
HE IG VD

Intelligence Artificielle pour les systèmes autonomes (IAA)

Vehicle dynamics and control

Prof. Yann Thoma - Prof. Marina Zapater

Février 2024

Basé sur le cours du Prof. A. Geiger

Hes·so Haute Ecole Spécialisée de Suisse occidentale

ReDS Reconfigurable & embedded Digital Systems

TE1 – QCM et questions théoriques

Contenu et préparation du TE du 23.4.2023

→ Contenu:

- Chapitres de théorie du 01 au 05 (aujourd’hui) y compris.

→ Déroulement:

- QCM
- Questions théoriques générales

→ Préparation:

- Slides du cours avec vidéos et liens fournis
- Vos notes prises durant les discussions des séances de cours

→ Pendant le test:

- Une feuille de notes manuscrites recto-verso

Summary

Today’s lesson

→ **Vehicle dynamics**
- Kinematics
- Bicycle model
- Tires and dynamic bicycle model

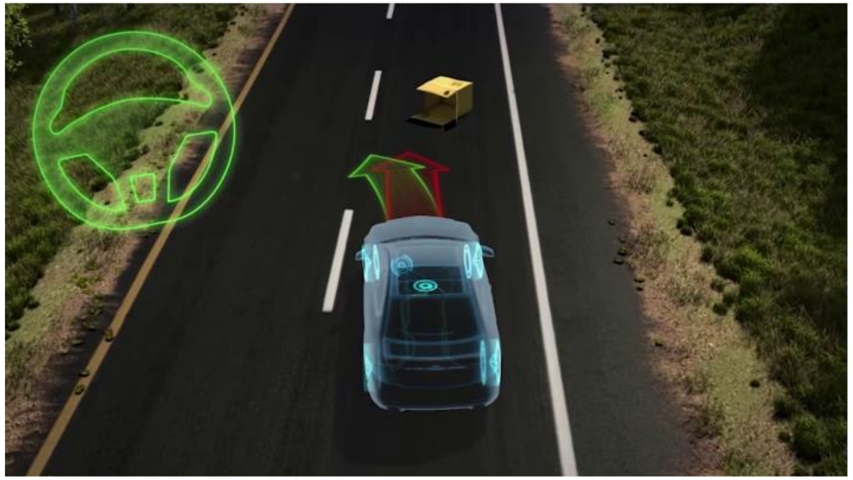
→ Vehicle control
- Controller basics
- Types of controllers

Importance of vehicle dynamics

Enabling accurate control

→ **Kinematics**: geometric description of motion in space (based on reference frames and coordinate systems)

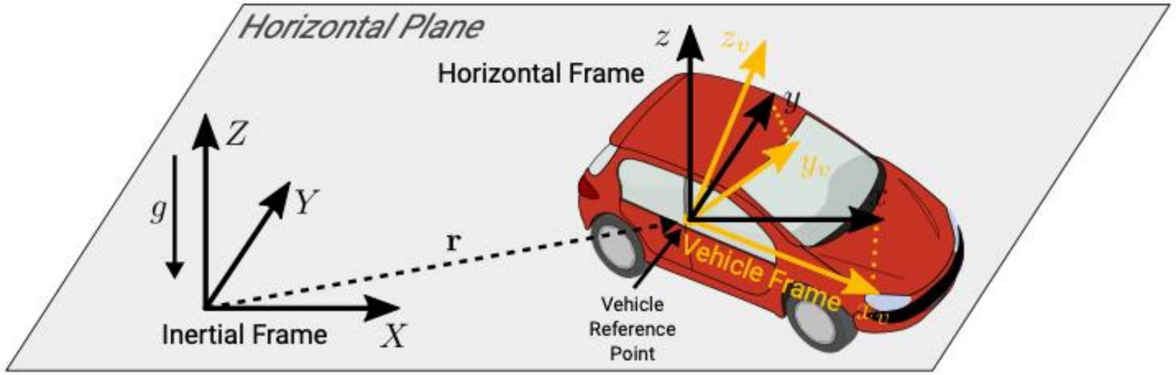
→ **Kinetics** (or dynamics): describe the laws of the causes of motion (forces/moments in Newton's laws)



Position, velocity and acceleration

→ **Inertial frame:** fixed to the earth

→ **Vehicle frame:** attached to the vehicle.



→ **Position** is given by 3 coordinates

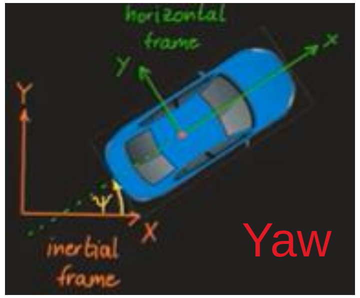
→ Vehicle **velocity** and **acceleration** are the first and second order derivatives of position

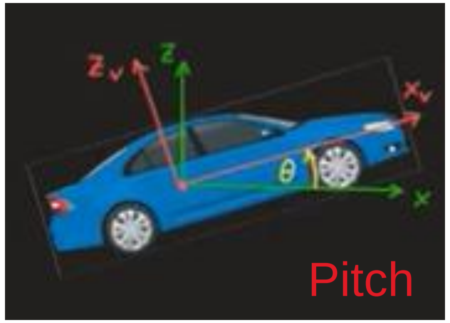
$$\mathbf{r}_P(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} \qquad \mathbf{v}_P(t) = \dot{\mathbf{r}}_P(t) = \begin{pmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \end{pmatrix} \qquad \mathbf{a}_P(t) = \ddot{\mathbf{r}}_P(t) = \begin{pmatrix} \ddot{x}(t) \\ \ddot{y}(t) \\ \ddot{z}(t) \end{pmatrix}$$

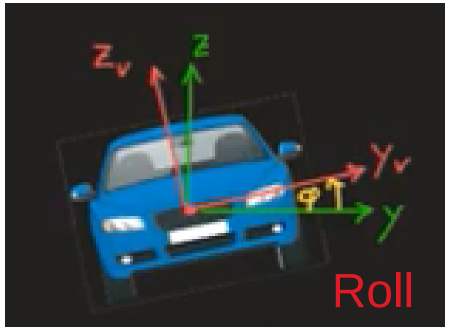
Relation between inertial and vehicle frame

Yaw, pitch and roll

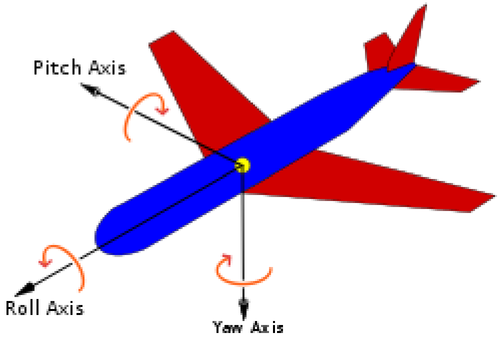
→ Angles in between inertial and vehicle frame define the yaw, pitch and roll







→ For an airplane:



Constraints on position and velocity

Holonomic vs Non-holonomic constraints

Holonomic constraints

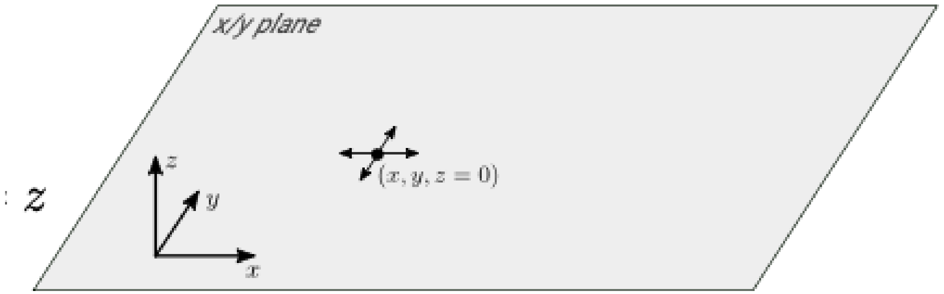
→ Constraints position (config) space

→ Can freely move in any direction

→ Controllable degrees of freedom equal the total degrees of freedom

→ A constraint is defined as f(x,y,z)=0

Example: a 3D-particle in which z=0



Non-holonomic constraints

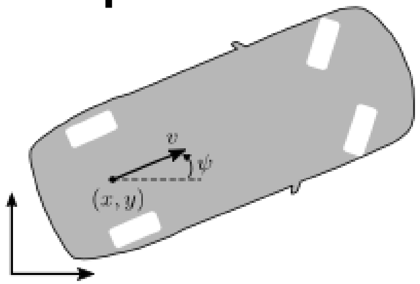
→ Constraints velocity space (i.e. the derivative of position)

