3. Longitudinal & Lateral Control
第3课：纵向、横向控制

金力 Li Jin
li.jin@sjtu.edu.cn
上海交通大学密西根学院
Shanghai Jiao Tong University UM Joint Institute

上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

Recap

- Architecture of autonomous driving
  - How human drive
  - How a computer drives
  - Decision-making architecture
- Action
  - Trajectory tracking
  - Trajectory planning
  - Route planning
  - Trip planning

Outline

- Longitudinal control
  - Speed tracking
  - Trajectory tracking
  - Vehicle following
- Lateral control
  - Lane keeping
  - Lane changing

Longitudinal & lateral control

Longitudinal control

- Device: throttle, brake, (clutch)
- Action: push/release pedals
- Influences speed and position

Lateral control

- Device: steering wheel
- Action: turn steering wheel
- Influences angular velocity and direction

Longitudinal control 纵向控制

- Speed tracking
- Position tracking
- Vehicle following
- Further discussions

Decision-making problem

- Data
  - Vehicle parameters
  - How you want the vehicle to move
- Decision variables
  - Torque generated by the engine
- Constraints
  - Vehicle dynamics
- Objective
  - Make the vehicle move as you want

- Reference:
  - Attia, R., Orjuela, R., & Basset, M. (2014). Combined longitudinal and lateral control for automated vehicle guidance. *Vehicle System Dynamics*, *52*(2), 261-279.

Longitudinal dynamics

- Vehicle mass = m [kg]
- Vehicle speed = v [m/s]
- Propelling force = F_p [N]
- Aggregate resisting force = F_r [N]
  - Aerodynamic force $F_a = 1/2\rho C_d v^2$
  - Gravitational force $F_g = mg\sin\theta$
  - Rolling resistance force $F_{rr} = C_r mg\cos\theta$
- Dynamic equation

$$m\dot{v} = F_p - F_r$$

where $\dot{v}$ is the time-derivative of v, i.e., the acceleration.

(# in the title means blackboard demonstration)

Longitudinal dynamics

- Propelling dynamics
  - Moment of inertia of wheel = I_w
  - Rotational speed of wheel = ω
  - Longitudinal force from the ground = F_l
  - Radius of wheel = R
  - Traction torque = T_c
  - Brake torque = T_b

$$I_w\dot{\omega} = -F_l R + T_c - T_b$$

- Assume non-slip rolling:
  - $v = R\omega$
  - $F_p = F_l$
- Assume no power transmission losses
  - $\omega = R_g\omega_e$ (R_g = gearbox ratio)
  - $T_e = R_g T_c$ (T_e = engine torque)

Longitudinal dynamics

- Longitudinal dynamics for synthesis

$$\frac{(mR^2 + I_w)R_g}{R}\dot{v} = T_e - R_gT_b - R_gRF_r$$

- Let $M_t = \frac{(mR^2+I_w)R_g}{R}$, we have

state-dependent

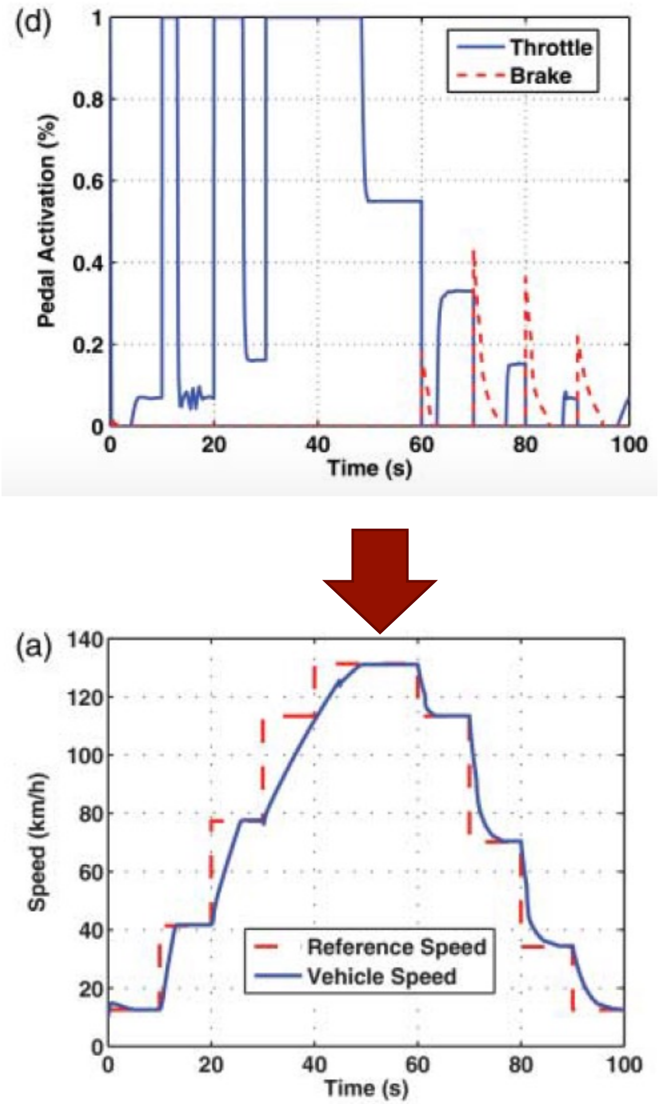
$$\dot{v} = \frac{1}{M_t}(T_e - R_gT_b - R_gRF_r)$$

