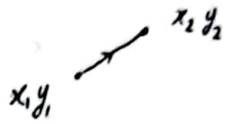
Theme: EEE 494

Date: 05 / 02 / 23

Sat Sun Mon Tue wed Thu Fri

Mobile Robot Kinematics

(Automatic Mobile Robots Ch: 3)



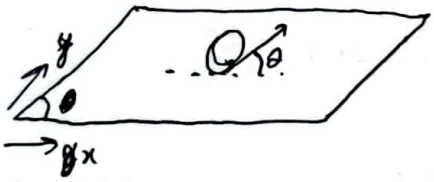
$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

↑ vector norm

distance

- Let $\xi = [\xi_1, \ldots \xi_n]^T$

$\therefore \xi = [x, y, \theta]^T$



$$\xi = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = [x, y, \theta]^T$$

$$\dot{\xi} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$$

Non holonomic ~~constraint~~ Examples

- System: Disk that rolls without slipping

- $\xi = [x, y, \theta]^T$

- No side slip constraint

$$[\dot{x}, \dot{y}] \cdot \begin{bmatrix} \sin\theta \\ -\cos\theta \end{bmatrix}$$

Theme:

Date: / /

Sat Sun Mon Tue wed Thu Fri

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}$$

State space

$\dot{x} = Ax + Bu$

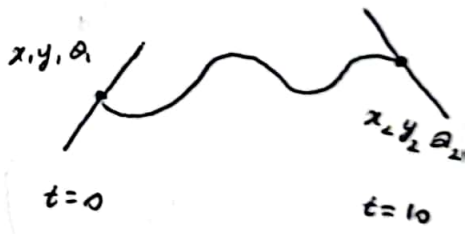
$u = kx$ control

Find k

$$\begin{bmatrix} v \\ \omega \end{bmatrix}_{2\times 1} = \begin{bmatrix} \quad \end{bmatrix} \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_{3\times 1}$$

↑ This has to be 2×3

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 \end{bmatrix} \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$



k values will be updated at every iteration

Theme:

Date: / /

Sat Sun Mon Tue wed Thu Fri

EEE494

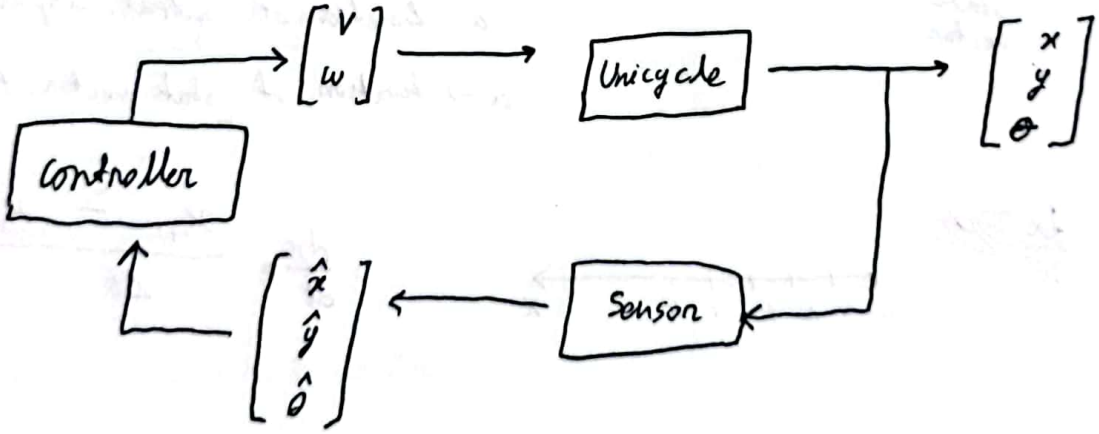
07/02/23

Unicycle model

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}$$

GPS or sensors provide estimation of the state coordinates $\begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{\theta} \end{bmatrix}$

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{\theta} \end{bmatrix}$$



Error Vector

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} - \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{\theta} \end{bmatrix}$$

