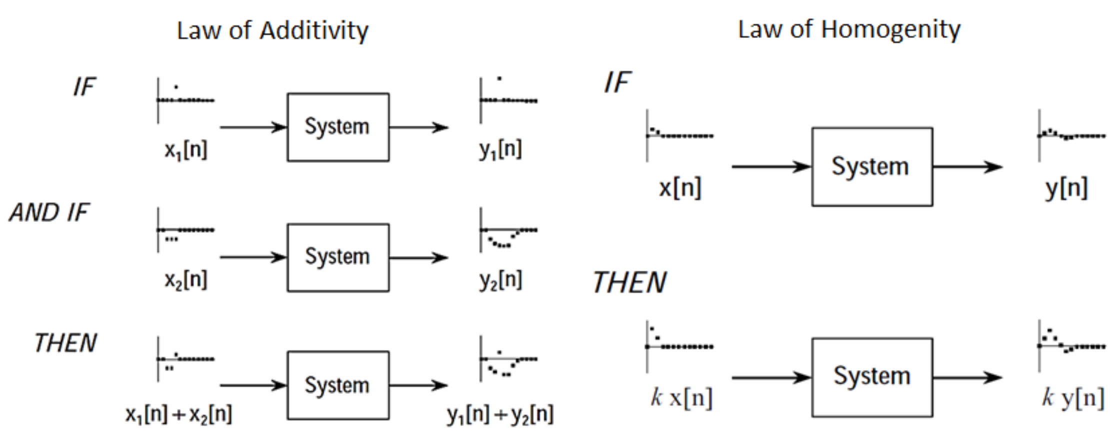
$$\frac{V_o(s)}{V_i(s)} = \left(\frac{s+\frac{1}{\tau_1}}{s+\frac{1}{\beta\tau_1}}\right)\left(\frac{s+\frac{1}{\tau_2}}{s+\frac{1}{\alpha\tau_2}}\right)$$

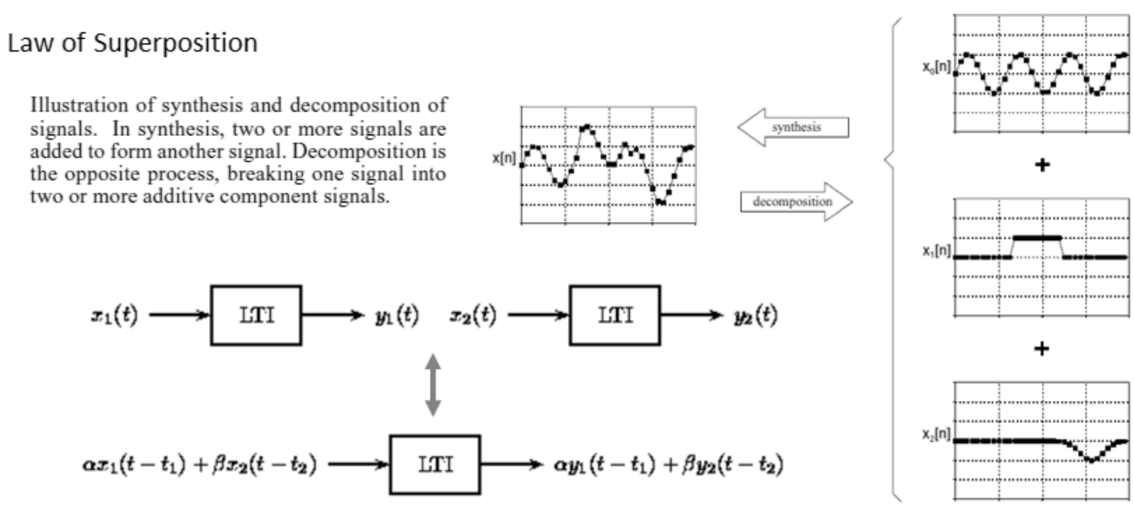
$$\tau_1 = R_1 C_1 \qquad \alpha\beta = 1 \qquad \tau_2 = R_2 C_2$$

Linear Time-Invariant (LTI) *Systems* are those with output(s) for a linear combination of inputs that are the same as a linear combination of individual responses to those inputs; output(s) do not depend on when an input was applied.

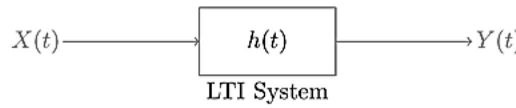


This type of system is superior to simple state machines for representation because they have more memory of past states and have the ability to predict the future. It is often used to model power plants to predict long-term system behavior. It also used in electrical circuits consisting inductors, transistors, and resistors that are the basis upon which modern technology is built.





Transfer Function (also known as *System Function* or *Network Function*) relates the output and the input of a certain system, and gives the device's output for each possible input. Often represented as a graph (*Transfer Curve* or *Characteristic Curve*), transfer functions are used in the mathematical analysis of systems, particularly using the block diagram technique, in electronics and control theory.



$$y(t) = x(t) * h(t) \Rightarrow Y(s) = X(s)H(s)$$

$$H(s) = \frac{Y(s)}{X(s)}$$

A transfer function has the following properties:

1. The transfer function is defined only for a linear time-invariant (LTI) system. It is not defined for nonlinear systems.
2. The transfer function between a pair of input and output variables is the ratio of the Laplace transform of the output to the Laplace transform of the input.

$$TF = H(s) = \frac{Y(s)}{X(s)}$$

3. All initial conditions of the system are set to zero.

$$y(0) = y'(0) = y'(0) \ldots = y^n(0)$$

3. The transfer function is independent of the input of the system.

The term *transfer function* is also used in the frequency domain analysis of systems using methods such as the *Laplace transform*; here it means the amplitude of the output as a function of the frequency of the input signal. For example:

- The transfer function of an electronic filter is the voltage amplitude at the output as a function of the frequency of a constant amplitude sine wave applied to the input.
- For optical imaging devices, the optical transfer function is the Fourier transform of the point spread function (hence a function of spatial frequency).

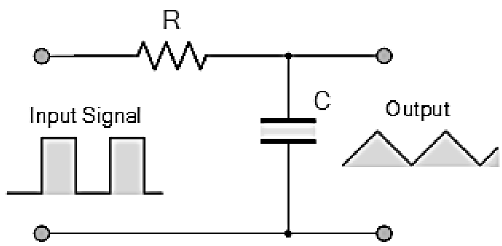
The dimensions and units of the transfer function model the output response of the device for a range of possible inputs. For example:

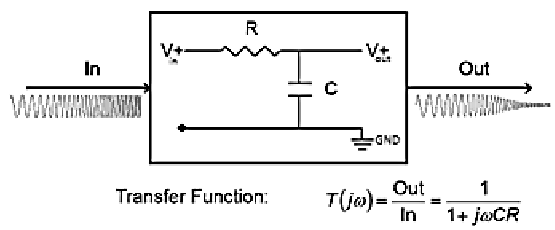
- The transfer function of a two-port electronic circuit like an amplifier might be a two-dimensional graph of the scalar voltage at the output as a function of the scalar voltage applied to the input.

- The transfer function of an electromechanical actuator might be the mechanical displacement of the movable arm as a function of electrical current applied to the device.
- The transfer function of a photodetector might be the output voltage as a function of the luminous intensity of incident light of a given wavelength.

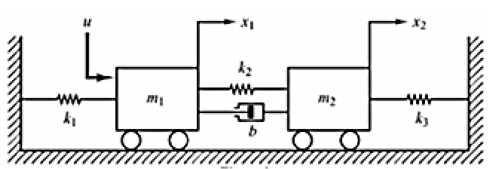
Transfer Function of Electrical Circuits

Component	Voltage-current	Current-voltage	Voltage-charge	Impedance $Z(s) = V(s)/I(s)$	Admittance $Y(s) = I(s)/V(s)$
Capacitor	$v(t) = \frac{1}{C}\int_0^t i(\tau)d\tau$	$i(t) = C\frac{dv(t)}{dt}$	$v(t) = \frac{1}{C}q(t)$	$\frac{1}{Cs}$	Cs
Resistor	$v(t) = Ri(t)$	$i(t) = \frac{1}{R}v(t)$	$v(t) = R\frac{dq(t)}{dt}$	R	$\frac{1}{R} = G$
Inductor	$v(t) = L\frac{di(t)}{dt}$	$i(t) = \frac{1}{L}\int_0^t v(\tau)d\tau$	$v(t) = L\frac{d^2q(t)}{dt^2}$	Ls	$\frac{1}{Ls}$

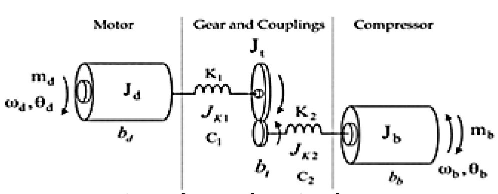




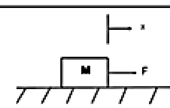
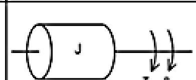
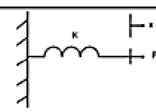
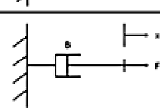
Transfer Function of Mechanical Systems



Translational Mechanical System



Rotational Mechanical System

	Symbol	Force/Torque
Mass (Translational)		$F = M\frac{d^2x}{dt^2}$
Mass (Rotational)		$T = J\frac{d^2\theta}{dt^2}$
Spring		$F = K_x$ (Trans'l.) $T = K_\theta$ (Rot'l.)
Dashpot/ Damper		$F = B\frac{dx}{dt}$ (Trans'l.) $T = B\frac{d_\theta}{dt}$ (Rot'l.)

ASSESSMENT

Q&A

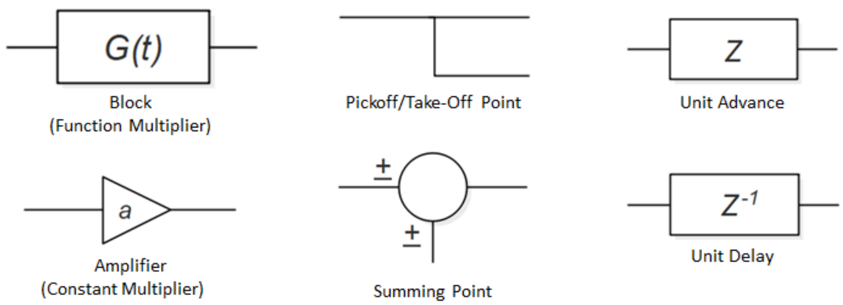
Watch the following YouTube tutorials about MATLAB and perform the tasks they feature:

- ✓ Open-Loop Control Systems
- ✓ Feedback Control Systems
- ✓ Components of a Feedback Control System
- ✓ Simulating Disturbance Rejection in Simulink
- ✓ Simulating Robustness by System Variations

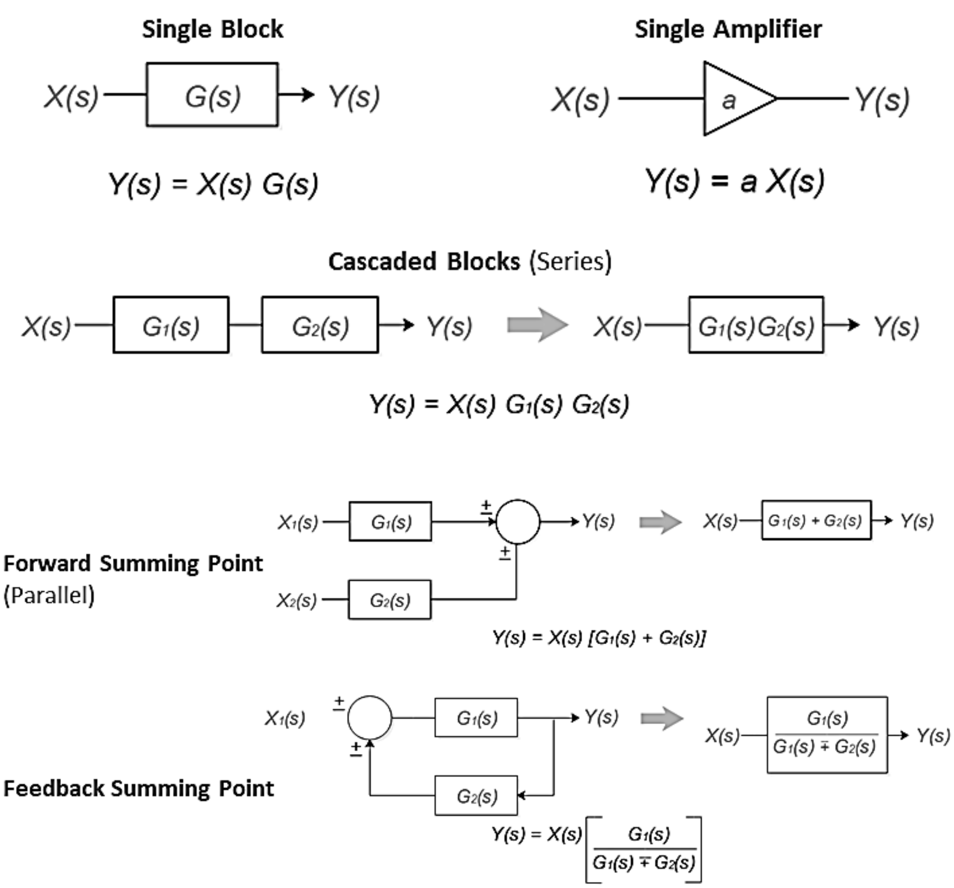
See the discussion proper on the Introduction to MATLAB (Module 4).

Module 3: **Block Diagrams**

Block diagram is a diagram of a system in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks. The following are some of its basic components:



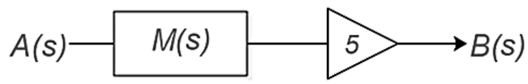
Basic Block Diagrams



Transfer Function to Block Diagram

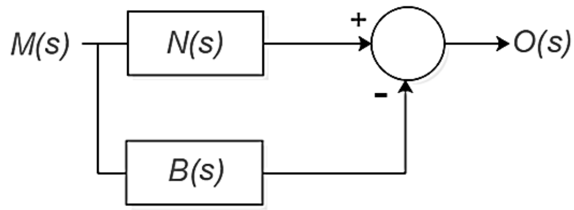
Example #1

$$\frac{B(s)}{A(s)} = 5M(s)$$

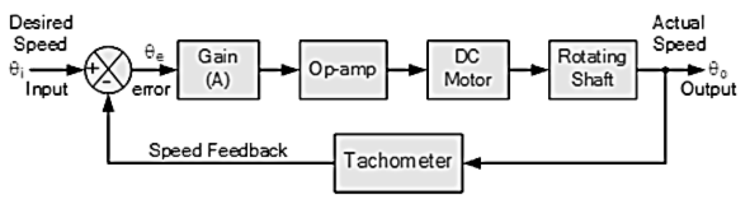


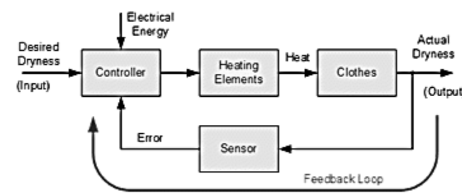
Example #2

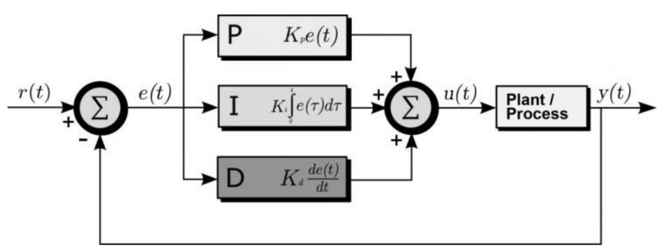
$$\frac{O(s)}{M(s)} = N(s) - B(s)$$



Examples of Real-World Control System Block Diagrams







The *transfer function* of a component is represented by a block, which has a single input and a single output. Its output is obtained by multiplying transfer function of the block with input.

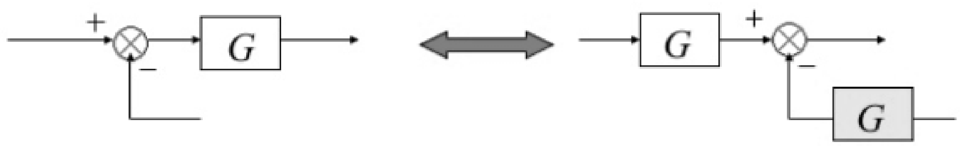
Steps in Finding the Transfer Function of a Block Diagram

$$TF = H(s) = \frac{Y(s)}{X(s)}$$

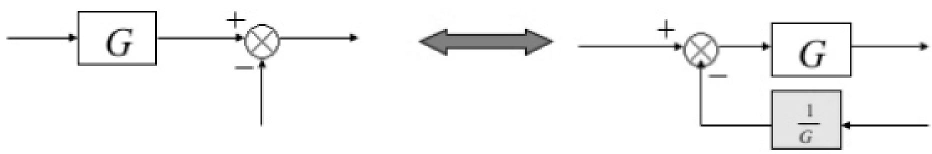
1. Locate the input and output of the diagram.
2. Reduce the blocks connected in series.
3. Reduce the blocks connected in parallel.
4. Reduce the simple feedback loops.
5. Try to shift take off points towards right and summing points towards left.
6. Repeat steps 2 to 5 until the simplest form is obtained.
7. Obtain the Transfer Function of Overall System.

Simplification of Block Diagrams

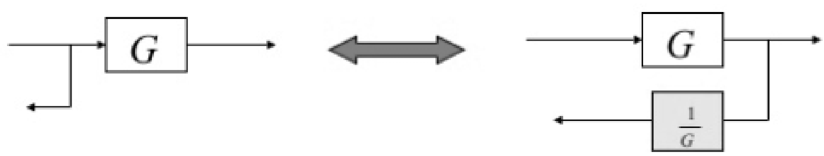
Moving a Summing Point Behind a Block



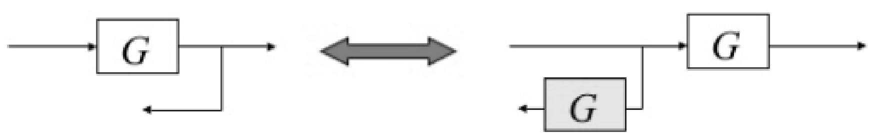
Moving a Summing Point Ahead a Block



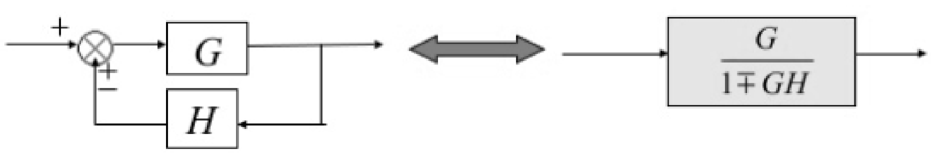
Moving a Pickoff Point Behind a Block



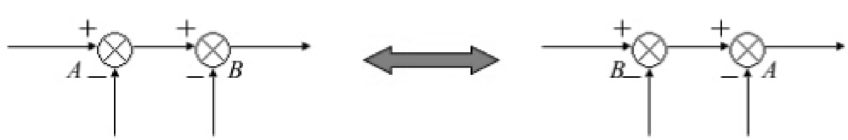
Moving a Pickoff Point Ahead a Block



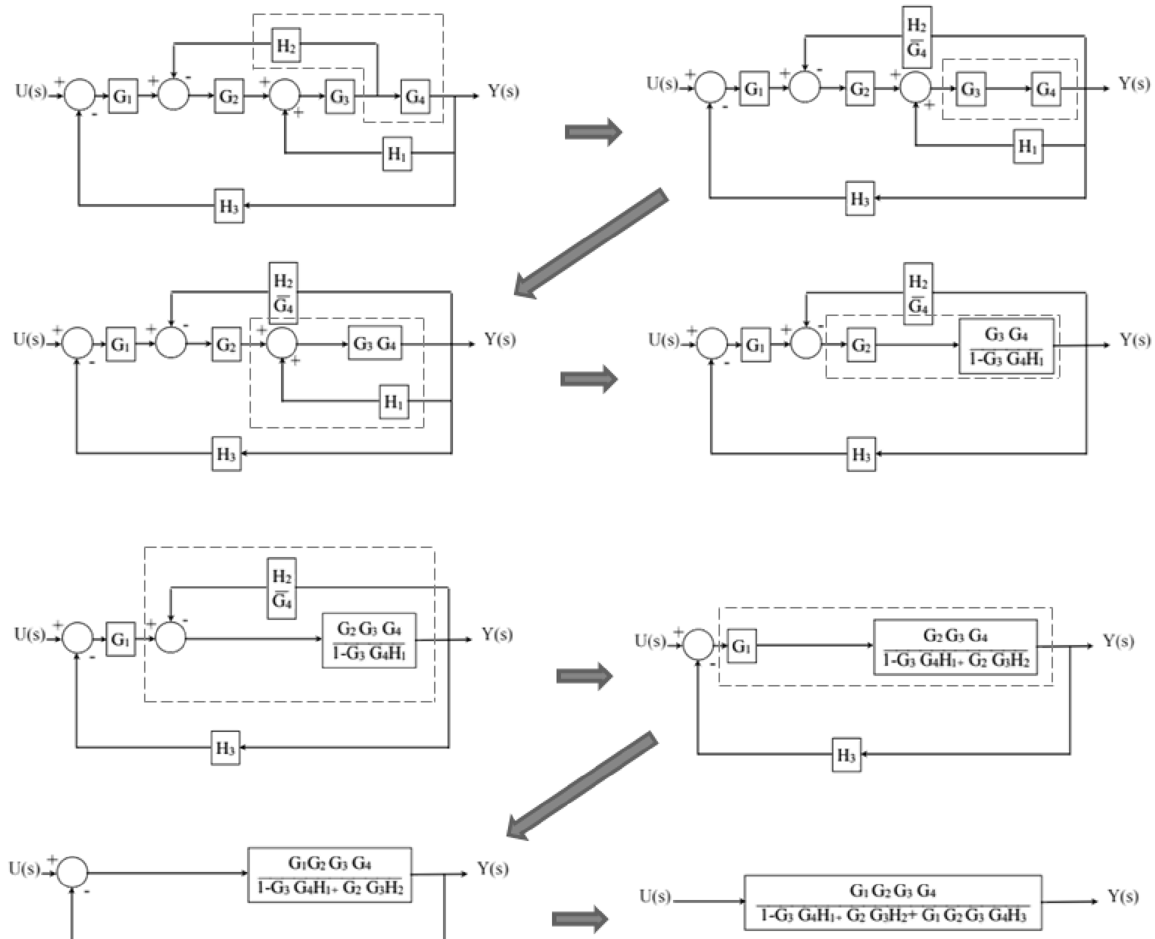
Eliminating a Feedback Loop



Swapping with Two Adjacent Summing Points

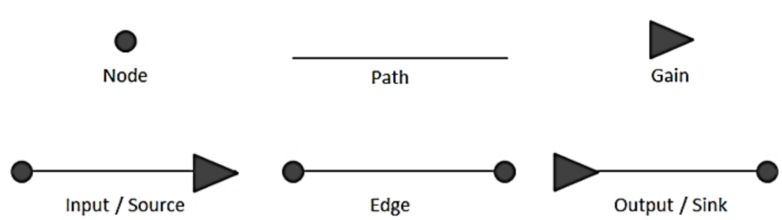


Example:

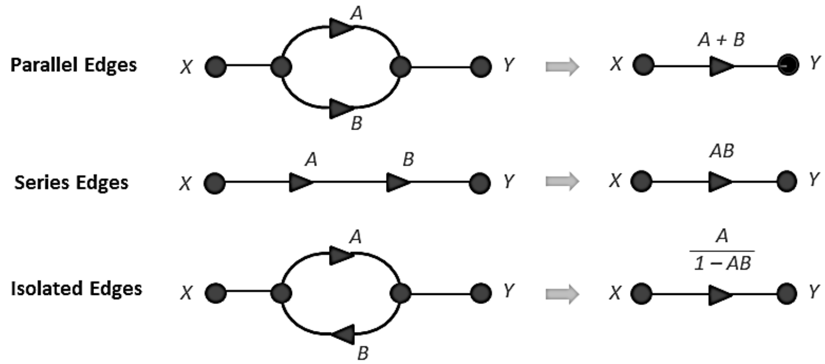


$$TF = \frac{G_1G_2G_3G_4}{1 - G_3G_4H_1 + G_2G_3H_2 + G_1G_2G_3G_4H_3}$$

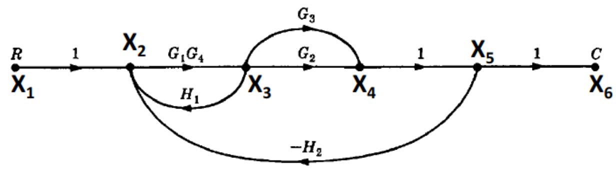
Signal-flow graph (SFG) is a directed graph used to model signal flow in a system. It can be used to derive the system's *transfer function* using *Mason's Rule* (or *Mason's Gain Formula*). The following are its basic components:



Basic Signal Flow Graphs



Steps in Finding the Transfer Function of a Signal Flow Graph



1. Identify the input and the output of the given system.
2. Determine all the possible forward paths (P_1, P_2, P_3, …) and their gains (G_1, G_2, G_3, …).
3. Determine all the possible loops (L_1, L_2, L_3, … or L_{11}, L_{12}, L_{13}, …) and their respective gains. (G_1, G_2, G_3, …). If possible, determine also the twin, triplet, etc. loops that don't touch each other. (ex. $L_{21} = L_{11}L_{12}$ *or* $L_{311} = L_{112}L_{123}$)
4. Solve for the determinant of the system (Δ) and the determinant of the i^{th} forward-path (Δ_i).
5. Substitute all the derived equations to the formula of ***Mason's Rule*** and simply the resulting transfer function.

Mason's Rule

$$TF(s) = \frac{1}{\Delta}\left(\sum_{i=1}^{n} P_i \Delta_i\right)$$

note:

$\Delta_i = \Delta$
for part of SFG that don't touch each other

$\Delta_i = 1$
if no non-touching loops to the i^{th} path, or if taking out i^{th} path breaks all the loops

where:

$n = no. of\ forward\ paths$

$P_i = the\ i^{th}\ forward - path\ gain$

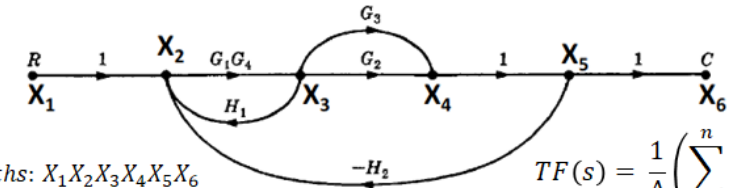
$\Delta_i = determinant\ of the\ i^{th}\ forward - path$

$\Delta = determinant\ of\ the\ system$

$= 1 - sum\ of\ individual\ loop\ gains + \ sum\ of\ products\ of\ gains\ of\ all\ 2\ loops\ that\ don't\ touch\ each\ other$

$- sum\ of\ products\ of\ gains\ of\ all\ 3\ loops\ that\ don't\ touch\ each\ other + \ ...$

Example 1



Input: X_1

Output: X_6

2 *Forward Paths*: $X_1X_2X_3X_4X_5X_6$

2 *Path Gains*: $P_1 = G_1G_2G_4$ *and* $P_2 = G_1G_3G_4$

Loops: $X_2X_3X_2$ *and* **2** *loops along* $X_2X_3X_4X_5\,X_2$

Loop Gains: $L_1 = G_1G_4H_1 \quad L_2 = -G_1G_2G_4H_2$ *and* $L_3 = -G_1G_3G_4H_2$

$\Delta = 1-(L_1+L_2+L_3)$

$\Delta_i = \Delta_1 = \Delta_2 = 1$

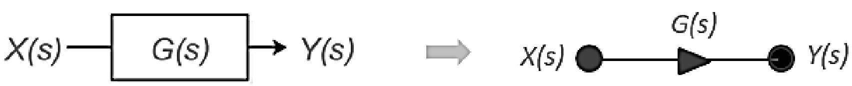
$$TF(s) = \frac{1}{\Delta}\left(\sum_{i=1}^{n} P_i\Delta_i\right)$$

$$= \frac{1}{\Delta}(P_1\Delta_1 + P_2\Delta_2)$$

$$= \frac{(P_1+P_2)}{1-(L_1+L_2+L_3)}$$

$$TF = \frac{G_1G_2G_4 + G_1G_3G_4}{1 - G_1G_4H_1 + G_1G_2G_4H_2 + G_1G_3G_4H_2}$$

Steps in Converting a Block Diagram into a Signal Flow Graph



1. Represent all the signals, variables, summing points and take-off points of block diagram as nodes in signal flow graph.
2. Represent the system blocks of block diagram as branches in signal flow graph.
3. Represent the transfer functions inside the blocks of block diagram as gains of the branches in signal flow graph.
4. Connect the nodes as per the block diagram. If there is connection between two nodes (but there is no block in between), then represent the gain of the branch as one. For example, between summing points, between summing point and takeoff point, between input and summing point, between take-off point and output.

