Introduction to Control Systems

Dr. ASHRAF E. SUYYAGH

THE UNIVERSITY OF JORDAN

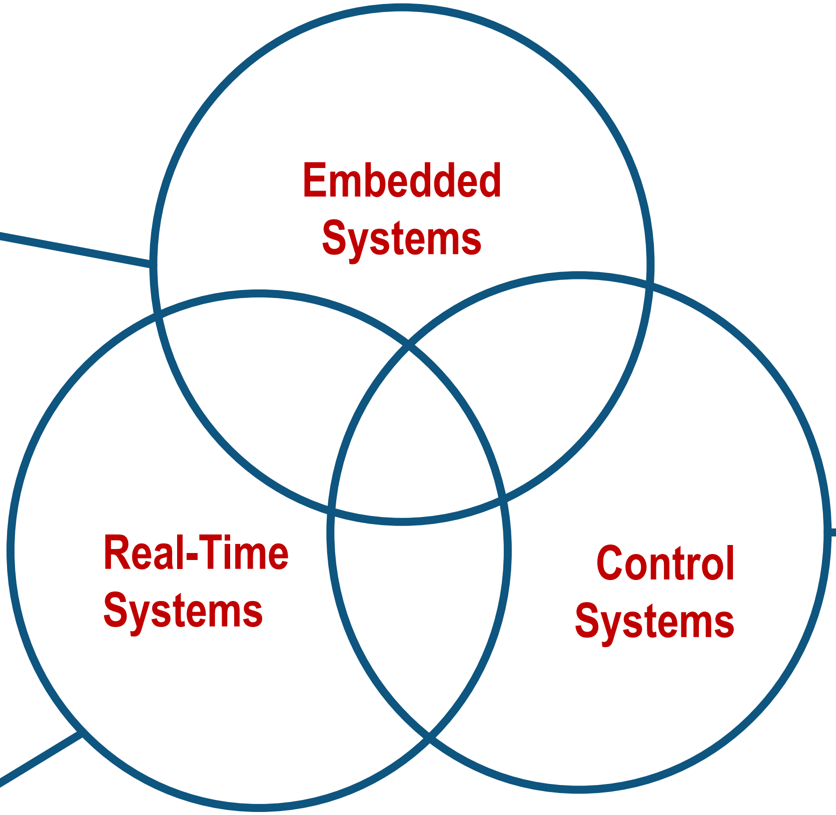
DEPARTMENT OF COMPUTER ENGINEERING

UPDATED - SPRING 2023

Chapter Overview

Introduction I

1. Not necessarily control systems, could only be sensor-based with human-machine interfaces.
2. Not necessarily real-time.
3. Vary from no to simple to complex control.
4. Flexible (Software-based)



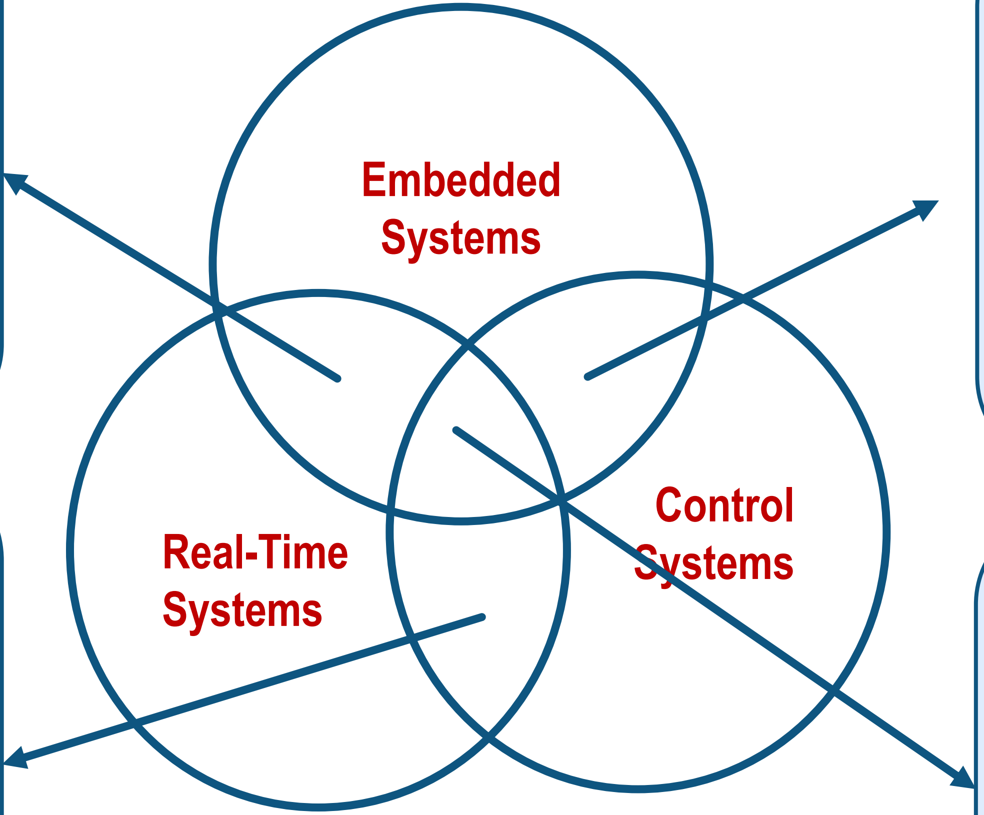
1. Not necessarily embedded. Could be running on PCs.
2. Not necessarily control but used in applications with timing constraints (*e.g.* video frame processing).
3. In some instances (*e.g.* soft real-time), subject to QoS.

1. Could be mechanical or electromechanical systems and no computers involved (Analogue control)
2. A computer or microcontroller / microprocessor or PLC can replace many components (Digital control)
3. Digital control not necessarily embedded or real-time
4. Control Systems have many advanced techniques and fields (see slide 8)

Introduction II

Real-Time Embedded Systems

- Systems with real-time requirements and constraints but not necessarily control applications
- *E.g.*, monitoring applications, some communication systems



Embedded Control Systems

- Control Systems that are hidden and embedded, but do not have real-time requirements.
- *E.g.*, many home appliances fall under this category like microwaves, washing machines, dryers, *etc*.

Real-Time Control Systems (RCS)

- Control Systems with real-time requirements and constraints that are not embedded or hidden
- *E.g.*, Air Traffic Control Systems, Process control in factories

Real-Time Embedded Control Systems

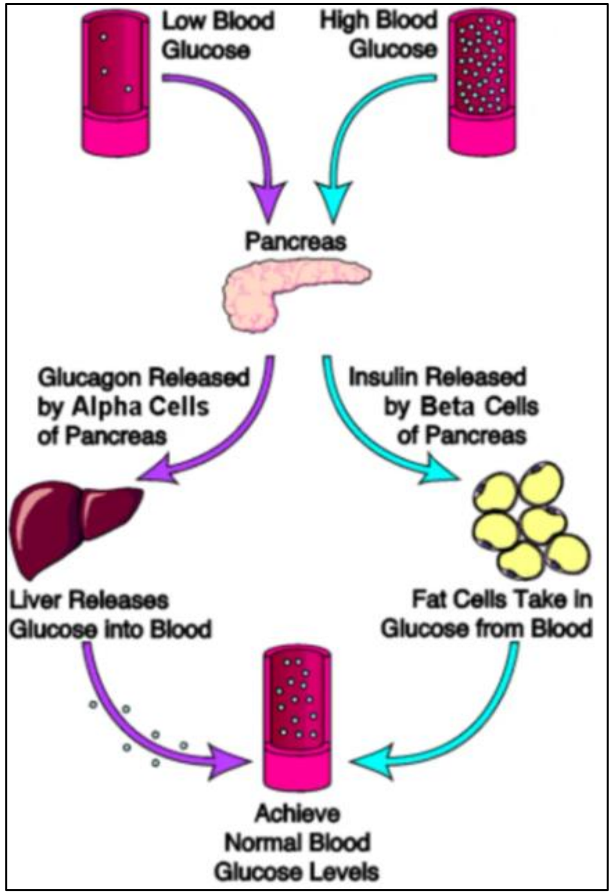
- Control Systems that are hidden and embedded, and have real-time requirements.
- *E.g.*, systems in automotive industry

Why do we need control system course?

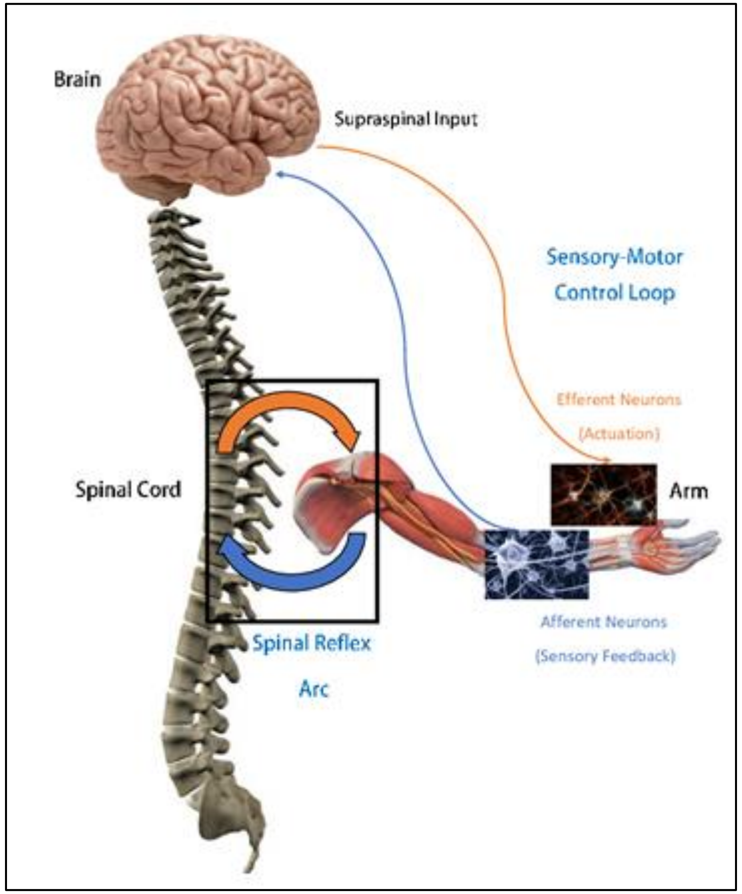
- In embedded systems course, you learned how to read from keypads, turn on LEDs and 7-segment displays, send and receive simple data through simple communication protocols (RS-232 or USART), run timers, and read from A/Ds. In all these cases, the controlled medium is quite simple (LEDs), and almost instantaneous in response.
- Real systems are far more complex; hydraulic, pneumatic, and mechanical systems are much slower to respond. They have limitations (*e.g.,* swing angles, maximum loads) as well as inertia (will not respond until certain forces are reached). These things need to be well-understood and modeled and how the physical system will respond based on physical models is quite important.
- If the controlled system is not well understood, well-modelled and well-controlled, it might not work as required or could go unstable, causing physical damage or even death.
- Also, in many applications, we need the system to operate smoothly and not abruptly (*i.e.,* smooth start/stop rather than sudden), so a simple on/off signal is no longer as simple. For example, you do not want a bumpy ride in an elevator.
- Also, real-life systems are subject to noises, disturbances, and operational situation that can change abruptly, we need to design control systems that will keep working despite these circumstances.

Control Systems in Nature

- Control Systems is not a human engineered discipline → **Best controllers are found in nature!**
- Biochemical Controllers → Human body has numerous control systems (central and distributed):
  * *pancreas,*
  * *immune system,*
  * *eyes tracking an object,*
  * *eye-hand coordination,*
  * *fight or flight response, etc.*
- Species declining or increasing population can be modeled as a self-regulating control system (wolves/rabbits' population)



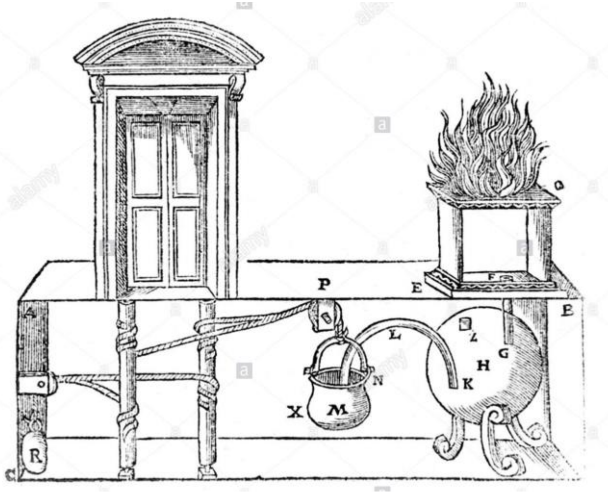
Pancreas Glucose Regulation



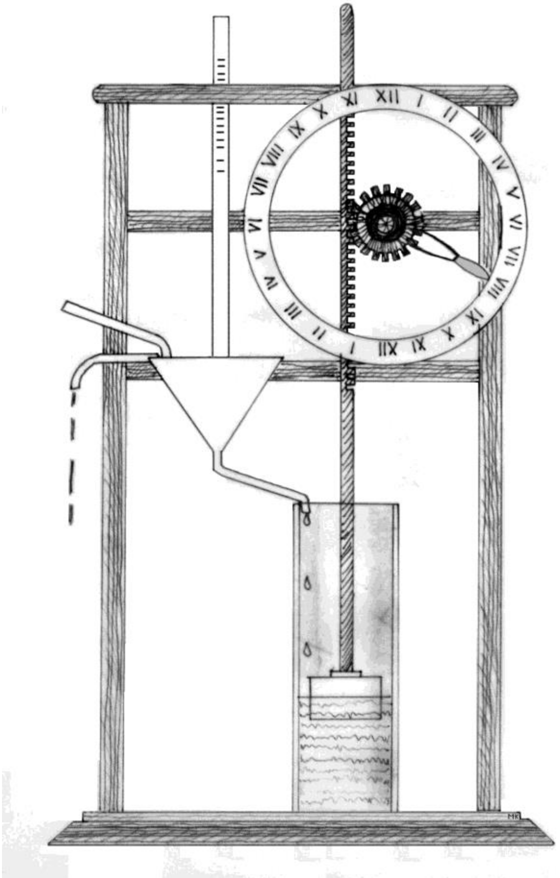
Sensory-Motor Control in a Human Arm

Control Systems History and Development I

- Control systems are ancient! Initially they were basically mechanical systems made from simple materials (*e.g.*, wood, metal, water and steam). Some were also automatic!
- First feedback control systems were developed in Greece in 3rd century B.C. (water clocks or clepsydra). The design was so successful it was still used in Baghdad until its fall in 1258 A.D. and in Europe up until the pendulum clocks were invented.
- Heron of Alexandria used steam pressure to control the automatic opening and closing of large temple doors in the 1st century A.D.
- The book "*Pneumatica*" by Heron described so many inventions and basic control systems, many of which are the basis of today's technology (*e.g.*, steam engines, windmills).
- ***Pneumatics*** (from Greek *πνεῦμα* pneuma 'wind, breath') is a technology that makes use of gas or pressurized/compressed gas whereas ***hydraulics*** (from Greek: Υδραυλική) makes use of liquids.