Date: / /

Theme:

Sat Sun Mon Tue wed Thu Fri

In state estimation, we want $\lim_{t \to \infty} \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = 0$

$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$ ↗ starting / current position

$\begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{\theta} \end{bmatrix}$ ↳ estimation

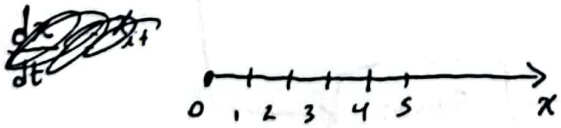
Kinematic / dynamic models

$$\dot{x}(t) = a\left(x(t), u(t), t\right)$$

↗ state vector

↑ a function of output, input, time

a → function of state vector, function of control vector



$$\frac{dx}{dt} = \frac{x_{i+1} - x_i}{\Delta t}$$

x_{i+1} → new position

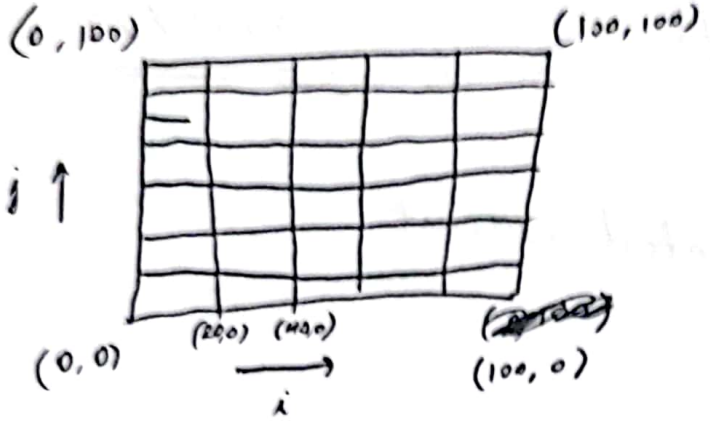
x_i → previous "

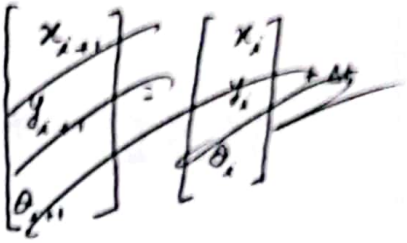
Δt → My sampling space

Theme:

Date: / /

Sat Sun Mon Tue wed Thu Fri





$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \\ \theta_{i+1} \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \\ \theta_i \end{bmatrix} + \Delta t \begin{bmatrix} \cos\theta_i & 0 \\ \sin\theta_i & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_i \\ w_i \end{bmatrix}$$

$\Delta t \rightarrow$ constant

? current — ↑ previous — correction / update

Optimal control problem

$$\min \quad h(x(t_f), t_f) + \int_{t_0}^{t_f} g(x(t), u(t), t)\, dt$$

minimize

- For optimal control → shortest path
  - → smallest time
  - → minimum energy → fuel

Date: / /

Sat Sun Mon Tue wed Thu Fri

Theme:

EEE 494

12/02/23

$$a = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \qquad b = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

$a.b = 4 + 10 + 18 = 26$

$a.b$ can also be found by $a^T \times b = 26$

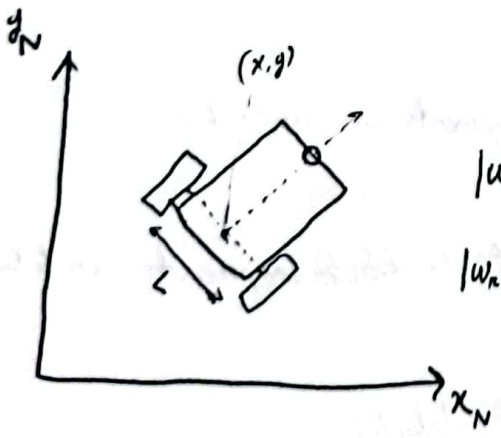
dot product is a linear function

$$f = m_1 2 + m_2 4 + m_3 7$$

m_1 m_2 m_3 are linear functions

then f is a linear function

From slide



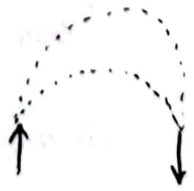
$|w_l| \leq w_l$, max

$|w_r| \leq w_r$, max

$w_l = w_r$ Reward

$w_l > w_r$ right

$w_l < w_r$ left



If car wants to change from 90° to -90° then it can multiple ways to change the angular displacement (U-turn)

