SRI CHANDRASEKHARENDRA SARASWATHI VISWA MAHAVIDYALAYA
(University U/S 3 of UGC Act 1956)
Accredited with "A" Grade by NAAC
ENATHUR, KANCHIPURAM - 631561

Course Material

SUBJECT	:	**COMPUTER CONTROL OF PROCESSES**
BRANCH	:	**EIE**
YEAR/SEM	:	**FOURTH/SEVENTH**

Prepared by

K.SARASWATHI,

Assistant Professor
Department of Electronics and Instrumentation Engineering
Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya
Enathur, Kanchipuram – 631561

PROGRAMME: B.E. BRANCH: Electronics and Instrumentation Engineering

Semester	Subject Code	Subject Name	Total Contact Hours	Weekly Hours			Credit
				L	T	P	
VII	**-**	**Computer Control of Processes**	**45**	**3**	**0**	**0**	**3**

(For Students admitted from 2018 onwards)

Pre-requisite: Control systems.

Aim

To learn the basic methods of design of discrete data systems and digital controller in multiloops.

Course Objectives

The course will enable the students to:

- To represent the linear time invariant system in discrete State Space form.
- To design Digital controllers
- To study the techniques of DAS, DDC, AI and SCADA.
- To introduce System identification techniques.
- To design Multi-loop and multivariable controller for multivariable system.

UNIT-I ANALYSIS OF DISCRETE DATA SYSTEM

State-space representation of discrete data systems: Selection of sampling process – Selection of sampling period – Review of z-transform – Pulse transfer function – Modified z-transform - Stability of discrete data system – Jury's stability test.

UNIT-II DESIGN OF DIGITAL CONTROLLER

Digital PID – Position and velocity form – Deadbeat's algorithm – Dahlin's algorithm – Kalman's algorithm - Pole placement controller – Predictive controller.

UNIT-III COMPUTER AS A CONTROLLER

Basic building blocks of computer control system – Data acquisition systems – SCADA – Direct digital control – Introduction to AI and expert control system – Case study - Design of computerized multi loop controller.

UNIT-IV SYSTEM IDENTIFICATION

Non Parametric methods: Transient Analysis, Frequency analysis, Correlation analysis, Spectral analysis. Parametric methods: Least Square method, Recursive least square method.

UNIT-V MULTI LOOP REGULATORY CONTROL

Multi-Loop Control: Introduction, Process Interaction, Pairing of Input and Outputs, Relative Gain Array (RGA) - Properties and Application of RGA, Multi-loop PID Controller - Decoupler.

TEXT BOOKS

1. P.B. Deshpande, and R.H.Ash, "Computer Process Control", ISA Publication, USA, 1995.
2. Sigurd Skogestad, Ian Postlethwaite,"Multivariable Feedback Control: Analysis and Design", John Wilry ans Sons, 2005.

REFERENCE

1. C.M.Houpis, G.B.Lamount, "Digital Control Systems Theory, Hardware and Software", International Student Edition, McGraw Hill Book Co., 1985.
2. G. Stephanoupoulis, "Chemical Process Control", Prentice Hall of India, New Delhi, 1990.
3. Singh, "Computer Aided Process Control", Prentice Hall of India, 2004.

Course Outcomes

At the end of the course the students will be able to

CO1. Able to understand the analysis of discrete data system

CO2. Able to design various digital control algorithms.

CO3. Able to learn the techniques of DAS, DDC, AI and SCADA.

CO4. Ability to build models from Input-Output data.

CO5. Ability to design Multi-loop and multivariable controller for multivariable system.

Pre-Test:

1. The Z transform of Z{Gho(s).Gp(s} is
   a) **GhoGp(z)**
   b) Gho(z).Gp(z)
   c) Gho(z).Gp(s)
   d) Gho(s).Gp(z)
2. The transfer function of Zero order hold is
   a) **1-e^{-st}/s**
   b) 1-s
   c) 1+e^{-st}/s
   d) None
3. Which of the following are the digital controller algorithms?
   a) Deadbeat Controller
   b) Dahlin Algorithm
   c) Kalman's Algorithm
   d) **All the mentioned**
4. In Dahlin's Algorithm α_f is used to
   a) **Enhance the robustness of the loop**
   b) Eliminate error
   c) For good response

d) None

5. Among the following which is not the part of SCADA System
   a) Remote terminal unit
   b) Communication network
   c) Data acquisition
   d) **Actuator**
6. What is the full form of SCADA?
   a) **Supervisory Control and Data Acquisition**
   b) Super Control and Data Acquisition
   c) Supervisory Control and Digital Acquisition
   d) Super Control and Digital Acquisition
7. Which among the following is a unique model of a system?
   a) **Transfer function**
   b) State variable
   c) Block diagram
   d) Signal flow graphs
8. Which among the following is a disadvantage of modern control theory?
   a) Implementation of optimal design
   b) Transfer function can also be defined for different initial conditions
   c) Analysis of all systems take place
   d) **Necessity of computational work**
9. Which mechanism in control engineering implies an ability to measure the state by taking measurements at output?
   a) Controllability
   b) Observability
   c) Differentiability
   d) **Adaptability**
10. State model representation is possible using __________
   a) Physical variables
   b) Phase variables
   c) Canonical state variables
   d) **All of the mentioned**

UNIT - 1

ANALYSIS OF DISCRETE DATA SYSTEM

State-space representation of discrete data systems: Selection of sampling process – Selection of sampling period – Review of z-transform – Pulse transfer function – Modified z-transform - Stability of discrete data system – Jury's stability test.

State space representation for discrete time systems

In control engineering, a state-space representation is a mathematical model of a physical system as a set of input, output and state variables related by first-order differential equations or difference equations. State variables are variables whose values evolve over time in a way that depends on the values they have at any given time and on the externally imposed values of input variables. Output variables' values depend on the values of the state variables. The state of the system can be represented as a state vector within that space. To abstract from the number of inputs, outputs and states, these variables are expressed as vectors.

The internal state variables are the smallest possible subset of system variables that can represent the entire state of the system at any given time. The minimum number of state variables required to represent a given system, n, is usually equal to the order of the system's defining differential equation, but not necessarily. If the system is represented in transfer function form, the minimum number of state variables is equal to the order of the transfer function's denominator after it has been reduced to a proper fraction. It is important to understand that converting a state-space realization to a transfer function form may lose some internal information about the system, and may provide a description of a system which is stable, when the state-space realization is unstable at certain points. In electric circuits, the number of state variables is often, though not always, the same as the number of energy storage elements in the circuit such as capacitors and inductors. The state variables defined must be linearly independent, i.e., no state variable can be written as a linear combination of the other state variables or the system will not be able to be solved.

The Discrete State Space (or State Space) component defines the relation between the input and the output in state-space form, where is the value of the discrete state at the previous sample time instant. The input is a vector of length, the output is a vector of the length, and is the number of states.

The dynamics of a linear time (shift)) invariant discrete-time system may be expressed in terms state (plant) equation and output (observation or measurement) equation as follows

$$x(k+1) = Ax(k) + Bu(k),$$

$$y(k) = Cx(k) + Du(k)$$

Where x(k) is a n-dimensional state vector at time t =kT, r-dimensional control (input) vector u (k), m-dimensional output vector y(k), respectively, are represented as

$$x(k) = [x_1(k), x_2(k), \dots . x_n(k)]^T \ ,$$

$$u(k) = [u_1(k), u_2(k), \dots . u_r(k)]^T,$$

$$y(k) = [y_1(k), y_2(k), \dots . y_m(k)]^T$$

The parameters (elements) of A, an $nX \ n$ (plant parameter) matrix. B an nX r control (input) matrix and C An m X r output parameter, D an m X r parametric matrix are constants for the LTI system. Similar to above equation state variable representation of SISO (single output and single output) discrete-rime system (with direct coupling of output with input) can be written as

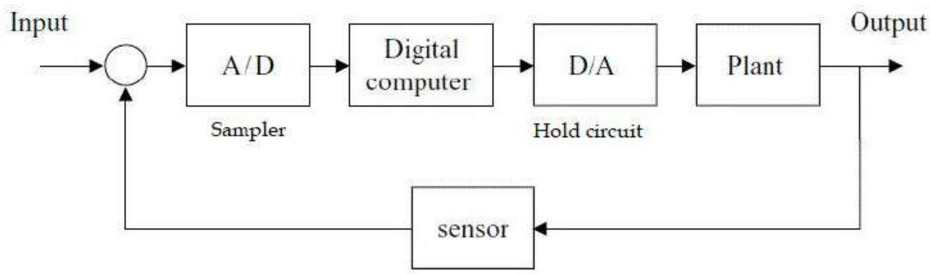
$$x(k+1) = Ax(k) + Bu(k),$$

$$y(t) = C^T x(k) + Du(k)$$

Where the input u, output y and d. are scalars, and b and c are n-dimensional vectors. The concepts of controllability and observability for discrete time system are similar to the continuous-time system. A discrete time system is said to be controllable if there exists a finite integer n and input mu(k); k [0,n 1] that will transfer any state (0) $x^0 = bx(0)$ to the state x^n $at\ k = n$ n.

Sampled Data System

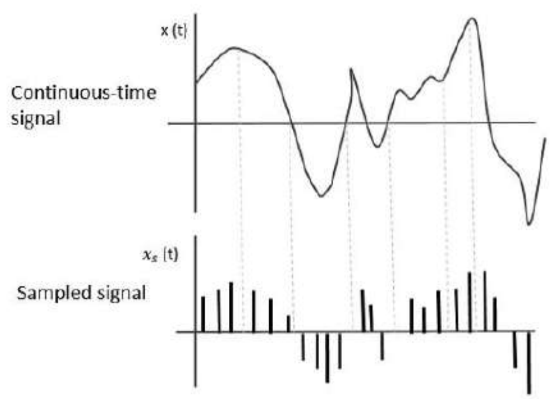
When the signal or information at any or some points in a system is in the form of discrete pulses, then the system is called discrete data system. In control engineering the discrete data system is popularly known as sampled data systems.



Sampled data control system

Sampling Theorem

Sampling is defined as, "The process of measuring the instantaneous values of continuous-time signal in a discrete form."**Sample** is a piece of data taken from the whole data which is continuous in the time domain. When a source generates an analog signal and if that has to be digitized, having **1s** and **0s** i.e., High or Low, the signal has to be discretized in time. This discretization of analog signal is called as Sampling.



Sampling Rate

To discretize the signals, the gap between the samples should be fixed. That gap can be termed as a **sampling period T_s**.

$$\text{Sampling Frequency} = 1/Ts = fs$$

Where,

- Ts is the sampling time
- fs is the sampling frequency or the sampling rate

Sampling frequency is the reciprocal of the sampling period. This sampling frequency, can be simply called as **Sampling rate**. The sampling rate denotes the number of samples taken per second, or for a finite set of values.

For an analog signal to be reconstructed from the digitized signal, the sampling rate should be highly considered. The rate of sampling should be such that the data in the message signal should neither be lost nor it should get over-lapped. Hence, a rate was fixed for this, called as Nyquist rate.

Nyquist Rate

A band limited continuous time signal with highest frequency fm hertz can be uniquely recovered from its samples provided that the sampling rate Fs is greater than or equal to 2fm samples per seconds.

Suppose that a signal is band-limited with no frequency components higher than **W** Hertz. That means, **W** is the highest frequency. For such a signal, for effective reproduction of the original signal, the sampling rate should be twice the highest frequency.

$$f_S = 2W$$

Where,

- f_S is the sampling rate
- **W** is the highest frequency

This rate of sampling is called as **Nyquist rate**.

A theorem called, Sampling Theorem, was stated on the theory of this Nyquist rate.

Aliasing

Aliasing can be referred to as “the phenomenon of a high-frequency component in the spectrum of a signal, taking on the identity of a low-frequency component in the spectrum of its sampled version.”

The corrective measures taken to reduce the effect of Aliasing are −

- In the transmitter section of PCM, a **low pass anti-aliasing filter** is employed, before the sampler, to eliminate the high frequency components, which are unwanted.
- The signal which is sampled after filtering, is sampled at a rate slightly higher than the Nyquist rate.

This choice of having the sampling rate higher than Nyquist rate also helps in the easier design of the **reconstruction filter** at the receiver.

Quantizing

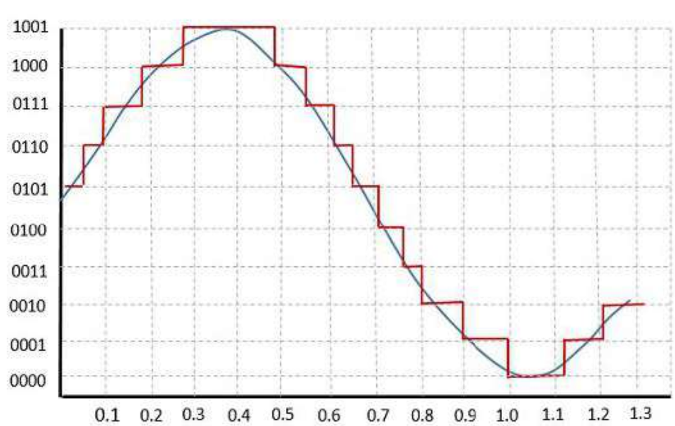
The digitization of analog signals involves the rounding off of the values which are approximately equal to the analog values. The method of sampling chooses a few points on the analog signal and then these points are joined to round off the value to a near stabilized value. Such a process is called as Quantization.

Quantizing an Analog Signal

The analog-to-digital converters perform this type of function to create a series of digital values out of the given analog signal. The following figure represents an analog signal. This signal to get converted into digital has to undergo sampling and quantizing.

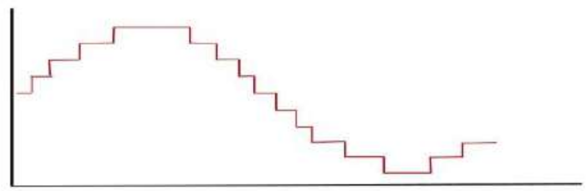
The quantizing of an analog signal is done by discretizing the signal with a number of quantization levels. **Quantization** is representing the sampled values of the amplitude by a finite set of levels, which means converting a continuous-amplitude sample into a discrete-time signal.

The following figure shows how an analog signal gets quantized. The blue line represents analog signal while the brown one represents the quantized signal.



Both sampling and quantization result in the loss of information. The quality of a Quantizer output depends upon the number of quantization levels used. The discrete amplitudes of the quantized output are called as representation levels or reconstruction levels. The spacing between the two adjacent representation levels is called a quantum or step-size.

The following figure shows the resultant quantized signal which is the digital form for the given analog signal.



Types of Quantization

There are two types of Quantization - Uniform Quantization and Non-uniform Quantization.

The type of quantization in which the quantization levels are uniformly spaced is termed as a Uniform Quantization. The type of quantization in which the quantization levels are unequal and mostly the relation between them is logarithmic, is termed as a Non-uniform Quantization.

There are two types of uniform quantization. They are Mid-Rise type and Mid-Tread type. The following figures represent the two types of uniform quantization.

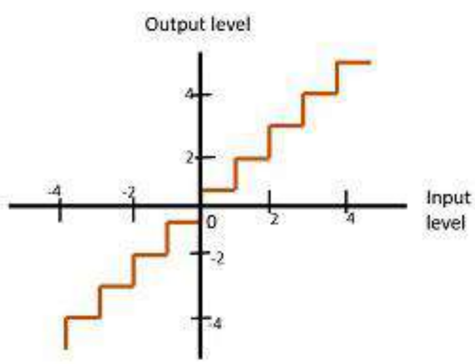


Fig 1 : Mid-Rise type Uniform Quantization

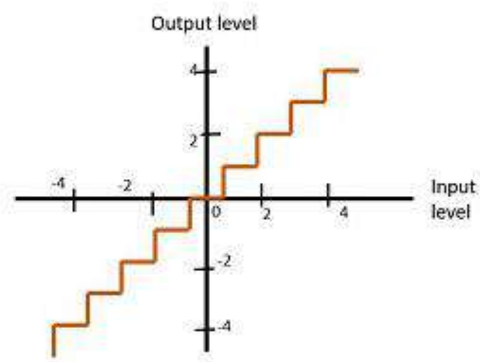


Fig 2 : Mid-Tread type Uniform Quantization

Figure 1 shows the mid-rise type and figure 2 shows the mid-tread type of uniform quantization.

- The Mid-Rise type is so called because the origin lies in the middle of a raising part of the stair-case like graph. The quantization levels in this type are even in number.
- The Mid-tread type is so called because the origin lies in the middle of a tread of the stair-case like graph. The quantization levels in this type are odd in number.
- Both the mid-rise and mid-tread type of uniform quantizer are symmetric about the origin.

Quantization Error

For any system, during its functioning, there is always a difference in the values of its input and output. The processing of the system results in an error, which is the difference of those values.

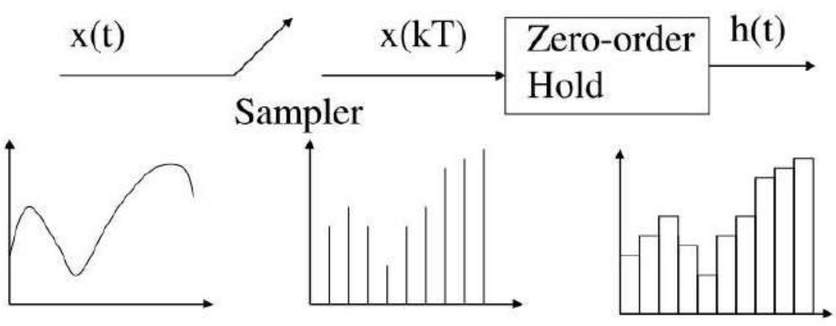
The difference between an input value and its quantized value is called a Quantization Error.

Quantization Noise

It is a type of quantization error, which usually occurs in analog audio signal, while quantizing it to digital. For example, in music, the signals keep changing continuously, where regularity is not found in errors. Such errors create a wideband noise called as Quantization Noise.

Sample and Hold Circuit

The Signal given to the digital controller is a sampled data signal and in turn the controller gives the controller output in digital form. But the system to be controlled needs an analog control signal as input. Therefore the digital output of controllers must be converters into analog form. This can be achieved by means of various types of hold circuits. The simplest hold circuits are the zero order hold (ZOH). In ZOH, the reconstructed analog signal acquires the same values as the last received sample for the entire sampling period.