Ancient Pneumatic Temple Gate Control



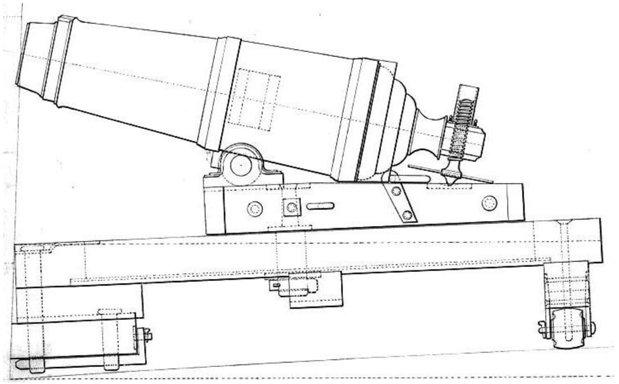
Clepsydra or water clock

Control Systems History and Development II

- Steam pressure and temperature control with safety valves as far back as 1681, windmill blade adjustment to wind speed in 1745.
- There was interest in building mechanical clocks, and automata figures that dance and move using steam and mechanical gears in the 17th and 18th century. These are prime examples of open-loop control systems (more on this later).

  If interested, the documentary video in the next slide presented by Professor Simon Schaffer is quite interesting!
- The foundation of control systems as we know them today can be traced to the year 1868.
- Mechanical and electrical systems modelling and description using differential equations, as well as understanding criteria for stability (*Maxwell, Routh-Hurwitz, Sturm, Chebyshev*).
- Mostly driven by wartime needs especially for warships and gun platforms → stability of steering ships and applying power through hydraulic systems, gun platforms for military ships (achieved finally in 1922 by Minorsky) (Adaptive Control field)
- In 1930's and 1940's: focus shifted to developing of math and techniques to facilitate analysis of control systems. Remember computers and calculator were not invented ye. (Bode, Nyquist) → Bode Plots and Root Locus

17th/18th Century Doll Automata



Naval Artillery Platform

Teaching Machines How To Act Like Humans And Animals

https://youtu.be/YAg66jrvpHA

Control Systems History and Development III

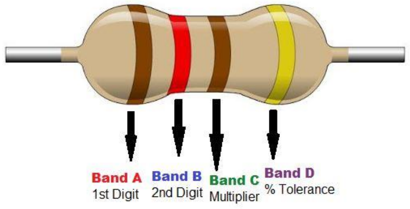
- Analogue control techniques mostly deals with electromechanical, hydraulic and pneumatic controllers, whereas digital control theory simply replaces the analogue components with a computer, microcontroller, microprocessor or PLC.
- Control theory is divided into two main branches:
  - **Linear Control Theory** (the focus introductory B.Sc. courses): applies to systems made of devices which obey the superposition principle, which means that the output is proportional to the input. They are governed by linear differential equations. It also involves systems whose parameters do not change with time, that is linear time invariant (LTI) systems (the ones you studied in EE signals course).
  - **Non-Linear Control Theory**: covers the class of systems that do not obey the superposition principle and applies to more real-world systems because all real control systems are nonlinear. Requires advanced numerical methods to analyze such complex systems.

Control Systems History and Development IV

- Some of the main control techniques/fields (usually topics taken at M.Sc. and Ph.D. level):
  - **Optimal Control:** a control technique in which the control signal optimizes a certain "cost index": for example, in the case of a satellite, the jet thrusts needed to bring it to desired trajectory that consume the least amount of fuel. Mainly applies two methods: Model Predictive Control and (MPC) and linear-quadratic-Gaussian control (LQG).
  - **Stochastic Control:** deals with control design with uncertainty in the model. It is assumed that there exist **random** noise and disturbances in the model and the controller, and the control design *must take into account these random deviations.*
  - **Adaptive control** (1950s): uses on-line identification of the process parameters, or modification of controller gains on the spot. Used widely in aerospace industry.
  - **Robust Control** (1960s, 1970s): Controllers designed using robust control methods tend to be able to cope with small differences between the true system and the nominal model used for design
  - **Intelligent Control** (2010s, 2020s): uses AI computing approaches, machine learning, evolutionary computations, genetic algorithms and reinforcement learning to control ***complex dynamic systems***.

Analogue Control Systems and their Disadvantages

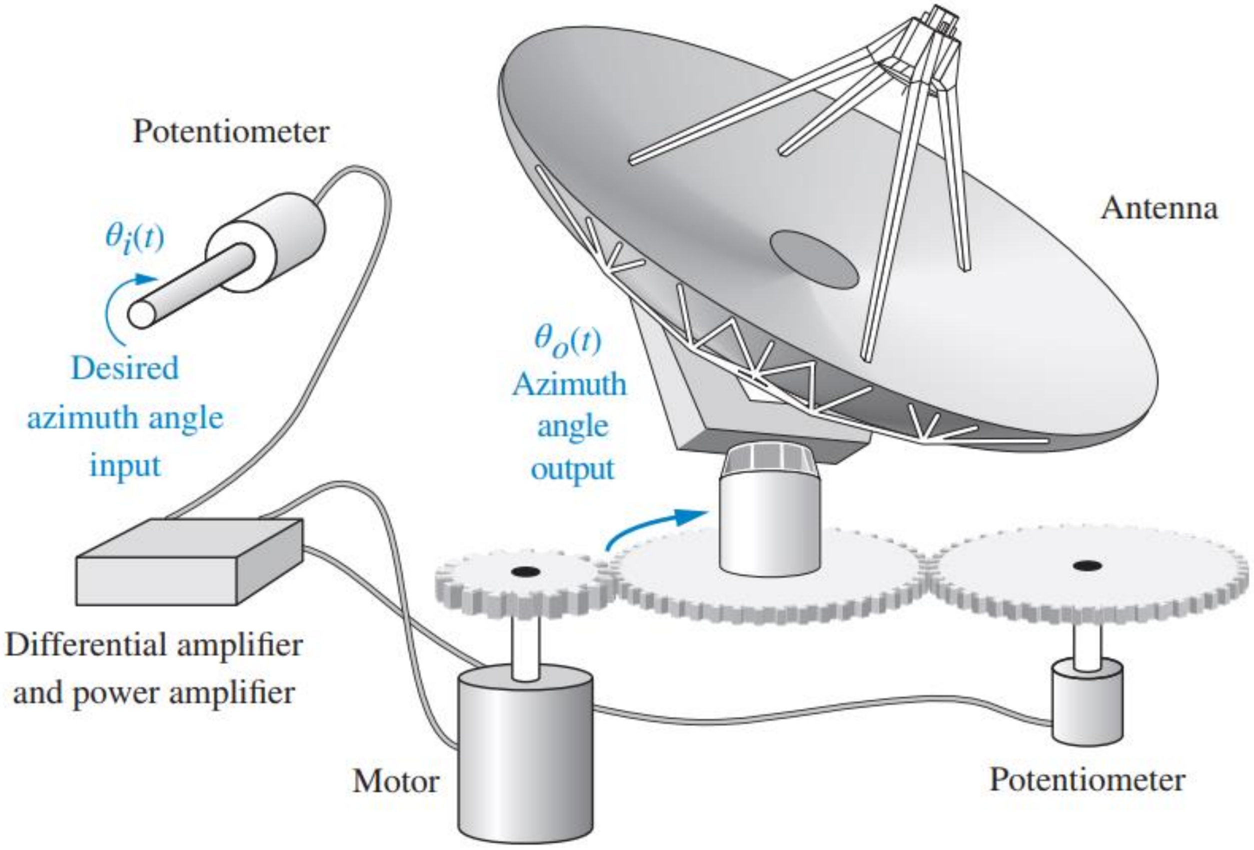
- The basis of every control course is to study classical control theory. All main ideas in control systems start from here before moving to digital (computer-controlled) systems, and more advanced topics in control theory (see slide 6).
- In analogue control, the controller is made of electromechanical parts (resistors, capacitors, inductors, *etc.*). All signals are continuous time domain (no sampling). The entire signal (voltage, current is processed)
- Analogue control systems have several disadvantages:



  - Could be quite complex to design, especially for complex control systems.
  - Difficult to modify or update → Complete circuit and system redesign.
  - Since they use analogue components (*e.g.*, resistors and capacitors), their values vary from their nominal values due to tolerances (e.g., 1000 ± 100Ω) , and resistance changes for example by circuit/environment temperature, and components quality and values change over time due to decay.
  - Highly susceptible (sensitive) to noise and power supply drift (DC shifting)
  - Each control subsystem needs its own circuit, no sharing of components is possible, thus quite costly.
  - Could be hard to achieve some mathematical computations in analogue hardware.

Simple Control System Example (All Analogue Components)

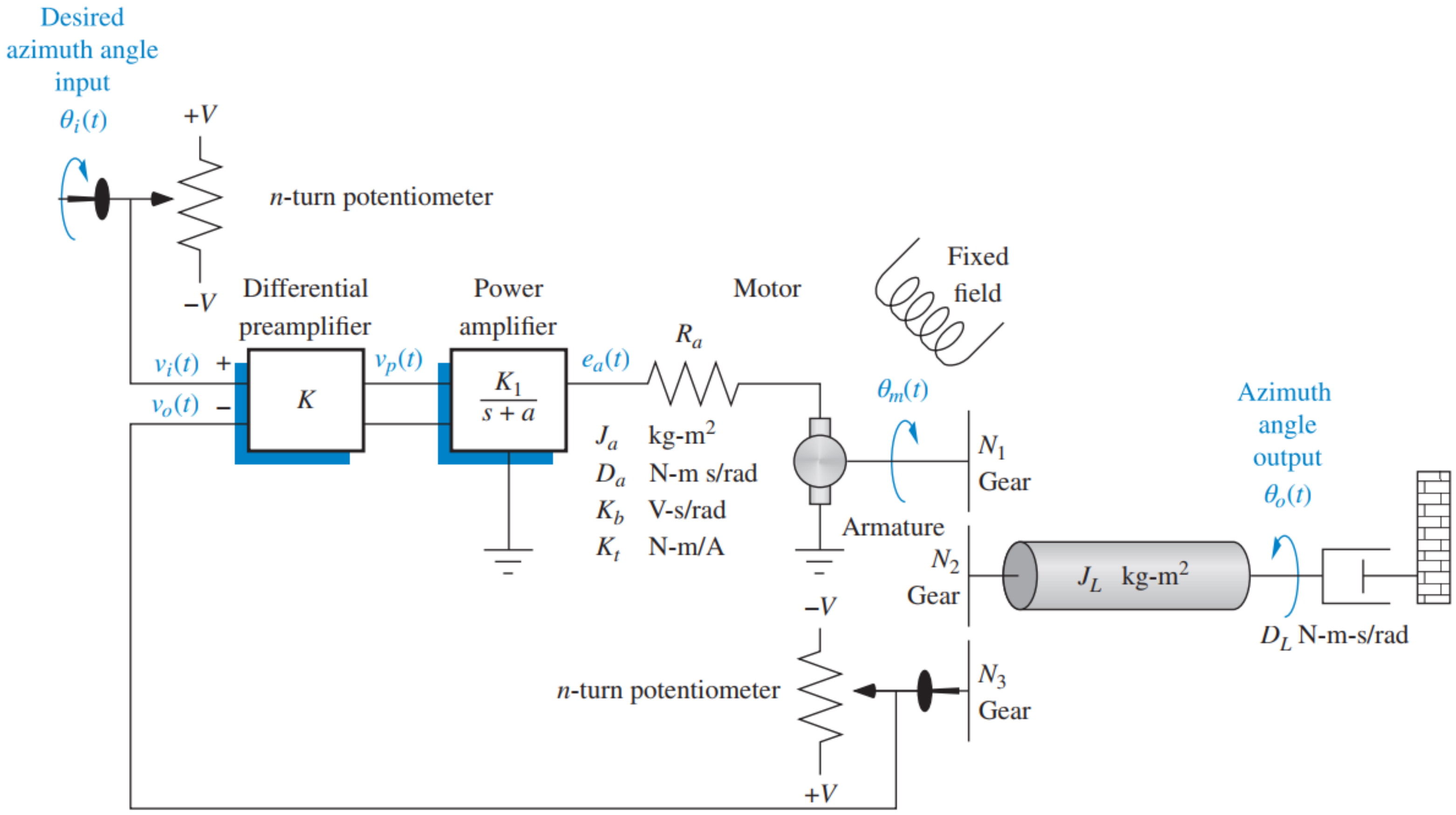
Layout



Antenna Azimuth Position Control System

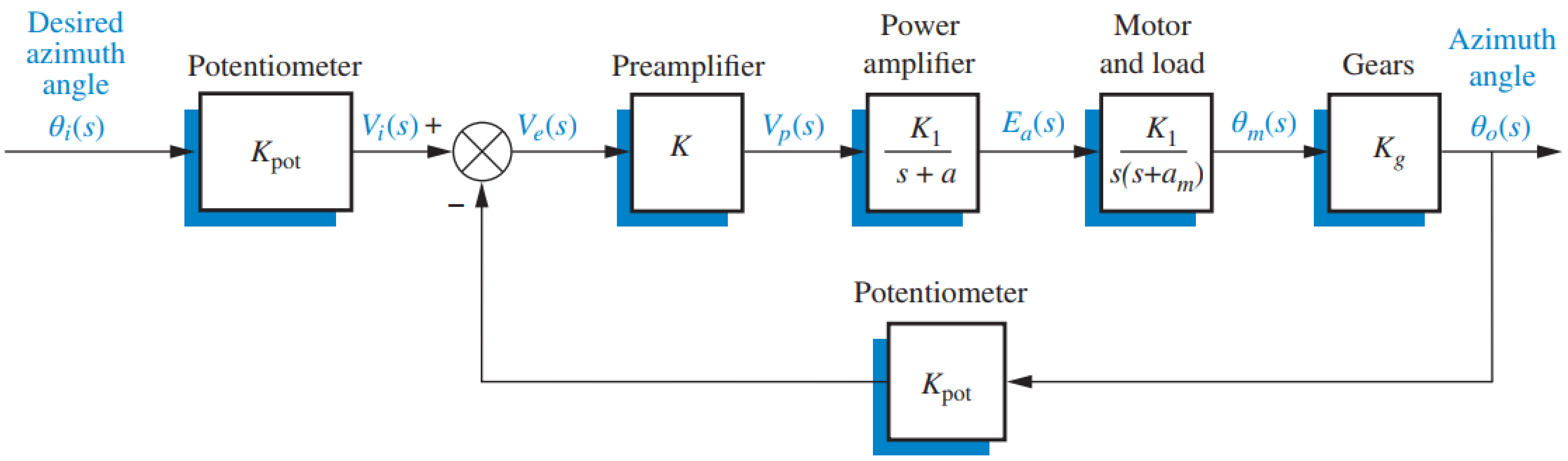
Simple Control System Example (All Analogue Components)

Schematic



Simple Control System Example (All Analogue Components)

Block Diagram



You can notice that there are **no computers** or **digital controllers** in the layout / schematic / block diagrams → analogue control

