4. Autonomous driving: Longitudinal control

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

Recap

- Trajectory tracking
  - State, dynamics, action, policy
  - Performance of policy
  - Simulation
- n-D dynamical system & control
  - Modeling
  - Objective
  - Theory*

Outline

- Longitudinal control
  - Dynamical equation
  - State-space model
- Synthesis with tracking & following algorithms
  - Trajectory tracking
  - Vehicle following
- Additional issues
  - Linearization
  - Saturation
  - Noise & perturbation
  - Model identification
  - Human driver behavior

Longitudinal & lateral control

Longitudinal control

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

Lateral control

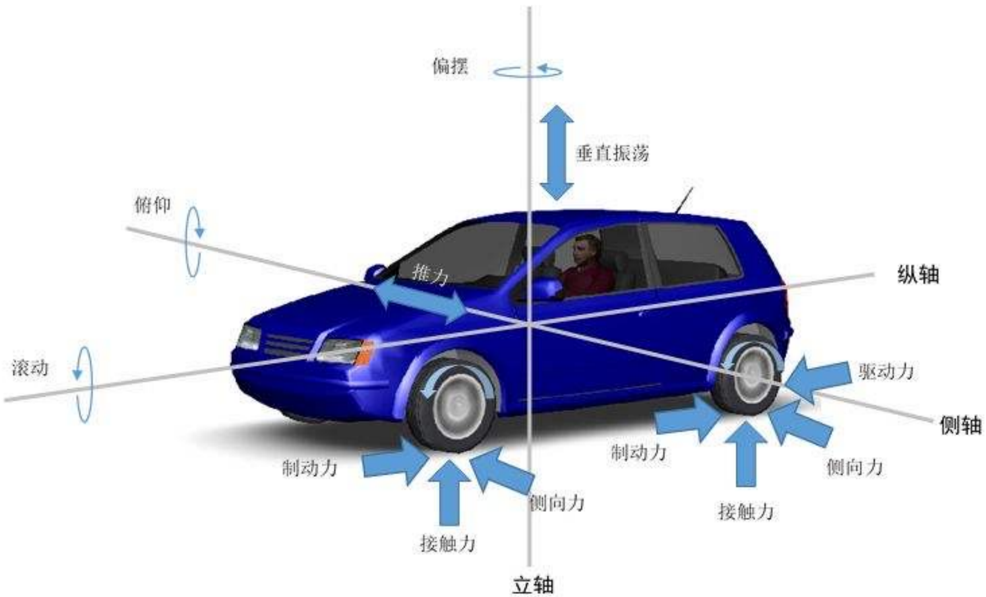
- Device: steering wheel
- Action: turn steering wheel
- Influences angular velocity and direction

Control problem

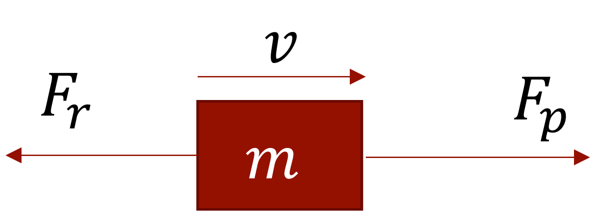
- Parameters
  - Vehicle mass, wheel radius, gear ratio, etc.
  - Air drag coefficient, ground friction coefficient, etc.
- Decision variable
  - Torque generated by the engine
  - Torque generated by the brake
- Constraints
  - Vehicle dynamics
- Objective
  - Make the vehicle move as you want
- Reference:
  - Attia, R., Orjuela, R., & Basset, M. (2014). Combined longitudinal and lateral control for automated vehicle guidance. *Vehicle System Dynamics*, *52*(2), 261-279.

Vehicle dynamics

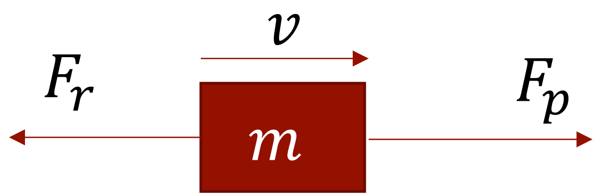
What automobile engineers do in the industry...



