Introduction to Control System

The terms, **control** and **system** are closely interrelated. Control is the process of forcing a **system output variable** to conform to some desired value, called **reference value**.

In order to gain a better understanding of the task of control, a simple example is considered. Driving a car is an excellent example of **control system**. The driver has to follow the given direction of a road. He/she observes the actual path of the car and then forces the car, operating the steering wheel, to track the desired path as closely as possible. The driver performs the following steps in detail:

- The driver uses his eyes as **sensors** for obtaining measurements, both of the car's actual path and the road course.
- Then he/she compares both directions and generates an **error signal**, which is used to decide in which direction to move the steering wheel.
- The driver **actuates** the steering wheel according to his decision, making the car, the controlled object, move to the desired direction.

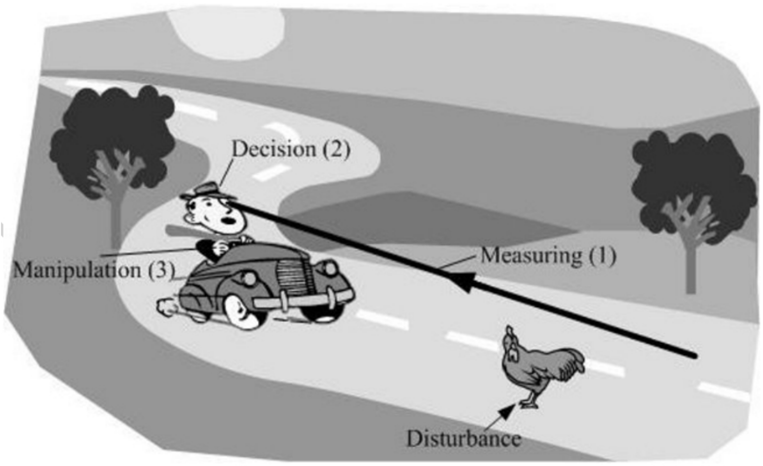


Figure 1: Manual control of a car's direction of travel

An animal or any hindrance on the road acts as a **disturbance** and should be avoided if possible. After reaching such a disturbance, the driver must return the car to the desired direction. These three steps of measuring, decision, and manipulation are characterizing the driver's control action (Figure 1).

There are 2 basics form of control system: open-loop system and closed-loop system.

What is an open-loop control system?

Example: Table Fan

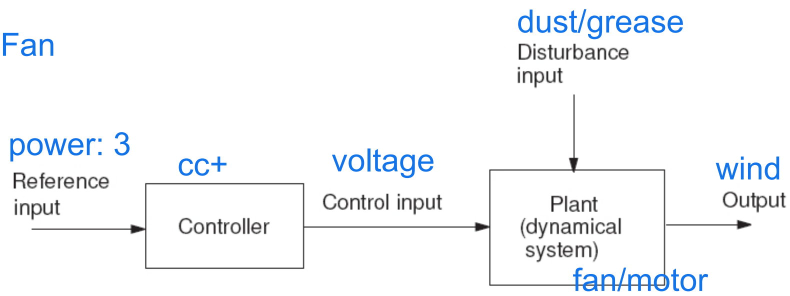


Figure 2: A general open-loop control system

Those systems in which the **output has no effect on the control action** are called ***open-loop control systems*** (Figure 2). In other words, in an open-loop control system the output is neither measured nor fed back for comparison with the input.

One practical example is a **simple fan**, where:

Plant/Process/System - refers to the system (simple fan) we want to control. It generally refers to a physical process which we can either model or measure.

Reference input/Desired Output - Switch on the fan (that is, press the switch and 230 V is applied). So, the reference input is the 230 V signal.

Controller - The electronic voltage controller (that is, turn the knob to the desired position). The effect is to reduce/change the voltage to the appropriate value. We may have approximately 230 V (= full speed) and 115 V (half-speed), and so on.

Once the speed is set there is nothing else that needs to be done. But suppose you have three fans. Even if you give their knobs the same amount of turn, the speeds are likely to be slightly different. This may happen due to inaccuracy in the settings, inconsistency in ball bearings performance, imperfect setting of the fan blades causing different amount of drag on the blades, or maybe due to non-standard performance of the electronic components.