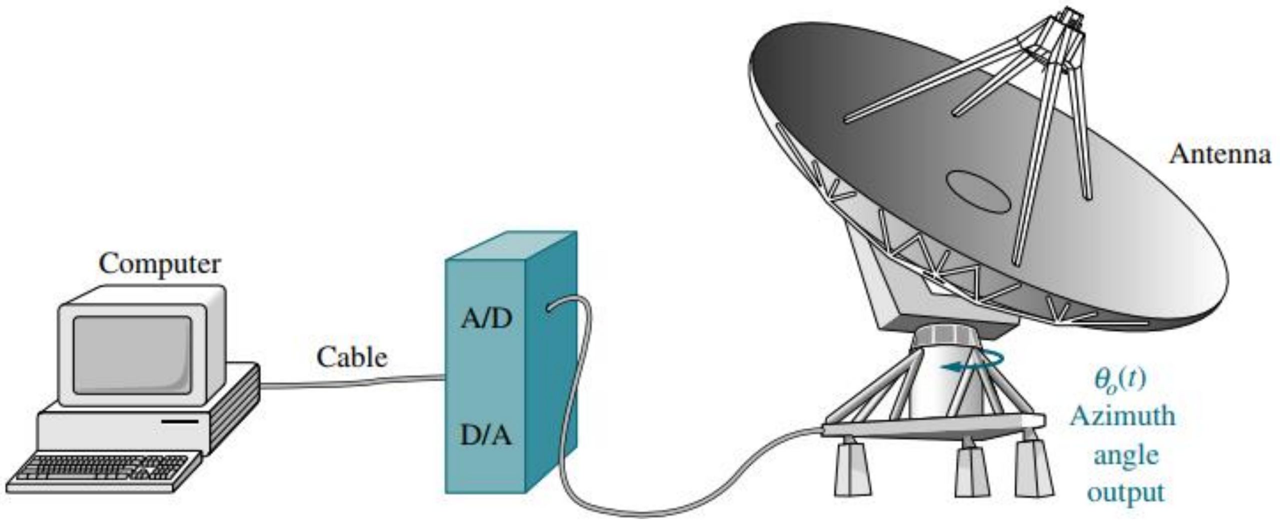
Digital Computer Role in Control Systems

- **Supervisor only role:** a digital computer might not take a direct control role at all. The whole plant might by controlled through conventional analogue controllers, however, computers take a **supervisor role.** That is, they simply collect data about the system operation, creating logs, or showing real-time sensory data input to remote screens where engineers can monitor the plant and see if there are hazards or safety actions that need to be taken.
- **Digital controller:** A digital computer can replace many analogue controllers, where it takes inputs, feedback data in digital form, process them through software code, equations, libraries, software PID, and issue commands to the plant for **direct control**. Programmer logic controllers (PLC) are common industrial examples, SoCs for other applications.
- **Hybrid role:** In Industrial automation and Industrial IoT **(IIoT**), the computer role can be **hybrid**; that is, it can take both a supervisory role and direct control role. These systems are quite famous and are referred to as **SCADA** (Supervisory Control and Data Acquisition).
    - The main objectives of a SCADA system are,
        1. To operate and interact with ground-level devices such as different types of sensors and actuators.
        2. To operate and control single or multiple machines / operations at local or remote locations through HMI in a central room.
        3. To gather and log data, schedule events, issue alarms

Digital Control (Computer-Controlled Systems)

Most modern control systems are digital → They use main computer frames, PLCs or embedded controllers.

- Could be simple: home automation, DVD player, HDD controller, washing machine
- Could be complex with many subsystems, or distributed control systems:
  - Industrial robots,
  - Chemical / nuclear process control,
  - Spacecraft / space rover:
    - orbital maneuvering system (OMS),
    - thrust controller for orientation,
    - reaction control systems (RCS),
    - life support systems
    - flight control systems to adjust for atmosphere disturbances,



Digital Control Systems and their Advantages

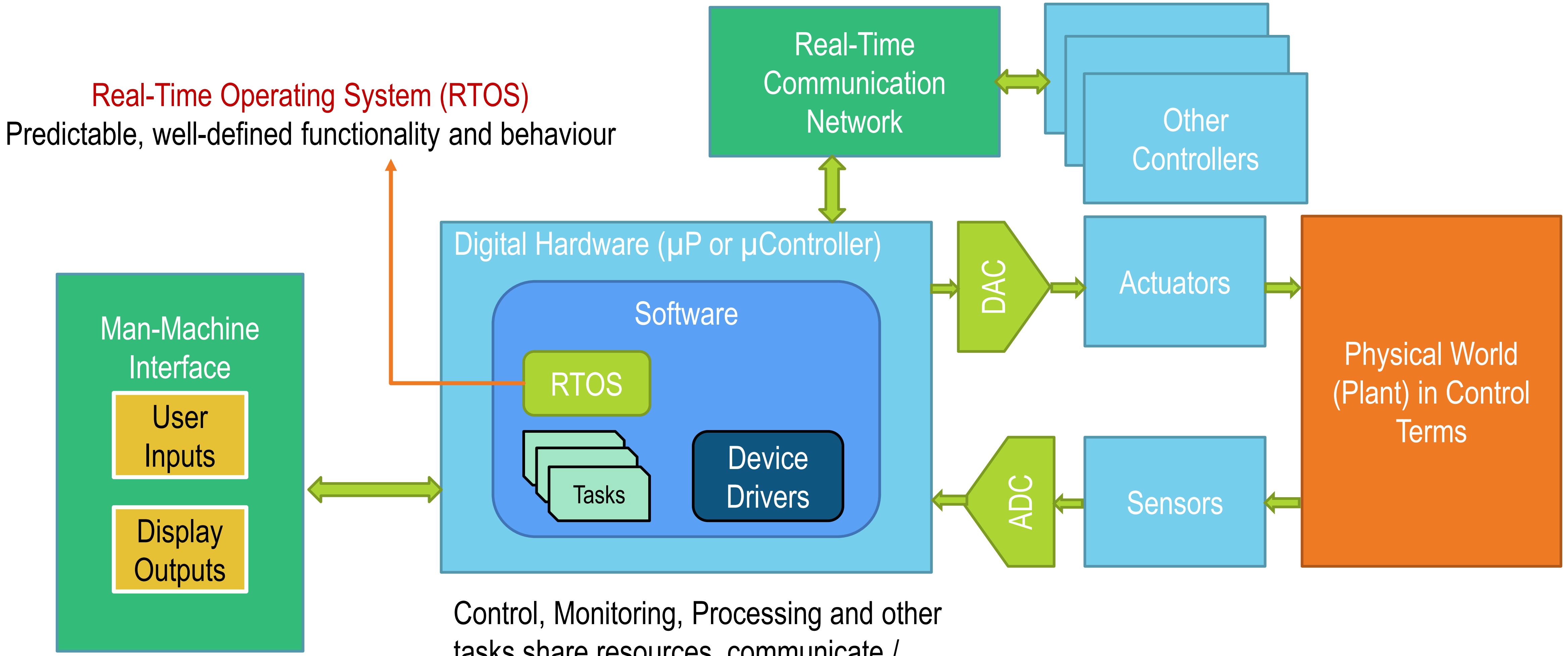
- Many hardware circuits can be replaced by software code (*e.g.*, some filters can be coded instead of being built as analogue circuits)
- Complex arithmetic operations which cannot be easily realized in hardware can be written in software or called from libraries → allowing us to design systems quickly (check ARM CMSIS DSP / Control) https://www.keil.com/pack/doc/CMSIS/DSP/html/modules.html
- Digital controllers can be easily modified without complete replacement of the original controller → **no circuit redesign or re-wiring, simply rewriting software, compiling and updating.**
- Since the control is now in software, the computer can run multiple control codes for multiple systems / subsystems at the same time.
- If appropriate digital controller is chosen, then real-time control is possible.
- Advances in VLSI technology provide better, faster and more reliable integrated circuits at lower prices; that is, a computer controller is **more reliable** that a controller built using analogue parts (slower decay, more accurate, less sensitive to noise and component decay and tolerances).

Digital Control Systems and Notes on A/D (ADC)

However:

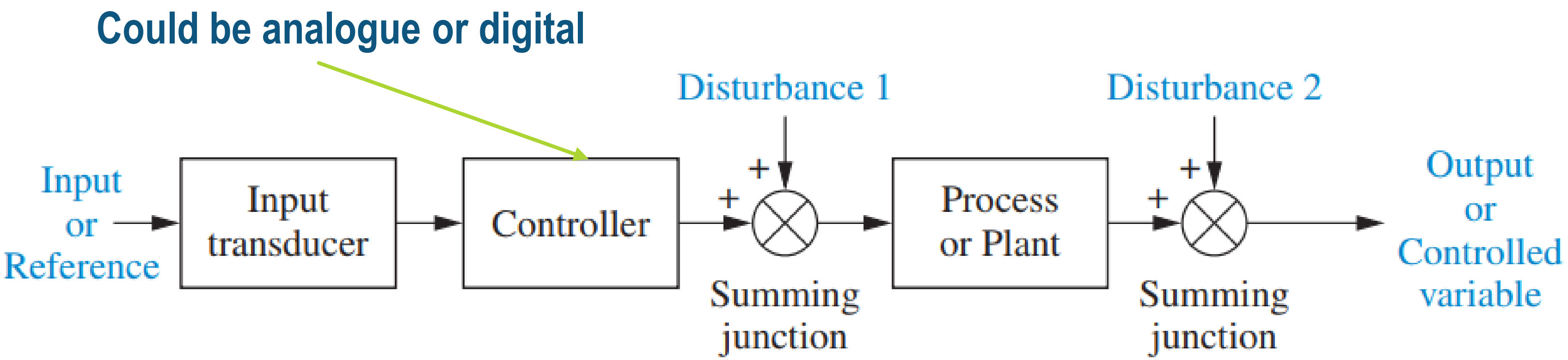
- Discrete time signals and approximation are always involved **(Sampling Theory : Shannon's Theory of Sampling / Nyquist's Rate).** We no longer have continuous time voltage or current signals for example.
- Quantization error due to A/D is always there. For more accurate representation of signals using "0" and "1" , one needs higher ADC resolution (12 bits, 14, 16 bits or more) → Negligible errors within required specification.
- Yet, more resolution means slower time to generate the number. (Sampling rate F_s ↓, Sampling Period T_s ↑)
- These days, faster ADC hardware allows short sampling periods (high sampling rate) but could cost more.
- With short sampling periods, digital controllers monitor controlled variables almost continuously

Overview of a Simple Digital Real-Time Control System



Control, Monitoring, Processing and other tasks share resources, communicate / synchronize with each other

Open-Loop Control Systems



Main characteristic of open-loop systems is that they:

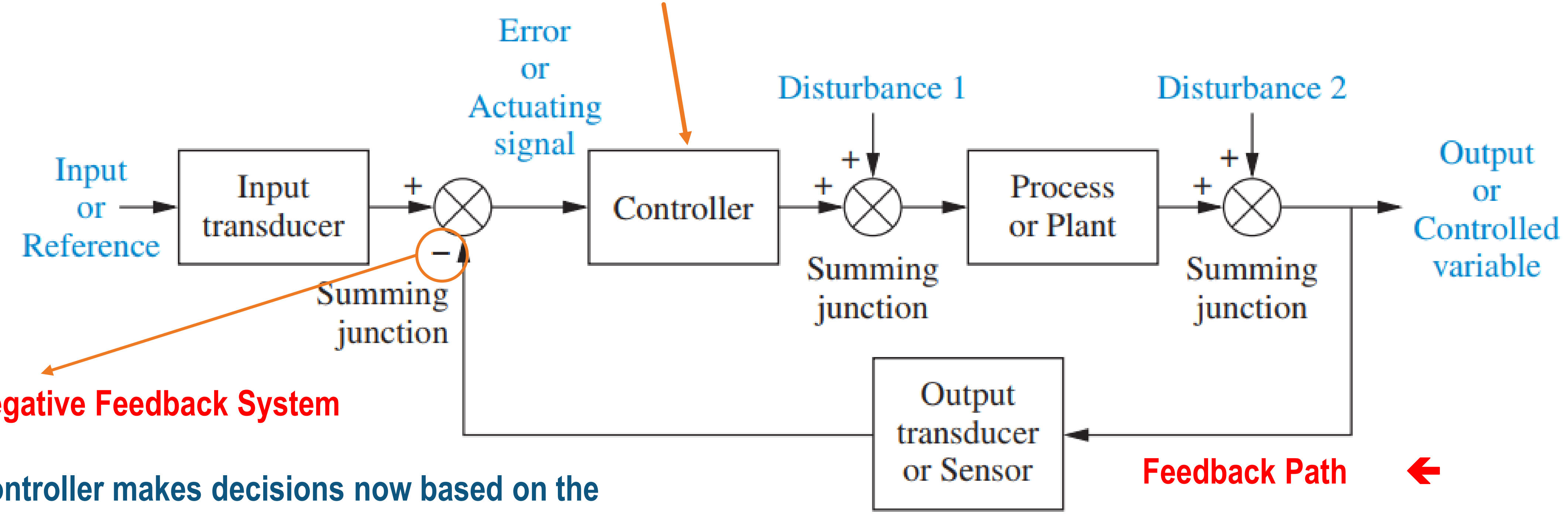
- Contain one or more **feed-forward paths.** That is, the flow of signals / commands is one-way.
- Are only commanded by the input(s) and do not check for the controlled variable value (no return or feedback data).
- Are therefore sensitive to disturbance because they cannot compensate / correct for any disturbances (noise):
  - either the one that add to the controller's driving signal (Disturbance 1, noisy control signal going into the actuator),
  - or disturbances at the output (Disturbance 2, for example a physical object in the way, they simply would not know about it)
- A kitchen microwave or toaster is a simple example. There is no information if the food reached the desired temperature, or if the bread got burned in the toaster or not!