Block Diagrams	Signal Flow Graphs
Visualize input output relations	Alternative to block diagrams
Useful in design and realization of (linear) components; help understand the flow of information between internal variables	Do not require iterative reduction to find transfer functions (using Mason's Gain Rule)
Equivalent to a set of linear algebraic equations (of rational functions)	Can be used to find the transfer function between any two variables (not just the input and output)
Mainly relevant where there is a cascade of information flow	Lock familiar to computer scientists (?)

ASSESSMENT

Q&A

Watch the following YouTube tutorials about MATLAB and perform the tasks they feature:

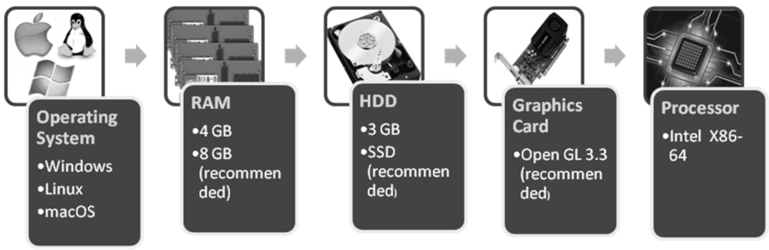
- ✓ Getting Started with Simulink 1
- ✓ Getting Started with Simulink 2
- ✓ Getting Started with Simulink 3
- ✓ Getting Started with Simulink 4
- ✓ Getting Started with Simulink 5

See the discussion proper on the Introduction to MATLAB (Module 4).

Module 4: **Introduction to MATLAB**

MATLAB is a high-level language and interactive environment used by engineers and scientists for numerical computation, visualization, and programming. It is useful tool for analyzing data, developing algorithms, and creating models and applications, and it provides functions that operate on real and complex numbers, as well as on vectors, matrices, and structures. ***Simulink*** is a MATLAB-based graphical programming environment for modeling, simulating and analyzing multi-domain dynamical systems.

MATLAB System Requirements



Before proceeding to work with MATLAB, it is recommended to be familiar with any other high-level or object-oriented programming (OOP) languages, such as C or C++ or Java, particularly about programming language elements, like variables, data type, loops, functions, methods, conditional statements, classes, objects.

Downloading MATLAB

- Step 1: Visit the official website www.mathworks.com.
- Step 2: Go to the bottom (footer) area of the webpage and click on the ***Trial Software*** link under ***Try or Buy*** *section*.
- Step 3: A new web page opens after clicking on the ***Trial Software*** link. Enter your email ID in the space provided
- Step 4: After clicking on ***Continue***, it again asks some more information to create your account with MathWorks, to get access to the products. Apart from the email ID and the location you previously entered, it asks for your purpose to get the MATLAB software- select hobbyist or personal use option, and select ***Yes*** for above 13 years option and click ***Create***.

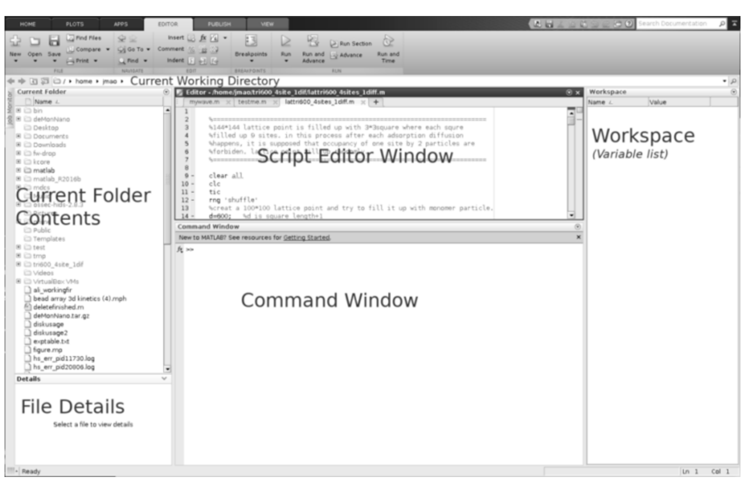
- Step 5: Again, a new webpage opens, click on ***Continue*** with ***Current Email*** option to proceed further. Now go to your email's inbox and open the email received from MathWorks. Now verify your email address by clicking on ***Verify Your Email*** button.
- Step 6: You will be directed to a new webpage, fill up all the details about you, accept the agreement, and click on ***Create*** button.
- Step 7: A new webpage opens, here your verified email id will be displayed. Below it, select the ***I agree*** button and click on ***Submit*** button. Now we have created a MathWorks account.
- Step 8: Here you will find different types of options for add-ons, you must select one of these and click on ***Continue***. On the next page, some options will be displayed, but hese all are optional. So, you can continue by selecting or without selecting these options.
- Step 9: A new page opens displaying your email ID & your license ID, note down these details.

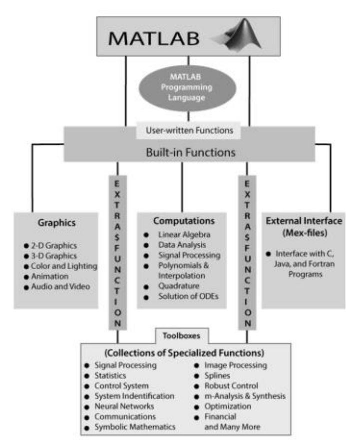
Installing MATLAB

- Step 1: Double click on the MATLAB icon (the binary file which we downloaded earlier). A pop-up will ask for the installer to run, click on the ***Run***. A MathWorks Installer window will pop-up on the screen.
  - By default, the first option, i.e., Log in with a MathWorks Account, is selected, we will proceed with this option. So, click on ***Next*** button.
  - Accept the license terms by selecting ***Yes*** in the next page and again click on the ***Next*** button.
  - A new page appears, by default, the first option is selected, Log in to your MathWorks Account. Enter here your email ID and password that we created during the creation of our account with MathWorks.
- Step 2: A ***License Selection*** window will appear, a preselected license id will be highlighted with a blue background. Here you have to select your license ID; this is the ID which we have saved during STEP 9 of downloading of the installer (we urged to note down that ID during that time) and again click on ***Next***. A new ***Folder Selection*** window appears, no need to change the folder location for installation of MATLAB, click on ***Next***.

- Step 3: Next is ***Product Selection*** window, the first product is MATLAB 9.6, this is mandatory to select because it is the MATLAB environment, and from other products, you can choose as many of your choices and click on ***Next***.
- In the ***Installation Options*** window, select options as per your choice. Any time you feel something to change, you can go back to the previous step by clicking on the ***Back*** button. Next is the ***Confirmation*** window, here you no need to do anything, confirm what you are going to download in the process of the installation of MATLAB, its other Add-on products, and what is the size of the downloads; and click on ***Install***. By clicking on ***Install***, downloading of all the products will be started. It's a massive download, so you have to wait for some time to complete the download.
- Step 4: After downloading of all products and completion of the installation, a window appears that says to ***Activate the MATLAB***, no need to do anything, click on the ***Next*** button.
    - After clicking on ***Next***, a new window appears that says about what is the meaning of activation. Proceed by clicking on ***Next***.
    - Again a new window appears displaying your email ID and your products' license ID, proceed by clicking on ***Confirm*** button.
    - Congratulations, you have completed the installation process and successfully installed the MATLAB and its other products. Now click on the ***Finish*** button.
- **Step 5:** A MATLAB shortcut will be created on the desktop as per our choice during the installation process. Now we can work with MATLAB by clicking on the icon placed there on the desktop.

MATLAB Environment and Built-In Functions



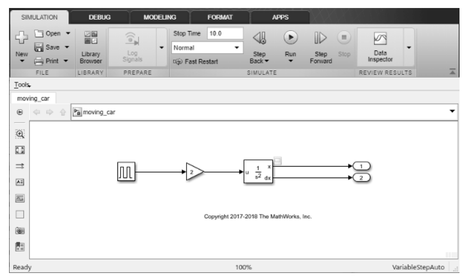


A schematic diagram of MATLAB's main features.

MATLAB File Types

- **M-files** (.m) – standard ASCII text data; these are either *Script Files* or *Function Files*
- **MAT-files** (.mat) – binary information files created by MATLAB when you save a record with save command. The record is written with a special format that only MATLAB can read. Mat-files can be loaded into MATLAB with the load commands.
- **Fig-Files** (.fig) – binary figure files that can be opened again in MATLAB as figures; a Fige file contains all the data required to recreate the figure. Specific files can be opened with the open *filename.fig* command.
- **P-files** (.p) – file that can be executed in MATLAB directly (without being parsed and compiled). These files are created with the *pcode* command.
- **Mex-files** (.mex) – MATLAB-callable FORTRAN, C, and Java programs.

You can use ***Simulink***® to model a system and then simulate the dynamic behavior of that system. This example simulates simplified motion of a car. A car is typically in motion while the gas pedal is pressed. After the pedal is released, the car idles and comes to a stop.



A ***Simulink block*** is a model element that defines a mathematical relationship between its input and output. To create this simple model, you need four Simulink blocks. Simulating this model integrates a brief pulse twice to get a ramp. The results display in a Scope window. The input pulse represents a press of the gas pedal — 1 when the pedal is pressed and 0 when it is not. The output ramp is the increasing distance from the starting point.

Block Name	Block Purpose	Model Purpose
Pulse Generator	Generate an input signal for the model	Represent the accelerator pedal
Gain	Multiply the input signal by a constant value	Calculate how pressing the accelerator affects the car acceleration
Integrator, Second-Order	Integrate the input signal twice	Obtain position from acceleration
Outport	Designate a signal as an output from the model	Designate the position as an output from the model

Open New Model

1. Use the Simulink Editor to build your models. Start MATLAB®. From the MATLAB toolstrip, click the **Simulink** button.
2. Click the ***Blank Model*** template. The Simulink Editor opens.

- From the ***Simulation*** tab, select ***Save > Save As***. In the ***File Name*** text box, enter a name for your model. For example, simple_model. Click ***Save***. The model is saved with the file extension .slx.

Open Simulink Library Browser

Simulink has block libraries, organized in the Library Browser, such as:

- Continuous — for systems with continuous states
- Discrete — for systems with discrete states
- Math Operations — implement algebraic and logical equations
- Sinks — store and show the signals that connect to them
- Sources — generate the signal values that drive the model

1. From the ***Simulation*** tab, click the ***Library Browser*** button .
2. Set the Library Browser to stay on top of the other desktop windows. On the Simulink Library Browser toolbar, select the ***Stay on Top*** button . To browse through the block libraries, select a category and then a functional area in the left pane. To search all of the available block libraries, enter a search term. For example, find the Pulse Generator block. In the search box on the browser toolbar, enter pulse, and then press Enter.

Get detailed information about a block. Right-click the Pulse Generator block, and then select **Help for the Pulse Generator block**. The Help browser opens with the reference page for the block. Blocks typically have several parameters. You can access all block parameters by double-clicking the block.

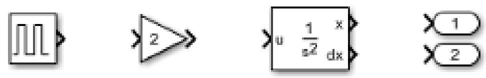
Add Blocks to a Model

1. To start building the model, browse the library and add the blocks. From the Sources library, drag the Pulse Generator block to the Simulink Editor. A copy of the Pulse Generator block appears in your model with a text box for the value of the ***Amplitude*** parameter. Enter 1.
2. Add the following blocks to your model using the same approach.

Block	Library	Parameter
Gain	Simulink/Math Operations	Gain: 2
Integrator, Second-Order	Simulink/Continuous	Initial condition: 0
Outport	Simulink/Sinks	Port number: 1

Add a second Outport block by copying the existing one and pasting it at another point using keyboard shortcuts. Your model now has the blocks you need.

3. Arrange the blocks by clicking and dragging each block. To resize a block, drag a corner.

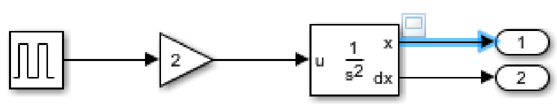


Connect Blocks

1. Click the output port on the right side of the Pulse Generator block. The output port and all input ports suitable for a connection are indicated by a blue chevron symbol .
2. Point to to see the connection cue. Click the cue. Simulink connects the blocks with a line and an arrow indicating the direction of signal flow.
3. Connect the output port of the Gain block to the input port on the Integrator, Second-Order block. Also, connect the two outputs of the Integrator, Second-Order block to the two Outport blocks.
4. Save your model. In the ***Simulation*** tab, click **Save**.

Add Signal Viewer

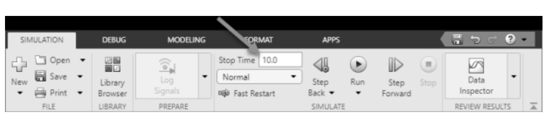
To view simulation results, connect the first output to a Signal Viewer. Click the signal. In the **Simulation** tab under **Prepare**, click **Add Viewer**. Select **Scope**. A viewer icon appears on the signal and a scope window opens.

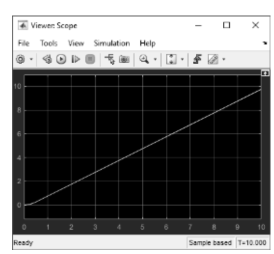


You can open the scope at any time by double-clicking the icon.

Run Simulation

1. In the **Simulation** tab, set the simulation stop time (default is 10.0) by changing the value in the toolbar. This time value has no unit.





2. To run the simulation, click the **Run** button .

ASSESSMENT

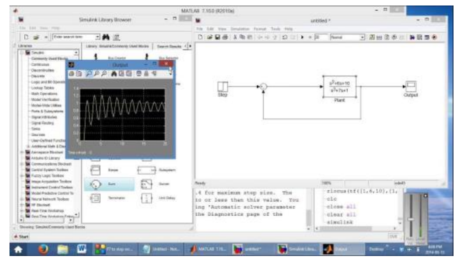
Group Task

Watch the following YouTube tutorials about MATLAB and perform the tasks they feature:

- ✓ Getting Started with Simulink 6
- ✓ Getting Started with Simulink 7
- ✓ Getting Started with Simulink 8
- ✓ Getting Started with Simulink 9
- ✓ MATLAB and C/C++

See more MATLAB activities in the next modules.

I. OVERVIEW



Control Systems are used to continually regulate the behaviour of devices in a repeatable and predicted way. They may be simple electronic or electrical devices or very sophisticated computer-controlled systems. The fundamentals of a control system include measuring an error signal and then adjusting the system to reach the desired course or outcome. If the system deviates too fast and the control system is unable to adjust the course or outcome, the control system is inadequate to manage the disturbances.

II. MODULE OBJECTIVES

After successful completion of modules 5-8 of Chapter 2, you should be able to:

1) Remember the essential engineering mathematics and their respective applications to engineering;
2) Understand the principles of Laplace transformation and other concepts of mathematical modelling of control systems;
3) Evaluate the methods of analyzing the time response and frequency response of a control system; and
4) Apply the key concepts of control theory to design a control system that meets the requirements of the real-world scenario.

III. COURSE MATERIALS

Suggested Online Resources for Further Learning

- Downloadable Course AVPs: https://bit.ly/3iLs1dM
- Downloadable Course PDFs: https://bit.ly/3mxuSs5

Module 5: **Engineering Mathematics**

Differential Equation refers to an equation that contains one or more terms involving derivatives of one variable (the dependent variable, y) with respect to another variable (the independent variable, x)

1. $$\frac{d^3y(t)}{dt^3} + a_2\frac{d^2y(t)}{dt^2} + a_1\frac{dy(t)}{dt} + a_0y(t) = 0$$

This is the one showing the most detailed information of the model. It shows function prototype, dependent variable, independent variable

2. $$\frac{d^3y}{dt^3} + a_2\frac{d^2y}{dt^2} + a_1\frac{dy}{dt} + a_0y = 0$$

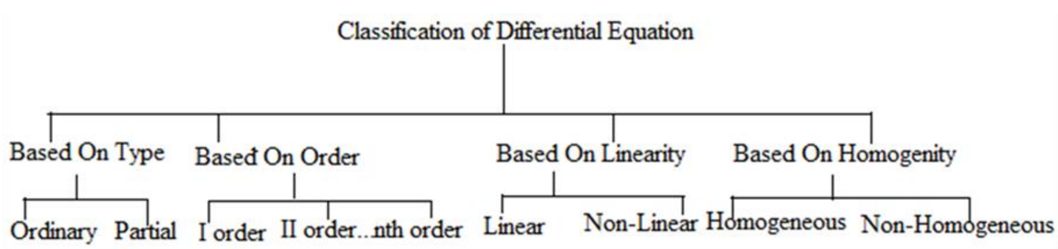
This is a little bit simpler than the previous one. It shows dependent variable, independent variable, but not function prototype

3. $$y''' + a_2y'' + a_1y' + a_0y = 0$$

This is the simplest form and is the most widely used form in engineering. It does not show the function prototype, nor independent variable. In most case, y represents y(t), which means the independent variable is t

4. $$\dddot{y} + a_2\ddot{y} + a_1\dot{y} + a_0y = 0$$

This is almost same representation as the previous one. The only difference is that it has dots on top of each dependent variables in stead of using 'prime(')' symbol.



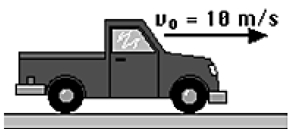
$\frac{d^2y}{dx^2} - \frac{dy}{dx} + 9 = 0$ has order 2 and is 1st degree

$\frac{d^3y}{dx^3} = 78xe^3$ has order 3 and is 1stdegree

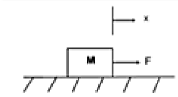
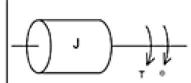
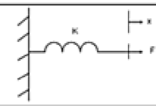
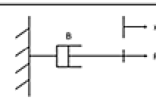
$\left(\frac{dy}{dx}\right)^2 = 5x + 6$ has order 1 and is 2nd degree

DE in Mechanical Engineering

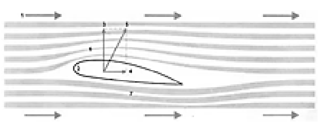
	Derivative Form	Integral Form
Position	$r(t)$	$r(t) = r_0 + \int_0^t v\,dt'$
Velocity	$v(t) = \dfrac{dr}{dt}$	$v(t) = v_0 + \int_0^t a\,dt'$
Acceleration	$a(t) = \dfrac{dv}{dt} = \dfrac{d^2r}{dt^2}$	$a(t)$



$$F = ma = m\,\frac{dv}{dt}$$

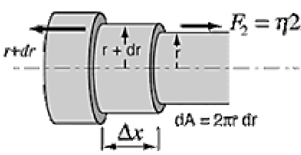
	Symbol	Force/Torque
Mass (Translational)		$F = M\dfrac{d^2x}{dt^2}$
Mass (Rotational)		$T = J\dfrac{d^2\theta}{dt^2}$
Spring		$F = Kx$ (Trans'l.) $T = K\theta$ (Rot'l.)
Dashpot/ Damper		$F = B\dfrac{dx}{dt}$ (Trans'l.) $T = B\dfrac{d\theta}{dt}$ (Rot'l.)

Fluid Mechanics and Thermodynamics



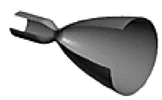
$$F_{viscous} = \eta A\frac{dv}{dr}$$

$$F_1 = \eta 2\pi(r+dr)\Delta x\left[\frac{dv}{dr}\right]_{r+dr} \qquad F_2 = \eta 2\pi r\Delta x\left[\frac{dv}{dr}\right]_r$$



net driving force — drag on outside — forward force on inside

$$-\Delta P 2\pi r dr = \eta 2\pi(r+dr)\Delta x\left[\left[\frac{dv}{dr}\right]_r + \left[\frac{d^2v}{dr^2}\right]_r dr\right] - \eta 2\pi r\Delta x\left[\frac{dv}{dr}\right]_r$$



S = Entropy, T = Temperature, V = Volume, E = Internal Energy, C_p = Heat Capacity (constant pressure)

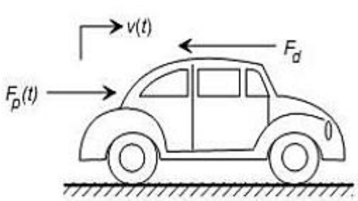
p = Pressure, H = Enthalpy, Q = Heat Transfer, R = Gas Constant, C_v = Heat Capacity (constant volume)

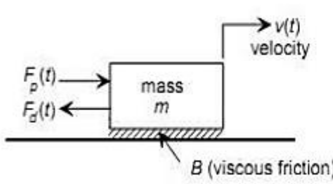
2nd Law of Thermodynamics:	$S_2 - S_1 = \dfrac{\Delta Q}{T}$	differential form: $dS = \dfrac{dQ}{T}$
1st Law for a Gas:	$dQ = dE + p\,dV$ or	$dQ = dH - V\,dp$
Ideal Gas:	$pV = RT$ $dE = C_v\,dT$	$dH = C_p\,dT$
Substitute:	$dQ = C_v\,dT + \dfrac{RT}{V}dV$	$dQ = C_p\,dT - \dfrac{RT}{p}dp$
Substitute:	$dS = C_v\dfrac{dT}{T} + R\dfrac{dV}{V}$	$dS = C_p\dfrac{dT}{T} - R\dfrac{dp}{p}$
Integrate:	$S_2 - S_1 = C_v\ln\dfrac{T_2}{T_1} + R\ln\dfrac{V_2}{V_1}$	$S_2 - S_1 = C_p\ln\dfrac{T_2}{T_1} - R\ln\dfrac{p_2}{p_1}$
specific form:	$s_2 - s_1 = c_v\ln\dfrac{T_2}{T_1} + R\ln\dfrac{v_2}{v_1}$	$s_2 - s_1 = c_p\ln\dfrac{T_2}{T_1} - R\ln\dfrac{p_2}{p_1}$

FINITE DIFFERENCED HEAT EQUATION

$$\boxed{\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\dot{q}}{k} = 0}$$

Dynamic Systems

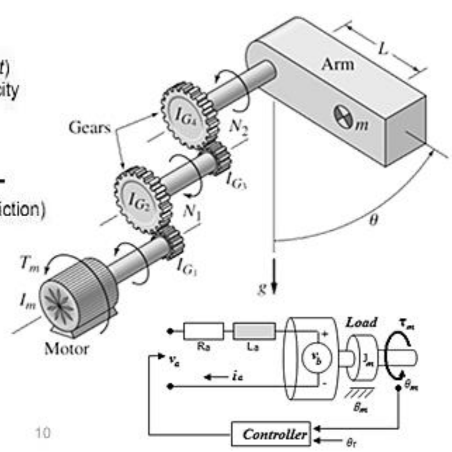




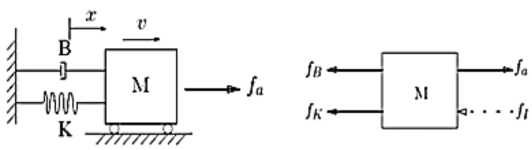
Then from a simple force balance:

$$m\frac{dv}{dt} + Bv = F_p(t) + F_d(t)$$

$$\rightarrow\; m\frac{dv}{dt} + Bv = K_e\theta(t) + F_d(t)$$



Translational Mechanical Systems are those that move along a <u>straight line</u> and mainly composed of basic mechanical elements like *mass*, *spring*, and *dashpot* or *damper*. These systems oppose applied forces through their mass and mechanical material properties like elasticity and frictional resistance.



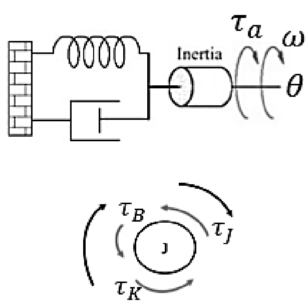
$$F_a(t) - F_M(t) - F_B(t) - F_K(t) = 0$$

$$F_a - M\frac{dv}{dt} - B\frac{dx}{dt} - Kx = 0$$

$$F_a - M\frac{d^2x}{dt} - B\frac{dx}{dt} - Kx = 0$$

$$F_a = M\frac{d^2x}{dt} + B\frac{dx}{dt} + Kx$$

Rotational Mechanical Systems are those that move about a <u>fixed axis</u> and mainly composed of basic mechanical elements like *moment of inertia*, *torsional spring*, and *dashpot*. These systems oppose applied torques through their moment and mechanical material properties like elasticity and frictional resistance



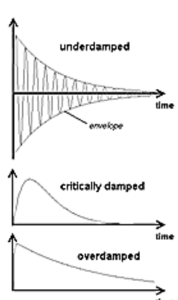
$$\tau_a(t) - \tau_J(t) - \tau_B(t) - \tau_K(t) = 0$$

$$\tau_a - J\frac{d\omega}{dt} - B\frac{d\theta}{dt} - K\theta = 0$$

$$\tau_a - J\frac{d^2\theta}{dt} - B\frac{d\theta}{dt} - K\theta = 0$$

$$\tau_a = J\frac{d^2\theta}{dt} + B\frac{d\theta}{dt} + K\theta$$

Mechanical Damping is generally defined, as the measure or way a system has to dissipate vibration energy, in the forms of sound and heat by generated friction. Despite being small in magnitude, it greatly affects the system response near resonant conditions. ***Natural Frequency*** is the frequency at which the system resonates. It is determined by the stiffness of the object, which acts as a spring, and the amount of mass.

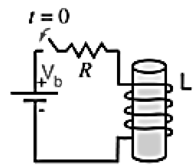


$\zeta = \frac{c}{2\sqrt{mk}}$ the **damping ratio**

$\omega_0 = \sqrt{\frac{k}{m}}$ the **natural frequency** in radians

where: $\zeta = damping\ ratio$
$m = mass$
$k = spring\ constant$
$\omega_0 = natural\ frequency$

DE in Electrical Engineering



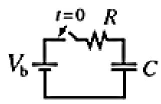
$V_b = V_R + V_L$

$V_b = IR + L\dfrac{\Delta I}{\Delta t}$

As the current increases

$V_b = IR + L\dfrac{\Delta I}{\Delta t}$

the rate of change of current decreases.