So, essentially an open-loop system is one where there is no way to correct the error between the desired output and the actual output.

What is a closed-loop control system?

Feedback control systems are often referred to as ***closed-loop control systems***. In practice, the terms feedback control and closed-loop control are used interchangeably.

Consider the same electronic fan control switch. Assume that you are looking at the fan blades to make sure that the speed is right. If it isn't, then you turn the knob continuously till the desired speed is achieved. The block diagram in Figure 3 is not an exact representation of this, but it conveys the idea in a broad sense. The measurement device or **transducer** is to convert the output signal to an equivalent electrical system (e.g. from kinetic energy to electrical energy). This facilitates the comparison of the output signal to the input.

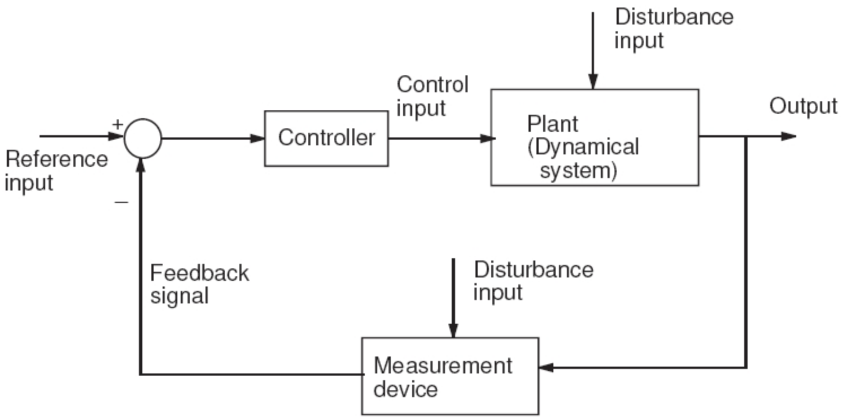


Figure 3: A general closed-loop control system

Figure 4 shows an example of closed-loop control in mobile robotics.

open loop and closed loop not tested in details, just understand the concept to write theory only

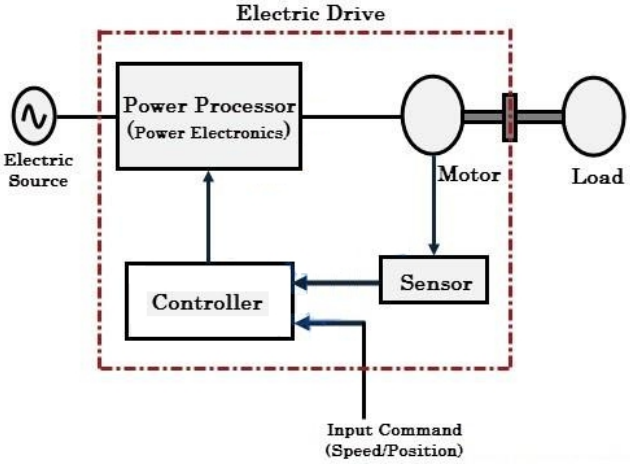


Figure 4: A closed-loop control in mobile robotics

- The measured speed at the motor shaft is fed back and compared with reference speed.
- The difference speed error is applied to the controller to generate a control voltage which controls the power processor and produces the desired terminal voltage.
- This terminal voltage controls the speed of the motor and thus the speed of the motor can maintained for any variations in the load torque.
- If the load torque has been increased, due to this high load the motor speed reduces momentarily from its desired value.

Closed-loop systems have different characteristics when compared to open-loop systems. The following summarizes the advantages and disadvantages of closed-loop system:

Advantages:
- Highly accurate as any error arising is corrected due to presence of feedback signal.
- They are less sensitive to disturbances.
- They are less sensitive to system characteristics/parameter variations.

Disadvantages:
- They have a tendency to oscillate.
- They are complicated to design.
- Required more maintenance.
- Stability is the major problem and more care is needed to design a stable closed loop system.