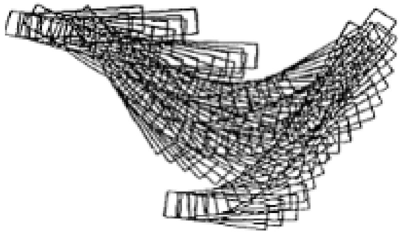
→ <u>Cannot</u> freely move in any direction

→ Controllable degrees of freedom <u>less</u> than the total degrees of freedom

→ Constraint cannot be defined as f(x,y,z)=0

Example: Car





Kinematics of a rigid body

“Cars” are not just one point P, they are a collection of points

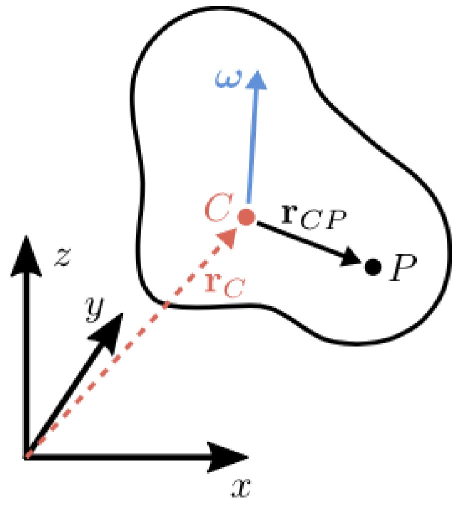
→ A rigid body refers to an infinite collection of small mass points rigidly connected

→ Its motion can be described by an arbitrary reference point (C), plus the relative motion of other points P with respect to C

- C: reference point fixed to the rigid body
- P: arbitraty point of the rigid body
- w: angular velocity of the rigid body

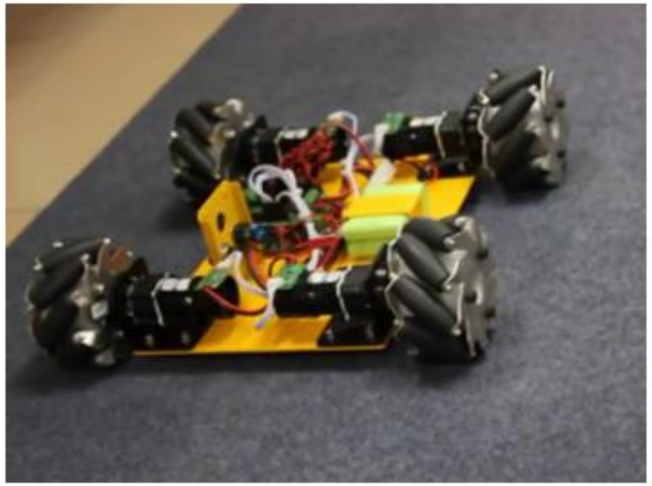
→ A rigid body has 6 Degrees-of-Freedom (DoF)

- 3 positions, 3 rotations



Holonomic vs. Non-holonomic systems

→ A robot can be subject to both holonomic and non-holonomic constraints
→ A car (rigid body in 3D) is kept on the ground by 3 holonomic constraints
→ One additional non-holonomic constraint prevents sideways sliding



Rigid body motion

How can we estimate the motion of the car?

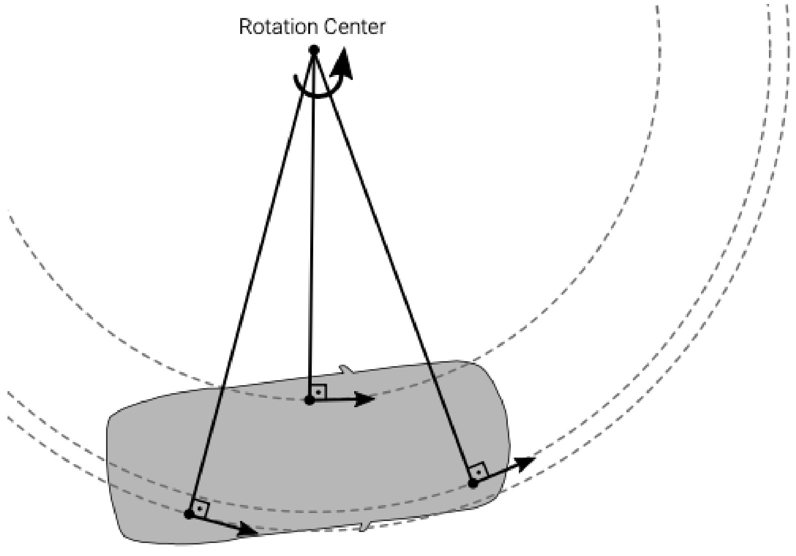
→ **Condition #1** : Different points of the body move along different circular trajectories

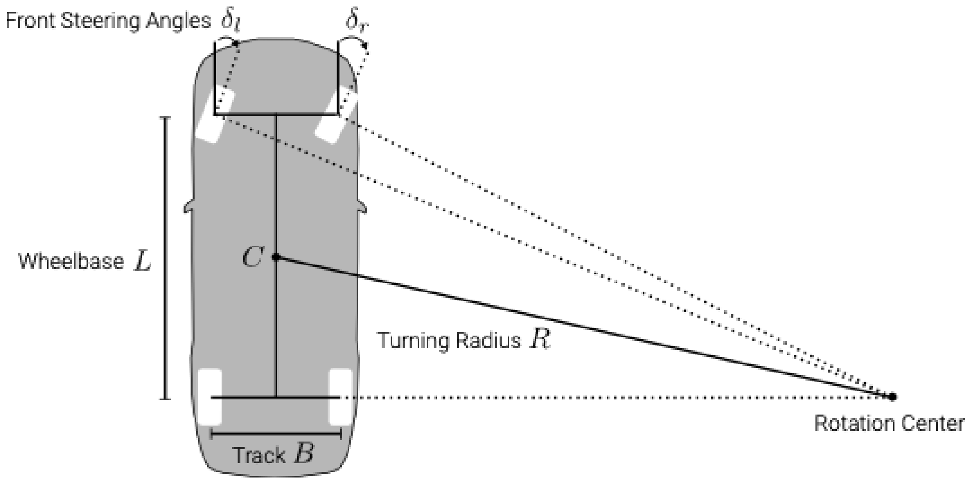
→ **Condition #2** : Rear wheels do not steer

- Slip angle: angle between the velocity of the wheel (trajectory) and the wheel orientation

→ Simplification: Kinematic bicycle model

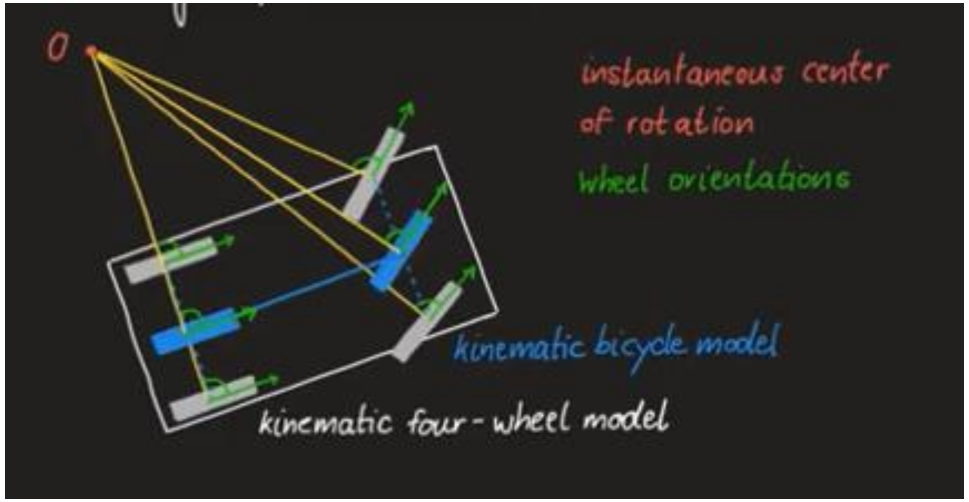
- Two rear weels and two front wheens are combined into one (imaginary) real wheel and one front wheel.