What we do in this course...



Longitudinal dynamics



- Vehicle mass = m [kg]
- Vehicle speed = v [m/s]
- Propelling force = F_p [N]
- Aggregate resisting force = F_r [N], i.e., the sum of
  - 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.

Propelling dynamics

Relevant quantities:

- 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

Newton's second law for wheel (simplified):

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

Propelling dynamics

Assumptions:

- non-slip rolling:

$$v = R\omega \text{ (angular speed to linear speed)}$$

- flat ground: $\theta = 0$

$$F_p = F_l \text{ (no impact due to gravity)}$$

- no power transmission losses

$$\omega = R_g \omega_e \ (R_g = \text{gearbox ratio})$$

$$T_e = R_g T_c \ (T_e = \text{engine torque})$$

Thus, wheel dynamics becomes:

$$\frac{I_w}{R}\dot{v} = -F_p R + \frac{T_e}{R_g} - T_b.$$

Longitudinal dynamics

- Combining

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

$$\frac{I_w}{R}\dot{v} = -F_pR + \frac{T_e}{R_g} - T_b$$

leads to

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

- Longitudinal dynamics for control design.
- Let $M_t = \frac{(mR^2+I_w)R_g}{R}$, we have

speed-dependent

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

variable

input

Longitudinal dynamics

- Recall resistant force

$$F_r = \frac{1}{2}\rho C_d v^2 + C_r mg.$$

- Thus, we have

$$\dot{v} = \frac{1}{M_t}\left(T_e - R_g T_b - R_g R\left(\frac{1}{2}\rho C_d v^2 + C_r mg\right)\right)$$

$$= -\frac{\rho C_d R_g R}{2M_t}v^2 + \frac{T_e - R_g T_b}{M_t} - \frac{C_r mg R_g R}{M_t}.$$

- That is, the longitudinal dynamics takes the form

$$\dot{v} = -av^2 + (b_1 T_e - b_2 T_b) - c.$$

- Nonlinear dynamics due to the v^2 term.

Longitudinal dynamics

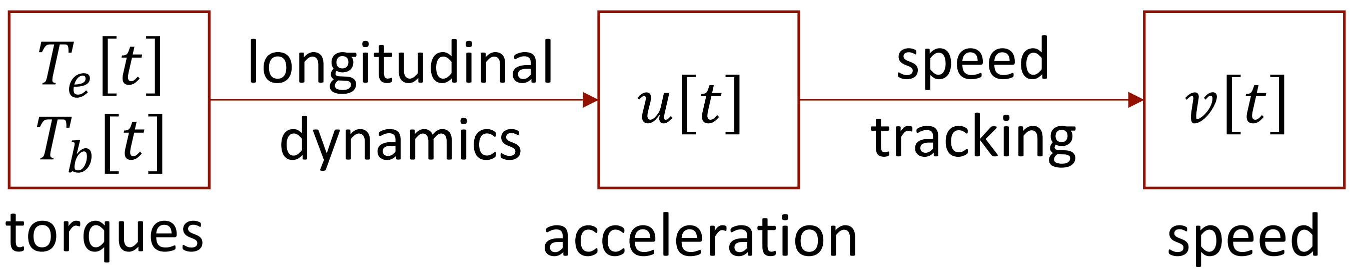
$$\dot{v} = -av^2 + (b_1 T_e - b_2 T_b) - c.$$

Main messages

- We can tune the speed by playing with the torques
- We can consider the total torque
$$T = b_1 T_e - b_2 T_b, \qquad T_e \geq 0, \qquad T_b \geq 0.$$
- One T is selected, we translate it to T_e or T_b.
- Note that either $T_e = 0$ or $T_b = 0$.
$$\dot{v} = -av^2 + T - c.$$
- We can reformulate the longitudinal dynamics in the standard state-space representation.

