Autonomous Vehicle Planning and Control

Wu Ning

Session 2

Vehicle Longitudinal Control

Outline

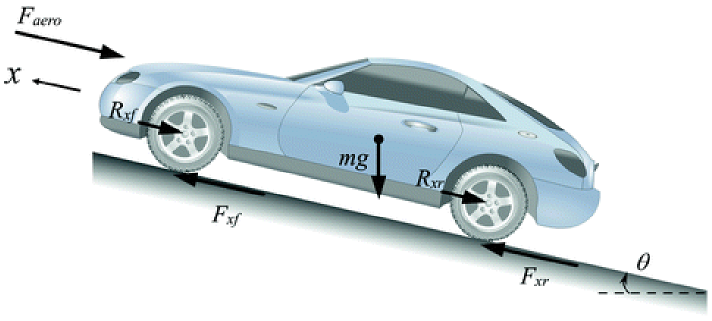
- Longitudinal vehicle dynamics
  - Aerodynamic drag force
  - Vehicle longitudinal tire force
  - Rolling resistance
- Classical feedback control
  - Transfer Function
  - Open loop VS closed loop
  - PID controller and it’s tuning
- Cruise control
- Adaptive cruise control

Longitudinal vehicle dynamics

Longitudinal force

A force balance along the vehicle longitudinal axis yields:

$$m\ddot{x} = F_{xf} + F_{xr} - F_{aero} - R_{xf} - R_{xr} - mg\sin(\theta)$$



where

- F_{xf} is the longitudinal tire force at the front tires,
- F_{xr} is the longitudinal tire force at the rear tires,
- F_{aero} is the equivalent longitudinal aerodynamic drag force,
- R_{xf} is the force due to rolling resistance at the front tires,
- R_{xr} is the force due to rolling resistance at the rear tires,
- m is the mass of the vehicle,
- g is the acceleration due to gravity,
- θ is the angle of inclination of the road on which the vehicle is traveling.

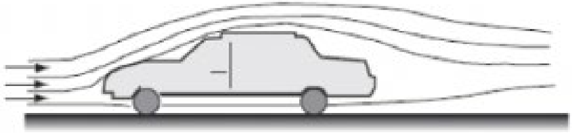
Aerodynamic drag force

Aerodynamic drag is the force needed to overcome these miniature forces when an object moves through air at a certain velocity.

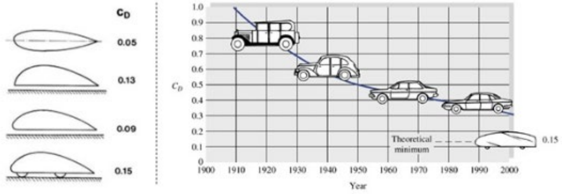
$$F_{aero} = \frac{1}{2}\rho C_d A_F (V_x + V_{wind})^2$$

where

- ρ is the mass density of air,
- C_d is the aerodynamic drag coefficient,
- A_F is the frontal area of the vehicle, which is the projected area of the vehicle in the direction of travel,
- V_x is the longitudinal vehicle velocity,
- V_{wind} is the wind velocity (positive for a headwind and negative for a tailwind).



v (km/h)	20	40	60	80	100	120	140	160
F_D (N)	10	50	109	180	300	420	565	771



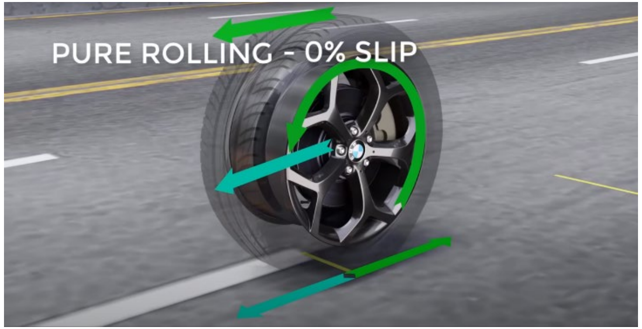
Drag coefficient of airfoil type vehicle (left), and change in drag coefficient through years proportional to the body of the vehicle (right)

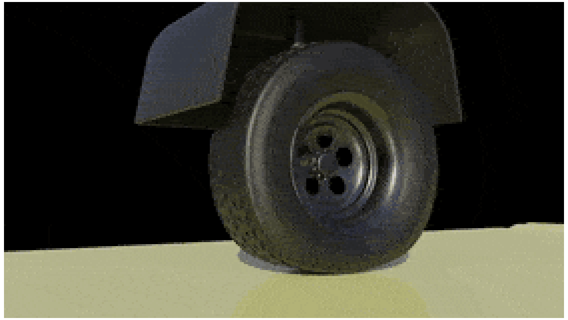
Longitudinal tire force

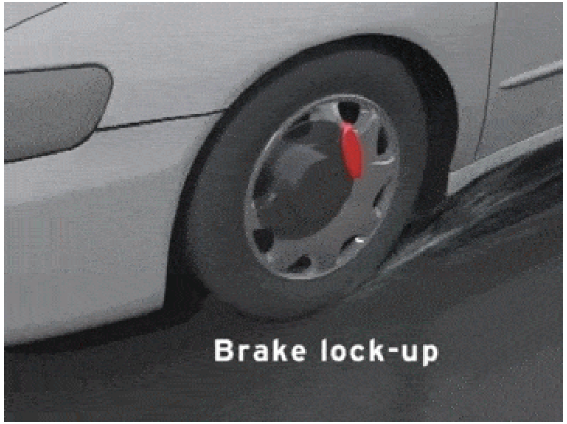
The longitudinal tire forces F_{xf} and F_{xr} are friction forces from the ground that act on the tires, which is generated by each tire depends on

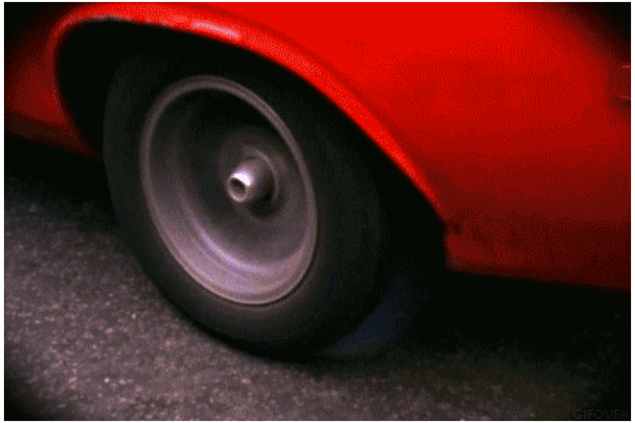
1. **the slip ratio.**
2. **the normal load** on the tire (vertical force F_{zf}, F_{zr})
   - comes from a portion of the weight of the vehicle
   - is influenced by the location of the c.g., vehicle longitudinal acceleration, aerodynamic drag forces and grade of the road
3. **the friction coefficient of the tire-road interface**.

Longitudinal slip ratio









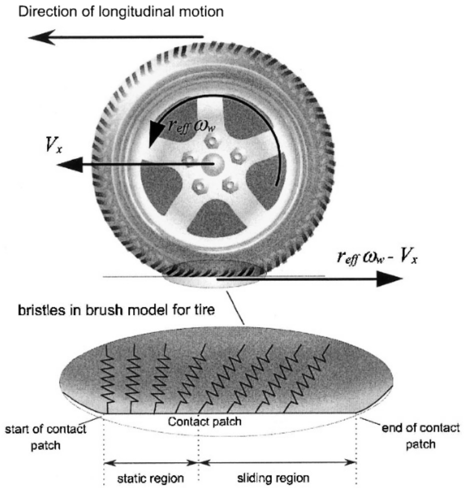
Deceleration

Acceleration

Longitudinal slip ratio

Slip ratio σ_x is defined as

$$\sigma_x = \begin{cases} \dfrac{r_{eff}\omega_w - V_x}{V_x} & \text{during braking} \\ \dfrac{r_{eff}\omega_w - V_x}{r_{eff}\omega_w} & \text{during acceleration} \end{cases}$$



$r_{eff}\omega_w < V_x$: Wheels are skidding, this happens during deceleration of vehicle, during normally braking.

$r_{eff}\omega_w > V_x$: Wheels are spinning, this happens in acceleration, especially in low friction driving (icy road)

$r_{eff}\omega_w = 0$: Wheels are locked, this happens during heavy or panic braking where the vehicle loses its desired traction.

Longitudinal tire force

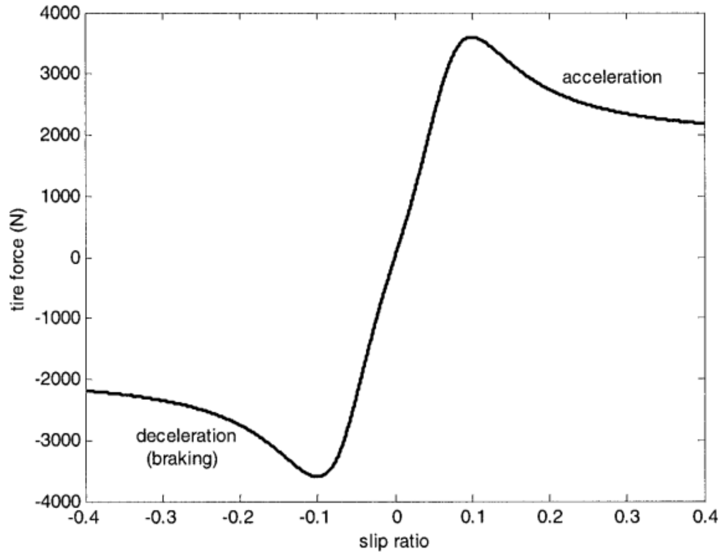
The tire force in this small-slip region can then be modeled as:

$$F_{xf} = C_{\sigma f}\sigma_{xf}$$

$$F_{xr} = C_{\sigma r}\sigma_{xr}$$

$$\sigma_x = \begin{cases} \dfrac{r_{eff}\omega_w - V_x}{V_x} & \text{during braking} \\[2ex] \dfrac{r_{eff}\omega_w - V_x}{r_{eff}\omega_w} & \text{during acceleration} \end{cases}$$

where $C_{\sigma f}$ and $C_{\sigma r}$ are called the longitudinal tire stiffness parameters of the front and rear tires respectively



