3. Autonomous Driving: Trajectory Tracking

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

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Recap

- Concept of autonomous driving
  - How human drive
  - How a computer drives
  - Decision-making architecture
- Speed tracking
  - State, dynamics, action, policy
  - Performance of policy
- 1D dynamical system & control
  - Modeling
  - Objective
  - Theory*

Outline

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

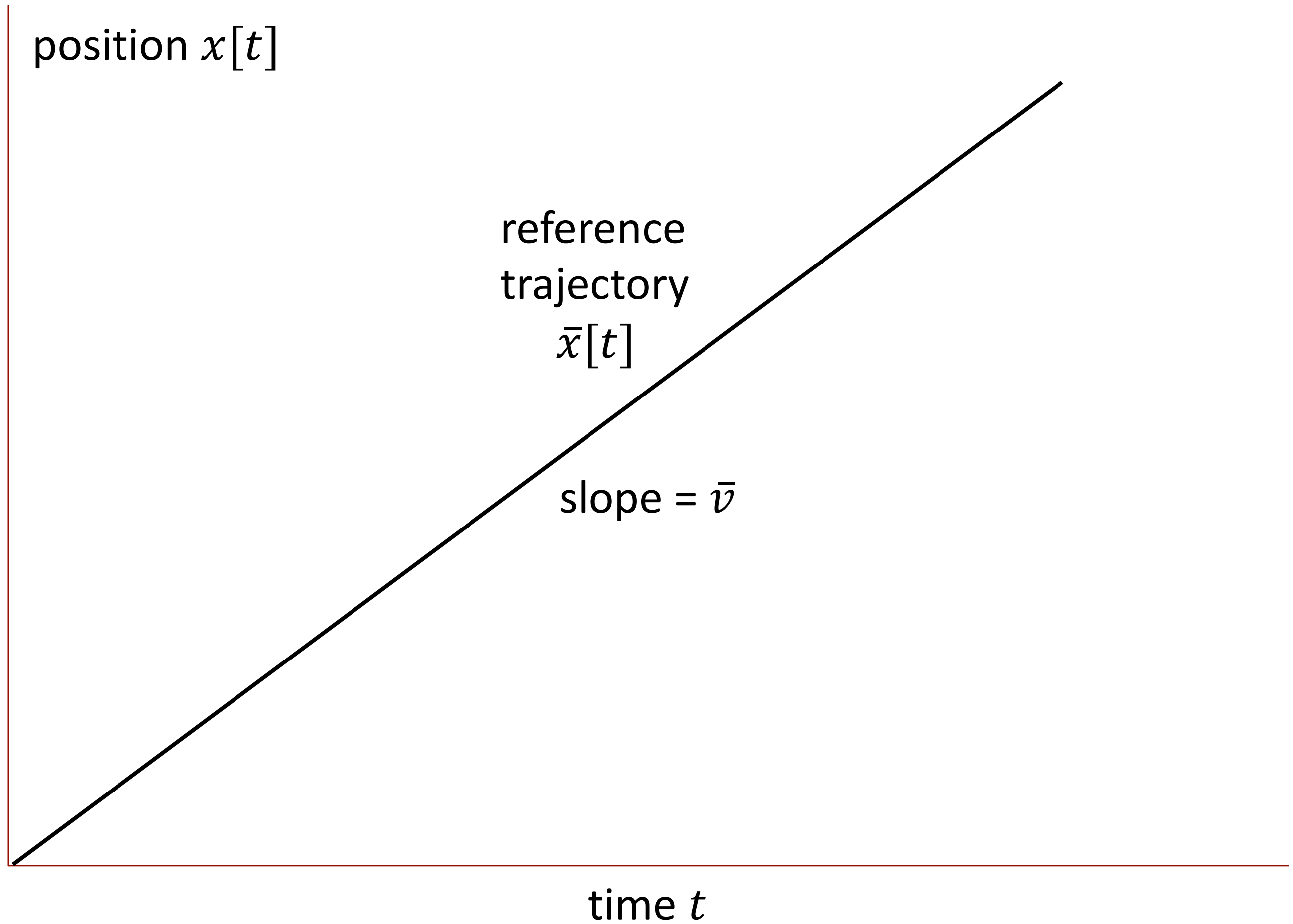
Trajectory tracking

- We have a trajectory for a vehicle to follow
- The vehicle should cover the trajectory with minimal cost
- Typically, cost = time + fuel
- Decision variables: engine torque
- Objective: asymptotic convergence
- Constraints: vehicle dynamics & kinematics