It is possible to use relatively inaccurate and inexpensive components to obtain the accurate control of a given plant, whereas doing so is impossible in the open-loop case. From the stability point of view, the open-loop control system is easier to build because system stability is not a major problem. On the other hand, **stability** is a major problem in the closed-loop control system, which may tend to overcorrect errors that can cause oscillations of constant or changing amplitude.

As control engineer we must be able to:

- model/measure the dynamic behaviour of the plant/system.
- choose an appropriate controller that allows the system output to meet a list of user designed criteria.

Obstacle avoidance

In order for a robot to avoid running into obstacles, different sensory systems that can be used. For example, whisker sensor that triggers a switch when an obstacle is encountered. If the robot is to avoid any contact, then a proximity sensor, such as reflected IR could be used. Range sensors such as sonar or laser sensors allow detection of obstacles to occur over longer distances. Besides reactively sensing the obstacle and taking evasive action, a robot can use prior knowledge of the location of obstacles and planning a route that avoids obstacles. This method requires the robot to have more explicit knowledge of the world around it, and its current position.

Dead reckoning

Dead reckoning or path integration uses basic geometry to work out the position of a robot, given an initial position and knowledge of the movement that occurred. A simple case is where the robot makes independent rotations and straight line movements. It could use an optical encoder on its wheels to estimate the heading angle θ and the distance travelled, d. Problems with dead reckoning occur because small errors in the distance or heading estimate can accumulate into large errors in position.

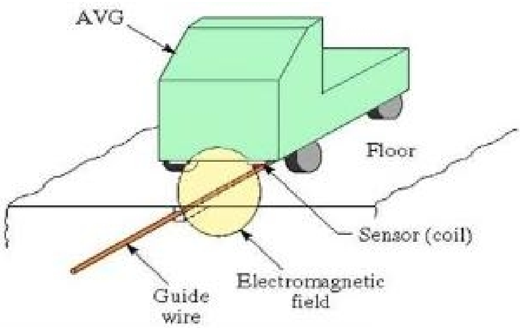
To overcome the limitation, a popular solution is to have some external reference points in the world that can be used to recalibrate the system. These points are called landmarks. They are perceptual distinctive features in the environment that the robot can recognise, preferably from different viewpoints, and use to orient itself. Good properties for a landmark is that it does not move or change, can be detected over a wide area, unique, and something already existing in the environment that can be readily adopted for navigation.

A mobile robot can use several control methods to orient towards a landmark.

- Assume the robot has detected a landmark at 45 degrees to the right and 2 metres away. Using open loop control, it would need to calculate the motor command(s) needed to turn 45 degrees and move 2 metres. It could do this using an inverse model: e.g. from the robot geometry an equation relating the motor signal to the end position of the robot could be derived, and this could be inverted to obtain an expression that, given a desired end position, specifies the motor commands.
- Feed-forward control would deal with this by trying to measure such disturbances and adjust the motor output accordingly, e.g. adding an appropriate sum to the motor command to compensate for the detected wheel slip.

Automated Guided Vehicle (AGV)

Automated Guided Vehicle (AGV) is an unmanned, computer-controlled mobile robot that is used to transport material and goods from one place to another.