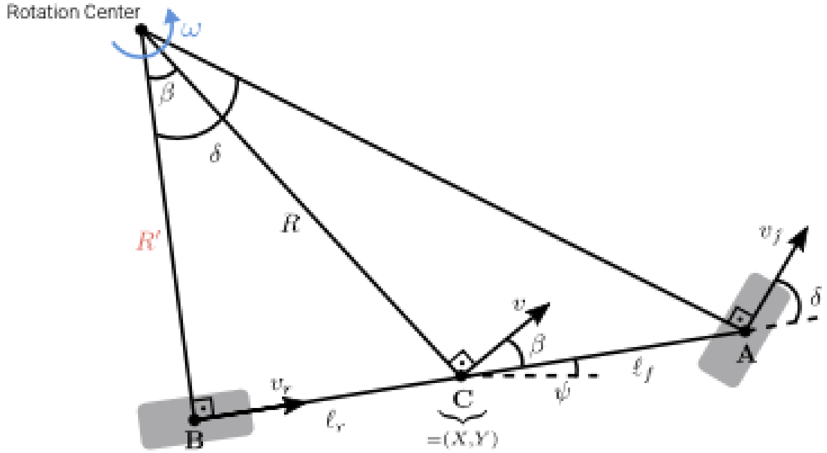




Kinematic Bicycle model

Principle, model and motion equations





$$X_{t+1} = X_t + v\cos(\psi)\,\Delta t$$

$$Y_{t+1} = Y_t + v\sin(\psi)\,\Delta t$$

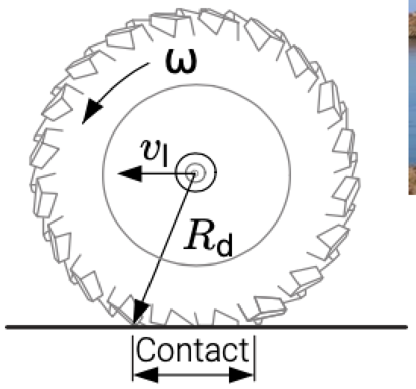
$$\psi_{t+1} = \psi_t + \frac{v\delta}{\ell_f + \ell_r}\,\Delta t$$

What about tires?

Slippery terrains

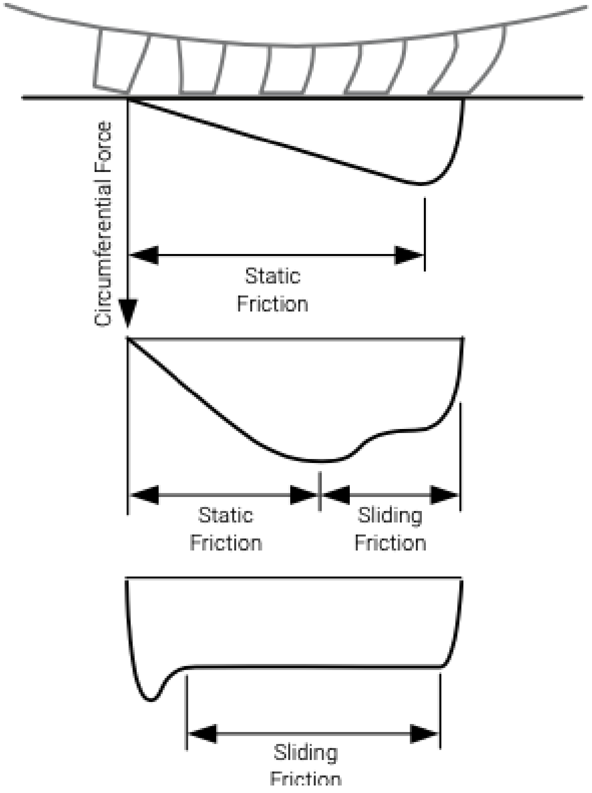
→ Tire models are necessary

→ They describe lateral and longitudinal forces on the tires



→ Longitudinal force:

- As soon as the wheel is driven, **tire tread blocks** adhere to the ground, deform and slip when loosing contact
- If the driving force increases beyond static friction, the blocks slip earlier
- If the tire tread blocks start sliding at the beginning, only friction can be applied



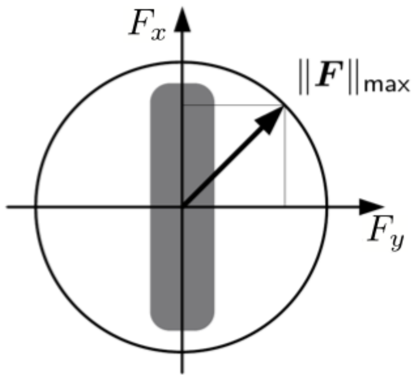
Tread block model

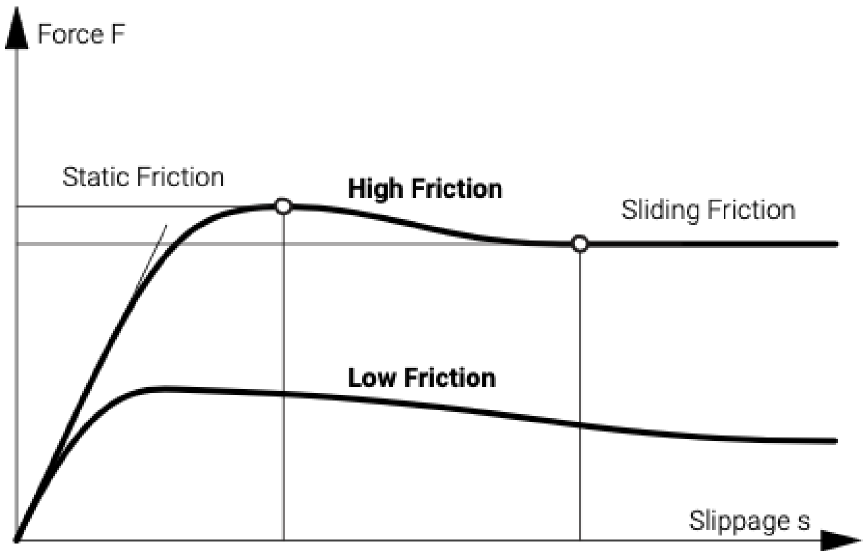
A simple qualitative description

→ Slippage: difference between surface speed of the wheel and vehicle speed
- Force grows linearly with slippage at the beginning
- When large slippage leads to reduction of F (sliding friction < static friction)

→ In slippery terrain (low friction) the maximum reduces due to a decreased static friction
- Tread blocks start sliding earlier