Theme:

Date: / /

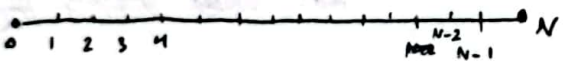
Sat Sun Mon Tue Wed Thu Fri

$$\dot{x}(t) = a(x(t), u(t), t)$$

state vector
1st derivative

function of output vector, input vector

$$\dot{x}(t) = \frac{x_{i+1} - x_i}{\Delta t}$$



$\rightarrow i$

Discritization

step size যত ছোট, তত accurate output পাবো

Now, using a for loop from i=0 to i=N-1 as $x_{i+1} = x_i + \Delta t_i \cdot a(x_i, u_i, t_i)$

Optimal solution

① Optimal solution is not always possible or practical.

② When optimal trajectories are not possible, we use "differential flatness" trajectories.

Date: / /

Sat Sun Mon Tue wed Thu F

$x(t) \in R^n$

The control vector consists of n - dimension

$x(t) \in X$ ← admissible state

$u(t) \in U$ ← admissable control

random করে number নিতে হবে U থেকে

$$\min_u \; h(x(t_f), t_f) + \int_{t_0}^{t_f} g(x(t), u(t), t)\, dt$$

$h(x(t_f), t_f)$ → cost fn

$g(x(t), u(t), t)$ → stage cost

$u^*(t) = \pi(x(t), t)$ → function of state vector → closed loop

$u^*(t) = P(x(t_0), t)$ → open loop → function of initial conditions

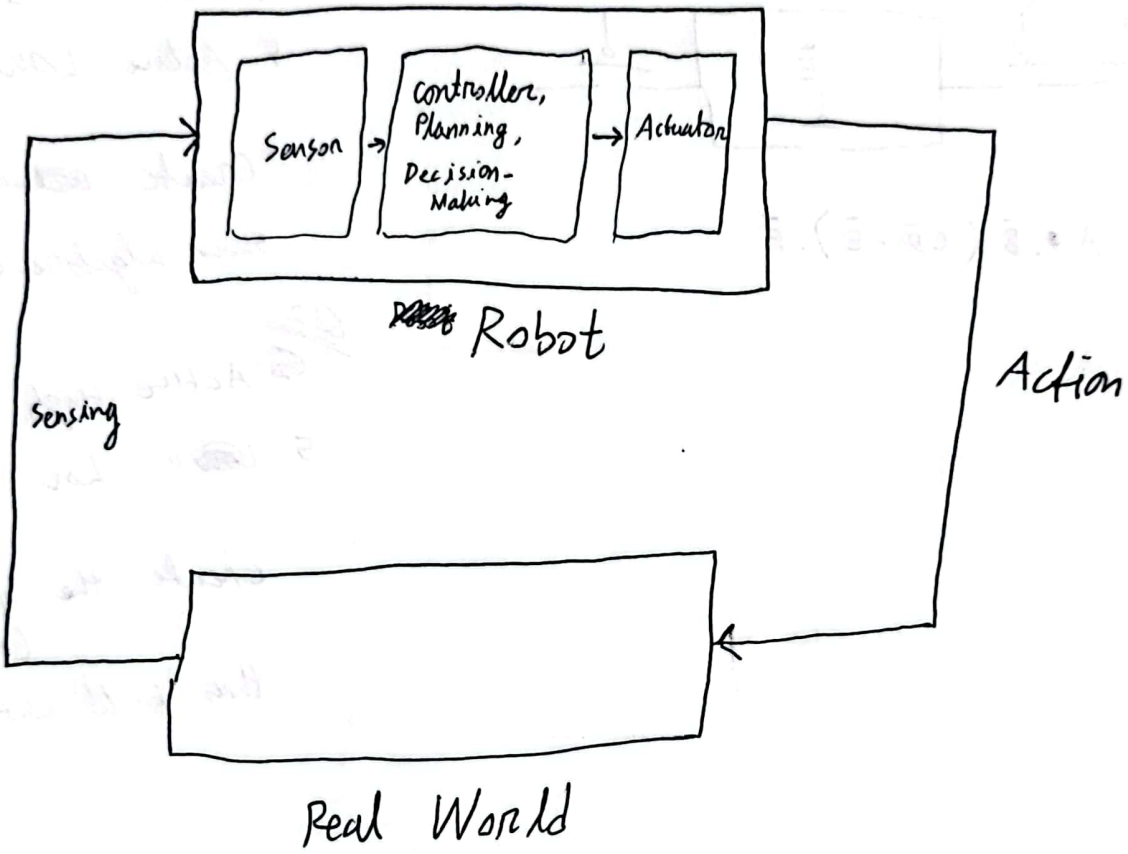
Date:

Theme:

Sat Sun M

EEE 494

19/02/23



See Slide pg: 3

Localization → detect where I am

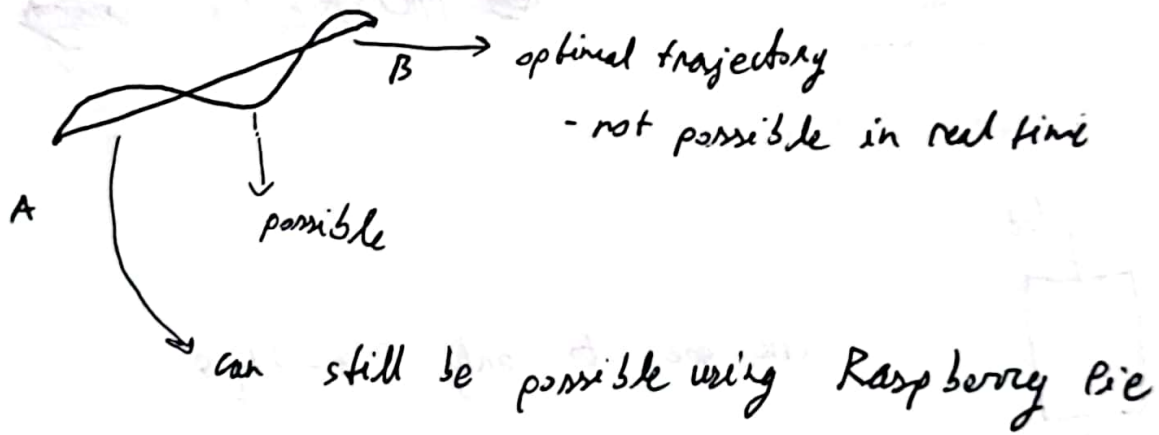
Top Level → set path . Low Level → confused