state & output   input

- Main messages
  - We can tune the speed by playing with the engine torque
  - Such influence is indirect; we can directly set $\dot{v}$ but not v
  - Recall that the resisting force F_r depends on speed (through aerodynamic force)

Speed tracking

- So how should we set the control input, i.e., the engine torque?
- It depends on how you want the vehicle to move.
- A simple objective:
  - You have a pre-defined reference speed profile $v_{ref}(t)$ for $t \geq 0$
  - You want the vehicle to track the pre-defined speed profile
  - Track means asymptotical convergence.



Speed tracking

- Data
  - Vehicle parameters
  - Reference speed profile $v_{ref}(t)$ for $t \geq 0$
- Decision variables
  - Torque generated by the engine
  - Equivalently, pedal movement: push throttle/brake to some angle…
  - Assume automatic transmission; no gear switch & no clutch…
- Constraints
  - Vehicle dynamics
- Objective
  - Make the actual vehicle speed $v(t)$ asymptotically converge to the reference speed $v_{ref}(t)$, i.e., "tracking"

Speed tracking

- Let's discuss insights before getting into the math.
- Basically, we want to drive the tracking error

$$e(t) = v_{ref}(t) - v(t)$$

to zero.

- Intuitively,
  - If $v(t) < v_{ref}(t)$, i.e., if actual speed < desired speed, then we want push throttle forward and apply more torque;
  - If $v(t) > v_{ref}(t)$, i.e., if actual speed > desired speed, then we want release throttle backward and apply less torque or disengage throttle and push brake;
  - If $v(t) = v_{ref}(t)$, i.e., if actual speed = desired speed, then we want to keep the current position of throttle/brake.

Speed tracking

- So, how to quantitatively implement the above insights?
- That is, if we observe $v(t)$ and compute $e(t) = v_{ref}(t) - v(t)$, how should we exactly select the torque or the throttle/brake position?
- Essentially, we need a mapping from our observation to our decision.
- Such a mapping is called a controller.
- Also called a control policy or control law.

$$\begin{array}{c} \text{tracking error} \\ e(t) = v_{ref}(t) - v(t) \end{array} \xrightarrow{\text{controller}} \begin{array}{c} \text{engine torque} \\ T_e(t) \end{array}$$

Speed tracking

- Let's clarify several definitions before proceeding…
- Input 输入: something that you or the environment injects into the system
- Output 输出: something that you observe from the system
- State 状态:
  - Physical interpretation: a set of information that describes the current status, configuration, condition etc. of a system.
  - Mathematical interpretation: a set of variables that, along with control inputs, uniquely specifies the system's output in the future.

Speed tracking

- The speed tracking problem is essentially the determination of the controller, which specifies the decision (action) according to the observed information (state).
- Formally, a controller is a function that maps states to actions.
- The process of designing this function is called control synthesis.
- If the function is linear -> linear control
- If the function is non-linear -> nonlinear control
- If the function is learned from data -> learning-based control

Speed tracking

- So, how can we synthesize a speed tracking controller?
  - Linear? Non-linear?
  - Parameter values?
- Recall our objective: make the actual speed $v(t)$ track the reference speed $v_{ref}(t)$.
- Equivalently, we want the tracking error $e(t)$ to vanish.

$$e(t) = v_{ref}(t) - v(t)$$

$$\dot{v} = \frac{1}{M_t}\left(T_e - R_g T_b - R_g R F_r\right)$$

$$\dot{e}(t) = \dot{v}_{ref}(t) - \frac{1}{M_t}\left(T_e(t) - R_g T_b - R_g R F_r(t)\right)$$

Speed tracking

- I claim that the following controller can achieve what we want

$$T_e(t) = M_t\left(ke(t) + \dot{v}_{ref}(t)\right) + R_g R F_r(t)$$

where k is some positive constant.