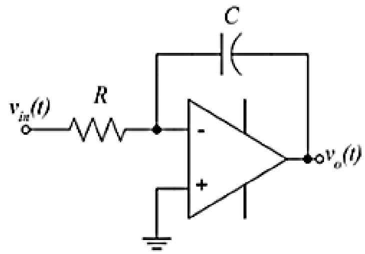
$V_b = V_R + V_C$

$V_b = IR + \dfrac{Q}{C}$

As charging progresses,

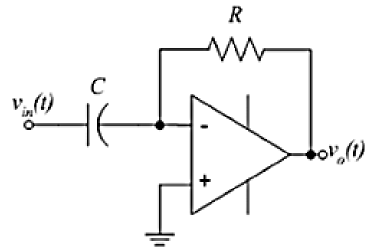
$V_b = IR + \dfrac{Q}{C}$

current decreases and charge increases.



(a) ideal integrator

$$v_o(t) = -\frac{1}{RC}\int_0^t v_{in}(t)dt + v_o(0)$$



(b) ideal differentiator

$$v_o(t) = -RC\frac{dv_{in}(t)}{dt}$$

Important Equations in Electrical Circuits

Ohm's Law $V = IR$

Time Constant $\tau = RC$

$\tau = \dfrac{L}{R}$

Transient Voltage

$V_C = V_0 e^{-t/\tau}$

$V_L = V_0 e^{-t/\tau}$

Resonant Frequency $f_r = \dfrac{1}{2\pi\sqrt{LC}}$

Series Impedance $|Z| = \sqrt{R^2 + {X_T}^2}$ $\quad \emptyset = \tan^{-1}\dfrac{X_T}{R}$

Angular Freq. $\omega = 2\pi f$

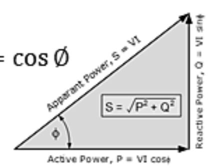
Reactance $X_C = \dfrac{1}{\omega C}$

$X_L = \omega L$

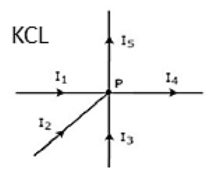
Total Reactance $X_T = X_L - X_C$

Power Factor

$pf = \dfrac{P}{S} = \cos\emptyset$



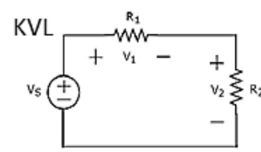
KCL



$-I_1 - I_2 - I_3 + I_4 + I_5 = 0$

$\Rightarrow I_1 + I_2 + I_3 = I_4 + I_5$

KVL

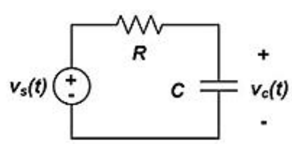


$V_S - V_1 - V_2 = 0$

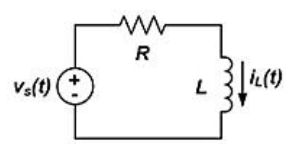
$\Rightarrow V_S = V_1 + V_2$

	Capacitors	Inductors
Differential i-v	$i = C\frac{dv}{dt}$	$v = L\frac{di}{dt}$
Integral i-v	$v_C(t) = \frac{1}{C}\int_{-\infty}^{t} i_C(\tau)\,d\tau$	$i_C(t) = \frac{1}{L}\int_{-\infty}^{t} v_L(\tau)\,d\tau$
DC equivalent	Open-circuit	Short-circuit
Two in series	$C_{eq} = \frac{C_1C_2}{C_1+C_2}$	$L_{eq} = L_1 + L_2$
Two in parallel	$C_{eq} = C_1 + C_2$	$L_{eq} = \frac{L_1L_2}{L_1+L_2}$
Stored energy	$W_C = \frac{1}{2}Cv_C^2$	$W_L = \frac{1}{2}Li_L^2$

1[st] Order Circuits (RC and RL)

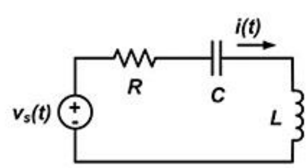


$$\frac{dv(t)}{dt} + \frac{1}{RC}v(t) = 0$$

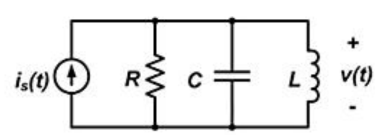


$$\frac{di(t)}{dt} + \frac{R}{L}i(t) = 0$$

2nd Order Circuits (RCL)

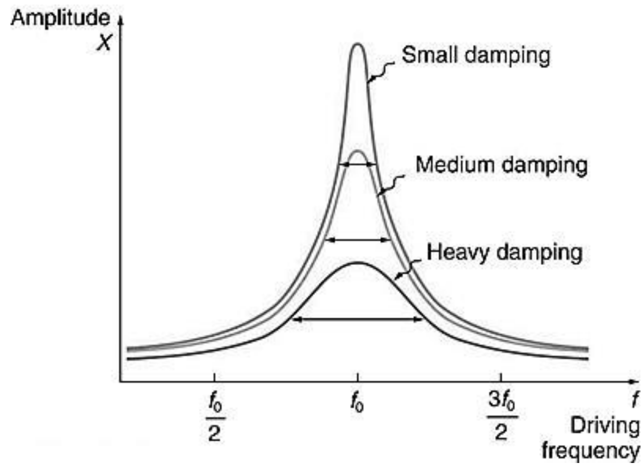


$$\frac{dv_s(t)}{dt} = L\frac{d^2i(t)}{dt^2} + R\frac{di(t)}{dt} + \frac{1}{C}i(t)$$

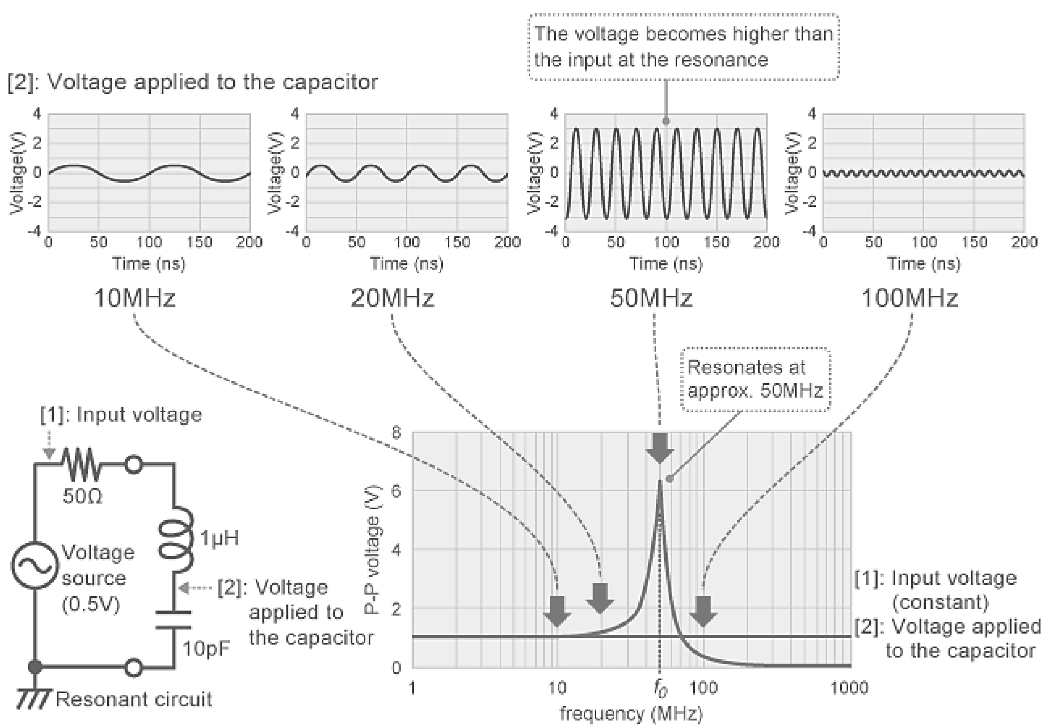


$$\frac{di_s(t)}{dt} = C\frac{d^2v(t)}{dt^2} + \frac{1}{R}\frac{dv(t)}{dt} + \frac{1}{L}v(t)$$

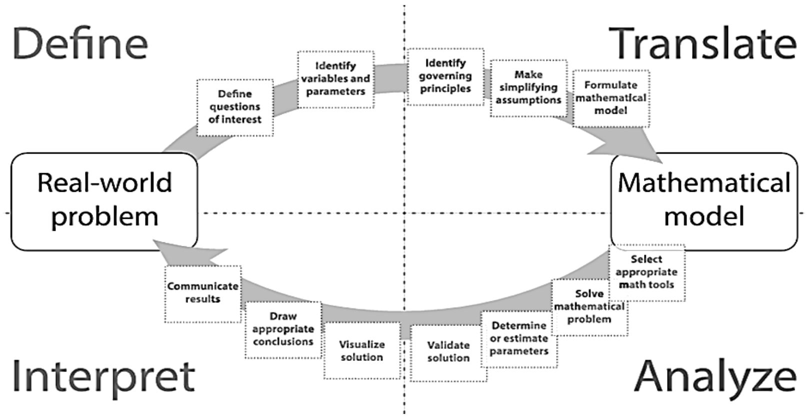
Electrical Damping is a system that provides damping over a wide range of frequencies including high frequencies. It includes a resistance in parallel with an electric motor or series connected resistance and capacitance in parallel with the electric motor.



Electrical Resonance occurs in an electric circuit at a particular *resonant frequency* when the impedances or admittances of circuit elements cancel each other. In some circuits, this happens when the impedance between the input and output of the circuit is almost zero and the transfer function is close to one. When damping is small, the resonant frequency is about equal to the *natural frequency* of the system



Mathematical Modelling is the art of translating problems from an application area into tractable mathematical formulations whose theoretical and numerical analysis provides insight, answers, and guidance useful for the originating application. It is indispensable in many applications.

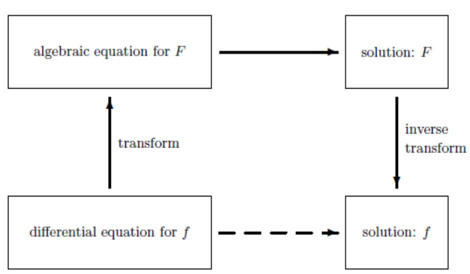


Integral Transforms (or simply *transforms*) are operators used in numerous complex mathematical analyses, especially as a map from functions to functions. They have been successfully used for almost two centuries in solving many problems in applied mathematics, mathematical physics, and engineering science.

Laplace Transform:	$x(t) \Leftrightarrow X(s)$	where $X(s) = \int_{-\infty}^{\infty} x(t)e^{-st}dt$
Continuous-Time Fourier Transform:	$x(t) \Leftrightarrow X(j\omega)$	where $X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt$
z Transform:	$x[n] \Leftrightarrow X(z)$	where $X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}$
Discrete-Time Fourier Transform:	$x[n] \Leftrightarrow X(e^{j\Omega})$	where $X(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n}$

Strategies for solving ODEs (and other similar problems) using integral transforms:

1. Convert the quantities entering our problem into image space, writing the transforms of the known quantities in the problem, and representing any unknown quantities in the problem by symbols denoting their transforms;
2. Simplify and solve the problem in the image space, thereby obtaining the transform of the solution;
3. Obtain the inverse of the transformed solution so as to get the direct-space solution to our problem.



ASSESSMENT

Group Task

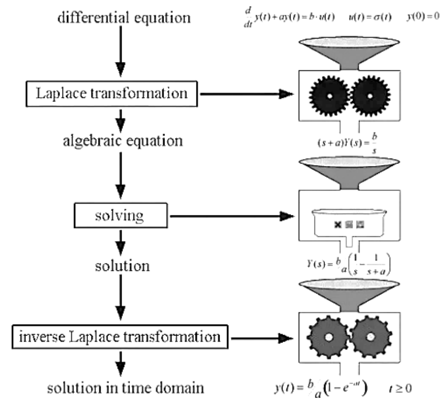
Watch the following YouTube tutorials about MATLAB and perform the tasks they feature:

- ✓ What is Robotics System Toolbox?
- ✓ Work with Mobile Robotics Algorithms in MATLAB
- ✓ Control LPR Manipulator Motion Through Joint Torque Commands
- ✓ Control PR2 Movements using ROS Actions and Inverse Kinematics
- ✓ Implement Simultaneous Localization and Mapping (SLAM)
- ✓ Simulink Blocks for Robot Manipulators and Safe Trajectory Tracking Control

See the discussion proper on the Introduction to MATLAB (Module 4),

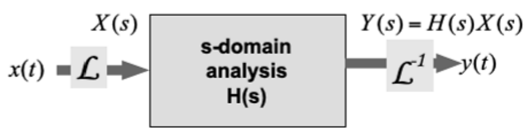
Module 6: **Laplace Transforms**

Laplace Transform is a technique for solving differential equations wherein the DE of time domain form is first transformed into its algebraic equation of frequency domain form, and once such algebraic equation is solved, the result then is finally transformed to time domain form to achieve the ultimate solution of the differential equation.



Laplace Transform is mathematically defined as:

$$F(s) = \mathcal{L}[f(t)] = \int_0^\infty e^{-st} f(t)dt$$



- Laplace Transform of an ***impulse*** function, $\delta(t)$

$$\mathcal{L}[\delta(t)] = \int_0^\infty \delta(t)e^{-st}dt = 1 \qquad \text{for all } s \qquad \boxed{\mathcal{L}[\delta(t)] \Leftrightarrow 1}$$

- Laplace Transform of a ***unit step*** function, $u(t)$

$$\mathcal{L}[u(t)] = \int_0^\infty u(t)e^{-st}dt = \int_0^\infty e^{-st}dt$$

$$= -\frac{1}{s}e^{-st}\Big|_0^\infty = \frac{1}{s} \qquad \operatorname{Re} s > 0 \qquad \boxed{\mathcal{L}[u(t)] \Leftrightarrow \frac{1}{s}}$$

- Laplace Transform of an ***exponential*** function, e^{at}

$$\boxed{\begin{aligned}\mathcal{L}[e^{at}u(t)] &= \int_0^\infty e^{at}e^{-st}dt \\ &= \int_0^\infty e^{-(s-a)t}dt = \frac{1}{s-a}\end{aligned}} \qquad \boxed{\mathcal{L}[e^{at}u(t)] \Leftrightarrow \frac{1}{s-a}}$$

- Laplace Transform of a ***cosine*** function, $\cos\omega_0 t$

$$\boxed{\begin{aligned}\mathcal{L}[\cos\omega_0 t\, u(t)] &= \frac{1}{2}\mathcal{L}[e^{j\omega_0 t}u(t) + e^{-j\omega_0 t}u(t)] \\ &= \frac{1}{2}\left[\frac{1}{s-j\omega_0} + \frac{1}{s+j\omega_0}\right] = \frac{s}{s^2+\omega_0^2}\end{aligned}} \qquad \boxed{\mathcal{L}[\cos\omega_0 t u(t)] \Leftrightarrow \frac{s}{s^2+\omega_0^2}}$$

- Laplace Transform of a ***differential*** function, $\frac{dx}{dt}$

$$\mathcal{L}[\frac{dx(t)}{dt}] = \int_{t=0}^{\infty} \frac{dx(t)}{dt} e^{-st}\, dt$$

$$\mathcal{L}[\dot{x}(t)] = sX(s) - x(0)$$

- Laplace Transform of an ***integral*** function, $\cos\omega_0 t$

Let $g(t) = \int_{\tau=0}^{t} x(\tau)d\tau$

then $x(t) = \frac{dg(t)}{dt}$, and $g(0) = 0$

$$\mathcal{L}[x(t)] = \mathcal{L}[\dot{g}(t)] = sG(s) - g(0) = sG(s)$$

$$\mathcal{L}[g(t)] = \frac{1}{s}X(s)$$

Inverse Laplace Transform is used to transform back the $F(s)$ frequency-domain equivalent of a function into its original time-domain form $f(t)$. Methods include:

- ✓ Direct Method (using the Laplace Transform Definition)
- ✓ Inspection Method (using the Laplace Transform Pairs table and Partial Fraction)
- ✓ Series Method (Taylor Series, Power Series, etc.)

Laplace Transform Pairs are a convenient tool for working on Laplace Transforms. These form a table that shows common functions $f(t)$ and their equivalent Laplace Transform $F(s)$.

$$f(t) \leftrightarrow F(s)$$

$$\mathcal{L}\{f(t)\} = F(s) \quad \leftrightarrow \quad \mathcal{L}^{-1}\{F(s)\} = f(t)$$

	$f(t)=\mathfrak{L}^{-1}\{F(s)\}$	$F(s)=\mathfrak{L}\{f(t)\}$		$f(t)=\mathfrak{L}^{-1}\{F(s)\}$	$F(s)=\mathfrak{L}\{f(t)\}$
1.	1	$\frac{1}{s}$	2.	e^{at}	$\frac{1}{s-a}$
3.	$t^n,\quad n=1,2,3,\ldots$	$\frac{n!}{s^{n+1}}$	4.	$t^p, p > -1$	$\frac{\Gamma(p+1)}{s^{p+1}}$
5.	$\sqrt{t}$	$\frac{\sqrt{\pi}}{2s^{\frac{3}{2}}}$	6.	$t^{n-\frac{1}{2}},\quad n=1,2,3,\ldots$	$\frac{1\cdot3\cdot5\cdots(2n-1)\sqrt{\pi}}{2^n s^{n+\frac{1}{2}}}$
7.	$\sin(at)$	$\frac{a}{s^2+a^2}$	8.	$\cos(at)$	$\frac{s}{s^2+a^2}$
9.	$t\sin(at)$	$\frac{2as}{(s^2+a^2)^2}$	10.	$t\cos(at)$	$\frac{s^2-a^2}{(s^2+a^2)^2}$
11.	$\sin(at)-at\cos(at)$	$\frac{2a^3}{(s^2+a^2)^2}$	12.	$\sin(at)+at\cos(at)$	$\frac{2as^2}{(s^2+a^2)^2}$
13.	$\cos(at)-at\sin(at)$	$\frac{s(s^2-a^2)}{(s^2+a^2)^2}$	14.	$\cos(at)+at\sin(at)$	$\frac{s(s^2+3a^2)}{(s^2+a^2)^2}$
15.	$\sin(at+b)$	$\frac{s\sin(b)+a\cos(b)}{s^2+a^2}$	16.	$\cos(at+b)$	$\frac{s\cos(b)-a\sin(b)}{s^2+a^2}$
17.	$\sinh(at)$	$\frac{a}{s^2-a^2}$	18.	$\cosh(at)$	$\frac{s}{s^2-a^2}$
19.	$e^{at}\sin(bt)$	$\frac{b}{(s-a)^2+b^2}$	20.	$e^{at}\cos(bt)$	$\frac{s-a}{(s-a)^2+b^2}$

Laplace Transform of Derivatives

$$\mathcal{L}\{f'(t)\} = s\mathcal{L}\{f(t)\} - f(0) = sF(s) - f(0)$$
$$\mathcal{L}\{f''(t)\} = s^2\mathcal{L}\{f(t)\} - sf(0) - f'(0) = s^2F(s) - sf(0) - f'(0)$$
$$\vdots$$
$$\mathcal{L}\left\{f^{(n)}(t)\right\} = s^nF(s) - s^{n-1}f(0) - s^{n-2}f'(0) - \cdots - f^{(n-1)}(0)$$

Laplace Transform of Integrals

$$\text{If } L\{f(t)\} = F(s), \quad then$$
$$L\left[\iint\iint\int \dots \int f(t)dt^n\right] = \frac{1}{s^n}L\{f(t)\} + \frac{f^{n-1}(0)}{s^n} + \frac{f^{n-2}(0)}{s^n} + \dots + \frac{f^n(0)}{s}$$

For eg. Let n = 1

$$L\left\{\int_0^t f(t)\right\} = \frac{1}{s}L\{f(t)\} + \frac{f'(0)}{s}$$

Properties of Laplace Transforms

1. Linearity: Let C_1, C_2 be constants. f(t), g(t) be the functions of time, t, then

$$L\{C_1f(t) + C_2g(t)\} = C_1L\{f(t)\} + C_2L\{g(t)\}$$

2. First shifting Theorem:

$$\text{If } L\{f(t)\} = F(s), \quad then\ L\{e^{at}f(t)\} = F(s-a)$$

3. Change of scale property:

$$\text{If } L\{f(t)\} = F(s), \quad then$$
$$L\{f(at)\} = \frac{1}{a}F\left(\frac{s}{a}\right); \text{ Frequency Scaling}$$
$$L\left\{f\left(\frac{t}{a}\right)\right\} = aF(as) \quad ; \text{ Time Scaling}$$

4. Time-Shifting

$$L\{f(t-T)u(t-T)\} = e^{-st}F(s)$$

5. Product

$$L\{f_1(t)f_2(t)\} = \frac{1}{2\pi j}\int_{c-j\infty}^{c+j\infty} F_1(\omega)F_2(\omega)\,d\omega$$

6. Final Value Theorem

$$\text{If } L\{f(t)\} = F(s), \quad then \ \lim_{s\to\infty} f(t) = \lim_{s\to 0} sF(s)$$

7. Initial Value Theorem

$$\text{If } L\{f(t)\} = F(s), \quad then \quad f(0^+) = \lim_{s\to\infty} sF(s)$$

Example 1

Solve $\frac{d^2y}{dt^2} + y = 0 \ given \ y = 1; \ \frac{dy}{dt} = 0 \ when \ t = 0$

⇨ The given differential equation can be written as

$y'' + y = 0, y(0) = 1, y'(0) = 0 \ - Eqn.(i)$

Taking Laplace Transform on both sides

$L(y'') + L(y) = L(0)$

By differentiation theorem

$p^2L(y) - p \cdot y(0) - y'(0) + L(y) = 0$

⇨ $p^2L(y) - p \cdot 1 - 0 + L(y) = 0$

⇨ $L(y) = \frac{p}{p^2+1}$

Taking Inverse Laplace Transform

$$y = L^{-1}\frac{p}{p^2+1} = \cos(t)$$

Example 2

$$\frac{\mathrm{d}^2x}{\mathrm{d}t^2} + \frac{\mathrm{d}x}{\mathrm{d}t} + x = 0 \ given \ x(0^+) = 0 \ and \ x'(0^+) = 1$$

$$\mathcal{L}\frac{\mathrm{d}^2 f(t)}{\mathrm{d}t^2} = s^2\mathcal{L}f(t) - sf(0^+) - f'(0^+)$$

$$\therefore \mathcal{L}\frac{\mathrm{d}^2 x}{\mathrm{d}t^2} = s^2\mathcal{L}(x) - sx(0^+) - x'(0^+)$$

$$= s^2\mathcal{L}(x) - s.0 - 1 = s^2\mathcal{L}(x) - 1$$

$$Again, \ \mathcal{L}\left[\frac{\mathrm{d}f(t)}{\mathrm{d}t}\right] = s\mathcal{L}f(t) - f(0^+)$$

$$\mathcal{L}\left[\frac{\mathrm{d}x}{\mathrm{d}t}\right] = s\mathcal{L}(x) - x(0^+) = s\mathcal{L}(x) - 0 = s\mathcal{L}(x)$$

$$Therefore, \ \mathcal{L}\left[\frac{\mathrm{d}^2x}{\mathrm{d}t^2} + \frac{\mathrm{d}x}{\mathrm{d}t} + x\right] = \mathcal{L}(0)$$

$$\mathcal{L}\left[\frac{\mathrm{d}^2x}{\mathrm{d}t^2}\right] + \mathcal{L}\left[\frac{\mathrm{d}x}{\mathrm{d}t}\right] + \mathcal{L}[x] = 0$$

$$\Rightarrow s^2\mathcal{L}(x) - 1 + s\mathcal{L}(x) + \mathcal{L}(x) = 0$$

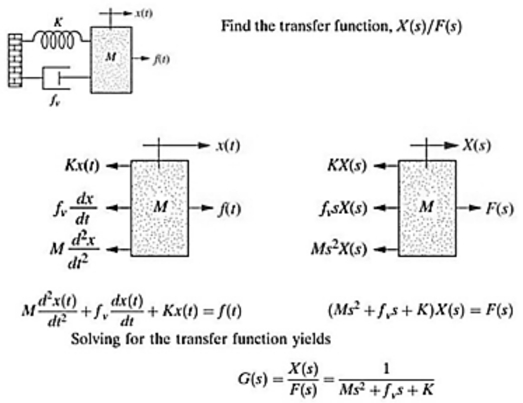
$$\Rightarrow \mathcal{L}(x)\left[s^2 + s + 1\right] = 1$$

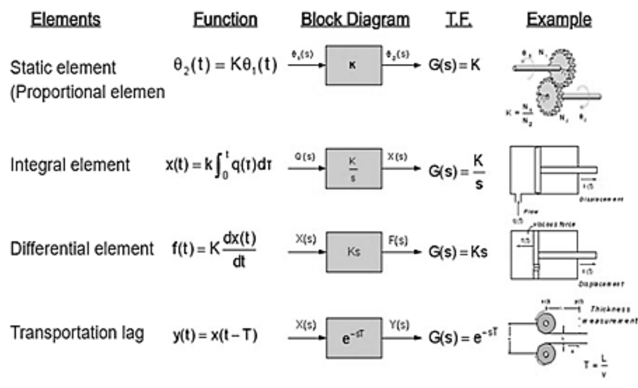
$$\Rightarrow \mathcal{L}(x) = \frac{1}{s^2 + s + 1} \quad = \frac{1}{s^2 + 2\frac{1}{2}s + \frac{1}{4} + \frac{3}{4}}$$

$$= \frac{1}{\left(s+\frac{1}{2}\right)^2 + \left(\frac{\sqrt{3}}{2}\right)^2} \quad = \frac{2}{\sqrt{3}} \times \frac{\left(\frac{\sqrt{3}}{2}\right)}{\left(s+\frac{1}{2}\right)^2 + \left(\frac{\sqrt{3}}{2}\right)^2}$$

$$\therefore \mathcal{L}^{-1}\mathcal{L}(x) = x = \frac{2}{\sqrt{3}}\, e^{-0.5t} \sin\frac{\sqrt{3}}{2}t$$

Laplace Transform in Mechanical Systems

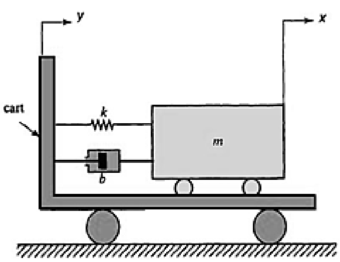




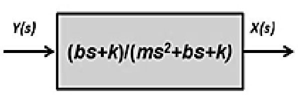
Elements	Function	Block Diagram	T.F.	Example
Static element (Proportional elemen	$\theta_2(t)=K\theta_1(t)$	$\theta_1(s) \rightarrow K \rightarrow \theta_2(s)$	$G(s)=K$	
Integral element	$x(t)=k\int_0^t q(\tau)d\tau$	$Q(s) \rightarrow \frac{K}{s} \rightarrow X(s)$	$G(s)=\frac{K}{s}$	
Differential element	$f(t)=K\frac{dx(t)}{dt}$	$X(s) \rightarrow Ks \rightarrow F(s)$	$G(s)=Ks$	
Transportation lag	$y(t)=x(t-T)$	$X(s) \rightarrow e^{-sT} \rightarrow Y(s)$	$G(s)=e^{-sT}$	

Transfer Function of Mechanical Systems

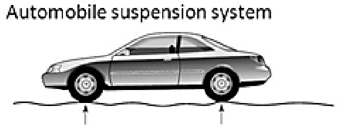
Consider the spring-mass-dashpot system mounted on a cart as shown. Derive the transfer function *G(s)* relating mass displacement (*x*) to cart displacement (y).