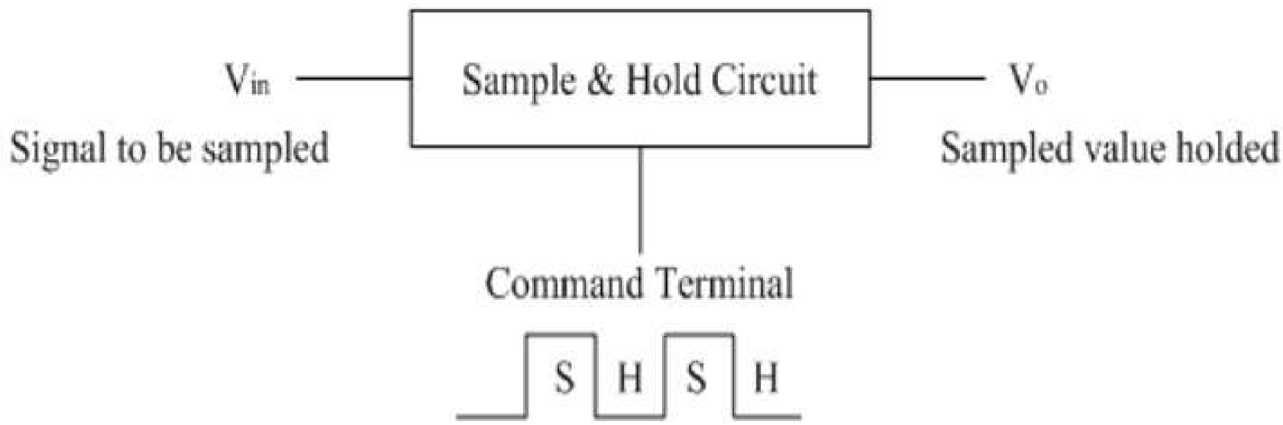
Zero Order Hold

The high frequency noises present in the reconstructed signal are automatically filtered out by the control system component which behaves like low pass filters. In a first order hold the last two signals for the current sampling period. Similarly higher order hold circuit can be devised. First or higher order hold circuits offer no particular advantage over the zero order hold.

A Sample and Hold circuit sometimes represented as S/H Circuit or S & H Circuit, is usually used with an Analog to Digital Converter to sample the input analog signal and hold the sampled signal.

In the S/H Circuit, the analog signal is sampled for a short interval of time, usually in the range of 10µS to 1µS. After this, the sampled value is hold until the arrival of next input signal to be sampled. The duration for holding the sample will be usually between few milliseconds to few seconds.

The following image shows a simple block diagram of a typical Sample and Hold Circuit.

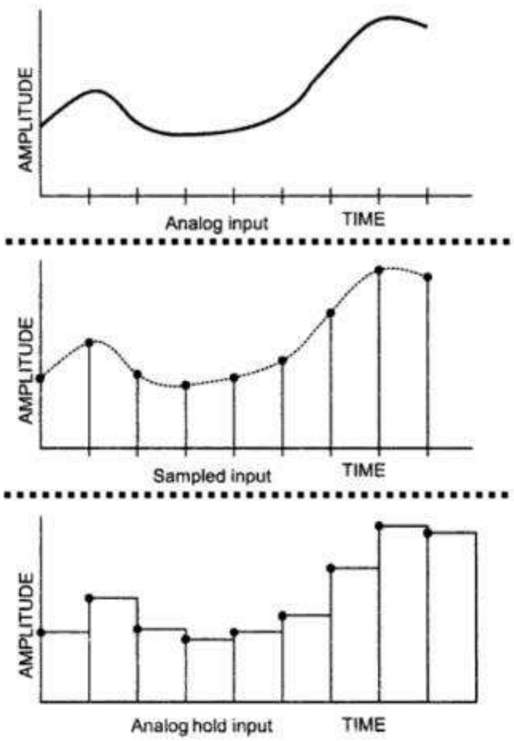


Need for Sample and Hold Circuits

If the input analog voltage of an ADC changes more than ±1/2 LSB, then there is a severe chance that the output digital value is an error. For the ADC to produce accurate results, the input analog voltage should be held constant for the duration of the conversion.

As the name suggests, a S/H Circuit samples the input analog signal based on a sampling command and holds the output value at its output until the next sampling command is arrived.

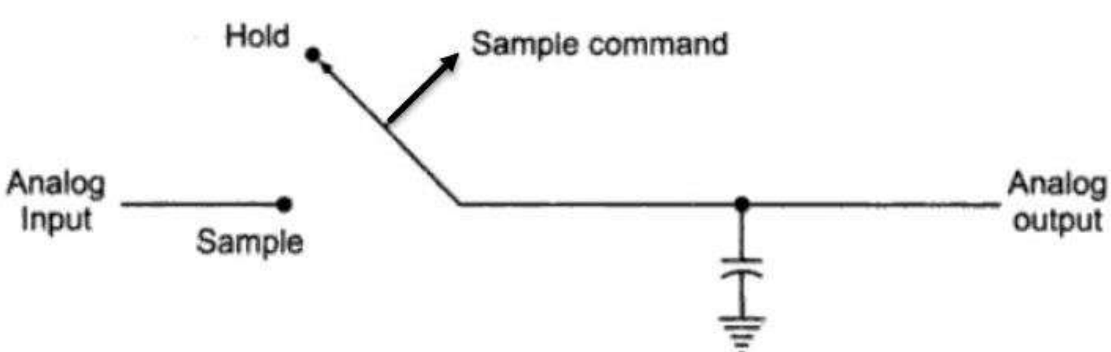
The following image shows the input and output of a typical Sample and Hold Circuit



Let us understand the operating principle of a S/H Circuit with the help of a simplified circuit diagram. This sample and hold circuit consists of two basic components:

- Analog Switch
- Holding Capacitor

The following image shows the basic S/H Circuit.



This circuit tracks the input analog signal until the sample command is changed to hold command. After the hold command, the capacitor holds the analog voltage during the analog to digital conversion.

Advantages of Sample and Hold Circuit

- The main and important advantage of a typical SH Circuit is to aid an Analog to Digital Conversion process by holding the sampled analog input voltage.
- In multichannel ADCs, where synchronization between different channels is important, an SH circuit can help by sampling analog signals from all the channels at the same time.
- In multiplexed circuits, the crosstalk can be reduced with an SH circuit.

Applications of Sample and Hold Circuit

Some of the important applications are mentioned below:

- Analog to Digital Converter Circuits (ADC)
- Digital Interface Circuits
- Operational Amplifiers
- Analog De-multiplexers
- Data distribution systems
- Storage of outputs of multiplexers
- Pulse Modulation Systems

Z- Transforms:

In signal processing, the Z-transform converts a discrete-time signal, which is a sequence of real or complex numbers, into a complex frequency-domain representation. It can be considered as a discrete-time equivalent of the Laplace transform. This similarity is explored in the theory of time-scale calculus.

SOME COMMON z-TRANSFORM PAIRS

Signal	Transform	ROC
1. $\delta[n]$	1	All z
2. $u[n]$	$\frac{1}{1-z^{-1}}$	$\|z\| > 1$
3. $-u[-n-1]$	$\frac{1}{1-z^{-1}}$	$\|z\| < 1$
4. $\delta[n-m]$	z^{-m}	All z, except 0 (if $m > 0$) or ∞ (if $m < 0$)
5. $\alpha^n u[n]$	$\frac{1}{1-\alpha z^{-1}}$	$\|z\| > \|\alpha\|$
6. $-\alpha^n u[-n-1]$	$\frac{1}{1-\alpha z^{-1}}$	$\|z\| < \|\alpha\|$
7. $n\alpha^n u[n]$	$\frac{\alpha z^{-1}}{(1-\alpha z^{-1})^2}$	$\|z\| > \|\alpha\|$
8. $-n\alpha^n u[-n-1]$	$\frac{\alpha z^{-1}}{(1-\alpha z^{-1})^2}$	$\|z\| < \|\alpha\|$
9. $[\cos \omega_0 n]u[n]$	$\frac{1-[\cos \omega_0]z^{-1}}{1-[2\cos \omega_0]z^{-1}+z^{-2}}$	$\|z\| > 1$
10. $[\sin \omega_0 n]u[n]$	$\frac{[\sin \omega_0]z^{-1}}{1-[2\cos \omega_0]z^{-1}+z^{-2}}$	$\|z\| > 1$
11. $[r^n \cos \omega_0 n]u[n]$	$\frac{1-[r\cos \omega_0]z^{-1}}{1-[2r\cos \omega_0]z^{-1}+r^2z^{-2}}$	$\|z\| > r$
12. $[r^n \sin \omega_0 n]u[n]$	$\frac{[r\sin \omega_0]z^{-1}}{1-[2r\cos \omega_0]z^{-1}+r^2z^{-2}}$	$\|z\| > r$

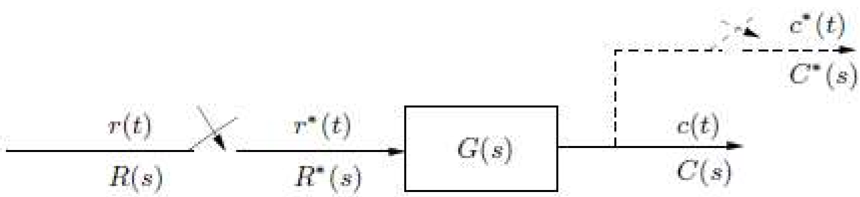
Pulse Transfer Function

Transfer function of an LTI (Linear Time Invariant) continuous time system is defined as

$$G(s) = \frac{C(s)}{R(s)}$$

Where R(s) and C(S) are Laplace transforms of input r (t) and output c (t) respectively. Assume that the initial conditions are zero.

Pulse transfer function relates z-transform of the output at the sampling instants to the z-transform of the sampled input. When the same is subject to a sampled data or digital data signal r*(t)



Block diagram of a system subject to a sampled input

The output of the system is C(s) = G(s) R*(s). The transfer function of the above system is difficult to manipulate because if contains a mixture of analog and digital components. Thus, it is desirable to express the system characteristics by a transfer function that relates r*(t) to c*(t), a fictitious sampler output.

$$G(z) = \frac{C(z)}{R(z)}$$

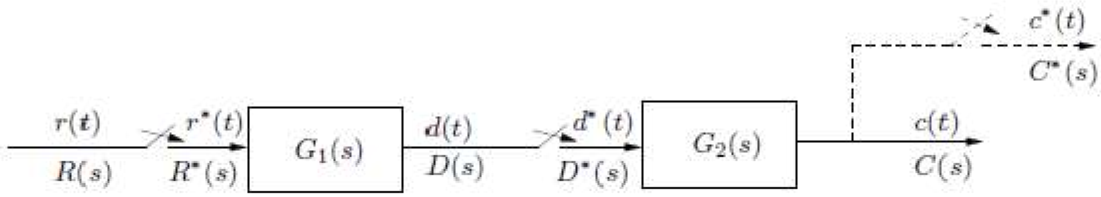
Overall Conclusion

1. Pulse transfer function or z-transfer characterizes the discrete data system responses only at sampling instants. The output information between the sampling instants is lost.
2. Since the input of discrete data system is described by output of the sampler, for all practical purpose the samplers can be simply ignored and the input can be regarded as r*(t).

Pulse transfer function for discrete data systems with cascaded elements

When discrete data systems has cascaded elements care should be taken in calculating the transfer function. Two cases of cascaded elements

1. Cascaded element are separated by a sampler
2. Cascaded element are not separated by a sampler



Discrete data system with cascaded elements, separated by a sampler

The input –output relation of the systems G1 and G2 are described by

$$D(z) = G_1(z)R(z)$$

And

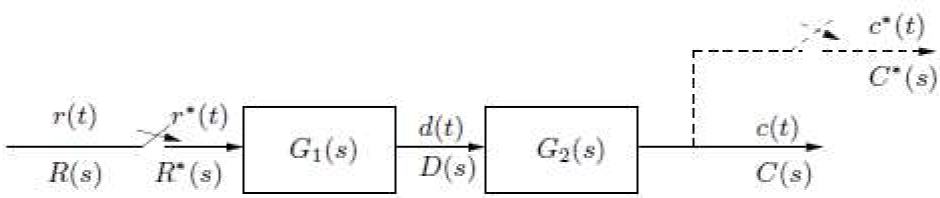
$$C(z) = G_2(z)R(z)$$

Thus the input-output relation of the overall system is

$$C(z) = G_1(z)G_2(z)R(z)$$

Therefore we can conclude that the z-transfer function of two linear system separated by sampler are the products of the individual z-transfer functions.

Cascaded element are not separated by a sampler



Discrete data system with cascaded elements, not separated by a sampler

The continuous output C(s) can be written as

$$C(s) = G_1(s)G_2(s)R^*(s)$$

The output of the fictitious sampler is

$$C(z) = Z[G_1(s)G_2(s)]\, R(z)$$

Z - Transform of the product $G_1(s)G_2(s)$ is denoted as

$$Z[G_1(s)G_2(s)] = G_1G_2(z) = G_2G_1(z)$$

One should note that in general $G_1G_2(z)\, not\ equal\ to\ G_1(z)G_2(z)$, except for some special cases. The overall output is thus,

$$C(z) = G_1G_2(z)R(z)$$

Modified Z Transform:

In mathematics and signal processing, the **advanced z-transform** is an extension of the z-transform, to incorporate ideal delays that are not multiples of the sampling time. It takes the form,

$$F(z,m) = \sum_{k=0}^{\infty} f(kT+m)z^{-k}$$

Where,

T is the sampling period

m (the "delay parameter") is a fraction of the sampling period

It is also known as the **modified z-transform**. The advanced z-transform is widely applied, for example to accurately model processing delays in digital control.

Properties

If the delay parameter, m, is considered fixed then all the properties of the z-transform hold for the advanced z-transform.

Linearity	$\mathcal{Z}\left\{\sum_{k=1}^{n} c_k f_k(t)\right\} = \sum_{k=1}^{n} c_k F_k(z,m).$
Time shift	$\mathcal{Z}\left\{u(t-nT)f(t-nT)\right\} = z^{-n}F(z,m).$
Damping	$\mathcal{Z}\left\{f(t)e^{-a\,t}\right\} = e^{-a\,m}F(e^{a\,T}z,m).$
Time multiplication	$\mathcal{Z}\left\{t^y f(t)\right\} = \left(-Tz\frac{d}{dz}+m\right)^y F(z,m).$
Final value theorem	$\lim_{k\to\infty} f(kT+m) = \lim_{z\to 1}(1-z^{-1})F(z,m).$

Example:

Consider the following example where $f(t) = \cos(\omega t)$:

$$\begin{aligned} F(z,m) &= \mathcal{Z}\left\{\cos(\omega\left(kT+m\right))\right\} \\ &= \mathcal{Z}\left\{\cos(\omega kT)\cos(\omega m) - \sin(\omega kT)\sin(\omega m)\right\} \\ &= \cos(\omega m)\mathcal{Z}\left\{\cos(\omega kT)\right\} - \sin(\omega m)\mathcal{Z}\left\{\sin(\omega kT)\right\} \\ &= \cos(\omega m)\frac{z\left(z-\cos(\omega T)\right)}{z^2-2z\cos(\omega T)+1} - \sin(\omega m)\frac{z\sin(\omega T)}{z^2-2z\cos(\omega T)+1} \\ &= \frac{z^2\cos(\omega m) - z\cos(\omega(T-m))}{z^2-2z\cos(\omega T)+1}. \end{aligned}$$

If $m = 0$ then $F(z,m)$ reduces to the transform

$$F(z,0) = \frac{z^2 - z\cos(\omega T)}{z^2-2z\cos(\omega T)+1},$$

which is clearly just the z-transform of $f(t)$.

Stability Analysis of Sample Data System

Stability Analysis of closed loop system in z-plane: Stability is the most important issue in control system design. Before discussing the stability test let us first introduce the following notions of stability for a linear time invariant (LTI) system.

1. BIBO stability or zero state stability
2. Internal stability or zero input stability

Since we have not introduced the concept of state variables yet, as of now, we will limit our discussion to BIBO stability only. An initially relaxed (all the initial conditions of the system are zero) LTI system is said tobe BIBO stable if for every bounded input, the output is also bounded.

However, the stability of the following closed loop system

$$\frac{C(z)}{R(z)} = \frac{G(z)}{1+GH(z)}$$

can be determined from the location of closed loop poles in z-plane which are the roots of thecharacteristic equation

$$1 + GH(z) = 0$$

1. For the system to be stable, the closed loop poles or the roots of the characteristic equation must lie within the unit circle in z-plane. Otherwise the system would be unstable.

2. If a simple pole lies at $|z| = 1$, the system becomes marginally stable. Similarly if a pair of complex conjugate poles lie on the $|z| = 1$ circle, the system is marginally stable. Multiple poles on unit circle make the system unstable.

Three stability tests can be applied directly to the characteristic equation without solving forthe roots.

- Schur-Cohn stability test
- Jury Stability test
- Routh stability coupled with bi-linear transformation.

Other stability tests like Lyapunov stability analysis are applicable for state space system models

Jury's Stability Test

Jury's test is a test that is similar to the Routh-Hurwitz criterion, except that it can be used to analyze the stability of an LTI digital system in the Z domain. To use Jury's test to determine if a digital system is stable, we must check our z-domain characteristic equation against a number of specific rules and requirements. If the function fails any requirement, it is not stable. If the function passes all the requirements, it is stable. Jury's test is a necessary and sufficient test for stability in digital systems.

Again, we call P(z) the characteristic polynomial of the system. It is the denominator polynomial of the Z-domain transfer function. Jury's test will focus exclusively on the Characteristic polynomial. To perform Jury's test, we must perform a number of smaller tests on the system. If the system fails any test, it is unstable.

Assume that the characteristic equation is as follows,

$$P(z) = a_0 z^n + a_1 z^{n-1} + \cdots + a_{n-1} z + a_n$$

Where $a_0 > 0$.

Jury Table

Row	z^0	z^1	z^2	z^3	z^4	$\dots$	z^n
1	a_n	a_{n-1}	$a_{n-2}\dots$	$\dots$	$\dots$		a_0
2	a_0	a_1	$a_2\dots$	$\dots$	$\dots$	$\dots$	a_n
3	b_{n-1}	b_{n-2}	$\dots$	$\dots$	b_0		
4	b_0	b_1	$\dots$	$\dots$	$\dots$	b_{n-1}	
5	c_{n-2}	$c_{n-3}\dots$	c_0				
6	c_0	$c_1\dots$	$\dots$	c_{n-2}			
.	....						
.	....						
.	....						
$2n-3$	q_2	q_1	q_0				

Where,

$$b_k = \begin{vmatrix} a_n & a_{n-1-k} \\ a_0 & a_{k+1} \end{vmatrix}$$

$$k = 0, 1, 2, 3, ..., n-1$$

$$c_k = \begin{vmatrix} b_{n-1} & b_{n-2-k} \\ b_0 & b_{k+1} \end{vmatrix}$$

$$k = 0, 1, 2, 3, ..., n-2$$

$$q_k = \begin{vmatrix} p_3 & p_{2-k} \\ p_0 & p_{k+1} \end{vmatrix}$$

This system will be stable if the following conditions are satisfied

1. $|a_n| \quad < a_0$
2. $P(z)|_{z=1} \quad > 0$
3. $P(z)|_{z=-1} \quad > 0$ for n even and $P(z)|_{z=-1} < 0$ for n odd
4.

$$\begin{aligned} |b_{n-1}| &> |b_0| \\ |c_{n-2}| &> |c_0| \\ . \quad & \ldots\ldots\ldots\ldots \\ . \quad & \ldots\ldots\ldots \\ |q_2| &> |q_0| \end{aligned}$$

Example : The characteristic equation: $P(z) = z^4 - 1.2z^3 + 0.07z^2 + 0.3z - 0.08 = 0$
Thus, $a_0 = 1 \quad a_1 = -1.2 \quad a_2 = 0.07 \quad a_3 = 0.3 \quad a_4 = -0.08$

We will now check the stability conditions.