Trajectory tracking

- Consider a single vehicle on a one-dimensional road.
- Suppose that the vehicle is at the starting point of the road.
- Its trajectory is characterized by its position at every time instant.
- We specify a target or reference trajectory $\bar{x}[t]$ for $t = 0,1,2, \ldots$ (discrete time, DT)
- A simple example is uniform motion with
$$\bar{x}[t] = \bar{v}t, \qquad t = 0,1,2, \ldots$$
- Now we want the vehicle to track this reference trajectory.

Trajectory tracking

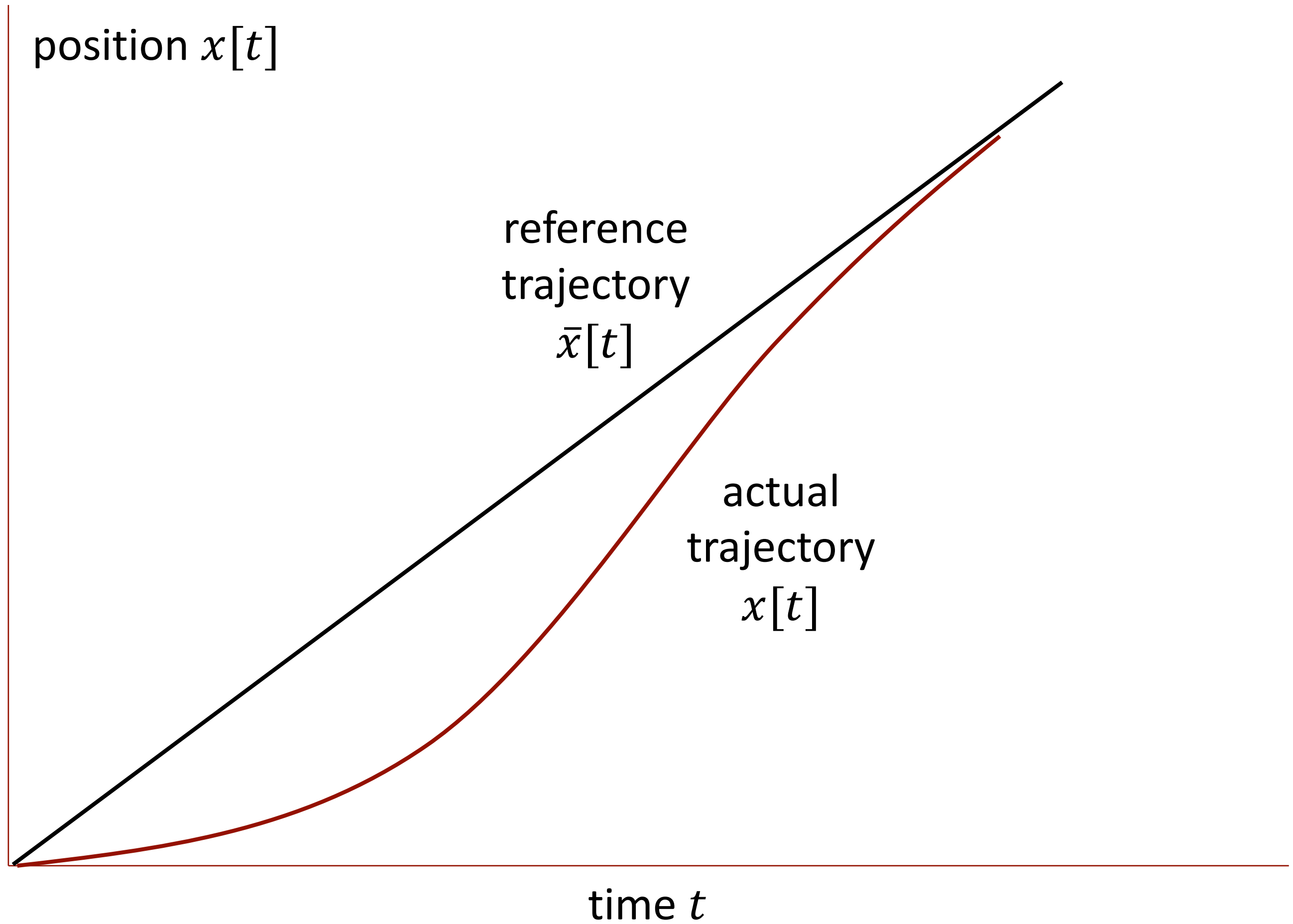


State variable

- What do we exactly mean by "tracking"?
- Typically, this means asymptotic convergence.
- Mathematically, let $x[t]$ be the actual position at time t.
- Important: don't confuse $x[t]$ with $\bar{x}[t]$!
- We call $x[t]$ a state variable of the control problem.
- What's the use of a state variable? Describe and predict the evolution of a dynamical system.
- Here dynamical system = the vehicle.
- Then, tracking or asymptotic convergence means

$$\lim_{t \to \infty} |x[t] - \bar{x}[t]| = 0.$$

Trajectory tracking



State

- Is the position $x[t]$ sufficient for us to describe or predict the motion of the vehicle?
- No! We also need $v[t]$, i.e. the speed, for the above purposes.
- Therefore, the state of the system is a two-dimensional vector $\left[x[t], v[t]\right]^T$.
- You can show that if $x[t] \to \bar{x}[t]$, then we must have $v[t] \to \bar{v}$, again in an asymptotic sense.
- Therefore, we (roughly) say that the system is convergent (or the tracking is successful) if

$$\begin{bmatrix} x[t] \\ v[t] \end{bmatrix} \to \begin{bmatrix} \bar{x}[t] \\ \bar{v} \end{bmatrix}.$$

State

A philosophical definition for state:

- A set of information that, along with all the future control inputs, will determine the future evolution of the system.