- When tracking error e is large, we need to increase T_e.
- When reference speed v_{ref} is increasing, we need to increase T_e
- When resisting force F_r is large, we need to increase T_e
- Why this particular form of controller? (Not required in this course.)

Speed tracking: some nonlinear control theory #*

- Consider a Lyapunov function

$$V = \frac{1}{2}e^2.$$

- If you plug the controller

$$T_e(t) = M_t\left(ke(t) + \dot{v}_{ref}(t)\right) + R_g R F_r(t)$$

into the dynamic equation

$$\dot{e}(t) = \dot{v}_{ref}(t) - \frac{1}{M_t}\left(T_e(t) - R_g T_b - R_g R F_r(t)\right),$$

you will find that

$$\dot{V}(t) = e(t)\dot{e}(t) = -\frac{k}{2}\left(e(t)\right)^2 = -kV(t).$$

(* means advanced materials not required in this course)

Speed tracking: some nonlinear control theory #*

- Recall results from ordinary differential equations
- Solution to

$$\frac{d}{dt}V(t) = -kV(t)$$

is

$$V(t) = V(0)\exp(-kt).$$

- The above can be re-written as

$$\big(e(t)\big)^2 = \big(e(0)\big)^2 \exp(-kt)$$
$$e(t) = e(0)\exp(-kt/2)$$

- That is, the tracking error $e(t)$ exponentially converges towards 0.

Homework 1

- Write codes that simulates the dynamics of the speed tracking problem.
- Verify that the claimed controller is exponentially convergent.
- For your convenience, let's simplify the problem a bit.
- We re-write the dynamic equation

$$\dot{v} = \frac{1}{M_t}\left(T_e - R_g T_b - R_g R F_r\right)$$

as

$$\dot{v} = \alpha(T_e - \beta - \gamma v^2)$$

and let $\alpha = \beta = \gamma = 1$.

Homework 1

Problem 1: Speed tracking

- Part a: Reformulate the problem in discrete time (so that you can code).
- Suppose that you have an ODE
$$\frac{d}{dt}x(t) = f\big(x(t)\big).$$
- You can discretize it as
$$\frac{x(t+\Delta t) - x(t)}{\Delta t} = f\big(x(t)\big).$$
- So, the discrete dynamic equation is
$$x(t+\Delta t) = x(t) + f\big(x(t)\big)\Delta t.$$
- This is how you can update the state in Python/Matlab/C++…
- Hint: your response should be in this form:
$$v(t+\Delta t) = v(t) + f\big(v(t)\big)\Delta t$$

Homework 1

- Part b: Find a controller T_e such that
$$e(t)\dot{e}(t) = -\frac{1}{2}\big(e(t)\big)^2.$$
- Part c: Assume $v_{ref}(t) = 1$ for all $t \geq 0$. Assume zero initial condition, i.e., $v(0) = 0$. Simulate the ODE
$$\dot{v}(t) = \alpha(T_e(t) - \beta - \gamma v(t)^2)$$

with the controller that you found in part 2. Plot $v(t)$ vs. t and label v_{ref}; please also label the axes!

- Part d: Select a time-varying reference speed $v_{ref}(t)$ and see if your controller still works. (Everyone should have his/her unique reference speed profile.) Plot $v(t)$ vs. t and label v_{ref}.

Position tracking

- Data
  - Vehicle parameters
  - Reference position profile $x_{ref}(t)$ for $t \geq 0$
- Decision variables
  - Pedal movement: push throttle/brake to some angle…
- Constraints
  - Vehicle dynamics
- Objective
  - Make the actual position $x(t)$ asymptotically converge to the reference position $x_{ref}(t)$, i.e., “tracking”

Position tracking

- Step 1: generate speed profile
  - Assume $\Delta = 1$ (unit time step)
  - Suppose a position profile $x(0), x(1), x(2), \ldots$
  - A simple (but not always feasible) scheme:
    $$v(t) = x(t+1) - x(t)$$
  - As long as the position profile is not bizarre, the above scheme should be OK.
  - You can say that the above scheme is a speed controller!