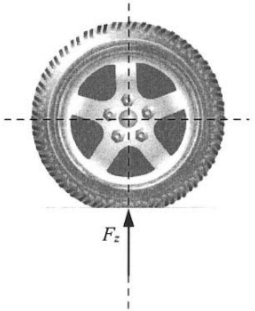
Longitudinal tire force as a function of slip ratio

Rolling resistance

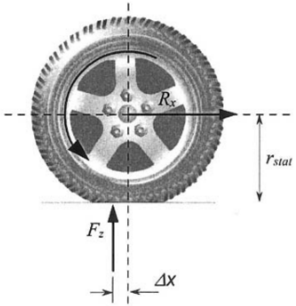
Typically, the rolling resistance is modeled as being roughly proportional to the normal force on each set of tires i.e.

$$R_{xf} + R_{xr} = f(F_{zf} + F_{zr})$$

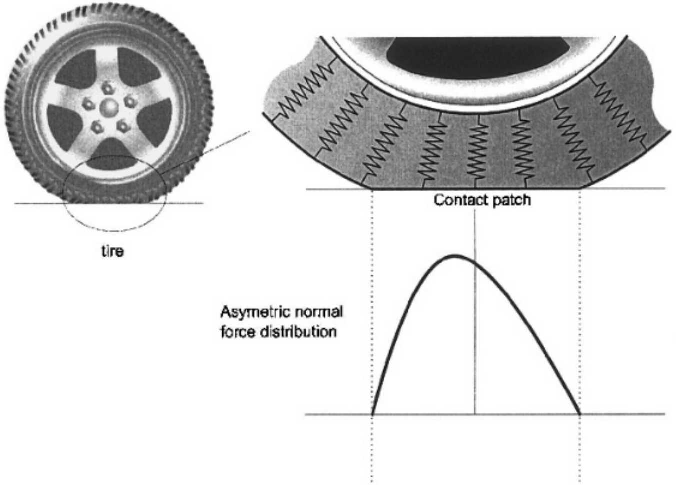
where f is the rolling resistance coefficient. A value of 0.015 is typical for passenger cars with radial tires (Wong, 2001). And F_{zf}, F_{zr} are the normal load (vertical force) on the tire.



Not rotating



Rotating



Rolling resistance

Typically, the rolling resistance is modeled as being roughly proportional to the normal force on each set of tires i.e. $R_{xf} + R_{xr} = f(F_{zf} + F_{zr})$

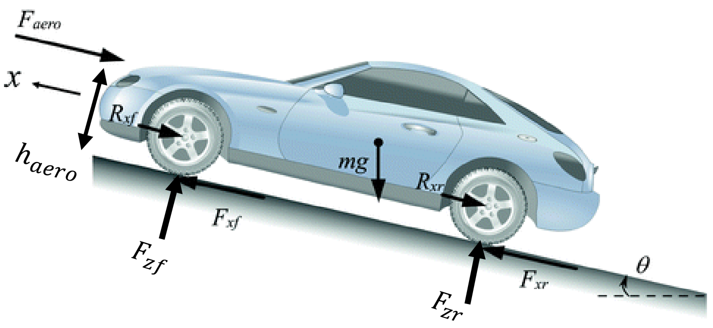
$$F_{zf}(\ell_f + \ell_r) + F_{aero}h_{aero} + mgh\sin(\theta) - mg\ell_r\cos(\theta) + m\ddot{x}h = 0$$

$$F_{zr}(\ell_f + \ell_r) - F_{aero}h_{aero} - mgh\sin(\theta) - mg\ell_f\cos(\theta) - m\ddot{x}h = 0$$

$$F_{zf} = \frac{-F_{aero}h_{aero} - m\ddot{x}h - mgh\sin(\theta) + mg\ell_r\cos(\theta)}{\ell_f + \ell_r}$$

$$F_{zr} = \frac{F_{aero}h_{aero} + m\ddot{x}h + mgh\sin(\theta) + mg\ell_f\cos(\theta)}{\ell_f + \ell_r}$$

- h is the hight of c.g
- h_{areo} is the hight of F_{aero}



Summary of Vehicle Longitudinal control

- Primary vehicle dynamic: $m\ddot{x} = F_{xf} + F_{xr} - F_{aero} - R_{xf} - R_{xr} - mg\sin(\theta)$

- Longitudinal tire force: $F_{xf} = C_{\sigma f}\sigma_{xf}$ $F_{xr} = C_{\sigma r}\sigma_{xr}$

$$\sigma_x = \begin{cases} \dfrac{r_{eff}\omega_w - V_x}{V_x} & \text{during braking} \\ \dfrac{r_{eff}\omega_w - V_x}{r_{eff}\omega_w} & \text{during acceleration} \end{cases}$$

- Rolling resistance: $R_{xf} + R_{xr} = f(F_{zf} + F_{zr})$

$$F_{zf} = \frac{-F_{aero}h_{aero} - m\ddot{x}h - mgh\sin(\theta) + mg\ell_r\cos(\theta)}{\ell_f + \ell_r}$$

$$F_{zr} = \frac{F_{aero}h_{aero} + m\ddot{x}h + mgh\sin(\theta) + mg\ell_f\cos(\theta)}{\ell_f + \ell_r}$$

- Aerodynamic drag force: $F_{aero} = \frac{1}{2}\rho C_d A_F (V_x + V_{wind})^2$

Vehicle Longitudinal control

"Longitudinal controller" is typically used in referring to any control system that controls the longitudinal motion of the vehicle.

Control state: longitudinal velocity, acceleration or its longitudinal distance

Control input: (the actuators used to implement longitudinal control): throttle and brakes

Actual implementation:

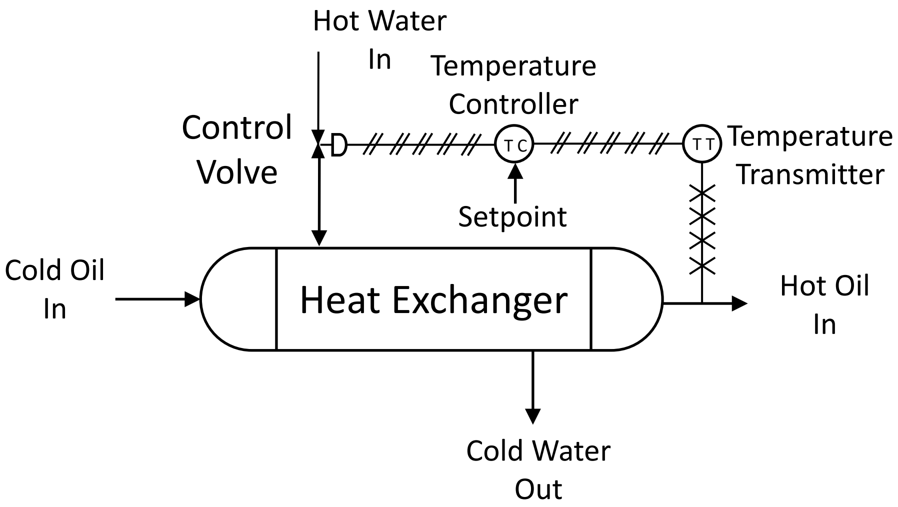
- Adaptive cruise control
- Collision avoidance
- Automated highway systems

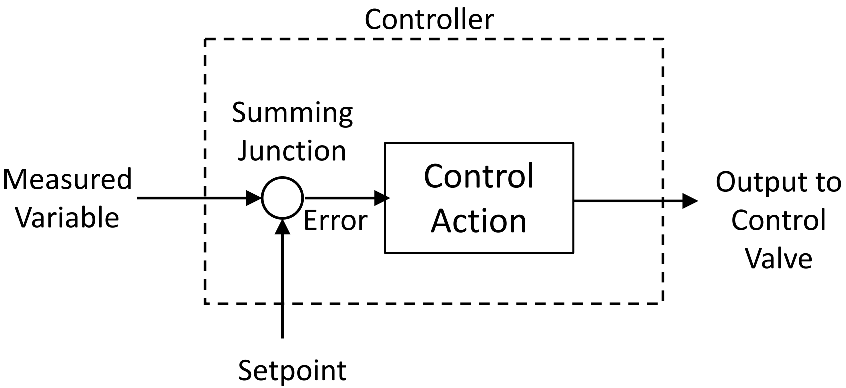
Classical feedback control

Background

What is a controller?

- A controller is a device that generates an output signal based on the input signal it receives.
- The input signal is often an error signal, which is the diff between the measured variable and the desired value (set point).





What is Classical Control and why?

Classical Control

- A Physical system can be modeled in the "time domain", where the response of a given system is a function of the various inputs, the previous system values, and time. As time progresses, the state of the system and its response change. However, time-domain models for systems are frequently modeled using high-order differential equations which can become impossibly difficult for humans to solve and some of which can even become impossible for modern computer systems to solve efficiently.

Laplace Transform

- To counteract this problem, classical control theory uses the Laplace transform to change an Ordinary Differential Equation (ODE) in the time domain into a regular algebraic polynomial in the frequency domain. Once a given system has been converted into the frequency domain it can be manipulated with greater ease.

What is Laplace Transform?