State-space representation

- Let' begin with the simpler speed tracking problem.
- That is, we want $\lim_{t\to\infty} v[t] = \bar{v}$.
- Recall that in lecture 2, we discussed how to select acceleration $u[t]$ according to speed $v[t]$.
- With longitudinal dynamics involved here, we further need to relate $u[t]$ to the torques $T_e[t]$ and $T_b[t]$.



State-space representation

- Recall our objective: make the actual speed $v(t)$ track the reference speed $\bar{v}$.
- The state is essentially speed $v(t)$:
$$\dot{v} = -av^2 + T - c.$$
- However, a more convenient representation is the tracking error
$$e(t) = \bar{v} - v(t).$$
- Since $v = \bar{v} - e$, we have $\dot{v} = -\dot{e}$ and thus
$$\begin{aligned} -\dot{e} &= -a(\bar{v} - e)^2 + T - c \\ &= -ae^2 + 2a\bar{v}e + T - (a\bar{v}^2 + c). \end{aligned}$$
- Let's use the standard notation x and u.
$$x = e, u = T.$$

State-space representation

- State: tracking error $x(t) \in \mathbb{R}$.
- Control input: "total torque" $u(\mathrm{t}) \in \mathbb{R}$.
- Dynamical equation
$$\dot{x} = ax^2 - 2a\bar{v}x - u + (a\bar{v}^2 + c).$$
- Objective: find a policy $\mu: \mathbb{R} \to \mathbb{R}$ such that
$$\lim_{t\to\infty} x(t) = 0. \qquad \text{(speed tracking)}$$
- One way is to linearize the model in the neighborhood of the origin:
$$\dot{x} \approx -2a\bar{v}x - u + (a\bar{v}^2 + c).$$
- Then, proceed as if we had a linear system.

State-space representation

- Linearization may or may not work.
- A more reliable way is to design a controller based on the nonlinear model.
- I claim that the following policy can achieve the above:
$$\mu(x) = ax^2 + (1 - 2a\bar{v})x + (a\bar{v}^2 + c).$$
- Idea of this policy: transfer the nonlinear system to a linear one.
- Total torque given by $u = \mu(x)$.
- Eventually need to translate u back to T_e and T_b.
- Why will this controller work?

Proof of convergence

- Plug the controller
$$\mu(x) = ax^2 + (1 - 2a\bar{v})x + (a\bar{v}^2 + c)$$
into the dynamic equation
$$\dot{x} = ax^2 - 2a\bar{v}x - u + (a\bar{v}^2 + c):$$
- we have
$$\dot{x} = ax^2 - 2a\bar{v}x - \left(ax^2 + (1 - 2a\bar{v})x + (a\bar{v}^2 + c)\right) + (a\bar{v}^2 + c) = -x.$$
- That is, the **controlled** dynamics is
$$\dot{x} = -x.$$
- This is an LTI system that converges!
- Don't confuse the above with linearization...

Interpretation of policy

$$\mu(x) = ax^2 + (1 - 2a\bar{v})x + (a\bar{v}^2 + c)$$

- When tracking error x is large, we need to increase torque u.
- When reference speed $\bar{v}$ is high, we need to apply a higher torque u.
- When (nominal) resisting force $(a\bar{v}^2 + c)$ is large, we need to apply higher u.
- Why this particular form of controller? (Not required in this course.)
- $\mu(0) = a\bar{v}^2 + c$, which means a non-zero torque is needed to maintain the reference speed.

Outline

- Longitudinal control
  - Dynamical equation
  - State-space model
- Synthesis with tracking & following algorithms
  - Trajectory tracking
  - Vehicle following
- Additional issues
  - Linearization
  - Saturation
  - Noise & perturbation
  - Model identification
  - Human driver behavior

Trajectory tracking

Trajectory tracking

- Data
  - Vehicle parameters (what are they?)
  - Reference trajectory $\bar{x}(t)$ for $t \geq 0$
- Decision variable
  - Engine & brake torques
  - Equivalently, the total torque
  - Intermediate variable: acceleration
- Objective
  - Make the actual position $x(t)$ asymptotically converge to the reference position $\bar{x}(t)$, i.e., "tracking"