    Speed controller: position -> speed
- Step 2: use the speed tracking strategy that we discussed before.

Homework 1

Problem 2: trajectory tracking

- Part a: Select a **linear** position profile, generate a corresponding speed profile
- Part b: Simulate the position tracking process. (Everyone should have his/her unique reference speed profile.) Plot the position $x(t)$ vs. t and label the reference position $x_{ref}(t)$.
- Part c: Select a **nonlinear** position profile, generate a corresponding speed profile
- Part d: Simulate the position tracking process. (Everyone should have his/her unique reference speed profile.) Plot the position $x(t)$ vs. t and label the reference position $x_{ref}(t)$.

Vehicle following

- Data
  - Vehicle parameters
  - Real-time relative position $x_{lead}(t)$ and absolute speed $v_{lead}(t)$ of the leading vehicle
- Decision variables
  - Pedal movement: push throttle/brake to some angle…
- Constraints
  - Vehicle dynamics
  - Safety constraints (no collision)
- Objective
  - Maintain a steady-state distance d to the leading vehicle., “following”

Vehicle following

- Differences w.r.t. position tracking
  - No pre-defined position/speed profile
  - Data received in a real-time manner rather than in a one-time manner at the beginning.
- Again, we assume unit time step.
- Dynamic equation

$$x(t+1) = x(t) + v_{lead}(t) - v(t)$$

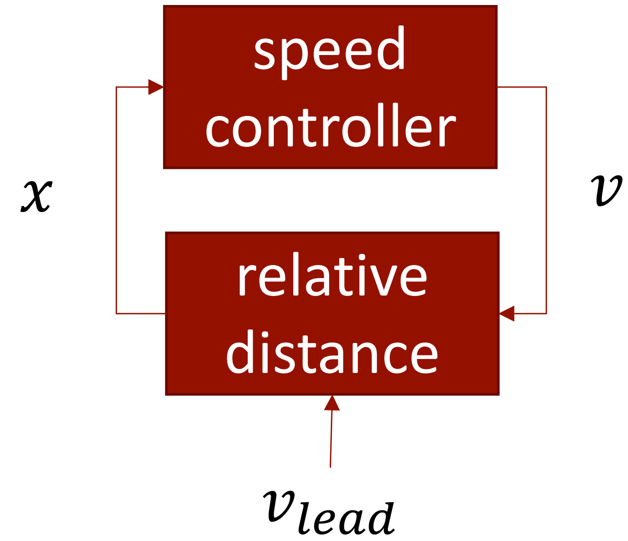
- We want x to be close to the reference distance d
- Steady-state:
  - $x = d$
  - $v = v_{lead}$

Vehicle following

- To do vehicle following, we first need a speed controller, which generates the speed $v(t)$ according to the observation and prediction of the leading vehicle's movement.
- Again, we can consider a naïve controller

$$v(t) = x(t) + v_{lead}(t) - d$$

- Then, implement the above speed via speed tracking



Further discussions

- Linearization 线性化
- Saturation 饱和
- Noise & perturbation 噪声、扰动
- Model identification 模型辨识
- Human driver behavior 人类驾驶行为

Linearization

- Recall that we had a model in the form
$$\dot{v} = \alpha(T_e - \beta - \gamma v^2)$$
- This is a nonlinear ODE, which may not be easy to deal with, if the right-hand side becomes more complex.
- A common technique is to linearize the equation, i.e. applying the idea of Taylor expansion.
- Suppose that the vehicle's speed v is not far away from the equilibrium speed v_0.
$$\dot{v} = \alpha\left(T_e - \beta - 2\gamma\left(v_0 + (v - v_0)\right)^2\right)$$
$$\approx \alpha\left(T_e - \beta - 2\gamma v_0^2 - 4\gamma v_0(v - v_0)\right)$$
- Thus, the RHS is linear in control input and state!

