Basics of Control Theory

1. What is Control Theory?

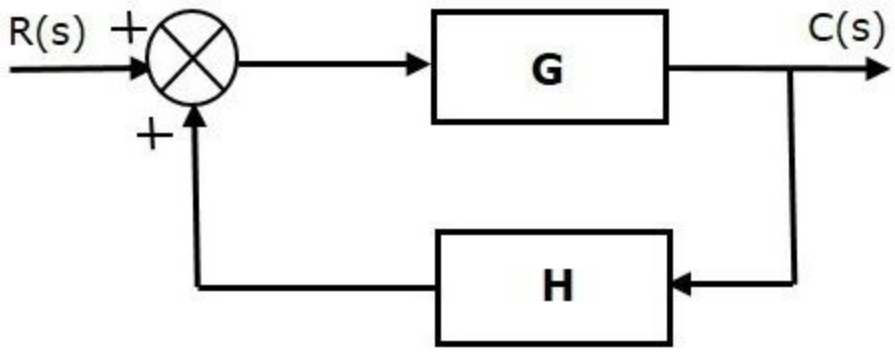
Control Theory is the study of various techniques that we can deploy to control dynamic systems. *A control system is a set of devices arranged in a specific configuration in order to produce a desirable output, given the input.* This might seem a little arbitrary at first, so check out this video.

A simple control system can be represented using the following block diagram:

Any control system falls under one of the following main categories :

Open Loop Control System : In an open loop control system, the control action, that is the processes/computations that the input undergoes to give the output are independent of the output. The block diagram shown above is an example of a simple open loop control system.

Closed Loop Control System : In a closed loop control system or a feedback control system, the output is continuously monitored in the form of a feedback signal. Thus, the control action of the controller depends on the output as well. A simple feedback controller can be represented using the following block diagram :

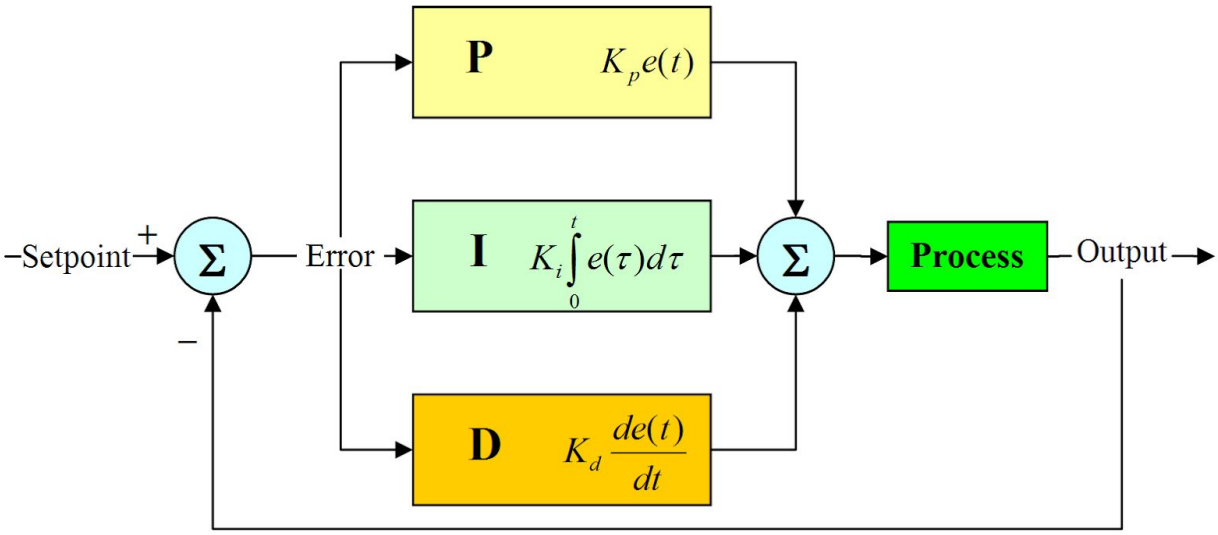


Where G is the control action, H is the feedback element, R is the reference input and C is the desired output. The output after undergoing some computation is added (positive feedback) or subtracted (negative feedback) from the input to provide an error signal which is then input into the controller G.

For a more detailed explanation with real life examples of control systems, look here.

2. Proportional, Integral, Derivative (PID) Controller

PID is the simplest and one of the most commonly used controllers in all kinds of controls applications (not just robotics).



Here, the P, I and D blocks represent the proportional, integral and the derivative controllers respectively. As you can see, the error signal is input into the three blocks and the final output signals are summed and then input into the *process* control action. Thus, the signal that is input into *process* $h(t)$ can be represented as

$$h(t) = K_p(t)e(t) + K_i\int_0^t e(\phi)d\phi + K_d de(t)/dt$$

Refer to this playlist by mathworks to get a more detailed insight into how the PID controller works. The first video gives a nice brief overview. Along with this, check out this video for a bunch of real life use cases of PID.