1. $|a_n| = |a_4| = 0.08 < a_0 = 1 \Rightarrow$ First condition is satisfied.
2. $P(1) = 1 - 1.2 + 0.07 + 0.3 - 0.08 = 0.09 > 0 \Rightarrow$ Second condition is satisfied.
3. $P(-1) = 1 + 1.2 + 0.07 - 0.3 - 0.08 = 1.89 > 0 \Rightarrow$ Third condition is satisfied.

Jury Table

$$b_3 = \begin{vmatrix} a_n & a_0 \\ a_0 & a_n \end{vmatrix} = \quad 0.0064 - 1 = -0.9936$$

$$b_2 = \begin{vmatrix} a_n & a_1 \\ a_0 & a_3 \end{vmatrix} = \quad -0.08 \times 0.3 + 1.2 = 1.176$$

Rest of the elements are also calculated in a similar fashion. The elements are $b_1 = -0.0756$
$b_0 = -0.204 \quad c_2 = 0.946 \quad c_1 = -1.184 \quad c_0 = 0.315$. One can see
$|b_3| = 0.9936 > |b_0| = 0.204$
$|c_2| = 0.946 > |c_0| = 0.315$

All criteria are satisfied. Thus the system is stable.

UNIT – 2

DESIGN OF DIGITAL CONTROLLER

Digital PID – Position and velocity form – Deadbeat's algorithm – Dahlin's algorithm – Kalman's algorithm - Pole placement controller – Predictive controller.

Digital PID

A digital PID controller reads the sensor signal, normally from a thermocouple or RTD and connects the measurement to engineering units, such as degree Fahrenheit or Celsius, which is then displayed in a digital format.

Position PID

Position PID is the algorithm typically used to perform closed-loop motion control on a position feedback axis. PID stands for the central gains used in this mode: Proportional, Integral, and Differential. The Position PID provides very good control and is suitable for nearly all motion control systems with position feedback.

The Position PID works on a position feedback only and controls both position and velocity. In certain advanced applications, other control modes may be preferred, such as Position I-PD, Velocity PID, or Velocity I-PD.

Position PID Advantages

- Tracks position very well.
- Is very well understood by most people in the motion control industry.

Position PID Disadvantages

- May chatter or oscillate when following an irregular target, such as a step jump or a noisy reference signal.
- Tendency to overshoot final position on some systems.

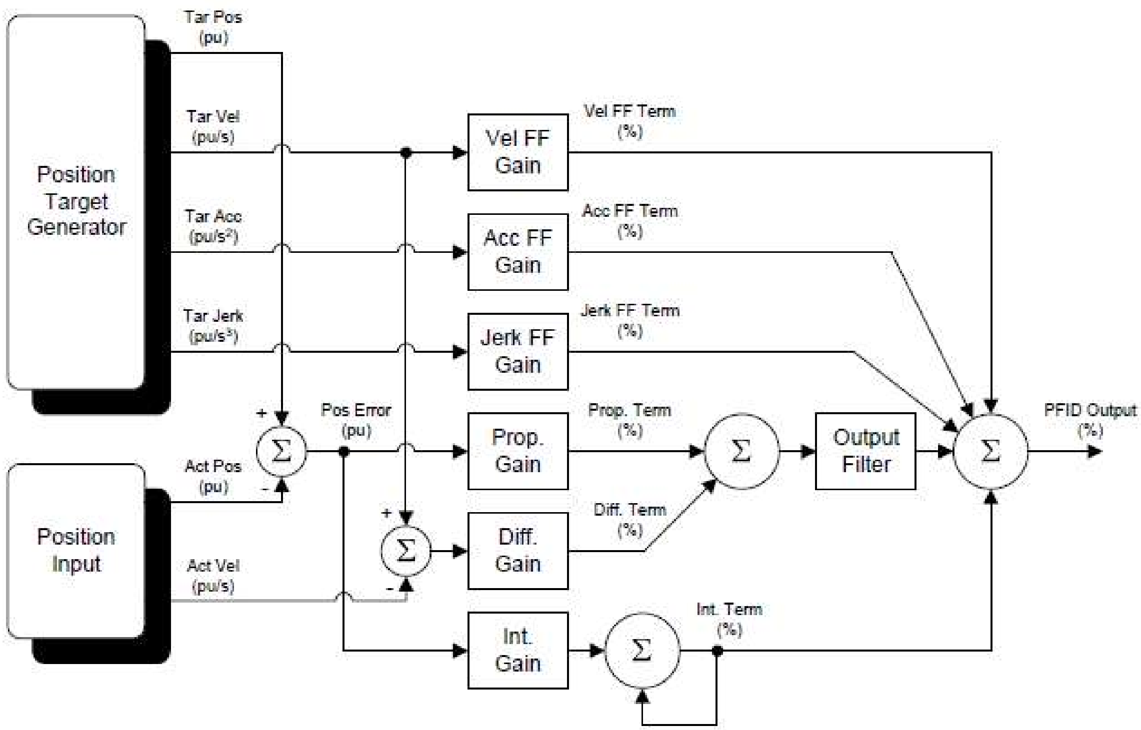
Algorithm

Each closed loop motion command issued to the RMC specifies a target profile, which defines where the axis should be at any given moment. For each loop time when the axis is in closed loop control, the RMC uses the specified target profile to calculate the desired position of the axis at that moment (called the Target Position) and subtracts the Actual Position to determine the Position Error. The Position PID algorithm then uses this information, together with the gains and feed forwards, to calculate how much Control Output should be generated to move the axis to the Target Position. The values of the gains and feed forwards must be set to achieve proper control. The process of setting the gains is called tuning and is done as part of the setup procedure.

The Position PID uses the gains and feed forwards listed below. Each gain or feed forward is multiplied by some quantity related to the Target Position and Actual Position to come up with a percentage. The resulting percentages are all summed and then multiplied by the maximum output (typically 10V), to come up with the Control Output voltage for that loop time.

- Proportional Gain
  The Proportional Gain is multiplied by the Position Error. This is the most important gain.
- Integral Gain
  The Integral Gain is multiplied by the accumulated Position Error. This helps the axis get into position over time.
- Differential Gain
  The Differential Gain is multiplied by the difference between the Target and Actual Velocities. This helps the axis keep up with quick changes in velocity.
- Velocity Feed Forward
  The Velocity Feed Forward is multiplied by the Target Velocity.
- Acceleration Feed Forward
  The Acceleration Feed Forward is multiplied by the Target Acceleration.
- Jerk Feed Forward
  The Jerk Feed Forward is multiplied by the Target Jerk. The Jerk Feed Forward is not necessary for most applications.

In addition, higher-order gains may be used if Acceleration Control or Active Damping is selected.



Position form PID controller

Velocity PID

Velocity PID is the algorithm typically used to perform closed-loop velocity control on a position or velocity axis. PID stands for the central gains used in this mode: Proportional, Integral, and Differential. The Velocity PID provides very good control and is suitable for nearly all motion control systems with velocity feedback. In certain cases, Velocity I-PD control may be preferred.

Velocity PID Advantages

- Excellent for controlling an axis that follows a smooth target, such as one generated by the RMC motion commands.

Velocity PID Disadvantages

- May not control very well with an irregular target, such as step jumps or a joystick.

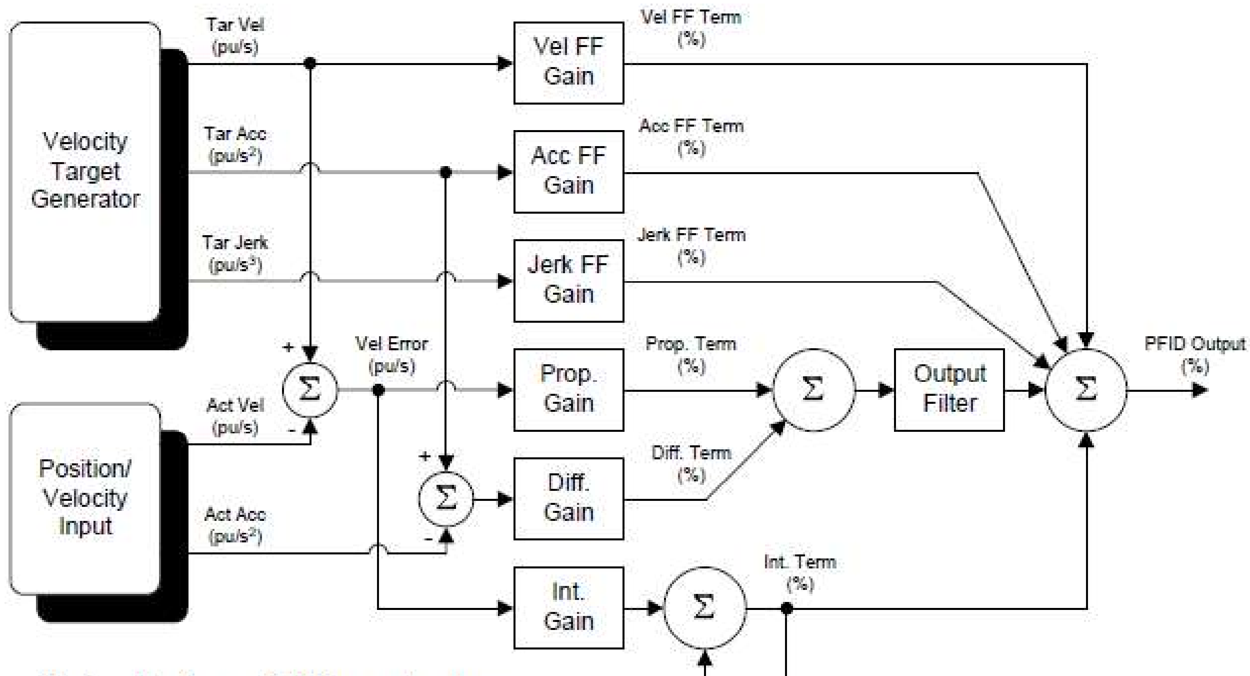
Algorithm

Each closed loop motion command issued to the RMC specifies a target profile, which defines where the axis should be at any given moment. For each loop time when the axis is in closed loop control, the RMC uses the specified target profile to calculate the desired velocity of the axis at that moment (called the Target Velocity) and subtracts the Actual Velocity to determine the Velocity Error. The Velocity PID algorithm then uses this information, together with the gains and feed forwards, to calculate how much Control Output should be generated to move the axis to the Target Velocity. The values of the gains and feed forwards must be set to achieve proper control. The process of setting the gains is called tuning and is done as part of the setup procedure.

The Velocity PID uses the gains and feed forwards listed below. Each gain or feed forward is multiplied by some quantity related to the Target Velocity and Actual Velocity to come up with a percentage. The resulting percentages are all summed and then multiplied by the maximum output (typically 10V), to come up with the Control Output voltage for that loop time.

- Proportional Gain
  The Proportional Gain is multiplied by the Velocity Error. This is the most important gain.
- Integral Gain
  The Integral Gain is multiplied by the integrated (sum of value x time) Velocity Error. This helps the axis get to velocity over time.
- Differential Gain
  The Differential Gain is multiplied by the difference between the Target and Actual Accelerations. This helps the axis keep up with quick changes in velocity.
- Velocity Feed Forward
  The Velocity Feed Forward is multiplied by the Target Velocity.
- Acceleration Feed Forward
  The Acceleration Feed Forward is multiplied by the Target Acceleration.
- Jerk Feed Forward
  The Jerk Feed Forward is multiplied by the Target Jerk. The Jerk Feed Forward is not necessary for most applications.

In addition, higher-order gains may be used if Acceleration Control or Active Damping is selected.



Velocity form PID controller

Deadbeat's algorithm:

An algorithm that requires the close-loop response to have finite settling time, minimum rise time and zero steady state error is referred as Deadbeat Algorithm. A specification that satisfies these criteria is

$$\frac{C(z)}{R(Z)} = Z^{-1}$$

This specification requires that the controlled variable shall reach the set point at time t = 1T, i.e. after a delay of one sampling period.

$$D(z) = \frac{1}{G(z)}\frac{z^{-1}}{1 - z^{-1}}$$

Design Criteria:

1. The system must have a zero steady state error at sampling instants.
2. The time to reach final output must be finite and minimum.
3. The controller should be physically realizable, i.e. it should be causal.

Example

$$G_p(z) = \frac{0.05(z+0.5)}{(z-0.9)(z-0.8)(z-0.3)}$$

When the input is a step function, $M(z) = z^{-2}$

$$\Rightarrow D_c(z) = \frac{1}{G_p(z)} \frac{M(z)}{1-M(z)}$$

$$= \frac{(z-0.9)(z-0.8)(z-0.3)}{0.05(z+0.5)} \frac{\frac{1}{z^2}}{1-\frac{1}{z^2}}$$

$$= \frac{20(z-0.9)(z-0.8)(z-0.3)}{(z+0.5)(z+1)(z-1)}$$

Thus

$$\begin{aligned} C(z) &= M(z)R(z) \\ &= \frac{1}{z^2}\frac{z}{z-1} = \frac{1}{z^2-z} \\ &= z^{-2} + z^{-3} + z^{-3} + \cdots \end{aligned}$$

Hence the output sequence follows the input after two sampling instants.

Dahlin's Algorithm:

Dahlin's Algorithm specifies that the closed –loop sampled data control system behave as though it were a first order process with dead time.

$$D(z) = \frac{1}{G(z)}\left[\frac{(1-\propto_f)z^{-(N+1)}}{1-\propto_f z^{-1} - (1-\propto_f)z^{-(N+1)}}\right]$$

The time constant of closed loop response t_f (or) equivalently, the parameter $\propto_f$ is an adjustable parameter that is selected by trial and error in the field.

Advantages and Disadvantages

- Dahlin's algorithm is essentially the same as the recently developed internal model control for first order system plus dead time types of processes.
- $\propto_f$ Can be used to enhance robustness of the loop in the presence of modeling errors.

- $\propto_f$= high value close to 1 given a high degree of robustness, but the response becomes quite sluggish.
- $\propto_f$= Smaller values equal to zero improve dynamic response, but the system becomes more sensitive to modeling errors.
- Even in the absence of modeling errors, the parameters $\propto_f$ can be used to shape the output response and to reduce ringing.
- It is unsuitable for the processes with inverse response due to the particular choice of $G_+(z) = Z^{-(N+1)}$ employed.

Example

Design a Dahlin's algorithm for $G_p(s) = \frac{e^{-1.4s}}{3.34s+1}$ T= 1 sec.

Solution:

Dahlin's Algorithm

$$D(z) = \frac{1}{G(z)}\left[\frac{(1-\propto_f)z^{-(N+1)}}{1-\propto_f z^{-1}-(1-\propto_f)z^{-(N+1)}}\right]$$

Where G(z) = Z{Gho(s). Gp(s)} = GhoGp(z)

$$G(z) = (1-z^{-1})Z\{\frac{e^{-1.4s}}{3.34s+1}\}$$

$$G(z) = [\frac{0.165z^{-2}+0.094z^{-3}}{1-0.741z^{-1}}]$$

$$D(z) = \frac{1}{G(z)}\left[\frac{(1-\propto_f)z^{-(N+1)}}{1-\propto_f z^{-1}-(1-\propto_f)z^{-(N+1)}}\right]$$

$$D(z) = \frac{0.6321z^{-1}-0.468z^{-2}}{0.094+0.1305z^{-1}-0.12z^{-2}-0.1043z^{-3}}$$

Kalman's Algorithm:

Kalman's filter is used in state and parameter estimation. To design a digital control algorithm by Kalman's approach one places restriction on C and M, instead of the usual C/R.

Design Procedure:

The control algorithm is to be designed for a unit step change in input, then

$$\frac{C(z)}{R(z)} = P(z)$$

And

$$\frac{M(z)}{R(z)} = Q(z)$$

Now

$$G(z) = \frac{P(z)}{Q(z)}$$

So the control algorithm can be derived as,

$$D(z) = \frac{1}{G(z)}[\frac{C(z)/R(z)}{1 - C(z)R(z)}]$$

$$D(z) = \frac{Q(z)}{1 - P(Z)}$$

<u>Pole Placement:</u>

Full state feedback (FSF), or **pole placement**, is a method employed in feedback control system theory to place the closed-loop poles of a plant in pre-determined locations in the s-plane. Placing poles is desirable because the location of the poles corresponds directly to the eigenvalues of the system, which control the characteristics of the response of the system. The system must be considered controllable in order to implement this method.

If the closed-loop dynamics can be represented by the state space equation

$$\dot{\underline{x}} = \mathbf{A}\underline{x} + \mathbf{B}\underline{u},$$

With output equation

$$\underline{y} = \mathbf{C}\underline{x} + \mathbf{D}\underline{u},$$

then the poles of the system transfer function are the roots of the characteristic equation given by

$$|s\mathbf{I} - \mathbf{A}| = 0.$$

Full state feedback is utilized by commanding the input vector u. Consider an input proportional (in the matrix sense) to the state vector,

$$\underline{u} = -\mathbf{K}\underline{x}.$$

Substituting into the state space equations above, we have

$$\dot{\underline{x}} = (\mathbf{A} - \mathbf{B}\mathbf{K})\underline{x}$$
$$\underline{y} = (\mathbf{C} - \mathbf{D}\mathbf{K})\underline{x}.$$

The poles of the FSF system are given by the characteristic equation of the matrix

$$\mathbf{A} - \mathbf{B}\mathbf{K}, \det\left[s\mathbf{I} - (\mathbf{A} - \mathbf{B}\mathbf{K})\right] = 0.$$

Comparing the terms of this equation with those of the desired characteristic equation yields the values of the feedback matrix **K** which force the closed-loop eigenvalues to the pole locations specified by the desired characteristic equation.