Springer Handbook → Motion Control

New Position = current position + Update

$$\min \quad h(x(t_f), t_f) + \int_{t_0}^{t_1} g(x(t), u(t), t)\, dt$$

↑ optimise করতে চাই



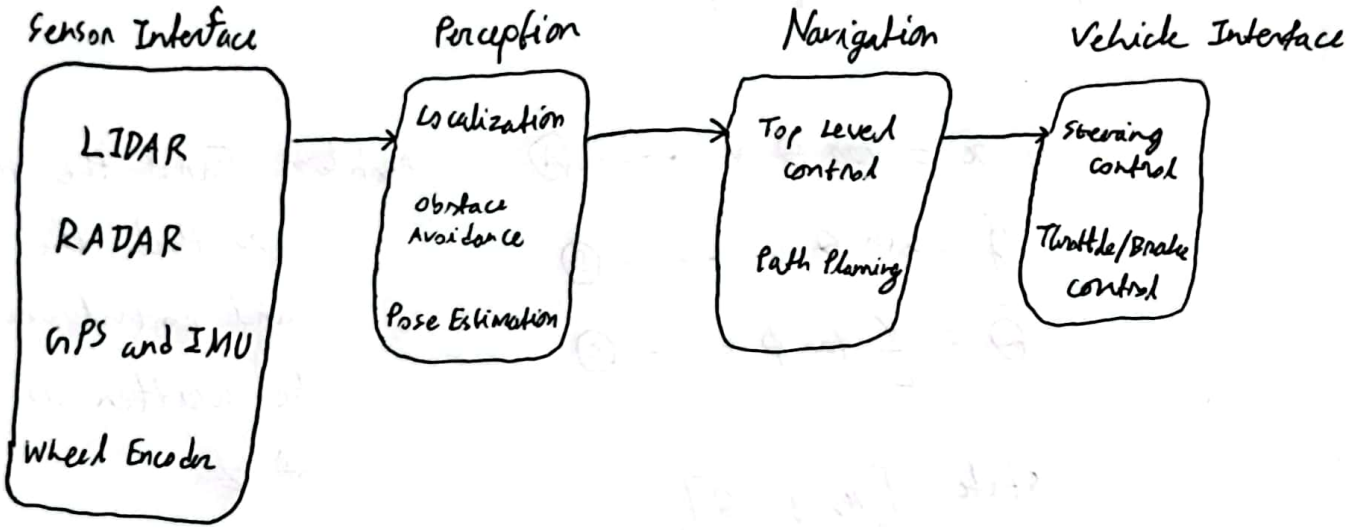
Differential Flatness

Theme: EEE494

Date: / /

Sat Sun Mon Tue wed Thu Fri

26/02/23



Optimal control will give the ideal trajectory but it is computationally ~~ex~~intensive. so we can find feasible trajectory instead.

Differential flatness : There exists a function z such that state$^{\text{vector}}$ can be written as a function of z and input vector can be written as z

∴ If there exists a funtion $z = \alpha(x, u, \ldots u^{(p)})$

Then state vector $x = \beta(z, \dot{z}, \ldots, z^{(q)})$

$u = \gamma(z, \dot{z}, \ldots, z^{(q)})$

Example: simple car Model

$$\dot{x} = \cos\theta\, v \quad \text{---- (A)}$$

$$\dot{y} = \sin\theta\, v \quad \text{---- (B)}$$

$$\dot{\theta} = \frac{v}{L}\tan\phi \quad \text{---- (C)}$$

state: $[x, y, \theta]$

control: $[v, \phi]$

Problem: Find the function z, so that the state vector and control vector can be written as functions of z

Solution: (x, y) denotes the position

let $z = (x, y)$

$\therefore \dot{z} = (\dot{x}, \dot{y})$

$$\frac{(B)}{(A)} = \frac{v\cos\theta}{v\sin\theta} = \frac{\dot{y}}{\dot{x}}$$

$$\Rightarrow \tan\theta = \frac{\dot{y}}{\dot{x}}$$

$$\theta = \tan^{-1}\left(\frac{\dot{y}}{\dot{x}}\right)$$

$$v = \frac{\dot{x}}{\cos\theta}$$

$$\tan\phi = \frac{\dot{\theta}L}{v} \qquad \therefore \phi = \tan^{-1}\left(\frac{\dot{\theta}L}{v}\right)$$

So far,

$z = (x, y)$

state: $[x, y, \theta]$ control: $[v, \phi]$

$\theta = \tan^{-1}\left(\frac{\dot{y}}{\dot{x}}\right)$ $\therefore \theta$ is a function of $\dot{z}$ $[f(\dot{z})]$

$v = \frac{\dot{x}}{\cos\theta}$ - v is a function of $\dot{z}$ $[f(\dot{z})]$

$\phi = \tan^{-1}\left(\frac{\dot{\theta}L}{v}\right)$ ϕ is a function of $\dot{z}$ $[f(\dot{z})]$

So, this is a differentially flat system because of θ and v

$\therefore$ We have now proved that the car model is a differentially flat system.