- Open loop control systems are very simple to design, stable for simple but could be unstable for complex systems

Negative Feedback Closed-Loop Control Systems I

Could be analogue or digital. Also called compensator

Feed-forward Path ➔



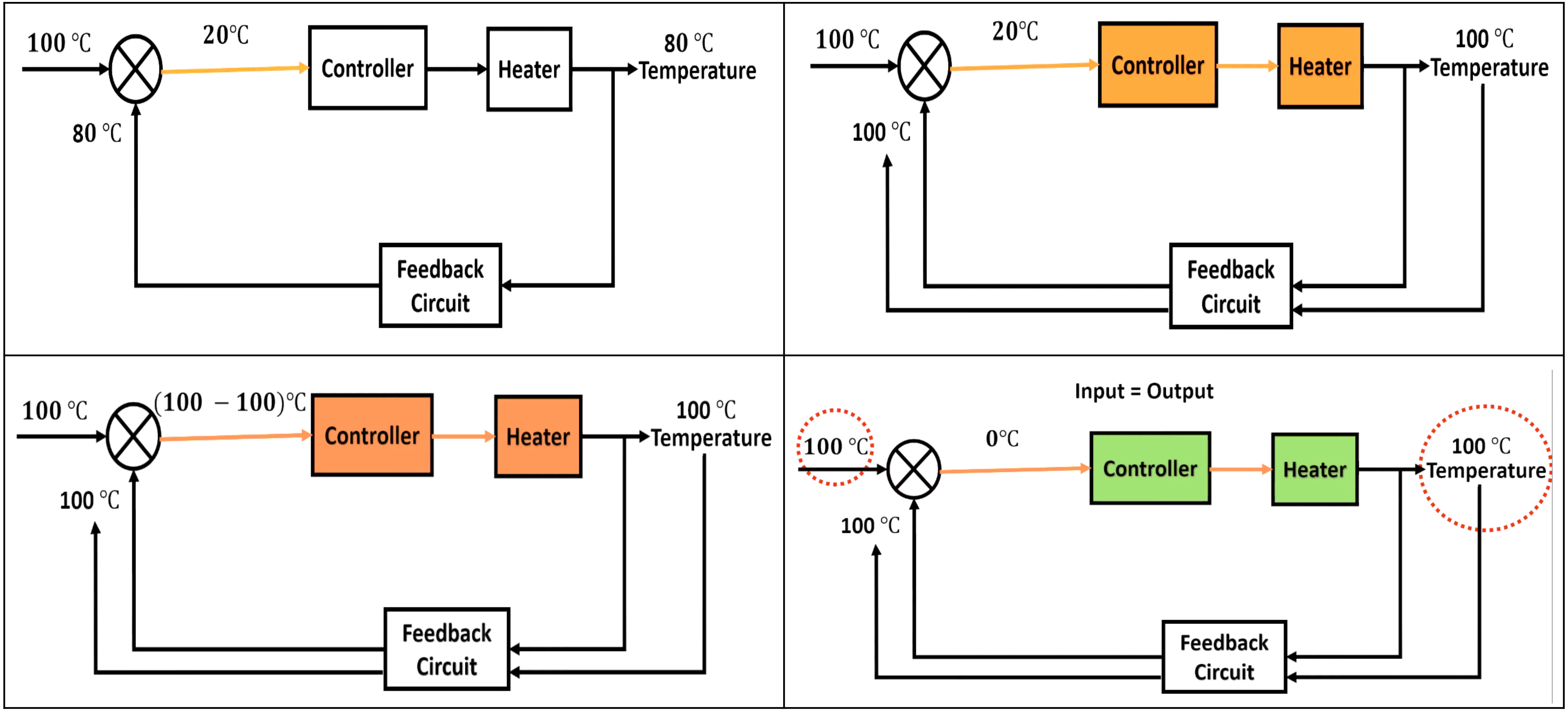
Negative Feedback System

Feedback Path 🡐

Controller makes decisions now based on the difference of the two signals (if both are unity gain)

This type of systems is also called Degenerative Feedback

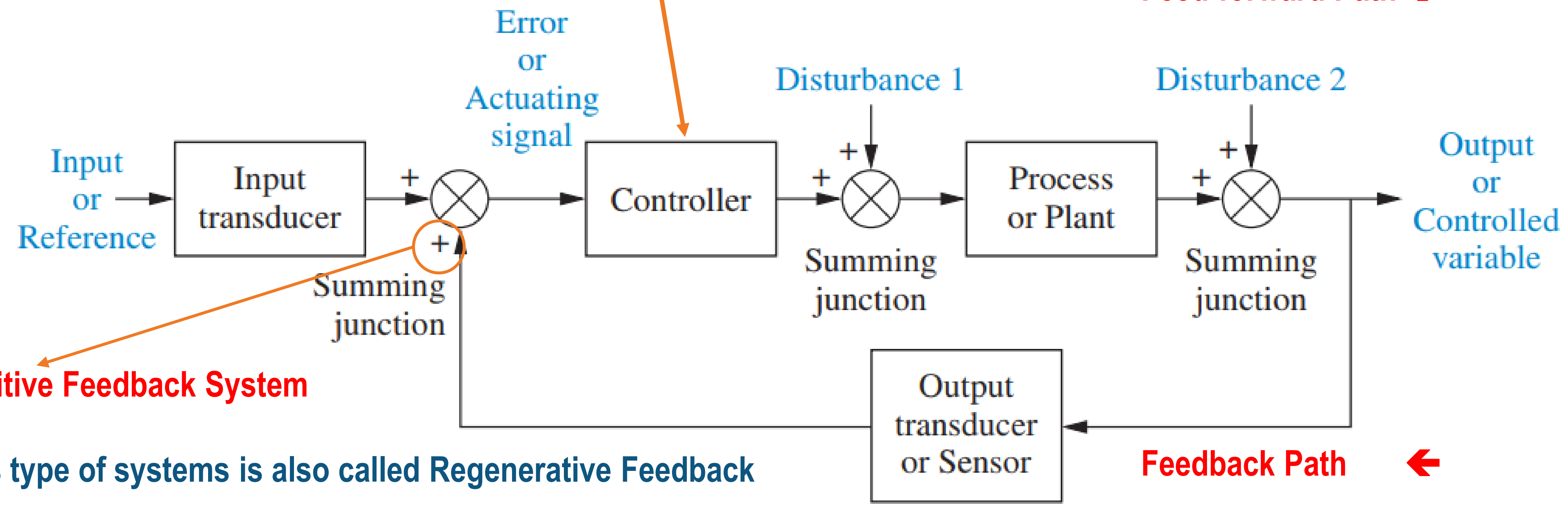
Negative Feedback Closed-Loop Control Systems II



Positive Feedback Closed-Loop Control Systems

Could be analogue or digital. Also called compensator

Feed-forward Path ➔



Positive Feedback System

This type of systems is also called Regenerative Feedback

Feedback Path ←

Positive Feedback could be desired

1. A simple NAND latch that forces that output to 0s and 1s is a positive feedback system with desirable output.
2. Used in building oscillators

Positive Feedback could be undesired

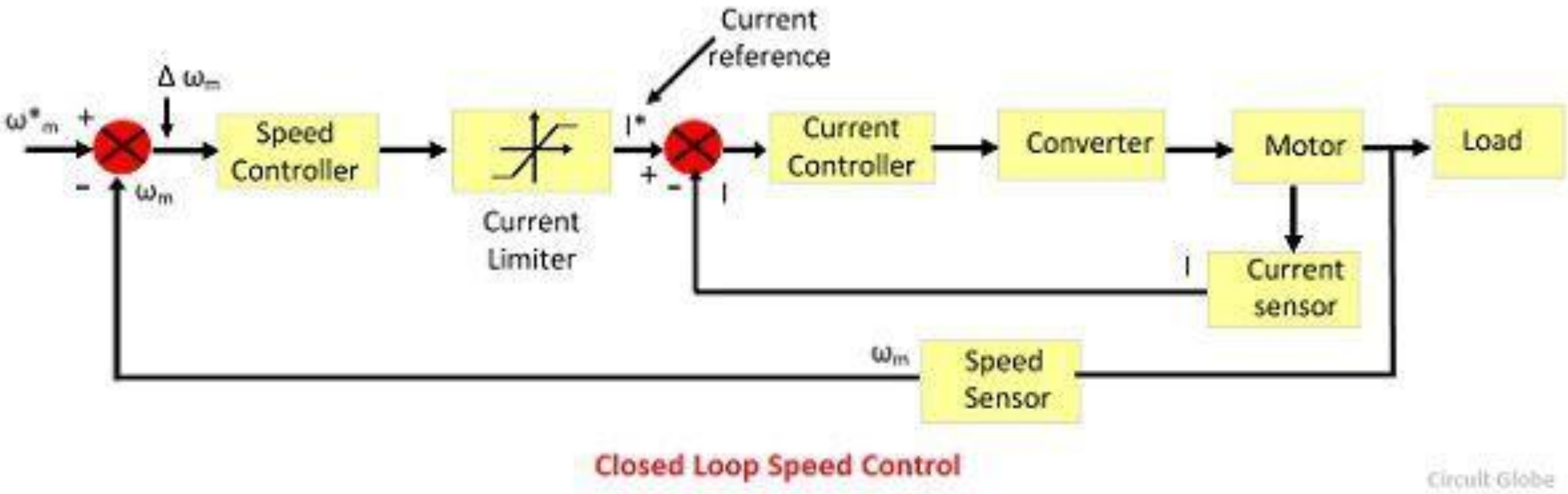
1. However, a microphone and speaker loop in a conference hall is a positive feedback system with undesirable or unstable output.
2. Also, one panicked person running in a crowd, makes other panic, who make in turn make others panic and so on: https://www.youtube.com/watch?v=oRQyu66zGE4

Multiple Closed-Loops Control Systems I

There could be multiple closed loops feedback to one or more controllers:

The feedback loops could be:

- Multiple sensors connected to the same physical object in the plant. For example, multiple sensors connected to one engine motor yet measuring different things such as engine temperature, engine vibration, engine current, engine speed, or engine fuel flow.



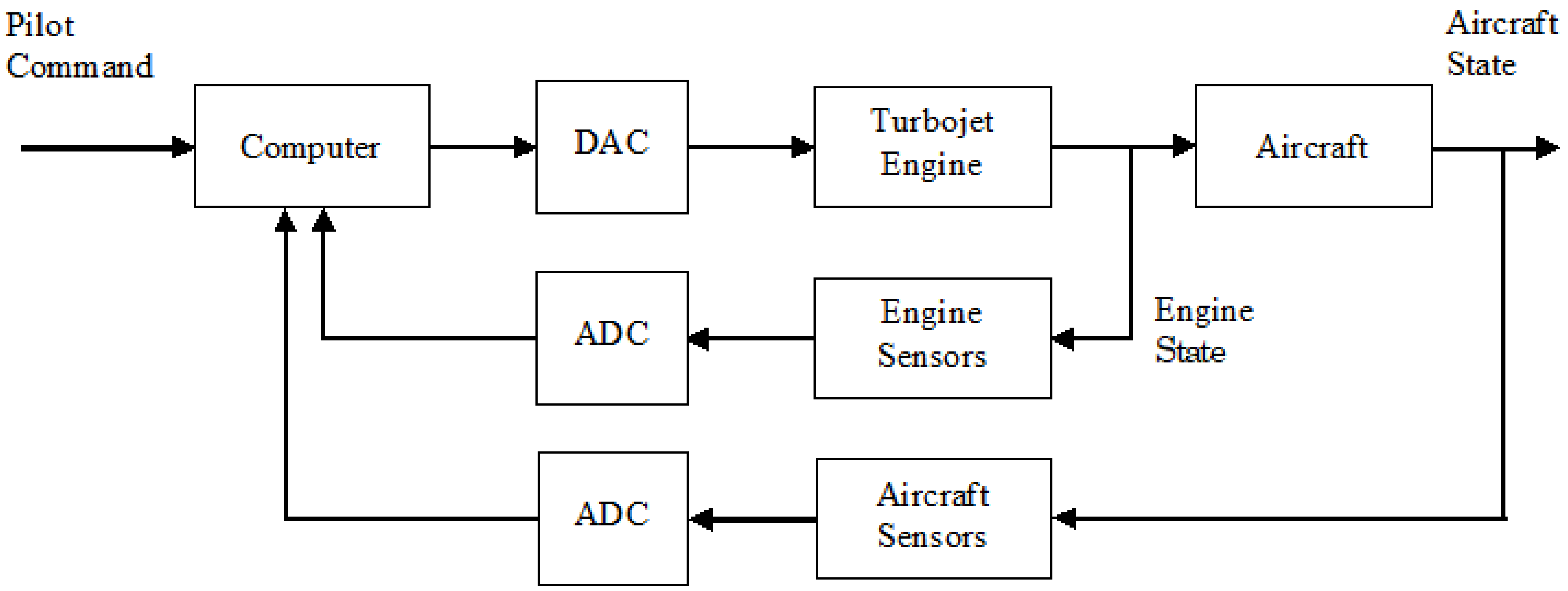
Closed Loop Speed Control

Multiple Closed-Loops Control Systems II

There could be multiple closed loops feedback to one or more controllers:

The feedback loops could be:

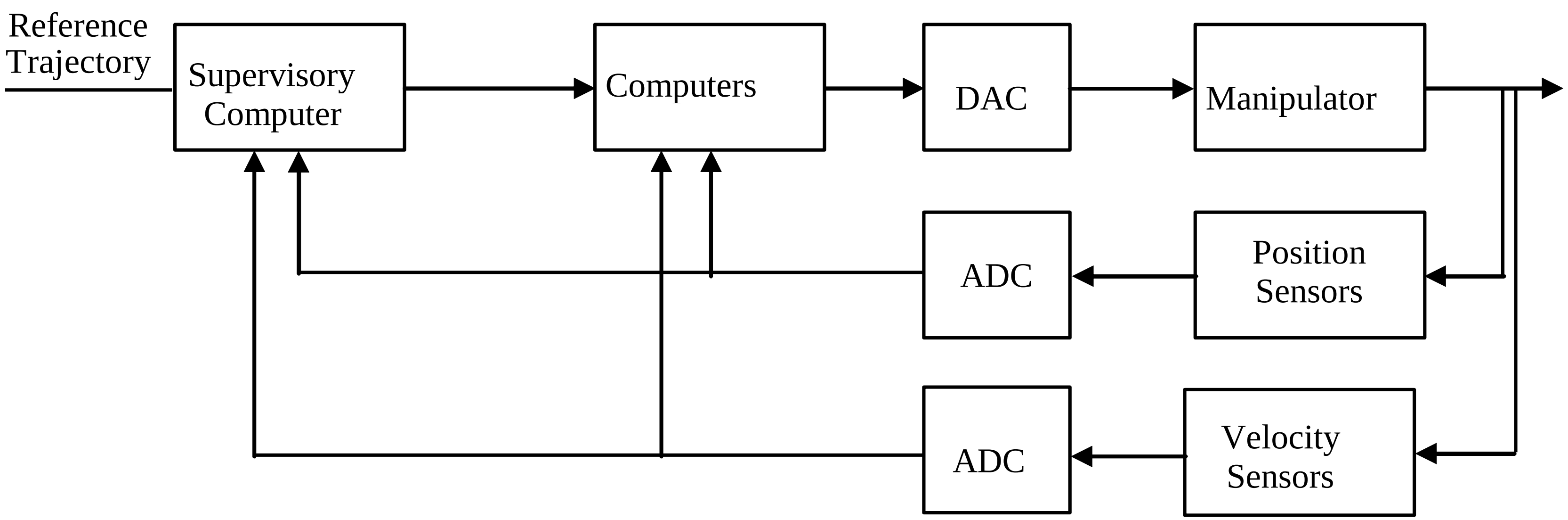
- Multiple sensors connected to different physical objects in the plant.



Multiple Closed-Loops Control Systems III

There could be multiple closed loops feedback to one or more controllers:

- In addition to a controller, there could be feedback loops to a supervisory computer for logging data and monitoring the plants.



Open-Loop vs Closed-Loop Control Systems

- Closed loop systems have the obvious advantage of greater accuracy than open-loop systems, as well as being more reliable.
- Closed-loop systems are more complex and expensive than open-loop systems (extra sensors, extra wiring, ADC if digital, more processing time and needs more maintenance)
- Closed-loop systems are less sensitive to noise, disturbances, and changes in the environment
- Transient response and steady-state error can be controlled more conveniently and with greater flexibility in closed-loop systems (more in following slides / chapters)
- Closed Loop systems could become unstable therefore is a major concern during design stage!

Question: What do you need to do to redesign the simple toaster as a closed-loop feedback system? What will you be sensing to get a perfect toasted piece of bread?

Home Exercise: Think of another existing open-loop system at your home, and think of what you need to convert it to a closed-loop system? Is it worth it?

