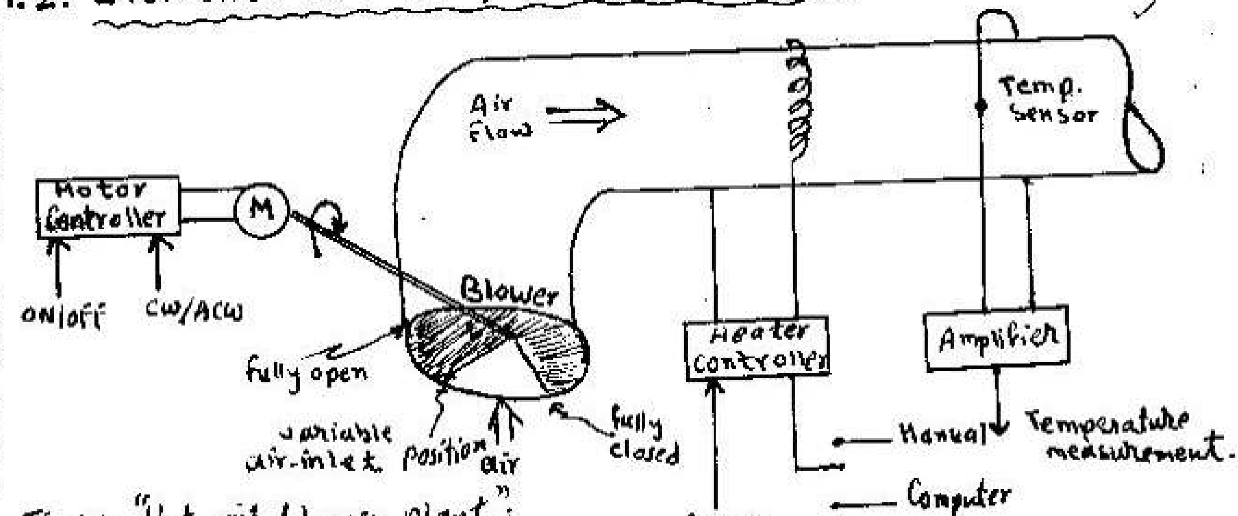
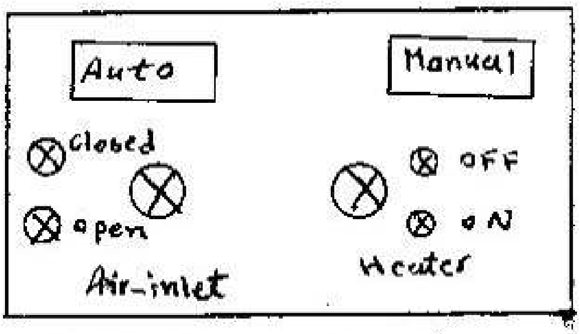
1.2: Elements of A Computer Control System:-

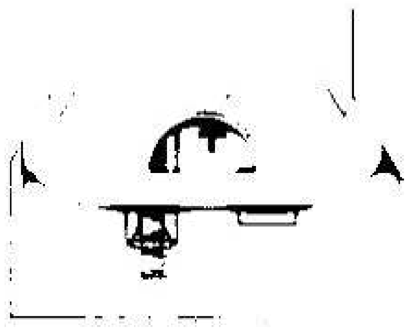


Fig(2): "Hot-air blower plant":

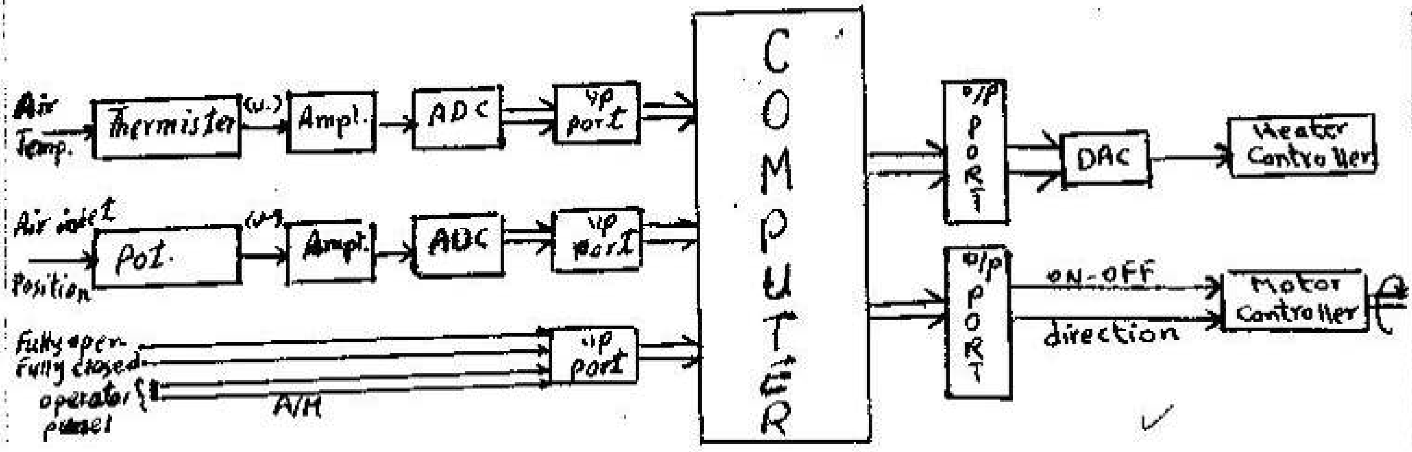
- A thermistor is placed at the outlet end of the tube to produce a signal proportional to the air temperature.
- The position of the air-inlet cover is adjusted by a reversible motor. The motor is controlled by an ON-OFF logic signal.
- A potentiometer wiper is attached to the air-inlet cover and the o/p voltage is proportional to the position of the cover.
- Two micro switches are used to detect when the cover is fully open and fully closed.
- The current supplied to the heating element can be varied by supplying a d.c. voltage.
- The operator is provided with a panel from which the control system can be switched from Auto to Manual, as in Fig(3):



Fig(3): operator Panel



The DDC of the described plant is as shown in Fig(4) below:-



Fig(4): DDC of a hot-air blower plant.

- The operation of the above Configuration requires Monitoring and Control calculations & actuation.

* Monitoring: involves: obtaining information about the current state of the plant. The Information is available from the plant instruments in the form of:
  - Analog Signals: air temp, air-inlet Cover position
  - Digital Signals: air-inlet Cover position (Fully open, Fully closed) and for Status Signals (Auto/Manual, Motor ON, Heater ON ...)

* Control Calculations: Involves the digital equivalent of Continuous feedback Control for the control of temp. There is feedback position Control for the air-inlet Cover position, and there is sequence and interlock Control;
  - The heater should not be ON if the fan is not running.
  - Automatic change from tracking to Controlling when the operator changes from manual to Auto...

1.3: Classification of Real-time Computer Control Systems:-

1.3.a: process - Related classification:

1.3.a1: Clock - based Systems:

A plant operates in real-time and thus we talk about the plant time-constants; these may be measured in hours for some chemical processes or in milliseconds for an aircraft system, for instant. For feedback control the required sampling rate will be dependent on the time-constant of the process to be controlled. "The shorter the time-constant of the process, the faster the required sampling rate." The computer which is used to control the plant must therefore be synchronized to real-time or natural time and must be able to carry out all the required operations, measurement, control and actuation within each sampling interval.

The completion of the operations within the specified time is dependent on the : 1- Number of operations to be performed & 2- the speed of the computer. Synchronization is usually obtained by adding to the computer system a clock - normally refered to as a 'Real-time' clock - and using a signal from this clock to "Interrupt" the operations of the computer at some predetermined fixed time interval.

The computer may run the plant input, plant output and control tasks in response to the clock interrupt: or, if the clock interrupt has been set at a faster rate than the sampling rate, it may count each interrupt until it is time to run the task. In larger systems the tasks may be subdivided into groups for controlling different parts of the plant and these may need to run at different sampling rates. The clock interrupt is frequently used to keep a clock and calander so that the computer system is aware of both the time & date.

1.3.a.2: Sensor-based Systems:-

There are many Systems where actions have to be performed, not at particular times or time intervals, but in response to some event. Typical examples are: turning off a pump or closing a valve when the level in a liquid tank reaches a predetermined value; or switching a motor off in response to the closure of a switch indicating that some desired position has been reached. Sensor-based systems are also used extensively to indicate alarm Conditions and initiate alarm actions, e.g. as an indication of too high a temperature or too great a pressure. The specification of sensor-based systems usually includes a requirement that the system must respond within a given maximum time to a particular event. ✓

Sensor-based systems normally employ interrupts to inform the Computer System that action is required. Some small, simple systems may use polling; that is, the Computer periodically asks (polls) the various sensors to see if action is required.

1.3.a.3: Interactive Systems:-

Interactive Systems represent the largest class of real-time system. The real-time requirement is usually expressed in terms of the average response time not exceeding a specified value.

1.3.b: Computer-Related classification - classification of Programms:-

There are three types of programming:-

1.3.b.1 - Sequential: Actions are ordered as a time sequence.

1.3.b.2 - Multi-tasking: Several Actions are performed in parallel.

1.3.b.3 - Real-time: differs from above types in that the actual time taken by an action is an essential factor.

2- Concepts of Computer Control:

Industrial and laboratory processes which use a Computer or Computers in their operation can be classified under one or more of the following categories of operation:

- Batch: this term is used to describe processes in which a sequence of operations is carried out to produce quantity of product.
- Continuous: is used for systems in which production is maintained for long periods of time w/o interruption.
- Laboratory (or Test): here Computer is used to Control some Complex experiemental test or some Complex equipment used for routine testing.

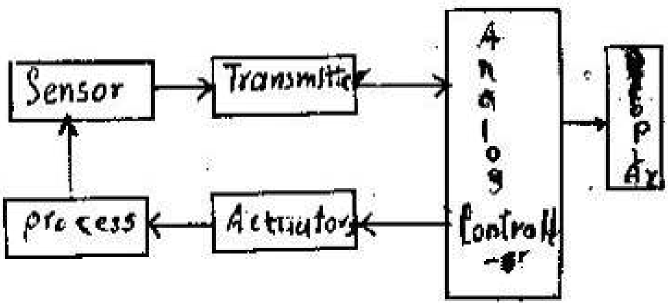
In all these Systems there'll be several different Computer activities being Carried out: Sequence Control, Supervisory Control, loop Control (DDC), the System may be also controlled using "centerlized Computer Control", Some loops may be Controlled using Analog Controller.

The overall objectives of the Control of these processes Can be summarized as:
* Safety. * product Specification. * environment, * ease of operation
* economics.