Linearization

- Why would we linearize?
- Recall that the solution to

$$\dot{x} = -kx$$

is

$$x(t) = x(0)\exp(-kt).$$

- That is, the state exponentially converges to the equilibrium, which happens to be the origin in the above ODE.
- The theory of dealing with a linear dynamical system

$$\dot{x} = ax + bu$$

is very well developed and validated in practice.
- Therefore, in many cases, linearization suffices; no nonlinear control has to be done…

Saturation

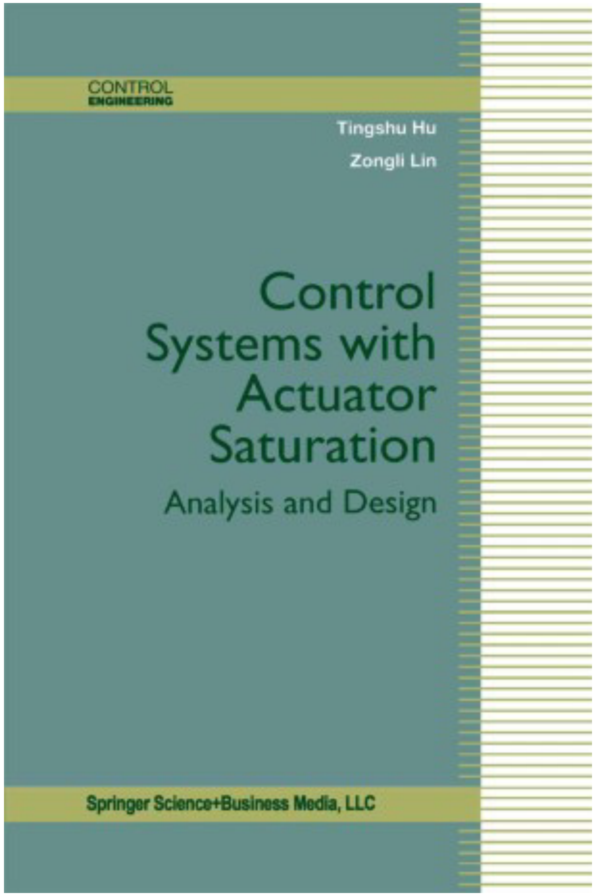
- Recall that the torque (control input) for speed tracking is given by

$$T_e(t) = M_t\left(ke(t) + \dot{v}_{ref}(t)\right) + R_g R F_r(t)$$

- What if the RHS in the above exceeds the maximally attainable torque?
- The attainable torque is upper-bounded by the physics of the engine.
- Let $\bar{T}_e$ be the upper bound.
- If $T_e \leq \bar{T}_e$, we are good.
- If $T_e > \bar{T}_e$, we say the controller to be saturated.
- That is, the machine cannot implement our command.

Saturation

- Saturation can be very tricky in control synthesis.
- At least, saturation slows down the system's convergence.
- Sometimes, saturation can even destabilize a system.
- Two approaches to addressing this problem
  - Lazy approach: design a controller that never reaches its limits
  - Systematic approach: incorporate the saturation nonlinearity in the design process (reachability, stabilizability, etc.)



Problem 3: Saturation

For the torque profile $T_e(t)$ that you generated in problem 2, let

$$\bar{T}_e = 0.8 \max_t T_e(t) .$$

Suppose now that the torque cannot exceed $\bar{T}_e$. That is, the actual torque $T_a(t)$ is given by

$$T_a(t) = \min\{\bar{T}_e, T_e(t)\}$$

where $T_a(t)$ is generated by the controller you obtained in Problem 2. Simulate the trajectory again and discuss the difference from the result in problem 2.

Noise & perturbations

- Recall the dynamic equation is

$$\dot{v} = \alpha(T_e - \beta - \gamma v^2)$$

- The above assumes that the model perfectly matches reality.
- But this never happens. What should we do?
- A random noise can be added to the RHS!

$$\dot{v} = \alpha(T_e - \beta - \gamma v^2) + \epsilon$$

- The noise may result from
  - Modeling error
  - Unmodeled dynamics
  - Environmental perturbations
  - Observation & actuation errors

Noise & perturbations

$$\dot{v} = \alpha(T_e - \beta - \gamma v^2) + \epsilon$$

- Control synthesis is more challenging, since everything becomes random.
- Essentially, we are no longer able to exactly tune $v(t)$, since the speed no becomes a stochastic process $\{V(t); t \geq 0\}$!
- Instead, we can tune its probability distribution!
- That is, we can specify the cumulative distribution function (PDF) $F_V(v, t)$.
- We can no longer ensure that the speed exactly tracks the reference speed profile.