→ Lateral and longitudinal forces should not exceed maximum friction force





Dynamic bicycle model

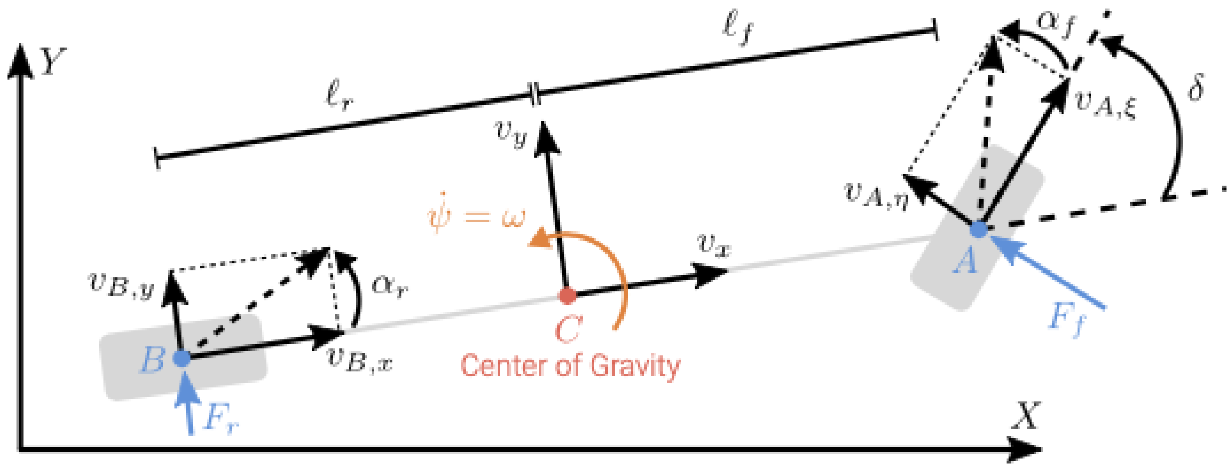
Taking into account tire forces and wheel slip

→ Assumes that the vehicle motion is restricted to the X/Y plane

→ The vehicle is a rigid body

→ Lateral tire forces are considered, generated by a linear tile model

→ It also considers slipping of tires



Summary

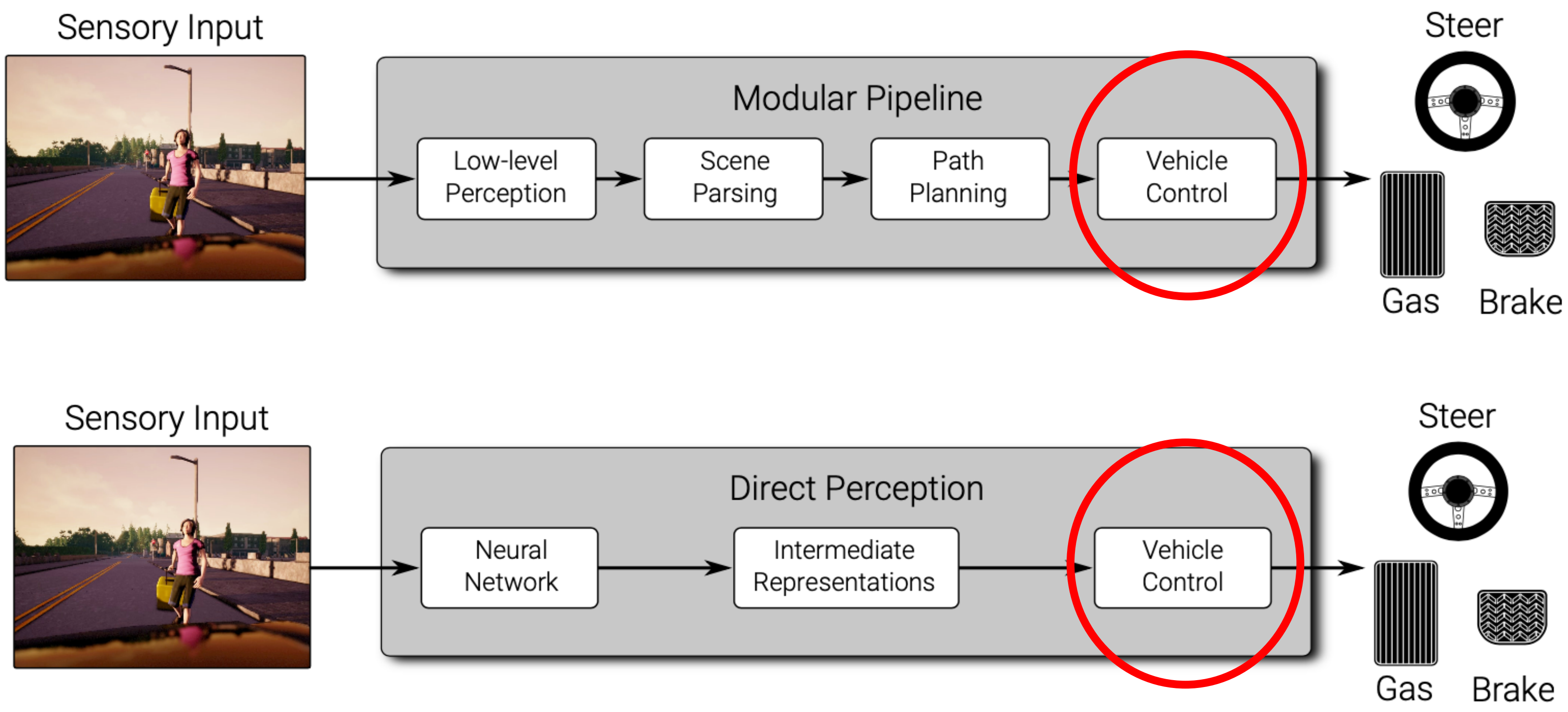
Today's lesson

→ Vehicle dynamics
- Kinematics
- Bicycle model
- Tires and dynamic bicycle model

→ **Vehicle control**
- Controller basics
- Types of controllers

Approaches to self-driving

Require vehicle control



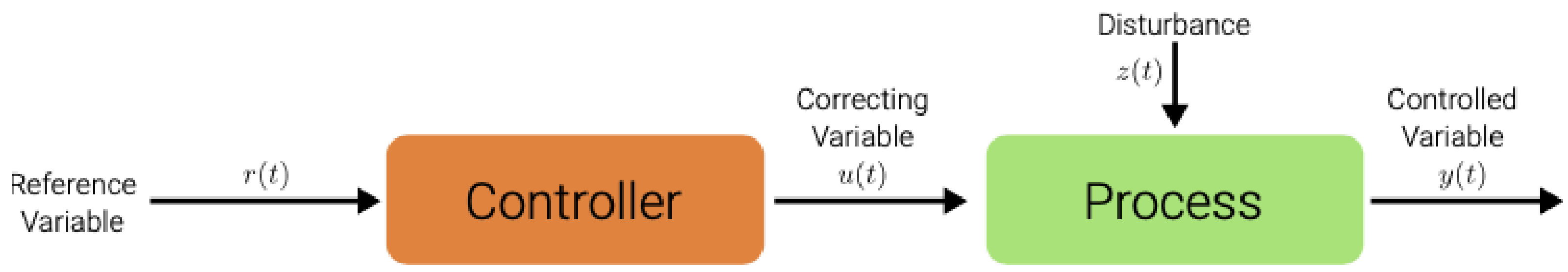
Vehicle control

A brief history of the controllers already deployed in today's cars

→ 1926: Servo braking (Pierce-Arrow)
→ 1951: Servo steering (Chrysler)
→ 1958: Cruise control (Chrysler)
→ 1978: Anti-lock braking system ABS (Bosch)
→ 1986: Traction control system ASR (Bosch)
→ 1995: Electronic stability program ESP (Bosch/BMW)
→ 2000: Adaptive cruise control ACC (Mitsubishi/Toyota/Bosch)
→ 2002: Emergency brake assistant (Mercedes Benz)
→ 2003: Lane-keeping assistant (Honda)
→ 2007: Automatic park assistant (Valeo)