For trajectory tracking:

- **Position** alone is not sufficient to be the state.
- **Position** & **speed** are sufficient to be the state; this turns out to be the minimal state representation.
- **Position** & **speed** & **acceleration** are sufficient to be the speed, but acceleration is redundant.

Dynamical equation

- Now we are ready to specify how the system evolves.
- State vector $\left[x[t], v[t]\right]^T$.
- Control input $u[t]$ (**same as speed tracking**)
- System dynamics:

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

- δ = discrete time step size.
- This is the DT, state-space model for a vehicle.
- Since we are restricted to linear motion, this is called longitudinal control.

More on state

- Standard form:

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

- Two implications:

1. As long as we know $\begin{bmatrix} x[t] \\ v[t] \end{bmatrix}$ and $u[t]$, we can fully predict $\begin{bmatrix} x[t+1] \\ v[t+1] \end{bmatrix}$. If we further know $u[t+1], u[t+2], \ldots$, we can fully predict all future states;
2. As long as we know $\begin{bmatrix} x[t] \\ v[t] \end{bmatrix}$, any previous history does not matter; i.e., all previous history is captured by the state.

Linear system

- System dynamics
$$\begin{bmatrix} x[t+1] \\ v[t+1] \end{bmatrix} = \begin{bmatrix} 1 & \delta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x[t] \\ v[t] \end{bmatrix} + \begin{bmatrix} 0 \\ \delta \end{bmatrix} u[t].$$
- The above system is linear, since the new state linear depends on the old state and on the control input.
- Linear systems are the most important type of systems in control theory.
- So linear algebra is the most important tool in control theory.
- The theory of linear system is comprehensive and extensive.