Noise & perturbations

- Instead, we can only say something as follows:
  - At time t, the probability that the actual speed lies in the interval $((t)-\delta, v_{ref}(t)+\delta]$ is given by

$$\Pr\{V(t)\in\left(v_{ref}(t)-\delta, v_{ref}(t)+\delta\right]\}$$
$$=F_V\left(v_{ref}(t)+\delta, t\right)-F_V(v_{ref}(t)-\delta, t)$$

  - The variance of the tracking error converges to 0, i.e.,

$$\lim_{t\to\infty}\mathrm{E}\left[\left(V(t)-v_{ref}(t)\right)^2\right]=0.$$

- In summary, when there is random noise,
  - The actual speed may or may not track the reference speed well;
  - But you can tune your controller so that the actual speed is more likely to track well than not.

Problem 4: Noise

Suppose that the update equation is

$$v(t+\Delta t) = v(t) + f\big(v(t)\big)\Delta t + \epsilon,$$

where f is your response to problem 1a and ϵ is a white noise, i.e., a normally distributed random variable with zero mean; select the variance of ϵ on your own. Simulate the trajectory again and discuss the difference from the result in problem 2.

Homework requirements

- Independent work
- Only pdf files accepted
- Clearly label the problem number and your response
- Attach codes in the end
- Neat formatting is much appreciated!!!

Model identification

- Recall the dynamic equation

$$\dot{v} = \frac{1}{M_t}\left(T_e - R_g T_b - R_g R F_r\right)$$

- In practice, how to obtain the parameters?
- An even more disturbing questions is: how do you know the form of the model is correct?
- This process is called model identification.
  - Objective 1: determine the form of the dynamic equation (linear? nonlinear?...)
  - Objective 2: determine the parameters of the dynamic equation.

Model identification

- Classical approach: measurements & experiments
- Vehicle mass = m [kg]
- Vehicle speed = v [m/s]
- Propelling force = F_p [N]
- Aggregate resisting force = F_r [N]
  - Aerodynamic force $F_a = 1/2\rho C_d v^2$



Model identification

- Modern (fancier) approach: learning
- Form of dynamic equation

$$\dot{x} = f(x)$$

  - We can conduct some simulations or experiments and record $x(t)$ and $\dot{x}(t)$ for $t \in [0, T]$
  - Then, we use a function $\hat{f}(x; \theta)$ to approximate the actual dynamics $f(x)$, where θ are parameters of the approximation function.
  - A fashionable (and usually good) choice of the approximation function $f(x)$: neural networks!
  - θ is determined by minimizing the error

$$\int_{t=0}^{T} \left(\dot{x}(t) - \hat{f}(x; \theta)\right)^2 dt$$

Model identification

- Estimation of parameters: we can use learning again!
- This is often reasonable, since many vehicle parameters vary over time.
  - Vehicle mass varies with passengers/freight/fuel…
  - Power varies with weather/fuel…
  - Air drag varies with weather…
- Instead of estimating the parameters at one time, we can gradually learn these parameters as the vehicle moves -> adaptive control
- Typical paradigm for adaptive control:
  - Start with a nominal controller
  - Fine-tune the nominal controller as real-time data come in

Model identification: Online & offline

Offline model identification

- Dynamics and parameters that do not significantly vary over time
- Use nominal relations/values
- Everything is done before the model is used for control synthesis
- Cannot adapt to perturbations

Online model identification

- Estimate parameters (and sometimes even dynamics) in real time, as the vehicle moves
- Mimics the learning process of a student driver
- Knowledge is updated continuously
- Can adapt to perturbations

Lateral control

Reference: Attia, R., Orjuela, R., & Basset, M. (2014). Combined longitudinal and lateral control for automated vehicle guidance. *Vehicle System Dynamics*, *52*(2), 261-279.

Lane keeping

- Data
  - Vehicle parameters
  - Geometry of the lane
- Decision variables
  - Steering wheel angle
- Constraints
  - Vehicle dynamics
- Objective
  - Keep the vehicle in the middle of the lane

Lane changing

- Data
  - Vehicle parameters
  - Geometry of the current lane and the target lane
- Decision variables
  - Steering wheel
- Constraints
  - Vehicle dynamics
  - No interference with other vehicles
- Objective
  - Leave the current lane and join the target lane

Summary

- Longitudinal control
  - Speed tracking
  - Trajectory tracking
  - Vehicle following
- Lateral control
  - Lane keeping
  - Lane changing

Next time

Trajectory planning

- Single-vehicle on an empty road
  - 1 dimensional
  - 2 dimensional
- Multi-vehicle planning
- Perception
  - Onboard perception
  - Vehicle-to-vehicle connectivity
  - Vehicle-to-infrastructure connectivity