Trajectory tracking

Step 1: use acceleration as decision variable

$$\begin{bmatrix} x[t+1] \\ v[t+1] \end{bmatrix} = \begin{bmatrix} 1 & \delta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x[t] \\ v[t] \end{bmatrix} + \begin{bmatrix} 0 \\ \delta \end{bmatrix} u[t].$$

- Consider linear controller

$$u = \mu(x, v) = -k_1(x[t] - \bar{x}[t]) - k_2(v[t] - \bar{v}).$$

Step 2: select total torque to attain desired acceleration:

$$\dot{v} = -av^2 + T - c = u$$
$$\Longrightarrow T = u + av^2 + c.$$

Step 3: translate T to T_e and T_b (recall $T = b_1 T_e - b_2 T_b$)

- If $T > 0$, then $T_e = T/b_1$ and $T_b = 0$.
- If $T < 0$, then $T_e = 0$ and $T_b = -T/b_2$.

Vehicle following

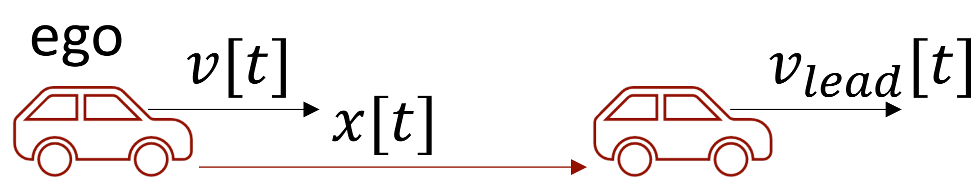
Vehicle following

- Data
  - Vehicle parameters
  - Real-time position $x_{lead}(t)$ and speed $v_{lead}(t)$ of the leading vehicle
- Decision variables: Engine/brake torque
- Constraints
  - Vehicle dynamics
  - Safety constraints (no collision)
- Objective: Maintain a steady-state distance d to the leading vehicle, i.e., "following" (simplistic objective)

https://www.bilibili.com/video/BV13X4y1M7L4?spm_id_from=333.337.search-card.all.click

Vehicle following

- Differences w.r.t. trajectory 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 consider a DT setting with state $\left[x[t], v[t]\right]^T$.
- This time we consider $x[t]$ to be the relative distance to leading vehicle.
- Illustration:



- Dynamic equation

$$x[t+1] = x[t] + v_{lead}[t]\delta - v[t]\delta.$$

- We want x to be close to the reference distance d

Vehicle following

- We can approximate the vehicle following problem as a trajectory tracking problem.
- Suppose trajectory of leading vehicle
$$x_{lead}[0], x_{lead}[1], x_{lead}[2], \ldots$$
- We want the ego vehicle (i.e., the one we control) to track the following trajectory:
$$x_{lead}[0] - d, x_{lead}[1] - d, x_{lead}[2] - d, \ldots$$
- The reference speed will be
$$v_{lead}[0], v_{lead}[1], v_{lead}[2], \ldots$$
- The transformed state will be
$$\begin{bmatrix} \tilde{x}[t] \\ \tilde{v}[t] \end{bmatrix} = \begin{bmatrix} x[t] - x_{lead}[t] + d \\ v[t] - v_{lead}[t] \end{bmatrix}.$$

Vehicle following

- The dynamics is more complex than (uniform) trajectory tracking:

$$\begin{bmatrix}\tilde{x}[t+1]\\ \tilde{v}[t+1]\end{bmatrix} = \begin{bmatrix}x[t+1]-x_{lead}[t+1]+d\\ v[t+1]-v_{lead}[t+1]\end{bmatrix}$$
$$= \begin{bmatrix}x[t]+v[t]\delta-x_{lead}[t]-v_{lead}[t]\delta+d\\ v[t]+u[t]\delta-v_{lead}[t]-u_{lead}[t]\end{bmatrix}$$
$$= \begin{bmatrix}\tilde{x}[t]+\tilde{v}[t]\delta\\ \tilde{v}[t]+(u[t]-u_{lead}[t])\delta\end{bmatrix}.$$

- Hence, vehicle following is analogous to trajectory tracking, except that the control input is shifted by $u_{lead}[t]$.
- If we set $\tilde{u}[t]=u[t]-u_{lead}[t]$, the tilde system appears to be a trajectory tracking problem.