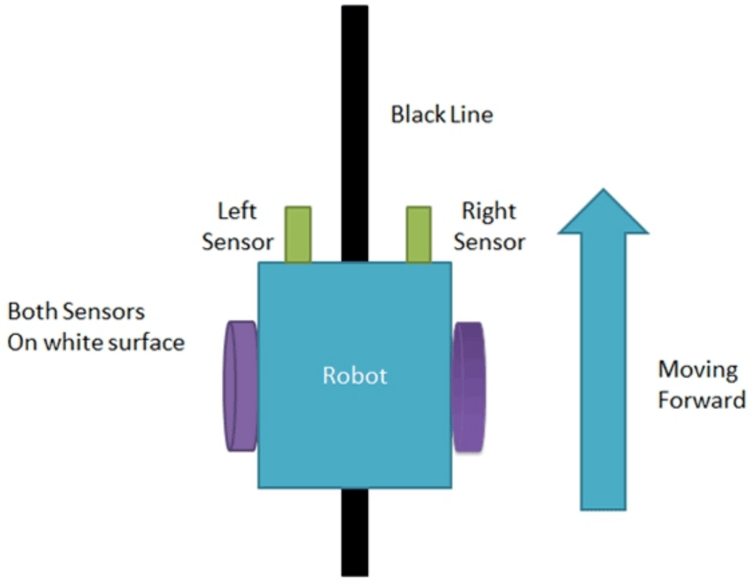


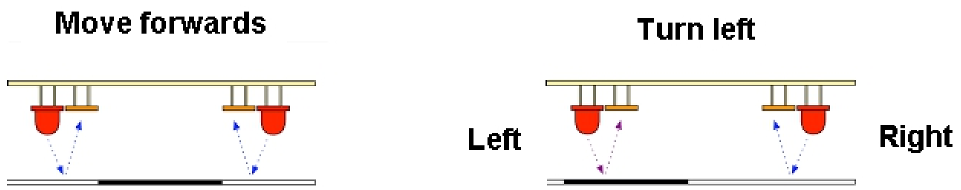
Wired navigation is one of the techniques used by AGV. The wired sensor is installed on the bottom of the AGV's and is placed facing the ground. A slot is cut in the ground and a wire is placed below the ground. This wire is used to transmit signal. The sensor detects the relative position of the signal being transmitted from the wire. This information is used to regulate the steering circuit, making the AGV follow the wire.

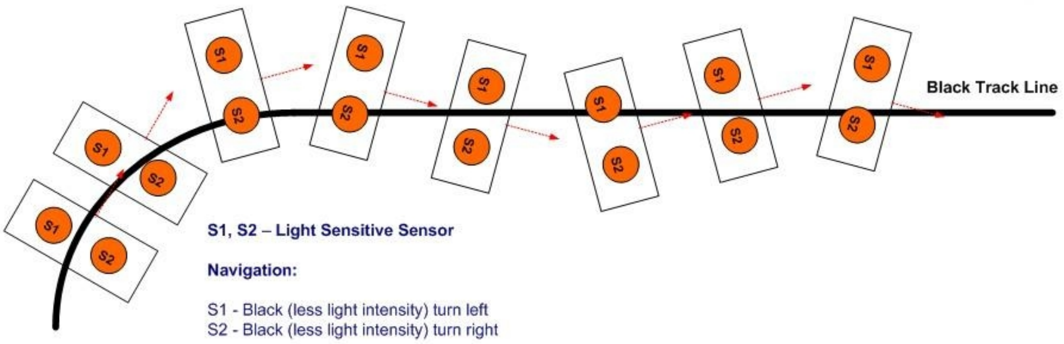
AGVs are commonly used to transport raw materials such as paper, steel, rubber, metal, and plastic. This includes transporting materials from receiving to the warehouse, and delivering materials directly to production lines. Pallet handling is an extremely popular application for AGVs as repetitive movement of pallets is very common in manufacturing and distribution facilities. AGVs can move pallets from the palletizer to stretch wrapping to the warehouse/storage or to the outbound shipping docks. Moving finished goods from manufacturing to storage or shipping often require the gentlest material handling because the products are complete and subject to damage from rough handling. Because AGVs operate with precisely controlled navigation and acceleration and deceleration this minimizes the potential for damage making them an excellent choice for this type of application