Control problem

- With the state-space model, we can formulate the trajectory tracking problem as follows:
- Given reference trajectory $\bar{x}[t]$ and initial condition $x[0], v[0]$,
- Find $u[0], u[1], u[2], \ldots$ such that
$$\begin{bmatrix} x[t] \\ v[t] \end{bmatrix} \to \begin{bmatrix} \bar{x}[t] \\ \bar{v} \end{bmatrix}.$$
- That is, select the time series of control inputs (acceleration) so that the reference trajectory is tracked.

Control policy

- There are two ways of selecting the control input $u[t]$.

1. Specify $u[t]$ as a function of time t, e.g., $u[t] = a^t$.
2. Specify $u[t]$ as a function of the state $\left[x[t], v[t]\right]^T$, e.g., $u[t] = k_0 + k_1 x[t] + k_2 v[t]$.

- The first way is feasible only in very special and peculiar cases; in general it is not easy.
- The second way is more popular, since we only need to specify the **mapping** from state to control

$$\mu: \mathbb{R}^2_{\geq 0} \to \mathbb{R},$$
$$\mu: [x, v]^T \mapsto u.$$

- Once the mapping is determined, we can obtain $u[t]$ via $u[t] = \mu\left(\left[x[t], v[t]\right]^T\right)$.

Control policy

- This mapping is called the control policy.
- Also called control law or controller or simply policy.
- Typically, the mapping is a function.
- (You may want to recall the difference between a mapping and a function).
- Every one is supposed to know the following:

1. **A control policy is a function;**
2. **This function maps a state to a control input.**

- I will ask this question in the first quiz.

Open-loop vs. closed-loop

- Recall the two ways of selecting control input.

Specify $u[t]$ as a function of time t.

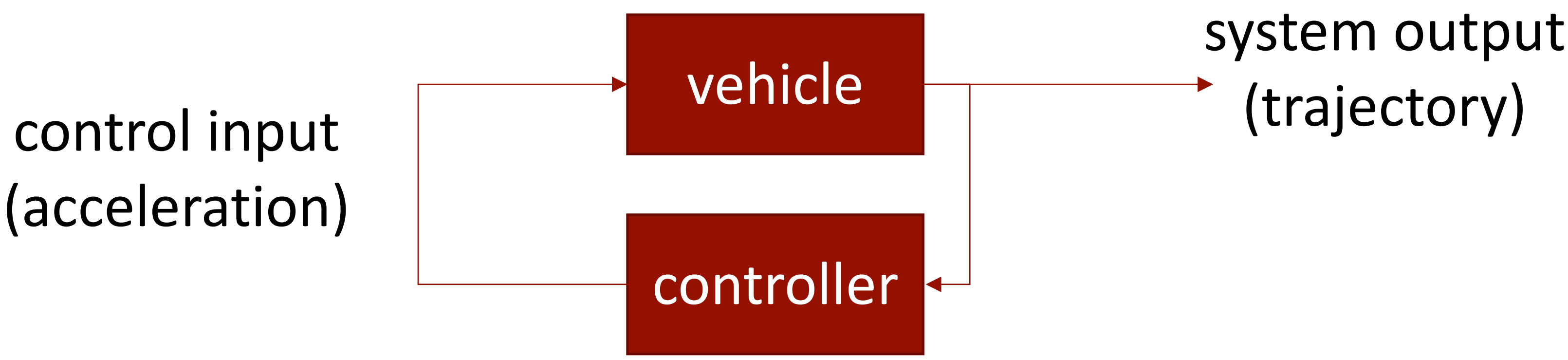
- In the context of vehicle control, this means you program a schedule of accelerations at every time step and send it to the vehicle.
- No more intervention after the schedule is sent.
- Send the instruction and let it go.
- This is called open-loop control.

Open-loop vs. closed-loop

- Recall the two ways of selecting control input.