Vehicle following

- Then we can select the control to the tilde system to be $\tilde{u} = [k_1 \quad k_2] \begin{bmatrix} \tilde{x} \\ \tilde{v} \end{bmatrix}$ (based on the eigenvalue argument.)
- The actual control policy will be
$$\mu(\tilde{x}, \tilde{v}; t) = [k_1 \quad k_2] \begin{bmatrix} \tilde{x} \\ \tilde{v} \end{bmatrix} + u_{lead}[t].$$
- Note that the controller is time-varying!
- The reason is that the reference trajectory results from motion of the leading vehicle.
- We cannot perfectly predict trajectory of the leading vehicle.
- Instead, we can only observe it in real time.
- Hence, we need to adjust the controller in real time

Vehicle following

- Alternative objective for vehicle following:
- The distance to leading vehicle satisfying

$$x[t] = d + \beta v[t].$$

- That is, longer distance on freeways than on streets.

Vehicle following

- How to evaluate a vehicle following policy?
- Note: performance depends leading vehicle!
- Typical evaluation methodology:

1. Assume uniform motion for leading vehicle.
2. Assume uniform motion with noise for leading vehicle.
3. Assume bounded speed range for leading vehicle.
4. Assume bounded acceleration for leading vehicle.
5. Assume human driver model for leading vehicle.

Mini project 1

- Step 1: search the internet and make your best guess for the parameters involved in

$$\dot{v} = -\frac{\rho C_d}{2M_t}v^2 + \frac{T_e - R_g T_b}{M_t} - \frac{C_r m g}{M_t},$$

$$M_t = \frac{(mR^2 + I_w)R_g}{R}.$$

- Step 2a: linearized the system at the origin and design that tracks a uniform trajectory in the face of additive Gaussian noise

$$v[t+1] = v[t] + u[t]\delta + w[t].$$

- Step 2b: use the nonlinear model to design a controller.
- Step 3: plot the reference and actual trajectories.

Outline

- Longitudinal control
  - Dynamical equation
  - State-space model
- Synthesis with tracking & following algorithms
  - Trajectory tracking
  - Vehicle following
- Additional issues
  - Linearization
  - Saturation
  - Noise & perturbation
  - Model identification
  - Human driver behavior

Linearization

- Recall that we had a model in the form
$$\dot{x} = ax^2 - 2a\bar{v}x - u + (a\bar{v}^2 + c)$$
- 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 relative speed x is not far away from the equilibrium speed 0.
$$\begin{aligned}\dot{x} &\approx \frac{\partial}{\partial x}\left[ax^2 - 2a\bar{v}x - u + (a\bar{v}^2 + c)\right]_{\substack{x=0\\u=0}} x \\ &+ \frac{\partial}{\partial u}\left[ax^2 - 2a\bar{v}x - u + (a\bar{v}^2 + c)\right]_{\substack{x=0\\u=0}} u + (a\bar{v}^2 + c) \\ &= -2a\bar{v}x - u + (a\bar{v}^2 + c)\end{aligned}$$
- 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