- Definition of *Laplace* Transform:

$$F(s) = \int_0^\infty f(t)\, e^{-st} dt,\ s = \sigma + j\omega$$

- Time-domain differentiation: if there is $L\{f(t)\} = F\ (s)$

$$L\{f'(t)\} = sF\ (s) - f(0)$$

$$L\{f''(t)\} = s^2 F(s) - sf(0) - f'(0)$$

$$\therefore L\{f^n(t)\} = s^n F(s) - \sum_{k-1}^{n} s^{k-1} f^{n-k}(0)$$

- Time-domain Integration: if there is $L\{f(t)\} = F\ (s)$

$$L\{\int_0^t f(\tau) d\tau\} = \frac{1}{s} F\ (s)$$

Common Laplace Transform

Function	Time domain	S domain
Unit impulse	$\delta(t)$	1
Delay impulse	$\delta(t-\tau)$	$e^{-\tau t}$
Unit step	$u(t)$	$\frac{1}{s}$
Delay step	$u(t-\tau)$	$\frac{1}{s}e^{-\tau t}$
Unit ramp	t	$\frac{1}{s^2}$
Exponential	e^{-at}	$\frac{1}{s+a}$
Time multiplied Exponential	$t^n e^{-at}$	$\frac{n!}{(s+a)^n}$

What is Transfer function?

The transfer function of a linear, time-invariant, differential equation system is defined as the ratio of the Laplace transform of the output (response function) to the Laplace transform of the input (driving function) under the assumption that all initial conditions are zero.

Consider the linear time-invariant system defined by the following differential equation:

$$a_0\frac{d^ny}{dt}+a_1\frac{d^{n-1}y}{dt}+\cdots+a_{n-1}\frac{dy}{dt}+a_ny=b_0\frac{d^mx}{dt}+b_1\frac{d^{m-1}x}{dt}+\cdots+b_{m-1}\frac{dx}{dt}+b_mx \qquad (n\geq m)$$

where y is the output of the system and x is the input. The transfer function of this system is the ratio of the Laplace transformed output to the Laplace transformed input when all initial conditions are zero

$$G(s)=\frac{\mathscr{L}[output]}{\mathscr{L}\ [input]}\Big|_{zero\ initial\ conditions}$$

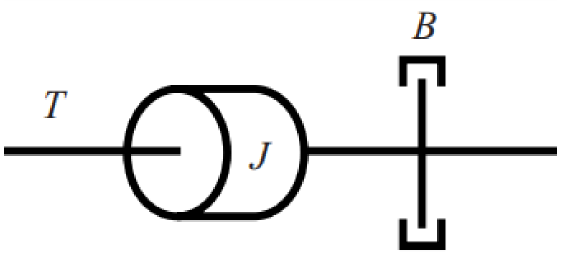
$$\frac{Y(s)}{X(s)}=\frac{b_0s^m+b_1s^{m-1}+\cdots+b_{m-1}s+b_m}{a_0s^n+a_1s^{n-1}+\cdots+a_{n-1}s+a_n}$$

Open loop system

The equation for the load elements is

$$J\ddot{c} + B\dot{c} = T$$

where T is the torque, c is the rotation angle.



By taking Laplace transforms of both sides of this last equation, assuming the zero initial conditions, we obtain

$$Js^2C(s) + BsC(s) = T(s)$$

Therefore, the open loop transfer function between $C(s)$ and $T(s)$ is:

$$\frac{C(s)}{T(s)} = \frac{1}{s(Js+B)}$$

Closed loop system

Open loop transfer function

$$G_{op}(s) = \frac{C(s)}{T(s)} = \frac{1}{s(Js+B)}$$

Closed loop transfer function between $C(s)$ and reference $R(s)$ is:

$$G_{cl}(s) = \frac{C(s)}{R(s)}$$

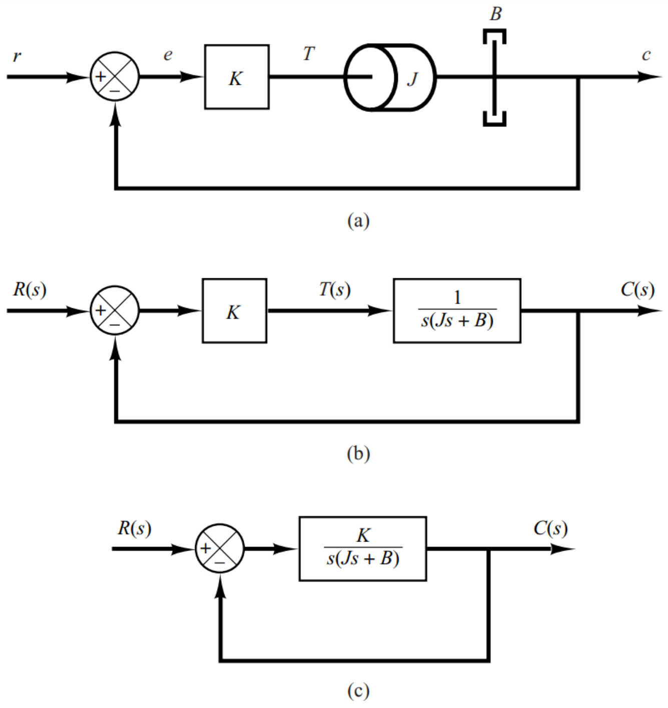
$$C(s) = \big(R(s) - C(s)\big)K\frac{1}{s(J_s + B)}$$

$$\frac{C(s)}{R(s)} = \frac{\frac{K}{s(J_s + B)}}{1 + \frac{K}{s(J_s + B)}} = \frac{\mathrm{K}}{Js^2 + Bs + \mathrm{K}}$$

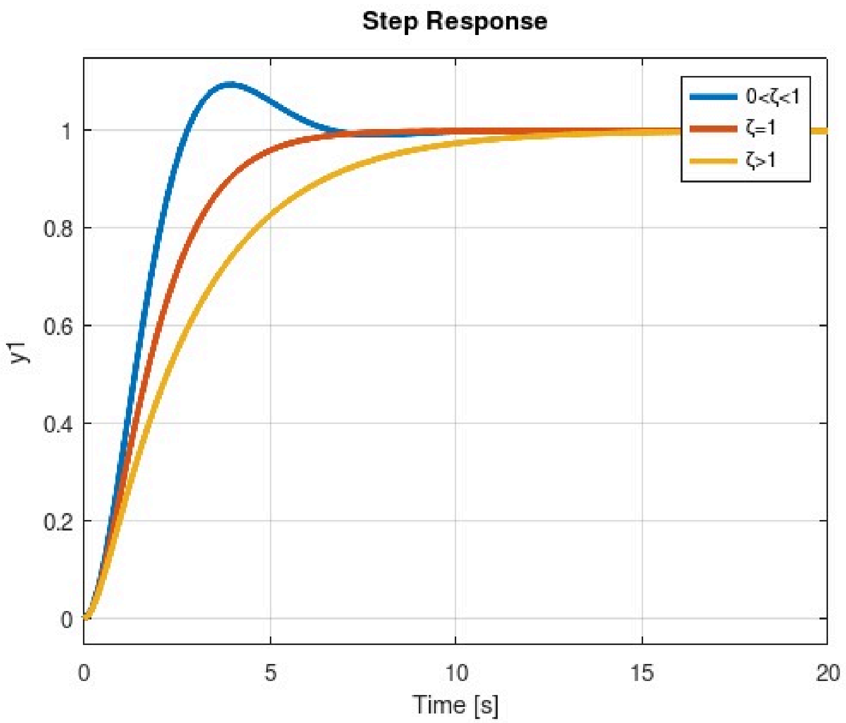
Rewrite it into the following form:

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

where $\zeta = \frac{B}{2\sqrt{JK}}$ is called damping ratio, $\omega_n = \sqrt{\frac{K}{J}}$ is natural frequency



Step Response of a Second Order System

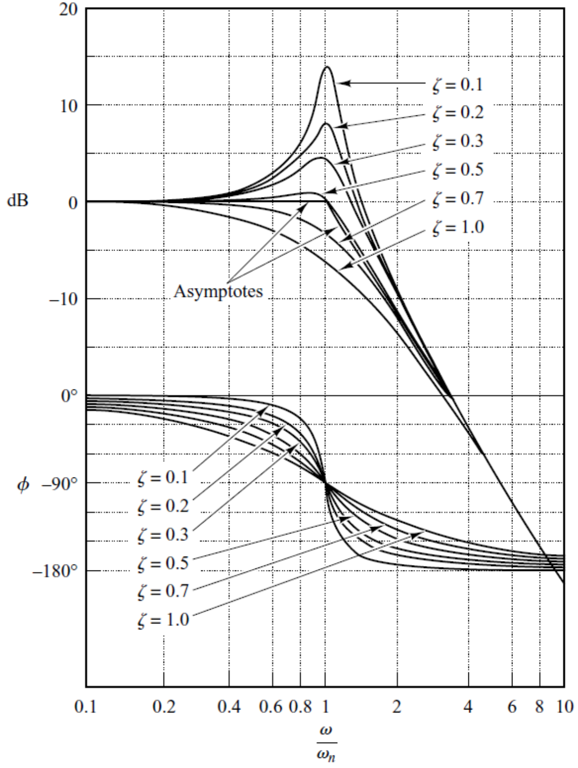


$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

- Underdamped case ($0 < \zeta < 1$)
  Faster reaches steady state, however will oscillate
- Overdamped case ($\zeta > 1$)
  Takes a long time to reach steady state, no overshoot
- Critically damped case ($\zeta = 1$)
  Reaches steady state with fairly short time without oscillation

Bode plot: Graphical forms representation of system frequency response characteristics

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$



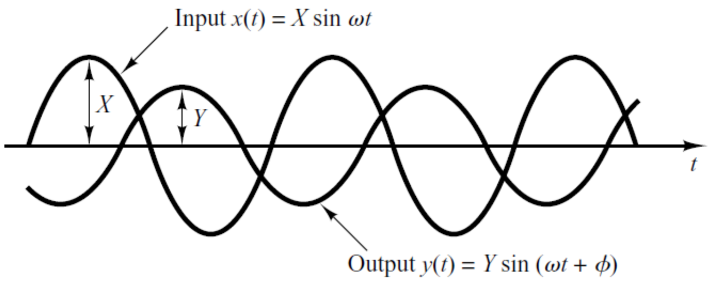
It consists of two graphs:

Logarithm of the magnitude of a sinusoidal transfer function:

$$20\log_{10}|G(j\omega)|$$

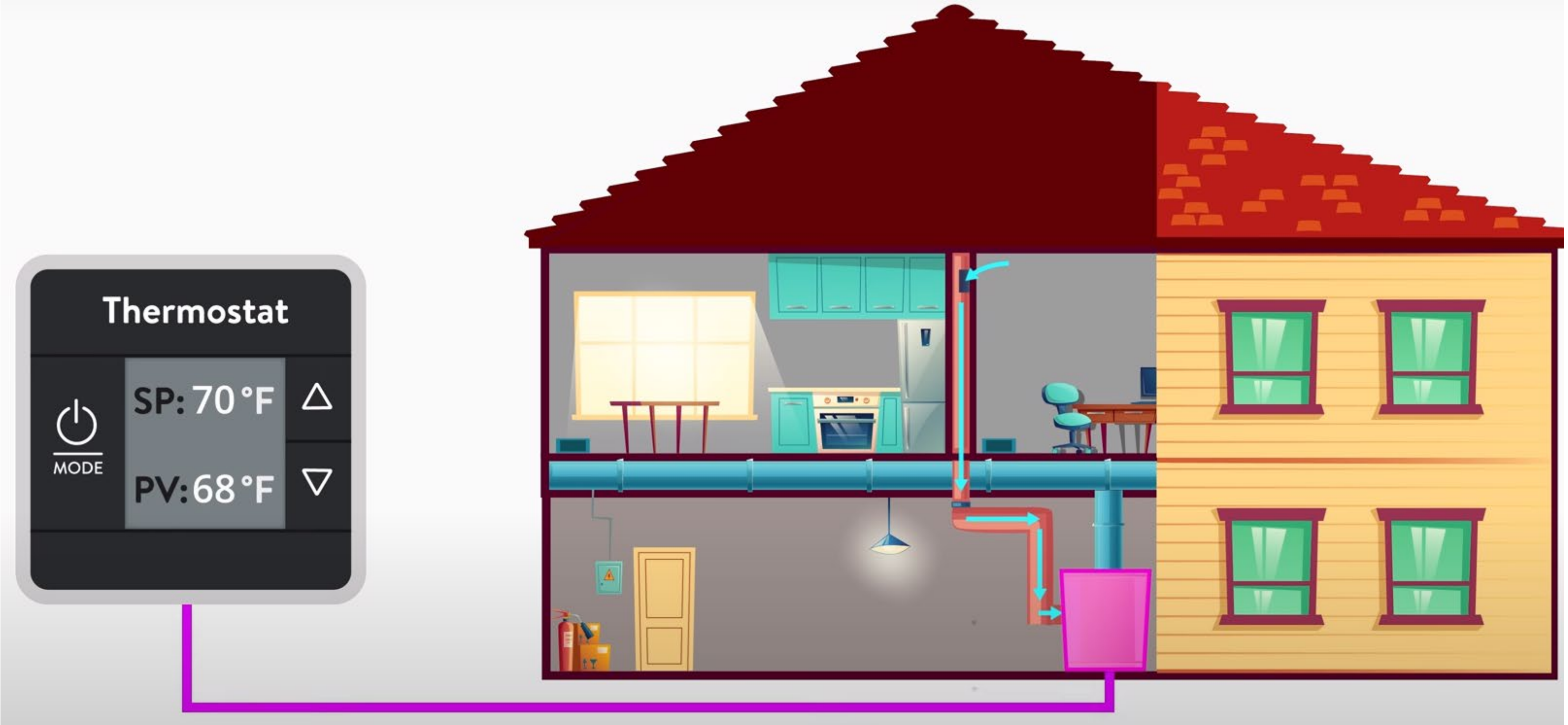
Phase angle:

$$\measuredangle G(j\omega) = \phi = \tan^{-1}\frac{imaginart\ part\ of\ G(j\omega)|}{real\ part\ of\ G(j\omega)|}$$



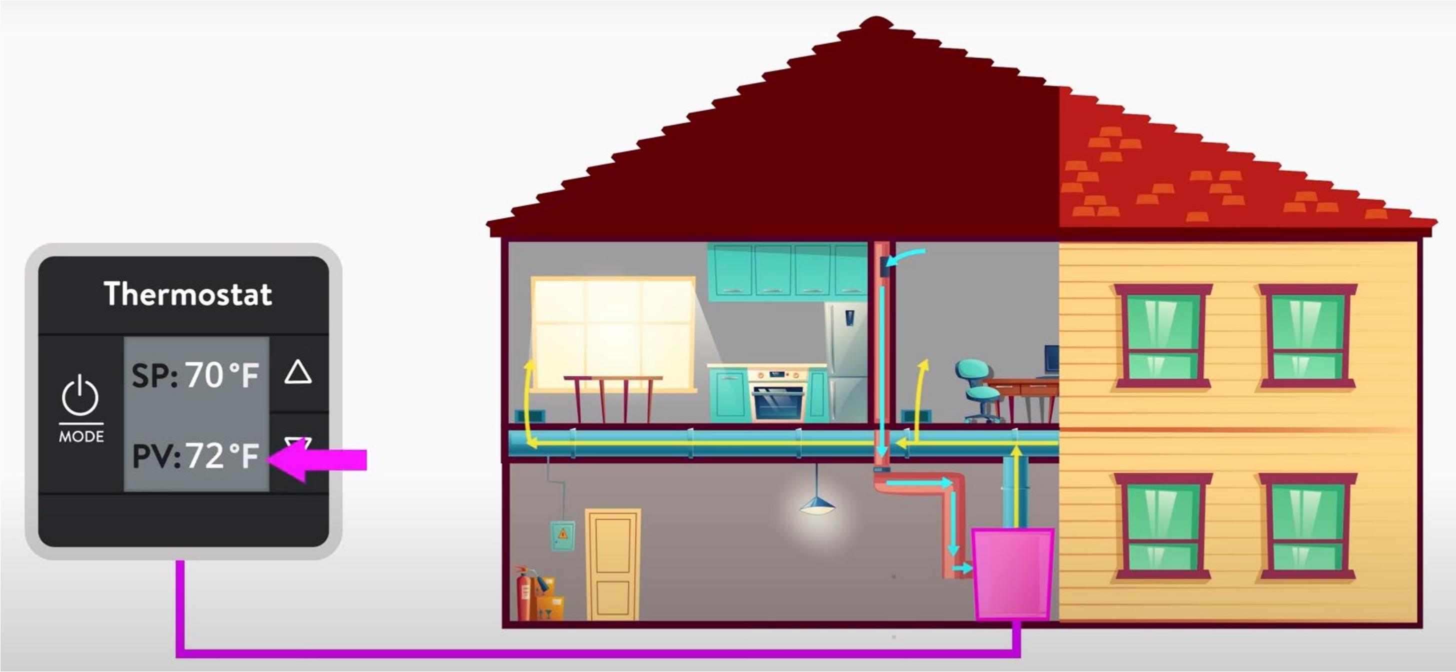
PID Controller

Why PID Control?



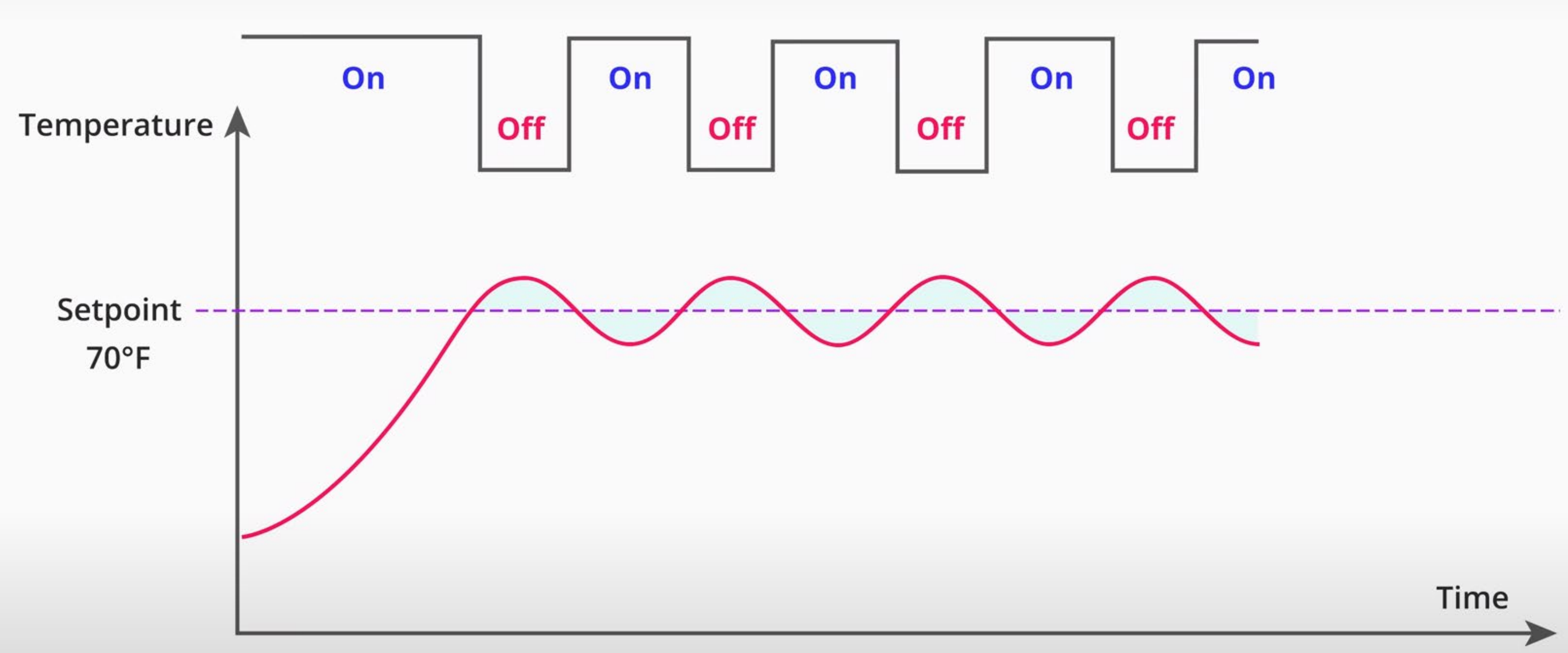
Below set point-> on

Why PID Control?