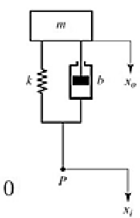


$$\sum F_x = ma_x$$
$$k(y-x)+b(\dot{y}-\dot{x})=m\ddot{x}$$
$$m\ddot{x}+b\dot{x}+kx=b\dot{y}+ky$$
$$ms^2X+bsX+kX=bsY+kY$$
$$G(s)=\frac{X(s)}{Y(s)}=\frac{bs+k}{ms^2+bs+k}$$



Automobile suspension system





The equation of motion for the system

$$m\ddot{x}_o + b(\dot{x}_o - \dot{x}_i) + k(x_o - x_i) = 0$$
$$m\ddot{x}_o + b\dot{x}_o + kx_o = b\dot{x}_i + kx_i$$

Taking the Laplace transform

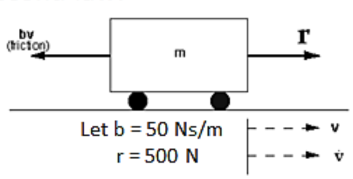
$$(ms^2 + bs + k)X_o(s) = (bs + k)X_i(s)$$

Transfer function $\frac{X_o(s)}{X_i(s)} = \frac{bs + k}{ms^2 + bs + k}$

Example: Vehicle Dynamics

- To get an ODE for the system we write down Newton's second law:

$$F = ma = m\frac{dv}{dt}$$



- Summing the interacting forces at the center of mass, we get:

$$r(t) - bv(t) = m\frac{dv}{dt}$$
$$m\frac{dv}{dt} + bv(t) = r(t)$$

- Assuming 0 initial conditions,

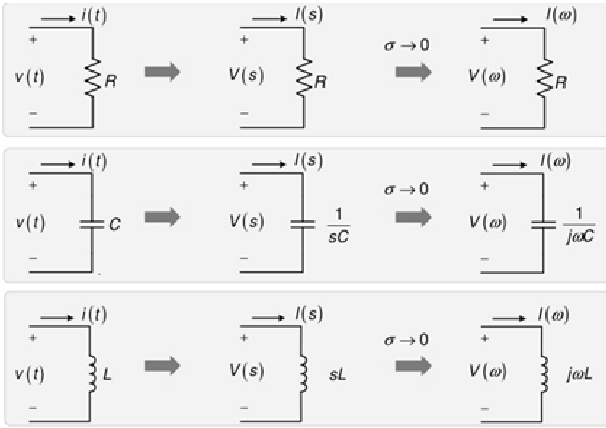
$$\Rightarrow msV(s) + bV(s) = R(s)$$
$$\Rightarrow G(s) = \frac{V(s)}{R(s)} = \frac{1}{ms + b}$$

$$G(s) = \frac{1}{1000s + 50} \longrightarrow$$ Transfer Function

- Therefore, we solve *v(t)* as

$$V(s) = \frac{500}{s(1000s + 50)} \qquad v(t) = 10\left(u(t) - e^{-\frac{t}{20}}\right)$$

Laplace Transform in Electrical Systems



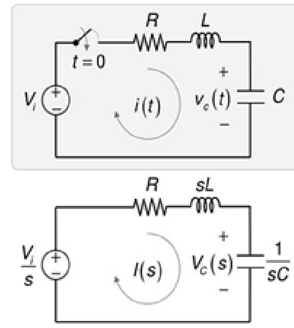
Second-order circuits are of special interest because they are capable of displaying, on a simple scales, the types of responses that appear in circuits of arbitrary order. In fact, second-order circuits and systems occur frequently in practical applications, so their behavior is subject of considerable interest.

Series *RLC* Circuit

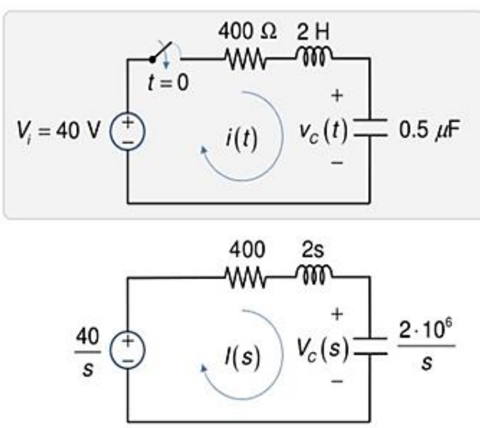
$$-\frac{V_i}{s} + sLI(s) + RI(s) + \frac{1}{sC}I(s) = 0$$

$$I(s) = \frac{\frac{V_i}{L}}{s^2 + s\frac{R}{L} + \frac{1}{LC}}$$

$$V_c(s) = \frac{1}{sC}I(s) = \frac{\frac{V_i}{LC}}{s\left(s^2 + s\frac{R}{L} + \frac{1}{LC}\right)}$$



Example: Determine the current $i(t)$ and voltage $v_c(t)$ for $t > 0$.



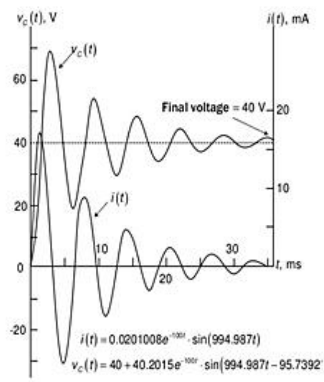
$$Z(s) = 2s + 400 + \frac{2\cdot 10^6}{s} = \frac{2s^2 + 400s + 2\cdot 10^6}{s}$$

$$I(s) = \frac{V(s)}{Z(s)} = \frac{\frac{40}{s}}{\frac{2s^2 + 400s + 2\cdot 10^6}{s}} = \frac{20}{s^2 + 200s + 10^6} = \frac{20}{(s+100)^2 + 994.987^2}$$

$$i(t) = 0.0201008e^{-100t}\cdot\sin(994.987t)$$

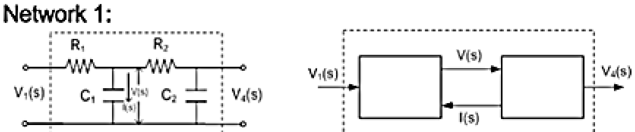
$$V(s) = I(s)Z(s) = \frac{40\cdot 10^6}{s(s^2 + 200s + 10^6)}$$

$$v_c(t) = 40 + 40.2015e^{-100t} \times\sin(994.987t - 95.7392^\circ)$$



Transfer Function in Electrical Systems

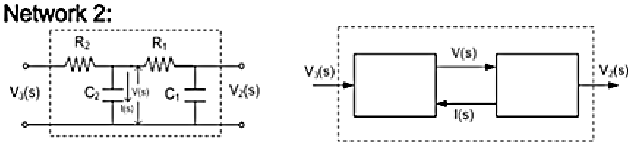
- **Network 1:**



$$\Rightarrow G^1(s) = \frac{1}{(R_1C_1R_2C_2)s^2 + (R_1C_1 + R_2C_2 + \boxed{R_1C_2})s + 1}$$

Loading effect

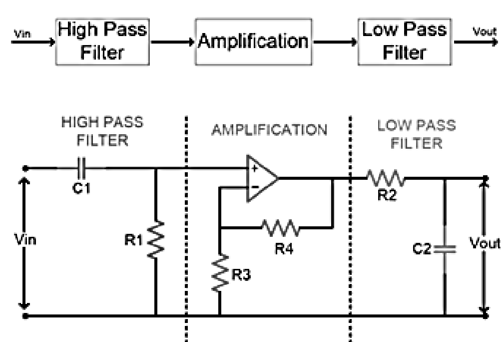
- **Network 2:**



$$\Rightarrow G^2(s) = \frac{1}{(R_1C_1R_2C_2)s^2 + (R_1C_1 + R_2C_2 + \boxed{R_2C_1})s + 1}$$

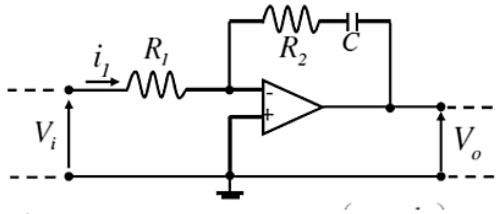
Loading effect

$$G^1(s) \neq G_1(s)G_2(s) \neq G^2(s)$$



Example: Operational Amplifiers

To get an ODE for the system we write down the s-domain equation of the circuit input and output voltages using their Laplace equivalents:



$$V_i(s) = R_1 I_1(s) \quad \text{and} \quad V_o(s) = -\left(R_2 + \frac{1}{s \cdot C}\right) I_1(s)$$

$$G(s) = \frac{V_o(s)}{V_i(s)} = -\frac{R_2 \cdot C \cdot s + 1}{R_1 \cdot C \cdot s} \qquad \text{or generally,} \qquad G(s) = \frac{V_o(s)}{V_i(s)} = -\frac{Z_f(s)}{Z_i(s)}$$

Transfer Function

MATLAB tf (Transfer Function Model)

Description

Use `tf` to create real-valued or complex-valued transfer function models, or to convert dynamic system models to transfer function form.

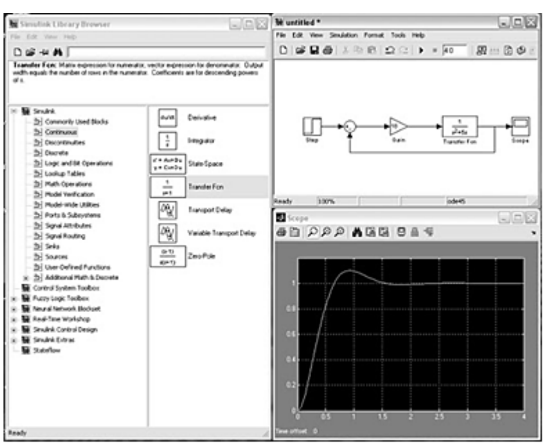
Transfer functions are a frequency-domain representation of linear time-invariant systems. For instance, consider a continuous-time SISO dynamic system represented by the transfer function `sys(s) = N(s)/D(s)`, where `s = jw` and `N(s)` and `D(s)` are called the numerator and denominator polynomials, respectively. The `tf` model object can represent SISO or MIMO transfer functions in continuous time or discrete time.

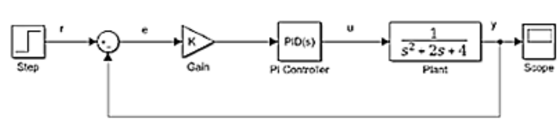
You can create a transfer function model object either by specifying its coefficients directly, or by converting a model of another type (such as a state-space model `ss`) to transfer-function form. For more information, see Transfer Functions.

You can also use `tf` to create generalized state-space (`genss`) models or uncertain state-space (`uss`) models.

Syntax	Description
`sys = tf(numerator,denominator)`	`sys = tf(numerator,denominator)` creates a continuous-time transfer function model, setting the `Numerator` and `Denominator` properties. For instance, consider a continuous-time SISO dynamic system represented by the transfer function `sys(s) = N(s)/D(s)`, the input arguments `numerator` and `denominator` are the coefficients of `N(s)` and `D(s)`, respectively.
`sys = tf(numerator,denominator,ts)`	`sys = tf(numerator,denominator,ts)` creates a discrete-time transfer function model, setting the `Numerator`, `Denominator`, and `Ts` properties. For instance, consider a discrete-time SISO dynamic system represented by the transfer function `sys(z) = N(z)/D(z)`, the input arguments `numerator` and `denominator` are the coefficients of `N(z)` and `D(z)`, respectively. To leave the sample time unspecified, set `ts` input argument to -1.
`sys = tf(numerator,denominator,ltiSys)`	`sys = tf(numerator,denominator,ltiSys)` creates a transfer function model with properties inherited from the dynamic system model `ltiSys`, including the sample time.
`sys = tf(m)`	`sys = tf(m)` creates a transfer function model that represents the static gain, `m`.
`sys = tf(__,Name,Value)`	`sys = tf(__,Name,Value)` sets properties of the transfer function model using one or more `Name,Value` pair arguments for any of the previous input-argument combinations.
`sys = tf(ltiSys)`	`sys = tf(ltiSys)` converts the dynamic system model `ltiSys` to a transfer function model.
`sys = tf(ltiSys,component)`	`sys = tf(ltiSys,component)` converts the specified `component` of `ltiSys` to transfer function form. Use this syntax only when `ltiSys` is an identified linear time-invariant (LTI) model.
`s = tf('s')`	`s = tf('s')` creates special variable `s` that you can use in a rational expression to create a continuous-time transfer function model. Using a rational expression can sometimes be easier and more intuitive than specifying polynomial coefficients.
`z = tf('z',ts)`	`z = tf('z',ts)` creates special variable `z` that you can use in a rational expression to create a discrete-time transfer function model. To leave the sample time unspecified, set `ts` input argument to -1.

See Examples





Converting From	Converting to	Matlab function
Transfer Function	Zero-pole-gain	`[z,p,k]=tf2zp(num,den)`
Transfer Function	State Space	`[A,B,C,D]=tf2ss(num,den)`
Zero-pole-gain	Transfer Function	`[num,den]=zp2tf(z,p,k)`
Zero-pole-gain	State Space	`[A,B,C,D]=zp2ss(z,p,k)`
State Space	Transfer Function	`[num,den]=ss2tf(A,B,C,D)`
State Space	Zero-pole-gain	`[z,p,k]=ss2zp(A,B,C,D)`

ASSESSMENT

Group Task

Watch the following YouTube tutorials about MATLAB and perform the tasks they feature:

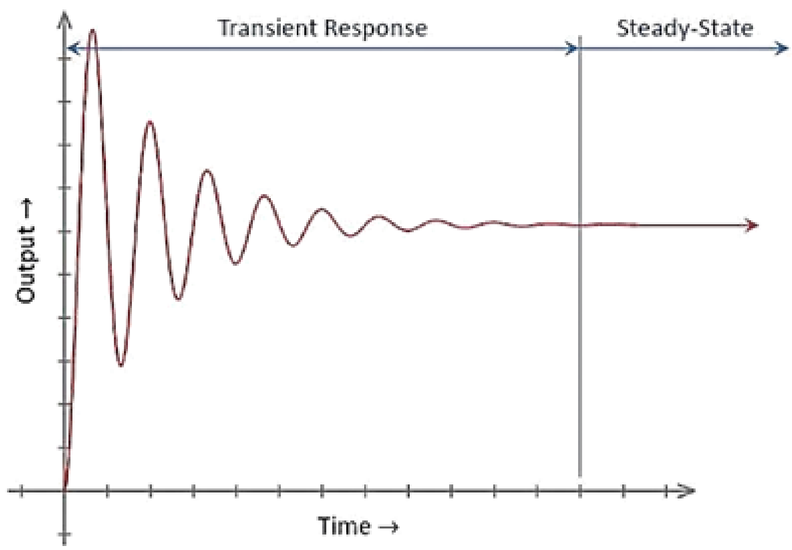
- ✓ Learning Robotics with MATLAB and Simulink
- ✓ Installing Hardware Support Packages for MATLAB and Simulink
- ✓ Simulink Quick Start for Student Competition Teams
- ✓ Stateflow Quick Start for Student Competition Teams
- ✓ LEGO Workshop for Mobile Robotics

See the discussion proper on the Introduction to MATLAB (Module 4),

Module 7: **Time Response and Stability**

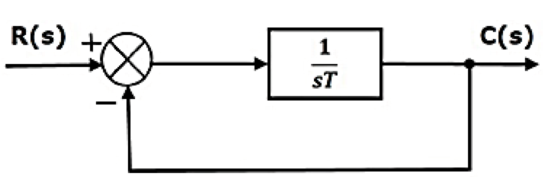
Time Response occurs when the output of control system for an input varies with respect to time. It consists of two parts:

- ✓ Transient Response – the response of the control system during the transient state
- ✓ Stead-State Response – part where the transient response will be zero



$$c(t) = c_{tr}(t) + c_{ss}(t)$$

Response of the First-Order System

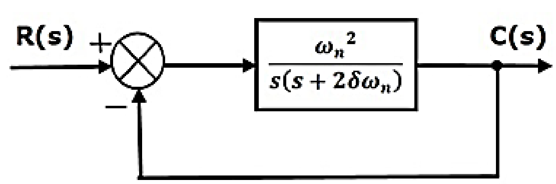


$$C(s) = \left(\frac{1}{sT+1}\right)R(s)$$

where,

- **C(s)** is the LT of the output signal c(t)
- **R(s)** is the LT of the input signal r(t)
- **T** is the time constant.

Response of the Second-Order System



$$C(s) = \left(\frac{\omega_n^2}{s^2+2\delta\omega_n s+\omega_n^2}\right)R(s)$$

Where,

- **C(s)** is the LT of the output signal c(t),
- **R(s)** is the LT of the input signal r(t),
- ω_n is the natural frequency
- **δ** is the damping ratio.

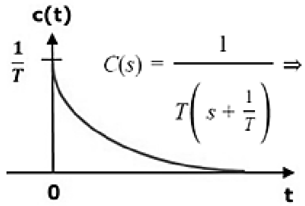
<u>Impulse Response</u> – consider the ***unit impulse signal*** as an input to the first order system

$$r(t) = \delta(t) \quad \Rightarrow \quad R(s) = 1$$

$$C(s) = \left(\frac{1}{sT+1}\right)R(s)$$

$$C(s) = \left(\frac{1}{sT+1}\right)(1) = \frac{1}{sT+1}$$

$$C(s) = \frac{1}{T\left(s+\frac{1}{T}\right)} \Rightarrow C(s) = \frac{1}{T}\left(\frac{1}{s+\frac{1}{T}}\right)$$



<u>Step Response</u> – consider the ***unit step signal*** as an input to first order system

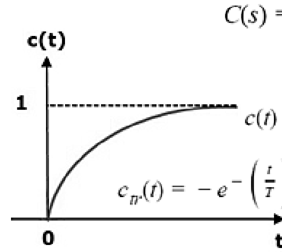
$$r(t) = u(t) \quad \Rightarrow \quad R(s) = \frac{1}{s}$$

$$C(s) = \left(\frac{1}{sT+1}\right)R(s) \qquad C(s) = \left(\frac{1}{sT+1}\right)\left(\frac{1}{s}\right) = \frac{1}{s(sT+1)}$$

$$C(s) = \frac{1}{s} - \frac{1}{s+\frac{1}{T}}$$

$$c(t) = \left(1 - e^{-\left(\frac{t}{T}\right)}\right)u(t)$$

$$c_{tr}(t) = -e^{-\left(\frac{t}{T}\right)}u(t) \qquad c_{ss}(t) = u(t)$$



<u>Ramp Response</u> – consider the ***unit ramp signal*** as an input to first order system

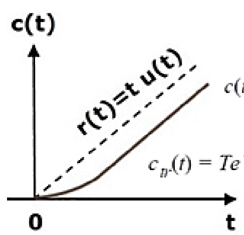
$$r(t) = tu(t) \quad \Rightarrow \quad R(s) = \frac{1}{s^2}$$

$$C(s) = \left(\frac{1}{sT+1}\right)R(s) \qquad C(s) = \left(\frac{1}{sT+1}\right)\left(\frac{1}{s^2}\right) = \frac{1}{s^2(sT+1)}$$

$$C(s) = \frac{1}{s^2} - \frac{T}{s} + \frac{T}{s+\frac{1}{T}}$$

$$c(t) = \left(t - T + Te^{-\left(\frac{t}{T}\right)}\right)u(t)$$

$$c_{tr}(t) = Te^{-\left(\frac{t}{T}\right)}u(t) \qquad c_{ss}(t) = (t-T)u(t)$$



<u>Parabolic Response</u> – consider the ***parabolic signal*** as an input to first order system

$$r(t) = \frac{t^2}{2}u(t) \quad \Rightarrow \quad R(s) = \frac{1}{s^3}$$

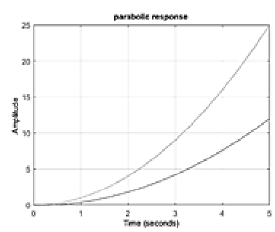
$$C(s) = \left(\frac{1}{sT+1}\right)R(s) \qquad C(s) = \left(\frac{1}{sT+1}\right)\left(\frac{1}{s^3}\right) = \frac{1}{s^3(sT+1)}$$

$$C(s) = \frac{1}{s^3} - \frac{T}{s^2} + \frac{T^2}{s} - \frac{T^2}{s+\frac{1}{T}}$$

$$c(t) = \left(\frac{t^2}{2} - Tt + T^2 - T^2e^{-\left(\frac{t}{T}\right)}\right)u(t)$$

$$C_{tr}(t) = -T^2e^{-\left(\frac{t}{T}\right)}u(t)$$

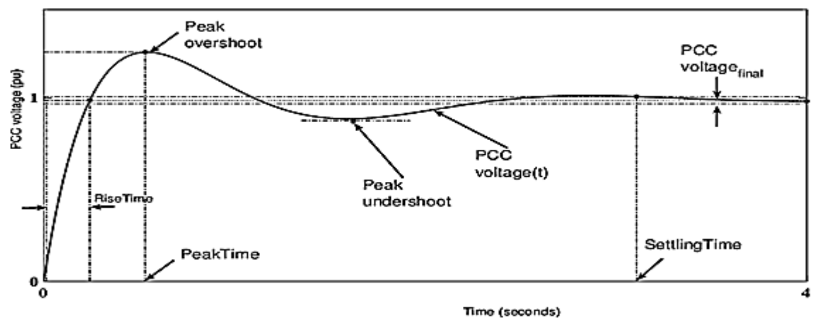
$$C_{ss}(t) = \left(\frac{t^2}{2} - Tt + T^2\right)u(t)$$



- From those responses, we can conclude that the first order control systems are <u>not stable</u> with the <u>ramp and parabolic inputs</u> because these responses go on increasing even at infinite amount of time.
- The first order control systems are stable with impulse and step inputs because these responses have bounded output. But, the impulse response doesn't have steady state term. So, the <u>step signal is widely used in the time domain</u> for analyzing the control systems from their responses.

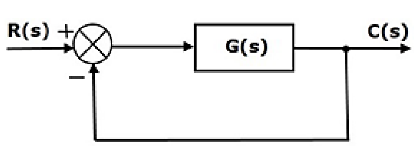
Time-Domain Specifications

- <u>Delay Time</u> (t_d) – time required for the response to reach *half of its final value* from zero instant
- <u>Rise Time</u> (t_r) – time required for the response to rise from *0% to 100%* for underdamped and from 10% to 90% for overdamped systems
- <u>Peak Time</u> (t_p) – time required for the response to reach the *peak value* for the first time
- <u>Peak Overshoot</u> (M_p) – deviation of the response at t_p from the final value of response.
- <u>Setting Time</u> (t_s) – time required for the response to reach the steady state and stay within the specified tolerance bands around the final value. In general, the tolerance bands are 2% and 5%.



Time domain specification	Formula	Substitution of values in Formula	Final value
Delay time	$t_d = \frac{1+0.7\delta}{\omega_n}$	$t_d = \frac{1+0.7(0.5)}{2}$	t_d=0.675 sec
Rise time	$t_r = \frac{\pi-\theta}{\omega_d}$	$t_r = \frac{\pi-(\frac{\pi}{3})}{1.732}$	t_r=1.207 sec
Peak time	$t_p = \frac{\pi}{\omega_d}$	$t_p = \frac{\pi}{1.732}$	t_p=1.813 sec
% Peak overshoot	$\%M_p = \left(e^{-\left(\frac{\delta\pi}{\sqrt{1-\delta^2}}\right)}\right) \times 100\%$	$\%M_p = \left(e^{-\left(\frac{0.5\pi}{\sqrt{1-(0.5)^2}}\right)}\right) \times 100\%$	$\%M_p$=16.32%
Settling time for 2% tolerance band	$t_s = \frac{4}{\delta\omega_n}$	$t_s = \frac{4}{(0.5)(2)}$	t_s=4 sec

Steady-State Error refers to the deviation of the output of control system from desired response during steady state.



$$E(s) = R(s) - C(s)$$

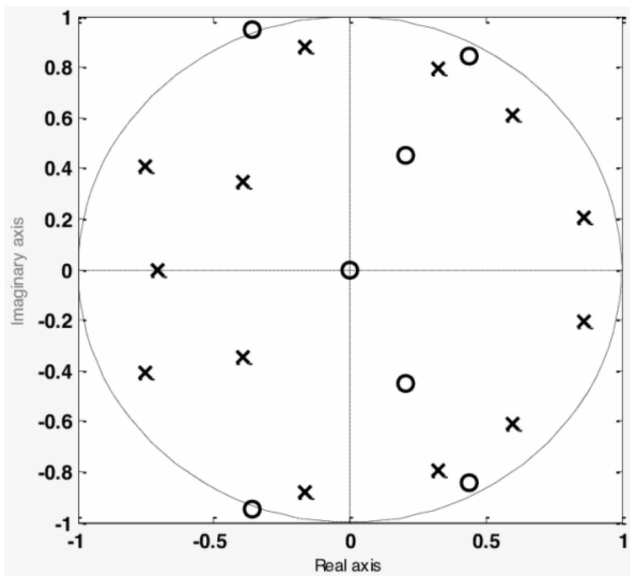
$$E(s) = \frac{R(s)}{1 + G(s)}$$

Input signal	Steady state error e_{ss}	Error constant
unit step signal	$\frac{1}{1+k_p}$	$K_p = \lim_{s\to 0} G(s)$
unit ramp signal	$\frac{1}{K_v}$	$K_v = \lim_{s\to 0} sG(s)$
unit parabolic signal	$\frac{1}{K_a}$	$K_a = \lim_{s\to 0} s^2G(s)$

S-Plane is a complex plane representing $s = \sigma + j\omega$ on which Laplace Transform is graphed; the position on the complex plane is given by $re^{j\theta}$and the angle θ, From the positive, real axis σ around the plane is denoted by where the *poles* and *zeros* are plotted to analyze the dynamic behavior of a transfer function.

$$G(s) = K\frac{N(s)}{D(s)} = \frac{K(s - z_1)(s - z_2)\ldots(s - z_m)}{(s - p_1)(s - p_2)\ldots(s - p_n)}$$

- The *zeros* $(z_1, z_2, \ldots z_n)$ are the roots of the numerator $N(s)$, plotted as O
- The *poles* $(p_1, p_2, \ldots p_n)$ are the roots of the denominator $D(s)$, plotted as X



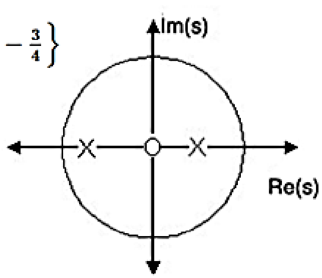
Example: Simple Pole/Zero Plot

$$H(s) = \frac{s}{\left(s - \frac{1}{2}\right)\left(s + \frac{3}{4}\right)}$$

The zeros are: $\{0\}$

The poles are: $\left\{\frac{1}{2}, -\frac{3}{4}\right\}$

Pole/Zero Plot



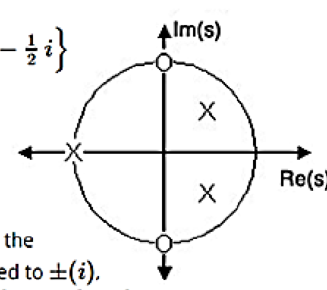
Example: Complex Pole/Zero Plot

$$H(s) = \frac{(s - i)(s + i)}{\left(s - \left(\frac{1}{2} - \frac{1}{2}i\right)\right)\left(s - \frac{1}{2} + \frac{1}{2}i\right)}$$