2.1 Classification of Control :-

a- Analog Controller:-

- For a Simple analog controller an individual process variable is controlled by a Single loop f/b Controller.



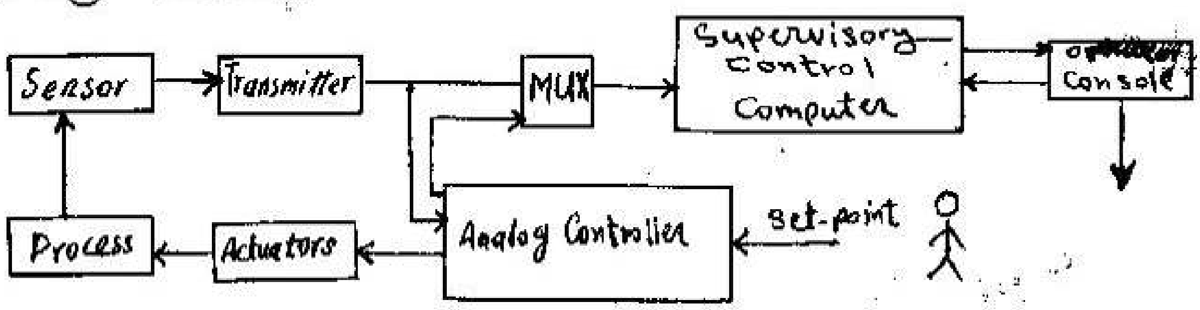
Fig(5): Single loop Analog Controller.

- The analog Controllers are phisically housed in a Centeralized Control room in such away that a process operator has direct access to the to the status of any Control loop and to the display of the process variable and its set points.

Advantadges:
1- Simplicity.
2- Reliability.

Disadvantages:
1- The lack of integrated process information display to the process operator.
2- The difficulty in implementing Complex Control Algorithms.

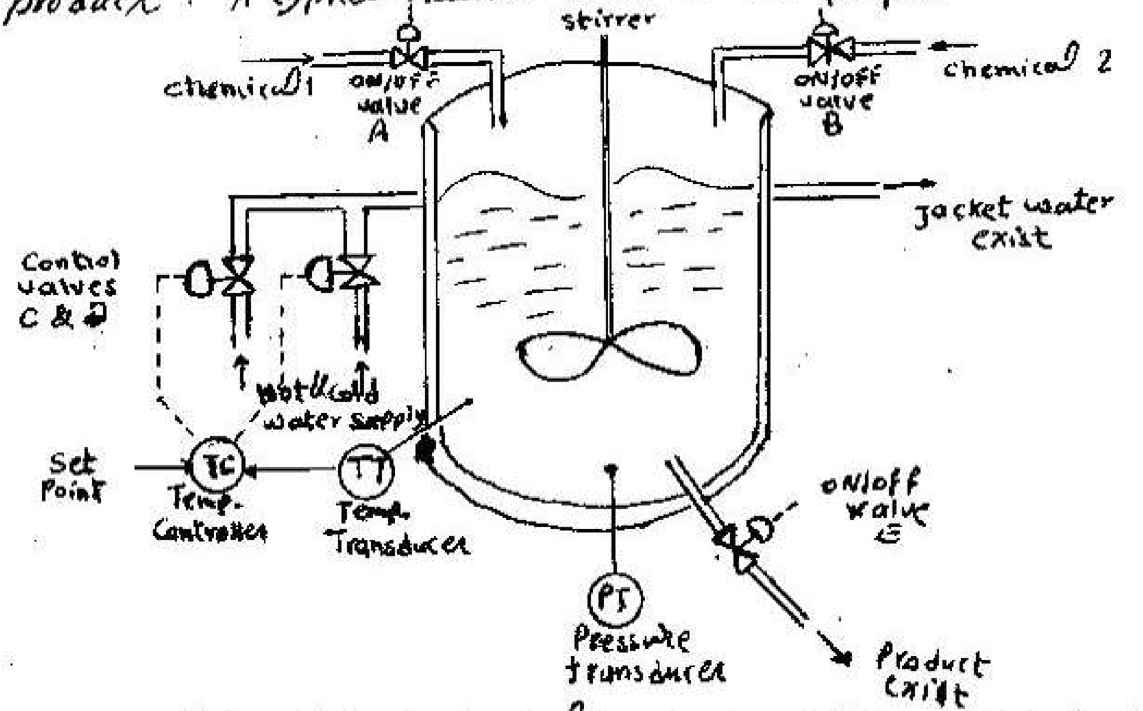
b- Supervisory Control:-
- This mode of Control is called "SUPERVISORY CONTROL" because the Computer is not directly involved in the dynamic f/b Control.
- Supervisory Control overcomes most of the disadvantages of analog Controller.
- In the supervisory Control System, the analog Controller is maintained as before, but a digital Computer is added which periodically scans, digitizes and Inputs process variables to the Computer. as in Fig(6).
- Here Digital Computer is used to:
1. Filter the data, 2- Compute Required Signals, 3- Specialized display, 4- plot Curves, 5- Compute Unmeasurable quantities.
- Many early Computer Control Systems used the Computer in a Supervisory role and not for direct Control. The main reasons were:
1- Computers were very expensive, 2- It was not economically worth to use a computer to replace the analog Control equipment. 3- Computers were not always very reliable.



Fig(6): Supervisory Control scheme.

C. Sequence Control:-

Although sequence control will occur in some part of most systems it often predominates in batch systems and hence a batch system is used to illustrate it. Batch systems are widely used in the food-processing and chemical industries where the operations carried out frequently involve mixing raw materials, carrying out some process and then discharging the product. A typical reactor vessel for this purpose is shown in fig (7).



Fig(7): A Simple chemical reactor vessel (Sequence Control)

The sequence of operations may be as follows:-

1- Open valve A to charge the vessel with chemical 1.

2. check the level of the chemical in the vessel (by monitoring the pressure in the vessel); when the correct amount of chemical has been admitted, close valve A.

3. Start the stirrer used to mix the chemical together.

4. Repeat 1&2 with valve B to admit chemical 2.

5. Switch on the three-term controller and supply a set-point so that the chemical mix is heated up to the required reaction temp.

6. Monitor the reaction temp. when it reaches the set-point, start a timer to time the duration of the reaction.

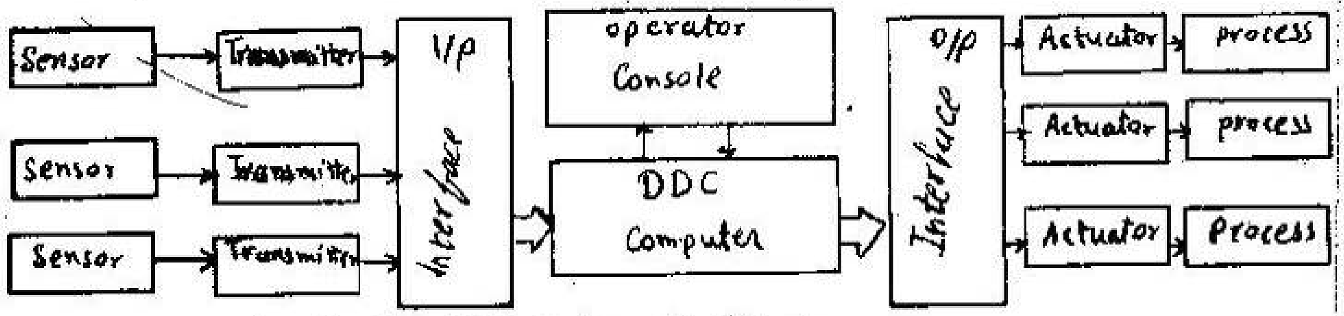
7- when the timer indicates that the reaction is complete, switch off the controller and open valve C to cool down the reactor contents. Switch off the stirrer.

8. Monitor the temperature; when the contents have cooled, open valve & remove the product from the reactor.

Note: when implemented by computer all of the above actions and timings would be based upon software. For a large chemical plant such sequences could become very lengthy and for plant efficiency a number of sequences may be taking place in parallel.

D_ DIRECT DIGITAL CONTROL (DDC):-

Fig(8) below, shows a schematic Diagram of DDC.



Fig(8): DDC schematic Diagram

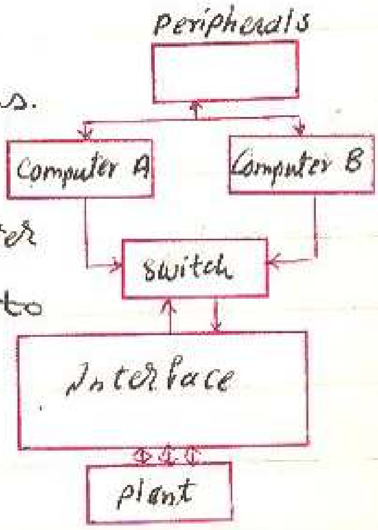
- DDC replaces the analog control with a periodically executed equivalent digital control algorithm carried out in the computer.
- The computer here is used to:
  1. Scan & digitize process variables periodically.
  2. Calculate the required control signal to reduce the difference between the set-point and process variable to zero.
  3. Output the calculated control signal in proper form to the actuator.
- The major difference between DDC & supervisory control is that the computer is directly involved with the dynamic control.
- The control calculations must be carried-out much more frequently and with more exact timing.

E: Centeralized Computer Control:-

In 1970 the cost of computer hardware had reduced such that it became possible to consider the use of dual computer system as in fig(8). In the event of the failure of one computer, the other takes over. In some schemes the changeover is MANUAL in others AUTOMATIC, failure detection and changeover is incorporated.

These falls in to two types:

E-1 Hierarchical Systems. E-2 Distributed Systems.



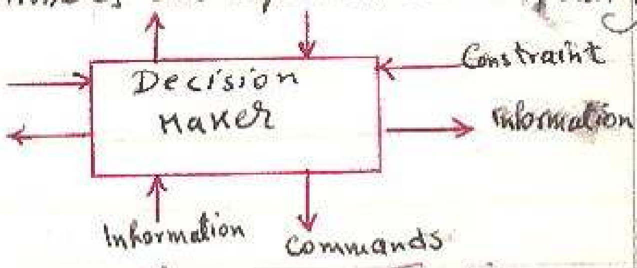
fig(9): Centeralized C.C. System.

E-1: Hierarchical Systems: In this system, Tasks are divided according to function, e.g. with one computer performing DDC calculations and being subservient to another which performs supervisory control.