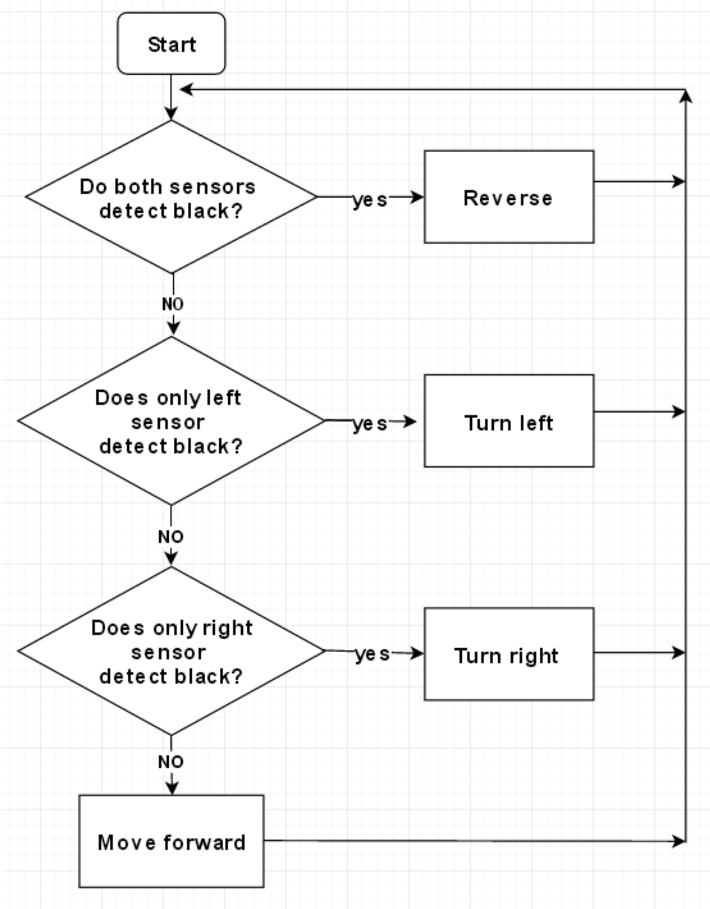
AGV is not suitable for non-repetitive tasks. This is because the hardware is coded to repeat the same task for equal interval of time. If the task is altered under certain condition, the AGV'S cannot be used unless the software is altered which is major drawback and the whole manufacturing unit will be stopped.

Line follower robot









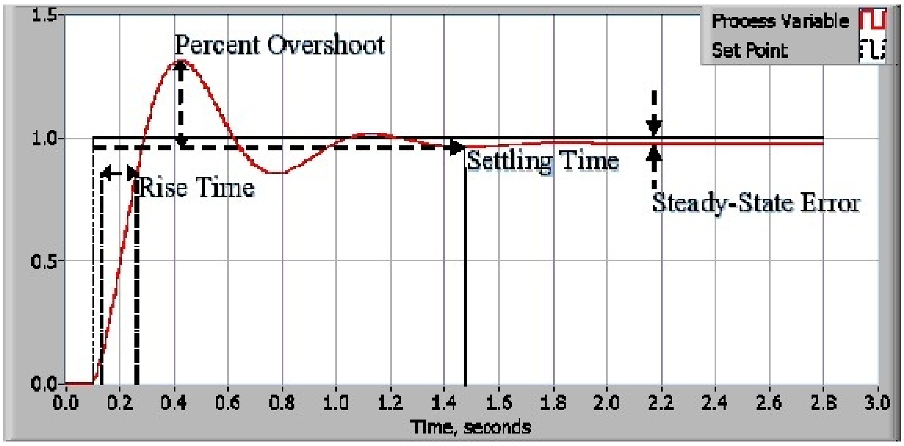
PID Controller

A **PID controller** is an instrument used in industrial control applications to regulate temperature, flow, pressure, speed and other process variables. PID (proportional, integral, derivative) controllers use a control loop feedback mechanism to control process variables and are the most accurate and stable controller.

- Proportional component: it can be considered as a virtual spring. A large proportional gain usually results in reduced rise time and steady-state error but it would also make the system overshoot and oscillate.

- Integral component: it ensures zero steady state error. A large integral gain will also make the system react faster to some constant error or disturbance, but can result in overshoot and makes transient response very poor.

- Derivative component: it should behave like a virtual damper. Increasing the derivative gain generally will reduce overshoot/oscillation. However, the derivative gain has no effect to the steady- state error and rise time.