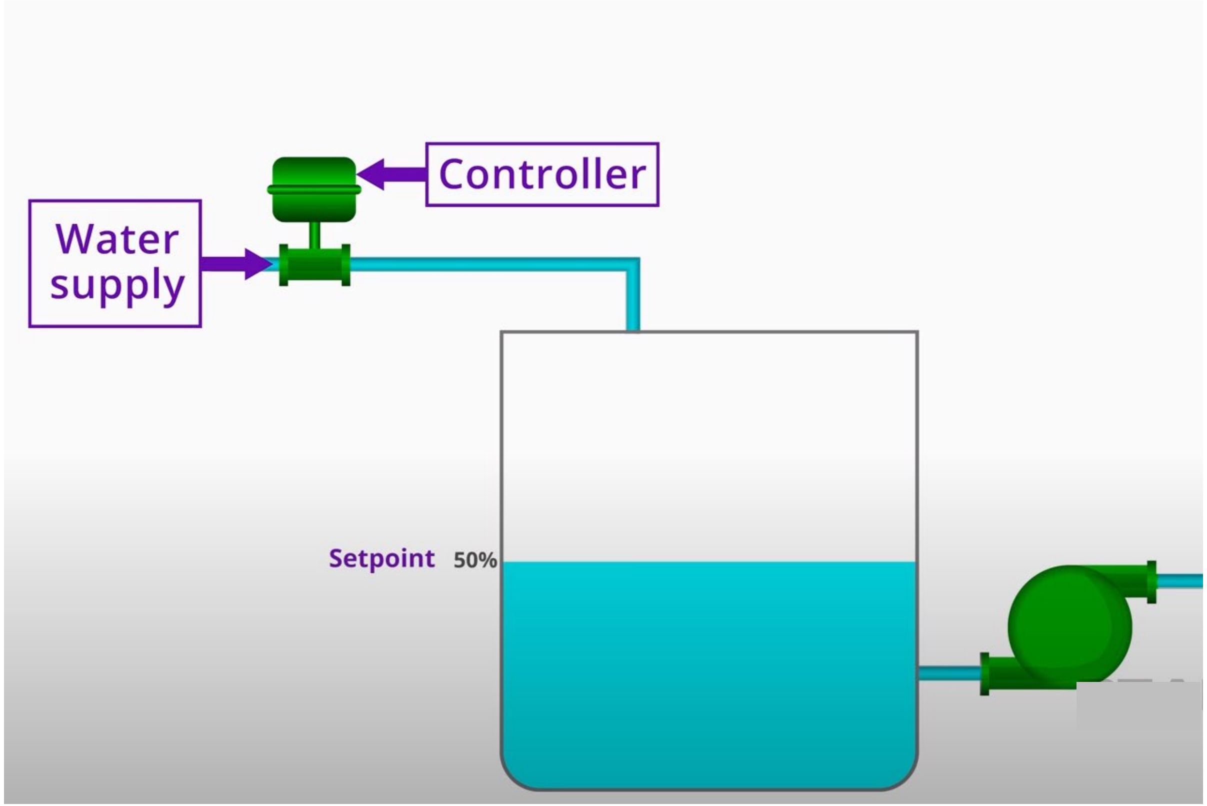
Above set point-> off

Why PID Control?

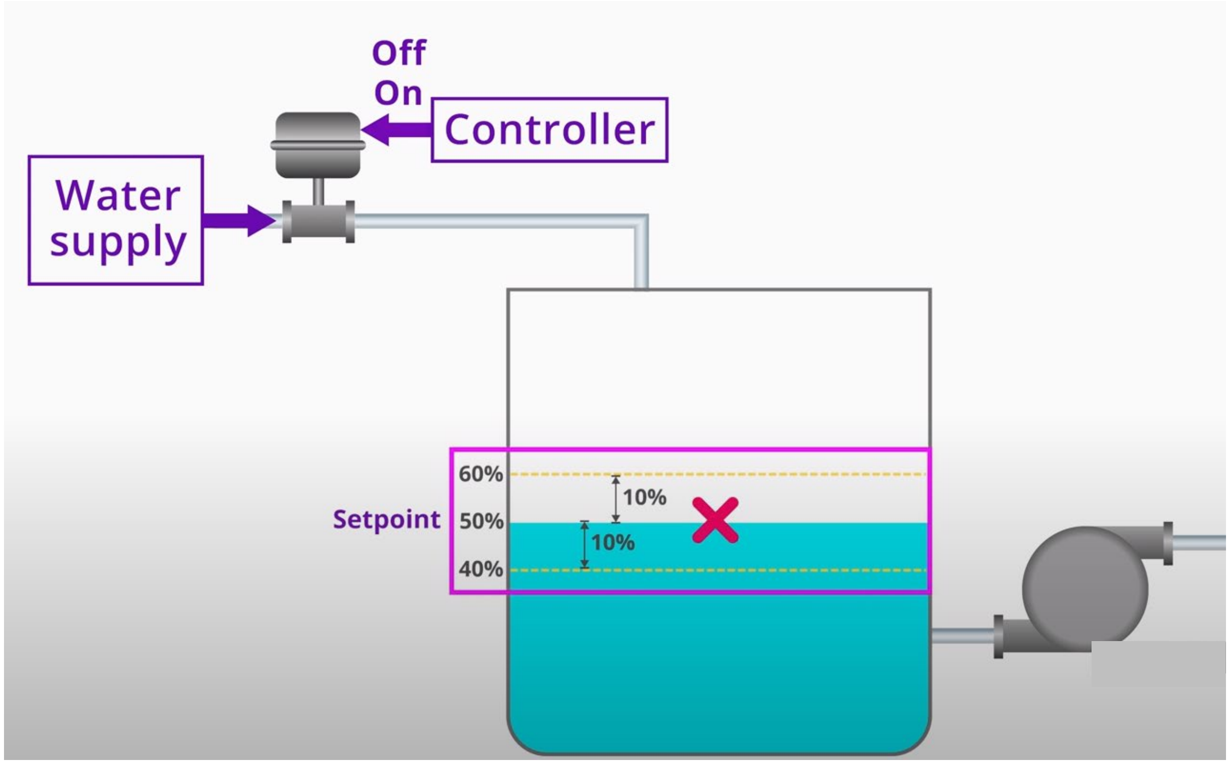


On/Off controller
Bang Bang controller

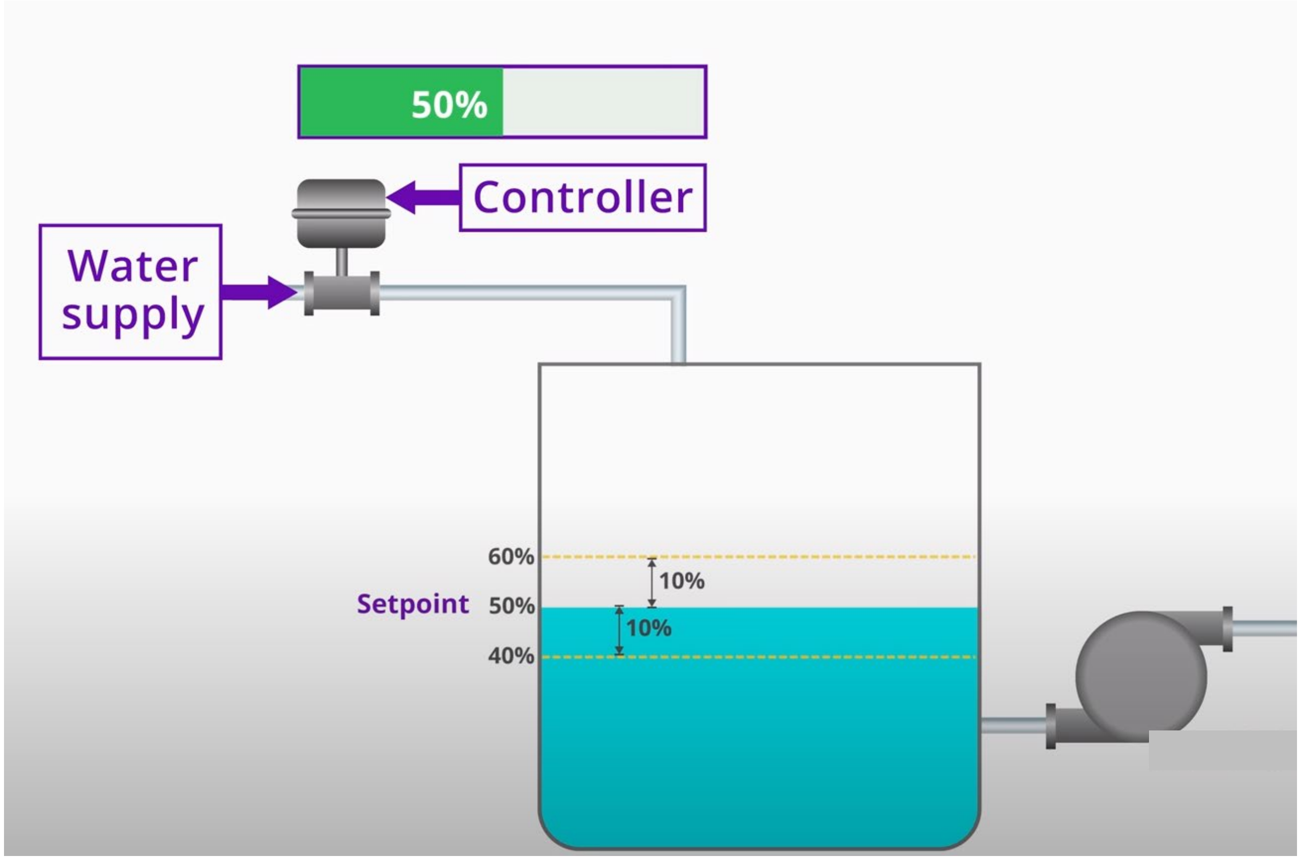
Why PID Control?