Specify $u[t]$ as a function of state $[x[t], v[t]]^T$.

- In the context of vehicle control, this means that at every time step you select the acceleration according to the current position and speed.
- Persistent intervention as time goes.
- This is called closed-loop control.



Linear feedback control

- Simplest feedback control: linear feedback
- Suppose that we determine the acceleration via
$$u[t] = -k_1(x[t] - \bar{x}[t]) - k_2(v[t] - \bar{v}).$$
- k_1, k_2 are positive coefficients.
- This is a linear controller.
- What it does:

1. If $x[t] - \bar{x}[t] > 0$, i.e., if the vehicle is ahead the reference trajectory, it should slow down;
2. If $v[t] - \bar{v} > 0$, i.e., if the vehicle is faster than the reference speed, it should slow down.
3. Larger deviation -> larger control input

Closed-loop dynamics

- With the linear controller, we can formulate the dynamics of the closed-loop system as follows

$$\begin{bmatrix} x[t+1] \\ v[t+1] \end{bmatrix} = \begin{bmatrix} 1 & \delta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x[t] \\ v[t] \end{bmatrix} + \begin{bmatrix} 0 \\ \delta \end{bmatrix} \begin{bmatrix} -k_1 & -k_2 \end{bmatrix} \left(\begin{bmatrix} x[t] \\ v[t] \end{bmatrix} - \begin{bmatrix} \bar{x}[t] \\ \bar{v} \end{bmatrix} \right)$$
$$= \left(\begin{bmatrix} 1 & \delta \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 \\ \delta \end{bmatrix} \begin{bmatrix} -k_1 & -k_2 \end{bmatrix} \right) \begin{bmatrix} x[t] \\ v[t] \end{bmatrix}$$
$$+ \begin{bmatrix} 0 \\ \delta \end{bmatrix} \begin{bmatrix} -k_1 & -k_2 \end{bmatrix} \begin{bmatrix} \bar{x}[t] \\ \bar{v} \end{bmatrix}.$$

- Black notations: state variables
- Brown notations: parameters or data
- The problem is now how to select k_1, k_2!

Closed-loop control

- Since feedback control focuses on the deviation between actual and reference trajectories, we reformulate the model as follows.
- Tracking errors as states:

$$\tilde{x}[t] = x[t] - \bar{x}[t],$$
$$\tilde{v}[t] = v[t] - \bar{v}.$$

- Then, we have

$$\begin{aligned}
&\tilde{x}[t+1] = x[t+1] - \bar{x}[t+1] \\
&= (x[t] + v[t]\delta) - (\bar{x}[t] + \bar{v}\delta) \\
&= (x[t] - \bar{x}[t]) + (v[t] - \bar{v})\delta = \tilde{x}[t] + \tilde{v}[t]\delta, \\
&\tilde{v}[t+1] = v[t+1] - \bar{v} = (v[t] + u[t]\delta) - \bar{v} \\
&= (v[t] - \bar{v}) + u[t]\delta = \tilde{v}[t] + u[t]\delta.
\end{aligned}$$

Closed-loop control

- Hence, the system is convergent if

$$\begin{bmatrix} \tilde{x}[t] \\ \tilde{v}[t] \end{bmatrix} \to \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

- With the linear controller $u[t] = -k_1(x[t] - \bar{x}[t]) - k_2(v[t] - \bar{v})$, we have

$$\begin{bmatrix} \tilde{x}[t+1] \\ \tilde{v}[t+1] \end{bmatrix} = \begin{bmatrix} 1 & \delta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x}[t] \\ \tilde{v}[t] \end{bmatrix} + \begin{bmatrix} 0 \\ \delta \end{bmatrix} \begin{bmatrix} -k_1 & -k_2 \end{bmatrix} \begin{bmatrix} \tilde{x}[t] \\ \tilde{v}[t] \end{bmatrix}$$

$$= \begin{bmatrix} 1 & \delta \\ -k_1\delta & 1 - k_2\,\delta \end{bmatrix} \begin{bmatrix} \tilde{x}[t] \\ \tilde{v}[t] \end{bmatrix}.$$