Temporal Characteristics of the Plant

Input: Press the 4th floor button representing our desired output (reach 4th floor)

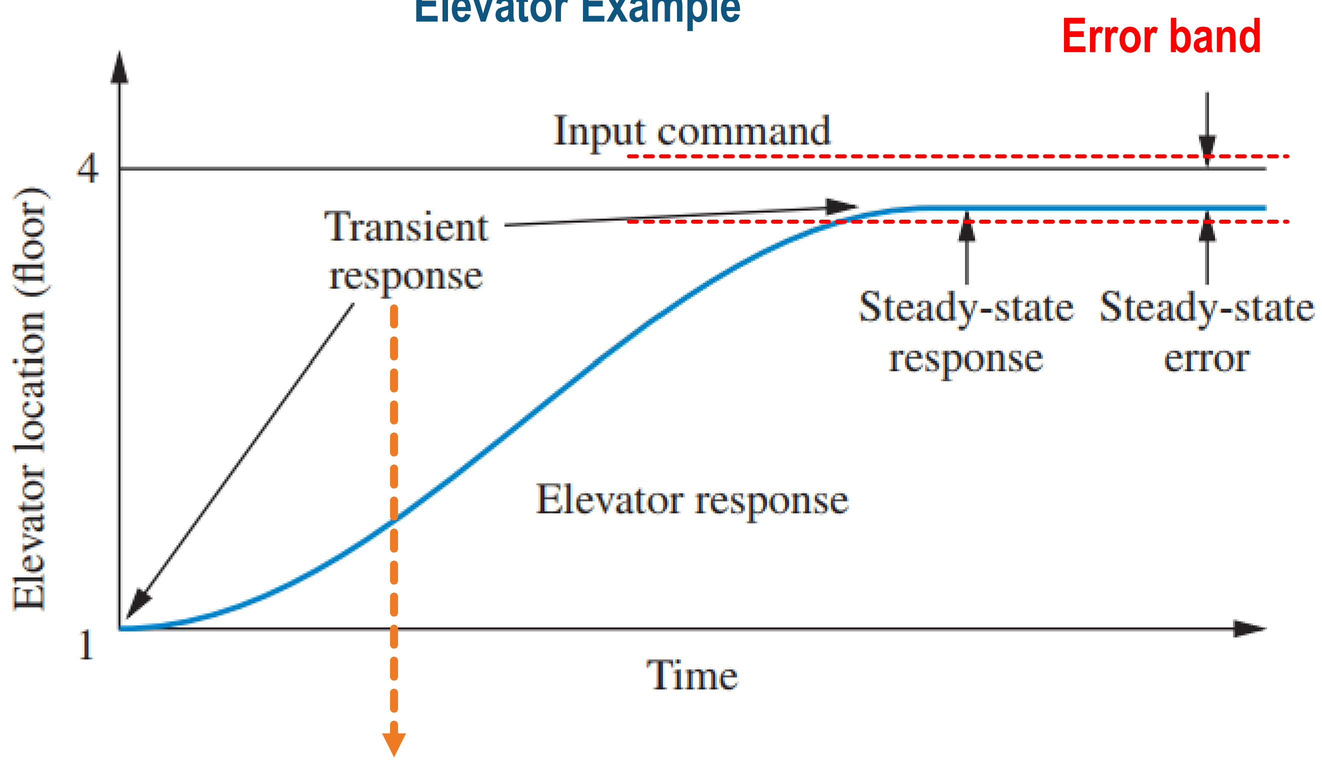
A button ideally represents a unit step function u(t)

Two measures of performance are apparent:
(1) the transient response and
(2) the steady-state error.

Analysis Stage: find values for transient response and steady state error that are within specifications and acceptable

Design Stage: apply control system parameters that satisfy the values analyzed

Elevator Example



Too slow: user patience is sacrificed
Too fast: user comfort is sacrificed, could cause damage/death

No oscillation
Levelling Correctly

Analysis and Design of Control Systems II

Temporal Characteristics of the Plant

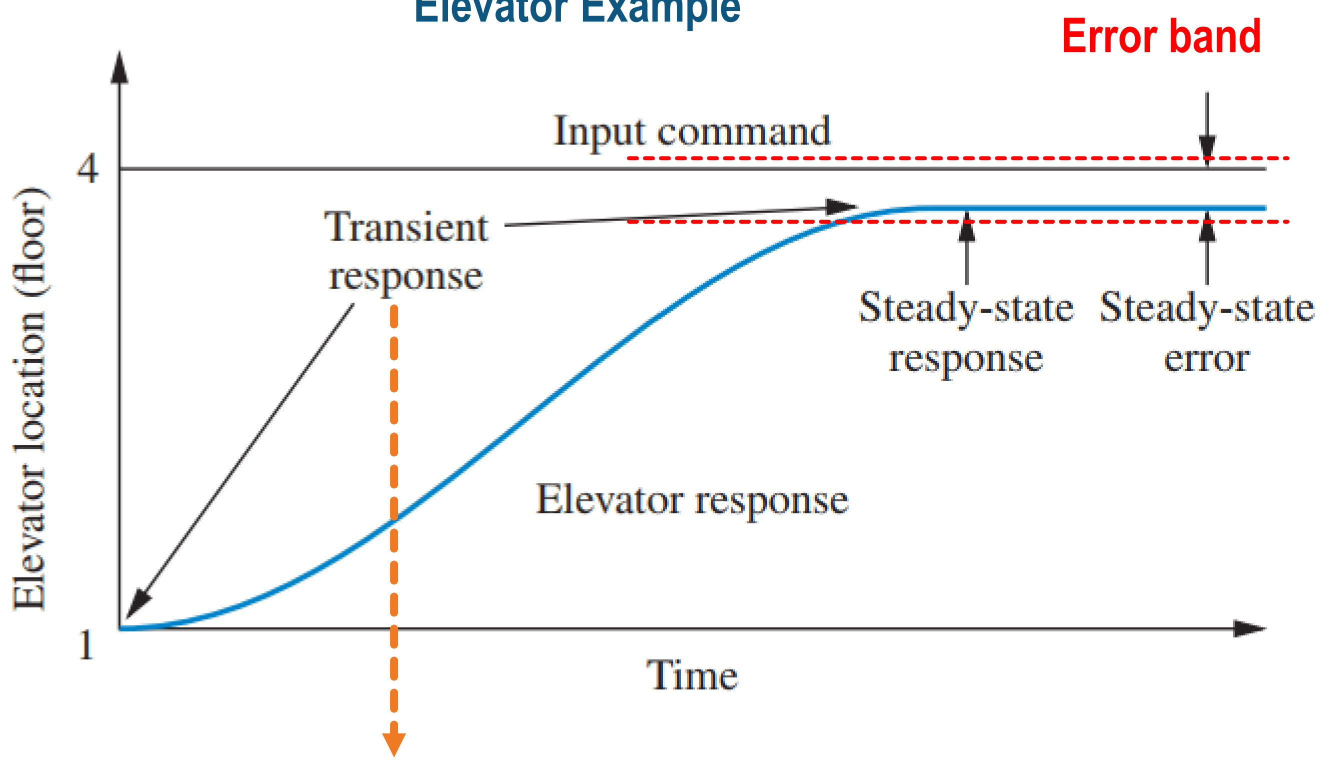
The transient response should be as quick as possible (ideal case), but in many systems, this is not feasible either due to its catastrophic effects or simply it is physically impossible:

e.g., an elevator going from G-level to 4th or 100th floor instantly will kill you (catastrophic effects)
Also, mechanical systems are slow and physically infeasible to have instantaneous response (physically impossible).

The ideal steady-state means we should reach our desired output without error. However; error could be there.
An error band specifies the maximum acceptable margins for errors (0.01%, 1%, 5%) and depends on the application.

Elevator could stop at the 4th floor with its level 1cm above the floor ground (acceptable), 5cm → You could trip and fall, 50 cm (something is visibly wrong, and it is dangerous)

Elevator Example



Too slow: user patience is sacrificed
Too fast: user comfort is sacrificed, could cause damage/death

No oscillation
Levelling Correctly

Analysis and Design of Control Systems III

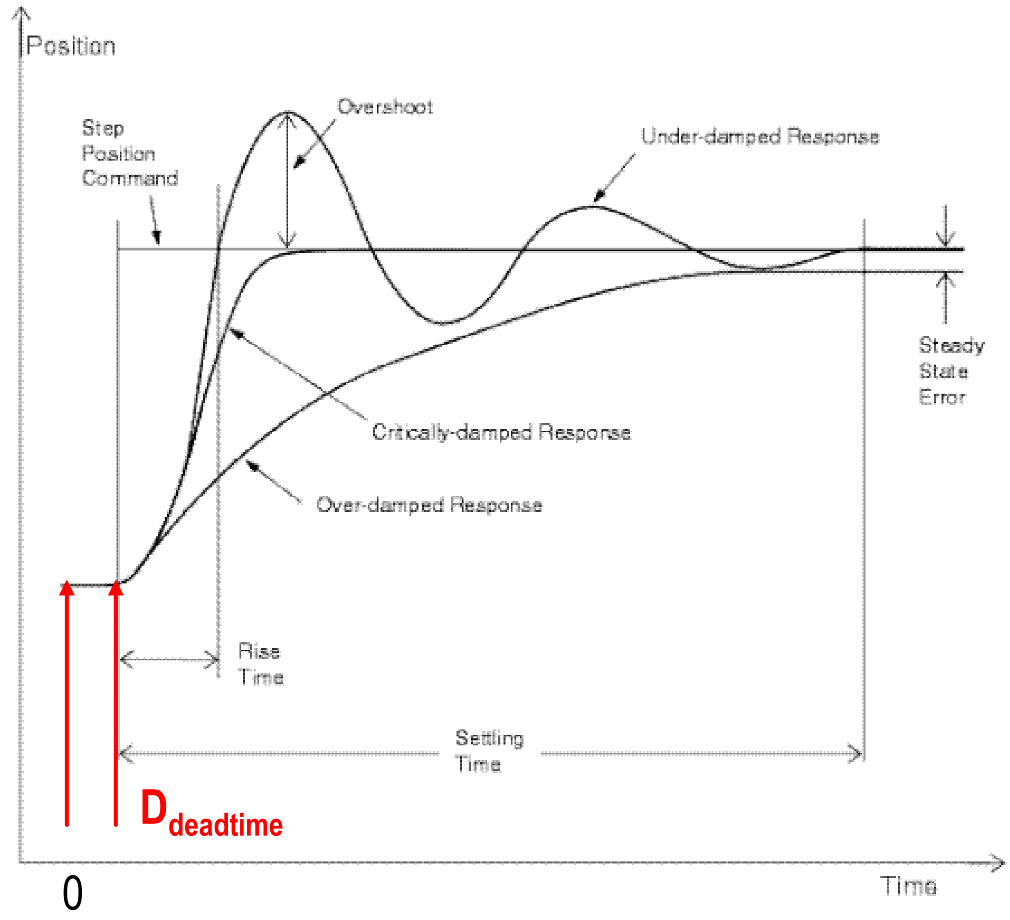
When responding to controllers' commands, changing the old state and reaching the new stead-state is not necessarily a smooth ride. It depends on the plant components, physical characteristics and equations

Given the system characteristics, we could have different type of responses:

1. **Overdamped** response
2. **Critically damped** response
3. **Underdamped** response → Overshoots

Overshooting means that the output oscillates until it settles on the stead-state value.

Imagine using an elevator to go to the 4th floor, but it first goes to the 5th floor, then 3rd floor, then goes to between 4th and 5th, then to between 3rd and 4th before eventually settling on the 4th floor.



Oscillations could be not only inconvenient but also dangerous, if they exceed the plant physical limitations.

Imagine if the 4th floor is the last floor, if they elevator overshoots, then means it will hit the roof causing damage, injury, or even death.

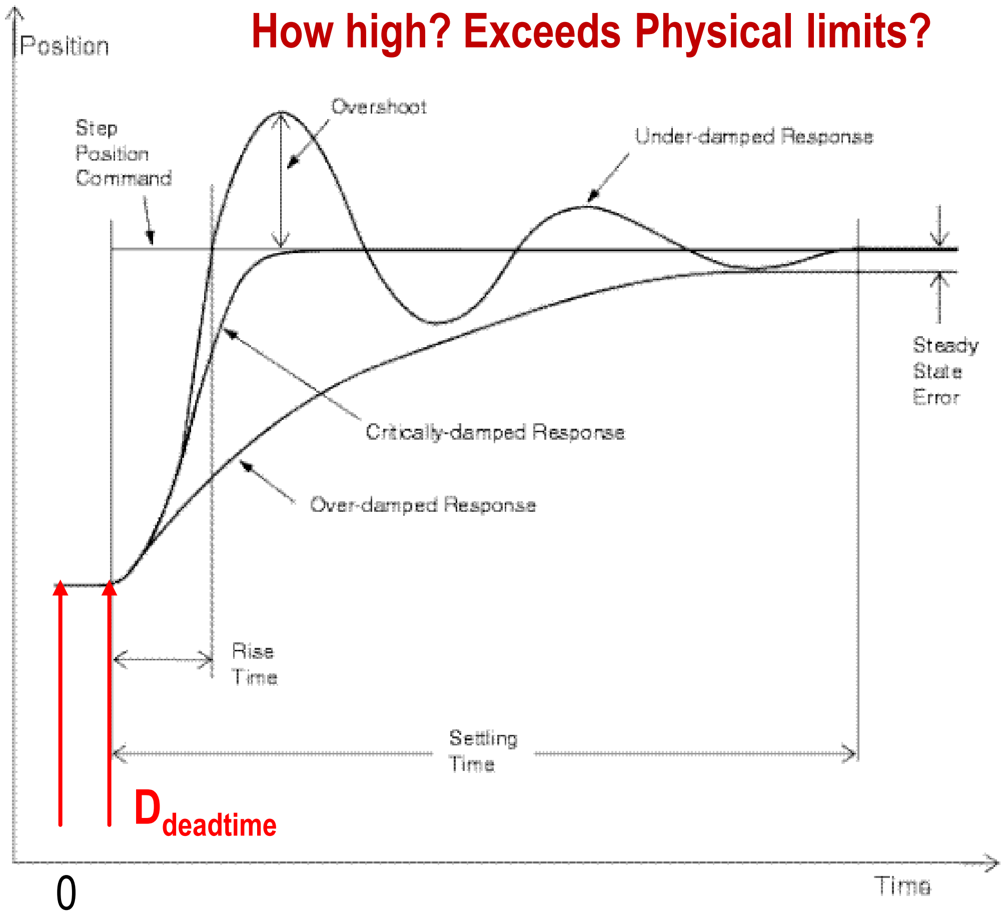
Other temporal characteristics of the physical plant:

Settling time: the time elapsed from the application of an ideal instantaneous step input to the time at which the output has entered and **remained** within a specified error band.

Rise time: the time required for the response to rise from 0% to 100% of its final value **(Ideal Definition), in practice to 90%, or 95%**

Dead time: Physical systems do not always respond immediately (*e.g.*, motor). When you send a current to a motor, it takes time to generate a field to overcome the inertia and initial torque to start moving

How high? Exceeds Physical limits?