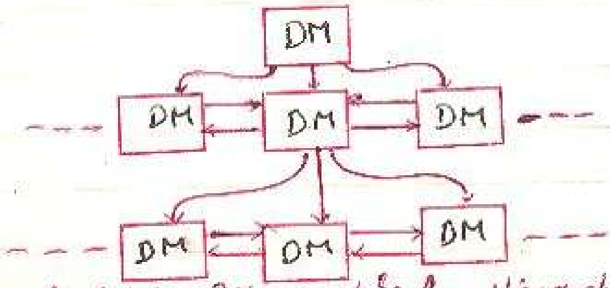
Each descision making (DM) element:

1. Recieves commands from the level above.
2. Sends Information back to the level.
3. Depending on Information recieved from elements below and constraints imposed by elements at the same level, sends commands to the elements below and Information to elements at the same level.

note: At the lowest level, the unit controllers are responsible for the operating the plant. See fig(10).



fig(10)-a DM Function.

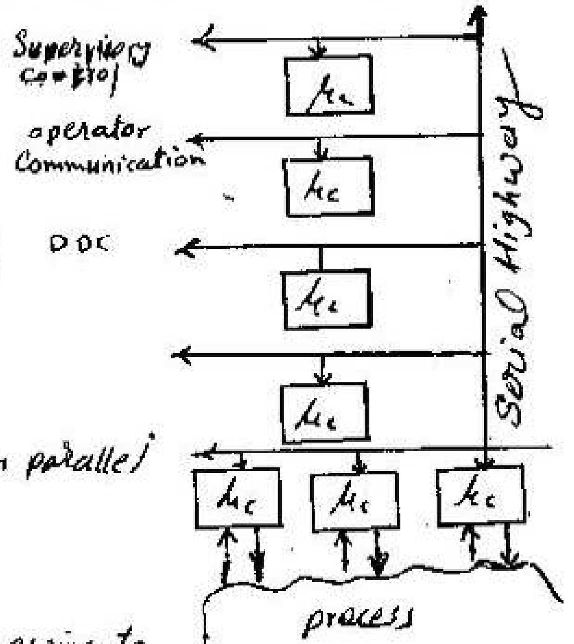


fig(10)-b- DM structure for hierarchical System.

E.2: Distributed Systems:-

In the distributed system, the total work is divided up & spread across several computers. In this case many computers perform essentially similar tasks in parallel. In the event of failure or overloading of a particular unit, all or some of the work can be transferred to other units.

In most modern computer control systems, a mixture of distributed and hierarchical approaches are used. Fig(11), shows the Distributed system scheme.



Fig(11): distributed System.

Why Distributed Control Systems?

1. Communication: Communication between computers is facilitated by a digital format using standard procedures.
2. Cost: Serial transmission is cheaper than parallel connections or analog wiring.
3. Reliability: Greater reliability is possible owing to the large numbers of processors.
4. Error: Data can be checked for errors.
5. performance: More complex control strategies can be implemented.
6. Modularity: it is easier to separate functions into logical modules, this provides for clean interfaces between modules, and allows ~~easier~~ testing and modifications.

2.2 Man_Machine Interface (MMI):

The Key to the successful adoption of a Computer Control scheme can often be the facilities provided for the plant operator. It is important that operator is provided with a simple and clear system for the day-to-day operation of the plant. All the Information relevant to the current state of its operation should be readily available and facilities to enable interaction with the plant - to change set points, to manually adjust actuators, to acknowledge alarm conditions, etc - should be provided. A large portion of the design and programming effort goes into the design and Construction of operator facilities and the major process Control equipment Companies have developed extensive schemes for the presentation of information.

The Control Engineer:

Assuming that a decision has been made on the most suitable Computer System the Control engineer's responsibility is as follows:-

1- To define the measurements and actuations and to set-up scaling and filter Constants, alarm and actuator limits, sampling intervals, etc.

2- To define the DDC Controllers, the interlinking or Cascading of such Controllers and the Connections with any other elements in the Control scheme.

3- To tune the Control scheme, i.e, to select the appropriate gains so that they perform according to some desired specifications.

4- To define and program the sequence control procedures necessary for the automation of plant operation.

5- To determine and implement satisfactory supervisory ~~control~~ schemes

3. Hardware Requirements for Real-time Applications:

3.1 Process related Interface:-

For real-time computer control, instruments and actuators connected to the process can take a wide variety of forms:

- They may be used for measuring temperature, (Thermo couples).
- They could be measuring flow-rate — (Impulse turbine).

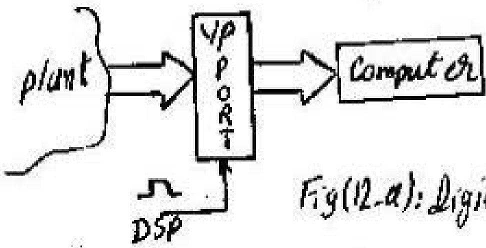
--- etc.

In all these operations, there is a need to convert a digital quantity to physical quantity, & vice-versa.

3.1.a. Digital Signal Interface:

* Digital i/p Interface:

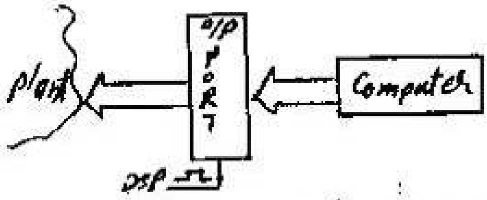
It is assumed that the plant outputs are logic signals which appear on lines connected to the digital i/p register. A device select pulse (DSP) is required for port interfacing. see. fig (12-a).



Fig(12-a): Digital i/p Interface

* Digital o/p Interface:

Digital o/p is the simplest form of o/p. A register or latch is required ~~to hold~~ the data o/p from the computer. if the computer o/p levels (0 to +5 v.) are not enough to operate the actuators on the plant, some signal conversion is necessary. This can be performed by using these low level signals to operate relays ~~which~~ carry to higher voltage signals.



Fig(12-b): Digital o/p Interface

3.1.b. Pulse Interfaces:

A Simple pulse i/p interface consists of a Counter Connected to a line from a plant. The Counter is reset under program Control, and after a fixed length of time the contents are read by the Computer.

The size of pulse Counter depends on the sensor o/p frequency and Sampling time.

for pulse o/p interface, Pulse generators can be of two types:

- They can either send a Series of pulses of fixed duration.
- OR, a Single pulse of variable length (PWM).

The operation, is to load the o/p port with the number of pulses to be transmitted, or to load the o/p port with the duty cycle of the PWM Signal.

Both, i/p & o/p pulse Interfaces are shown in fig(13) below:-

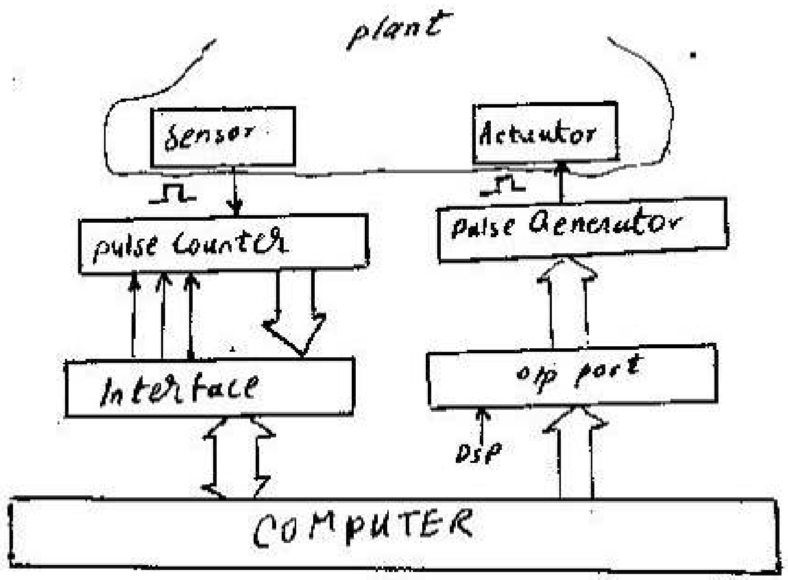


Fig (13): Pulse i/p-o/p Interfaces scheme.

3.1.C- Analog Interfaces:-

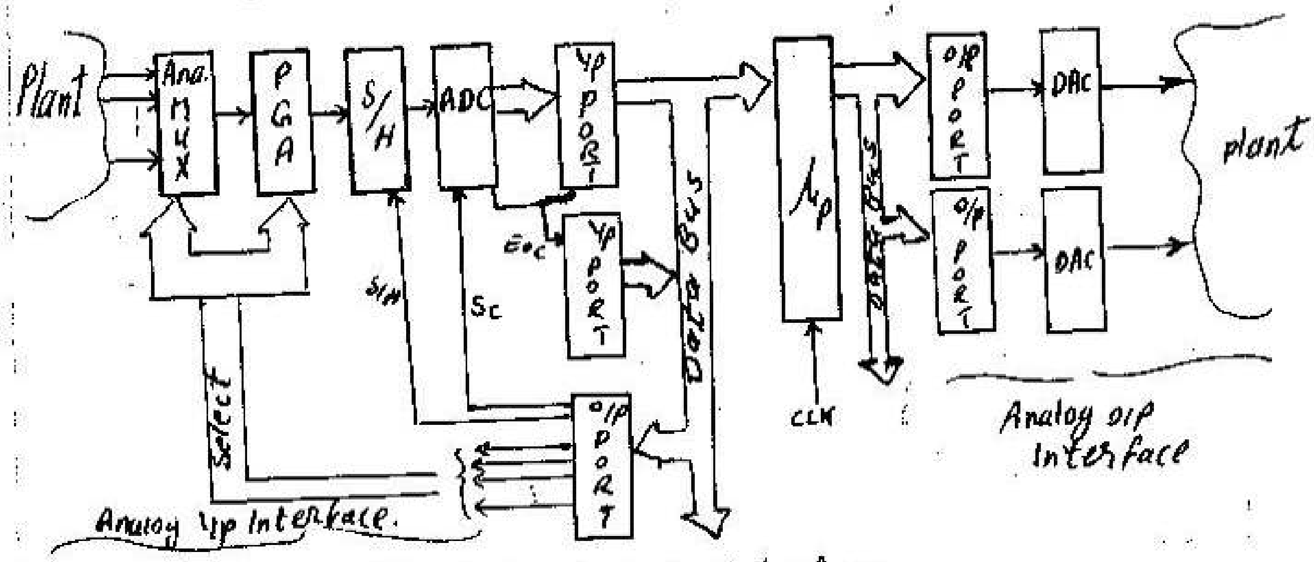


Fig (14): Analog i/p & o/p Interfaces.