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

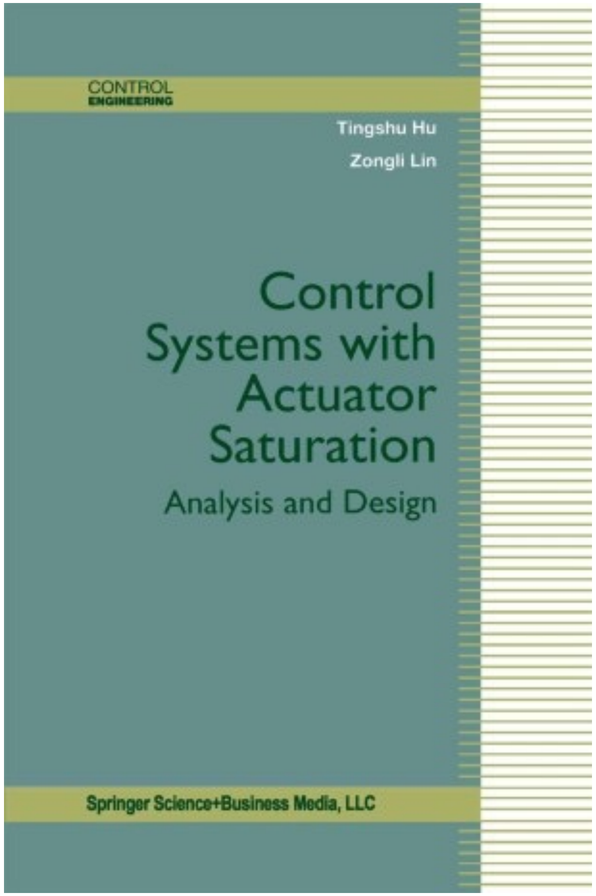
is very well developed and validated in practice.
- Therefore, in many cases, linearization suffices; no nonlinear control has to be done…

Saturation

- Recall that the torque (control input) for speed tracking is given by
$$\mu(x) = ax^2 + (1 - 2a\bar{v})x + (a\bar{v}^2 + c).$$
- 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 for engine torque.
- If $\mu(x) \le b_1\bar{T}_e$, we are good.
- If $\mu(x) > b_1\bar{T}_e$, 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.)



Noise & perturbations

- Recall the dynamic equation is

$$\dot{x} = ax^2 - 2a\bar{v}x - u + (a\bar{v}^2 + c)$$

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

$$\dot{x} = ax^2 - 2a\bar{v}x - u + (a\bar{v}^2 + c) + w$$

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

Noise & perturbations

$$\dot{x} = ax^2 - 2a\bar{v}x - u + (a\bar{v}^2 + c) + w$$

- Control synthesis is more challenging, since everything becomes random.
- Essentially, we are no longer able to exactly tune $x(t)$, since the relative speed now becomes a stochastic process $\{X(t); t \geq 0\}$!
- Instead, we can tune its probability distribution!
- That is, we can specify the cumulative distribution function (CDF) $F_X(x, t)$.
- 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 $(\bar{v}(t)-\delta, \bar{v}(t)+\delta]$ is given by
$$\begin{aligned}&\Pr\{V(t)\in(\bar{v}(t)-\delta,\bar{v}(t)+\delta]\}\\&=F_V(\bar{v}(t)+\delta,t)-F_V(\bar{v}(t)-\delta,t)\end{aligned}$$
  - The variance of the tracking error converges to 0, i.e.,
$$\lim_{t\to\infty}\mathrm{E}\left[\left(V(t)-\bar{v}(t)\right)^2\right]=0.$$
- 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.

Noise & perturbations

*Formulation of noise is **significantly simpler** in DT than in CT.*

- **DT**: add a noise term to the dynamical equation:
  $$x[t+1] = x[t] + \left(ax^2[t] - 2a\bar{v}x[t] - u[t] + (a\bar{v}^2 + c)\right)\delta + w[t].$$
- $w[t]$ follows, say, Gaussian distribution.
- Also, $w[0], w[1]. w[2], \ldots$ are. independent and identically distributed (IID).
- Also easy to simulate in Python/MATLAB/C.
- **[Not required] CT**: $w(t)$ is a Wiener process...