Hopefully you have gotten some intuition into the PID controller now. But it might be hard to see how you can implement this in real life. Consider a situation where you want your robot to turn to a specific angle. The only control you have over your robot is a turn signal (like a throttle). Below we have given a rough pseudocode as to how you could implement a simple PID control.

```
PID_Control (bot, goal, K_p, K_i, K_d) {
   1. Initialise error_prev = goal - bot.theta
   2. Initialise error_int = 0
   3. Loop:
      3.1 Set error = goal - bot.theta
      3.2 Update error_int = error_int + error
      3.3 Set error_dvt = error - error_prev
      3.4 Set u = K_p*error + K_i*error_int + K_d*error_dvt
      3.5 Execute bot.turn_control(u)
      3.6 Set error_prev = e
}
```

You can see that the error is just the difference between our desired state and the current state. And we don't really sit around calculating the derivative or integral, instead we use crude but effective approximations. The derivative is taken as the difference between the current and previous error and the integral is just the sum of all errors upto the current point. You could go one step further and use dt (a small constant) in calculating derivative and integral. Once you have an approximation for the three terms, you take their sum (weighted with their respective constants) as the signal that you want to provide to the robot.

The values `K_p, K_i, K_d` have to be set by the designer - you. Deciding these values usually involves some trial and error. This process is called tuning the controller and it depends heavily on your system. A recommended way to go about doing this is to first keep `K_i` and `K_d` zero (effectively a proportional controller) and find a near optimum value. Then you could introduce the other two components one by one.

For a detailed tutorial on the topic, you can also look at this document.

3. Laplace Transform

A Laplace Transform on a function $f(t)$ is defined by :

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

where $F(s)$ denotes the laplace transform of the function $f(t)$. The variable *t* (usually time) is a real variable whereas the variable *s* is a complex variable. Thus, laplace transforms convert functions from real variable domains to complex variable domains. For example applying laplace transform on a signal represented in the time(real) domain by $f(t) = t^2$ can be represented in term of it's frequency(complex) as $2/s^3$ where s is the complex frequency. Although Laplace Transformation is a very important part of control theory in general, you can understand most of the basic concepts without it. But it's essential to have some basic knowledge on this subject to tackle more advanced topics.

You can read more about laplace transforms here and about block diagram algebra here.

4. Further Resources

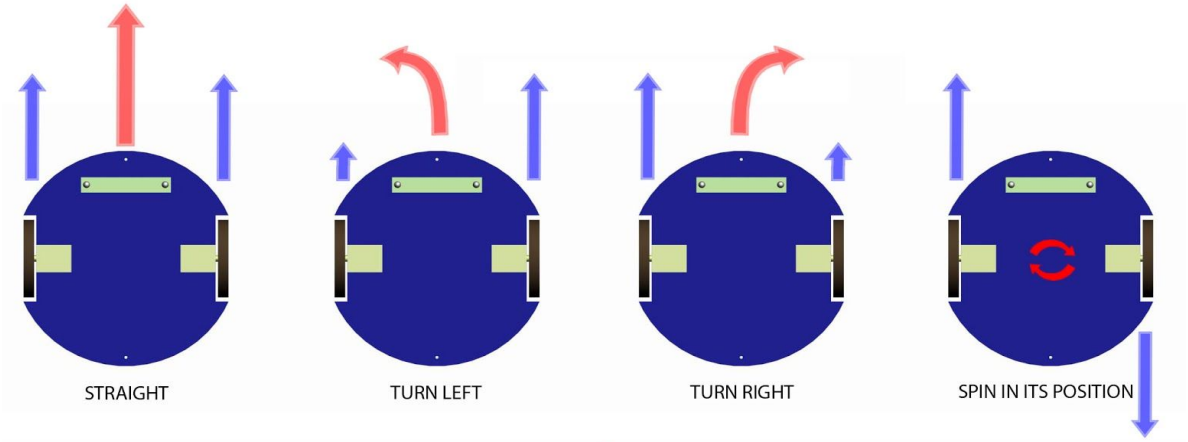
We have tried to give a brief overview of the core concepts in control theory in this document. We recommend you check out the following resources for more detailed explanation along with real life examples of how control theory can be used -

- This video series by Brian Douglas

- This video series by Steve Brunton
- This video series by Mathworks for PID controller
- This document gives a brief tutorial on PID controllers.

Differential Drive Robot

A differential drive robot or a differential wheeled robot is a mobile robot whose movement is based on two independently driven wheels placed on either side of the robot body. These robots can change their direction of motion by changing the difference between the velocities of independently controlled wheels. E.g. If the robot has to turn left, the velocity of the right wheel will be greater than the velocity of the left wheel. Contrary to this we have the Ackermann mechanism in our cars where we steer the wheels by an external control to change the direction of the car. We have already seen one such differential wheeled robot - the Turtlebot!



In the above figures the the length of the blue arrows represent the velocities of the wheels they represent and the red arrow depicts the direction of motion of the robot