Fig (14) shows An Analog i/p & o/p Interfaces one can conclude the following:-

- Separate ADCs are not normally used for each analog i/p., Instead one analoge multiplexer is required to switch the i/ps from several i/p channels to a Single ADC.
- A programmable gain Amplifier (PGA) can be used to amplify the mux. output. The gain of this amplifier is selected by the computer software according to the sensor output.
- A Sample/Hold (S/H) unit is used to prevent a change in the measured quantity while it is being converted to a discrete quantity. The Sampling time is much shorter than the sample time required for the process.

DATA ACQUISITION SYSTEM OPERATION:-

1. Select the channel and required gain.
2. Generate a Sample Signal.
3. Send the SC signal & wait for EOC.
4. Read the digital signal from ADC.

Steps 1→4 must be within Sampling time.

3.3- Data - Transfer Techniques:-

a)- Polling:-

In real-time Computer Control, a major problem in data transfer is "Timing". Under programmed transfer, the Computer can read or write at any time to a device, i.e, can make an unconditional transfer.

- For some process o/p devices which are always READY to recieve data, such as switches, LEDs, DACs, ... unconditional transfer is possible.
- For other devices, such as printers, which are not fast enough to keep up with the Computer. In this case unconditional transfer cannot be used, the Computer must always be sure that the device is READY to accept the next item of data, hence one of these methods can be used:-

a-1- Using timing loop;

A timing loop (Delay Subroutine) can be used to synchronise the Computer to the external device. To ensure that no transfer is made before the peripheral is ready, the time delay must be slightly greater than the max. delay expected in the peripheral.

a-2- Using Conditional transfer:-

A simple example of Conditional transfer is shown in fig (16). Assuming that the data is being transferred to a printer which operates at 40 chr/sec. Device is READY once every 25 msec., loop execution time $\simeq$ 8 μsec.

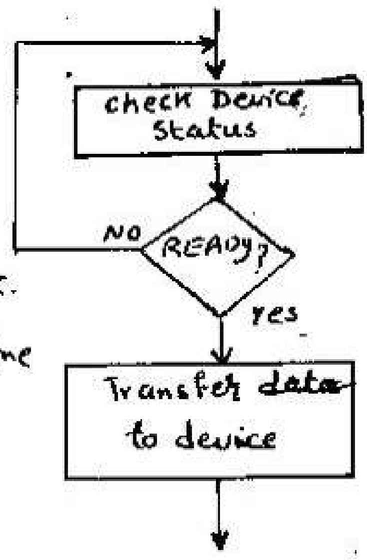
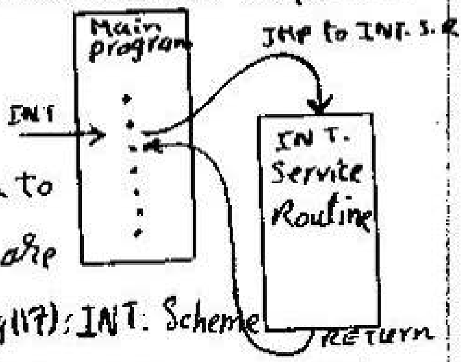


fig (16): Conditional Data Transfer

b)- Data-Transfer using INTERRUPTS :-

An iterrupt is a mechanism by which the flow of the program can be temprorarly stopped to allow a special set of instructions (Interrupt Service routine) to run. When this routine has finished, the program which has been suspended is resumed. see fig (17).

Interrupts are essential for the correct operation of most real-time computer control systems. In addition to providing a solution to the conditional wait problem they are used for:-



Fig(17): INT. Scheme

(1)- Real-time clock : External H/w provides a signal at regularly spaced intervals of time. The INT. S.R counts the signals & keeps a clock.

(2)- ALARM INPUTS:- Various sensors can be used to provide a change in a logic level in the event of an alarm. Alarms may need rapid response times; the use of interrupt provides an effective and efficient solution.

(3)- MANUAL ACTION:- Use of an interrupt can allow external control of a system to allow for maintenance and repair.

(4)- H/W FAILURE INDICATION:- Failure of interface units can be signalled to the μp by using an interrupt line.

(5)- Debugging Aids:- Interrupts can be used to insert breakpoints in the program during program testing.

(6)- Power Failure Warning:- It is useful to include in the computer system a circuit that detects very quickly the loss of power and provides a few milliseconds warning before the loss, such that the system stops working. This circuit can be connected to an interrupt line.

4- Real-time programming:-

4-1 Real-time Operating System (RTOS).

a)- Operating System (O.S.):-

- O.S. is a collection of programms that controls and sequences the excution of the user programms on a computer. It also provides the interface s/w between the user program and the H/w of the computer system. And allows the programs to communicate among themselves. i.e O.S. is a program which controls the operation of a computer itself.
- Application Program (user program): A program which is designed to carry out specific tasks which represent the end use of the computer.

a-1 Basic Elements of O.S.

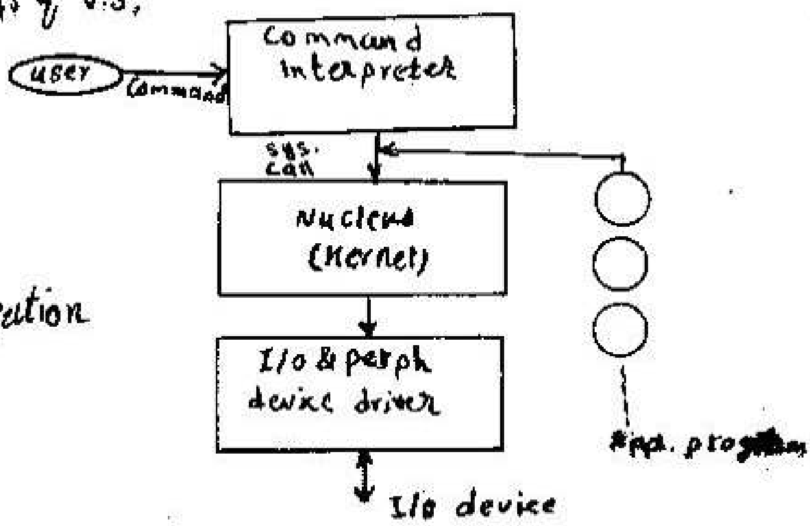
Command interpreter
"Allows user to give commands to O.S.", Command Interpreter + Application program gives a system call to the Kernel.

Nucleus (Kernel)
Supplies a logic interface between computer s/w & H/w running on the computer.

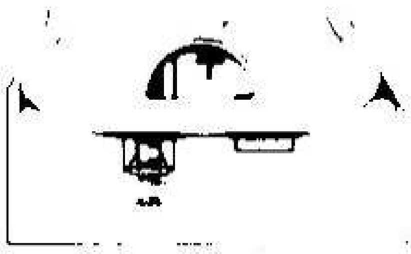
set of I/o device drives
performs the actual control of H/w perph. device it depends on the device it self and I/o Address in the computer. Hence it must be modified for every H/w configuration.

Hence: A system call is a subroutine that calls the O.S. with parameter specifing what the O.S. function to be performed.

Fig(17). shows the Basic Elements of O.S.



Fig(17): Basic O.S. organization



- are Sometimes called "Software buses" since application programms Can be plugged into any Computer system which runs a version of O.S.

a-2- O.S. Facilities:

① File system: provides file sys. to the user's application program. file hides the detail accessing tracks and sectors on the storage media from the appl. program.

② Logical I/O sys: Appl. program Can be isolated from physical details of I/O device by dealing with logical device.

③ Multitasking: O.S. gives the appearance of allowing several programs to operate simultanously on the Computer. This is called a Multitasking or Multiprogram O.S. while Multitasking O.S. that service multiusers are called multiusers or time-shared O.S.

④ Memory & resource management: For Multiuser each user must have a private memory Space and a private file sys., Also I/O device must be carefully managed if a program sending data for a printer and is suspended to allow another program that begins to run cannot get access to the printer.

⑤ Manage Passing: Transfer data from one prog. to another in Multitasking O.S. Communication links between programs are called Mailboxes, Queues, Buffers or pipes.

a-3 Examples of O.S.

① CP/M O.S.:

- Single user, Single task O.S., used with 8080, 8085, Z80 μps, - Versions of it have also been developed for 8086 & Motorolla 68000

- MSDOS O.S. for μsoft Corp. is similar to CP/M.

- The Command Interpreter called Consol Command processor (CCP).
- The kernel is called Basic Disk O.S (BDOS).
- The Basic I/O System (BIOS) refers to the set of I/O drives.

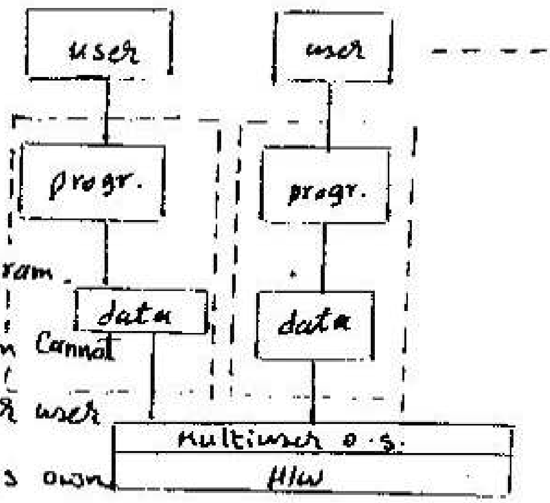
② The Unix O.S.

- It's multiuser, multitasking O.S. suitable for large development projects.
- It's large O.S. that'll not work with the 64 kbyte memory of 8-bit μp.
- A hard disk based sys. is required.
- Unix Command interpreter is called (shell).

a-4 Multiuser & Multitasking O.S.

① Multiuser O.S.

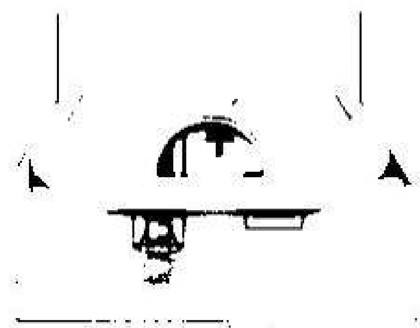
- This O.S. ensures that each user can run a single program as if they had the whole of the computer sys. for their program.
- This O.S. ensures that one user program cannot interfere with the operation of another user program. Each user program runs in its own protected environment.
- At any given instant, it is not possible to predict which user will have the use of the CPU or even if the user's code is in the memory.