Predictive Controller:

Model predictive control (MPC) is an advanced method of process control that is used to control a process while satisfying a set of constraints. It has been in use in the process industries in chemical plants and oil refineries since the 1980s. In recent years it has also been used in power system balancing models and in power electronics. Model predictive controllers rely on dynamic models of the process, most often linear empirical models obtained by system identification. The main advantage of MPC is the fact that it allows the current timeslot to be optimized, while keeping future timeslots in account. This is achieved by optimizing a finite time-horizon, but only implementing the current timeslot and then optimizing again, repeatedly, thus differing from Linear-Quadratic Regulator (LQR). Also MPC has the ability to anticipate future events and can take control actions accordingly. PID controllers do not have this predictive ability. MPC is nearly universally implemented as a digital control, although there is research into achieving faster response times with specially designed analog circuitry.

MPC is based on iterative, finite-horizon optimization of a plant model. At time the current plant state is sampled and a cost minimizing control strategy is computed (via a numerical minimization algorithm) for a relatively short time horizon in the future: [t, t+T]. Specifically, an online or on-the-fly calculation is used to explore state trajectories that emanate from the current state and find (via the solution of Euler–Lagrange equations) a cost-minimizing control strategy until time [t+T]. Only the first step of the control strategy is implemented, then the plant state is sampled again and the calculations are repeated starting from the new current state, yielding a new control and new predicted state path. The prediction horizon keeps being shifted forward and for this reason MPC is also called receding horizon control. Although this approach is not optimal, in practice it has given very good results. Much academic research has been done to find fast methods of solution of Euler–Lagrange type equations, to understand the global stability properties of MPC's local optimization, and in general to improve the MPC method.

Principles of MPC

Model Predictive Control (MPC) are a multivariable control algorithm that uses:

- an internal dynamic model of the process
- a cost function *J* over the receding horizon
- an optimization algorithm minimizing the cost function *J* using the control input *u*

An example of a quadratic cost function for optimization is given by:

$$J = \sum_{i=1}^{N} w_{x_i} (r_i - x_i)^2 + \sum_{i=1}^{N} w_{u_i} \Delta {u_i}^2$$

without violating constraints (low/high limits) with

x_i: i^{th} controlled variable (e.g. measured temperature)
r_i: i^{th} reference variable (e.g. required temperature)
u_i: i^{th} manipulated variable (e.g. control valve)
w_{x_i}: weighting coefficient reflecting the relative importance of x_i
w_{u_i}: weighting coefficient penalizing relative big changes in u_i

etc.

Nonlinear MPC

Nonlinear Model Predictive Control, or NMPC, is a variant of model predictive control (MPC) that is characterized by the use of nonlinear system models in the prediction. As in linear MPC, NMPC requires the iterative solution of optimal control problems on a finite prediction horizon. While these problems are convex in linear MPC, in nonlinear MPC they are not necessarily convex anymore. This poses challenges for both NMPC stability theory and numerical solution.

The numerical solution of the NMPC optimal control problems is typically based on direct optimal control methods using Newton-type optimization schemes, in one of the variants: direct single shooting, direct multiple shooting methods, or direct collocation. NMPC algorithms typically exploit the fact that consecutive optimal control problems are similar to each other. This allows initializing the Newton-type solution procedure efficiently by a suitably shifted guess from the previously computed optimal solution, saving considerable amounts of computation time. The similarity of subsequent problems is even further exploited by path following algorithms (or "real-time iterations") that never attempt to iterate any optimization problem to convergence, but instead only take a few iterations towards the solution of the most current NMPC problem, before proceeding to the next one, which is suitably initialized; see, e.g.

While NMPC applications have in the past been mostly used in the process and chemical industries with comparatively slow sampling rates, NMPC is being increasingly applied, with advancements in controller hardware and computational algorithms, e.g., preconditioning, to applications with high sampling rates, e.g., in the automotive industry, or even when the states are distributed in space (Distributed parameter systems).As an application in aerospace, recently, NMPC has been used to track optimal terrain-following/avoidance trajectories in real-time.

Explicit MPC

Explicit MPC (eMPC) allows fast evaluation of the control law for some systems, in stark contrast to the online MPC. Explicit MPC is based on the parametric programming technique, where the solution to the MPC control problem formulated as optimization problem is pre-computed offline. This offline solution, i.e., the control law, is often in the form of a piecewise affine function (PWA), hence the eMPC controller stores the coefficients of the PWA for each a subset (control region) of the state space, where the PWA is constant, as well as coefficients of some parametric representations of all the regions. Every region turns out to geometrically be a convex polytope for linear MPC, commonly parameterized by coefficients for its faces, requiring quantization accuracy analysis. Obtaining the optimal control action is then reduced to first determining the region containing the current state and seconds a mere evaluation of PWA using the PWA coefficients stored for all regions. If the total number of the regions is small, the implementation of the eMPC does not require significant computational resources (compared to the online MPC) and is uniquely suited to control systems with fast dynamics. A serious drawback of eMPC is exponential growth of the total number of the control regions with respect to some key parameters of the controlled system, e.g., the number of states, thus dramatically increasing controller memory requirements and making the first step of PWA evaluation, i.e. searching for the current control region, computationally expensive.

Robust MPC

Robust variants of Model Predictive Control (MPC) are able to account for set bounded disturbance while still ensuring state constraints are met. There are three main approaches to robust MPC:

- Min-max MPC. In this formulation, the optimization is performed with respect to all possible evolutions of the disturbance. This is the optimal solution to linear robust control problems; however it carries a high computational cost.
- Constraint Tightening MPC. Here the state constraints are enlarged by a given margin so that a trajectory can be guaranteed to be found under any evolution of disturbance.
- Tube MPC. This uses an independent nominal model of the system, and uses a feedback controller to ensure the actual state converges to the nominal state. The amount of separation required from the state constraints is determined by the robust positively invariant (RPI) set, which is the set of all possible state deviations that may be introduced by disturbance with the feedback controller.
- Multi-stage MPC. This uses a scenario-tree formulation by approximating the uncertainty space with a set of samples and the approach is non-conservative because it takes into account that the measurement information is available at every time stages in the prediction and the decisions at every stage can be different and can act as recourse to counteract the effects of uncertainties. The drawback of the approach however is that the size of the problem grows exponentially with the number of uncertainties and the prediction horizon.

UNIT-3

COMPUTER AS CONTROLLER

Basic building blocks of computer control system – Data acquisition systems – SCADA, Direct digital control – Introduction to AI and expert control system – Case study - Design of computerized multi loop controller.

Need of computer in a control system

In the modern cities, many facilities and works are controlled by the automation system and computers, for example, MTR, electricity supply system, traffic light control system, elevators and CNC machines in factories. There are many advantages of automation system such as the increase in efficiency; reduction of the cost, the number of operators, repetitive and boring work; enhancing the safety of workers; improvement of the working performance and completing the work that cannot be done manually.

The merits of computer control system

Computer is not the only ways to control the automation system, other methods include: mechanical systems, electrical (relay) systems, pneumatic system, electronic system, etc. Yet computer control system has many advantages over the other control systems. For example, it has fast calculation speed, multiple forms of input and output devices, large memory, programmable control, telecommunication possible, small and light, etc.

Fast calculation speed

The central processing unit of modern computers can do a large number of calculations within one second, and it can manage a lot of work and data in a short time. The fast calculation speeds of computer make it possible to control a lot of facilities at the same time under different conditions. For example, the MTR computer system can control trains in different routes to ensure the system runs normally.

Multiple forms of input and output devices

Computer has many input devices, including keyboard, mouse, scanner, tape, etc. Moreover, it can also use electronic circuits to transform signals from electronic sensors to digital data and input them to the computer. For example, by using a sensor, the temperature can be transformed into digital data and input in the computer.

After processing the data, the computer can generate output signal. Similarly, computer can use electronic circuits to output digital signals which can then control various output devices, for examples, printer, monitor, relay, motor, electromagnetic detecting valve, etc.

High capacity of data storage

Computer can use a large number of information storage devices, such as floppy disk, magnetic tape, hard disk, CD-ROM, DVD, etc. Therefore, computer can use those stored information to perform the controlling work. For example, we can use computer to design complicated shape of work pieces. Then we input the relevant data into the storage device of the CNC machine and let the computer to control the cutting work of the work pieces according to the data.

Programmable control

A program is a set of instructions. Computer can operate according to the program. Therefore, operators can either input new program or modify the existing program to change the working procedures or methods according to the needs. So, computer control system not only can operate those simple task and repetitive work, it can also operate complicated work under different conditions and feedback according to the program. For example, there is a visual system in the computer controlled six legs machine. The computer of the robot can analyze the image and choose the way without any obstacle.

Telecommunications possible

Computers can be used to input or output digital signals, and using the wires to transmit signals, exchange information and communicate. So, computer information can be transmitted to distant areas by private cables, telephone lines, internet or radio wave. For example, computers can control distant video cameras or machines through the internet; computers can also be used to control spacecrafts in space through the use of radio wave.

Small and light

With the advancement of technology, the size and weight of computers have been reduced a lot. The microcomputers developed recently are so sophisticated that it is possible to put the microcomputers into a number of utensils, such as washing machines, refrigerators, air-conditioners, car etc. Take the car as an example. For those cars having traditional emulsifier engine, the amount of petrol supply to the engine is controlled by the emulsifier. But nowadays, some of the car has an EFI system that can decide the best amount of petrol supply to the engine according to the amount of air input, air temperature, spinning speed of the engine, coolant's temperature, valves' open width, oxygen concentration in the air vent pipe and other important factors.

The restrictions of computer control system

There are many advantages of the computer control system, but there are some restrictions as well. For example, the cost is high, the maintenance is difficult and the control is complicated. It is difficult to withstand adverse environment and it needs a large amount of accessory devices.

High cost

To design and make a computer control system takes a long period of time. The equipment is expensive and hence at the total cost is high. For example, when a computer is used to control a cutting machine, a suitable set of control program is needed to control the moving path of the tools, moving speed, spinning speed, feeding rate, change of tools, adding of coolants, etc.

Difficulties in maintenance

Nowadays, computers are used in mass production of integrated circuit and electronic circuit board. They need new parts to replace those damaged. With the rapid development of computer technology, the newly designed parts may not be used in those old-fashioned electronic circuit boards, and so the maintenance will be more difficult. For example, when newly designed random access memory (RAM) is produced, the production of the old designed RAM will cease. After some time, when the old designed RAM is damaged, it cannot be replaced.

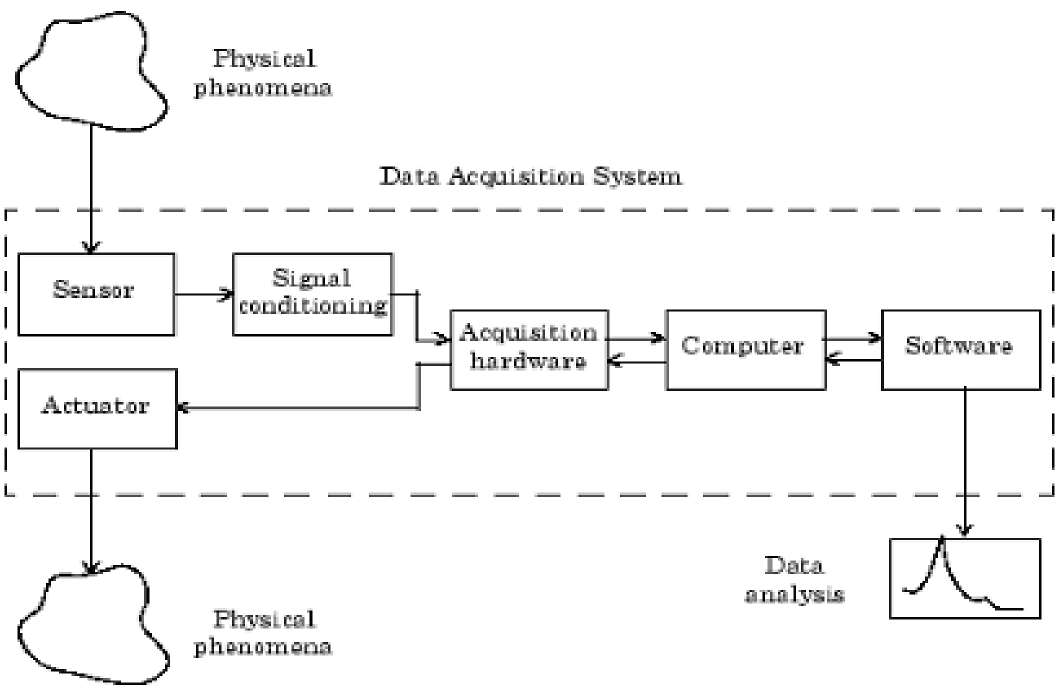
Intolerance under adverse environment

The electronic parts of the computer control system cannot work under bad environment, for example, too hot, too cold, humid, dusty, vigorous vibration etc. So, we should avoid using compute control system under these environments. For example, we should avoid using cooling fan to cool down the computer in dusty and hot place.

Requirement of large amount of supporting device

Computers can manage large amount of data in a short time, but it needs some supporting devices to provide information. That is why a computer control system needs many supporting devices. For example, a system is needed to detect the position of the tools and work pieces when a computer is used to control a cutting machine. The system can give feedback to the computer to allow it to control the machine accordingly.

Functional block diagram of a computer control system



Data loggers

A data logger (also data logger or data recorder) is an electronic device that records data over time or in relation to location either with a built in instrument or sensor or via external instruments and sensors. Increasingly, but not entirely, they are based on a digital processor (or computer).

Data loggers are normally more economical than chart recorders. They offer more flexibility and are available with a greater variety of input types. Most data loggers collect data which may be directly transferred to a computer. Although this option is available with some recorders, it normally adds significant expense to the recorder price.

Data acquisition systems offer a great deal of flexibility and are certainly attractive when high sample rates are required, however, since they require connection or installation into a computer, the computer must also be present and active when collecting the data. Data loggers can collect data independently of a computer. Data is normally collected in non-volatile memory for later download to a computer. The computer does not need to be present during the data collection process. This makes them ideally suited for applications requiring portability.

Wireless Data Loggers

Wireless data logger that measures and records ambient pressure, humidity and temperature. This type of dynamic data logger is ideal for monitoring weather data, measuring barometric pressure in high-rise buildings or monitoring storage or warehouse environments that contain sensitive material

Miniature Single Input Data Loggers

Miniature single input data loggers are generally low cost loggers dedicated to a specific input type. These types of data loggers are often used in the transportation industry. A typical application would be to include a temperature data logger in a shipment of food products to insure that the food temperature does not exceed acceptable limits. In addition to temperature miniature data loggers are available for a large variety of input types.

Fixed Mount Multi-Channel Data Loggers

Fixed input loggers have a fixed number of input channels which are generally dedicated for a specific type of input. OMEGA offers fixed input data loggers ranging from one to 8 channels

Handheld Multi-Channel Data Loggers

Handheld multi-channel loggers are commonly used in applications where the data logger is to be carried from one location to another. They are also commonly used in bench top or laboratory environments. In addition to storing data internally some models even contain on board printers which can produce an immediate hardcopy of the data.

Modular Data Loggers

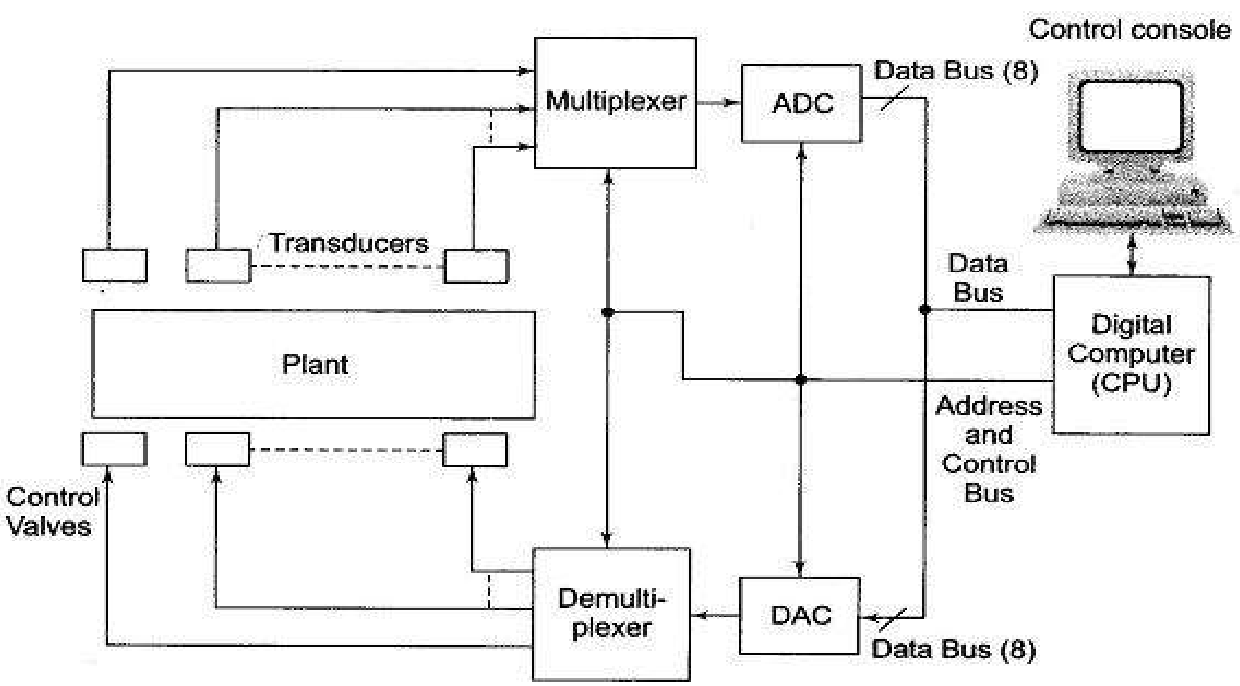
A modular data logger is configurable and expandable through the use of plug-in modules. The modules are normally field configurable and the user has the option of adding as many channels to satisfy the application requirement.

Direct Digital Control (DDC)

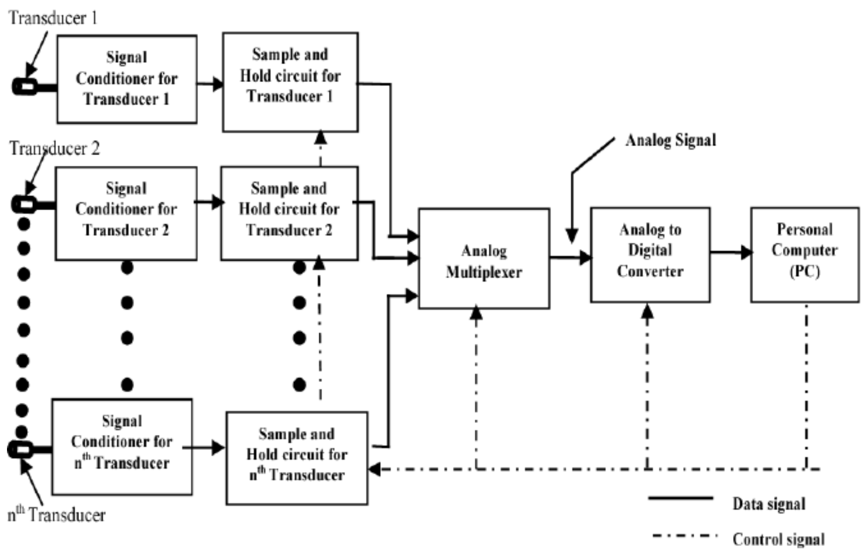
Developments in digital electronics have led to many industrial processes being computer controlled. The first system used was Direct Digital Control (DDC), in which a computer measured each variable in the process, these signals being used to maintain the required set points in the process.