Analysis and Design of Control Systems V

Another Example is the HDD

- In a computer HDD, transient response contributes to the time required to read from or write to the computer's disk storage
- Since reading and writing cannot take place until the head stops, the speed of the read/write head's movement from one track on the disk to another influences the overall speed/performance of the computer. The faster the cylinder rotates; the quicker we move the information to the head to read.
- Steady state response: head of a disk drive finally stopped at the correct track → otherwise errors

Think: Why do we rarely see HDDs with speeds more than 7200 RPMs?

Think: What do you think are the error margins for the steady state?

Control System Block Diagram Home Exercise

The control of the recording head of a dual actuator hard disk drive (HDD) requires two types of actuators to achieve the required high real density:

- The first is a coarse voice coil motor with a large stroke but slow dynamics.
- The second is a fine piezoelectric transducer (PZT) with a small stroke and fast dynamics.

A sensor measures the head position, and the position error is fed to a separate controller for each actuator.

Draw a block diagram for a dual actuator digital control system for the HDD.

Control System Stability

Transient Response and Steady-State error are irrelevant if the system is not stable.

Let's see unstable systems in action first:

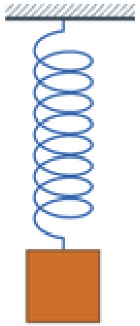
https://youtu.be/Dn1bVbstcOw

https://youtu.be/gFbP--OCZ-o

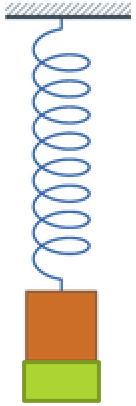
A Simple Stability Example

Initially, suppose a certain spring can hold a maximum weight of 7KGs, beyond which, the spring loses its elasticity and cannot return back to its original shape, and is therefore damaged.

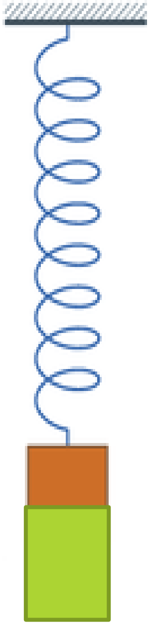
Consider that our system design initially requires attaching a mass of 3KGs to the spring. The spring stretches and reaches an equilibrium state. Now the spring system (spring + mass) is said to have a natural response. In this state, it responded to the initial conditions and no other forced inputs.



Now, during system operation, say we exert an external weight of 2KGs and the spring stretches. Since 2 + 3 < 7, when we remove the 2KG weight, the system will bounce and return to its former state. The 2KG weight is a forced input which gave a forced response.



Again, during system operation, say we exert an external weight of 5KGs and the spring stretches beyond its physical limitations. Since 5 + 3 > 7, when we remove the 5KG weight, the system will NOT bounce back and return to its former state. The 5KG weight is a forced input which gave a forced response that caused Instability and irrecoverable system damage



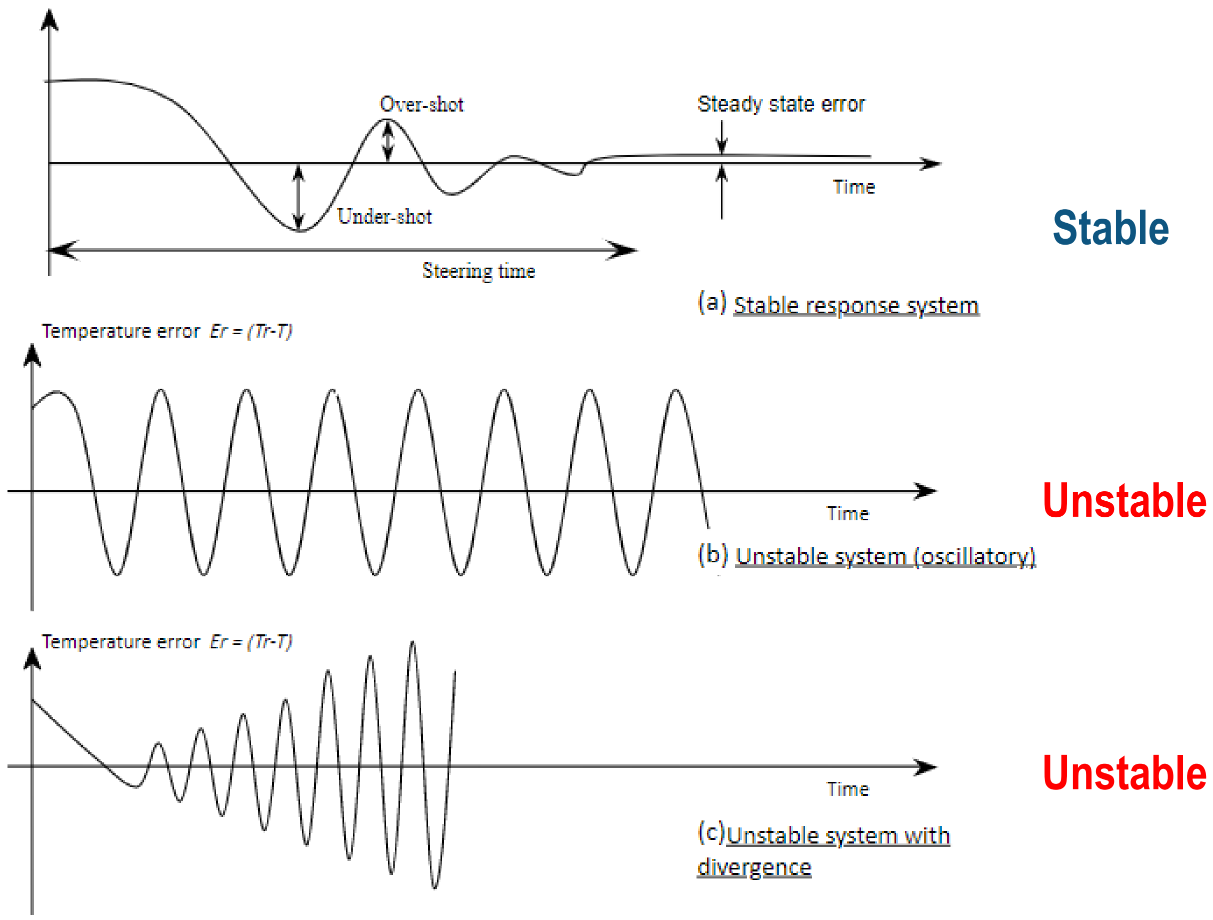
Instability

Instability, could lead to self-destruction of the physical device if limit stops (protection) are not part of the design.

Examples:

- an elevator would crash through the floor or exit through the ceiling;
- an aircraft would go into an uncontrollable roll;
- an antenna commanded to point to a target would rotate, line up with the target, but then begin to oscillate about the target with *growing* oscillations and *increasing* velocity until the motor or amplifiers reached their output limits or until the antenna was damaged structurally.

A time plot of an unstable system would show a transient response that grows without bound and without any evidence of a steady-state response.



Temporal Requirements in Computer (Digital) Control

In Analogue Control

Plant Temporal Characteristics
(settling time, rise time, dead time …)

In Digital (Computer) Control

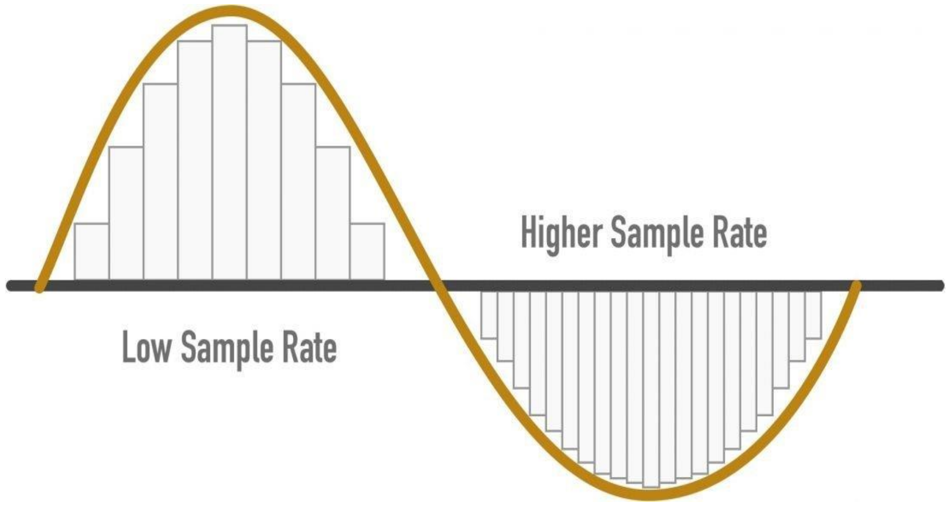
Plant Temporal Characteristics
(settling time, rise time, dead time, *etc.*)

+

Digital Controller Temporal Characteristics
(sampling time, processing time, deadlines, jittering and delays, *etc.*)

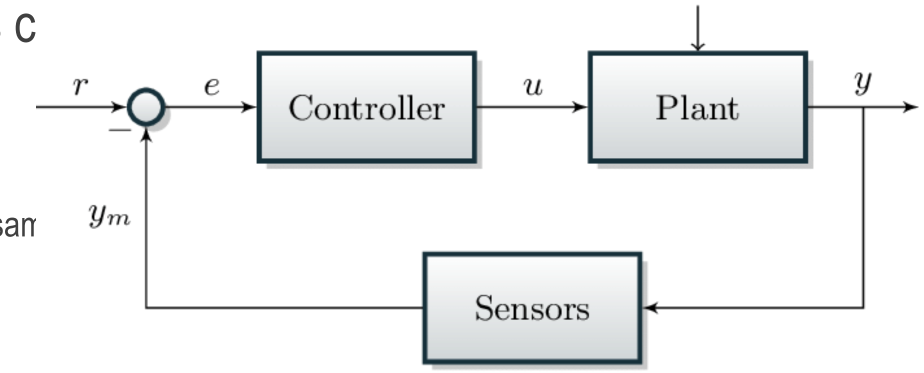
Temporal Requirements in Computer Control

- In an analogue control system without digital controllers, the temporal requirements are mainly concerned with analyzing the timing of the physical system (settling time, deadtime, risetime, transient time, *etc.*).
- In a digital system, more timing requirements are introduced → *e.g.*, ADC sampling rate ($T_{sampling}$)
- Choosing the sampling rate is not easy!
  - Must comply with Nyquist rate (*min* 2πf)
  - Might need to oversample to compensate for low ADC resolution.
  - Yet, at certain sampling rates, the system could become unstable.
  - If a control action needs to be computed every incoming sample, can the controller finish the required tasks ($T_{processing}$) related to one sample before the new sample comes in? That is; is $T_{processing} \leq T_{sampling}$? If not, what to do?



Exercise

- Suppose we have a control system as the one shown in the figure below. The controller is digital (e.g., computer / micrcontroller). Suppose that the sensors provide feedback data about the plant at a sampling rate of 1KHz. When a sample is received, some processing takes place such as calibration, filtering, computations and control decisions. Suppose it takes 1200 assembly instructions for this processing and on average, each instruction takes 1.5 clock cycles to execute. If the control processor is running at 1MHz, would this c

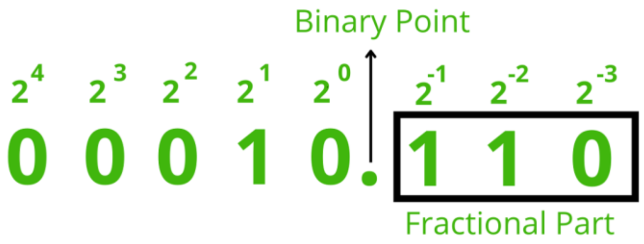


- Sampling Rate F_s = 1KHz → T_s = 0.001 seconds.
- In order for the system to satisfy the temporal requirments, $T_{processing} \leq T_{sam}$
- $T_{processing}$ = (1 / 1,000,000) x 1200 x 1.5 = 0.0018 seconds
- 0.0018 > 0.001
- This means it takes longer to process one sample before the next one arrives!
- What can we do to meet the deadline?

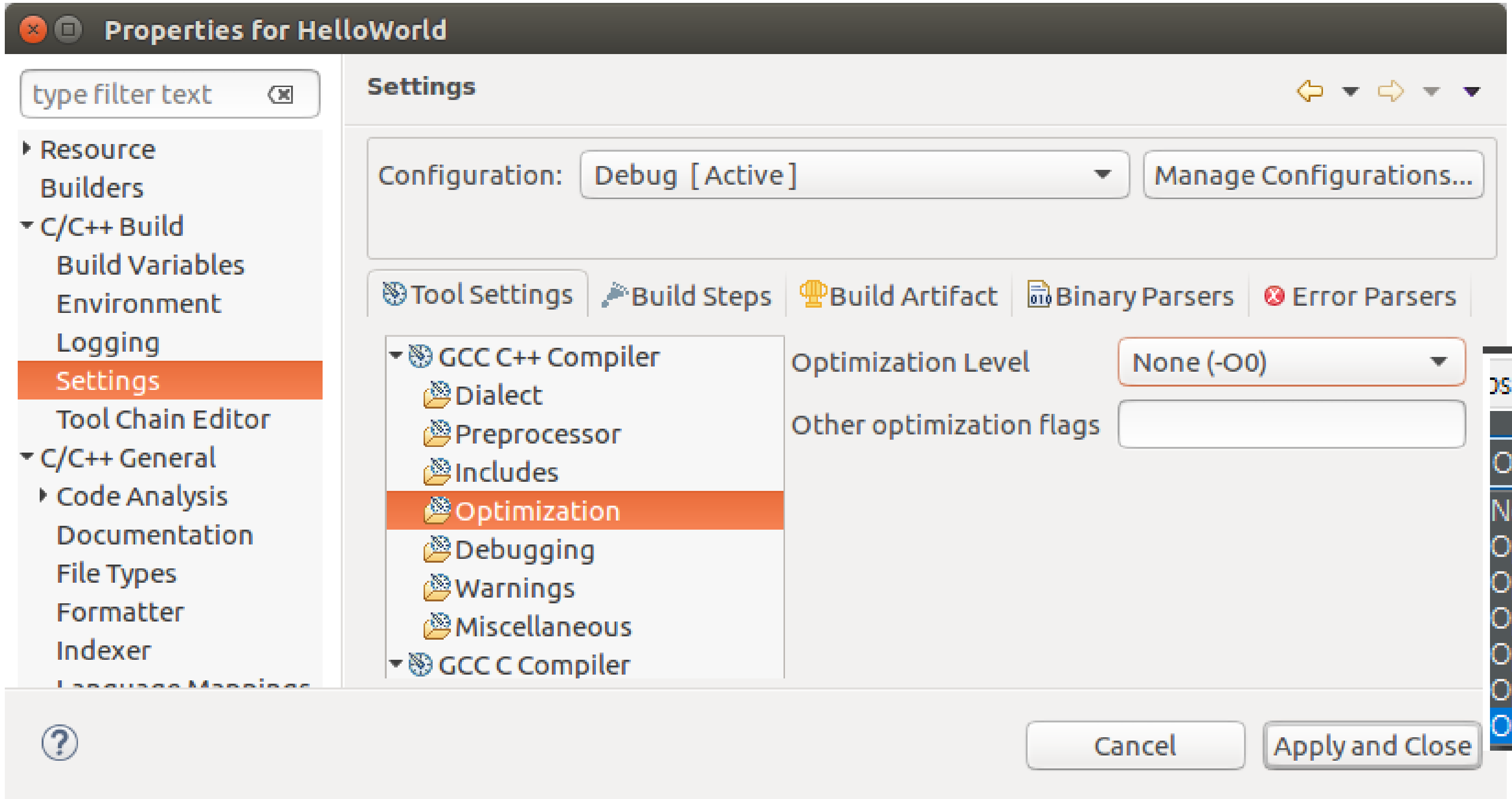
Exercise → Continued

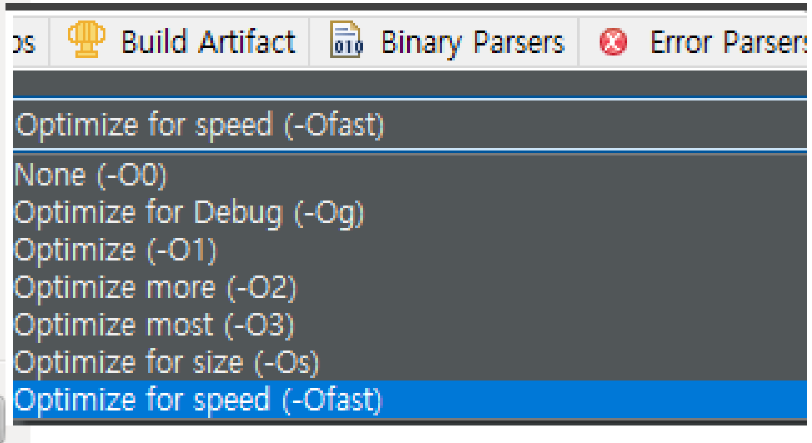
What can we do to meet the deadline?