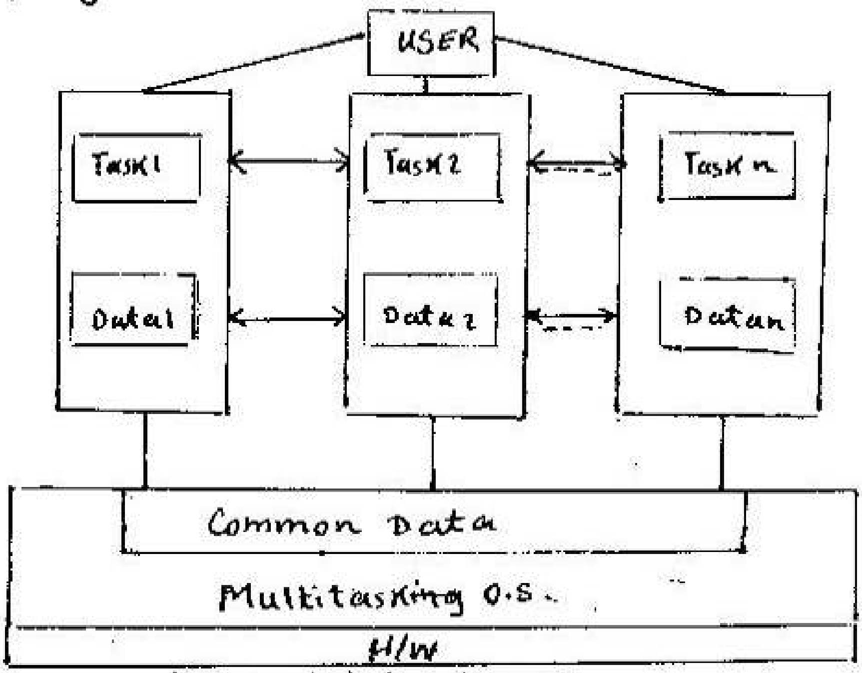
Fig(18): Multiuser O.S.

② Multitasking O.S.

- It is assumed that there's a single user.
- Various tasks are to cooperate to serve the requirement of the user.
- operation will require that the tasks communicate each other and share common data.



- A task may require certain activities which are contained in another task to be performed and it may itself be made by other task.
- The tasks may need to communicate with each other. The O.S. therefore has to have some means of enabling tasks, either to share memory for the exchange of data or to provide a mechanism by which tasks can send message to each other.
- Tasks may need to be invoked by external events, hence the O.S. must support the use of interrupts.
- Tasks ~~may~~ need to share data and they may require access to various H/w & S/w components. Hence there has to be a mechanism for preventing two tasks from attempting to use the same resources at the same time.



Fig(19): Multitasking O.S.

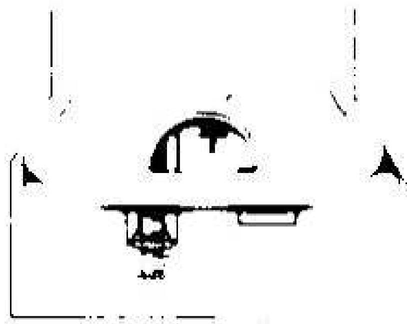
b): Real-time O.S.:

- A Real-time O.S. differs from normal multitasking O.S. by:-

① Its ability to schedule tasks on the basis of external events caused by interrupts.

② Their command interpreter are usually simple and they don't include any utility programs.

③ RTOS is usually used to serve the sys. it controls rather than human users.



Task:

- def. Task is an activity carried out by the Computer. it consists of a program, data associated with the program and computer resources such as memory or I/O devices required to execute the program.

- Task states:- with Computer of one Cpu only one task (activity) Run at a time. The other tasks must be in some state other than Running.

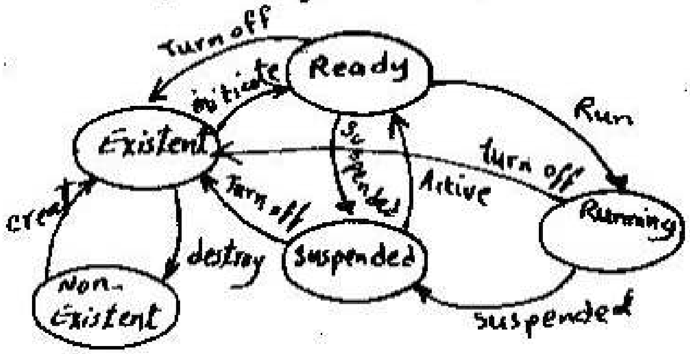
① Active (Running): it has the control of the Cpu. it has the highest priority of the tasks which are ready to run.

② Ready: Several tasks may be in this state.

③ Suspended: (waiting, delayed): The execution is delayed because * Task requires some resource which is not available or * Because task is waiting for some signal from the plant.

④ Existent: The O.S. is aware of the existance of this task, but the task has not been allocated a priority and has not been made runnable.

⑤ Non-existent: (Terminated): The O.S. has not as yet been made aware of the existance of this task, although it may be resident in the memory of the Computer.



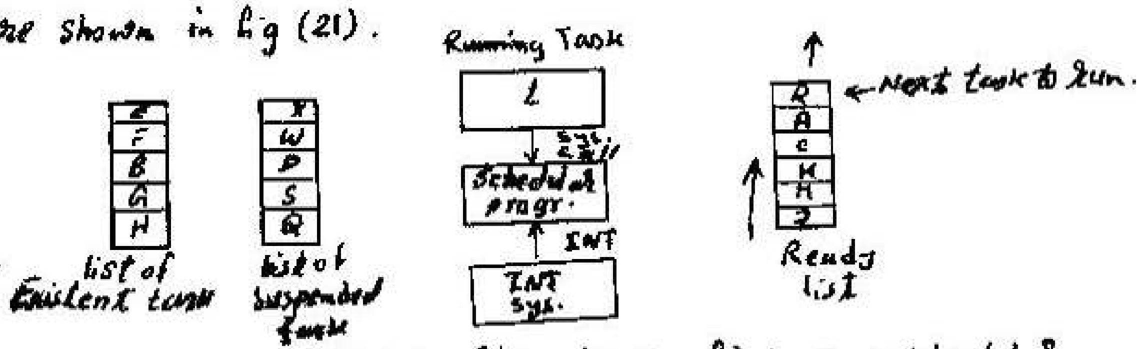
Fig(20): Task State Transition Diagram

b-2: Task Descriptor (TD): Each task in the Sys. has status information held in a TD. The Information held in the TD will vary from Sys. to Sys. but will typically consist of the following.

- Task identifier was un add. to task on a disc.
- Task priority.
- Current state of task.
- Area to store volatile environment.

b-3: Task Schedular:

It is a program which is responsible for Controlling the transitions of the tasks among their states. The elements of a Simple task schedualar are shown in fig (21).



Fig(21): Elements of a Simple Task schedular

When Running task stoped the task on the Ready list is started.
A Running task will Continue Running Until one of the following events occurs

① The task makes a Sys. Call asking to be suspended Until an event (INT Request) occurs.

② The task makes a Sys. call requesting a sys. resource which is unavailable. The task will then be placed in the list of suspended tasks

(3) The task makes a sys. call asking to be rescheduled, since it has completed its function. The task will then be placed in the Ready list or the suspended list.

(4) The schedular preempts the task to allow another task to execute. The task is then placed in the Ready list.

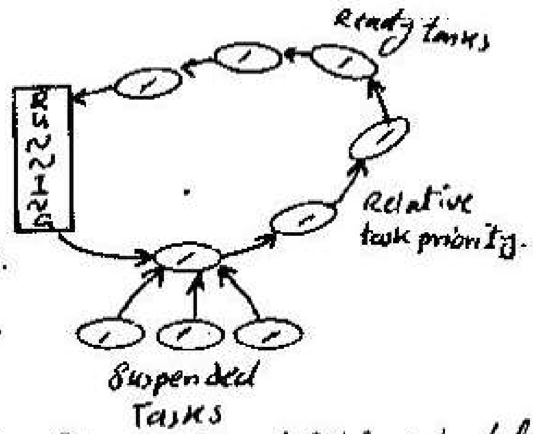
When one of these events, occurs, the schedular select the next task from the Ready list and cause the computer to start executing it.

6-4: Scheduling Strategies:-

The schedular's ability to Control the execution of tasks is the key to the efficiency and speed response of a RTOS.

Strategies used by Task schedular :-

(1) Round-Robin scheduling:-



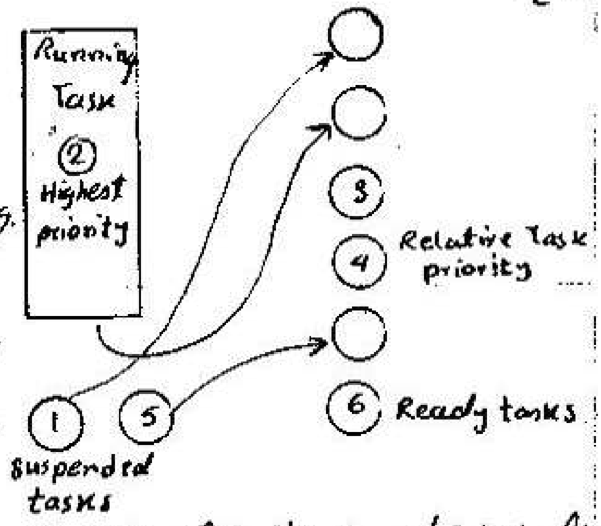
- All task have Equal priority.
- The Ready list is Configured as a Simple FIFO.
- The schedular gives each task an Equal time-slice for execution.
- The length of time-slices is usually determined by real-time clock which interrupts the schdular at regular intervals. when clk interrupt takes place the scheduler stopping execution of the currently running task and starting execution of the next task in the Ready list.

Fig(22): Round Robin Scheduling.

Disadvantage:- Response to external events such as INT. can be rather slow, because it must wait until all tasks infront of it have recieved their time-slices before it can begin execution.

② Priority-Based Preemitive Schedualing:

- Tasks are amigned different priorities.
- The highest priority task that is not in the suspended or Existent state is always Running.
- A task stops running only when it must wait for an event or a resource. The task with the next highest priority can the execute.



Fig(23) : Priority-Based Schedualing

- advantage:- guarantees very fast response to interrupts.
- disadvantages: Highly priority tasks that rarely become suspended may hog the Cpu preventing the lower-priority tasks from executing.

- In practice a mixture of Round-Robin & Priority-based schedualing is often used. In this case INT. SERV. Routine are amigned a high priority. Other tasks are amigned equal priorities which are lower than that amigned to the INT SERV. Routine. These low-priority tasks are schedualed in Round-Robin fashion. However the Highest-priority INT. SERV. Routines are schedualled in a priority-based fashion.