As digital electronics developed, minicomputers become more competitive, and with the advent of electronic Multiplexer (Mux) and Demultiplexer (Demux), it is possible to sample the state of many analog and digital transducers at high speed and also to control many values or devices in the plant.

A basic Direct Digital Control System is shown in Fig. In this system a large number of transducers are sited around the plant, each transducer being connected to one input of a Mux. A Mux can be considered as- the electronic equivalent of a switch with a contact or blade which rotates very rapidly so that it moves from one transducer to another, the blade remaining in contact with the transducer long enough for an ADC to sample and digitise or to quantise the analog signal. The quantised data are then transmitted along the data bus of the system to the CPU.



Direct Digital Control of an Industrial Process



When the CPU has analysed the data from one or more transducers and has compared them with the appropriate set points in the computer program, it sends signals along the data bus to the values controlling the system as follows. The digital signal produced by the CPU is converted into an analog signal by a DAC and the analog signal is transmitted to the appropriate control value through a demux. As this is being performed, data are displayed on the operator's visual display unit and if necessary, he can remotely change the set points associated with various sections of the process.

DDC SOFTWARE:

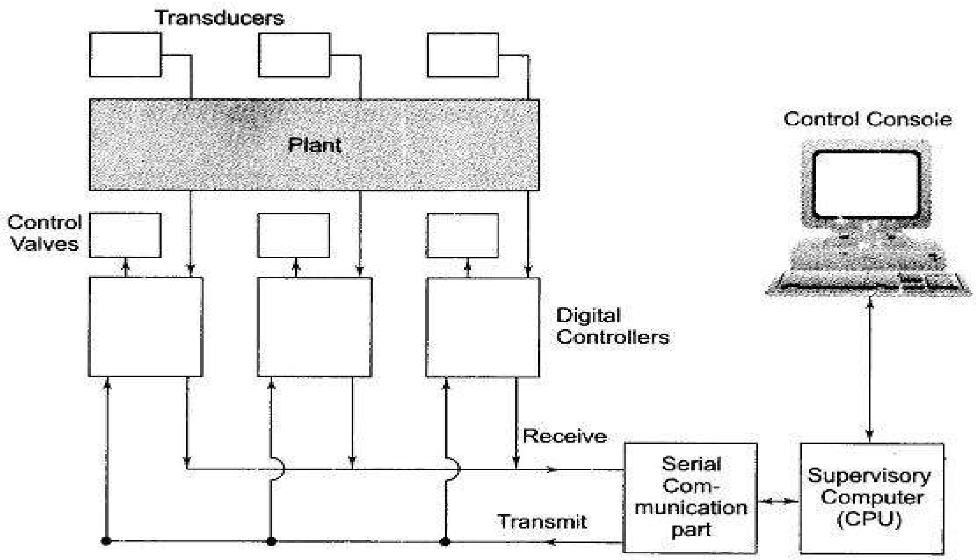
There are two algorithms for programming a three mode PID control loop

- Position Algorithm
- Velocity Algorithm

Computer Supervisory Control System

An alternative control system known as Computer Supervisory Control (CSC) is used in microprocessor based systems. In this case, the process is controlled by a number of local feedback loops using individual process controllers, the main computer merely acting in a supervisory role in which it monitors the measured variable (although it has the capability of remotely controlling set points of the controllers). This method provides the local operator with his own controller.

A basic Computer Supervisory Control system is shown in Fig. Each digital control controls one section of the plant by means of a closed loop involving its own sensors and control values. Each controller can either be operated manually to give the local operator complete control of his own section of the plant or it can be remotely controlled by the computer operator.



Computer Supervisory Control (CSC) of an Industrial Process

The computer is connected to each controller by means of a communication port which allows the computer either to receive information from the process controllers or to transmit data to the controllers to change, say, a set point. To minimise the cost of cabling, the communication port operates in a serial mode, that is, the data are sent along the communicating link in the form of pulses (which requires only two wires — a "send" wire and a "receive" wire). This method is clearly more economical than using a 8-bit data bus throughout the plant, but it is slower in operation, since it takes more time to send eight individual bits along one wire then it takes to send 8-bits simultaneously along eight wires. The digital process controller (also called the central computer) must contain a special interface which converts the pulses it receives in a serial mode along the supervisory serial data bus into the parallel mode used in microprocessor chip itself. Such an interface is known as a UART (Universal Asynchronous Receiver/Transmitter).

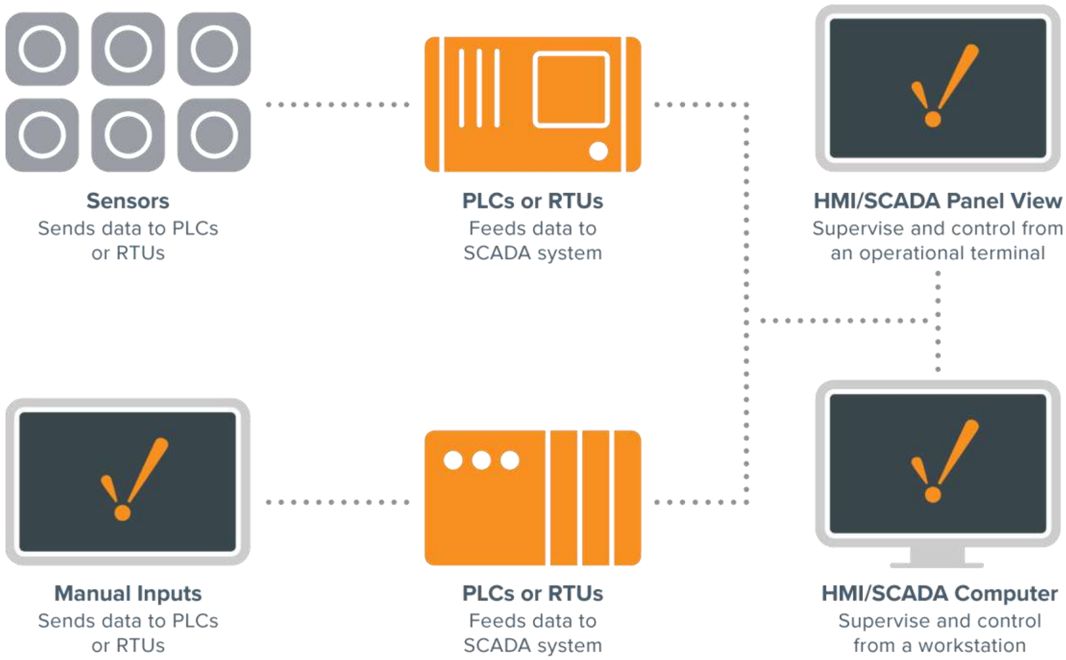
Supervisory Control and Data Acquisition - SCADA System

SCADA stands for Supervisory Control and Data Acquisition. It is a type of software application program for process control. SCADA is a central control system which consists of controllers, network interfaces, input/output, communication equipment, and software. SCADA systems are used to monitor and control the equipment in the industrial process which includes manufacturing, production, development, and fabrication. The infrastructural processes include gas and oil distribution, electrical power, water distribution. Public utilities include bus traffic system, airport and Metro systems. The SCADA system takes the reading of the meters and checks the status of sensors in regular intervals so that it requires minimal interference of humans.

A large number of processes occur in large industrial establishments. Every process you need to monitor is very complex because each machine gives a different output. The SCADA system used to gather data from sensors and instruments located in remote areas. The computer then processes this data and presents it promptly. The SCADA system gathers the information (like a leak on a pipeline occurred) and transfers the information back to the system while giving the alerts that leakage has occurred and displays the information in a logical and organized fashion. The SCADA system used to run on DOS and UNIX operating systems.

Generally, the SCADA system is a centralized system that monitors and controls the entire area. It is a pure software package that is positioned on top of the hardware. A supervisory system gathers data on the process and sends the commands control to the process. The SCADA is a remote terminal unit which is also known as RTU. Most control actions are automatically performed by RTUs or PLCs. The RTUs consists of the programmable logic converter which can be set to specific requirement. For example, in the thermal power plant, the water flow can be set to a specific value or it can be changed according to the requirement.

The SCADA system allows operators to change the set point for the flow, and enable alarm conditions in case of loss of flow and high temperature, and the condition is displayed and recorded. The SCADA system monitors the overall performance of the loop. The SCADA system is a centralized system to communicate with both wired and wireless technology to Clint devices. The SCADA system controls can run completely all kinds of the industrial process.

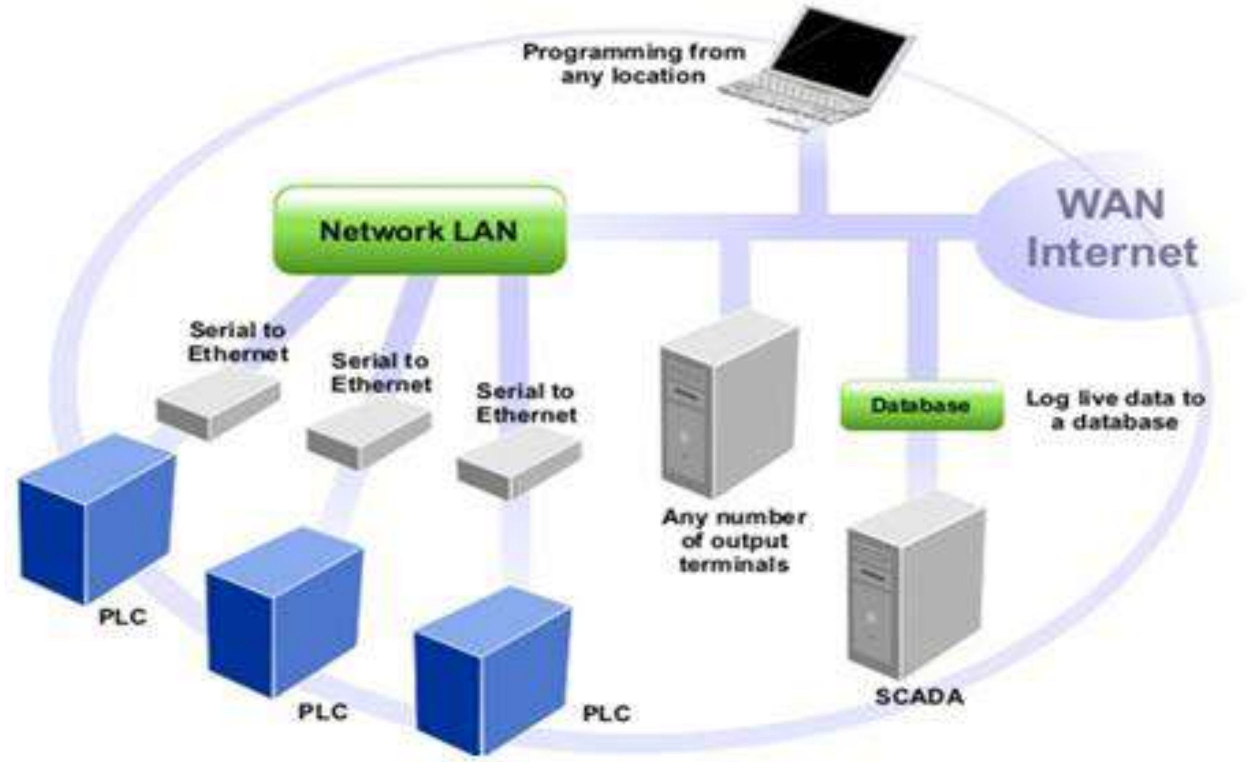


Hardware Architecture:

The generally SCADA system can be classified into two parts:

- The Clint layer which caters for the man-machine interaction.
- The data server layer which handles most of the process data activities.

The SCADA station refers to the servers and it is composed of a single PC. The data servers communicate with devices in the field through process controllers like PLCs or RTUs. The PLCs are connected to the data servers either directly or via networks or buses. The SCADA system utilizes a WAN and LAN networks, the WAN and LAN consist of internet protocols used for communication between the master station and devices. The physical equipment like sensors connected to the PLCs or RTUs. The RTUs convert the sensor signals to digital data and sends digital data to the master. According to the master feedback received by the RTU, it applies the electrical signal to relays. Most of the monitoring and control operations are performed by RTUs or PLCs as we can see in the figure.



Software Architecture

- Most of the servers are used for multitasking and real-time database. The servers are responsible for data gathering and handling. The SCADA system consists of a software program to provide trending, diagnostic data, and manage information such as scheduled maintenance procedures, logistic information, detailed schematics for a particular sensor or machine, and expert-system troubleshooting guides. This means the operator can see a schematic representation of the plant being controlled.
- **EX:** alarm checking, calculations, logging, and archiving; polling controllers on a set of parameters, those are typically connected to the server.

Working Procedure of SCADA Software system

The SCADA system performs the following functions:

- Data Acquisitions

- Data Communication
- Information/Data presentation
- Monitoring/Control
- These functions are performed by sensors, RTUs, controller, a communication network. The sensors are used to collect the important information and RTUs are used to send this information to the controller and display the status of the system. According to the status of the system, the user can give the command to other system components. This operation is done by the communication network.

Data Acquisitions:

- The real-time system consists of thousands of components and sensors. It is very important to know the status of particular components and sensors. For example, some sensors measure the water flow from the reservoir to the water tank and some sensors measure the value pressure as the water is a release from the reservoir.

Data Communication:

- The SCADA system uses a wired network to communicate between users and devices. Real-time applications use a lot of sensors and components which should be controlled remotely. The SCADA system uses internet communications. All information is transmitted through the internet using specific protocols. Sensor and relays are not able to communicate with the network protocols so RTUs used to communicate sensors and network interfaces.

Information/Data presentation:

- The normal circuit networks have some indicators which can be visible to control but in the real-time SCADA system, there are thousands of sensors and alarm which are impossible to be handled simultaneously. The SCADA system uses the human-machine interface (HMI) to provide all of the information gathered from the various sensors.

Human-machine interface:

- **The** SCADA system uses the human-machine interface. The information is displayed and monitored to be processed by a human. HMI provides access to multiple control units which can be PLCs and RTUs. The HMI provides the graphical presentation of the system. For example, it provides a graphical picture of the pump connected to the tank. The user can see the flow of the water and the pressure of the water. The important part of the HMI is an alarm system that is activated according to the predefined values.

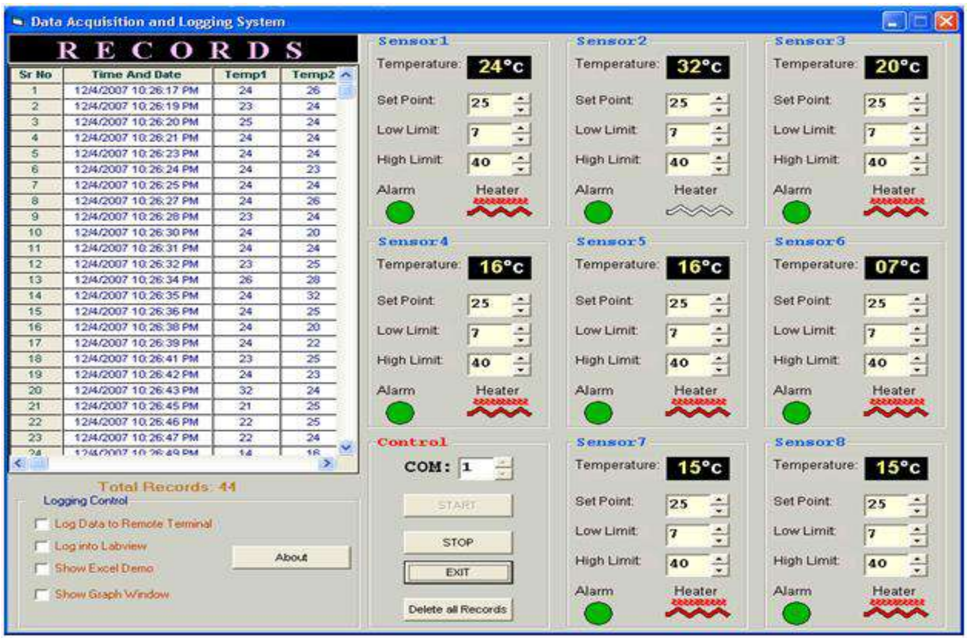
Monitoring/Control:

- The SCADA system uses different switches to operate each device and displays the status of the control area. Any part of the process can be turned ON/OFF from the control station using these switches. SCADA system is implemented to work automatically without human intervention but in critical situations, it is handled by manpower.

SCADA for Remote Industrial plant:

- In large industrial establishments, many processes occur simultaneously and each needs to be monitored, which is a complex task. The SCADA systems are used to monitor and control the equipment in the industrial processes which include water distribution, oil distribution, and power distribution. The main aim of this project is to process the real-time data and control the large scale remote industrial environment. In the real-time scenario, a temperature logging system for a remote plant operation is taken.
- The temperature sensors are connected to the microcontroller, which is connected to the PC at the front end, and software is loaded on the computer. The data is collected from the temperature sensors. The temperature sensors continuously send the signal to the microcontroller which accordingly displays these values on its front panel. One can set the parameters like low limit and high limit on the computer screen. When the temperature of a sensor goes above-set point the microcontroller sends a command to the corresponding relay. The heaters connected through relay contacts are turned OFF and ON.
- This is a temperature logging system. Here 8 temperature sensors in multiplexing mode are connected to the microcontroller through ADC 0808. Then the values of all the sensors are sent serially by the microcontroller through Max 32 to the com port of the PC. Software “DAQ System” loaded on the PC takes these values and shows them on its front panel, and also logs them to the database “daq.mdb”. One can set by the interactive way some parameters like a set point, low limit, and high limit on the computer screen. When the

temperature of some sensor increases beyond the set point, the microcontroller sends commands to relay driver IC. The heaters connected through relay contacts are (specific for that sensor) turned OFF (or ON in opposite case).High limit and low limits are for alarm. When the temperature goes above the high limit or below low limit the alarm will be turned on.