- Use compiler optimisations (if available) to reduce code size or increase speed. For example, the *gcc* compiler has different options (see next slide).
- Use algorithm optimizations (if you are using an algorithm with O(n^2) complexity, rewrite your code with a better algorithm that takes less time, for example one with O(n log(n)) or O(log (n)) complexity.
- If you are using real numbers, and if precision can be slightly sacrificed without issue, use the fixed-point notation instead of floating-point notation to represent real-numbers. There are libraries for fixed-point representation. The fixed-point representation used the same hardware and pipelines used by integer operations, and is therefore much faster.
- If it is possible and the design requirements allow for it, increase the sampling time by sampling at a lower rate.
- Configure the processor to run at ahigher clock rate than 1MHz if it supports higher speeds.
- If none of the above is viable, then simply change the processor to one which can offer a better performance, but beware of costs.



GCC Compiler Optimization Options





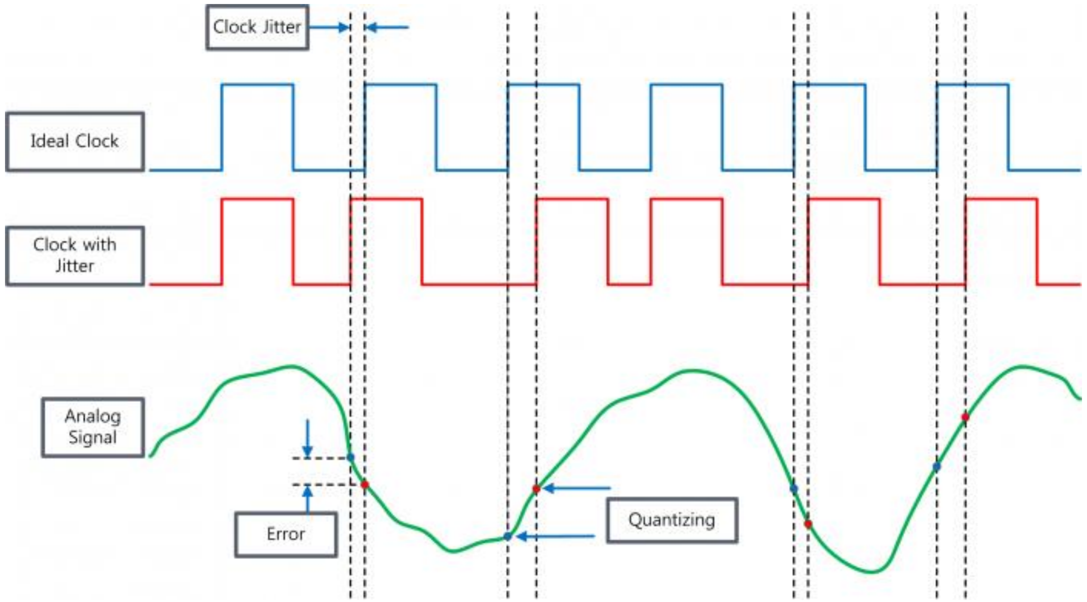
Jittering Issue in Digital and Control Systems

Sampling Jittering

- Samples should be always taken at the same time interval (even sampling). However, this period might vary by some time maxΔt (*i.e.* uneven sampling periods).
- If the variation is very small max(Δt) << $T_{sampling}$, it is almost negligible. The system can handle it and assume that the periods are still the same and constant. However, if
- If max(Δt) is proprtianally significant, this could pose a big problem (not capturing the correct signal properties)

Processing Jittering

- Suppose we have an event that we need to respond to, the controller could respond to it immediately, or if it is busy with a higher priority event, it could delay it a while. There is a variation in response time. This source of jittering is denoted as $\Delta t_{computer}$. However, some digital controllers can guarantee a maximum worst-case response time. This is useful to design for the worst case!



Temporal Requirements in Real-Time Control Systems I

Timing Requirements (*e.g.* deadlines) are categorized into**:**

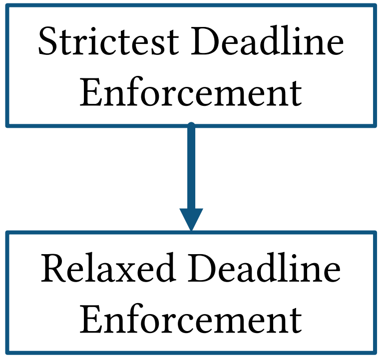
1. Absolute: response must occur at defined deadlines.
2. Relative: response must occur within a specified period of time following an event.

Real-time means:

1. Tasks must finish within a certain deadline
2. Tasks must not start before a certain time
3. The system must respond to external events quickly

Real-Time Systems Classification by Deadline Type:

1. **Hard** Real-Time Systems
2. **Firm** Real-Time Systems
3. **Soft** Real-Time Systems



Real-Time Control Systems Classification by Deadline Type I

1. **Hard** Real-Time (HRT) Systems

→ Under all circumstances, ALL 'hard (critical)' tasks MUST meet ALL their deadlines.

→ If not, system failure causes catastrophe or death.

→ Imperative that responses occur within the required deadline.

→ Response after deadline has no value!

- Guaranteed services required → functional correctness and timing correctness.
- Hard Real-Time Systems must be PREDICTABLE and DETERMINISTIC
- Analysis of estimated worst-case time → Scheduling algorithm and system must pass schedulability test
- In practice, the time bounds for HRT ranges from microseconds to milliseconds.
- Deadline does not necessarily imply "imminency" → Hard real-time task does not need to be completed within the shortest time possible (fast computing) → Only within the bound

Real-Time Systems Classification by Deadline Type II

2. Firm Real-Time Systems

→ Tasks missing their deadline **will not** result in a system failure, and no catastrophe or death

→ Infrequent misses lead to ***performance degradation*** **(loss of QoS)**

→ Response following a deadline has no value

3. Soft Real-Time Systems

→ Deadlines desired to be enforced, but they are not strict. (Best-effort service → deals with average response times)

→ Frequent deadline misses do not cause errors, but the result of the task **might** no longer be as useful.

→ Response following a deadline is not wasted, but degrades as more time passes

→ Usually specified by some probability? What is the probability that task A misses its deadlines 10% of the time?

→ Probabilistic analysis → complexity at design time!

→ Time bounds between fraction of a second to few seconds

- Complex real-time control systems could consist of subsystems of any of the three types (airplane control systems)

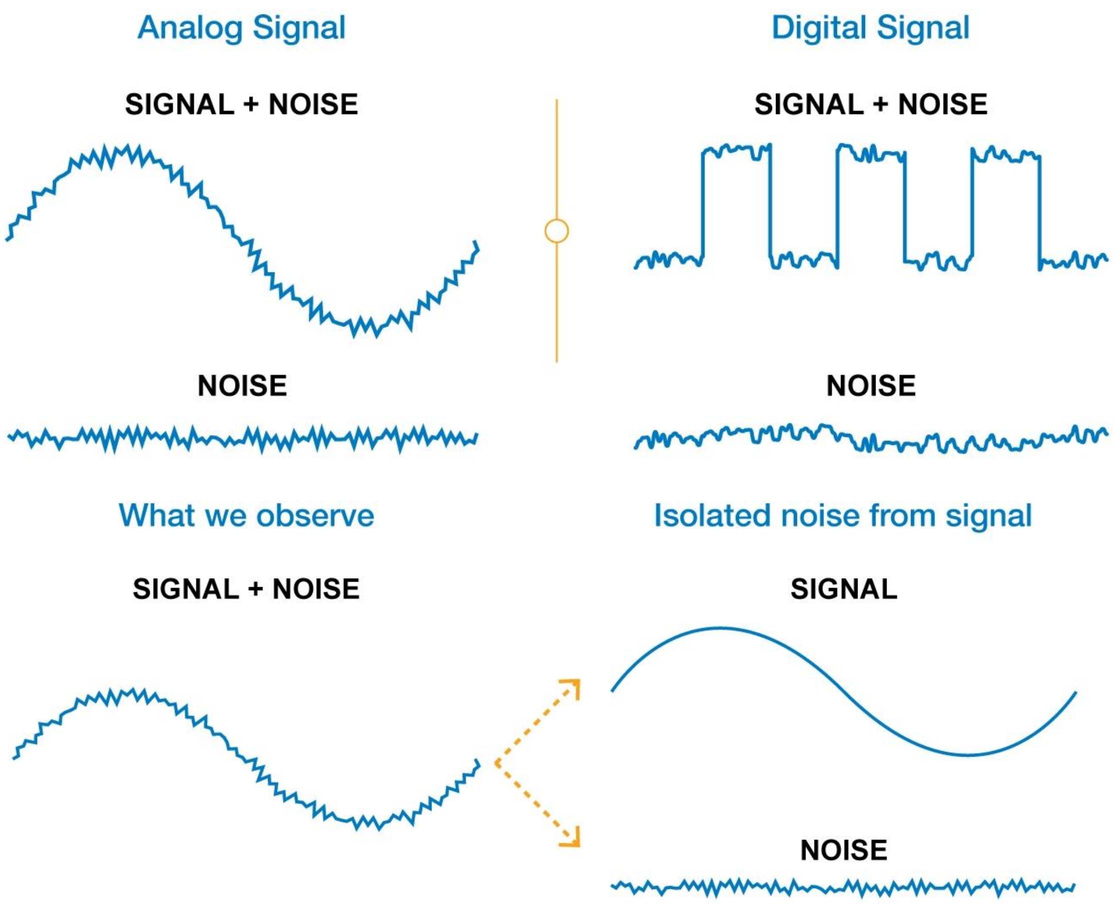
Functional Requirements

A. Data Collection / Acquisition

B. Signal Conditioning

Signal conditioning is used to refer to all the processing steps that are necessary to obtain meaningful *measured data* of plant from the raw sensor data.

- Sensors produce raw data! (*e.g.,* voltages, currents).
- Scaling to required values (*e.g.,* sensor voltage to input voltage of port pin)
- Inherent Measurement errors (*e.g.* A/D Quantization).
- Sensors also need calibration at many times.
- Noise effects → Must be filtered out (Anti-Aliasing filters, DC Filters, Digital Filters, etc.)



C. Plant / Process Control

D. (Optional) Alarm Monitoring / Data Logging / Man Machine Interaction

Non-Functional Requirements Overview

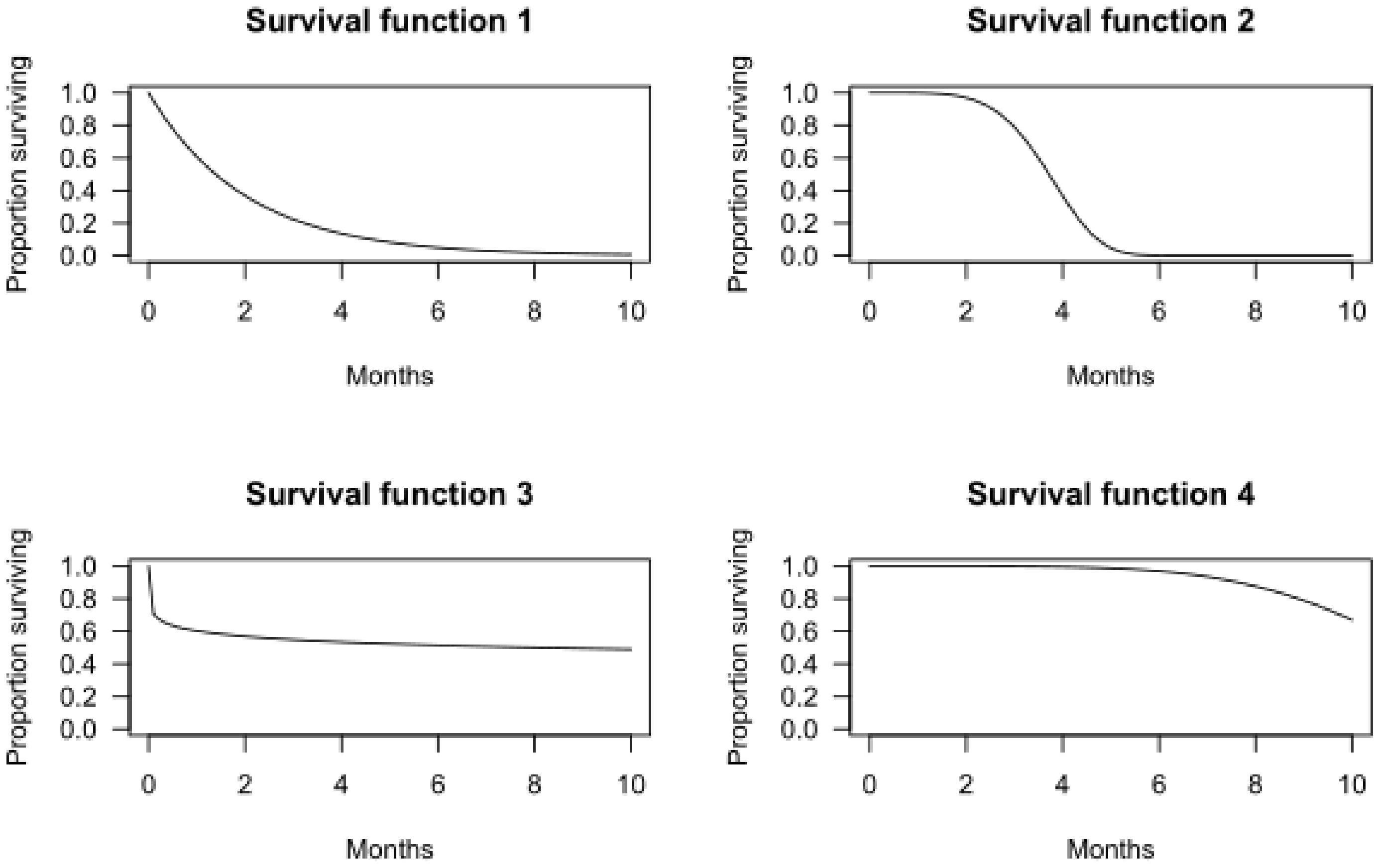
1. Dependability:
    - Reliability (survivability)
    - Safety
        - Fault-tolerance
        - Security
2. Performance
3. Robustness
4. Scalability

Dependability Requirements I

Reliability (survivability): probability that a control system will provide the plant control task until time t, given that the system was operational at $t = t_0$

There are many different survivability functions based on observations. One example is the exponential case $R(t) = e^{-\lambda(t-t_0)}$ corresponding to the first graph, where λ is constant failure rate in failures/hour.