Closed-Loop Response	Rise Time	Overshoot	Settling Time	Steady-State Error	Stability
Increasing K_P	Decrease	Increase	Small Increase	Decrease	Degrade
Increasing K_I	Small Decrease	Increase	Increase	Large Decrease	Degrade
Increasing K_D	Small Decrease	Decrease	Decrease	Minor Change	Improve

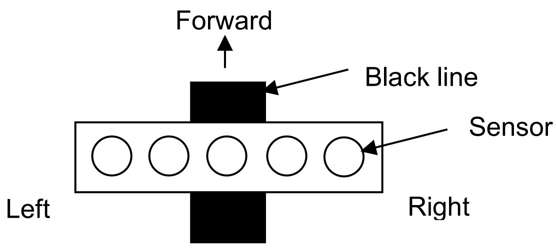
General tips for designing a PID controller When you are designing a PID controller for a given system, follow the steps shown below to obtain a desired response.
1. Obtain an open-loop response and determine what needs to be improved
2. Add a proportional control to improve the rise time
3. Add a derivative control to improve the overshoot
4. Add an integral control to eliminate the steady-state
5. Adjust each of Kp, Ki, and Kd until you obtain a desired overall response.

For example, if a robot takes a reasonable amount of time to settle. Then it begins to oscillate again with bigger and bigger oscillations. This is probably due to Ki is too high and is causing ramp up. To fix this, reduce Ki.

You can always refer to the table to find out which controller controls what characteristics. Lastly, please keep in mind that you do not need to implement all three controllers (proportional, derivative, and integral) into a single system, if not necessary. For example, if a PI controller gives a good enough response, then you don't need to implement derivative controller to the system. Keep the controller as simple as possible.

Proportional-Derivative (PD) Control can be used to make the robot follows the line smoothly and make less error.

Diagram:



- Next, determine the error of deviation using the formula:

  Error = current position – set point value

- Let the left most sensor is be '1', the middle sensor be '3', and the right most sensor is be '5'. This will give negative value when the robot is deviating to the right, or positive value when the robot is deviating to the left.

- Then, applying the PD formula to find the required change in speed,
  speed change = Kp * error + Kd * (error – previous error)

  where Kp and Kd are the constants that need to be determine through experiments, while the previous error is the error before this iteration.

- The last step is to adjust the robot motors speed by how much they need to change as follows,

  right motor speed = base speed – speed change
  left motor speed = base speed + speed change

  where, the base speed is the initial speed of the robot motors

Example:
Using an array of 5 sensors, the error value can be generated and used to control the robot's position over the black line as follows,

Sensor Array
0 0 0 0 1 ==> error = 4
0 0 0 1 1 ==> error = 3
0 0 0 1 0 ==> error = 2
0 0 1 1 0 ==> error = 1
0 0 1 0 0 ==> error = 0
0 1 1 0 0 ==> error = -1
0 1 0 0 0 ==> error = -2
1 1 0 0 0 ==> error = -3
1 0 0 0 0 ==> error = -4
Where bit 1 represents the detection of black line by the sensor.

Case 1:
- For example, if the error is 1, the robot has to turn right in order to align the black line at the centre.
- The right motor must slow down while the left motor must speed up.
- The speed of right motor has to be reduced by Kp*error, while the speed of left motor has to be increased by Kp*error.

Case 2:
- On the other hand, if the error is -1, the robot has to turn left.
- The right motor must speed up while the left motor must slow down.
- The speed of right motor has to be increased by Kp*error, while the speed of left motor has to be reduced by Kp*error.

- Note that Kp is the proportional gain and it can be determined experimentally or by using optimisation algorithm.
- Hence, as much the robot driven to one side, bigger will be the error and faster it must return to centre.
- The velocity with the robot will react to the error will be proportional to it.

- Firstly, an array of 5 sensors is mounted below the follower robot on a way that only one sensor is centered with relation to the line. The middle sensor will be a set point and the robot will always try to adjust itself to center at the set point.

Explaining Open and Closed loop Systems in Robotics - Control System Engineering:
https://www.youtube.com/watch?v=f32SDx3C6sA

Introduction to PID Control
https://www.youtube.com/watch?v=wkfEZmsQqiA&ab_channel=MATLAB
https://www.youtube.com/watch?v=UR0hOmjaHp0&ab_channel=BrianDouglas
https://www.youtube.com/watch?v=XfAt6hNV8XM&t=110s&ab_channel=BrianDouglas
https://www.youtube.com/watch?v=8MVjQZp8Xcw&ab_channel=DImuthuUpeksha
https://medium.com/luos/an-introduction-to-pid-control-with-dc-motor-1fa3b26ec661