In 2-D any vector can be written as a combo of the basis vectors $(1,0)$ and $(0,1)$

Date: / /

Sat Sun Mon Tue wed Thu Fri

Theme:

Example

$$\begin{bmatrix} 3 \\ 0 \\ 2 \end{bmatrix} = 3\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + 0\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + 2\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Any vector can be written as a linear combo of basis vector

∴ Any functions can be written as a linear combination of the basis function

∴ Next step,

Write Z as a linear combination of basis functions ψ_i,

$$\therefore Z = \sum_{i=1}^{N} a_i \psi_i$$

N is the number of basis functions

The more amount of basis functions I take, the more accurate (less errors) my function will be.

a_i are the weights (constants)

Theme:

Date: / /

Sat Sun Mon Tue wed Thu Fri

Reason

My z is a function of time but I don't know what z is. So we need to find z using the formula give in the previous page.

One potential choice is to have $\psi_1(t) = 1$ $\psi_2(t) = t$ $\psi_3(t) = t^2$

We know, $Ax = b$ Matrix x vector = vector

A and b is known
matrix vector

$x = A^{-1} b$ [for square matrices]

inverse
↳ pseudo matrix [when not square matrix]

$x = A^{\dagger} b$

↑ A dagger

Because Inverse occurs only for square matrices

or

rank full

↓ rank = no. of rows + no. of columns

Matlab

inv → inverse

pinv → pseudo inverse

Date: / /

Sat Sun Mon Tue wed Thu Fri

N = 4

$$Z = a_1 \psi_1 + a_2 \psi_2 + a_3 \psi_3 + a_4 \psi_4$$

$$Z(0) = a_1 \psi_1(0) + a_2 \psi_2(0) + a_3 \psi_3(0) + a_4 \psi_4(0)$$

$$Z(T) = a_1 \psi_1(T) + a_2 \psi(T) + a_3 \psi(T) + a_4 \psi(T)$$

$$\Downarrow$$

$$\underbrace{\begin{bmatrix} \psi_1(0) & \psi_2(0) & \psi_3(0) & \psi_4(0) \\ \psi_1(0) & \psi_2(0) & \psi_3(0) & \psi_4(0) \\ \psi_1(T) & \psi_2(T) & \psi_3(T) & \psi_4(T) \\ \psi(T) & \psi_2(T) & \psi_3(T) & \psi_4(T) \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} x(0) \\ y(0) \\ x(T) \\ y(T) \end{bmatrix}}_{b}$$

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = A^{+} b$$

Note: A is not a full rank or square matrix

Because 2 rows are same

Date: / /

Theme:

□ Sat □ Sun □ Mon □ Tue □ wed □ Thu

EEE494

28/02/22

Last lecture: we found trajectory

Now what if we have constrains like max speed limit

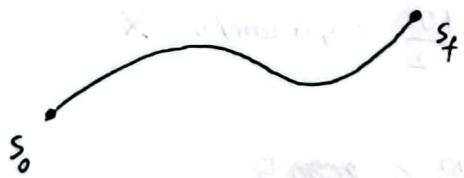
Divide trajectory into 2 components

- Geometrical path
- Timing on the path

$$v = \frac{dx}{dt} = \frac{dx}{ds} \cdot \frac{ds}{dt}$$

$\frac{ds}{dt}$ ← timing

$\frac{dx}{ds}$ ↑ geometrical path



$s(t)$ curve as function of time

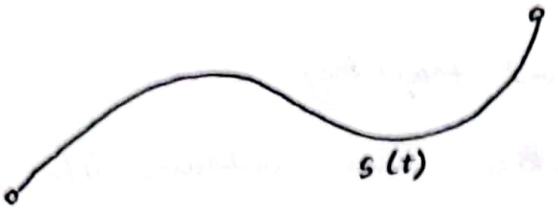
$$s(0) = s_0$$

$$s(t_f) = s_f$$

Theme:

Date: / /

Sat Sun Mon Tue wed Thu Fri



$$s = Vt$$

$$V = \frac{s}{t}$$

Increase t ~~not~~ meet 'velocity constraint'

Ex: Suppose $V_{max} = 5$

$s = 100k$

What is the constraint to meet the V_{max} requirement.

$$V = \frac{dx}{dt} = \frac{dx}{ds} \cdot \frac{ds}{dt}$$ x is the state vector

$$V = \frac{100}{t}$$

$t = 2 \quad \therefore V = \frac{100}{2} = 50 \text{ cm/s} \quad X$

$\therefore$ Requirement $\frac{100}{t} \leq 5$

$100 \leq 5t$

$t \geq 20 \text{ s}$

Theme:

Date: / /

Sat Sun Mon Tue wed Thu Fri

$s_1 = s_0 + \Delta s$

$s_2 = s_1 + \Delta s$

* Differential flatness gives the curve

Found the distance travelled L

If the car were to finish the track in a half a second then the car's velocity will be 200 m/s

∴ we need to place a constraint

Example

Show that the following system is differentially flat

$$\dot{x}_1 = u_1$$

$$\dot{x}_2 = u_2$$

$$\dot{x}_3 = x_2 u_1$$

states : x_1, x_2, x_3

controls : u_1, u_2

Define $z = (x_1, x_2)$

$x_1 = z_1$

$x_3 = z_2$

$u_1 = \dot{x}_1 = \dot{z}_1$

$x_1 = z_1$

$x_3 = z_2$

$$x_2 = \frac{\dot{x}_3}{u_1} = \frac{\dot{x}_3}{\dot{x}_1} = \frac{\dot{z}_2}{\dot{z}_1}$$

$u_1 = \dot{z}_1$

$$u_2 = \frac{\ddot{z}_2}{\ddot{z}_1} \qquad \left[u_2 = \dot{x}_2 \;,\; x_2 = \frac{\dot{z}_2}{\dot{z}_1} = \frac{\ddot{z}_2}{\ddot{z}_1} \right]$$

$\therefore$ the system is differentially flat

Let's say

$$Z_1 = \sum_{i=1}^{4} a_i \psi_i$$

$$\psi_1 = 1 \quad , \psi_2 = t \quad , \quad \psi_3 = t^2 \quad , \quad \psi_4 = t^3$$

$$\therefore Z_1 = a_1 \psi_1 + a_2 \psi_2 + a_3 \psi_3 + a_4 \psi_4$$

Theme:

Date: / /

Sat Sun Mon Tue wed Thu Fri

EEE 494

05/03/2023

To find optimal trajectory

$\dot{X} = f(x, u)$

$\min_{u} h(x, u, t)$

Feasible trajectory

$Ax = b$

$x = A^{+} b$ → pseudo matrix

The raspberry Pi can simply find the feasible trajectory by computing the matrix

Note

For $z = (x_2, x_3)$

Not differentially flat

Example: $\dot{x}_1 = u_1$

$\dot{x}_2 = u_2$

$\dot{x}_3 = \dot{x}_2 u_1$

Initial point : X_{10} , X_{20} , X_{30} at $t=0$

Final point : X_{1f} , X_{2f} , X_{3f} at $t = T$

(a) Show the system is differentially flat $z = (x_1, x_3)$

(b) compute a feasible trajectory for this system.