4_2: Real_time Languages:-

a_ Real_time Software Requirements:-

① Real-time s/w should be reliable, the failure of RT system can be expensive both in terms of lost components or the loss of human life.

② Real-time system are large and complex, which makes developments and maintenance costly.

③ Real-time systems have to respond to external events with a guaranteed response time.

④ Real-time systems should involve a wide range of interference devices, including non-standard devices.

b_ USER REQUIREMENTS:-

The user requirements for a real-time language can be divided into:-

① Security:- Security is a measure of the extent to which a language ~~is~~ able to detect errors automatically either at compile time or ~~through~~ the run-time.

Economically, it is important to detect errors at compilation stage rather than at run-time, since the earlier the error is detected the less it costs to correct it.

② Readability: The readability of a program is a measure of the ease with which its operation can be understood without resort to supplementry documentation such as flowcharts or natural language description. The benefit of good readability are:-

- Reduction in documentation costs.
- ~~Easy~~ Error detection.
- Easy maintenance.

③ Flexibility:- For a general purpose language, the programmer should be able to express all the operations required in a program without the need to use assembly coding. The flexibility of a language is a measure of this facility.

④ Simplicity:- In language design, the simple is to be preferred to the complex.
- Simplicity contributes to security.
- It reduces the cost of training.
- It reduces the probability of programming errors.
- It reduces compiler size and leads to more efficient object code.

⑤ Portability:- The ability to transfer a program from one computer to another and it will compile and run on the computer to which it has been transferred.

⑥ Scope & Visibility:-
- The scope of a variable is defined as the region of a program in which the variable is potentially accessible or modifiable..
- The regions in which it may actually be accessed or modified are the regions in which it is said to be visible.

⑦ Independent & Separate Compilation:-
One of the reasons for the popularity & widespread of use of FORTRAN for engineering and scientific work is the fact that subroutines can be compiled independently from the main program, and from each other.

⑧ Exception Handling:-
One of the functions of the run-time support system is to trap errors which have not been anticipated. In a real-time system every attempt must be made to keep the system running. This can be done by using error trapping mechanism which can be set to pass control to a routine provided by the system designer when a run-time error is detected.

(9)_ Efficiency:-

- In the early Computer Control systems great emphasis was placed on efficiency of the coding (size of the object code and speed of operation), as Computers were both expensive and slow.

- The falling costs of H/w and the increase in the Computational speed of Computers has changed this emphasis. Also in a large number of RT applications the concept of an efficient language has changed to include:

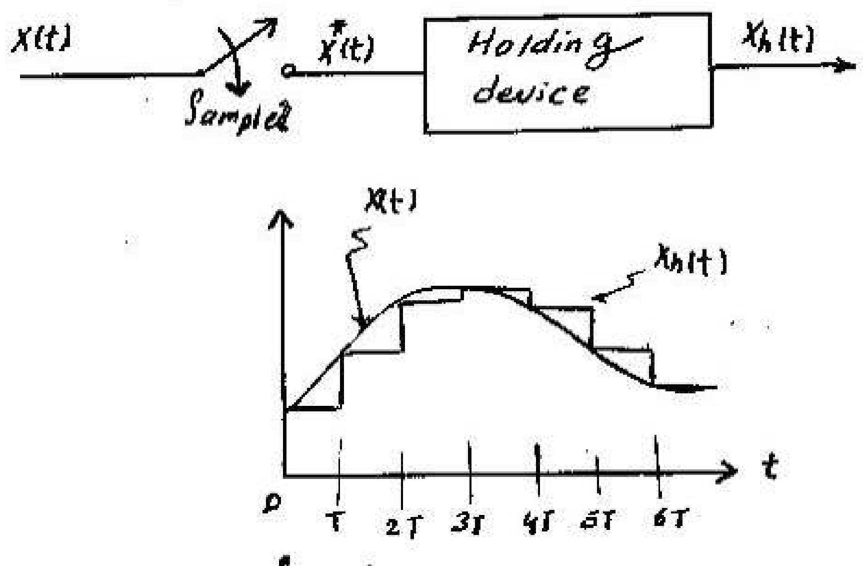
* The security and the costs of writing and maintaining the program.
* Speed and Compactness of the object code have become of secondary importance.

2- Sampled- Data Control Systems:-

2.1 Sampling and Reconstruction:-

The essential element of a discrete- time system is the sampler. In a conventional sampler, a switch closes to admit an input signal every T seconds. In practice, the sampling duration is very short in comparison with the most significant time constant of the plant. A sampler converts a continuous signal into a train of pulses occurring at the sampling instants 0, T, 2T, ... where T is the sampling period. (Between sampling instants, the sampler transmits no information.).

A holding device converts the sampled signal into a continuous signal, which approximately reproduces the signal applied to a sampler. The simplest holding device converts the sampled signal into one which is constant between two consecutive sampling instants. The transfer function G_h of a zero- order holding device is $G_h = \frac{1-e^{-Ts}}{s}$.



Fig(1): Signals before & after the Sampler & holding device.

2.2. The effect of Z.O.H on the performance of a Control System.

$G_h = \frac{1-e^{-Ts}}{s}$, Can be approximated by a Transfer function which is a ratio of polynomial in S. Knowing that $e^{-Ts} = \frac{1-\frac{Ts}{2}+\frac{(Ts)^2}{8}-\cdots}{1+\frac{Ts}{2}+\frac{(Ts)^2}{8}+\cdots} \simeq \frac{1-\frac{Ts}{2}}{1+\frac{Ts}{2}}$

$$\therefore G_h = \frac{1-e^{-Ts}}{s} \simeq \frac{1}{s}\left[1-\frac{1-\frac{Ts}{2}}{1+\frac{Ts}{2}}\right] = \frac{T}{1+\frac{T}{2}s}$$ $\therefore G_h$ add a pole at $s=-\frac{2}{T}$

(i.e) when T = Sampling time is chosen to be very short the effect of G_h will be very small (far pole is added and its effect is neglected).

$G_h = \frac{1}{1+T/2s}$ will be used since (T) is the sampler duration.

2.3: Shannon's Sampling Theorem:-

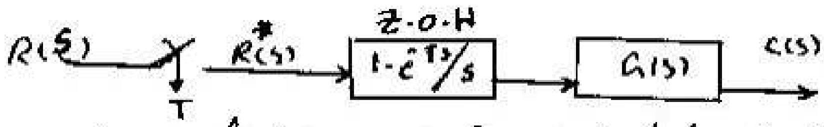
The choice of Sampling period T is approximately one-half to one-tenth, depending on the circumstances, of the smallest significant time constant involved in the plant.

2.4 Analysis of Discrete Control Systems:-

a) O/L Systems.

In order to develop an expression for the Z-transform of the o/p of open-loop Systems." This expression will be required later when we form closed-loop Systems by feeding back this output signal & Sampling it. Consider the open-loop system of fig(2). Note that $G(s)$ must contain the transfer function of the data hold. It is seen that

$$\frac{C(z)}{R(z)} = Z\{G_h(s) \cdot G(s)\}.$$



fig(2): O/L Sampled-data System.

Ex:- Given $G(s) = \frac{1}{s+1}$ in fig(2), determine the o/p $C(z)$ for Unit step change in input.

Sol. $\frac{C(z)}{R(z)} = Z\left[\frac{1-e^{-Ts}}{s} \cdot \frac{1}{s+1}\right] = (1-z^{-1}) Z\left[\frac{1}{s(s+1)}\right]$

$$\frac{1}{s(s+1)} = \frac{A}{s} + \frac{B}{s+1} , \quad 1 = As + A + Bs , \quad \therefore A = 1, \ B = -1$$

$$\therefore Z\left[\frac{1}{s} - \frac{1}{s+1}\right] = \frac{z}{z-1} - \frac{z}{z-e^{-T}}$$

$$\therefore \frac{C(z)}{R(z)} = \left(1 - \frac{1}{z}\right) \cdot \left(\frac{z}{z-1} - \frac{z}{z-e^{-T}}\right) = \frac{z-1}{z} \cdot \left(\frac{z}{z-1} - \frac{z}{z-e^{-T}}\right) = 1 - \frac{z-1}{z-e^{-T}}$$

for $R(z) = \frac{z}{z-1}$, $\therefore C(z) = \frac{z}{z-1} - \frac{z}{z-e^{-T}}$, $\therefore c(kT) = 1 - e^{-kT}$

b): Pulse transfer functions of closed-loop Systems.

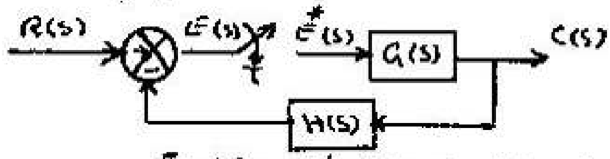
Consider the closed-loop System shown in fig (3). In this System, the actuating error is sampled. from the block diagram:-

$E(s) = R(s) - H(s)C(s)$, $C(s) = G(s).E^*(s)$

Hence, $E(s) = R(s) - G(s).H(s).E^*(s)$

$\therefore E^*(s) = R^*(s) - GH^*(s)\, E^*(s)$

$\therefore E^*(s) = \frac{R^*(s)}{1+GH^*(s)}$, Since $C^*(s) = G^*(s) E^*(s)$

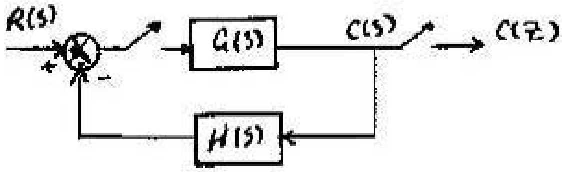


Fig(3): c/L discrete-time System.

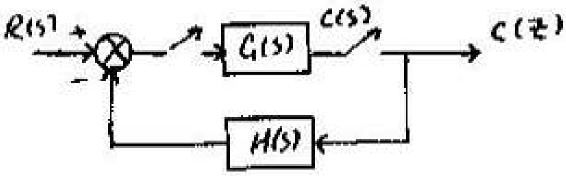
$$\therefore C^*(s) = \frac{G^*(s) R^*(s)}{1+GH^*(s)} \qquad \therefore C(z) = \frac{G(z) R(z)}{1+GH(z)} \quad \text{OR} \quad \frac{C(z)}{R(z)} = \frac{G(z)}{1+GH(z)}$$

Therefore the Pulse Transfer Function of fig(3) is $\frac{C(z)}{R(z)} = \frac{G(z)}{1+GH(z)}$.