- Hence, we have

$$\begin{bmatrix} \tilde{x}[t] \\ \tilde{v}[t] \end{bmatrix} = \begin{bmatrix} 1 & \delta \\ -k_1\delta & 1 - k_2\,\delta \end{bmatrix}^t \begin{bmatrix} \tilde{x}[0] \\ \tilde{v}[0] \end{bmatrix}$$

initial condition

Closed-loop control

$$\begin{bmatrix} \tilde{x}[t] \\ \tilde{v}[t] \end{bmatrix} = \begin{bmatrix} 1 & \delta \\ -k_1\delta & 1-k_2\,\delta \end{bmatrix}^t \begin{bmatrix} \tilde{x}[0] \\ \tilde{v}[0] \end{bmatrix}$$

- The above formula clearly indicates that the system is convergent if and only if

$$\lim_{t\to\infty} \begin{bmatrix} 1 & \delta \\ -k_1\delta & 1-k_2\,\delta \end{bmatrix}^t = 0.$$

- Therefore, we should select k_1, k_2 such that the above holds.
- [**Not required**] Recall from linear algebra: for a square matrix A, $\lim_{k\to\infty} A^k = 0$ if and only if the magnitude of every eigenvalue of A is less than 1.

Control design

- In conclusion, you can select k_1, k_2 such that the eigenvalues of $\begin{bmatrix} 1 & \delta \\ -k_1\delta & 1-k_2\,\delta \end{bmatrix}$ all have magnitudes less than one.
- The selection of k_1, k_2 is called control design.
- The above procedures will lead to a stabilizing controller, i.e., one that ensures convergence to the reference trajectories.
- This controller also restricts the impact of the noise term $w[t]$.
- There are infinitely many stabilizing controllers.
- To select the “best” one, we need more advanced tools.

DT vs. CT

- Recall the DT dynamics
$$\begin{bmatrix} x[t+1] \\ v[t+1] \end{bmatrix} = \begin{bmatrix} 1 & \delta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x[t] \\ v[t] \end{bmatrix} + \begin{bmatrix} 0 \\ \delta \end{bmatrix} u[t].$$
- We can also formulate the problem in continuous time (CT):
$$\begin{bmatrix} \dot{x}(t) \\ \dot{v}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t).$$
- Notation: $\dot{x}(t) = \frac{d}{dt} x(t)$.
- CT model is easier to treat mathematically and is usually used for theoretical analysis.
- Computer simulation always uses DT model.

From CT to DT

- How to discretize a CT model to obtain a DT model?
- Consider a CT model

$$\dot{y} = Ay + Bu.$$

- Consider a small time increment δ:

$$\frac{y(t + \delta) - y(t)}{\delta} \approx Ay(t) + Bu(t).$$

- Rearrangement leads to

$$y(t + \delta) \approx (A\delta + I)y(t) + B\delta u(t).$$

- I is the identity matrix of appropriate dimension.
- So the corresponding DT model is

$$y[t + 1] = (A\delta + I)y[t] + B\delta u[t].$$

Evaluating a policy

How do we know whether a policy is satisfactory or not?

- **Primary objective**: the actual trajectory asymptotically converges to the reference trajectory:

$$\lim_{t\to\infty}\begin{bmatrix}x[t]\\ v[t]\end{bmatrix}=\begin{bmatrix}\bar{x}[t]\\ \bar{v}\end{bmatrix}.$$

- Feedback control can also attain the following: if the vehicle is deviated from the reference trajectory, it can be steered back to the reference trajectory.
- **Secondary objective**: minimize a cost of interest.

Fuel: $\sum_t v^2[t]$. Comfort: $\sum_t u^2[t]$. Time: $\sum_t \mathbb{I}\{x[t]<L\}$.