Controller basics

Open-loop control



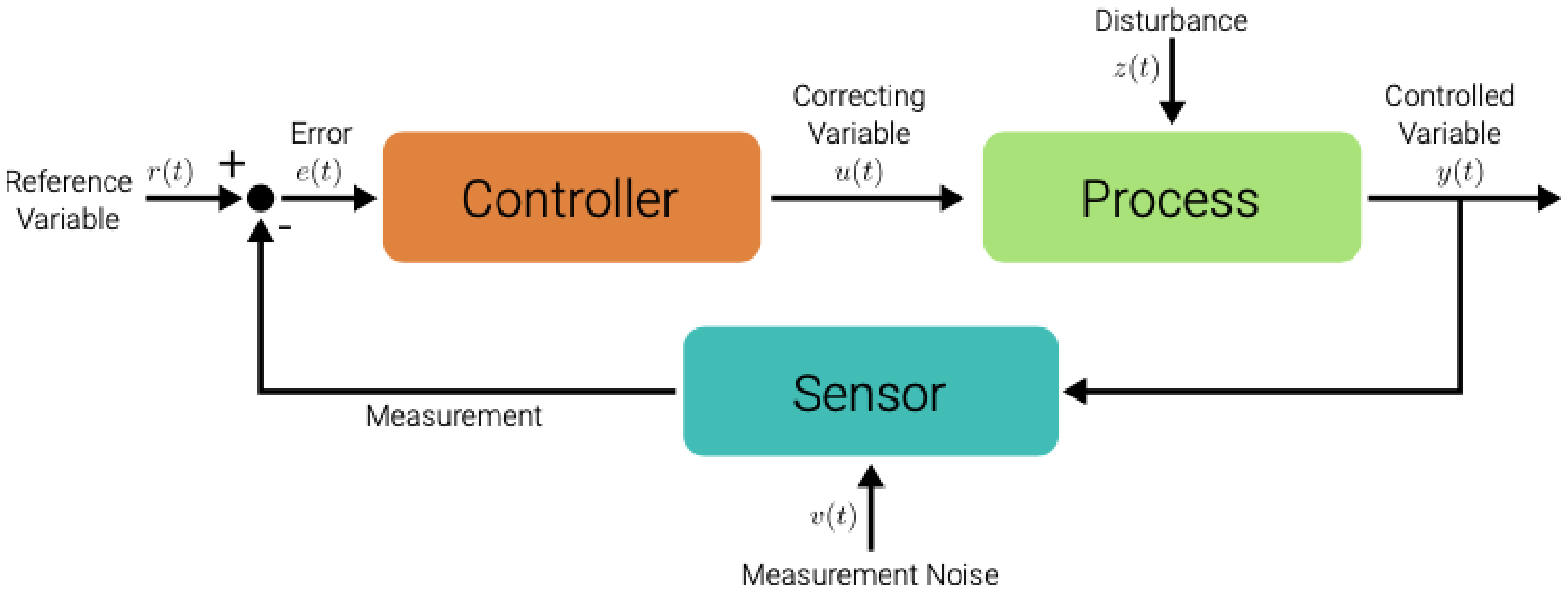
→ Requires precise knowledge of the plan and the influence factors

→ No feedback about the controlled variable

→ Cannot handle unknown disturbances, resulting in drifts

Controller basics

Closed-loop control



→ Exploits feedback to minimize error between reference and measurement

→ This is what is used for controlling a car

Closed-loop control

Basics

→ A vehicle needs to be controlled longitudinally and laterally

→ Three types of controllers:

1. **Black box** : don't require knowledge about the process
2. **Geometric** : exploit geometric relationships between the vehicle and the path, resulting in compact control laws for path tracking
3. **Optimal** : use knowledge of the system and minimize an objective function over future time steps

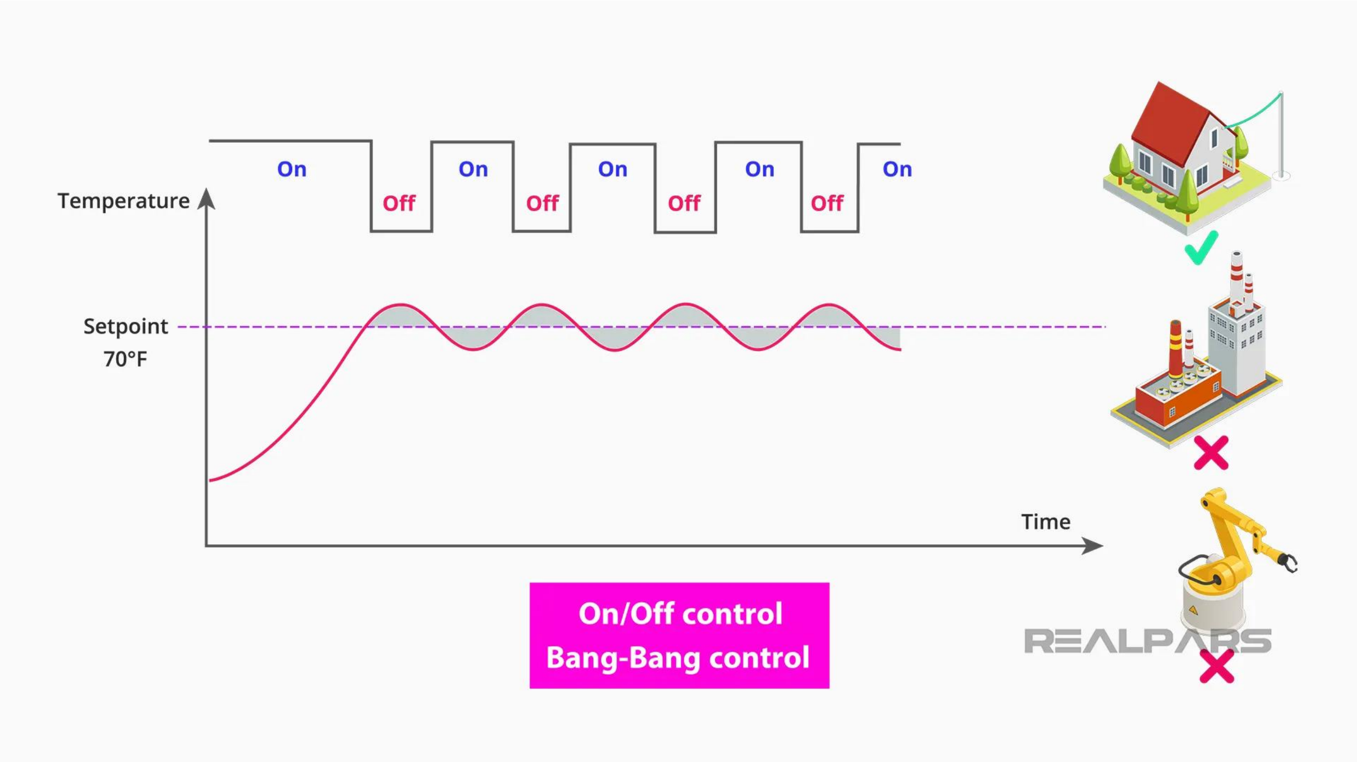
1.Black-box controller

Bang-bang control (a.k.a. “hysteresis controller” or “on/off control”)

→ Simple example: a house thermostat

→ Switches between two states

$$u(t) = \begin{cases} u_1, & \text{if} \quad e(t) \geq \tau \\ u_2, & \text{otherwise} \end{cases}$$