When λ is smaller, it is better. In large scale industrial applications and manufacturing, Ultra-high reliability when $\lambda < 10^{-9}$ as in the avionics standards.



Dependability Requirements I

One equation to represent the failure late $\lambda$is (as given in W. Bolton's book ***Instrumentation and Control Systems***, 3rd edition, 2021):

$$Failure\ Rate\ (\lambda) = \frac{Number\ of\ Failures}{Number\ of\ Systems\ Observed\ \times Time\ Observed}$$

Example:

If a car company produces **1 million cars** in one year, and on average each cars drives **two hour a day each day of the year**, and only **one car fails** during this whole year:

Num of failures = 1

Num of systems observed x time observed = 1,000,000 $\times$ 2 $\times$ 365 = 730,000,000 hours

Failure rate λ = 1/ 730,000,000 = 1.36 x 10^{-9}, which means that the reliability λ is close to the order of 10^{-9} set in some standards

Dependability Requirements II

Safety requires certification

Safety: defined as 1. responses to protect the system from unintentional harm (*e.g.,* error detection)
2. reliability against critical failure modes (*e.g.,* plane crash, self-driving car accident)

For example, *safeRTOS* is a certified safe version of *freeRTOS*

Components must successfully pass certain tests like the FCC, CE, EMC

Must comply with certified industry safety standards (*e.g.* aviation or automotive safety standards)

- **Fault-tolerance:** protection from design and operational faults? How?
  Hardware redundancy → *E.g.,* Two lock-step processors in tandem / multiple sensors.
  In software, roll back/recovery and checkpoints (similar to computer games ☺), however; in hard real-control systems:
  1. Difficult to guarantee a deadline when error occurs → roll-back and recovery can take unpredictable time.
  2. The error could have caused irrevocable action (remember we are connected to other hardware which affects the plant controlled)
  3. Temporal accuracy of the checkpoint data is invalidated by passage of time
- **Security**: protect system from intentional harm or access

Dependability Requirements III

2. Performance: timing of responses or throughput necessary?

3. Robustness: protection from external interference and perturbations

4. Scalability: Perform reasonably in an environment with added load

Fail-safe and Fail-Operational Control Systems

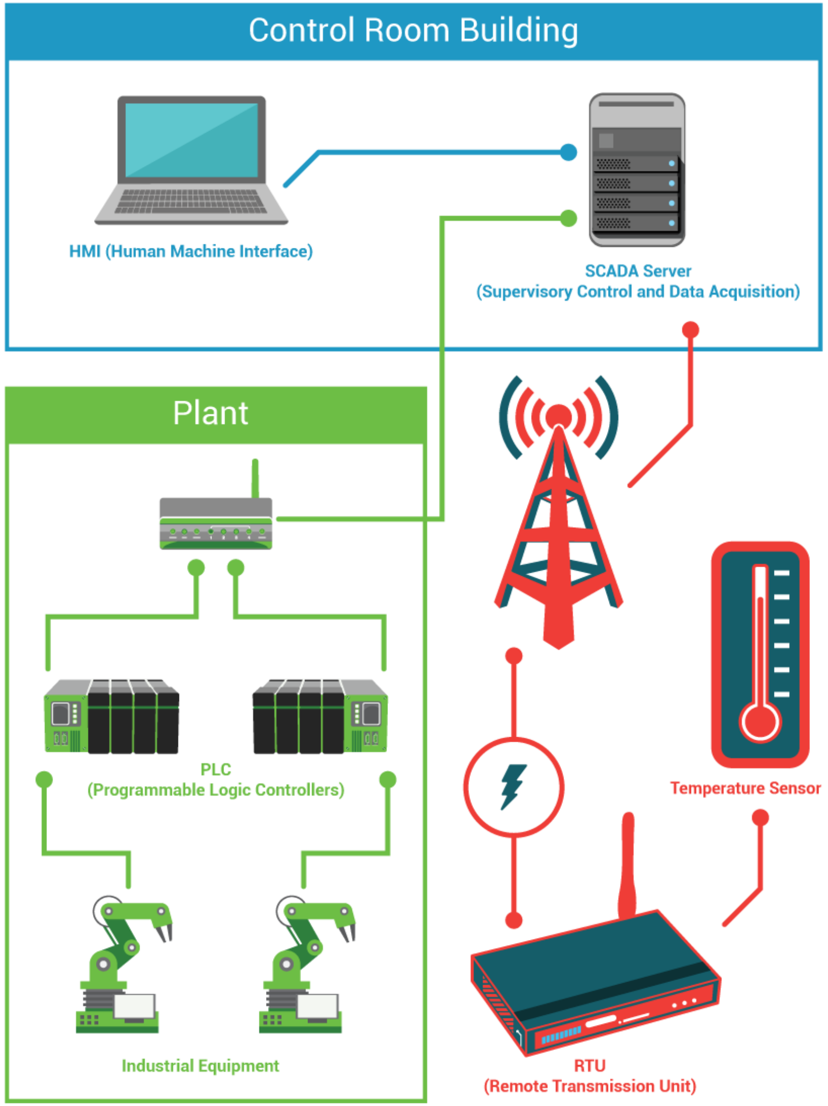
- Some control systems can have safe states **(fail-safe)** → When system fails, go to safe state
  Examples:
  electronic → Simple Electric Fuse
  In control and embedded software → Watchdog timers
  In Nuclear reactors → Magnetically held lead rods

  * Requires high error-detection coverage → the probability that an error is detected, provided it has occurred, must be close to one

- In certain applications, you cannot identify a safe state!
  Example: Flight control system of airplane or space craft!
  * Must provide minimum level of service to avoid catastrophe even if failures occur, or sound alarms.
  * If main power shuts down, switch to auxiliary power in hospitals.
  * These systems are called ***fail-operational***

SCADA System Definition

- SCADA is an acronym that stands for ***S***upervisory ***C***ontrol ***A***nd ***D***ata ***A***cquisition.
- A SCADA system is a combination of hardware and software that enables the automation of industrial processes by capturing operational real-time data.
- SCADA refers to a system that collects data from various sensors that monitor equipment like motors, pumps, and valves at a factory, plant or in other remote locations, and then sends this data to a central computer which then manages and controls the data.
- The size of such systems range from a few 1000 to several 10 thousands input/output (I/O) channels.
- A SCADA system empowers organizations to:
  - Control processes locally or at remote locations
  - Acquire, analyze and display real-time data
  - Directly interact with industrial equipment such as sensors, valves, pumps, and motors
  - Record and archive events for future reference or report creation or issue alarms when necessary



SCADA System Architecture Layers

Level 0 (Field level - Sensors/ Actuators):

- The field level includes field devices, such as sensors, used to forward data relating to field processes and actuators used to control processes.

Level 1 (Programmable Devices for Direct Control):

- The direct control level includes local controllers, such as **PLCs (Programmable Logic Controllers)** and **RTUs (Remote Terminal Units)**, that interface directly with field devices, including accepting data inputs from sensors and sending commands to field device actuators. A SCADA system can be built with only local area networks or a combination of both local and wide area networks. The PLC helps to build the SCADA system with local area network only whereas the RTU system helps to build the SCADA system with the wide-area network.

Level 2 (Local Plant Control and HMI):

- The plant supervisory level includes local supervisory systems that aggregate data from level 1 controllers and issue commands for those controllers to carry out. From these computers, the actual instructions and commands are given to do the operations. The supervisory computer may be connected to a particular machine or multiple same types of machines or a whole manufacturing plant. These computers are operated by machine operators, plant supervisors, and technicians of a manufacturing plant. The main functions of these computers are to observe and control the production, errors, *etc*.

SCADA System Architecture Layers

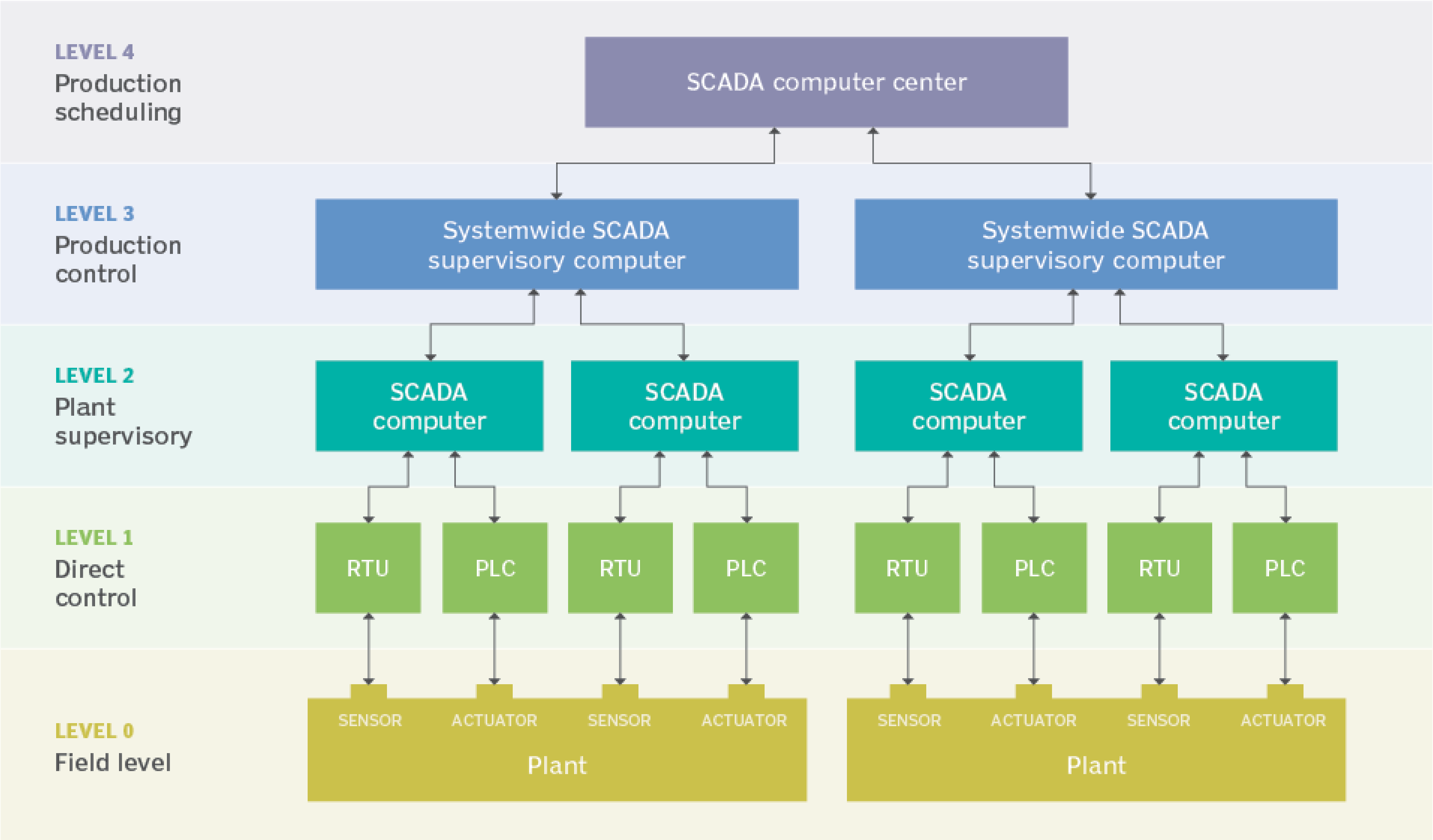
Level 3 (Coordination and Production Control):

- The production control level includes systemwide supervisory systems that aggregate data from Level 2 systems to produce ongoing reporting to the production scheduling level, as well as other site or regionwide functions, like alerts and reporting. Generally, these computers are connected to multiple plants. So, it can help to gather data from different plants from one place. This can take place using the cloud.

Level 4 (Central Control and Production Scheduling):

- The production scheduling level includes business systems used to manage ongoing processes. At this level, a central computer is connected to all the plants and machinery. Generally, this is operated and controlled by the management team. All the data and information are collected and stored here. Using these data and information they can take any decision. From this computer, the management team can see all the actions and operations, *etc*.

SCADA System Architecture Layers



SCADA System HMI Example

Control Systems Definitions Summary I

Plant	A plant may be a piece of equipment, perhaps just a set of machine parts functioning together, the purpose of which is to perform a particular operation (*e.g.,* hydraulic arm, motor)
Controlled Variable	The quantity or condition that is measured and controlled (system output)
Transducer	A transducer is a device that converts a signal from one form to another. Transducers are often employed at the boundaries of automation, measurement, and control systems, where electrical signals are converted to and from other physical quantities (energy, force, torque, light, motion, position, *etc*.). Transducers examples are sensors and actuators.
Sensor	A sensor is a transducer that receives and responds to a signal or stimulus from a physical system. It produces a signal, which represents information about the system, which is used by some type of telemetry, information or control system
Actuator	An actuator is a device that is responsible for moving or controlling a mechanism or system. t is operated by a source of energy, which can be mechanical force, electrical current, hydraulic fluid pressure, or pneumatic pressure, and converts that energy into motion. An actuator is the mechanism by which a control system acts upon an environment.
Control signal	The quantity or condition that is varied by the controller so as to affect the value of the controlled variable

Control Systems Definitions Summary II

Control	Control is the act of measuring the value of the controlled variable of the system and applying the control signal to the system to correct or limit deviation of the measured value from a desired value.
Processes	A process is the totality and series of actions affecting each other in a system and are carried out in order to achieve a particular result.
Disturbance	A Disturbance is signal that tends to adversely affect the value of the output of a system and could be: **internal:** generated within the system **external:** generated outside the system and is an external input to the process.
Open-Loop Control	Open Loop Control is a process where one or more input variables of a system act on a process variable. The actual value of the process variable is not being checked, with the result that possible deviations *e.g.* caused by disturbances are not compensated for in the open loop control process. T
Closed-Loop (Feedback) Control	Closed loop control is a process whereby one variable, namely the variable to be controlled (controlled variable) is continuously monitored, compared with another variable, namely the reference variable and, depending on the outcome of this comparison, influenced in such a manner as to bring about adaptation to the reference variable even in the presence of disturbances which are eventually reduced / eliminated. The characteristic feature of closed loop control is the closed action flow in which the controlled variable continuously influences itself in the action path of the control loop.

Required Reading

- Chapter One from Control Systems Engineering, Norman S. Nise, 6th (2010) or 7th (2014) or 8th Edition (2020), John Wiley And Sons
- Chapter One from Digital Control Engineering, Sami Fadali and Antonio Visioli, 3rdEdition, Elsevier Press, 2020