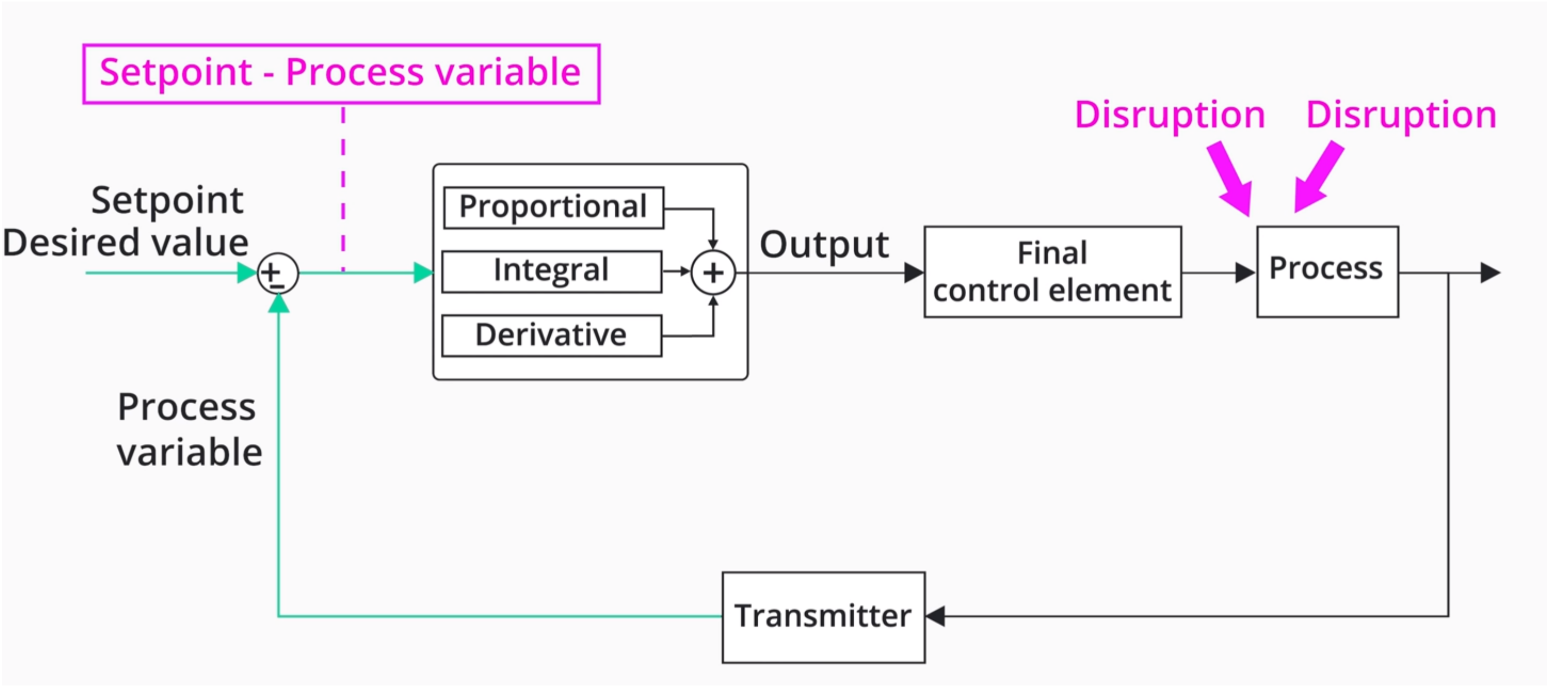
Why PID Control?



Why PID Control?



PID Control



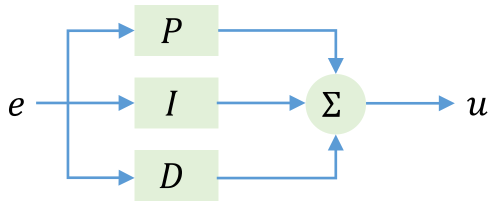
Setpoint - Process variable
Disruption
Disruption
Setpoint
Desired value
±
Proportional
Integral
Derivative
+
Output
Final
control element
Process
Process
variable
Transmitter

PID Control

- "Textbook" version of PID algorithm in transfer function representation:

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau)\, d\tau + K_D \frac{de(t)}{dt}$$

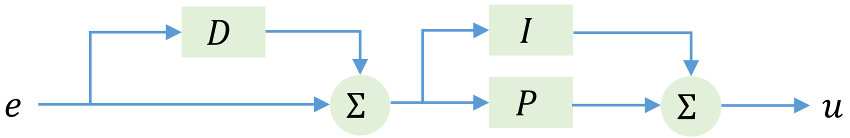
$$G(s) = \frac{\mathrm{U}(s)}{E(s)} = K_P + \frac{K_I}{s} + sK_D$$



(a) Non-interacting form

- It is also known as non-interacting (parallel) form PID controller.
- A slightly different version (series or interacting form) commonly found in commercial controllers is given by

$$G'(s) = \left(K_P' + \frac{K_I'}{s}\right)(1 + sK_D')$$

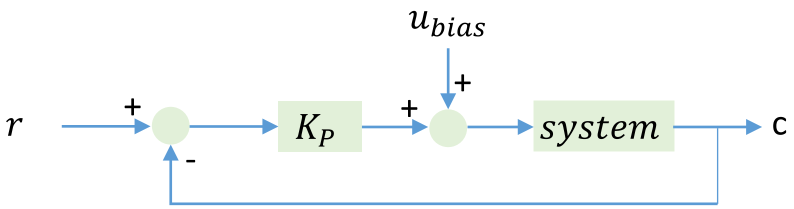


(b) Interacting form

Proportional Action

- For the pure proportional control, the control law:

$$u = K_P(r - c) + u_{bias}$$



- There will be always a steady state error in proportional control. The error will decrease with increasing gain, but the tendency towards oscillation will also increase. Let's design a P controller for the following system:

$$G(s) = \frac{1}{s^2 + s + 1} \qquad H(s) = K_P$$

- The closed loop transfer function will be:

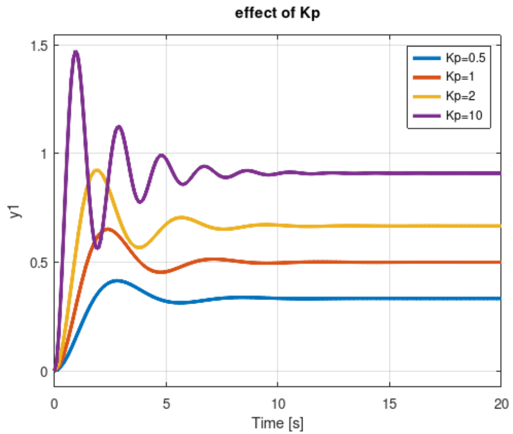
$$G_c(s) = \frac{G(s)H(s)}{1 + G(s)H(s)} = \frac{K_P}{s^2 + s + 1 + K_P}$$

- Steady state:

$$G_c(0) = \frac{K_P}{1 + K_P}$$

- Damping ratio:

$$\zeta = \frac{1}{2\sqrt{1 + K_p}}$$



Integral Action: PI controller

- The proportional part creates immediate output change to reposition the final control element within relatively short period of time in response to the error and the integral part continuously reposition final control element until the error is reduced to 0.
- The main disadvantage of the integral mode is that the controller output does not **immediately** direct the final control element to a new position in response to an error signal.
- The controller output changes at a defined rate of change, and the time is needed for the final control element to be repositioned.

Effect of K_I

Let's design a PI controller for the following system

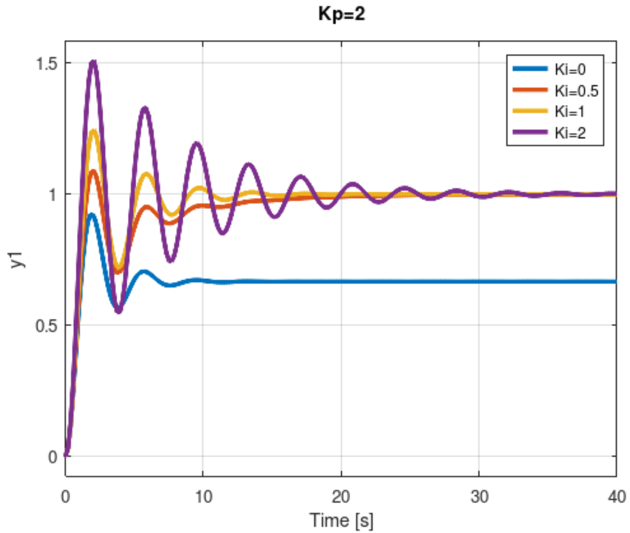
$$G(s) = \frac{1}{s^2 + s + 1} \qquad H(s) = K_P + \frac{K_I}{s}$$

$$G_c(s) = \frac{G(s)H(s)}{1 + G(s)H(s)} = \frac{K_P s + K_I}{s^3 + s^2 + (1 + K_P)s + K_I}$$

- Steady state: $G_c(0) = 1$
- Given K_P is fixed, we can see that the higher the K_I, the faster the output converges to the set point.

(can be seen from red and orange line in the right plot)

- However, it creates more oscillations into the system.