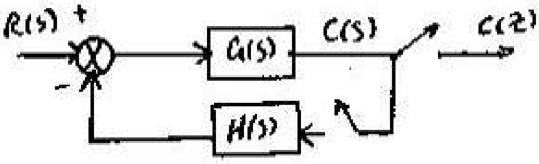
Fig(4): shows five typical Configurations of closed-loop discrete-time Systems. For each Configuration, the Corresponding output $C(z)$ is shown.



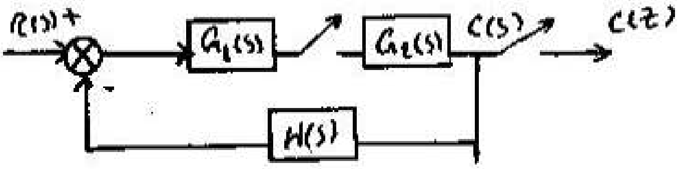
$$C(Z) = \frac{G(Z).R(Z)}{1+GH(Z)}.$$



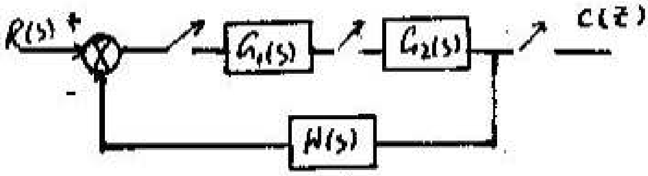
$$C(Z) = \frac{G(Z).R(Z)}{1+G(Z).H(Z)}.$$



$$C(Z) = \frac{RG(Z)}{1+HG(Z)}.$$



$$C(Z) = \frac{G_2(Z)\,RG_1(Z).}{1+G_1G_2H(Z)}.$$



$$C(Z) = \frac{G_1(Z)\,G_2(Z)\,R(Z)}{1+G_1(Z)\,G_2H(Z)}.$$

Fig(4): Typical Configurations of closed-loop Discrete-Time Systems And the Corresponding outputs C(Z).

This comes from the fact that the Pulse transfer functions of Cascaded elements to clarify the idea see fig(5) below:-

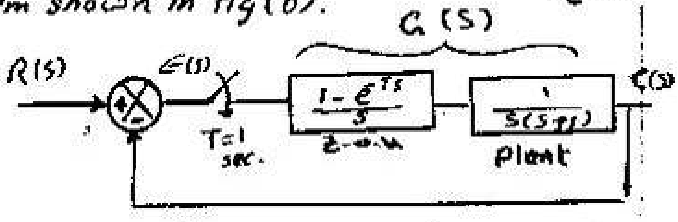
$$G_1G_2(Z) \neq G_1(Z).G_2(Z)$$

fig(5): P.T.F of Cascaded Elements.

Therefore, we must be careful and observe whether or not there is a sampler between cascaded Elements.

2.5: Time-Response:

Ex:- obtain the Unit-step response of the system shown in fig (6).



fig(6): C/L discrete-time system

Sol. $\frac{C(z)}{R(z)} = \frac{G(z)}{1+G(z)}$ $\quad H(z)=1$

$$G(s) = \frac{1-e^{-TS}}{s^2(s+1)} = (1-e^{-TS}) \cdot \frac{1}{s^2(s+1)}$$

$$\mathcal{Z}[G(s)] = G(z) = (1-z^{-1}) \cdot \mathcal{Z}\left\{\frac{A}{s^2} + \frac{B}{s} + \frac{C}{(s+1)}\right\}$$

$$\therefore G(z) = \frac{0.368z + 0.264}{z^2 - 1.368z + 0.368} \quad \text{for } T = 1\,\text{sec.}$$

Thus: $\frac{C(z)}{R(z)} = \frac{0.368z + 0.264}{z^2 - z + 0.632}$. for $R(z) = \text{Unit-step} = \frac{z}{z-1}$.

$$\therefore C(z) = \frac{0.368z^{-1} + 0.264z^{-2}}{1 - 2z^{-1} + 1.632z^{-2} - 0.632z^{-3}} = 0.368z^{-1} + z^{-2} + 1.4z^{-3} + 1.4z^{-4} + 1.147z^{-5} + 0.895z^{-6} + 0.802z^{-7} + \cdots$$ (by long division).

$\therefore$ $C(0)=0$, $C(1)=0.368$, $C(2)=1$, $C(3)=1.4$, $C(4)=1.4$, $C(5)=1.147$, $C(6)=0.895$, $C(7)=0.802$ + ——

كما بالرسم (ش)

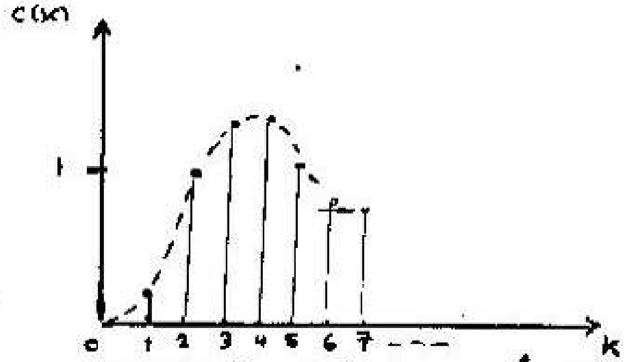


fig (7): Transient-Response for Unit-step input.

As for Computer Implementation:-

① off-line Simulation:

$$\frac{C(z)}{R(z)} = \frac{0.368z^{-1} + 0.264z^{-2}}{1 - z^{-1} + 0.632z^{-2}}$$

$\therefore C(KT) = C(K-1)T - 0.632\,C(K-2)T + 0.368\,R(K-1) + 0.264\,R(K-2)$

$K=0$, $C(0)=0$,

$K=1$, $C(1) = \cancel{C(0)} + 0.368R(0) + 0.264\cancel{R(-1)} = 0.368$.

$C(2) = C(1)T - 0.632\,C(0) + 0.368\,R(1) + 0.264\,R(0) = 0.368 + \cancel{0.368} + 0.264 = 1$.

In general for $K \geq 2$, $C(K)T = C(K-1)T - 0.632\,C(K-2)T + 0.368 + 0.264 \Rightarrow$ to be programmed.

② on-line Simulation or Implementation.

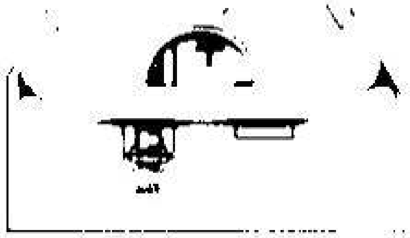
$R=1$, $C(0)=0$, $E(0) = R(0) - C(0) = R = 1$.

$\therefore E(K)T = R(K)T - C(K)T$ ——— ①

$$G(z) = \frac{C(z)}{E(z)} = \frac{0.368z^{-1} + 0.264z^{-2}}{1 - 1.368z^{-1} + 0.368z^{-2}}$$

$\therefore C(K)T = 1.368\,C(K-1)T - 0.368\,C(K-2)T + 0.368\,E(K-1)T + 0.264\,E($[illegible] ——②

[illegible] updated from (1), Eq.(2) is used for Simulation & for Real-time Implementation. C[illegible] is read from field & Error is calculated from (1). Then apply Control Algorithm.



2.6: Steady_state Error Analysis:-

An important c/cs of a control system is its ability to follow, or track, certain inputs with a min. of error. The control system designer attempts to minimize the system error to certain inputs. In this section the evaluation of steady-state error is considered.

$$\because \quad \frac{C(z)}{R(z)} = \frac{G(z)}{1+G(z)} \quad \text{for } H(z)=1, \quad \& \quad E(z) = R(z) - C(z)$$

Then $E(z) = \dfrac{R(z)}{1+G(z)}$ $\quad \therefore e_{ss}(kT) = \lim\limits_{z\to 1}(z-1)E(z) = \lim\limits_{z\to 1}(z-1)\cdot\dfrac{R(z)}{1+G(z)}$

provided that $e_{ss}(kT)$ has a final value.

The steady_state errors will now be derived for two common inputs $R(z)$:

* A position (step) input: $R(z) = \dfrac{z}{z-1}$

$$\therefore e_{ss}(kT) = \lim_{z\to 1}(z-1)\cdot\frac{z}{(z-1)[1+G(z)]} = \lim_{z\to 1}\frac{z}{1+G(z)} = \frac{1}{1+\lim\limits_{z\to 1}G(z)}$$

Now, define the position error constant $K_p = \lim\limits_{z\to 1} G(z)$ then $e_{ss}(kT) = \dfrac{1}{1+K_p}$.

Now, Recall $G(z)$ of Ex. in 2.5 $\quad G(z) = \dfrac{0.368z + 0.264}{z^2 - 1.368z + 0.368}$ then $K_p = \lim\limits_{z\to 1} G(z) = \infty$

$\therefore e_{ss}(kT) = \dfrac{1}{1+K_p} = 0$ this is true for type (1) 2^nd^ order system subjected to step input.

* Ramp Input $r(t) = t$, $\therefore R(z) = \dfrac{Tz}{(z-1)^2}$.

$$\because e_{ss}(kT) = \lim_{z\to 1}(z-1)\cdot\frac{R(z)}{1+G(z)} = \lim_{z\to 1}(z-1)\cdot\frac{Tz}{(z-1)^2[1+G(z)]} = \frac{T}{\lim\limits_{z\to 1}(z-1)G(z)}$$

Now, define the velocity error constant $K_v = \lim\limits_{z\to 1}\dfrac{1}{T}\cdot(z-1)\cdot G(z)$

Then $\quad e_{ss}(kT) = \dfrac{1}{K_v}$.

$\& \quad e_{ss}(kT) = \dfrac{1}{K_v} = \dfrac{T}{K_{d.c.}} = \dfrac{T}{K_p}$.

The development above illustrates that, in general, increased system gain ($K_p = K_{d.c.}$) and/or the addition of poles at $z=1$ to the open_loop transfer function tend to decrease steady-state errors.

2.7: Mapping S-plane / Z-plane:

A linear dynamic system is stable if all poles of the transfer function lie in the left-half s-plane. In the Z plane, the left-half s plane corresponds to the unit circle centered at the origin, or the left-half s plane maps into the inside of the Unit Circle in the Z-plane. This can be proved easily.

Since $Z = e^{Ts}$, $s = \sigma + j\omega$ we obtain $|Z| = e^{T\sigma}$, $\angle Z = \omega T$

In the left-half s-plane, $\sigma < 0$. Therefore, the magnitude of Z varies between 0 & 1. while The Imaginary axis, or $\sigma = 0$, corresponds to the Unit Circle in the Z-plane. when $\sigma > 0$ [unstable s plane] the $|Z| > 1$ i.e the location is outside the Unit Circle.