($\mathbb{I}$ = indicator function, L = road section length.)

Summary of trajectory tracking

- State variable: $x[t], v[t]$
- State space: $\mathbb{R}^2$
- Control input: $u[t]$; input space = $\mathbb{R}$
- Dynamical equation
$$\begin{bmatrix} x[t+1] \\ v[t+1] \end{bmatrix} = \begin{bmatrix} 1 & \delta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x[t] \\ v[t] \end{bmatrix} + \begin{bmatrix} 0 \\ \delta \end{bmatrix} u[t].$$
- Control objective: asymptotic convergence
- Control policy:
$$\mu(x, v) = -k_1(x[t] - \bar{x}[t]) - k_2(v[t] - \bar{v}).$$
- Control design task: select k_1, k_2.

Simulation

Initialize:

- Data & parameters: $\bar{x}[t]$, $\bar{v}$, δ.
- Initial condition: $x[0], v[0]$.
- Design parameters: k_1, k_2.

Iterate:

- $u[t] = \begin{bmatrix} -k_1 & -k_2 \end{bmatrix} \begin{bmatrix} \tilde{x}[t] \\ \tilde{v}[t] \end{bmatrix}$,
- $\begin{bmatrix} x[t+1] \\ v[t+1] \end{bmatrix} = \begin{bmatrix} 1 & \delta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x[t] \\ v[t] \end{bmatrix} + \begin{bmatrix} 0 \\ \delta \end{bmatrix} u[t]$.

Terminate:

- If $|x[t] - \bar{x}[t]| < \epsilon$ for $T - S, T - S + 1, \dots, T$.

Outline

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

State & control

- Consider a DT n-dimensional dynamical system.
- State variable: x. (Be careful; don't confuse with previous notation.)
- State space: $\mathbb{R}^n$.
- $x[t] \in \mathbb{R}^n$ for $t = 0,1,2, \ldots$
- Control input: u.
- Set of inputs: $\mathbb{R}^m$.
- $u[t] \in \mathbb{R}^m$ for $t = 0,1,2, \ldots$
- Note that $x[t]$ and $u[t]$ are sufficient for predicting future evolution; $x[t-1], x[t-2]$ no longer matter.

Dynamics

- Initial condition: $x[0] \in \mathbb{R}^n$.
- Dynamical equation:
$$x[t+1] = f(x[t], u[t]).$$
- If the function $f: \mathbb{R}^{n+m} \to \mathbb{R}^n$ is linear in its arguments, then the system is linear.
- That is, the system is linear if f takes the form
$$f(x, u) = Ax + Bu + c.$$
- Note that $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}$.
- Typically we assume $c = 0$.
- Then, we have
$$x[t+1] = Ax[t] + Bu[t].$$
- Such a system is called linear time-invariant (LTI).

Non-LTI systems

- A system is not LTI if it is either nonlinear or time-varying.
- If the function $f(x, u)$ is nonlinear in x and u, the system is nonlinear.
- For example, the following system is nonlinear:
$$\begin{bmatrix} x_1[t+1] \\ x_2[t+1] \end{bmatrix} = \begin{bmatrix} x_1^2[t] + x_2[t] \\ u_2[t] \end{bmatrix}.$$
- If the function $f(x, u; t)$ depends on time t, the system is time-varying.
- For example, the following system is time-varying:
$$x[t+1] = (A_0 + A_1 t)x[t] + Bu[t].$$

Control of LTI systems

- Consider an LTI system
$$x[t+1] = Ax[t] + Bu[t].$$
- A typical control objective is to steer the system to a desired state.
- To make math simpler, we usually set the desired state to be 0 (n-dimensional vector of 0's).
- This can be done by shifting the origin of the state space $\mathbb{R}^n$.
- Hence, the control problem is: given initial condition $x[0]$, select $u[t]$ for $t = 0,1,2,\ldots$ such that $x[t]$ converges to 0 in some sense...