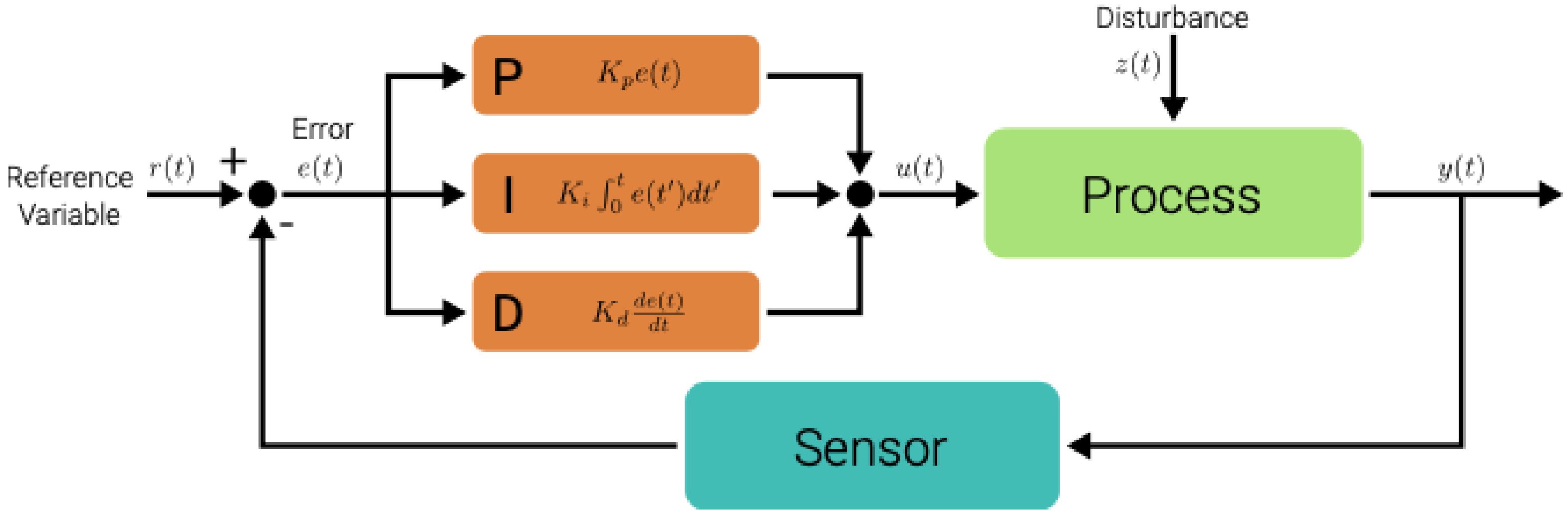
1.Black-box controller

PID Control

→ **P**roportional : the P element alone leads to overshooting/oscillation

→ **I**ntegral : corrects residual errors by integrating past error measurements

→ **D**erivative : alleviates oscillation by introducing a damping behavior

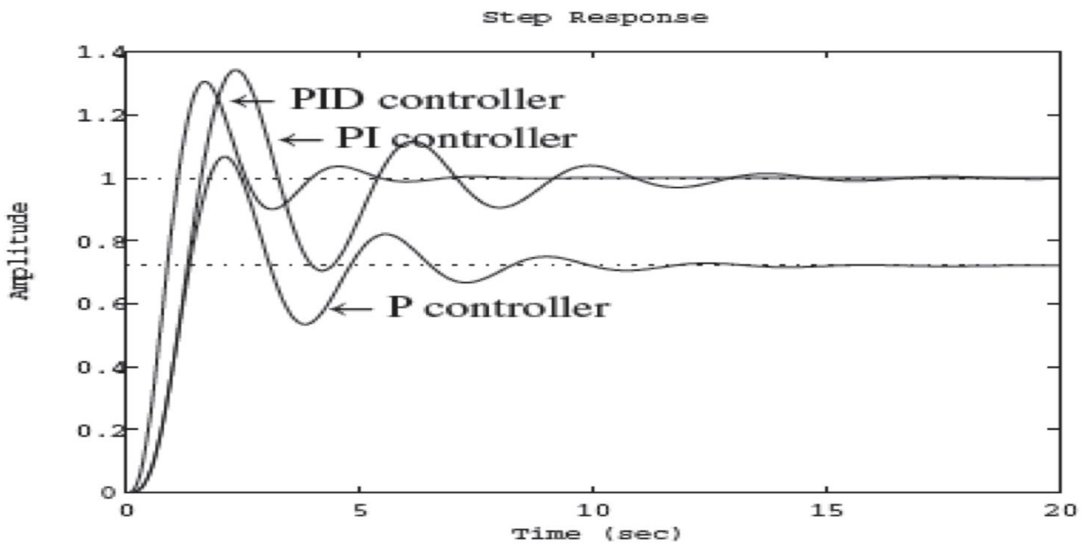


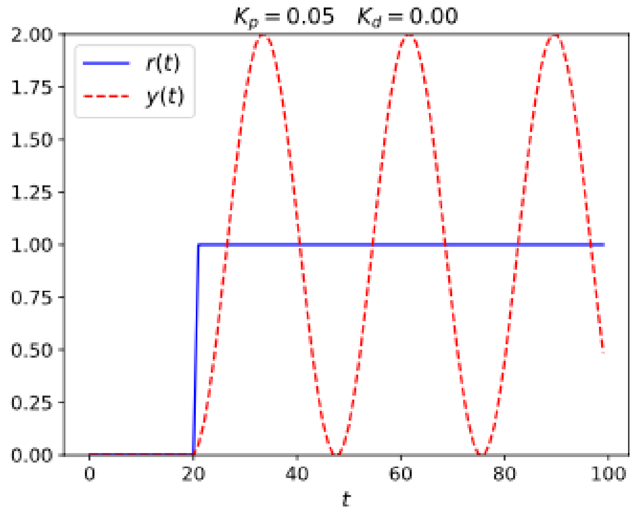
P vs PD vs PID

Controller performance

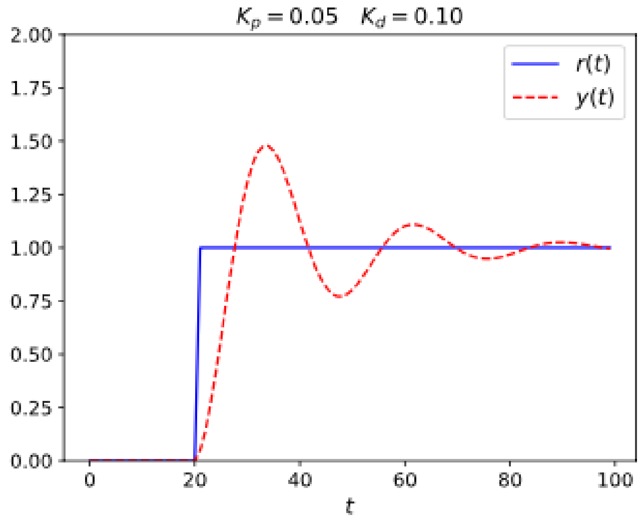
→ Controlled variable : position y(t) = x(t)

→ Correcting variable : acceleration u(t) = a(t)