Derivative Action: PD controller

- The higher the error signal rate of change, the sooner the final control element is positioned to the desired value.
- The added derivative action reduces initial overshoot of the measured variable, and therefore aids in stabilizing the process sooner.
- This control mode is called proportional plus rate (PD) control because the derivative section responds to the rate of change of the error signal.

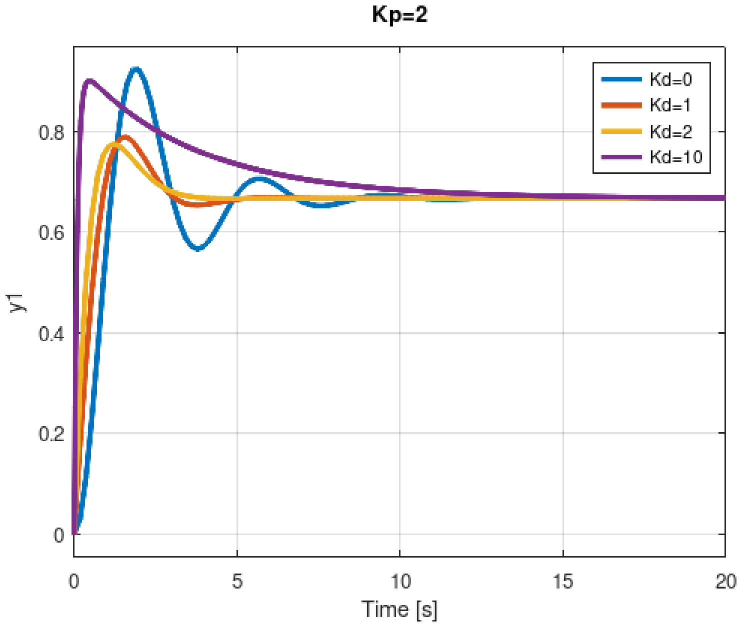
Effect of K_D

Let's design a PD controller for the following system

$$G(s) = \frac{1}{s^2 + s + 1} \qquad H(s) = K_P + K_D s$$

$$G_c(s) = \frac{G(s)H(s)}{1 + G(s)H(s)} = \frac{K_D s + K_P}{s^2 + (1 + K_D)s + (1 + K_P)}$$

- Steady state: $G_c(0) = \frac{K_P}{1+K_P}$
- Given K_P is fixed, we can see that the higher the K_D, the less the oscillations .
- However, it will result in slower convergence to the set point.



Modifications of the PID algorithm

Derivative approximation

Consider a PD controller in in the following format, where $K_i = 0$:

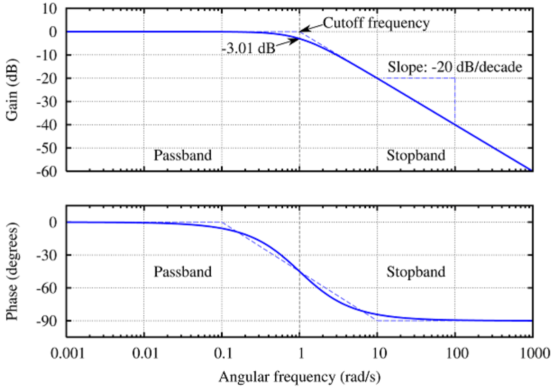
$$G(s) = K_P + \frac{K_I}{s} + \boxed{sK_D}$$

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau)\, d\tau + \boxed{K_D \frac{de(t)}{dt}}$$

$\frac{de(t)}{dt}$ requires differentiation of a the error signal which in many cases can be very noisy. Such noise will result in large control signal especially when noise is in high frequency. Therefore, it is essential to limit the gain in high frequency region. This can be done by implementing the derivative term as:

$$D = s\mathrm{K}_D \frac{1}{\frac{K_D}{N}s + 1}$$

This can be seen as a the ideal derivative sK_d filtered by a first-order system with the time constant K_D/N. The approximation acts as a derivative for low-frequency signal components.



Modifications of the PID algorithm

Integrator Windup

- All actuators have limitations: a motor has limited speed, a valve cannot be more than fully opened or fully closed, etc. For a control system with wide range of operating conditions, it may happen that control variable reaches the actuator limits.
- When this happens the feedback path will be broken, because the actuator will remain saturated even if the process output changes.
- If the controller with integrating action is used, the error will continue to be integrated. This means that the integral term may become very large or is "winds up": integrator windup.
- It is then required that error has opposite sign for a long period before things return to normal. The consequence is that any controller with integral action may give large transients when the actuator saturates.

Modifications of the PID algorithm

Integrator Windup: how to avoid?

- **Setpoint Limitation**

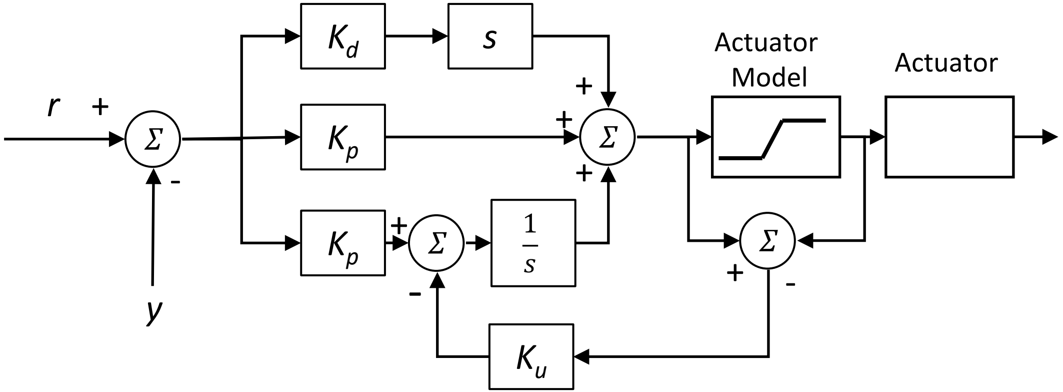
  Introduce limiters on the setpoint variables so that the controller output will never reach the actuator bounds. Does not work for windup causes by disturbances, another problem is limitations on controller performance.

- **Clamping**

  Limit the upper and lower bound of integration

- **Back-calculation**

  Windup is avoided by inhibiting the integration whenever the output saturates. This method is equivalent to back-calculation. It is also easy to limit the rate of change of the control signal



Common methods of PID tuning

Example: Ziegle-Nichols(Z-N) Oscillation Method

Zero K_I and K_D, set K_P low.

↓

Set command to zero.

↓

Raise K_P to K_{MAX}, the minimum value that causes sustained oscillation.

Note f_0, the frequency of oscillation.

↓

Set control law.

P control	PI control	PID control
$K_P = 0.5K_{MAX}$	$K_P = 0.45K_{MAX}$	$K_P = 0.6K_{MAX}$
$K_I = 0$	$K_I = 2f_0$	$K_I = 2f_0$
$K_D = 0$	$K_D = 0$	$K_D = 0.125/f_0$

↓

Done

Other PID Tuning Techniques

	Pros	Cons
heuristic tuning	• It's a quick and easy way to obtain a reasonable result. • It has an intuitive approach, i.e. no (explicit) assumptions are made about the process. • Little process knowledge is required.	• It's time-consuming. It takes a long time to achieve good performance. • It doesn't guarantee reaching a robust and stable solution. This can represent a risk for the entire plant.
Ziegler-Nichols tuning method	• It's simple, intuitive and you get a reasonable performance for simple loops. • Little process knowledge is required.	• Results in an oscillatory closed loop response (max overshoot at 25%). • Only suitable for small dead time processes (dead time is smaller than the process time constant) • High proportional gains (due to the 25% overshoot design specification), low integral action with too low damping of the closed loop system and too low robustness against changes in the process dynamics, including non-linearities. • It can't define control objectives or closed loop performance requirements.
Cohen-Coon tuning method	• Same advantages as Ziegler-Nichols. • Additionally, works well specifically for systems with a larger time delay.	• Same disadvantages as Ziegler-Nichols

Other PID Tuning Techniques

	Pros	Cons
Kappa-Tau tuning method	• Less oscillatory response. • Originally designed for load disturbance response. It can also deal with setpoint tracking by using setpoint weighting. • Results in optimal disturbance rejection with no overshoot. • The tuning parameter for the design is the sensitivity of the controller towards process disturbances. Allows choosing between faster or slower response.	• It can't define control objectives or closed loop performance requirements.
Lambda tuning method	• Enables choosing a desired closed-loop time constant, i.e. how fast the controller responds. • Works well specifically for systems with a large time delay (dead time is close to the process time constant). • Results in high robustness against changes in the process dynamics, including non-linearities. • Results in a response with no overshoot.	• Results in a slow rejection of disturbances, especially for slow systems. • It can't define control objectives and is limited in closed loop performance requirements. • Only suited for PI controller tuning. The parameter derivative cannot be taken into consideration.
Model-based tuning	• Allows a structured tuning method that considers both your process behavior and your control needs. • Enables a balance between the engineering objectives' performance and robustness. • It's a flexible method. It will search for the optimal solution close to your requirements. You can compare and test scenarios.	• You need to follow a strict workflow. The method requires making explicit how you want the process to behave in closed loop. • Requires you to identify a sufficiently accurate model, otherwise you will never get the right loop tuning.

Cruise Control

Cruise Control

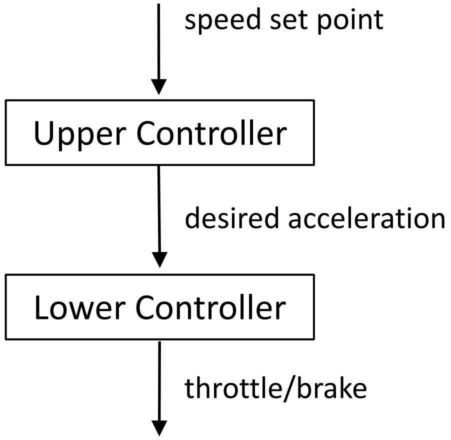
In a standard cruise control system, the speed of the vehicle is controlled to a desired value using the throttle control input.