Control of LTI systems

- A typical definition for asymptotic convergence:

$$\lim_{t\to\infty} x^T[t]x[t] = 0. \text{ (BE notation)}$$

- We have a name for the above "distance" from the origin:

$$\text{2-norm: } \|x\|_2 = \sqrt{x^T x} = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}.$$

(PhD notation)

- Alternative norms (PhD vs. BE notation):

$$\text{1-norm: } \|x\|_1 = \sum_i |x_i|.$$

$$p\text{-norm: } \|x\|_p = \left(\sum_i |x_i|^p\right)^{1/p}.$$

$$\infty\text{-norm: } \|x\|_\infty = \max_i |x_i|.$$

Feedback control

- Feedback control means to select $u[t]$ according to $x[t]$.
- Mathematically, we look for a function $\mu: \mathbb{R}^m \to \mathbb{R}^n$ such that
$$u[t] := \mu(x[t]), \qquad t = 0,1,2, \ldots$$
- The function μ is called

control policy/control law/controller.

- If the control policy is linear, i.e., if
$$\mu(x) = Kx,$$
the feedback-controlled system is also linear:
$$x[t+1] = Ax[t] + BKx[t] = (A + BK)x[t].$$

CT LTI system

- Sometimes we also consider CT LTI systems:

$$\frac{d}{dt}x(t) = Ax(t) + Bu(t) \text{ or simply } \dot{x} = Ax + Bu.$$

- To obtain the DT counterpart, consider

$$\begin{aligned} x(t+\delta) &= x(t) + \big(Ax(t) + Bu(t)\big)\delta \\ &= (I + A\delta)x(t) + B\delta u(t). \end{aligned}$$

- This is called discretization.
- Key: only $t + \delta$ on the left hand side, only t on the right hand side.
- If a linear controller is applied,

$$\frac{d}{dt}x(t) = Ax(t) + BKx(t).$$

Control design

- Linear feedback controller:
$$\mu(x) = Kx.$$
- Recall that the desired state is 0.
- System dynamics:
$$x[t] = Ax[t] + BKx[t] = (A + BK)x[t].$$
- Thus, we have
$$x[t] = (A + BK)^t x[0], \qquad t = 0,1,2, \dots$$
- Therefore, the system is convergent in the sense that $\lim_{t\to\infty} x[t] = 0$ if and only if
$$\lim_{t\to\infty} (A + BK)^t = 0.$$
- Again, we need to adjust eigenvalues of $A + BK$.

[Not required] Eigendecomposition

- Suppose that $n \times n$ matrix A has n eigenvalues

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|.$$

- Let $\Lambda = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \in \mathbb{C}^{n \times n}$.
- Let $v^{(i)}$ be a right eigenvector of λ_i: $Av^{(i)} = \lambda_i v^{(i)}$.
- Let $V = \left[v^{(1)} \middle| v^{(2)} \middle| \ \ldots v^{(n)}\right]$.
- Thus,

$$\begin{aligned} AV &= \left[Av^{(1)} \middle| Av^{(2)} \middle| \ldots Av^{(n)}\right] \\ &= \left[\lambda_1 v^{(1)} \middle| \lambda_2 v^{(2)} \middle| \ldots \lambda_n v^{(n)}\right] = V\Lambda \\ &\Rightarrow AV = V\Lambda. \end{aligned}$$

[Not required] Eigendecomposition

- If the n eigenvectors are linearly independent, then V is invertible.
- Then we have
$$AVV^{-1} = V\Lambda V^{-1} \Longrightarrow A = V\Lambda V^{-1}.$$
- The above is called eigendecomposition.
- Thus, $A^k = (V\Lambda V^{-1})^k = V\Lambda^k V^{-1}$.
- Note that
$$\Lambda^k = \begin{bmatrix} \lambda_1^k & & \\ & \ddots & \\ & & \lambda_n^k \end{bmatrix}.$$
- $\Lambda^k \to 0$ (and thus $A^k \to 0$) if $|\lambda_1| < 1$.

Summary

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