The zeros are: $\{i, -i\}$

The poles are: $\left\{-1, \frac{1}{2} + \frac{1}{2}i, \frac{1}{2} - \frac{1}{2}i\right\}$

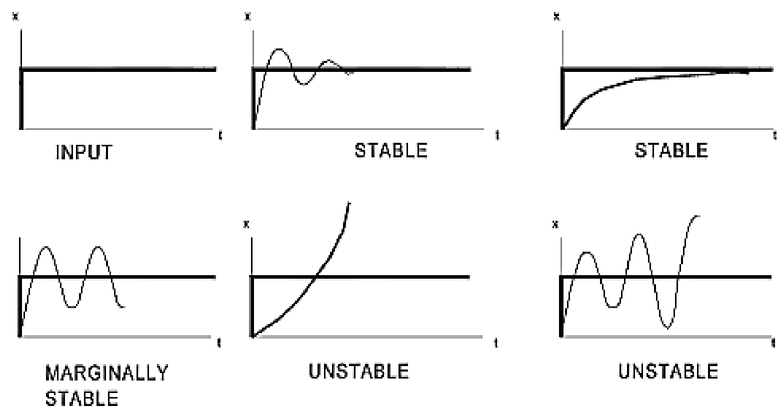
Pole/Zero Plot



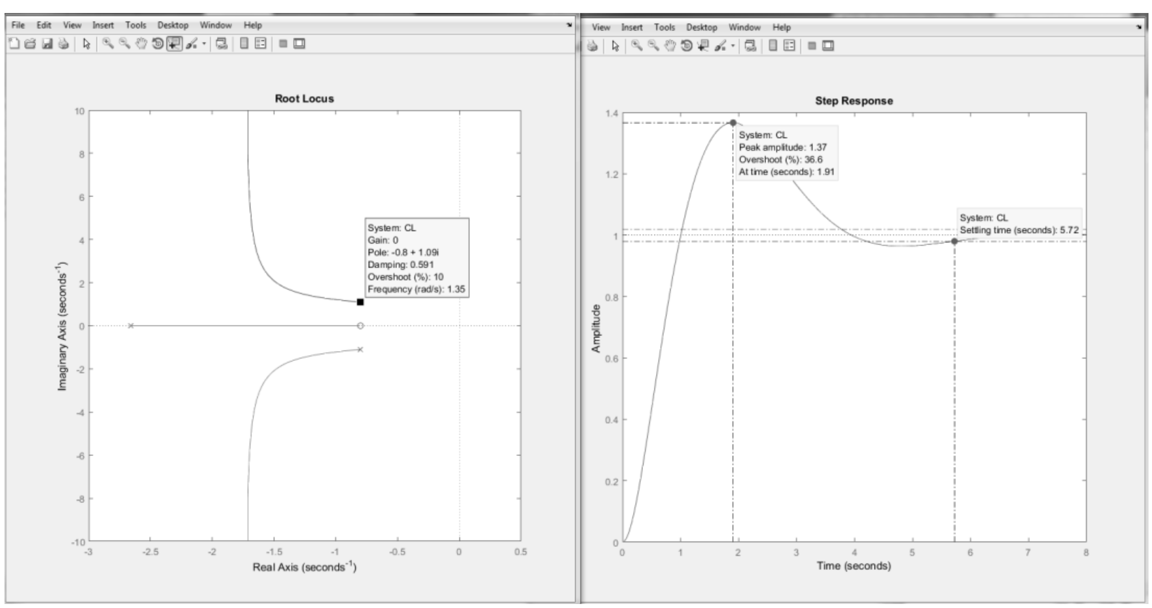
Using the zeros and poles found from the transfer function, the zeros are mapped to $\pm(i)$, and the poles are placed at -1, $\frac{1}{2} + \frac{1}{2}i$ and $\frac{1}{2} - \frac{1}{2}i$

A ***stable system*** produces a bounded output for a given bounded input. Its output is under control. Otherwise, it is said to be unstable. The following are the basic conditions for system stability analysis using the *s-plane*:

- <u>Stable</u> – all poles of the system lie in the left half plane
- <u>Unstable</u> – all poles of the system lie in the right half plane
- <u>Marginally Stable</u> – all poles of the system lie on the imaginary axis
- <u>Absolute Stability</u> – independent on location of zeros



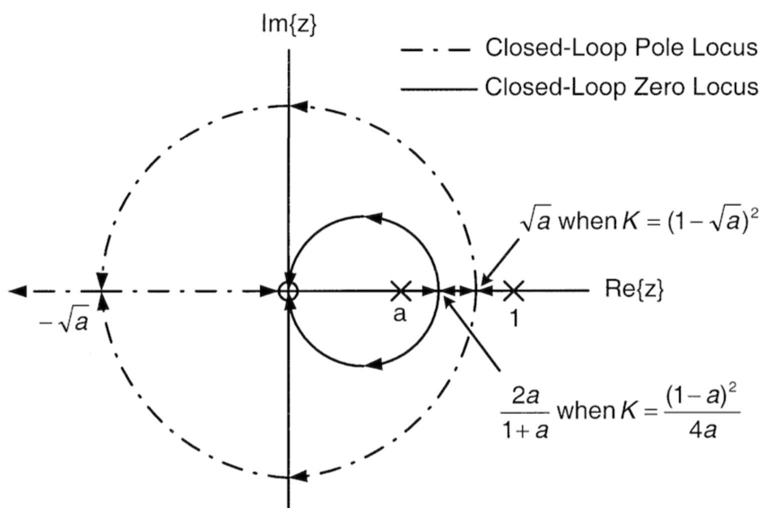
Root Locus Diagram is used to observe the path of the closed loop poles to identify the nature of the control system. It uses an open loop transfer function to know the stability of the closed loop control system. It is also a graphical representation in *s-domain* and it is symmetrical about the real axis. *Root Locus* is the set of all points (locus) of the roots of the characteristic equation by varying system gain from zero to infinity.



The characteristic equation of the closed loop control system is $1 + G(s)H(s) = 0$, so we can represent $G(s)H(s)$ as

$$G(s)H(s) = K\frac{N(s)}{D(s)}$$

- Case 1: If $K = 0$, then $D(s) = 0$. Closed loop poles are equal to open loop poles when *K* is zero.
- Case 2: If $K = \infty$, then $N(s) = 0$. Closed loop poles are equal to the open loop zeros when *K* is infinity.



Therefore, the root locus branches <u>start at open loop poles</u> and <u>end at open loop zeros</u>.

Angle Condition describes the point at which the angle of the open loop transfer function is an odd multiple of ***180^O***, and the magnitude of $G(s)H(s)$ is

$$|G(s)H(s)| = \sqrt{(-1)^2 + 0^2} = 1 \qquad \left|\frac{N(s)}{D(s)}\right| = \left|-\frac{1}{K}\right| = \frac{1}{K}$$

Magnitude Condition describes the point (which satisfied the *angle condition*) at which the magnitude of the open loop transfer function is ***one***.

$$\angle\left(\frac{N(s)}{D(s)}\right) = \angle\left(-\frac{1}{K}\right) = 180°$$

Example: Determine system Gain at S = -3 + j3 to Transfer Function of

$$G(S)H(S) = \frac{k}{s(s+6)}.$$

Solution:

$$G(S)H(S)_{s=-3+j3} = \frac{K}{(-3+j3)(3+j3)}$$
$$= \frac{K\angle 360^\circ}{\sqrt{18}\angle 135^\circ * \sqrt{18}\angle 45^\circ}$$
$$= \frac{K\angle 360^\circ}{18\angle 180^\circ}$$

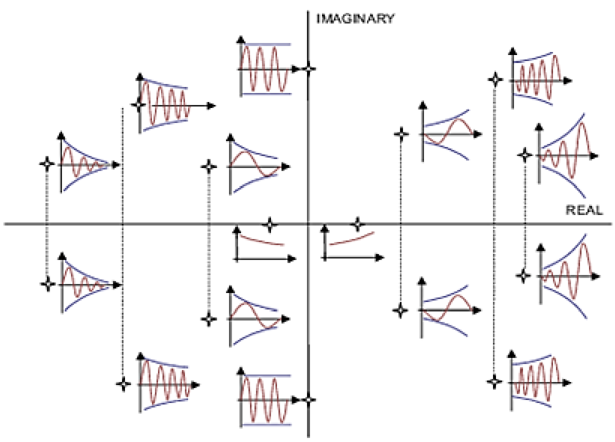
Angle: $\angle G(S)H(S) = \angle 180^\circ = 1 * 180^\circ$

Here 1 is an odd multiple so -3 + j3 lies on Root locus

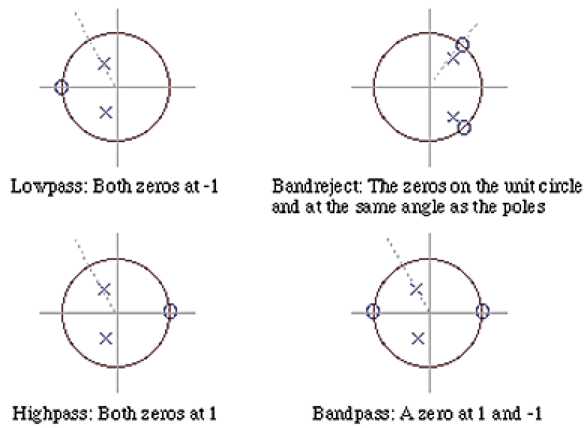
Magnitude: $|G(S)H(S)| = 1 \qquad \frac{K}{18} = 1$

$$\therefore k = 18$$

Analysis of Control Systems using the S-Plane



Time-Response Analysis



Frequency-Response Analysis

MATLAB rlocus (Root Locus Plot of Dynamic Systems)

Description

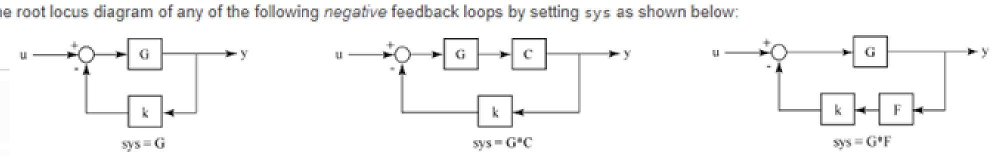
`rlocus(sys)` calculates and plots the root locus of the SISO model `sys`. The root locus returns the closed-loop pole trajectories as a function of the feedback gain `k` (assuming negative feedback). Root loci are used to study the effects of varying feedback gains on closed-loop pole locations. In turn, these locations provide indirect information on the time and frequency responses.

You can use `rlocus` to plot the root locus diagram of any of the following *negative* feedback loops by setting `sys` as shown below:

Syntax

```
rlocus(sys)
rlocus(sys1,sys2,...)
```

```
[r,k] = rlocus(sys)
r = rlocus(sys,k)
```



For instance, if `sys` is a transfer function represented by $sys(s) = \frac{n(s)}{d(s)}$ the closed-loop poles are the roots of $d(s) + kn(s) = 0$

The root locus plot depicts the trajectories of closed-loop poles when the feedback-gain k varies from 0 to infinity. `rlocus` adaptively selects a set of positive gains k to produce a smooth plot. The poles on the root locus plot are denoted by x and the zeros are denoted by o.

`rlocus(sys1,sys2,...)` plots the root loci of multiple LTI models `sys1, sys2,...` on a single plot. You can specify a color, line style, and marker for each model. For even more plot customization options, see `rlocusplot`.

`[r,k] = rlocus(sys)` returns the vector of feedback gains `k` and the complex root locations `r` for these gains.

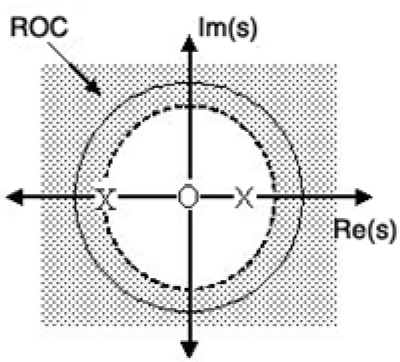
`r = rlocus(sys,k)` uses the user-specified vector of feedback gains k to output the closed-loop poles `r` that define the root locus plot.

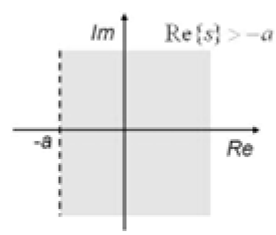
See EXAMPLES

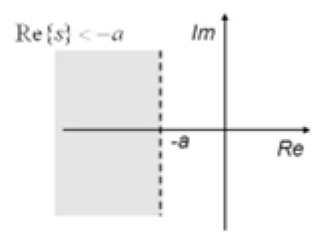
Region of Convergence (ROC) defines the region in the s-plane where the Laplace Transform exists. It is the range of *t* for which the Laplace Transform converges.

Properties of ROC

- ✓ ROC does not contain any poles.
- ✓ ROC contains strip lines parallel to jω axis in s-plane.
- ✓ If x(t) is absolutely integral and it is of finite duration, then ROC is entire s-plane.
- ✓ If x(t) is a right sided sequence then ROC : Re{s} > σ_o.
- ✓ If x(t) is a left sided sequence then ROC : Re{s} < σ_o.
- ✓ If x(t) is a two sided sequence then ROC is the combination of two regions.







f(t)	F(s)	ROC	f(t)	F(s)	ROC
$u(t)$	$\frac{1}{s}$	ROC: Re(s) > 0	$t\,e^{at}\,u(t)$	$\frac{1}{(s-a)^2}$	ROC:Re(s) > a
$t\,u(t)$	$\frac{1}{s^2}$	ROC:Re(s) > 0	$t^n e^{at}\,u(t)$	$\frac{n!}{(s-a)^{n+1}}$	ROC:Re(s) > a
$t^n\,u(t)$	$\frac{n!}{s^{n+1}}$	ROC:Re(s) > 0	$t\,e^{-at}\,u(t)$	$\frac{1}{(s+a)^2}$	ROC:Re(s) > -a
$e^{at}\,u(t)$	$\frac{1}{s-a}$	ROC:Re(s) > a	$t^n\,e^{-at}\,u(t)$	$\frac{n!}{(s+a)^{n+1}}$	ROC:Re(s) > -a
$e^{-at}\,u(t)$	$\frac{1}{s+a}$	ROC:Re(s) > -a	$t\,e^{at}\,u(-t)$	$-\frac{1}{(s-a)^2}$	ROC:Re(s) < a
$e^{at}\,u(t)$	$-\frac{1}{s-a}$	ROC:Re(s) < a	$t^n\,e^{at}\,u(-t)$	$-\frac{n!}{(s-a)^{n+1}}$	ROC:Re(s) < a
$e^{-at}\,u(-t)$	$-\frac{1}{s+a}$	ROC:Re(s) < -a	$t\,e^{-at}\,u(-t)$	$-\frac{1}{(s+a)^2}$	ROC:Re(s) < -a

ASSESSMENT

Group Task

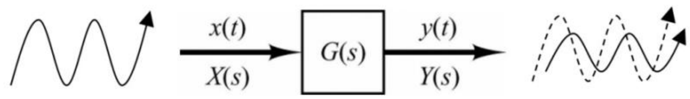
Watch the following YouTube tutorials about MATLAB and perform the tasks they feature:

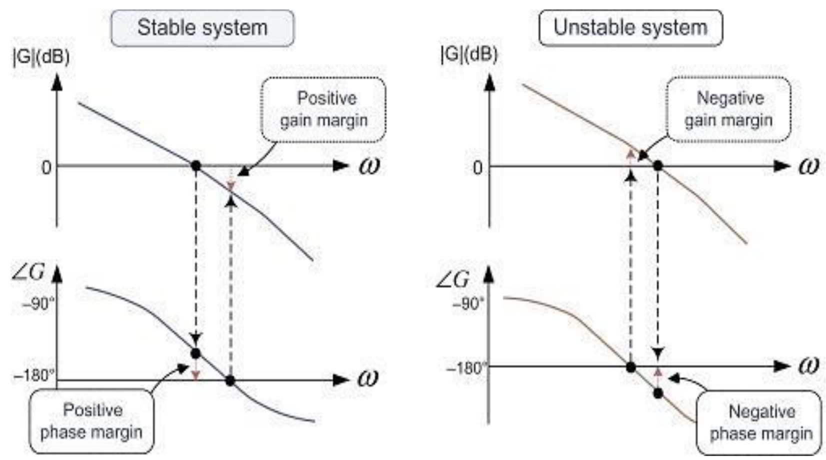
- ✓ Programming the VEX Arm-Cortex using Simulink
- ✓ Robot Navigation using VEX Encoders and Simulink
- ✓ Obstacle Detection for VEX Robots using Simulink
- ✓ Debugging of VEX Simulink Models using External Mode
- ✓ Getting Started with Simulink and VEX EDR V5

See the discussion proper on the Introduction to MATLAB (Module 4),

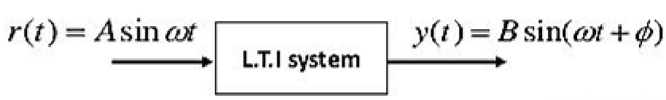
Module 8: **Frequency-Response Analysis**

Frequency Response defines the quantitative measure of the output spectrum of a system that provides useful insights into its stability and performance characteristics. It is the system response to sinusoidal inputs at varying frequencies.

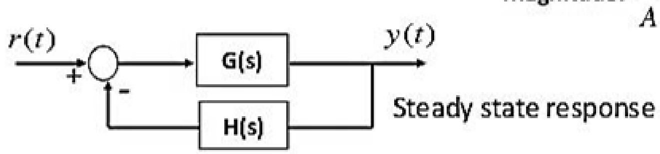




The ***amplitude*** of the output sinusoidal signal is obtained by multiplying the amplitude of the input sinusoidal signal and the magnitude of $G(j\omega)$ at $\omega = \omega_r$, while the ***phase*** of the output sinusoidal signal is obtained by adding the phase of the input signal and the phase of $G(j\omega)$ at $\omega = \omega_r$. We can write ***angular frequency*** ω_r as $\boldsymbol{\omega_r = 2\pi f_0}$; f_0 is the input signal frequency.



Magnitude: $\frac{B}{A}$ **Phase:** ϕ

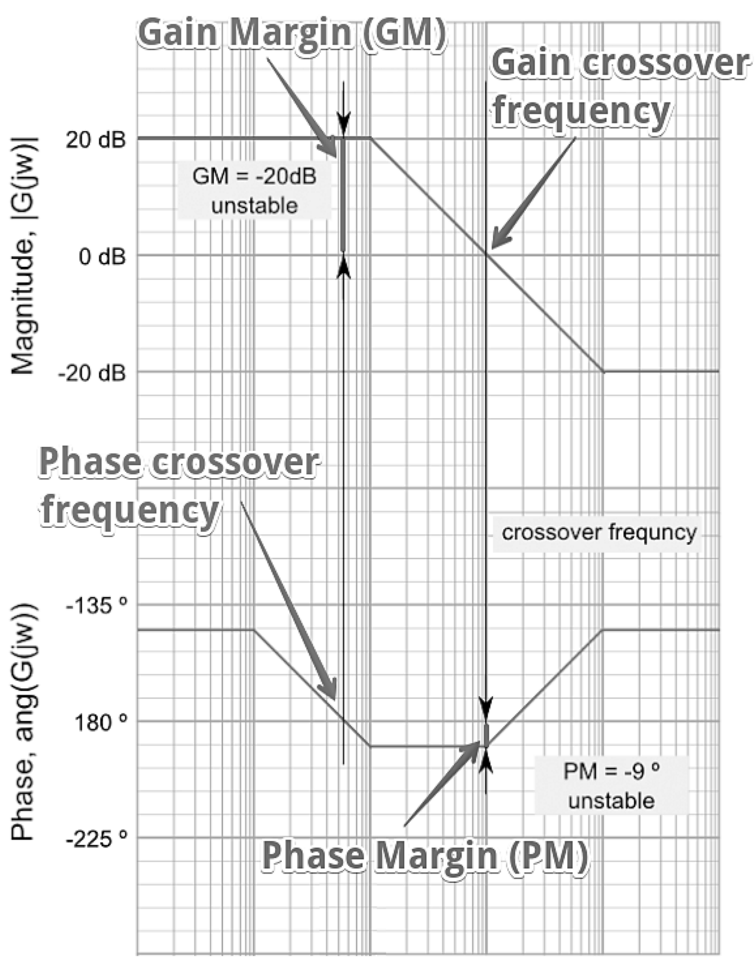


$$\frac{Y(s)}{R(s)} = \frac{G(s)}{1+G(s)H(s)} \qquad s = \sigma + j\omega \Rightarrow s = j\omega$$

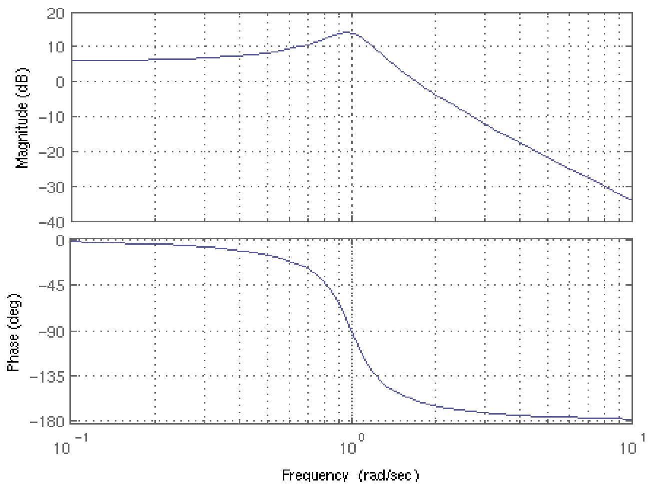
Magnitude: $\dfrac{|G(j\omega)|}{|1+G(j\omega)H(j\omega)|}$ **Phase:** $\dfrac{\angle G(j\omega)}{\angle[1+G(j\omega)H(j\omega)]}$

Frequency Domain Specifications

- Resonant Frequency – where the peak magnitude of the frequency response falls, denoted by ω_r.
- Resonant Peak – maximum magnitude of $T(j\omega)$, denoted by M_r; corresponds to the peak overshoot in the time domain response for certain values of damping ratio δ
- Bandwidth –frequencies range over which, the magnitude of $T(j\omega)$ drops to 70.7% from its 0-frequency value
- Cut-off Rate – slope of the log-magnitude curve near the cutoff frequency
- Gain Margin – gain to be added to the system in order to bring the system to the verge of instability
- Phase Margin – phase lag to be added at the gain cross over frequency to bring the system to the verge of instability



Bode Plot is a standard graph format for plotting frequency response of LTI systems. It is usually a combination of a *Bode Magnitude Plot*, expressing the magnitude of the frequency response, and a *Bode Phase Plot*, expressing the phase shift. It is used in control system engineering to determine and analyze the system stability.



The values of these Bode Plot parameters can be used to analyze system stability:

1. <u>Phase Cross over Frequency</u> (ω_{PC}) – frequency (rad/sec) at which the phase plot has $-\mathbf{180}°$ phase
2. <u>Gain Cross over Frequency</u> (ω_{GC}) – frequency (rad/sec) at which the magnitude plot has a $\mathbf{0}\ \boldsymbol{dB}$ magnitude
   - Stable if $\omega_{PC} > \omega_{GC}$
   - Marginally Stable if $\omega_{PC} = \omega_{GC}$
   - Unstable if $\omega_{PC} < \omega_{GC}$
3. <u>Gain Margin</u> (GM) – negative of the magnitude (dB) at phase cross over frequency

$$GM = 20\log\left(\frac{1}{M_{pc}}\right) = 20logM_{pc}$$

M_{PC} = magnitude at phase cross over frequency

4. <u>Phase Margin</u> (PM) – amount of change in open-loop phase (degree) needed to make a closed-loop system unstable
   - Stable if both GM and PM are $(+)$
   - Marginally Stable if both GM and PM are 0
   - Unstable if both/either GM and/or PM are/is $(-)$

$$PM = 180^0 + \phi_g$$

$\emptyset_G$ = phase angle at gain cross over frequency

MATLAB bode (Bode Plot of Frequency Response, or Magnitude and Phase Data)

Syntax

```
bode(sys)
bode(sys1,sys2,...,sysN)
bode(sys1,LineSpec1,
     ...,sysN,LineSpecN)
bode( __ ,w)

[mag,phase,wout] = bode(sys)
[mag,phase,wout] = bode(sys,w)
[mag,phase,wout,sdmag,sdphase]
                 = bode(sys,w)
```

Description

`bode(sys)` creates a Bode plot of the frequency response of a dynamic system model `sys`. The plot displays the magnitude (in dB) and phase (in degrees) of the system response as a function of frequency. `bode` automatically determines frequencies to plot based on system dynamics.

If `sys` is a multi-input, multi-output (MIMO) model, then `bode` produces an array of Bode plots, each plot showing the frequency response of one I/O pair.

`bode(sys1,sys2,...,sysN)` plots the frequency response of multiple dynamic systems on the same plot. All systems must have the same number of inputs and outputs.

`bode(sys1,LineSpec1,...,sysN,LineSpecN)` specifies a color, line style, and marker for each system in the plot.

`bode( __ ,w)` plots system responses for frequencies specified by `w`.

- If `w` is a cell array of the form `{wmin,wmax}`, then `bode` plots the response at frequencies ranging between `wmin` and `wmax`.
- If `w` is a vector of frequencies, then `bode` plots the response at each specified frequency.

You can use `w` with any of the input-argument combinations in previous syntaxes.

`[mag,phase,wout] = bode(sys)` returns the magnitude and phase of the response at each frequency in the vector `wout`. The function automatically determines frequencies in `wout` based on system dynamics. This syntax does not draw a plot.

`[mag,phase,wout] = bode(sys,w)` returns the response data at the frequencies specified by `w`.

- If `w` is a cell array of the form `{wmin,wmax}`, then `wout` contains frequencies ranging between `wmin` and `wmax`.
- If `w` is a vector of frequencies, then `wout = w`.

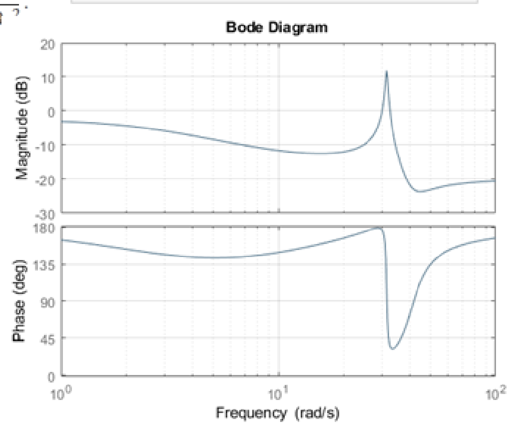
`[mag,phase,wout,sdmag,sdphase] = bode(sys,w)` also returns the estimated standard deviation of the magnitude and phase values for the identified model `sys`. If you omit `w`, then the function automatically determines frequencies in `wout` based on system dynamics.