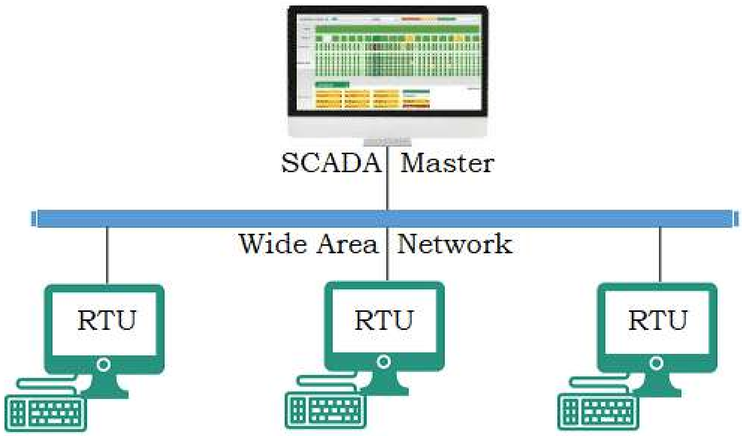
Types of SCADA System

There are four different types of SCADA systems from four generations. They are:

1. Early or Monolithic SCADA Systems (First Generation)
2. Distributed SCADA Systems (Second Generation)
3. Networked SCADA Systems (Third Generation)
4. IoT SCADA Systems (Fourth Generation)

1. Early or Monolithic SCADA Systems (First Generation)

Minicomputers were used in the earlier Supervisory Control and Data Acquisition systems. Monolithic systems were developed during times when ordinary network services were unavailable. These were designed to be independent systems without any connection to other systems.

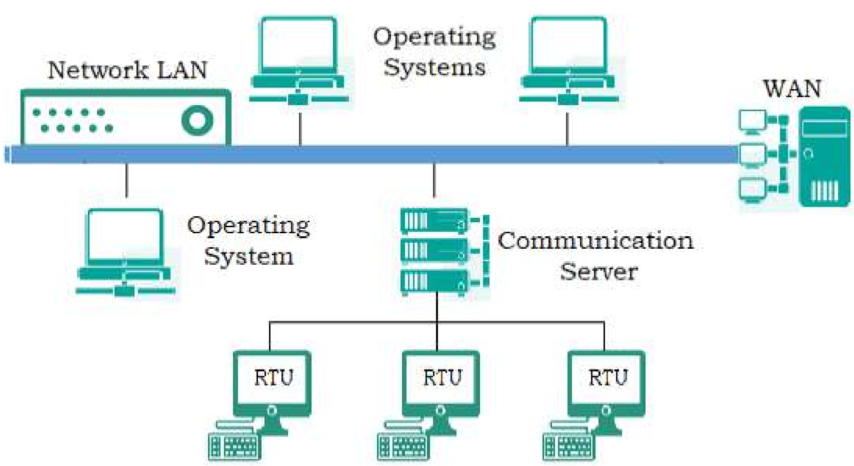


Monolithic SCADA System

Back up mainframe was used to gather data from all remote terminal units. Functions of these early systems were limited to flagging of operations in case of emergency and monitoring the sensors.

2. Distributed SCADA Systems (Second Generation)

Here, the control function was distributed across several systems that were connected using LAN. Command processing and real-time data were shared to perform control operations.

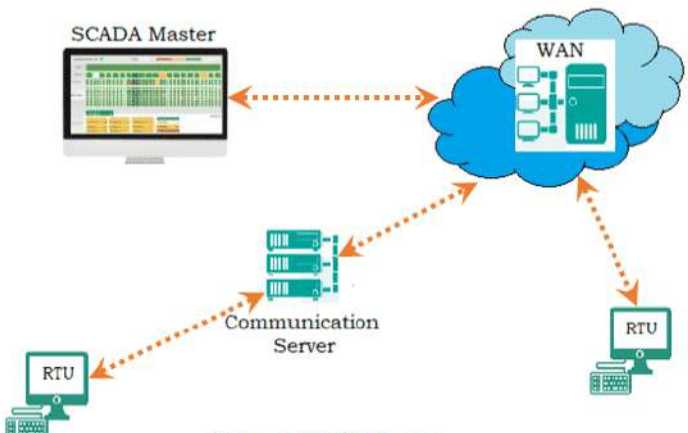


Distributed SCADA Systems

The second generation resulted in the reduction of size and cost of each station but there were no standardized network protocols. Since the protocols were proprietary, very few people understood the security of Supervisory Control and Data Acquisition system installation and this factor was largely ignored.

3. Networked SCADA Systems (Third Generation)

Present Supervisory Control and Data Acquisition systems are networked and communicate over WAN system through phone or data lines. Fiber optic connections or Ethernet is used for data transmission between the nodes.

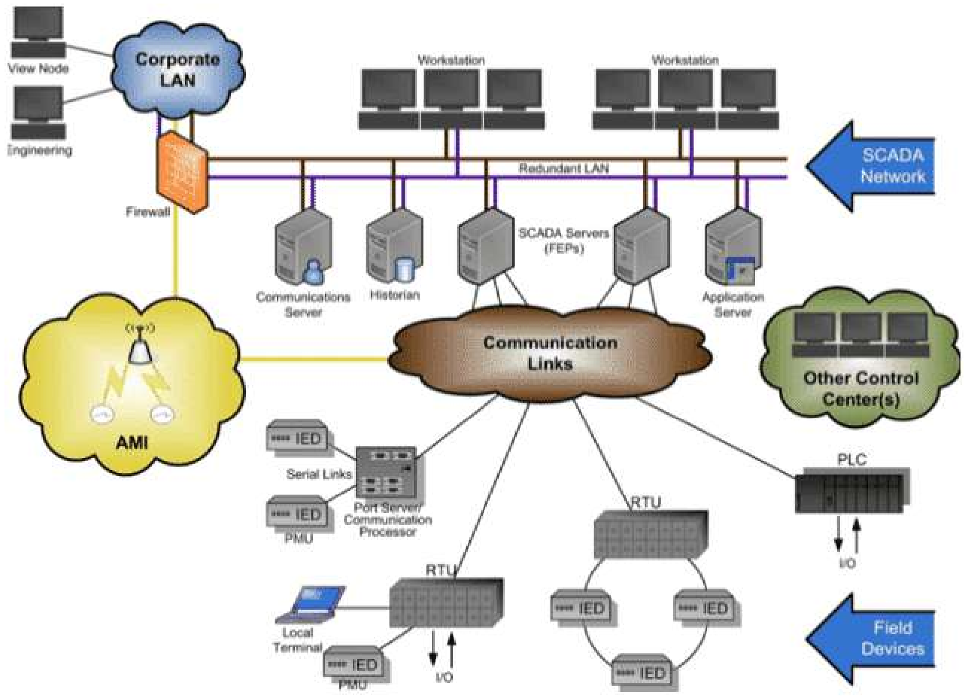


Network SCADA System

These systems use PLC for adjusting and monitoring the flagging operations only when there is a requirement for major decisions.

4. Internet of Things SCADA Systems (Fourth Generation)

The fourth generation is seeing a reduction in infrastructural cost of these systems by adopting IoT with cloud computing. Integration and maintenance is also made very easy in the fourth generation system.



IoT SCADA Systems

These systems can report the state in real time using cloud computing. Thus intricate control algorithms can be implemented that are often used on traditional PLCs.

Applications of SCADA System

Supervisory Control and Data Acquisition systems are mainly used to monitor a wide data variety like currents, voltages, temperature, pressure, water levels etc. in several industries. If any abnormal conditions are detected, alarms at remote or central sites are triggered for operator alert. The various applications of SCADA Systems include:

1. **Power Generation & Distribution**: Used to monitor current flow, voltage, circuit breaker functions. Also used in remotely switching on/ off of power grids.
2. **Water & Sewage System**: Used by municipal corporations for regulating and monitoring water flow, reservoir status, pressure in distribution pipes, etc.
3. **Industries and Buildings**: Used to control HVAC, central air conditioning, lighting, entry/ exit gates, etc.
4. **Oil and Gas Industries**: Used for regulating and monitoring flow, reservoir status, pressure in distribution pipes, etc.

5. **Communication Networks**: Used for monitoring and controlling servers, networks and nodes.
6. **Manufacturing**: Used for managing inventories for controlling over manufacturing/ stocking. Also used for monitoring and regulating instrumentation, process and product quality.
7. **Public Transport**: Used for regulating subway electricity, automating traffic signals/ railway crossing and live tracking of flights/ trains/ buses.

Advantages of SCADA System

The advantages of Supervisory Control and Data Acquisition system include:

- Improvement in Service Quality
- Improvement in Reliability
- Reduction in operation and maintenance costs
- Easy to monitor large system parameters
- Real time information on demand
- Reduction in Manpower
- Value added services
- Ease in Fault Detection and Fault Localization (FDFL)
- Reduction in Repair Time (System Down Time)

Artificial Intelligence

Artificial Intelligence is composed of two words **Artificial** and **Intelligence**, where Artificial defines "man-made," and intelligence defines "thinking power", hence AI means "a man-made thinking power."

Artificial Intelligence is not just a part of computer science even it's so vast and requires lots of other factors which can contribute to it. To create the AI first we should know that how intelligence is composed, so the Intelligence is an intangible part of our brain which is a combination of **Reasoning, learning, problem-solving perception, language understanding, etc**.

Advantages of Artificial Intelligence

- **High Accuracy with less error:** AI machines or systems are prone to less errors and high accuracy as it takes decisions as per pre-experience or information.
- **High-Speed:** AI systems can be of very high-speed and fast-decision making, because of that AI systems can beat a chess champion in the Chess game.
- **High reliability:** AI machines are highly reliable and can perform the same action multiple times with high accuracy.
- **Useful for risky areas:** AI machines can be helpful in situations such as defusing a bomb, exploring the ocean floor, where to employ a human can be risky.
- **Digital Assistant:** AI can be very useful to provide digital assistant to the users such as AI technology is currently used by various E-commerce websites to show the products as per customer requirement.
- **Useful as a public utility:** AI can be very useful for public utilities such as a self-driving car which can make our journey safer and hassle-free, facial recognition for security purpose, Natural language processing to communicate with the human in human-language, etc.

Disadvantages of Artificial Intelligence

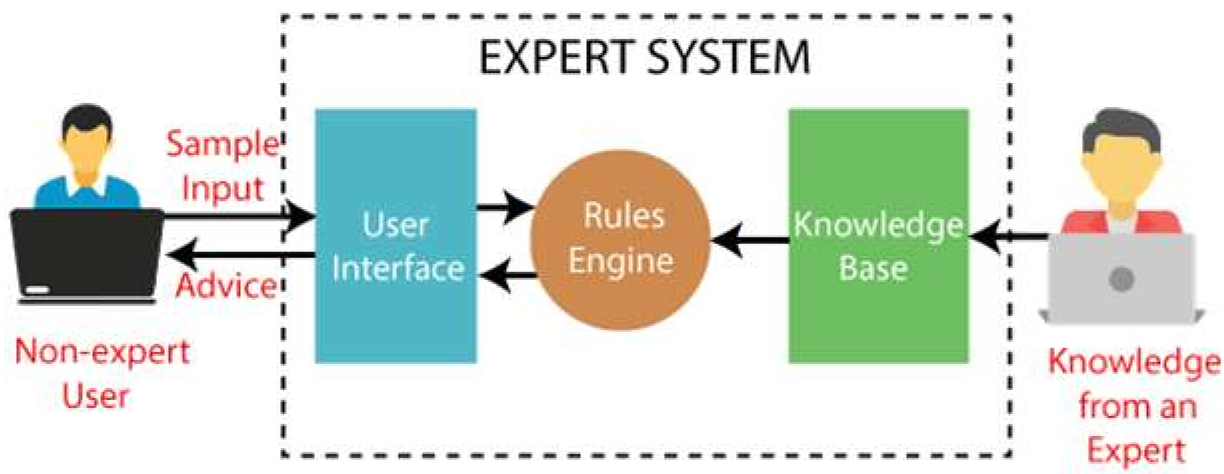
- **High Cost:** The hardware and software requirement of AI is very costly as it requires lots of maintenance to meet current world requirements.
- **Can't think out of the box:** Even we are making smarter machines with AI, but still they cannot work out of the box, as the robot will only do that work for which they are trained, or programmed.
- **No feelings and emotions:** AI machines can be an outstanding performer, but still it does not have the feeling so it cannot make any kind of emotional attachment with human, and may sometime be harmful for users if the proper care is not taken.
- **Increase dependency on machines:** With the increment of technology, people are getting more dependent on devices and hence they are losing their mental capabilities.
- **No Original Creativity:** As humans are so creative and can imagine some new ideas but still AI machines cannot beat this power of human intelligence and cannot be creative and imaginative.

Expert System:

An expert system is a computer program that is designed to solve complex problems and to provide decision-making ability like a human expert. It performs this by extracting knowledge from its knowledge base using the reasoning and inference rules according to the user queries.

It solves the most complex issue as an expert by extracting the knowledge stored in its knowledge base. The system helps in decision making for complex problems using both facts and heuristics like a human expert. It is called so because it contains the expert knowledge of a specific domain and can solve any complex problem of that particular domain. These systems are designed for a specific domain, such as medicine, science, etc.

The performance of an expert system is based on the expert's knowledge stored in its knowledge base. The more knowledge stored in the KB, the more that system improves its performance. One of the common examples of an ES is a suggestion of spelling errors while typing in the Google search box.



Characteristics of Expert System

- **High Performance:** The expert system provides high performance for solving any type of complex problem of a specific domain with high efficiency and accuracy.

- **Understandable:** It responds in a way that can be easily understandable by the user. It can take input in human language and provides the output in the same way.
- **Reliable:** It is much reliable for generating an efficient and accurate output.
- **Highly responsive:** ES provides the result for any complex query within a very short period of time.

Components of Expert System

An expert system mainly consists of three components:

- **User Interface**
- **Inference Engine**
- **Knowledge Base**

1. User Interface

With the help of a user interface, the expert system interacts with the user, takes queries as an input in a readable format, and passes it to the inference engine. After getting the response from the inference engine, it displays the output to the user. In other words, it is an interface that helps a non-expert user to communicate with the expert system to find a solution.

2. Inference Engine (Rules of Engine)

- The inference engine is known as the brain of the expert system as it is the main processing unit of the system. It applies inference rules to the knowledge base to derive a conclusion or deduce new information. It helps in deriving an error-free solution of queries asked by the user.
- With the help of an inference engine, the system extracts the knowledge from the knowledge base.
- There are two types of inference engine:
- Deterministic Inference engine: The conclusions drawn from this type of inference engine are assumed to be true. It is based on facts and rules.
- Probabilistic Inference engine: This type of inference engine contains uncertainty in conclusions, and based on the probability.

Inference engine uses the below modes to derive the solutions:

- **Forward Chaining:** It starts from the known facts and rules, and applies the inference rules to add their conclusion to the known facts.
- **Backward Chaining:** It is a backward reasoning method that starts from the goal and works backward to prove the known facts.

3. Knowledge Base

- The knowledgebase is a type of storage that stores knowledge acquired from the different experts of the particular domain. It is considered as big storage of knowledge. The more the knowledge base, the more precise will be the Expert System.
- It is similar to a database that contains information and rules of a particular domain or subject.
- One can also view the knowledge base as collections of objects and their attributes. Such as a Lion is an object and its attributes are it is a mammal, it is not a domestic animal, etc.

Components of Knowledge Base

- **Factual Knowledge:** The knowledge which is based on facts and accepted by knowledge engineers comes under factual knowledge.
- **Heuristic Knowledge:** This knowledge is based on practice, the ability to guess, evaluation, and experiences.

Knowledge Representation: It is used to formalize the knowledge stored in the knowledge base using the If-else rules.

Knowledge Acquisitions: It is the process of extracting, organizing, and structuring the domain knowledge, specifying the rules to acquire the knowledge from various experts, and store that knowledge into the knowledge base.

Needs of Expert System

1. **No memory Limitations:** It can store as much data as required and can memorize it at the time of its application. But for human experts, there are some limitations to memorize all things at every time.

2. **High Efficiency:** If the knowledge base is updated with the correct knowledge, then it provides a highly efficient output, which may not be possible for a human.
3. **Expertise in a domain:** There are lots of human experts in each domain, and they all have different skills, different experiences, and different skills, so it is not easy to get a final output for the query. But if we put the knowledge gained from human experts into the expert system, then it provides an efficient output by mixing all the facts and knowledge
4. **Not affected by emotions:** These systems are not affected by human emotions such as fatigue, anger, depression, anxiety, etc.. Hence the performance remains constant.
5. **High security:** These systems provide high security to resolve any query.
6. **Considers all the facts:** To respond to any query, it checks and considers all the available facts and provides the result accordingly. But it is possible that a human expert may not consider some facts due to any reason.
7. **Regular updates improve the performance:** If there is an issue in the result provided by the expert systems, we can improve the performance of the system by updating the knowledge base.

Capabilities of the Expert System

- **Advising:** It is capable of advising the human being for the query of any domain from the particular ES.
- **Provide decision-making capabilities:** It provides the capability of decision making in any domain, such as for making any financial decision, decisions in medical science, etc.
- **Demonstrate a device:** It is capable of demonstrating any new products such as its features, specifications, how to use that product, etc.
- **Problem-solving:** It has problem-solving capabilities.
- **Explaining a problem:** It is also capable of providing a detailed description of an input problem.
- **Interpreting the input:** It is capable of interpreting the input given by the user.
- **Predicting results:** It can be used for the prediction of a result.

- **Diagnosis:** An ES designed for the medical field is capable of diagnosing a disease without using multiple components as it already contains various inbuilt medical tools.

Advantages of Expert System

- These systems are highly reproducible.
- They can be used for risky places where the human presence is not safe.
- Error possibilities are less if the KB contains correct knowledge.
- The performance of these systems remains steady as it is not affected by emotions, tension, or fatigue.
- They provide a very high speed to respond to a particular query.

Limitations of Expert System

- The response of the expert system may get wrong if the knowledge base contains the wrong information.
- Like a human being, it cannot produce a creative output for different scenarios.
- Its maintenance and development costs are very high.
- Knowledge acquisition for designing is much difficult.
- For each domain, we require a specific ES, which is one of the big limitations.
- It cannot learn from itself and hence requires manual updates.

Applications of Expert System

- **In designing and manufacturing domain** it can be broadly used for designing and manufacturing physical devices such as camera lenses and automobiles.
- **In the knowledge domain**
  These systems are primarily used for publishing the relevant knowledge to the users. The two popular ES used for this domain is an advisor and a tax advisor.
- **In the finance domain**
  In the finance industries, it is used to detect any type of possible fraud, suspicious activity, and advise bankers that if they should provide loans for business or not.
- **In the diagnosis and troubleshooting of devices** in medical diagnosis, the ES system is used, and it was the first area where these systems were used.