Note that, since $\angle Z = \omega T$, the angle of Z varies from $-\infty$ to ∞ as ω varies from $-\infty$ to ∞.

2.8: Stability Analysis:-

(a) Bilinear Transformation:-

A few methods are available for determining whether or not a polynomial in Z contains a root or roots on or outside the Unit Circle. One Method is the Bilinear Transformation, which is used to modify the Routh stability criterion. The Routh stability criterion tells us whether or not any of the roots of a polynomial lie in the right half of the Complex plane. Since the following transformation $Z = \frac{r+1}{r-1}$ maps the Unit circle in the Z-plane to the left r plane, with this transformation the Routh stability criterion may be applied to the polynomial in the r domain in the same manner as continuous-time systems.

(b): Routh - stability criterion:-

Ex:- if the f/f pulse Transfer function of a Unity f/b system is
$G(Z) = \frac{10(1-e^{-T})Z}{(Z-1)(Z-e^{-T})}$ for T=1, The c/cs Equation $1+G(Z)=0$ is:

$Z^2 + 4.952 Z + 0.368 = 0$ the roots are $Z = -0.076$, $Z = -4.876$ (unstable) $|Z|>1$.

to apply Routh - stability Criterion using the Bilinear Transformation
$Z = \frac{r+1}{r-1}$ we obtain: $\frac{(r+1)^2}{(r-1)^2} + 4.952\left(\frac{r+1}{r-1}\right) + 0.368 = 0$

or; $6.32 r^2 + 1.264 r - 3.584 = 0$

r^2	6.32	-3.584
r^1	1.264	0
r^0	-3.584	

This one change of sign in the first column of the Routh array, indicates that there is one root in the right-half r plane, which implies that there is one root outside the Unit Circle in the Z-plane. and this corresponds to the result obtained previously.

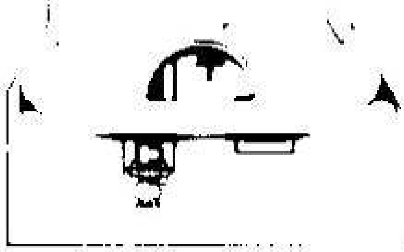
Note:- check stability for G(Z) with T=0.1 sec.

(c):- Jury - Test of stability of Discrete-time Systems.

Consider the following C/L P.T.F $\frac{C(Z)}{R(Z)} = \frac{G(Z)}{1+GH(Z)}$. The stability of the Sys. as defined by the above Equation is determined from the locations of the poles in the Z-plane (i.e the roots of the C/cs Equation) $P(Z) = 1+GH(Z) = 0$.

$P(Z) = 1 + GH(Z) = 0$

- **Stable**: if the C/L poles located inside the Unit circle.
- **Critically stable**: if a simple pole or poles located at $Z = 1, -1$ or one pair of Complex Conjugate poles located at the Unit circle.
- **Unstable**: if one of the poles of the C/L poles located outside the Unit circle or there are two or more Multiple Complex Conj. poles at the Unit Circle.



The procedure of Jury-Test is as below:- Ex: $P(z) = 1 + GH(z) = a_4 z^4 + a_3 z^3 + a_2 z^2 + a_1 z + a_0 = 0$

(1) $|a_0| < |a_4|$. if satisfied then go to (2).

(2) $p(z)\big|_{z=1} > 0$ if satisfied go to (3).

(3) $p(z)\big|_{z=-1}$ > 0 for n = order = even. (here $n=4$).
< 0 for n = order = odd. if (3) is satisfied go to (4).

(4) $|b_0| > |b_3|$ This condition comes from the array shown below:-
$|c_0| > |c_2|$

	z^0	z^1	z^2	z^3	z^4
①	a_0	a_1	a_2	a_3	a_4
②	a_4	a_3	a_2	a_1	a_0
③	b_0	b_1	b_2	b_3	
④	b_3	b_2	b_1	b_0	
⑤	c_0	c_1	c_2		

no. of rows are equal to $2n-3$.
for $n=4$. $\therefore$ Rows No. $= 8-3 = 5$.

where $b_0 = \begin{vmatrix} a_0 & a_1 \\ a_4 & a_3 \end{vmatrix}$, $b_1 = \begin{vmatrix} a_0 & a_2 \\ a_4 & a_2 \end{vmatrix}$, $b_2 = \begin{vmatrix} a_0 & a_3 \\ a_4 & a_1 \end{vmatrix}$, $b_3 = \begin{vmatrix} a_0 & a_4 \\ a_4 & a_0 \end{vmatrix}$

$c_0 = \begin{vmatrix} b_0 & b_1 \\ b_3 & b_2 \end{vmatrix}$, $c_1 = \begin{vmatrix} b_0 & b_2 \\ b_3 & b_1 \end{vmatrix}$, $c_2 = \begin{vmatrix} b_0 & b_3 \\ b_3 & b_0 \end{vmatrix}$.

as for previous Example in (a), $p(z) = z^2 + 4.952z + 0.368 = 0$

① $|0.368| < |1|$ satisfied.

② $p(z)\big|_{z=1} = 1 + 4.952 + 0.368 > 0$ satisfied.

③ $p(z)\big|_{z=-1} > 0$ for $n=2$ (even), $= 1 - 4.952 + 0.368 < 0$ is not satisfied

$\therefore$ $p(z)$ is unstable c/cs Eq. [Note: check stability for $T = 0.1$ sec. for the same c/cs].

No. of Rows $= 2n-3 = 1$ (gives no information) (i.e) no array.

Ex:- Test the stability of the following c/cs Equations:-

① $z^4 - 1.2z^3 + 0.07z^2 + 0.3z - 0.08 = 0$ (stable), ② z^3 [illegible] $- 0.1z + 0.2 = 0$ (unstable).

③. $z^3 - 1.3z^2 - 0.08z + 0.24 = 0$ (unstable). ④

$G(z) = \dfrac{k(0.3179z + 0.2642)}{(z-1)(z-0.3679)}$ (for Stability $2.3925 > k > 0$.

2.9:- Transient Response Specifications:-

Absolute stability is a basic requirement of all control systems. In addition, good relative stability and steady-state accuracy are also required of any control system, whether continuous-time or discrete-time.

In many practical cases, the desired performance cks of control systems, whether they are continuous-time or discrete-time, are specified in terms of time-domain quantities. This is because system with energy storage cannot respond instantaneously and will always exhibit transient response whenever they are subjected to inputs or disturbances. Frequently, the performance cks of a control system are specified in terms of the transient response to a unit step input, since the unit step input is easy to generate and is sufficiently drastic to provide useful information on both transient response & the steady-state response cks of the system. The transient response specifications are summarized as below:-

for $s = \sigma + j\omega_d = -\zeta\omega_n + j\omega_n\sqrt{1-\zeta^2}$

$\therefore |Z| = e^{\sigma T}$, $\angle Z = \omega_d T$ since $Z = e^{sT} = e^{(\sigma \mp j\omega_d)T} = e^{\sigma T}(\cos\omega_d T \mp j\sin\omega_d T)$

Specifications	Theoritically	Graphically $= e^{\sigma T} \angle \omega_d T$
1- Rise-time (t_r)	$t_r = \dfrac{\pi - \tan^{-1}\omega_d/\sigma}{\omega_d}$	t_r = first value of o/p at 90% of i/p.
2- Peak-time (t_p)	$t_p = \dfrac{\pi}{\omega_d} = \dfrac{\pi}{\omega_n\sqrt{1-\zeta^2}}$	t_p = time at which maximum o/p occurs.
3- Maximum-overshoot M_p	$M_p = e^{-(\sigma/\omega_d)\pi} = e^{-(\zeta/\sqrt{1-\zeta^2})\pi}$	M_p = Max. value of o/p from i/p.
4- Settling time t_s,	(a) $T_{settl.}$ for 2% criteria $= \dfrac{4}{\sigma} = \dfrac{4}{\zeta\omega_n}$	(a) T_s 2%, time when response (o/p) get in tolerance of ~~2% of~~ input & doesn't get out of it.
	(b) $T_{settl.}$ for 5% criteria $= \dfrac{3}{\sigma} = \dfrac{3}{\zeta\omega_n}$	(b) when response gets 5% from i/p value & stay within it.

2.10: Problems:-

1- (a) obtain the P.T.F of a Unity f/b C/L System with $G(s) = \dfrac{K}{s(s+a)}$.

Ans: $$\frac{C(z)}{R(z)} = \frac{K[(aT-1+e^{-aT})z + (1-e^{-aT}-aTe^{-aT})]}{a^2z^2+[K(aT-1+e^{-aT})-a^2(1+e^{-aT})]z + (1-e^{-aT}-aTe^{-aT}+a^2e^{-aT})}$$

(b) write a Computer program to simulate $c(z)$. with $R(z)$ = Unit step.

(c) check the values $K = 0, 1, ---5$, with $a=1$, $T=1$ sec. both manually & through Computer simulation.

(d) check $e_{ss}(KT)$ Compare your results.

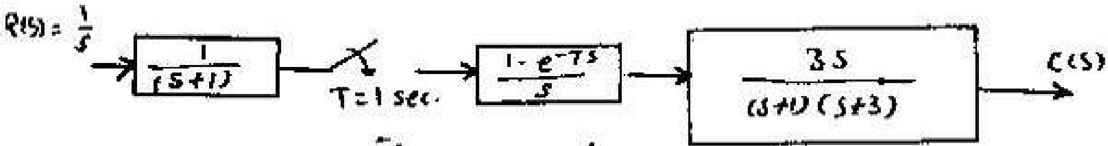
(e) check stability using two methods, then Compare your results.

2- for $G(z) = \dfrac{K(1-e^{-T/T_1})z}{(z-1)(z-e^{-T/T_1})}$ prove that the system is stable iff $0 < K < \dfrac{2(1+e^{-T/T_1})}{(1-e^{-T/T_1})}$ using two methods.

3- With $T=0.1$ sec., find the z-transform of ① $G(s) = \dfrac{(s+1)}{(s-1)(s+2)}$.
② $G(s) = \dfrac{1}{s(s+1)^2}$, ③ $G(s) = \dfrac{(s+1)}{(s^2+25)}$

then Compare poles location in both s-plane & z-plane. & check stability for both Continuous & discrete Representation. compare!

4- Find the System response at the Sampling instants to a Unit step input for the System of Fig.(8) below:-



Fig(8): B.D of problem 4.

for C/L Unity f/b System with $G(s) = \dfrac{k}{(s+a)(s+b)}$ ① find $G(z)$. ② for $a=0.3$ $b=0.5$, $T=1$sec., check stability ③ check e_{ss}. for both $R(z)$ = step, Ramp.