See EXAMPLES

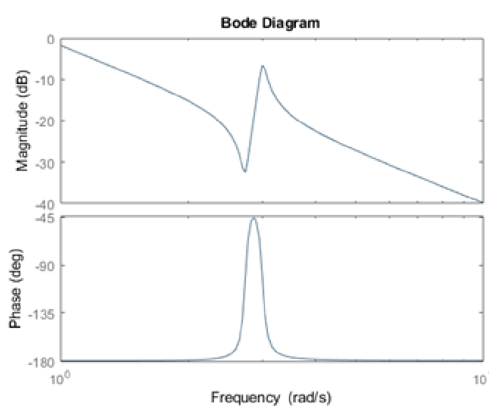
Example

$$H(s) = \frac{s^2 + 0.1s + 7.5}{s^4 + 0.12s^3 + 9s^2}.$$

```
H = tf([-0.1,-2.4,-181,-1950],[1,3.3,990,2600]);
bode(H,{1,100})
grid on
```



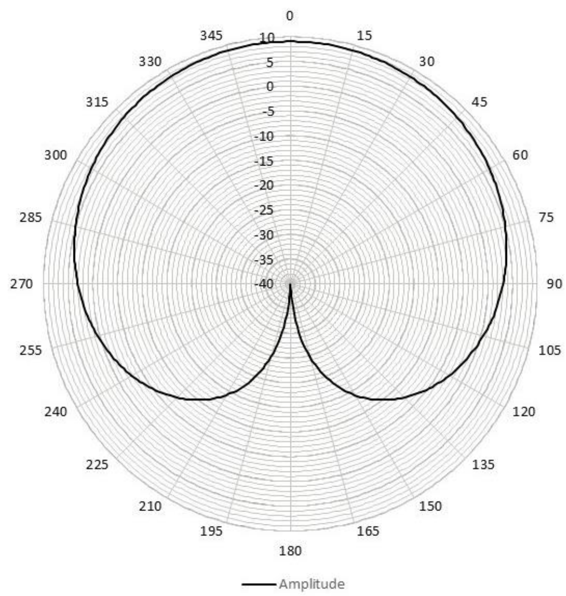

```
H = tf([1 0.1 7.5],[1 0.12 9 0 0]);
bode(H)
```



bode automatically selects the plot range based on the system dynamics.

Polar Plot is a type of plot drawn between magnitude and phase where the magnitudes are represented by normal values. It is drawn between the magnitude and the phase angle of $G(j\omega)H(j\omega)$ by varying ω from 0 to ∞. The concentric circles and the radial lines represent the magnitudes and phase angles, respectively. These angles are represented by positive values in anti-clock wise direction.

$$G(j\omega)H(j\omega) = |G(j\omega)H(j\omega)| \angle G(j\omega)H(j\omega)$$



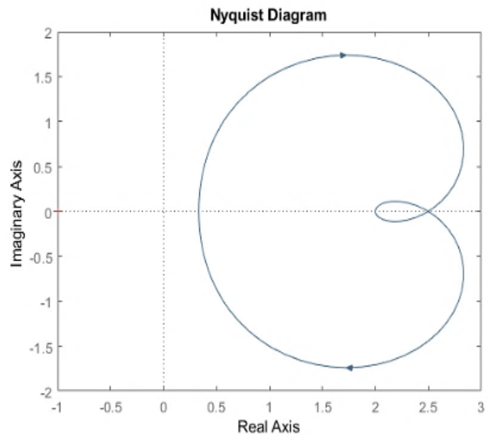
MATLAB polarplot (Polar Plot)

Syntax	Description
polarplot(theta,rho) polarplot(theta,rho,LineSpec) polarplot(theta1,rho1,...,thetaN,rhoN) polarplot(theta1,rho1,LineSpec1, ...,thetaN,rhoN,LineSpecN)	polarplot(theta,rho) plots a line in polar coordinates, with theta indicating the angle in radians and rho indicating the radius value for each point. The inputs must be vectors with equal length or matrices with equal size. If the inputs are matrices, then polarplot plots columns of rho versus columns of theta. Alternatively, one of the inputs can be a vector and the other a matrix as long as the vector is the same length as one dimension of the matrix.
	polarplot(theta,rho,LineSpec) sets the line style, marker symbol, and color for the line.
	polarplot(theta1,rho1,...,thetaN,rhoN) plots multiple rho,theta pairs.
	polarplot(theta1,rho1,LineSpec1,...,thetaN,rhoN,LineSpecN) specifies the line style, marker symbol, and color for each line.
polarplot(rho) polarplot(rho,LineSpec)	polarplot(rho) plots the radius values in rho at evenly spaced angles between 0 and 2π.
	polarplot(rho,LineSpec) sets the line style, marker symbol, and color for the line.
polarplot(Z) polarplot(Z,LineSpec)	polarplot(Z) plots the complex values in Z.
	polarplot(Z,LineSpec) sets the line style, marker symbol, and color for the line.
polarplot(__,Name,Value) polarplot(pax,__) p = polarplot(__)	polarplot(__,Name,Value) specifies properties of the chart line using one or more Name,Value pair arguments. The property settings apply to all the lines. You cannot specify different property values for different lines using Name,Value pairs.
	polarplot(pax,__) uses the PolarAxes object specified by pax, instead of the current axes.
	p = polarplot(__) returns one or more chart line objects. Use p to set properties of a specific chart line object after it is created. For a list of properties, see Line Properties.

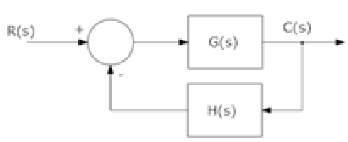
See EXAMPLES

Nyquist Plot is the continuation of polar plot for finding the stability of the closed loop control systems by varying ω from $-\infty$ to ∞. It is used to draw the complete frequency response of the open loop TF. The ***Nyquist stability criterion*** states that if there are P poles and Z zeros are enclosed by the s-plane closed path, the corresponding $G(s)H(s)$ plane must encircle the origin $P - Z$ times. Mathematically,

$$N = P - Z$$

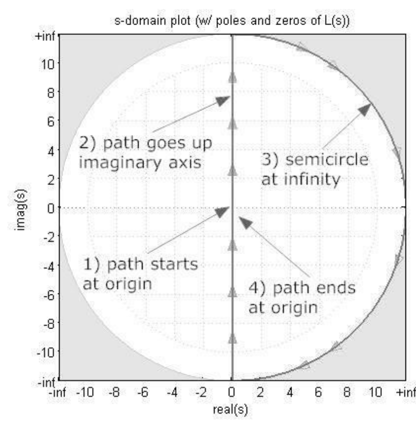


Determining Stability using Nyquist Plot



$$T(s) = \frac{C(s)}{R(s)} = \frac{G(s)}{1 + G(s) \cdot H(s)} = \frac{G(s)}{1 + L(s)}$$

$$L(s) = \text{loop gain} = G(s) \cdot H(s)$$



To determine the stability of a system:

- Start with a system that has $1 + L(s) = 0$ characteristic equation.
- Make a mapping from the s domain to the $L(s)$ domain where the path of s encloses the entire right half plane.
- From the mapping we find the number N, which is the number of encirclements of the $-1 + j0$ point in $L(s)$.
- Factor $L(s)$ to determine P in the right half plane.
- With N and P, we can determine Z of $1 + L(s)$ in the right half plane (which is the same as the number of poles of $T(s)$).
- If $Z > 0$, the system is unstable.

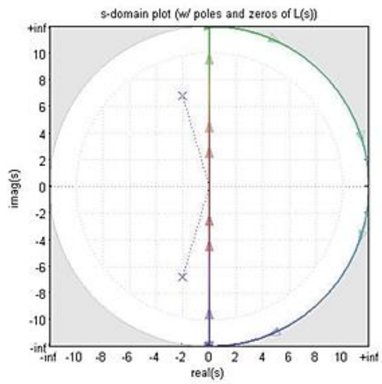
Example: Nyquist Path, no poles on $j\omega$ axis, stable

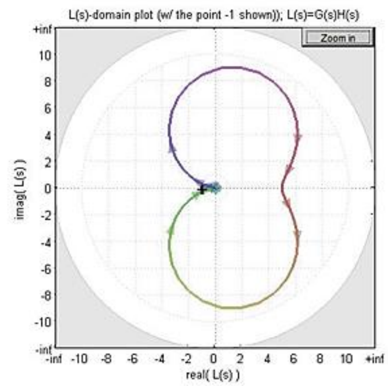
- Consider a system with plant $G(s)$, and unity gain feedback $(\mathrm{H}(s) = 1)$

$$G(s) = \frac{250}{s^2 + 4s + 50}; \quad H(s) = 1$$

$$L(s) = G(s) \cdot H(s) = \frac{250}{s^2 + 4s + 50}$$

If we map this function from s to $L(s)$ with the variable s following the Nyquist path we get the following image *(note: the image on the left is the "Nyquist path" the image on the right is called the "Nyquist plot")*





If we zoom in on the graph in $L(s)$, the first thing we notice is the multiple arrowheads at the origin. This is because as the path in s traverses a semicircle at ∞ the path in $L(s)$ remains at the origin, but the angle of $L(s)$ changes.

More importantly, we can see that it does not encircle the $-1 + j0$, so $N = 0$. We also know that P=0, and since $N = Z - P$; $Z = 0$ also. This tells us that the system is stable. And, if we close the loop, we find that the characteristic equation of the closed loop transfer function is

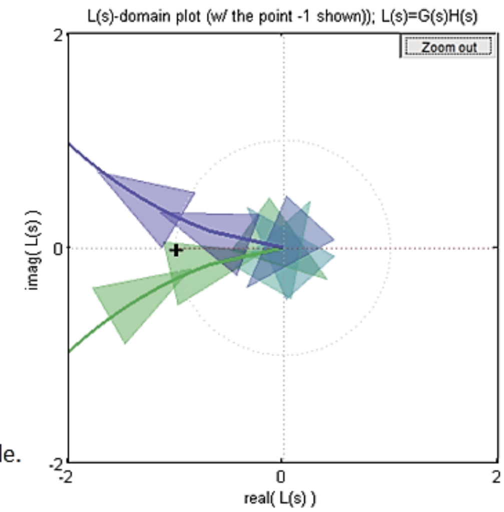
$$1 + L(s) = 0$$

$$1 + \frac{250}{s^2 + 4s + 50} = 0$$

$$s^2 + 4s + 50 + 250 = 0$$

$$s^2 + 4s + 300 = 0$$

which has roots at $-2 \pm 17.2j$ so the system is indeed stable.



MATLAB nyquist (Nyquist Plot of Frequency Response)

Syntax

```
nyquist(sys)
nyquist(sys,w)
nyquist(sys1,sys2,...,sysN)
nyquist(sys1,sys2,...,sysN,w)
nyquist(sys1,'PlotStyle1',...,sysN,'PlotStyleN')
[re,im,w] = nyquist(sys)
[re,im] = nyquist(sys,w)
[re,im,w,sdre,sdim] = nyquist(sys)
```

Description

`nyquist` creates a Nyquist plot of the frequency response of a dynamic system model. When invoked without left-hand arguments, `nyquist` produces a Nyquist plot on the screen. Nyquist plots are used to analyze system properties including gain margin, phase margin, and stability.

`nyquist(sys)` creates a Nyquist plot of a dynamic system `sys`. This model can be continuous or discrete, and SISO or MIMO. In the MIMO case, `nyquist` produces an array of Nyquist plots, each plot showing the response of one particular I/O channel. The frequency points are chosen automatically based on the system poles and zeros.

`nyquist(sys,w)` explicitly specifies the frequency range or frequency points to be used for the plot. To focus on a particular frequency interval, set `w = {wmin,wmax}`. To use particular frequency points, set `w` to the vector of desired frequencies. Use `logspace` to generate logarithmically spaced frequency vectors. Frequencies must be in `rad/TimeUnit`, where `TimeUnit` is the time units of the input dynamic system, specified in the `TimeUnit` property of `sys`.

`nyquist(sys1,sys2,...,sysN)` or `nyquist(sys1,sys2,...,sysN,w)` superimposes the Nyquist plots of several LTI models on a single figure. All systems must have the same number of inputs and outputs, but may otherwise be a mix of continuous- and discrete-time systems. You can also specify a distinctive color, linestyle, and/or marker for each system plot with the syntax `nyquist(sys1,'PlotStyle1',...,sysN,'PlotStyleN')`.

`[re,im,w] = nyquist(sys)` and `[re,im] = nyquist(sys,w)` return the real and imaginary parts of the frequency response at the frequencies `w` (in `rad/TimeUnit`). `re` and `im` are 3-D arrays (see "Arguments" below for details).

`[re,im,w,sdre,sdim] = nyquist(sys)` also returns the standard deviations of `re` and `im` for the identified system `sys`.

See EXAMPLES

ASSESSMENT

Group Task

Watch the following YouTube tutorials about MATLAB and perform the tasks they feature:

- ✓ Programming VEX Smart Motors using Simulink
- ✓ Simulating Mobile Robotics using Virtual Worlds
- ✓ Switching between Driver and Autonomous Control of VEX Robots
- ✓ Using Infrared Sensors for Robot Autonomy
- ✓ Using Vision Sensors for Robot Autonomy

See the discussion proper on the Introduction to MATLAB (Module 4),

Control Systems for Building Management

(CMPE 30133 – Feedback and Control Systems)

3 Chapter

I. OVERVIEW

Building management system (BMS) is typically a personal or embedded computer, operating dedicated BMS software, and communicating through an industrial control network to the HVAC equipment. It is an overarching control system that is responsible for the automatic regulation and control of non-GMP facility subsystems, maintaining predefined parameters (or set points) and the control of their functionality. Its major aim of the BMS is to guarantee the safety of facility operation, while also monitoring and optimizing the use and efficiency of its supervised subsystems to allow more efficient operation.

II. MODULE OBJECTIVES

After successful completion of modules 9-12 of Chapter 3, you should be able to:

1) Remember the key concepts of building management systems;
2) Understand the purpose and functions of fire detection and alarm systems;
3) Evaluate the processes in designing CCTV and PA systems; and
4) Apply the working principles of mixed-signal circuits and sensors in developing access control systems for building management.

III. COURSE MATERIALS

Suggested Online Resources for Further Learning

- Downloadable Course AVPs: https://bit.ly/3iLs1dM
- Downloadable Course PDFs: https://bit.ly/3mxuSs5

Module 9: **Essential Components of a BMS**

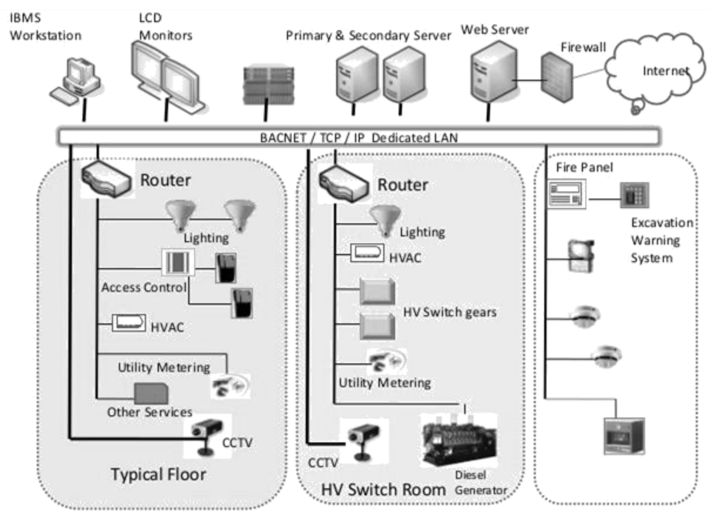
Building Management System (BMS) is a computer-based control system used in buildings to control and monitor the building's mechanical and electrical equipment such as ventilation, lighting, power systems, fire systems, and security systems. It is otherwise known as a *Building Management System* (BMS) or *Smart Building*.

Core Functions of a BMS:

1. To control the building environment
2. To operate systems according to occupancy and energy demand
3. To monitor and correct system performance
4. To alert or sound alarms when needed

Essential Components of a BMS:

1. Sensors
2. Controllers
3. Output Devices
4. Communication Protocols
5. User Interface (UI)



3 Main Categories of BMS Subsystems

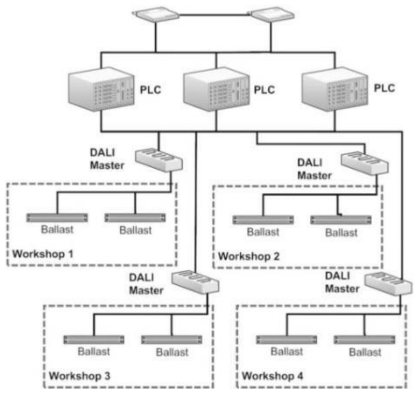
1. <u>Electrical Systems</u> – network of conductors and equipment designed to carry, distribute, and convert electrical power safely from the point of delivery or generation to the various loads around the building that consume the electrical energy; includes lighting control systems and load management systems
2. <u>Mechanical Systems</u> – any essential building services that use machines; include plumbing systems such as water supply and sanitation systems, electromechanical systems such as elevators and escalators, and HVAC systems
3. <u>Extra-Low Voltage (ELV) Systems</u> – subsystems that need electricity to run but are not part of the building's main electrical system; include data network, CCTV, public address systems, audio/video solutions and access control systems

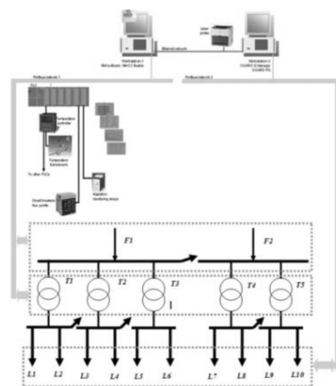
Lighting Control System

- intelligent networked system of devices related to lighting control
- devices may include relays, occupancy sensors, photocells, light control switches or touchscreens, and signals from other building systems (such as fire alarm or HVAC)

Load Management System

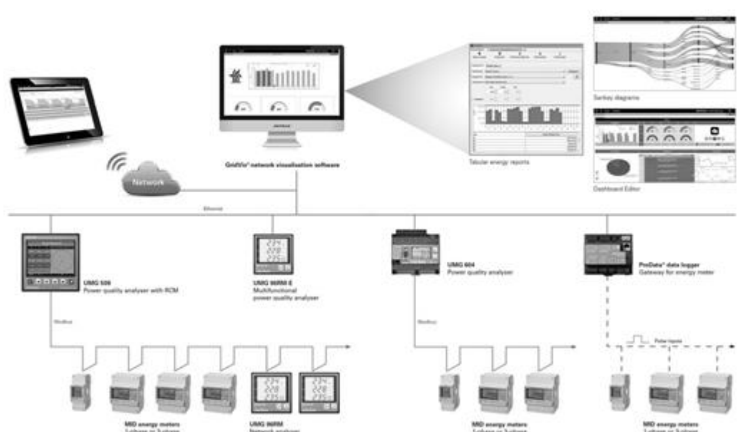
- primarily intended for balancing the supply of electricity on the network with the electrical load by adjusting or controlling the load rather than the power station output
- can also help reduce harmful emissions, since peaking plants or backup generators are often dirtier and less efficient than base load power plants





Energy Management System (EMS)

- computer-based system that measures energy consumption and look for areas where there might be room to improve energy efficiency
- requires a careful balancing between efforts to use energy efficiently and meet the quality of life requirements, while insuring that primary mission requirements are met

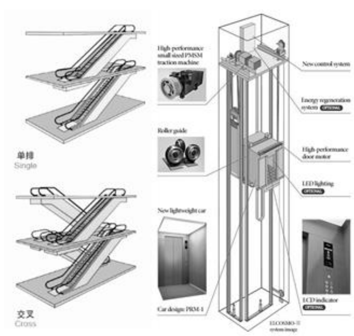
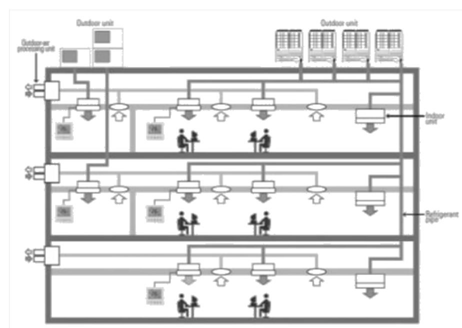


Electromechanical System

- primarily involves electric motors and solenoids that are used in combination with mechanical parts to provide actuation or movement
- includes elevator and escalator systems, as well as industrial and high-powered motor-based machines installed and operated within the building

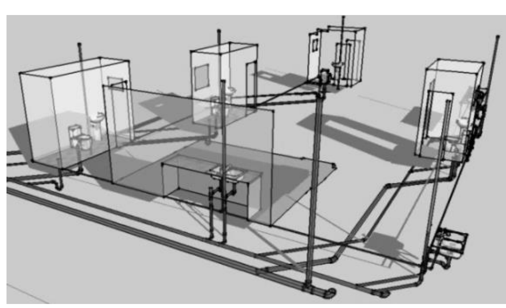
Heating, Ventilation, and Air Conditioning (HVAC) **System**

- used for moving air between indoor and outdoor areas, along with heating and cooling building interiors
- includes a natural or forced ventilation element used to filter and clean indoor air to keep a healthy building indoor environment and maintain humidity levels at optimal comfort levels
- uses heating elements such as furnace or boiler, as well as a pipe system for the fluid carrying the heat or duct work for forced air system

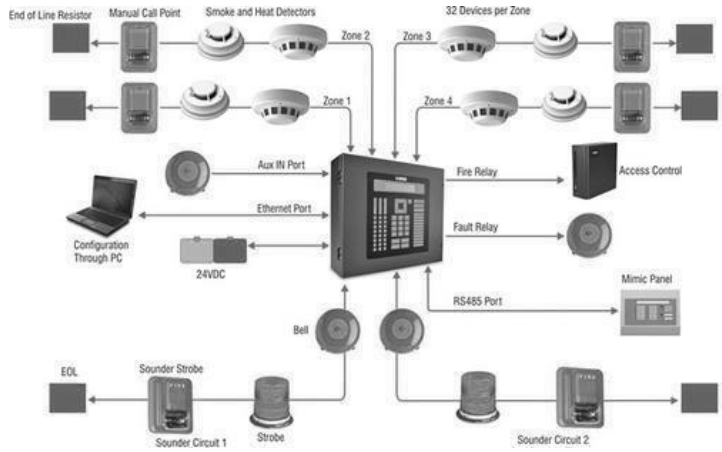
Plumbing Systems

- network of pipes and fixtures installed in a building for the distribution and use of potable (drinkable) water and the removal of waterborne wastes
- system that conveys fluids for a wide range of applications using pipes, valves, plumbing fixtures, tanks, and other apparatuses to convey fluids



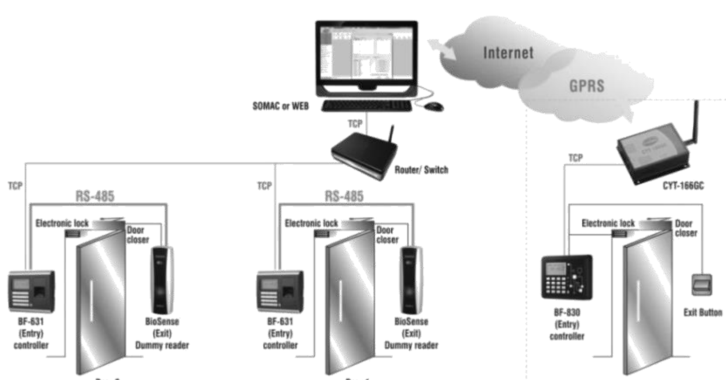
Fire Detection and Alarm System (FDAS)

- control equipment that utilizes detectors, warning devices and other components to detect fires and provide warning
- detection elements include heat detectors, flame detectors, smoke detectors, carbon monoxide detectors and multi- sensor detectors, or an alarm can be triggered at manual call points
- alarms may consist of bells, sirens, horns, lights or a combination these



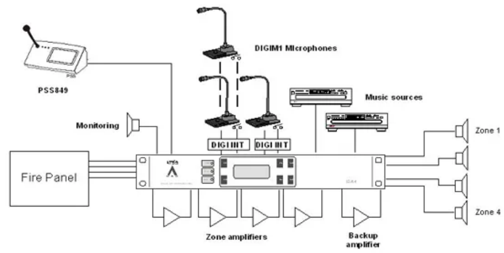
Access Control System (ACS)

- determines who is allowed to enter or exit, where they are allowed to exit or enter, and when they are allowed to enter or exit
- security features that control how users and systems communicate and interact with other systems and resources
- may also provide logistics and location monitoring functions, which are important in buildings that require utmost security features



Public Address with Background Music (PA BGM) **System**

- may include multiple microphones or other sound sources, a mixing console, multiple amplifiers, and loudspeakers for louder volume or wider distribution
- used in any public venue that requires that an announcer, performer, etc. be sufficiently audible at a distance or over a large area; basically intended to provide “public address”, or a way to transmit audio communication to a group

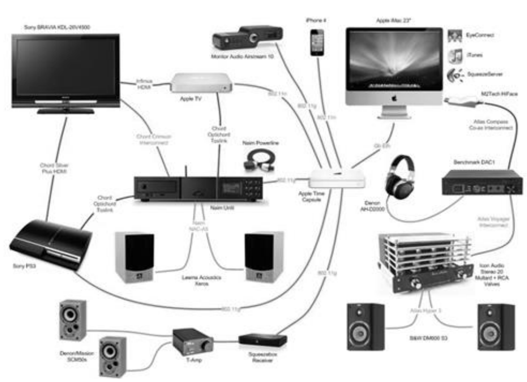


Closed-Circuit Television (CCTV) **System**

- allows the use of video cameras to monitor the interior and exterior of a property, transmitting the signal to a monitor or set of monitors and can utilize digital video recorders (DVRs) to provide footage recording with a variety of options and extra features (such as motion detection and email alerts
- commonly used for security and surveillance, as well as for observing parts of a process from a central control room (e.g. when the environment is not suitable for humans)

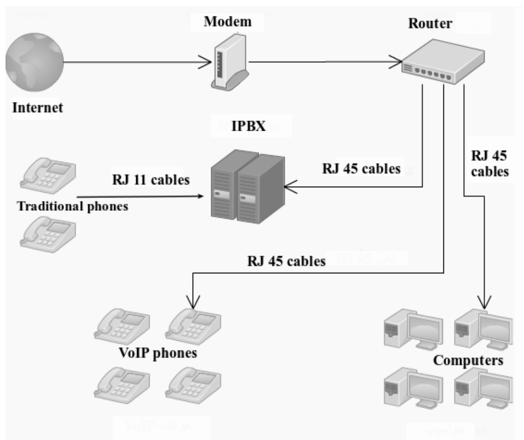
Audiovisual (AV) **Control Systems**

- can be comprised of conference telephones, video cameras, interactive whiteboards, digital signage, computers, smartphones, tablets, wireless connectivity, and more
- designed to simplify audiovisual technologies so the events and collaborations that depend on them run seamlessly



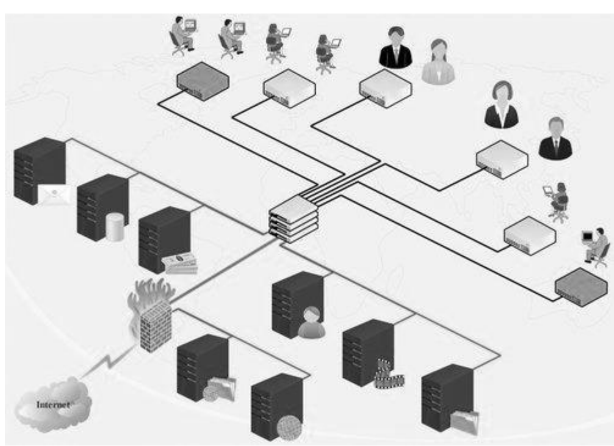
Internet Protocol Private Branch Exchange (IP PBX or IPBX) **Systems**