P control

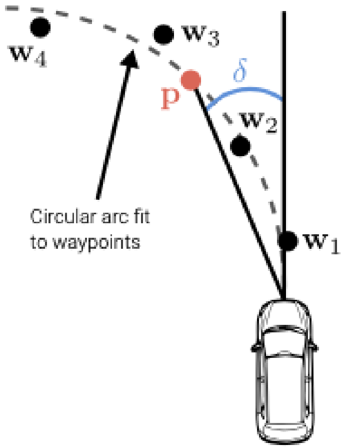


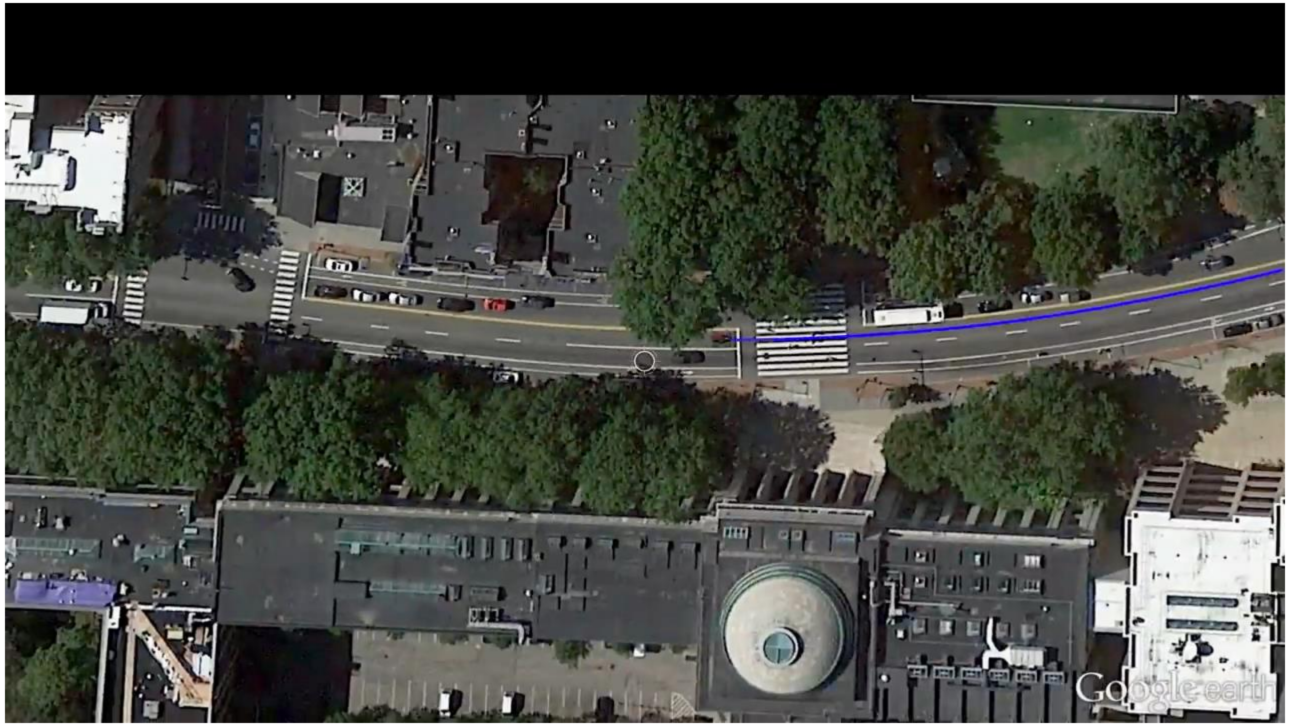
PD control

An example for cars

Waypoint-based Vehicle control

→ We want the vehicle to follow waypoints
→ Input: waypoints
→ Velocity: longitudinal PID control
→ Steering angle: lateral PID control





https://www.youtube.com/watch?v=4Y7zG48uHRo

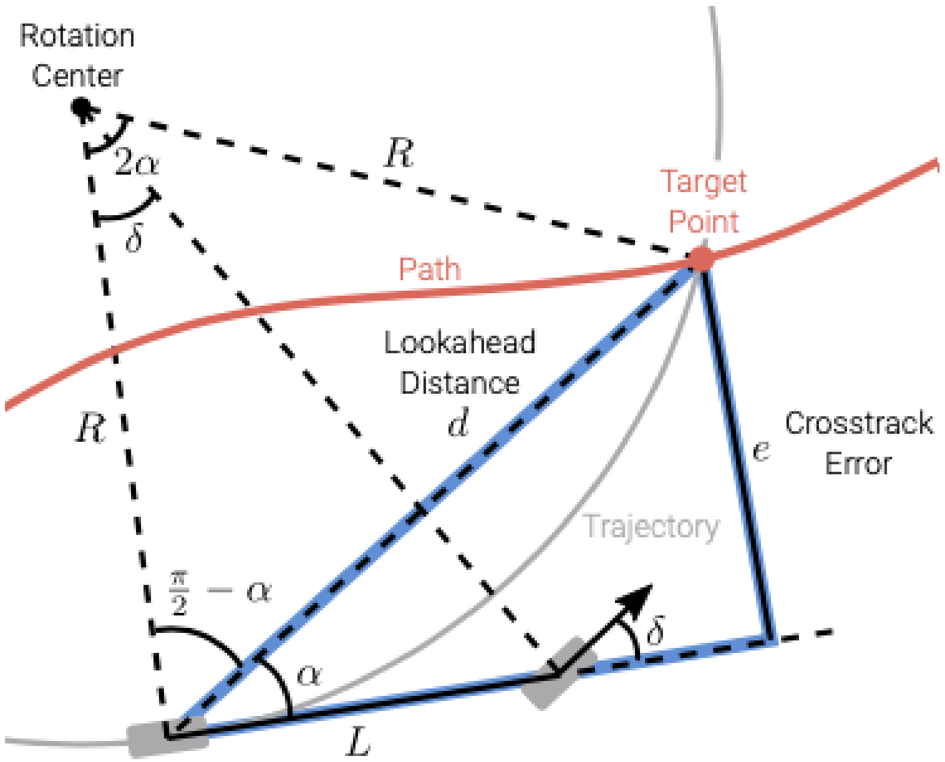
2.Geometric control

Pure pursuit control

→ **Goal:** track a target point at a lookahead distance ”d” to follow path.

→ Exploiting geometric relationship between vehicle and path to follow

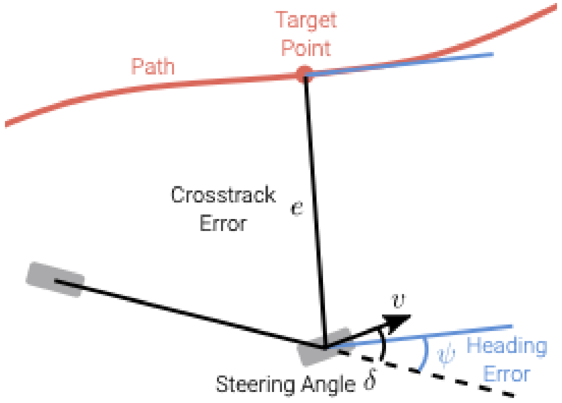
→ Steering angle determined by angle between vehicle heading direction and the lookahead direction