Model identification

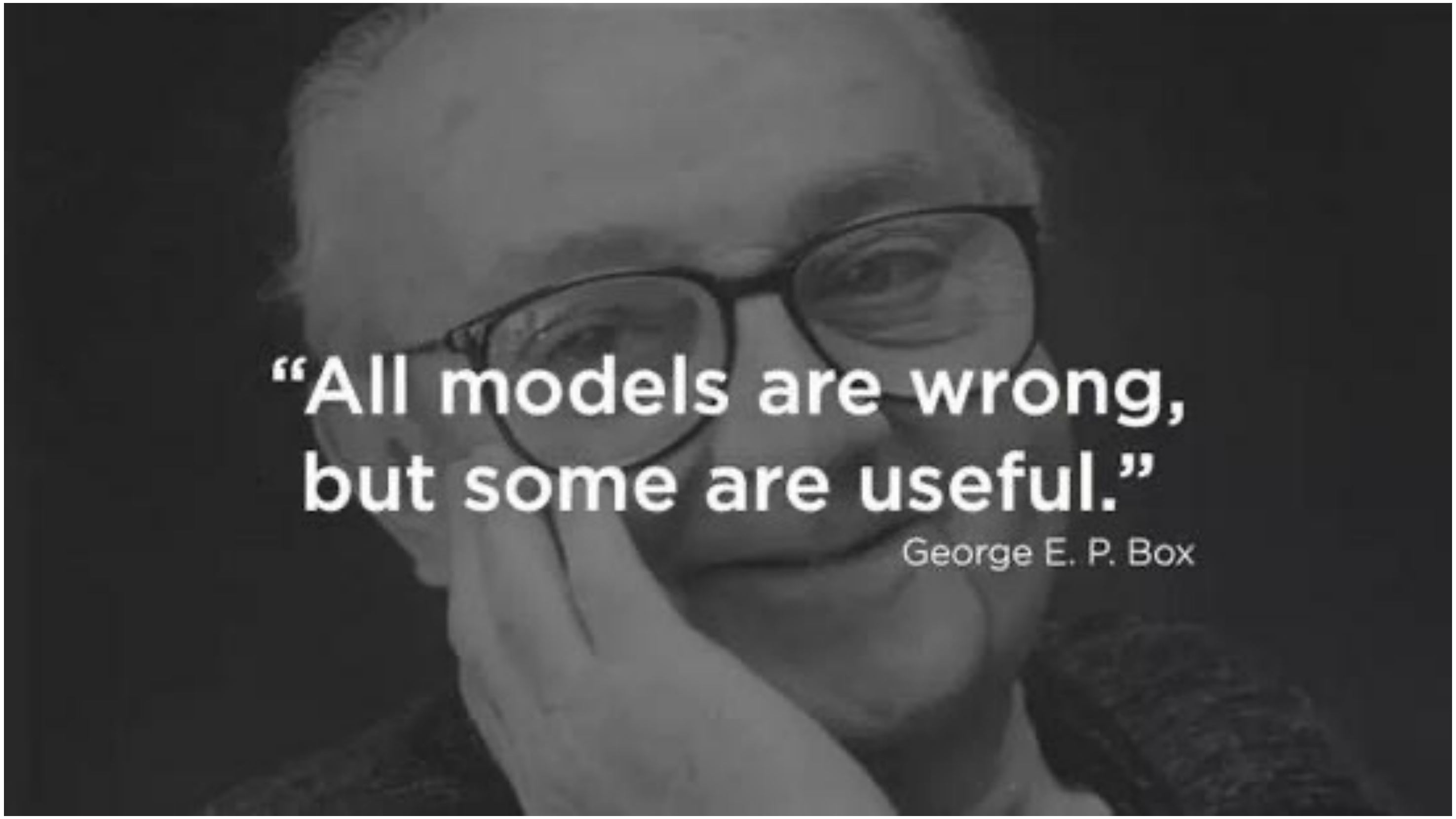
- Recall the dynamic equation

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

- 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

Yet another piece of philosophy...



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: machine 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

Final project option 1

- Based on mini project 1.
- Controls a three-vehicle platoon.
- Addresses at least 3 out 5 problems listed above...

Summary

- Longitudinal control
  - Dynamical equation
  - State-space model
- Synthesis with tracking & following algorithms
  - Trajectory tracking
  - Vehicle following
- Additional issues
  - Linearization
  - Saturation
  - Noise & perturbation
  - Model identification
  - Human driver behavior