- central system that switches and routes calls between the telephone network and Voice over IP (VoIP) users
- assesses the best way to route many calls simultaneously to efficiency allow business users to share limited resources, like a set number of external phone lines
- PBX system with IP connectivity and may provide additional audio, video, or instant messaging communication utilizing the TCP/IP protocol stack



Network Infrastructure

- hardware and software resources of an entire network that enable network connectivity, communication, operations and management of an enterprise network
- provides the communication path and services between users, processes, applications, services and external networks or the internet
- services include: routers, switches, load balancing, secure remote access, servers, back-office software, storage, and IP telephony



ASSESSMENT

Group Task

Watch the following YouTube tutorials about MATLAB and perform the tasks they feature:

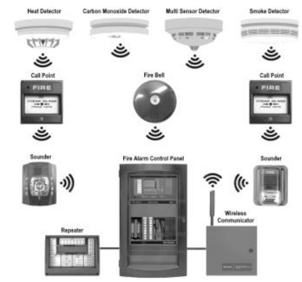
- ✓ What is Systems Engineering
- ✓ Towards a Model-Based Approach
- ✓ The Benefits of Functional Architectures
- ✓ An Introduction to Requirements
- ✓ Some Benefits of Model-Based Systems Engineering

See the discussion proper on the Introduction to MATLAB (Module 4),

Module 10: **Fire Detection and Alarm Systems**

Fire Detection and Alarm System (FDAS) is a early warning device intended to protect the property and to help people evacuate safely in case of fire. It has the following components:

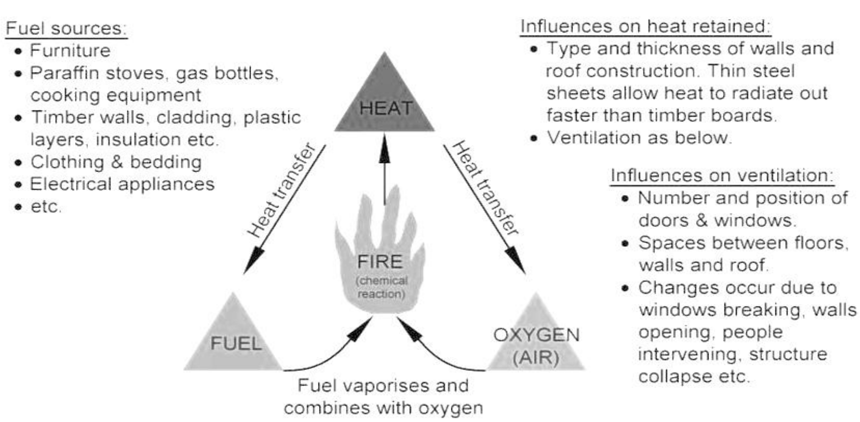
- ✓ inputs: detectors/sensors (smoke, heat, flame, etc.) and manual switches
- ✓ outputs: sound and light signals, voice evacuation procedures, interface to emergency power shutdown
- ✓ control panel: configurable device that integrates inputs and outputs, and allows users control them

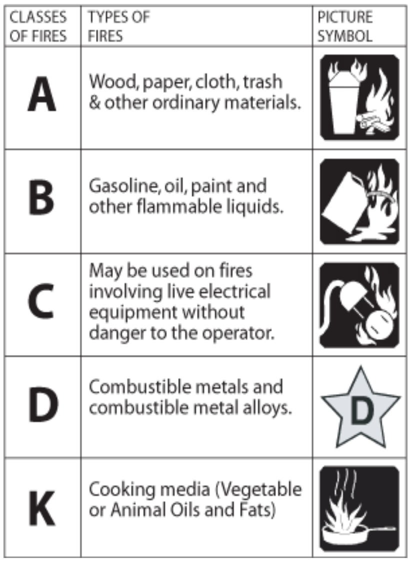


Philippine Electronics Code Vol. 2 (Fire Detection and Alarm Systems)

- discusses the design and application, invitation, maintenance, testing, and inspection of fire detection and alarm system (fire suppression systems are not covered)
- defines the means of signal initiation, transmission, notification, annunciation, and emergency voice evacuation in a fire detection and alarm system
- establishes the minimum required levels of performance and quality of installation of fire detection and alarm systems
- requires the installment of standards-compliant FDAS in buildings, facilities, houses, structures, and premises that are not more than 400 sq. m. in total floor area, nor more than two floors, and with a height of not more than 10 m. (in relation to the relevant provision of RA 9514)

Fire Hazards

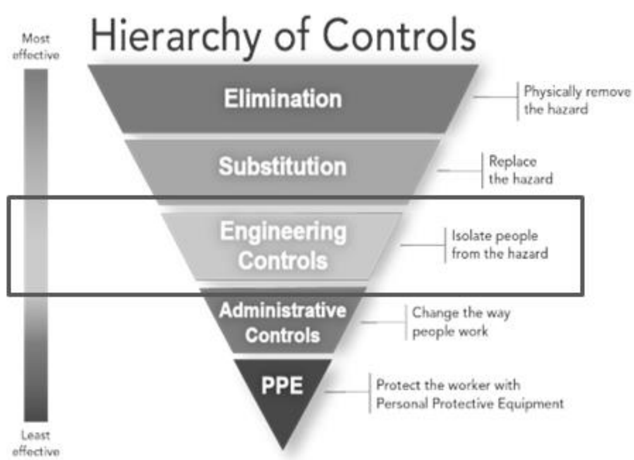




CLASSES OF FIRES	TYPES OF FIRES	PICTURE SYMBOL
A	Wood, paper, cloth, trash & other ordinary materials.	
B	Gasoline, oil, paint and other flammable liquids.	
C	May be used on fires involving live electrical equipment without danger to the operator.	
D	Combustible metals and combustible metal alloys.	D
K	Cooking media (Vegetable or Animal Oils and Fats)	

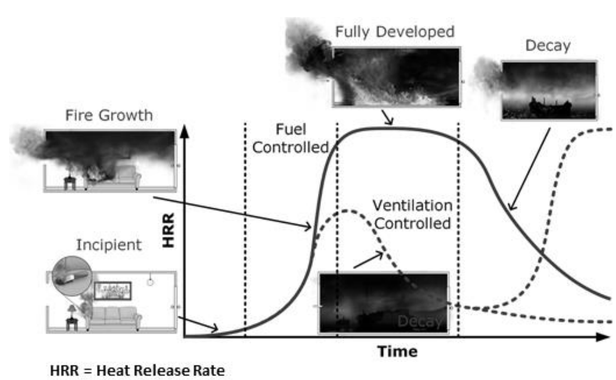
Factors that Determine the Degree of Fire Hazard

- Construction Materials
- Combustibility of Contents
- Type of Occupancy
- Number of Occupants (Load Capacity)
- Height from the Ground Level
- Longest Horizontal Travel Distance
- Fire Safety Construction



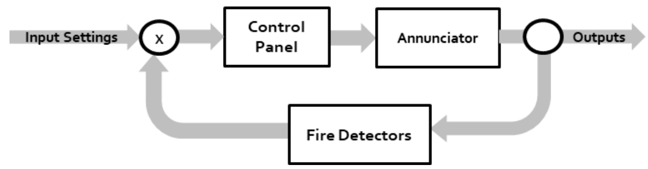
Stages of Fire Development

1. <u>Incipient</u>: Ignition has occurred but there has been no spread.
2. <u>Growth</u>: Fire is self-sustaining.
3. <u>Fully Developed</u>: Fire is at its hottest point, burning all its available fuels.
4. <u>Decay</u>: Fire is running out of fuel but is still very dangerous. This is the longest stage



FDAS shall have at least one of these functions

- ✓ Automatic detection, and manual fire alarm signal initiation
- ✓ Activation of fire alarm notification appliances
- ✓ Emergency communications system
- ✓ Activation of annunciators
- ✓ Monitoring of abnormal conditions in fire suppression system
- ✓ Activation of fire safety functions
- ✓ Transmission of alarm signal to an off-premises central station



Fire Alarm Control Panel (FACP)

- receives signals from input devices of the FDAS, and controls the outputs
- required in each FDAS; may be multiple in a multi-building complex with at least 3 buildings or great than 40,000 sq. m. total floor area of all the buildings
- has 3 types:
    - ✓ **Conventional –** for small offices or retail shops
    - ✓ **Addressable** – for large building complexes
    - ✓ **Semi-Addressable** – used with a series of detectors and devices connected back to the central control panel

Fire detector is a device/component used to sense fire, smoke etc. *Automatic fire detectors* (AFDs) are meant to imitate one or more of the human senses of touch, smell or sight. In FDAS, one or more of the following detectors are used:

- **Smoke Detector –** particles of combustion
- **Heat Detector** – abnormally high temperature
- **Multi-Criteria Detector –** smoke, heat, flame, and CO
- **Duct Detector –** smoke circulating in ventilating ducts
- **Beam Detector** – smoke along a continuous path
- **Flame Detector –** radiant energy emitted by flame
- **Air Sampling-Type Detector** – consist of a piping lay-out
- **Carbon Monoxide Detector –** abnormally high levels of CO

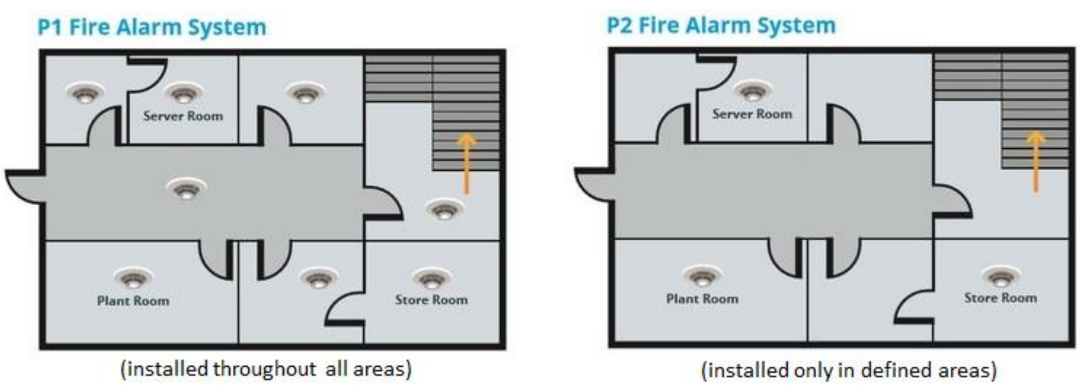
Areas where fire detectors are required:

- Spaces under floor more than 600 mm. in height
- Spaces above drop ceiling more than 1 m. in height
- Concealed spaces under the roof more than 1 m. in height
- Stairwells, vertical shafts, and vaults
- Return air ducts of air conditioning systems
- Tunnels linking two or more buildings

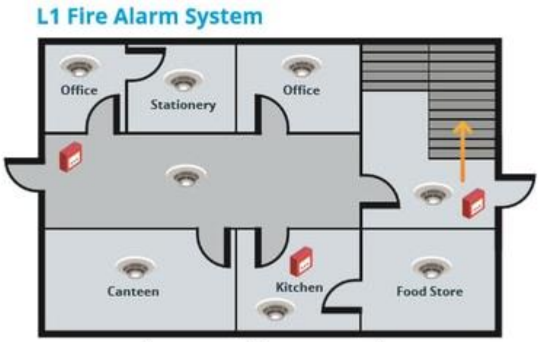
Areas where fire detectors are NOT required:

- Spaces that don't meet the above specifications
- Toilet and/or bathroom less than 4 sq. m. in floor area
- Exhaust ducts from toilets and bathrooms
- Any walk-in type enclosure with less than 1 sq. m floor area
- Covered paths and areas under structures (platforms, ducts)

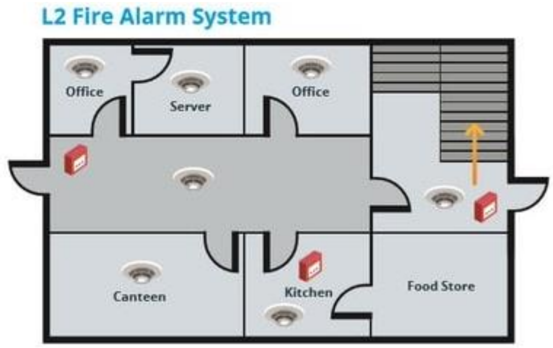
P category fire alarm systems are those that are designed to protect property. It is appropriate to label a system under this category when considering how a business and its operations will be protected from the risk of fire.



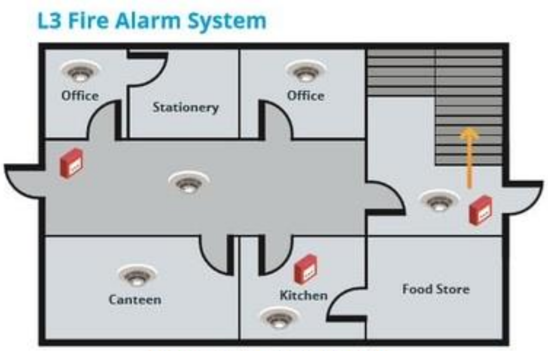
L category fire alarm systems are those deemed most suitable for protecting life within a building. L1 is the most comprehensive system, while L2 features detectors in high-risk rooms and those that form part of an escape route, including corridors. L3 features detectors in all escape routes (e.g. corridors and stairways) and rooms that open onto an escape route, while L4 consists of detectors within escape route areas only. These categories of systems are recommended to lower-risk buildings.



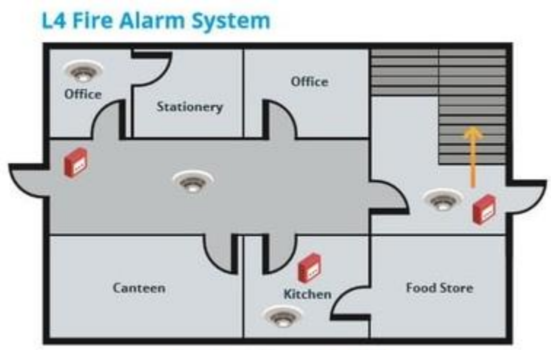
(maximum life protection)



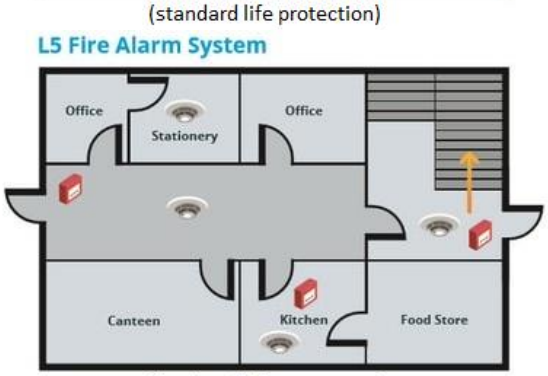
(additional life protection)



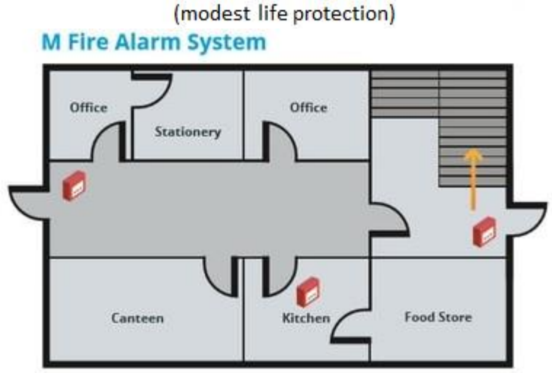
(standard life protection)



(modest life protection)



(localized life protection)



(manual system)

L5 is one installed to tackle a specific fire risk in a certain area of a building, while M types is the least sophisticated of systems, which rely on the building's occupants to detect a fire and provide a warning to others.

Alarms shall be clearly audible throughout the floor and/or building where they are installed with a typical sound level ranging from 65 dbA (10 dB higher than ambient room noise) to 115 dBA. ***Annunciators*** or visual notification devices such as indicating lamps, strobe lights, or text displays shall be used along with audible notification for areas where hearing protection is worn. The primary display of fire point/zone annunciation shall be at the FACP of the fire command center.

Central Station Alarm Monitoring

- where all installed FDAS should be subscribed to
- all alarm signals received are forwarded or redirected to firefighting authorities

A 1-way emergency communication subsystem shall be required for the following facilities:

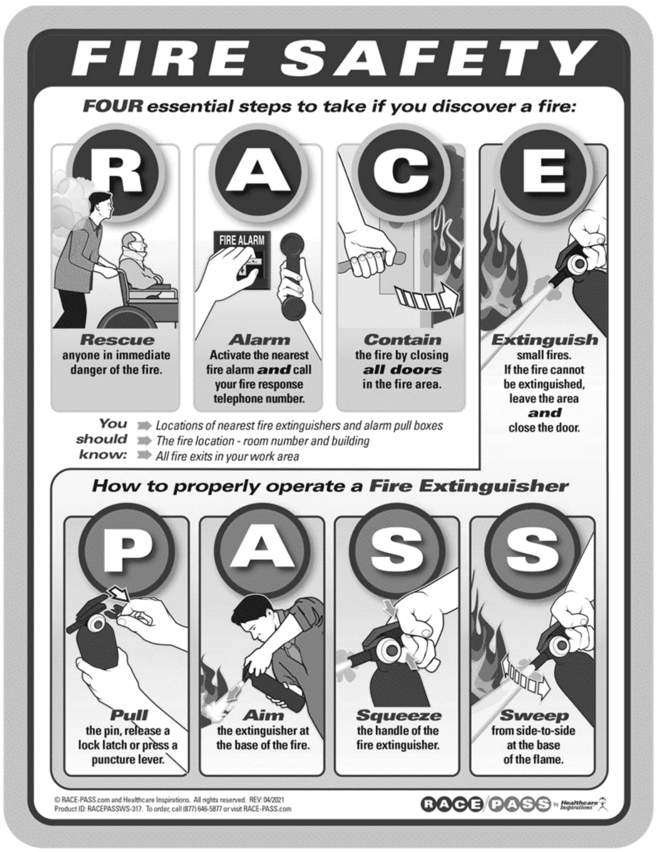
- Transportation Terminals
- Shopping Malls and Hotels
- Office Buildings
- Buildings with floor area greater than 15,000 sq. m. or with more than 15 floors (25,000 sq. m. floor area or more than 25 floors if residential)

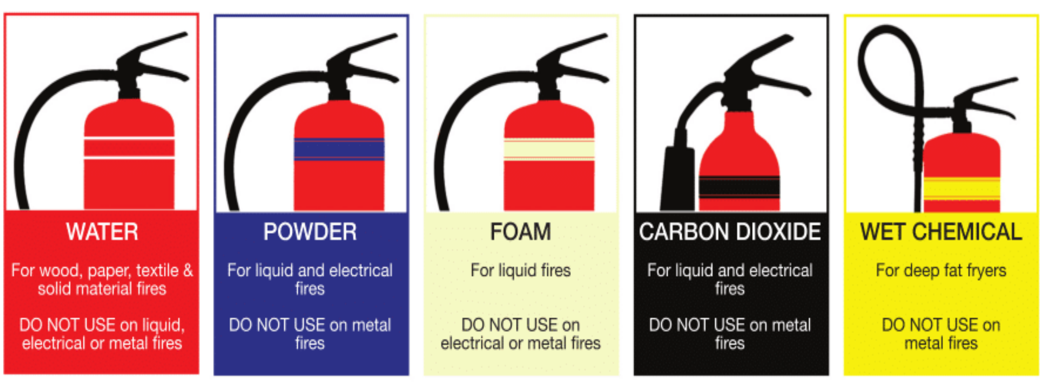
Two-way telephone communication sub-system shall be required for conventional system with FACP of more than 10 zones and for addressable and semi-addressable system with more than 200 automatic detectors.

Components of ***Fire Protection System***

- ✓ Fire Suppression System
- ✓ Fire Sprinkler System
- ✓ Fire Hose Cabinets
- ✓ Fire Dept. Connection
- ✓ Fire Pump and Jockey Pump
- ✓ Firewater Tank
- ✓ Stairwell Pressurization System
- ✓ Gas Extractor

The ***Fire Code of the Philippines*** (RA 9514) was enforced by the *Bureau of Fire Protection* (BFP) to ensure public safety, promote economic development through the prevention and suppression of all kinds, of destructive fires, and promote the professionalization of the fire service as a profession.





The contents of an extinguisher is indicated by a zone of colour on the red body of the extinguisher

ASSESSMENT

Group Task

Watch the following YouTube tutorials about MATLAB and perform the tasks they feature:

- ✓ What is PID Control?
- ✓ Expanding Beyond a Simple Integral
- ✓ Expanding Beyond a Simple Derivative
- ✓ A PID Tuning Guide
- ✓ Three Ways to Build a Model
- ✓ Manual and Automatic Tuning Methods
- ✓ Important PID Concepts

See the discussion proper on the Introduction to MATLAB (Module 4),

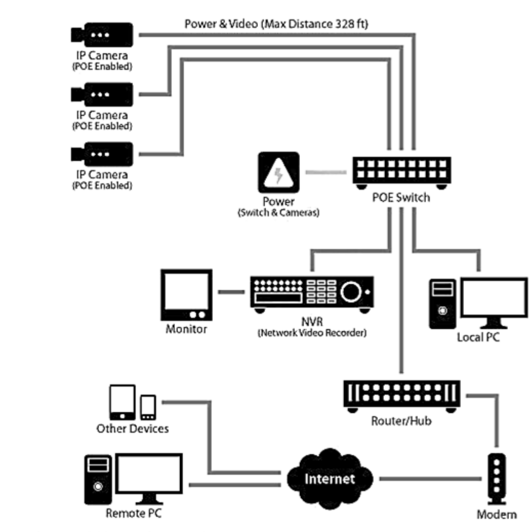
Module 11: **CCTV and Public Address Systems**

Closed-Circuit Television (CCTV) **System** is a video system in which specifically placed cameras record video images and transmit these to a monitor/monitors, with only a small number of people being able to access it. Also known as ***video surveillance***, this system is primarily used for security purposes to monitor both public and property security, and for monitoring and surveillance purposes to prevent or investigate crimes.

“Closed-circuit” means broadcasts are usually transmitted to a limited (closed) number of monitors, unlike “regular” TV, which is broadcast to the public at large.

Types of CCTV System

- **Analog** - Uses BNC connectors on coaxial cables to transmit continuous video signals. It has relatively low resolution but cheap and effective. The images require a video capture card and can be stored on a PC or recorder.
- **Digital** – It digitalizes signals at camera level and doesn’t require a video capture card as images are stored directly to a computer but require a (relatively) large amount of space to store recordings, so they are usually heavily compressed.
- **Network** (IP) – Used with analog or digital cameras, it utilizes a video server to stream footage over the internet. IP cameras have the ability to contain more cameras in one, which can cover a wide angle that may normally take multiple cameras or camera systems to cover. Advantages include:
  - the possibility of WiFi and audio
  - *Distributed Artificial Intelligence* (DAI) for analyzing image footage
  - remote access
  - *Power Over Ethernet* (POE)
  - better resolution.



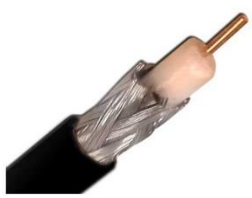
Description	Analog	Digital
Number of "Channels"	1 per coax (video) 100+ via RF modulation	Virtually Infinite using IP Multicasting
Video Sources	Analog cameras, tuners, satellite feeds	Analog cameras, tuners, satellite feeds
Reach	Anywhere there is a coax	Anywhere there is Ethernet
Quality	Variable	DVD-quality
Network Input	RF Modulators	MPEG Encoders
Network Output	Analog Set Top, TV monitor	Digital Set Top, PC Screens
Video On Demand	Limited: Requires RF Frequency Management (one VoD per RF channel)	Limited only by network bandwidth (one Ethernet Switch supports 20 to 200 VoD sessions)
Two Way Television (conferencing)	Difficult: each source requires its own RF modulator and dedicated RF TV "channel".	Unlimited, automatic setup. Very few network dependencies
Installation	Costly: Coax cable to every location	None: Uses existing Ethernet network
Live Television Channel Selection	IR Remote Control	IR Remote Control, mouse-click for PC
Management	Static channel assignment	Automatic – channels exist as needed
Channel Guide	Fixed or scheduled channel guide on dedicated RF channel	Dynamically generated digital display on TV and PC
Recording	Analog VCR's connected via coax spider	Digital recording scheduled or on demand
Viewer Statistics	Difficult	Automatic
Skills Required	Video, RF engineering	IP Networking
Cost	Low	Medium (low compared to installing new coax everywhere)

IP Camera Cabling:

Twisted Pair Network Cable

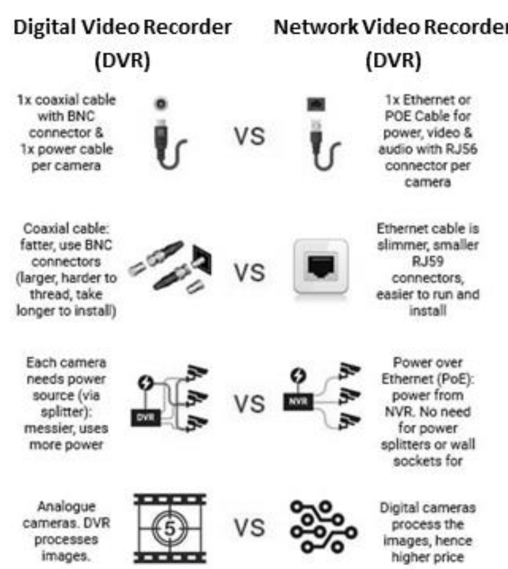
RJ45 terminated cables

Analog Camera Cabling:



Coaxial Cable

BNC terminated cables



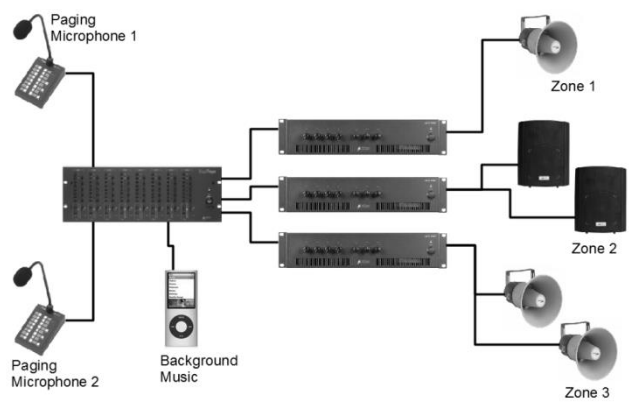
CCTV Design Considerations

1. **Purpose**. This is a legal requirement for people or organizations installing CCTV. In most countries, you are not allowed to install a camera unless there is an ‘identified, pressing need’. This should be justified at the design stage of the system.
2. **Location**: Design should begin with identifying where the cameras are to be located and exactly what they are going to view.
3. **Camera ID Level**. The wording of this standard should correspond to the ID standards within British Standard BS IEC 62676 – 4. This indicates that you will need a minimum resolution of 125 pixels per meter if you are to obtain Facial Recognition of a Known Person and 250 pixels per meter to obtain Facial Identification of an Unknown Person.
4. **Pictures per Second** (PPS). Normally, a minimum of 6 PPS is required but with the price of HDD storage, reducing it is becoming increasingly common to record in true time (25 PPS). This indicates camera’s lowlight capability.
5. **Camera Lens**. If you double the width of the picture you will half the resolution obtained. This indicates the appearance of the camera the common styles being box, bullet and dome cameras.
6. **Dynamic Range**. Using a camera without a wide dynamic range when viewing doorways or windows from the inside will result in silhouetting.
7. **Hard Drive**. With IP CCTV systems you can easily be looking at 1TB or more per camera for a month’s recording. This will also depend on how many megapixels each camera can produce (e.g. A 4K camera will require around 8 times the storage of a 1 megapixel camera.). 25 PPS will require over 4 times the storage of 6 PPS.
8. **Bandwidth**. Often the usable bandwidth is only 60% of that which is available. If you have insufficient bandwidth then the cameras will record at either a lower resolution or less pictures per second than is available.
9. **Record Resolution**. A 5-megapixel (MP) camera can be set to record at a fraction of its resolution. The quality of the monitor should match that of the pictures produced by the cameras
10. **Download Writer**. Best practice precludes downloading images to a USB stick. WORM (Write Once Read Many) media, correctly packaged, can prove that images have not been tampered with. If you wish to have resilient evidence in a court of law a CD/DVD writer is strongly recommended.