- **Planning and scheduling**

  The expert systems can also be used for planning and scheduling some particular tasks for achieving the goal of that task.

Difference between AI and Expert System

Artificial Intelligence	EXPERT SYSTEM
AI is the ability of a machine or a computer program to think, work, learn and react like humans.	Expert systems represent the most successful demonstration of the capabilities of AI.
AI involves the use of methods based on the intelligent behavior of humans to solve complex problems.	Experts systems are computer programs designed to solve complex decision problems.
Components of AI: 1. Natural Language Processing (NLP) 2. Knowledge representation 3. Reasoning 4. Problem solving 5. Machine learning	Components of expert system: 1. Inference engine 2. Knowledge base 3. User interface 4. Knowledge acquisition module
AI is the study is systems that act in a way to any observer would appear to be intelligent.	Expert system represent the most successful demonstration of the capabilities of AI
AI systems are used in a wide range of industries, from healthcare to finance, automotive, data security, etc.	Expert systems provide expert advice and guidance in a wide variety of activities.

UNIT – 4

SYSTEM IDENTIFICATION

Non Parametric methods: Transient Analysis, Frequency analysis, Correlation analysis, Spectral analysis. Parametric methods: Least Square method, Recursive least square method.

System identification

System identification is a methodology for building mathematical models of dynamic systems using measurements of the input and output signals of the system. System identification methods are divided into two groups: parametric and nonparametric. Parametric methods identify system model with an underlying mathematical structure that is associated with a coefficient set or parameters, whereas nonparametric methods model a system directly with its responses.
The process of system identification requires that you:

- Measure the input and output signals from your system in time or frequency domain.
- Select a model structure.
- Apply an estimation method to estimate values for the adjustable parameters in the candidate model structure.
- Evaluate the estimated model to see if the model is adequate for your application needs.

A method for obtaining a transfer function of a system is a parametric method. The system parameters in this case are coefficients of the transfer function, and the number of parameters is less than or equal to $2n + 1$ where n is the order of the system. In the same way, a state equation or a difference equation method belongs to the parametric group. In contrast, a method for obtaining an impulse response, step response or frequency response of the system belongs to the nonparametric group.

Similarly, identification methods for nonlinear systems are also divided into the parametric and nonparametric groups. Nonparametric methods of nonlinear system identification include those system representation methods using Volterra kernels or Wiener kernels. Hence a nonparametric method for nonlinear system identification usually means a method for obtaining Volterra kernels or Wiener kernels.

In a specific sense, system identification is concerned with coming with an accurate model given the input-output signals recorded during working of the studied system. Hence, it becomes plain that system identification is closely related to other fields of mathematical modeling. We mention here the various domains concerned with parameter estimation including statistical inference, adaptive filtering and machine learning. Historically, system identification originates from an engineering need to form models of dynamical systems: it then comes as no surprise that traditionally emphasis is laid on numerical issues as well on system-theoretical concerns.

Progress in the field has much been reinforced by introducing good software to execute the various algorithms. This makes the methodology semi-automatic: that is a user needs still have a conceptual overview on what is to be done, but the available software tools take care of most technical details. In this course, the use of the MATLAB System Identification toolbox is discussed in some detail.

Stirred Thank: The following is a prototypical example in the context of process control. Consider a bio- chemical reactor, where two different substances go in via respective pipelines. Both inflows come at a certain flow-rate and have a certain concentration, either of which can be controlled by setting valves. Then the substances interact inside the stirred tank, and the yield is tapped from the tank. Maybe the aim of such process is to maximize the concentration of the yield at certain instances. A mathematical approach to such automatic control however requires a mathematical description of the process of interest. That is, we need to set up equations relating the setting of the valves and the output. Such model could be identified by experimenting on the process and compiling the observed results into an appropriate model.

Speech: Consider the apparatus used to generate speech in the human. In an abstract fashion, this can be seen as a white noise signal generated by the glottis. Then the mouth is used to filter this noise into structured signals which are perceived by an audience as meaningful. Hence, this apparatus can be abstracted into a model with unknown white noise input, a dynamical system shaped by intention, and an output which can be observed. Identification of the filter (dynamical system) can for example be used to make an artificial speech.

Industrial: The prototypical example of an engineering system is an industrial plant which is fed by an inflow of raw material, and some complicated process converts it into the desired yield. Often the internal mechanism of the studied process can be worked out in some detail. Nevertheless, it might be more useful to come up with a simpler model relating input-signals to output- signals directly, as it is often (i) easier (cheaper) to develop, (ii) is directly tuned to our need, and (iii) makes abstraction of irrelevant mechanisms in the process, and (iv) might better handle the unforeseen disturbances.

Acoustic: The processing of acoustical signals can be studied in the present context. Let us for example study the room which converts an acoustic signal (say a music signal) into an acoustic signal augmented with echo. It is then often of interest to compensate the signal sent into the room for this effect, so as to 'clean' the perceived signal by the audience. In this example, the room is conceived as the dynamical system, and it is of interest to derive a model based on acoustic signals going into the room, and the consequent signals perceived by an audience.

Econometric: The following example is found in a financial context. Consider the records of the currency exchange rates. This multivariate time-series is assumed to be driven by political, socio economic or cultural effect. A crude way to model such non measureable effect is as white noise. Then the interesting bit is how the exchange rates are interrelated: how for example a injection of resources in one market might alter other markets as well.

Multimedia: Finally, consider the sequence of images used to constitute a cartoon on TV say. Again, consider the system driven by signals roughly modeling meaning, and outputting the values projected in the different pixels. It is clear that the signals of neighboring pixels are inter- related, and that the input signal is not as high-dimensional as the signals projected on the screen.

The System Identification Procedure

Different steps in system identification experiment. The practical way to design is typically according to the following steps, each one raising their own challenges:

- Description of the task. What is a final desideratum of a model? For what purpose is it to be used? How will we decide at the end of the day if the identified model is satisfactory? On which properties to we have to focus during the identification experiments?
- Look at initial Data. What sort of effects are of crucial importance to capture? What are the challenges present in the task at hand. Think about useful graphs displaying the data. Which phenomena in those graphs are worth pinpointing?
- Nonparametric analysis. If possible, do some initial experiments: apply an pulse or step to the system, and look at the outcome. Perhaps a correlation or a spectral

analysis is possible as well. Look at where random effects come in. If exactly the same experiment is repeated another day, how would the result differ? Is it possible to get an idea of the form of the disturbances?

- Design Experiment. Now that we have acquired some expertise of the task at hand, it is time to set up the large identification experiment. At first, enumerate the main challenges for identification, and formalize where to focus on during the experiment. Then design an experiment so as to maximize the information which can be extracted from observations made during the experiment. For example. Make sure all the dynamics of the studied system are sufficiently excited. On the other hand, it is often paramount to make sure that the system remains in the useful 'operation mode' throughout the experiment. That is, it is no use to inject the system with signals which do not apply in situations where/when the model is to be used.

- Identify model. What is a good model structure? What are the parameters which explain the behavior of the system during the experiment?
- Refine Analysis: It is ok to start of with a hopelessly naive model structure. But it is then paramount to refine the model structure and the subsequent parameter estimation in order to compensate for the effects which could not be expressed in the first place. It is for example common practice to increase the order of the dynamical model. Is the noise of the model reflecting the structure we observe in the first place, or do we need more flexible noise models? Which effects do we see in the data but are not captured by the model: time-varying, nonlinear, aging, saturation...
- Verify Model: is the model adequate for the purpose at hand? Does the model result in satisfactory results as written down at the beginning? Is it better than a naive approach? Is the model accurately extracting or explaining the important effects? For example, analyze the residuals left over after subtracting the modeled behavior from the observed data. Does it still contain useful information, or is it white? Implement the model for the intended purpose, thus it work satisfactory?

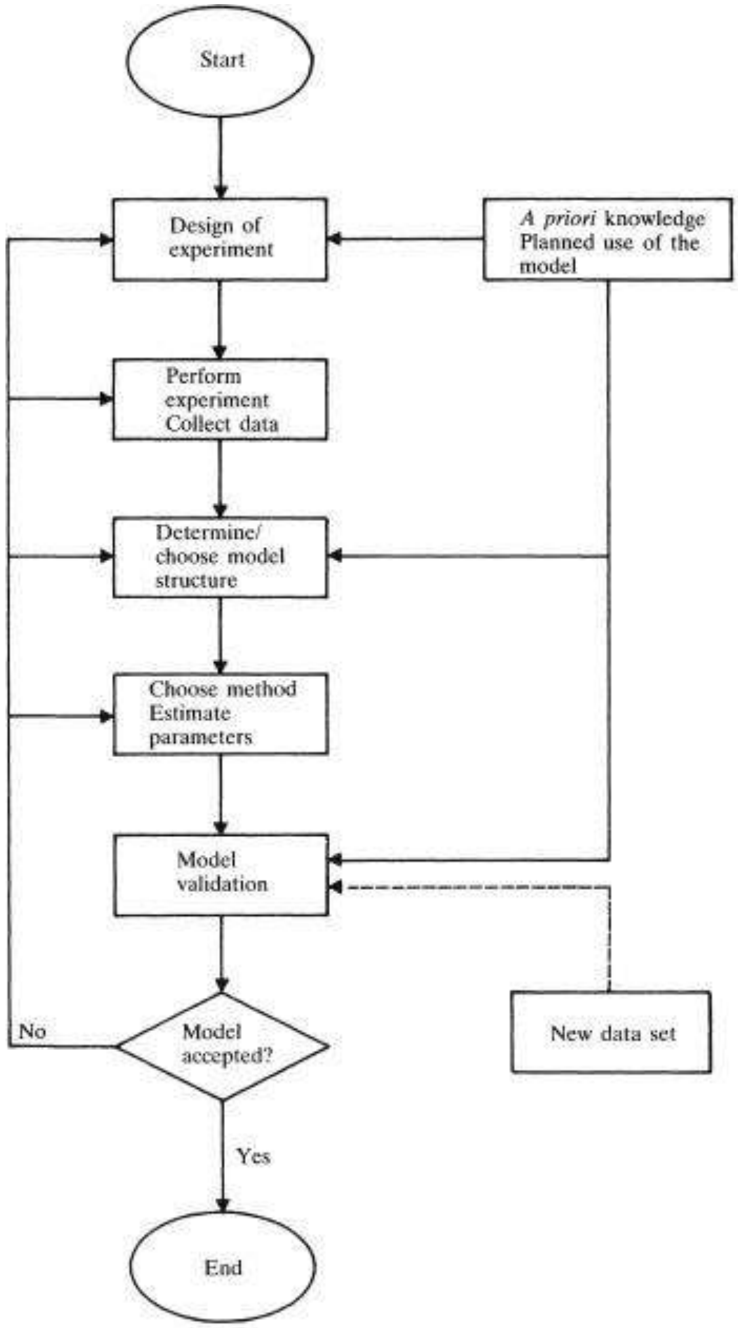
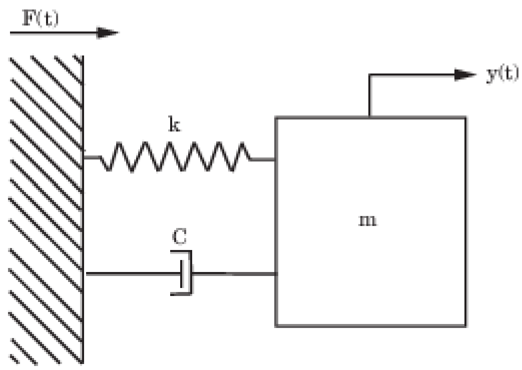


FIGURE 1.3 Schematic flowchart of system identification.

Dynamic Systems and Models

In a dynamic system, the values of the output signals depend on both the instantaneous values of the input signals and also on the past behavior of the system. For example, a car seat is a dynamic system—the seat shape (settling position) depends on both the current weight of the passenger (instantaneous value) and how long the passenger has been riding in the car (past behavior). A model is a mathematical relationship between the input and output variables of the system. Models of dynamic systems are typically described by differential or difference equations, transfer functions, state-space equations, and pole-zero-gain models. One can represent dynamic models in both continuous-time and discrete-time form.

An often-used example of a dynamic model is the equation of motion of a spring-mass-damper system. As the following figure shows, the mass moves in response to the force $F(t)$ applied on the base to which the mass is attached. The input and output of this system are the force $F(t)$ and displacement $y(t)$, respectively.



Nonparametric Techniques

Transient Analysis

A first approach is to inject the studied system with a simple input as a pulse or a step, and to record the subsequent output of the system. This gives then an impression of the impulse response of the studied system. The pros of this approach are that (i) it is simple to understand or to (ii) implement, while the model need not be specified further except for the LTI property. The downsides are of course that (i) this method break down when the LTI model fits not exactly the studied system. Since models serve merely as mathematical convenient approximations of the actual system, this is why this approach is in practice not often used. (ii) It cannot handle random effects very well. (iii) Such experiment is not feasible in the practical setting at hand. As for this reason it is merely useful in practice to determine some structural properties of the system.

Frequency Analysis

An LTI is often characterized in terms of its reaction to signals with a certain frequency and phase. It is hence only natural to try to learn some properties of the studied system by injecting it with a signal having such a form. If repeating this procedure for a range of frequencies, one can obtain a graphical representation of complex variable. Such Bode plots (or Nyquist or related plots) are well suited for the design and analysis of automatic control systems. This procedure is rather sensitive to disturbances. And due to the presence of noise it will be difficult to extract good estimates of Amplitude and Phase from those signals.

A Correlation Analysis

The above ideas are taken a step further into a correlation analysis. But instead of using simple input signals, the system is injected with a random signal $\{u_t\}_t$ which has zero mean or which has finite values. This technique is related to Least Square estimate and the Prediction Error Method.

Spectral Analysis

Now both the correlation technique and the frequency analysis method can be combined into a signal nonparametric approach as follows. The idea is to take the Discrete Fourier Transforms (DFT) of the involved signals, and find the transfer function relating them. This estimate is sometimes called the empirical transfer function estimate. However the above estimate to the spectral densities and the transfer function will give poor results.

Least Square Estimation

The "least squares" method is a form of mathematical regression analysis used to determine the line of best fit for a set of data, providing a visual demonstration of the relationship between the data points. Each point of data represents the relationship between a known independent variable and an unknown dependent variable.

The least squares method provides the overall rationale for the placement of the line of best fit among the data points being studied. The most common application of this method, which is sometimes referred to as "linear" or "ordinary", aims to create a straight line that minimizes the sum of the squares of the errors that are generated by the results of the associated equations, such as the squared residuals resulting from differences in the observed value, and the value anticipated, based on that model.

This method of regression analysis begins with a set of data points to be plotted on an x- and y-axis graph. An analyst using the least squares method will generate a line of best fit that explains the potential relationship between independent and dependent variables.

In regression analysis, dependent variables are illustrated on the vertical y-axis, while independent variables are illustrated on the horizontal x-axis. These designations will form the equation for the line of best fit, which is determined from the least squares method.

In contrast to a linear problem, a non-linear least squares problem has no closed solution and is generally solved by iteration. The discovery of the least squares method is attributed to Carl Friedrich Gauss, who discovered the method in 1795.

- The least squares method is a statistical procedure to find the best fit for a set of data points by minimizing the sum of the offsets or residuals of points from the plotted curve.
- Least squares regression is used to predict the behavior of dependent variables.

Example of the Least Squares Method
An example of the least squares method is an analyst who wishes to test the relationship between a company's stock returns, and the returns of the index for which the stock is a component. In this example, the analyst seeks to test the dependence of the stock returns on the index returns. To achieve this, all of the returns are plotted on a chart. The index returns are then designated as the independent variable, and the stock returns are the dependent variable. The line of best fit provides the analyst with coefficients explaining the level of dependence.

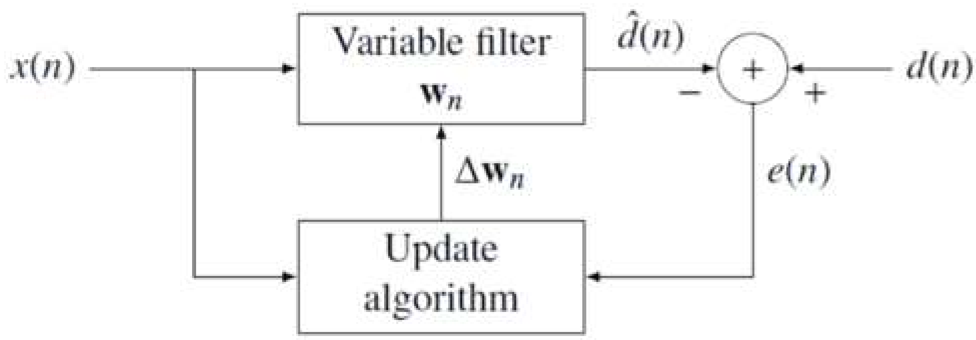
The Line of Best Fit Equation
The line of best fit determined from the least squares method has an equation that tells the story of the relationship between the data points. Line of best fit equations may be determined by computer software models, which include a summary of outputs for analysis, where the coefficients and summary outputs explain the dependence of the variables being tested.

Least Squares Regression Line
If the data shows a leaner relationship between two variables, the line that best fits this linear relationship is known as a least squares regression line, which minimizes the vertical distance from the data points to the regression line. The term "least squares" is used because it is the smallest sum of squares of errors, which is also called the "variance".

Recursive least squares (RLS)

Recursive least squares (RLS) is an adaptive filter algorithm that recursively finds the coefficients that minimize a weighted linear least squares cost function relating to the input signals. This approach is in contrast to other algorithms such as the least mean squares (LMS) that aim to reduce the mean square error. In the derivation of the RLS, the input signals are considered deterministic, while for the LMS and similar algorithm they are considered stochastic. Compared to most of its competitors, the RLS exhibits extremely fast convergence. However, this benefit comes at the cost of high computational complexity.

The idea behind RLS filters is to minimize a cost function C by appropriately selecting the filter coefficients Wn, updating the filter as new data arrives. The error signal e(n) and desired signal d(n) are defined in the negative feedback diagram below:



Unit – 5

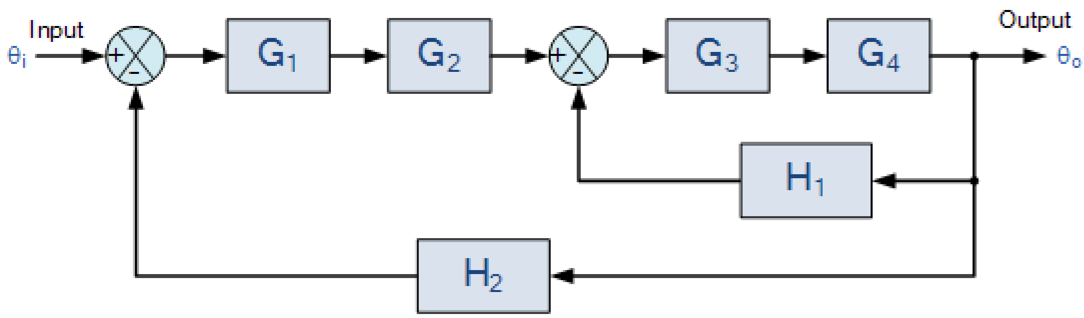
MULTI LOOP REGULATORY CONTROL

Multi-Loop Control: Introduction, Process Interaction, Pairing of Input and Outputs, Relative Gain Array (RGA) - Properties and Application of RGA, Multi-loop PID Controller - Decoupler.