Input:

- Speed set point (target speed)

Output:

- Desired acceleration=> Throttle/brake



Upper level PI controller

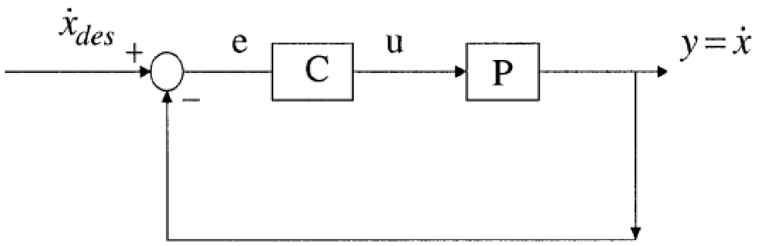
When designing upper level, we will consider the imperfection of the lower level controller due to finite bandwidth. Therefore, we will assume a 1st order system to model the response in lower level controller:

$$\ddot{X} = \frac{1}{\tau s + 1}\ddot{X}_{des}$$

The plant model for the upper controller is the transfer function between desired acceleration and actual vehicle speed

Since $\ddot{x} = \frac{1}{\tau s+1}\ddot{x}_{des}$, or $sV(s) = \frac{1}{\tau s+1}A_{des}(s)$

$$\mathrm{P(s)} = \frac{V_x(s)}{A_{des}(s)} = \frac{1}{s(\tau s + 1)}$$



Upper level PI controller

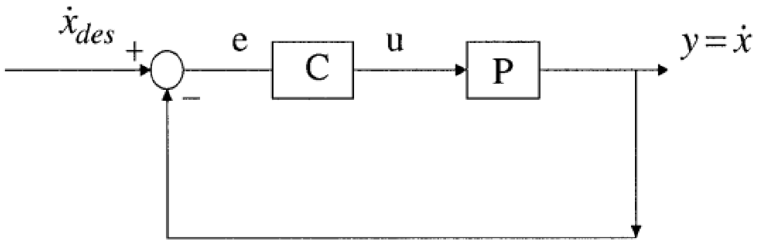
A typical algorithm used for the upper controller is PI control using error in speed as the feedback signal:

$$u(t) = -k_p(\dot{x} - \dot{x}_{des}) - k_I(x - x_{des})$$

$$u(t) = -k_p\left(V_x - V_{ref}\right) - k_I \int_0^t \left(V_x - V_{ref}\right)dt$$

The PI controller is $C(s) = k_p + \frac{k_i}{s}$, $\mathrm{P(s)} = \frac{1}{s(\tau s+1)}$

the closed-loop transfer function is



$$\frac{V_x}{V_{des}} = \frac{PC}{1+PC} = \frac{k_p s + k_i}{\tau s^3 + s^2 + k_p s + k_i}$$

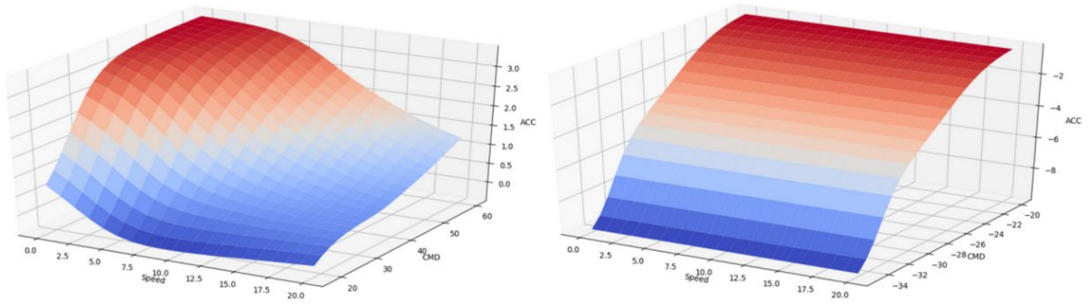
Lower level controller

In the lower controller, the throttle input is calculated so as to track the desired acceleration determined by the upper controller.

The key is finding the relationship between acceleration and throttle/brake.

Approaches:

1. Vehicle model
2. Look up table



(a) Throttle (b) Brake

Lower level controller

Vehicle model:

- Assume the vehicle speed directly to the engine speed $\dot{x} = r_{eff}\omega_w$; $\ddot{x} = r_{eff}R\dot{\omega}_e$

where ω_e is the angular speed of the engine output and *R* is the gear ratio.

- The vehicle longitudinal dynamic model: $m\ddot{x} = F_x - R_x - F_{aero}$;

$$F_x = mr_{eff}R\dot{\omega}_e + R_x + F_{aero}$$

- From wheel rotational dynamics, transmission dynamics, and engine rotational dynamics, and assumption of F_{aero} as $F_{aero} = c_a\left(r_{eff}R\omega_e\right)^2$ (since F_{aero} is quadratic function of vehicle velocity and can also be expressed in terms of a quadratic in ω_e), we can get the relationship of torque and $\ddot{x}_{ides}$.

Lower level controller

Vehicle model:

- The net torque:

$$T_{net} = \frac{J_e}{R\mathrm{r}_{\mathrm{eff}}}\ddot{x}_{ides} + \left[c_a R^3 {r_{eff}}^3 \omega_e^2 + R(r_{eff} R_x)\right]$$

- The dynamics relating engine acceleration $\dot{\omega}_{\mathrm{e}}$ to the pseudo-inputs "net combustion torque" T_{net} can be modelled by the single first-order ode (ordinary differential equation)

where $J_e = I_e + \left(m{r_{eff}}^2 + I_\omega\right)R^2$ is the effective inertia reflected on the engine side, R is the gear ratio and r_{eff} the tire radius.

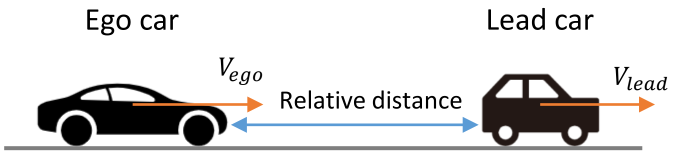
Adaptive Cruise Control

What is ACC?

Adaptive cruise control (**ACC**) is an available cruise control advanced driver-assistance system for road vehicles that automatically adjusts the vehicle speed to maintain a safe distance from vehicles ahead by using certain sensors, such as Radar.

Input:

- Driver-set velocity (V_{set}: similar as cruise control's target speed)
- Velocity of the ego car V_{ego}
- Relative distance to the lead car (preceding vehicle in the same lane)
- Relative velocity of the lead car
- Time gap



Output: Acceleration of ego car =>Throttle/Brake (same as Cruise control)

ACC

ACC system have two modes of steady state operation:

- **Speed control**: The ego car travels at a driver-set speed.
- **Spacing control** (i.e. Vehicle following): The ego car maintains a safe distance from the lead car.

Basic Principle: *Switch* between

- *Too close*: Spacing control, The control goal is to maintain the safe distance
- *Further away*: Speed control, The control goal is to track the driver-set velocity

Why time gap not constant distance?

Steady state spacing control is called vehicle following. In the vehicle following mode, the longitudinal controller must ensure that the following two properties are satisfied:

1. **Individual vehicle stability**, in which spacing error converges to zero if the preceding vehicle travels at constant velocity
2. **String stability,** in which spacing error does not amplify as it propagates towards the tail of a string of vehicles.

The constant time-gap spacing policy in which the desired spacing is proportional to speed should be used. With the constant time-gap spacing policy, both string stability and individual vehicle stability can be ensured in an autonomous manner.

ACC

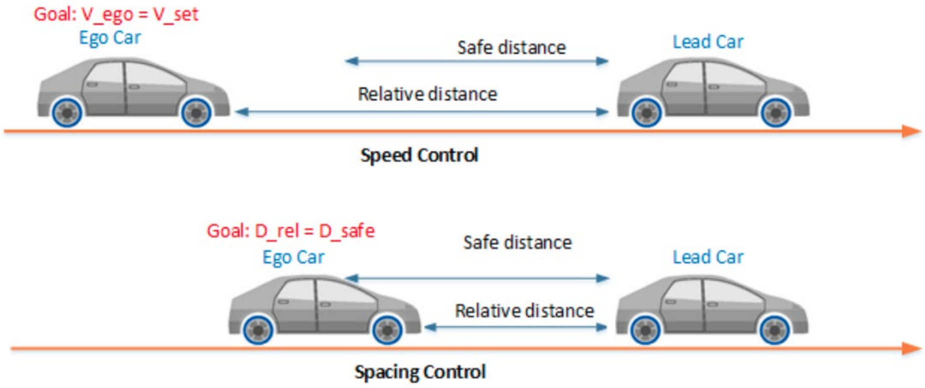
- The safe distance between the lead car and the ego car is a function of the ego car velocity,

$$D_{safe} = D_{default} + T_{gap} \times V_{ego}$$

- The ACC can be summaried as:

$$D_{rel} < D_{safe}, \qquad \ddot{x}_{des}(t) = -k_p(D_{rel} - D_{set}) - k_i \int_0^t (D_{rel} - D_{set})\,\mathrm{dt}$$

$$D_{rel} \geq D_{safe}, \qquad \ddot{x}_{des}(t) = -k_p(V_{ego} - V_{set}) - k_i \int_0^t (V_{ego} - V_{set})\,\mathrm{dt}$$



- $D_{default}$ is the standstill default spacing
- T_{gap} is the time gap between the vehicles

Summary

In this session, we have learned:

- Vehicle dynamics in longitudinal direction
- Classical control theory and its application
    - Modeling of a mechanical system
    - PID controller and its usage in real application
- Cruise Control
    - Upper level control and lower level control
    - Adaptive Cruise Control: time gap

感谢聆听!

Thanks for Listening