Public Address System (PA System) is a system of microphones, amplifiers, and loudspeakers used to amplify speech or music in a large building or at an outdoor gathering. It is a sound enhancement system that increases the apparent volume of vocals, musical instruments, other sound sources, or recorded sounds or music, and this can be used in public places , whether in a remote or a large area.

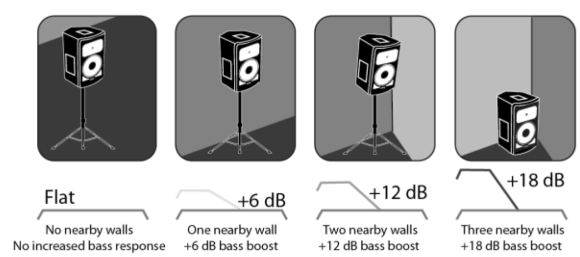
Factors to be Considered in a PA System Design

1. The output power of the PA system (should be adequate for a giving application).
2. Number of microphones that can be connected at the input.
3. Facility of operating the PA system on the DC batteries in the event of power failure.
4. Provision to connect a tape recorder or audio player at its input.
5. Provision of a tone control/graphic equalizer circuit.
6. Separate amplitude control for each input microphone.
7. Number of speakers that can be driven by the PA system.
8. Facility to use the wireless microphones.
9. Size, weight and cost.
10. Guarantee for reliable operation and after sale service.

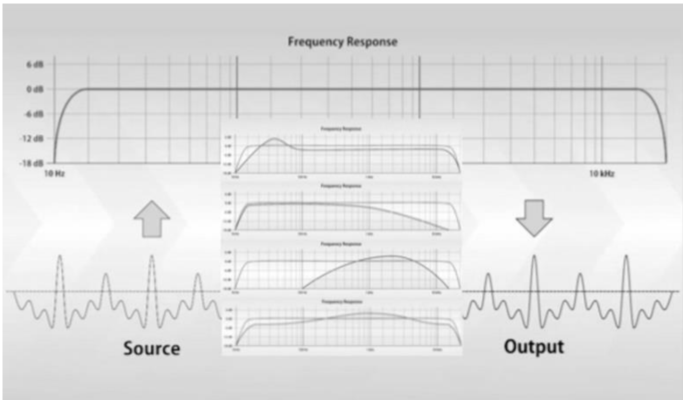


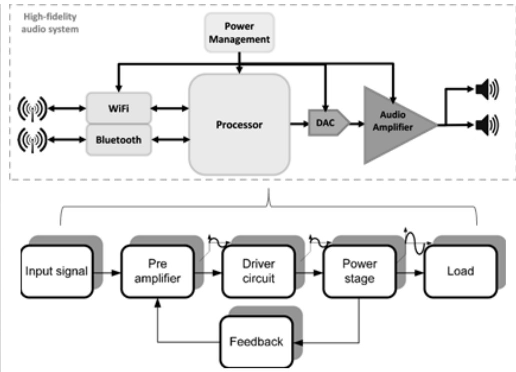
Requirements of a PA System

1. Avoid the acoustic feedback.
2. Distribute the sound intensity uniformly.
3. Reduce reverberations.
4. Create a sense of direction.
5. Use proper speaker orientation.
6. Select proper microphones and speakers.
7. Loud speaker impedances should be matched properly.
8. Use closed ring connection for loud speakers.
9. Proper grounding should be provided.



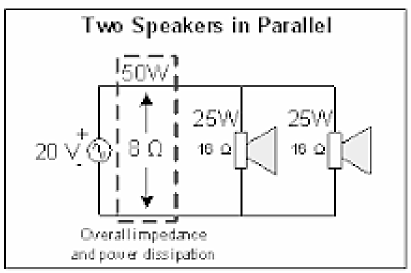
High Fidelity (Hi-Fi) audio systems reproduce sound with minimal background disturbances and offer the closest resemblance to the original sound.





Hi-Fi Audio Control System Block Diagram

Impedance matching

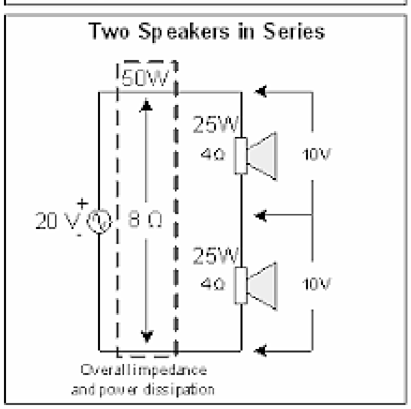


Formula for calculating the equivalent overall impedance of speakers wired in parallel

Z(total) = Equivalent Overall Impedance

Z(1) = Impedance of speaker one

$$Z(\text{total}) = \frac{1}{\frac{1}{Z(1)} + \frac{1}{Z(2)} + \frac{1}{Z(3)} + \cdots}$$

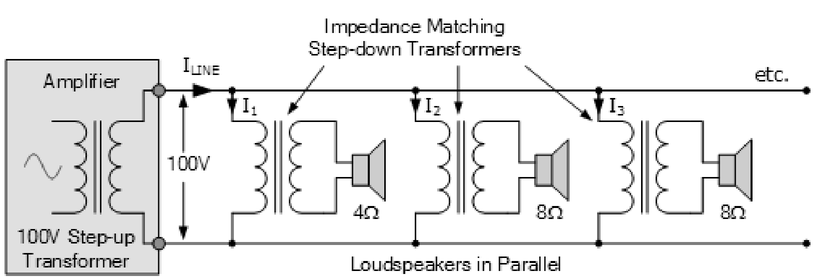


Formula for calculating the equivalent overall impedance of speakers wired in series

Z(total) = Equivalent Overall Impedance

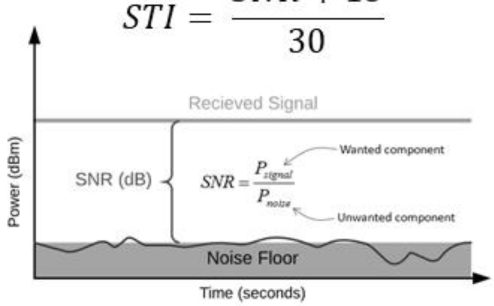
Z(1) = Impedance of speaker one

$$Z(\text{total}) = Z(1) + Z(2) + Z(3) + \ldots$$



The ***Speech Transmission Index*** (STI) provides the sound transmission with an objective value, focusing the physical phenomenon of sound mixing. In short, the easiness or difficulty of speech listening can be assessed by the resemblance of the original sound wave and transmitted sound wave to ears. The STI values range from 0 to 1, denoting worst to best intelligibility, respectively.

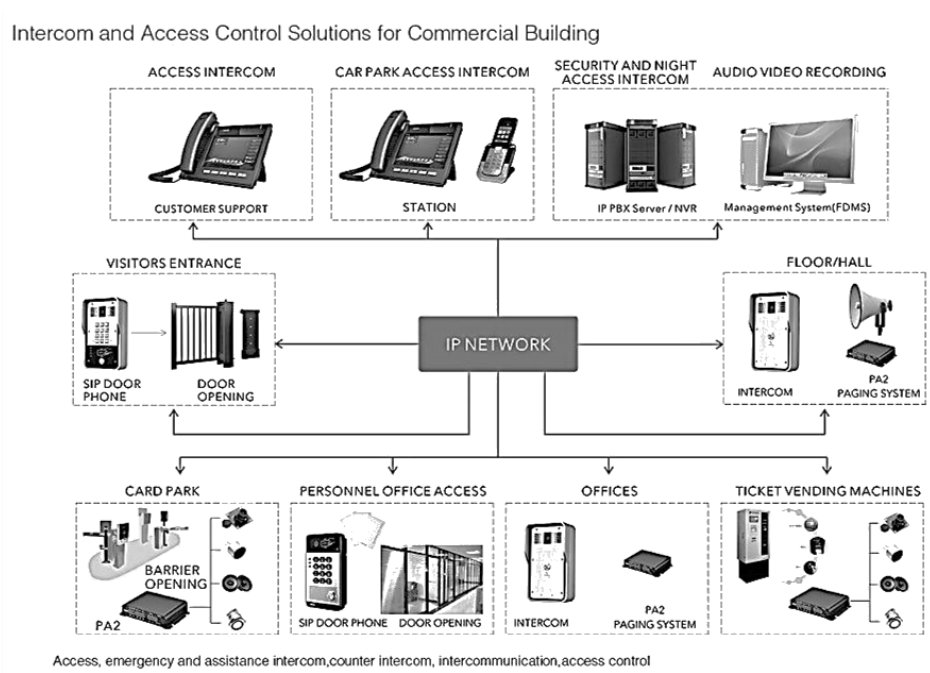
$$STI = \frac{SNR + 15}{30}$$



STI-Value	Quality1)	Intelligibility of syllables in %	Intelligibility of words in %	Intelligibility of sentences in %
0 – 0.3	Bad	0 – 34	0 – 67	0 – 89
0.3 – 0.45	Poor	34 – 48	67 – 78	89 – 92
0.45 – 0.6	Fair	48 – 67	78 – 87	92 – 95
0.6 – 0.75	Good	67 – 90	87 – 94	95 – 96
0.75 – 1	Excellent	90 – 96	94 – 96	96 – 100

1) A more detailed classification of speech intelligibility (STI) into 12 categories between A+ (better than 0.76) and U (worse than 0.36) can be found in EN 60268-16 Annex F and Annex G.

Intercom systems, installed in many buildings, have both speakers throughout a building, and microphones in many rooms so occupants can respond to announcements. PA and Intercom systems are commonly used as part of an ***emergency communication system*** (ECS).



ASSESSMENT

Group Task

Watch the following YouTube tutorials about MATLAB and perform the tasks they feature:

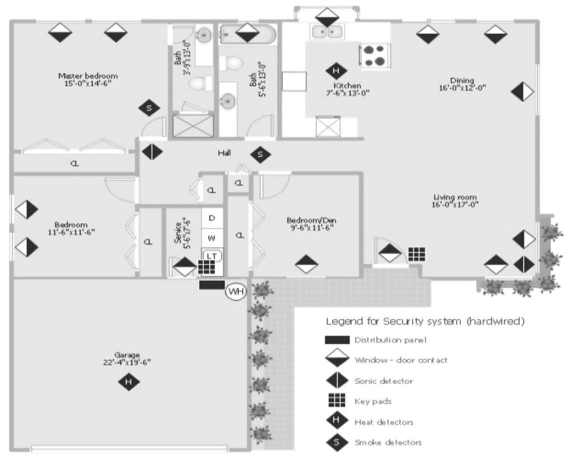
- ✓ Camera Calibration with MATLAB
- ✓ Real-Time Face Detection using MATLAB
- ✓ Active Noise Cancellation
- ✓ Real-Time Audio Processing for Algorithm Prototyping and Custom
- ✓ Prototyping Audio Processing Applications on a Raspberry Pi

See the discussion proper on the Introduction to MATLAB (Module 4),

Module 12: **Security Access Control Systems**

Access Control System (ACS) is a security system in which an access in a specified location is selectively restricted and permitted only upon presentation of proper credentials. It is basically intended to determine:

- ✓ <u>Who</u> is allowed to enter or exit
- ✓ <u>Where</u> they are allowed to enter or exit
- ✓ <u>When</u> they are allowed to enter or exit

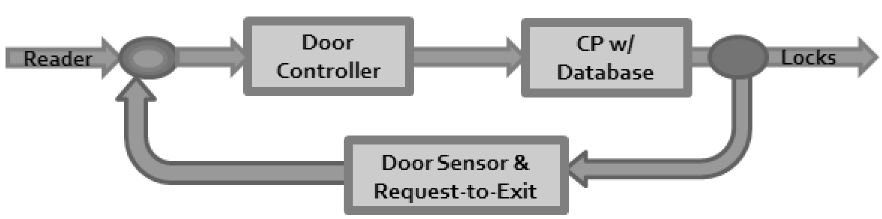


Defining the Purpose of an ACS

- Every door to be restricted have to be defined with a specific ***Restriction Level*** within the same Access Control System. Their security level will determine their ***Door Profile*** for the restricted doors well as the ***Access Profile*** for the users.
- Ideally, the coverage requirement for any ACS is 100%. However due to various constraints such as financial, environmental and/or structural factors, certain tolerances are accepted so long as the main design objective is not compromised.

How an ACS Works

1. A *credential* is presented to a reader.
2. The *reader* sends the credential's info to the control panel.
3. The *control panel* compares the credential's info to an access list.
4. Based on the outcome of the comparison, the control panel can either grant or deny the presented request.
5. The control panel sends a transaction log to the *database*.



ACS Credentials (or *authenticating information*)

- ✓ Something the user **knows** --- *Password, Pass-phrase, PIN code*
- ✓ Something the user **has** --- *Key, RFID Card, Key Fob*
- ✓ Something the user **is** --- *Biometrics*

ACS Reader could be a keypad where a code is entered, or a card/biometric scanner that reads the credential of the party requesting access, but don't usually make an access decision, but send a code to an access control panel that verifies the number against an access list.

Basic Reader Semi-Intelligent Reader Intelligent Reader

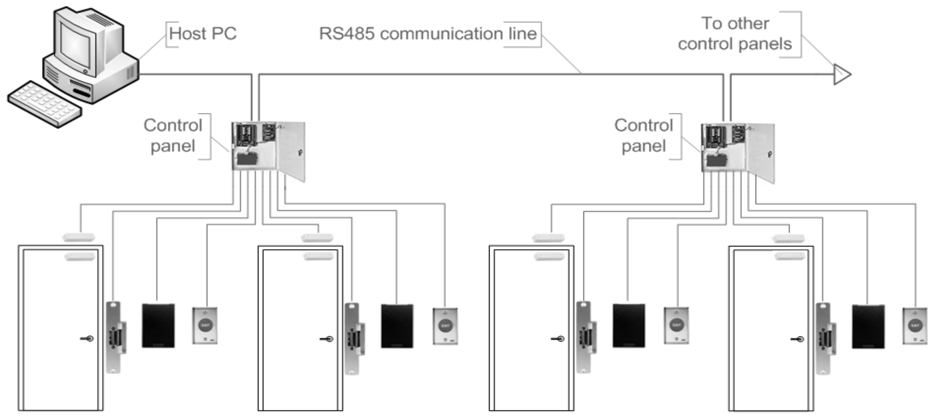
ACS Locks can be as simple as a mechanical, an electromagnetic or an electric lock, or as complex as electromechanical or pneumatic-powered barriers that restrict access.

Request-to-Exit (REX) is the component when triggered, the door alarm is temporarily ignored while the door is opened; exiting a door without having to electrically unlock the door is called *mechanical free egress*.

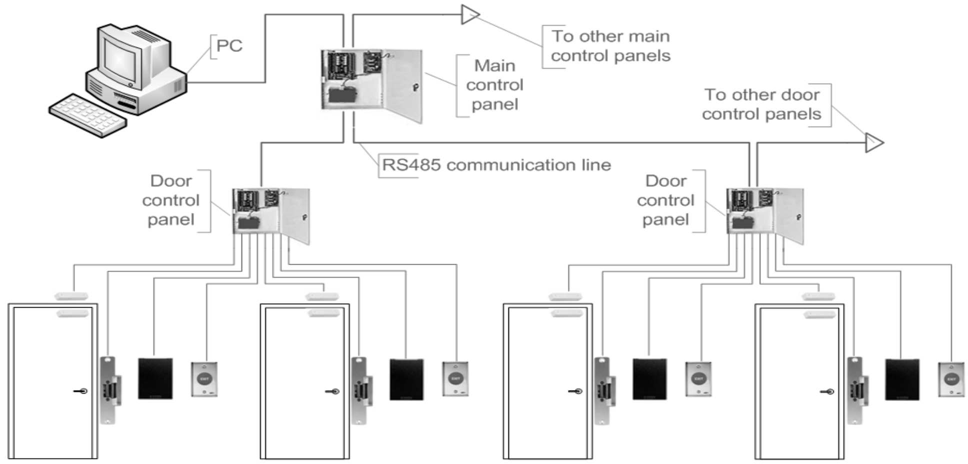
Door Sensors monitor the status of the physical door, whether it is closed, open or ajar. They provide feedback information to the control panel.

Access Control System Topologies

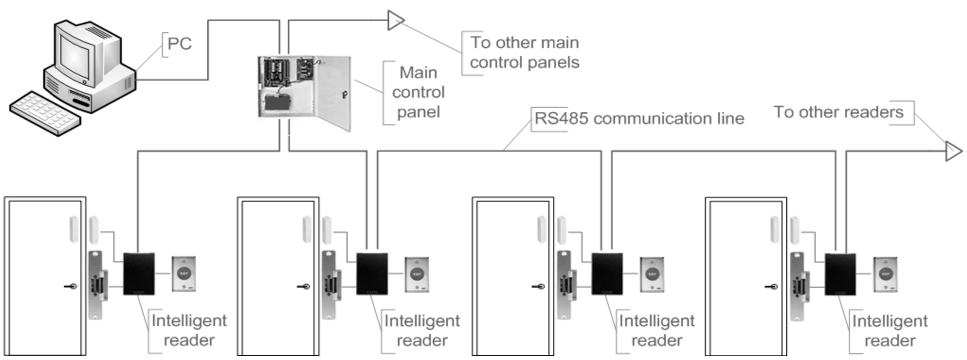
1. Serial Controllers



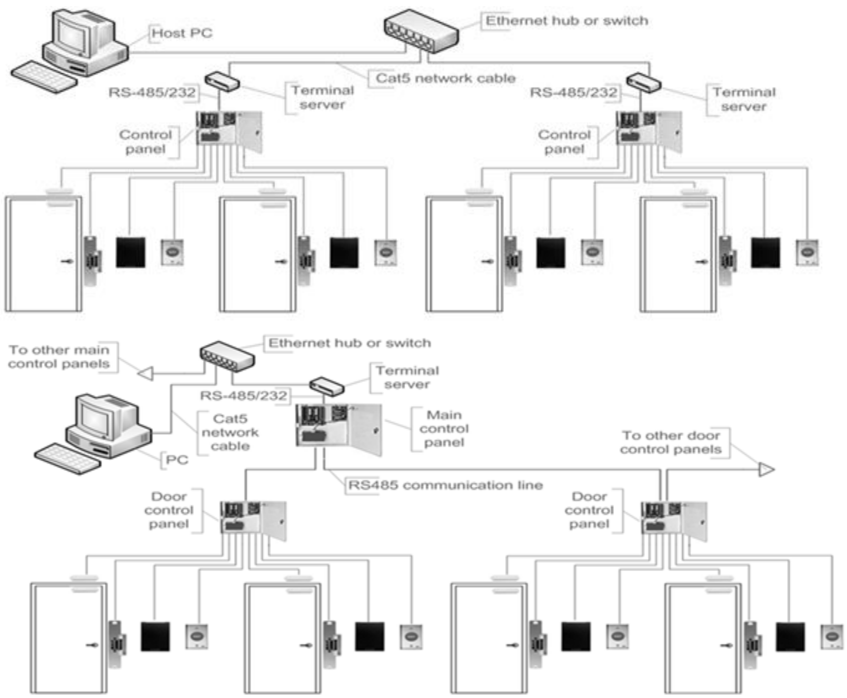
2. Serial Main Controller and Sub-Controllers



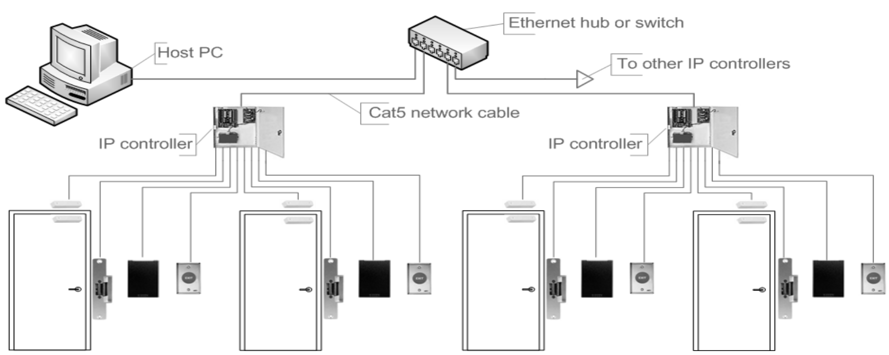
3. Serial Main Controller and Intelligent Readers



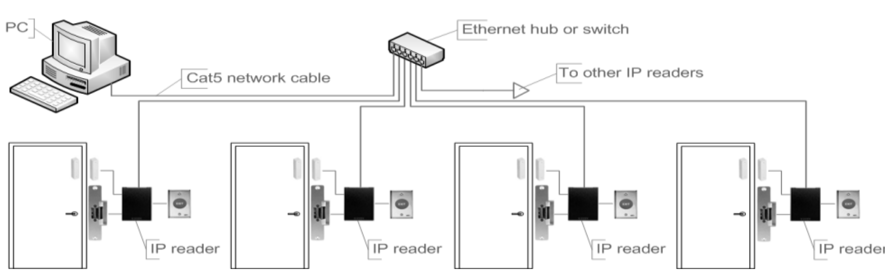
4. Serial Controllers and Terminal Servers



5. IP Controllers



6. IP Readers



Door Policy (Set of Door Responses)

- ✓ Normal Mode – usually the normal operating condition of the system.
- ✓ Degraded Mode – usually the operating condition of the system when the communication link between controllers had been disconnected.
- ✓ FA Mode – usually the operating condition of the system under fire alarm activation.
- ✓ Loss of Power – usually the operating condition of the system when power supply to the field devices had been cut-off

Door Response

- Fail-Safe – automatic unlock response without restriction on ingress and egress.
- Fail-Secure – automatic lock response with restriction on ingress only

Restriction Level	**Authentication Type**
Type 1 – Highly Restrictive	2-3 Factor Credentials Authentication
Type 2 – Moderately Restrictive	Single Factor Credential Authentication
Type 3 – Low Restrictive	Mechanical Key

Restriction Type	**Applied Main Field Device**	**Response Type**
Type A	Access Restricted by EM Lock	Fail-Safe
Type B	Access Restricted by other automatic locking mechanism	Fail-Secure
Type C	Access Restricted by Mechanical Lock	n/a

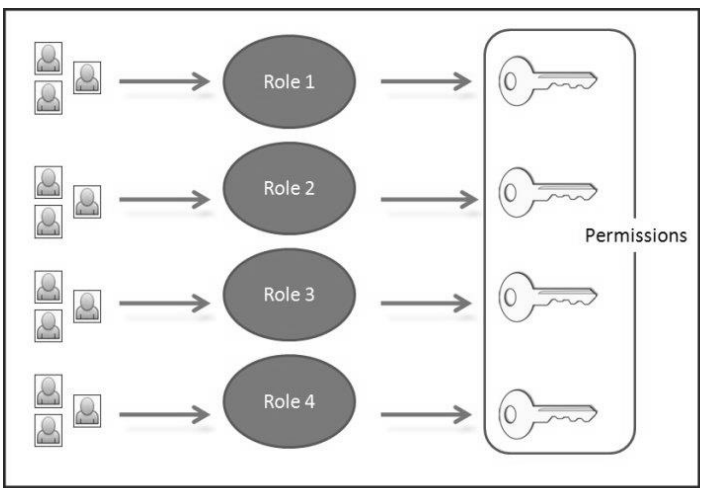
Door Profiles (Access Door Architecture)

- Type 1A – with EM Lock, Door Sensor, Contact-less Smart Card Reader with Keypad, Push Button to request EXIT, Junction Box and Door Controller.
- Type 1B – with Electric Door Strike, Push Bar, Door Sensor, Contact-less Smart Card Reader with Keypad (WP-Weatherproof), Junction Box & Door Controller.
- Type 1C – with Mechanical Lock, Push Bar, Door Sensor, EM Door Lock, Contact-less Smart Card Reader with Keypad, Junction Box and Door Controller.
- Type 2A – with EM Lock, Door Sensor, Contact-less Smart Card Reader without Keypad, Junction Box and Door Controller.

- Type 2B – with Push Button, Vehicle Sensor, Contact-less Smart Card Reader without Keypad (WP-Weatherproof), Junction Box and Door Controller.
- Type 2C – with EM Lock, Door Sensor, Contact-less Smart Card Reader without Keypad (WP-Weatherproof), Push Button to request EXIT, Junction Box and Door Controller.
- Type 3A – with Mechanical Lock, Door Sensor, Junction Box and Door Controller.
- Type 3B – with Mechanical Lock, Push Bar, Door Sensor, Junction Box and Door Controller.
- Type 3C – with Push Bar, Door Sensor, Junction Box and Door Controller.
- Type 3D – with Door Sensor, Junction Box and Door Controller.

Access Profile

- pertains to the level of access rights defined for a specific user.
- often defined during card registration or after the biometric specimen had been taken.



ASSESSMENT

Group Task

Watch the following YouTube tutorials about MATLAB and perform the tasks they feature:

- ✓ Motor Control on Embedded Devices
- ✓ What is a Field-Oriented Control?
- ✓ Modeling Motor, Inverter, and Controller
- ✓ Automatic Tuning of Controllers for an Induction Motor
- ✓ Auto-Tuning a Field-Oriented Controller

See the discussion proper on the Introduction to MATLAB (Module 4),

References and Suggested Readings

- R. C. Dorf and R. H. Bishop, *Modern Control Systems*
- G. F. Franklin, J. D. Powell. A. Emami-Naeini, *Feedback Control of Dynamic Systems*,
- *Sensors & Transducers Journal and Magazine* (S&T e-Digest)
- Nicolas P. Sands, Ian Verhappen, *A Guide to the Automation Body of Knowledge*
- Don Johnson, *Fundamentals of Electrical Engineering I*
- Louis E. Frenzel Jr., *Principles of Electronic Communication Systems*
- Manuel Gooding, P.E., *Introduction to Control and Instrumentation*
- Tracy Adams, P.E., *SCADA System Fundamentals*
- John R. Hackworth, Frederick D. Hackworth, Jr., *Programmable Logic Controllers*
- Shakhatreh, Fareed, *The Basics of Robotics*
- **Various Online Resources**

Downloadable Course AVPs: https://bit.ly/3iLs1dM
Downloadable Course PDFs: https://bit.ly/3mxuSs5