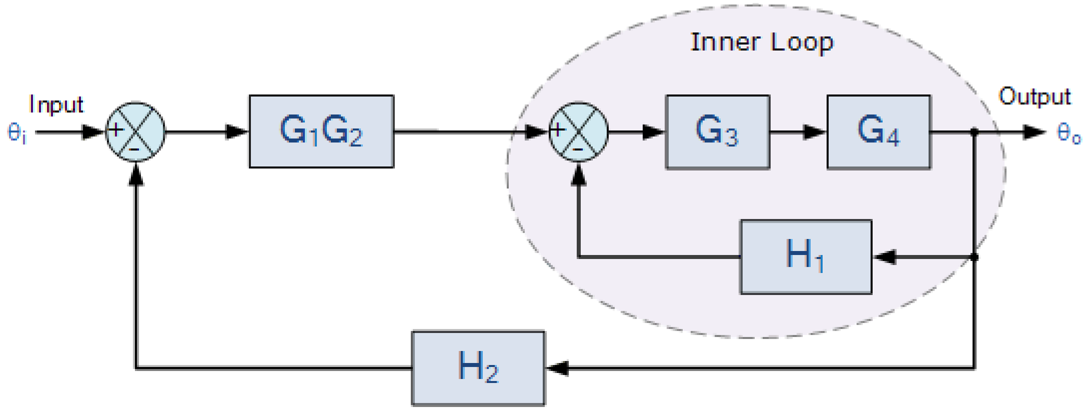
Multi-loop Closed-loop System

The basic transfer function still applies to more complex multi-loop systems. Most practical feedback circuits have some form of multiple loop control, and for a multi-loop configuration the transfer function between a controlled and a manipulated variable depends on whether the other feedback control loops are open or closed.

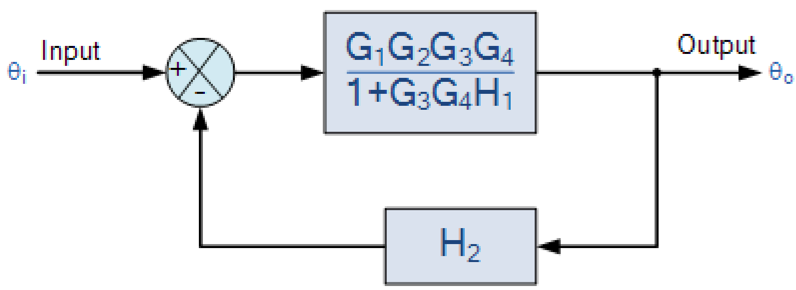
Consider the multi-loop system below.



Any cascaded blocks such as G_1 and G_2 can be reduced, as well as the transfer function of the inner loop as shown.



After further reduction of the blocks we end up with a final block diagram which resembles that of the previous single-loop closed-loop system.



And the transfer function of this multi-loop system becomes:

$$\frac{\text{Output}}{\text{Input}} = \frac{\theta_o}{\theta_i} = \frac{G_1 G_2 G_3 G_4}{1 + G_3 G_4 H_1 + G_1 G_2 G_3 G_4 H_2}$$

Then we can see that even complex multi-block or multi-loop block diagrams can be reduced to give one single block diagram with one common system transfer function.

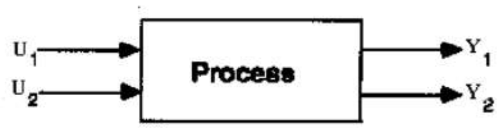
Multiloop control: Each manipulated variable depends on only a single controlled variable, i.e., a set of conventional feedback controllers.

Multivariable control: Each manipulated variable can depend on two or more of the controller variables.

Examples: decoupling control, model predictive control.

Multiloop control Strategy:

- Typical industrial approach
- Consists of using n standard feedback controllers e.g., PID, one for each controller variable.
- Control system design
    1. Select controlled and manipulated variables
    2. Select pairing of controlled AND MANIPULATED VARIABLES
    3. Specify types of feedback controllers.



Two possible controller pairings:

U_1 with Y_1, U_2 with Y_2 (1-1/2-2 pairing)

or

U_1 with Y_2, U_2 with Y_1 (1-2/2-1 pairing)

For *n* x *n* system, *n!* possible pairing configurations.

Transfer function model (2 X 2 system)

Two controlled variables and two manipulated variables
(4 transfer functions required)

$$\frac{Y_1(s)}{U_1(s)} = G_{P11}(s), \quad \frac{Y_1(s)}{U_2(s)} = G_{P12}(s)$$

$$\frac{Y_2(s)}{U_1(s)} = G_{P21}(s), \quad \frac{Y_2(s)}{U_2(s)} = G_{P22}(s)$$

Thus, the input-output relations for the process can be written as:

$$Y_1(s) = G_{P11}(s)U_1(s) + G_{P12}(s)U_2(s)$$

$$Y_2(s) = G_{P21}(s)U_1(s) + G_{P22}(s)U_2(s)$$

Or in vector-matrix notation as,

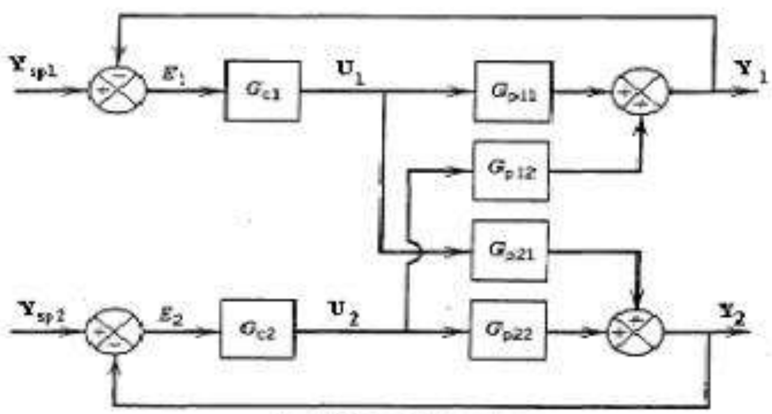
$$\boldsymbol{Y}(s) = \boldsymbol{G}_p(s)\boldsymbol{U}(s)$$

where $\boldsymbol{Y}(s)$ and $\boldsymbol{U}(s)$ are vectors,

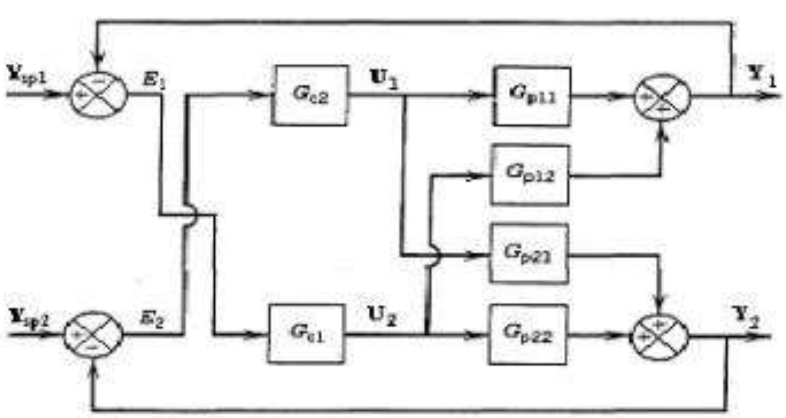
$$\boldsymbol{Y}(s) = \begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} \quad \boldsymbol{U}(s) = \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix}$$

And $\boldsymbol{G}_p(s)$ is the transfer function matrix for the process

$$\boldsymbol{G}_p(s) = \begin{bmatrix} G_{P11}(s) & G_{P12}(s) \\ G_{P21}(s) & G_{P22}(s) \end{bmatrix}$$



1-1/2-2 controller pairing



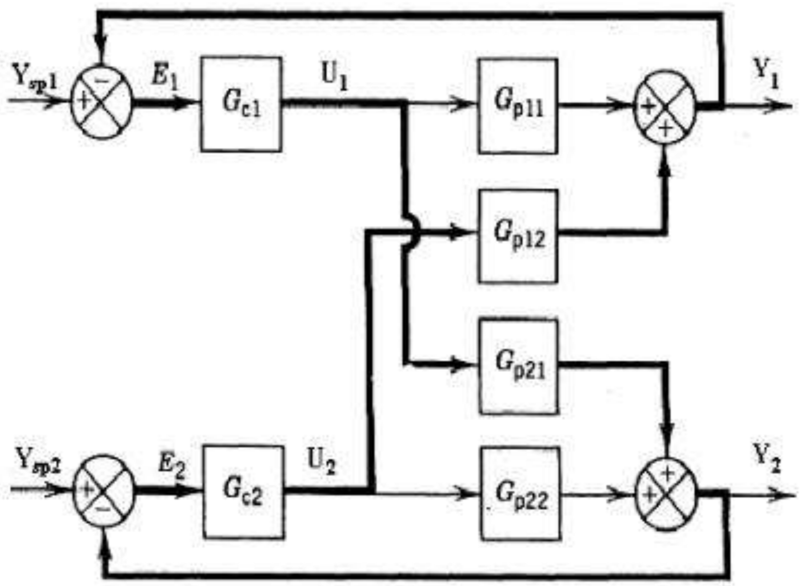
(b) 1-2/2-1 controller pairing

Block diagrams for 2 X 2 multiloop control schemes

Loop interactions:

Process interactions may induce undesirable interactions between two or more control loops. Example for a 2X2 system, control loop interactions are due to the presence of third feedback loop. The problems due to these loop interactions are

- Closed loop system may become destabilized.
- Controller tuning becomes more difficult.



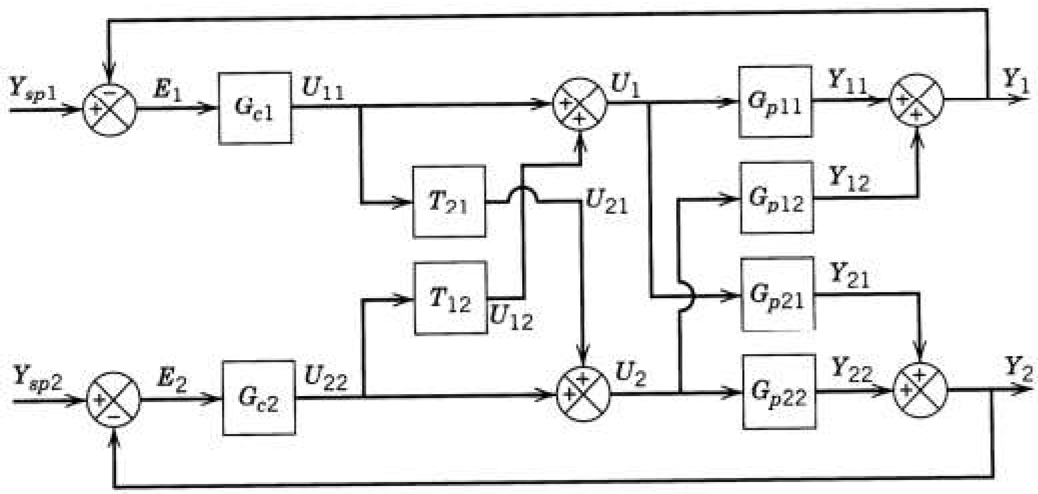
The hidden feedback control loop (in dark lines) for a 1-1/2-2 controller pairing.

Strategies for dealing with undesirable control loop interactions

1. Detune one or more FB controllers.
2. Select different manipulated or controlled variables.
3. Use a decoupling control scheme.
4. Use some other type of multivariable control scheme.

Decoupling control System

- Use additional controllers to compensate for process interactions and thus reduce control loop interactions.
- Ideally, decouple control allows set- point changes to affect only the desired controlled variables.
- Typically, decoupling controllers are designed using a simple process model like a steady-state model or transfer function model.



A decoupling control system.

Decoupler Design Equation

We want cross-controller, T_{12}, to cancel the effect of U_2 on Y_1. Thus, we would like,

$$T_{12}G_{P11}U_{22} + G_{P12}U_{22} = 0$$

Because $U_{22} \neq 0$ in general, then

$$T_{12} = -\frac{G_{P12}}{G_{P11}}$$

Similarly, we want T_{12} to cancel the effect of U_1 on Y_2. Thus, we require that,

$$T_{21}G_{P22}U_{11} + G_{P21}U_{11} = 0$$

$$\therefore\ T_{21} = -\frac{G_{P21}}{G_{P22}}$$

Compare with the design equations for feedforward control based on block diagram analysis

Relative Gain Array

One method for designing and analyzing a MIMO control scheme for a process in steady state is with a **R**elative **G**ain **A**rray (RGA). RGA is useful for MIMO systems that can be decoupled (see the article about determining if a system can be decoupled). For systems that cannot be decoupled, model predictive control or neural networks are better choices of analysis tool than RGA. A good MIMO control scheme for a system that can be decoupled is one that can control a process variable without greatly affecting the other process variables. It must also be stable with respect to dynamic situations, load changes, and random disturbances. The RGA provides a quantitative approach to the analysis of the interactions between the controls and the output, and thus provides a method of pairing manipulated and controlled variables to generate a control scheme.

Relative **G**ain **A**rray is an analytical tool used to determine the optimal input-output variable pairings for a multi-input-multi-output (MIMO) system. In other words, the RGA is a normalized form of the gain matrix that describes the impact of each control variable on the output, relative to each control variable's impact on other variables. The process interaction of open-loop and closed-loop control systems is measured for all possible input-output variable pairings. A ratio of this open-loop 'gain' to this closed-loop 'gain' is determined and the results are displayed in a matrix.

$$RGA = \Lambda = \begin{bmatrix} \lambda_{11} & \lambda_{12} & \cdots & \lambda_{1n} \\ \lambda_{21} & \lambda_{22} & \cdots & \lambda_{2n} \\ \vdots & & & \\ \lambda_{n1} & \lambda_{n2} & \cdots & \lambda_{nn} \end{bmatrix}$$

The array will be a matrix with one column for each input variable and one row for each output variable in the MIMO system. This format allows a process engineer to easily compare the relative gains associated with each input-output variable pair, and ultimately to match the input and output variables that have the biggest effect on each other while also minimizing undesired side effects.

Properties and Application of RGA

The following are some of the linear algebra properties of RGA:

1. Each row and column of ф (G) sums to 1.
2. For nonsingular diagonal matrices D and E, ф (G) = ф (DGE).
3. For Permutation matrices P and Q, Pф (G) Q = ф (PGQ).
4. Lastly, ф (G^{-1}) = ф $(G)^{T}$ = ф (G^{T}).

Reference:

[1]. https://eng.libretexts.org
[2]. https://en.wikibooks.org
[3]. https://www.inductiveautomation.com
[4]. https://www.javatpoint.com
[5]. http://www.brainkart.com

Post-Test:

1. Which among the following constitute the state model of a system in addition to state equations?
   a) Input equations
   b) **Output equations**
   c) State trajectory
   d) State vector
2. Which among the following plays a crucial role in determining the state of dynamic system?
   a) **State variables**
   b) State vector
   c) State space
   d) State scalar
3. State space analysis is applicable even if the initial conditions are _____
   a) Zero
   b) **Non-zero**
   c) Equal
   d) Not equal
4. Conventional control theory is applicable to ______ systems
   a) **SISO**
   b) MIMO
   c) Time varying
   d) Non-linear
5. Forward chaining systems are _____________ where as backward chaining systems are ___________
   a) Goal-driven, goal-driven
   b) Goal-driven, data-driven
   c) **Data-driven, goal-driven**
   d) Data-driven, data-driven

6. a) An expert system is a computer program that contains some of the subject-specific knowledge of one or more human experts.
   b) A knowledge engineer has the job of extracting knowledge from an expert and building the expert system knowledge base.
   a**) True, True**
   b) False, True
   c) True, false
   d) False, false
7. a) In a backward chaining system you start with the initial facts, and keep using the rules to draw new conclusions given those facts.
   b) In a backward chaining system, you start with some goal trying to prove, and keep looking for rules that would allow you to conclude that goal, perhaps setting new sub-goals to prove.
   a) **False, True**
   b) False, False
   c) True, true
   d) None
8. A rule-based system consists of a bunch of___________ rules.
   a) If-Than
   b) loops
   c) And – OR
   d) All the above
9. The formula to calculate the Kth element in the row is
   a. $\boldsymbol{b_k = \begin{bmatrix} a_0 & a_{n-k} \\ a_n & a_k \end{bmatrix}}$
   b. $b_k = \begin{bmatrix} a_{n-1} & a_n \\ a_n & a_k \end{bmatrix}$
   c. $b_k = \begin{bmatrix} a_0 & a_{n-k} \\ a_{n-k} & a_k \end{bmatrix}$
   d. $b_k = \begin{bmatrix} a_1 & a_{n-k} \\ a_n & a_k \end{bmatrix}$
10. The Z transform of e^{-NTs} is?
   a. Z^{-N}
   b. Z^{-T}
   c. Z^{-s}
   d. Z^{-1}
11. Check the Sufficient condition for the system represented as $z^3 - 0.2z^2 - 0.25z + 0.05 = 0$
   a. **Sufficient conditions for stability are satisfied**.
   b. Sufficient conditions for stability are not satisfied.

c. Sufficient conditions for stability are marginally satisfied.
d. None of the mentioned.

12. For the sample data control system G(s)= 1/(S+1) find the response.
a. $\mathbf{\frac{0.632}{Z+0.264}}$
b. $\frac{0.632}{Z-0.264}$
c. $\frac{0.32}{Z+0.264}$
d. $\frac{0.632}{Z+0.64}$

13. A system is said to be_____________ if it is possible to transfer the system state from any initial state to any desired state in finite interval of time.
a) **Controllable**
b) Observable
c) Cannot be determined
d) Controllable and observable

14. A system is said to be_________________ if every state can be completely identified by measurements of the outputs at the finite time interval.
a) Controllable
b) **Observable**
c) Cannot be determined
d) Controllable and observable

15. A transfer function of the system does not have pole-zero cancellation? Which of the following statements is true?
a) System is neither controllable nor observable
b) **System is completely controllable and observable**
c) System is observable but uncontrollable
d) System is controllable and unobservable