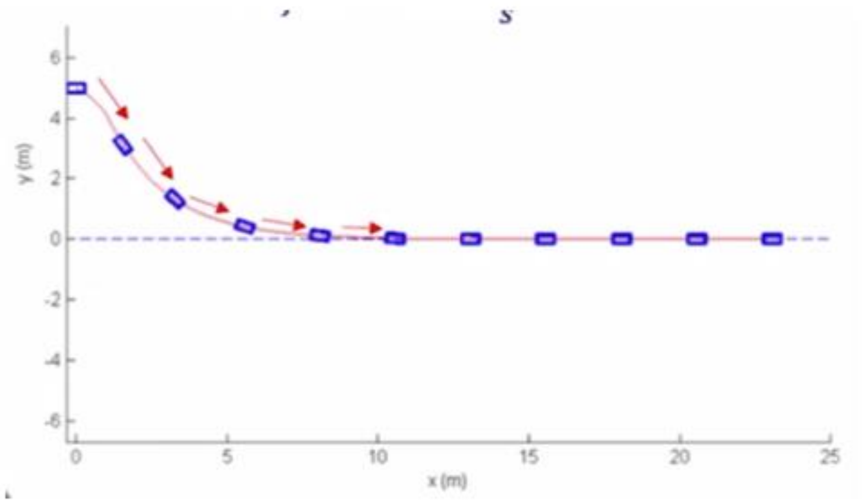


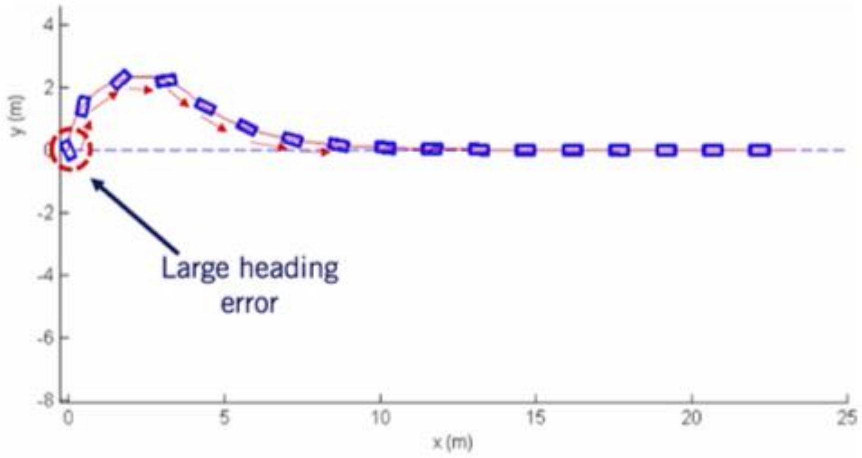
2.Geometric control

Stanley control

→ Control Law used by Stanley in a DARPA Challenge 2008
→ Reference at front axle, no lookahead
→ Combines heading and crosstrack error
→ Crosstrack error converges exponentially (so, fast) to zero
→ Works for small velocities without disturbances







2.Geometric control

Stanley control

→ Video of DARPA Challenge 2008

https://www.youtube.com/watch?v=M2AcMnfzpNg

3.Optimal control

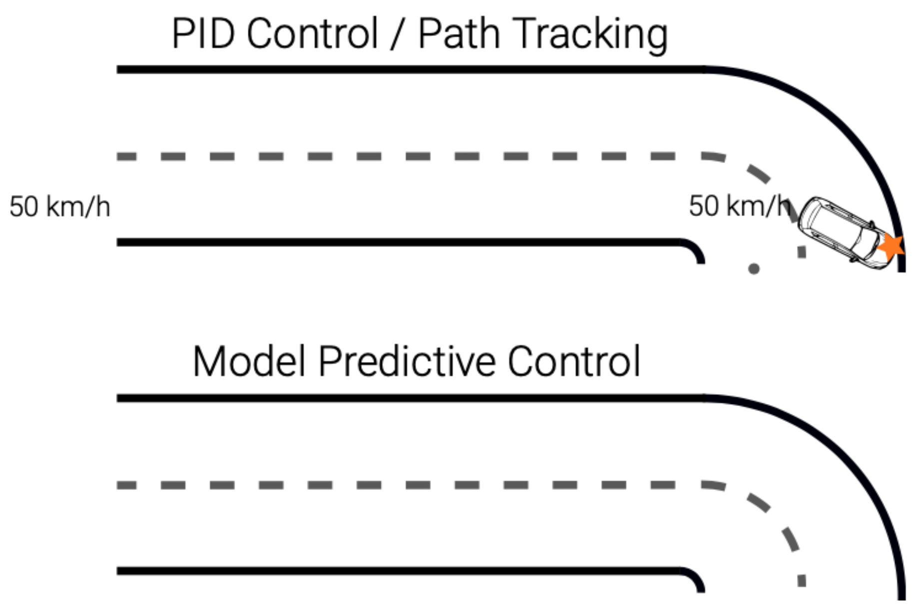
Model Predictive Controller (MPC)

→ We formalize the problem as an optimization
  - Based on LQR (Linear Quadratic Regulator)

→ Non-linear cost function and dynamics

→ Flexible: allows for rececing window and incorporation of constraints

→ But expensive : non-linear optimization required at every iteration



$$
\begin{aligned}
\underset{\delta_1,\dots,\delta_T}{\operatorname{argmin}} \quad & \sum_{t=1}^{T} C_t(\mathbf{x}_t, \delta_t) && \text{(sum of costs)} \\
s.t. \quad & \mathbf{x}_1 = \mathbf{x}_{\text{init}} && \text{(initialization)} \\
& \mathbf{x}_{t+1} = f(\mathbf{x}_t, \delta_t) && \text{(dynamics model)} \\
& \underline{\delta} \leq \delta_t \leq \overline{\delta} && \text{(constraints)}
\end{aligned}
$$

3.Optimal control

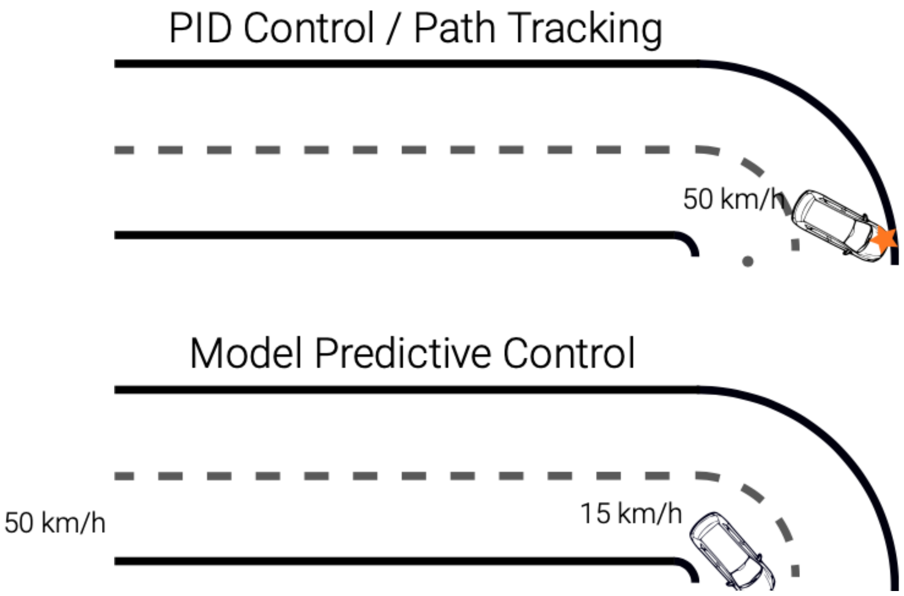
Model Predictive Controller (MPC)

→ We formalize the problem as an optimization
  - Based on LQR (Linear Quadratic Regulator)

→ Non-linear cost function and dynamics

→ Flexible: allows for rececing window and incorporation of constraints

→ But expensive : non-linear optimization required at every iteration



$$
\begin{aligned}
\underset{\delta_1,\ldots,\delta_T}{\operatorname{argmin}} \quad & \sum_{t=1}^{T} C_t(\mathbf{x}_t, \delta_t) && \text{(sum of costs)} \\
s.t. \quad & \mathbf{x}_1 = \mathbf{x}_{\text{init}} && \text{(initialization)} \\
& \mathbf{x}_{t+1} = f(\mathbf{x}_t, \delta_t) && \text{(dynamics model)} \\
& \underline{\delta} \leq \delta_t \leq \overline{\delta} && \text{(constraints)}
\end{aligned}
$$

Summary

Vehicle control

→ Open-loop controllers cannot handle unknown disturbances

→ In practice, we thus require closed-loop control with sensor feedback

→ Black box controllers don't require knowledge about the process

- Most popular black box controller: **PID controller**

→ Geometric controllers exploit geometric relationships for path tracking

→ Optimal controllers use a vehicle model and optimize a cost function

- **MPC** is the most flexible and powerful approach
- However, MPC requires solving an optimization problem at every time step

HE IG VD

TODO's for today

Exercises

1. Vehicle dynamics – sommaire des techniques

2. Controller of the Crazyflie: https://www.bitcraze.io/documentation/repository/crazyflie-firmware/master/functional-areas/sensor-to-control/controllers/
    - How many and what type of controllers come into the crazyflie and how do they work?

REDS
Institut
Reconfigurable
and Embedded
Digital Systems

Haute Ecole Spécialisée
de Suisse occidentale