Solution : (a)

$$z = (x_1 , x_3)$$

$$x_1 = z_1$$

$$x_3 = z_2$$

Then,

$$\dot{x}_1 = \dot{z}_1 \quad \Rightarrow u_1 = \dot{z}_1$$

$$\dot{x}_3 = \dot{z}_2$$

$$x_2 = \frac{\dot{x}_3}{u_1} = \frac{\dot{z}_2}{\dot{z}_1} \qquad \therefore x_2 = \frac{\dot{z}_2}{\dot{z}_1}$$

$$u_2 = \dot{x}_2 * [\text{illegible}] = \beta(z, \dot{z}, \ddot{z})$$

∴ All states and controls are functions of $z, \dot{z}, \ddot{z}$. So the system differentially flat for this choice of $z = (x_1 , x_3)$

Theme:

Date: / /

Sat Sun Mon Tue wed Thu Fri

solution (b)

Write z as a combination of basis functions

$$z_1 = \sum_{i=1}^{4} a_{1i}\, \psi_{1i}$$

$$= a_{11}\psi_{11} + a_{12}\psi_{12} + a_{13}c_3 + a_{14}c_{14}$$

Function এর degree যত বেশি, graph তত smooth.

$$z_2 = \sum_{i=1}^{4} a_{2i}\, \psi_{2j} = a_{21}\psi_{21} + a_{31}c_{13} + a_{24}c_{24}$$

$$\therefore z_1 = a_{11}\psi_{11} + a_{12}\psi_{12} + a_{13}\psi_{13} + a_{14}\psi_{14}$$

$$\boxed{z_1 = a_{21}1 + a_{22}t + a_{22}t^2 + a_{24}t^3}$$

$$\boxed{z_2 = a_{21} + a_{22}t + a_{23}t^2 + a_{24}t^3}$$

$$\boxed{X_2 = \frac{\dot{z}_2}{\dot{z}_1} = \frac{a_{22} + 2a_{23}t + 3a_{24}t^2}{a_{12} + 2a_{13}t + 3a_4 t^2}}$$

unknown

Known

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & T & T^2 & T^3 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2T & 3T^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T & T^2 & T^3 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2T & 3T^2 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{24} \end{bmatrix} = \begin{bmatrix} z_1(0) \\ \dot{z}_1(0) \\ z_2(0) \\ \dot{z}_2(0) \\ z_1(T) \\ \dot{z}_1(T) \\ z_2(T) \\ \dot{z}_2(T) \end{bmatrix} = \begin{bmatrix} x_{10} \\ 1 \\ x_{30} \\ x_{20} \\ x_{1f} \\ 1 \\ x_{3f} \\ x_{2f} \end{bmatrix}$$

A $\quad$ a $\quad$ = $\quad$ b

For first row

$$x_2 = \frac{\dot{z}_2}{z_1} = \frac{a_{22} + 2a_{23}t + 3a_{24}t^2}{a_{12} + 2a_{13}t + 3a_{14}t^2}$$

$$x_2(0) = \frac{a_{22}}{a_{12}}$$

Set $a_{12} = 1$, $x_2(0) = a_{22}$

For second row

$$\dot{z}_1 = a_{12} + 2a_{13}t + 3a_{14}t^2$$

$$\dot{z}_1(0) = a_{12} = 1$$

For third row

$$z_2 = a_{21} + a_{22}t + a_{23}t^2 + a_{24}t^2$$

$$x_{30} = z_2(0) = a_{21}$$

For fourth row

$$\frac{\dot{z}_2}{\dot{z}_1} = \frac{a_{22} + 2a_{23}t + 3a_{24}t^2}{a_{12} + 2a_{13}t + 3a_{14}t^2}$$

$\dot{z}_1$ ↑ set aside

$$\dot{z}_2(0) = \frac{a_{22}}{a_{12}} = a_{22} \rightarrow x_{20}$$

For fifth row

$$Z_1 = a_{11}\ 1 + a_{12}\ t + a_{13}\ t^2 + a_{14}\ t^3$$

$$Z_1(T) = a_{11}\ 1 + a_{12}\ T + a_{13}\ T^2 + a_{14}\ T^3$$

For sixth row

Let $\dot{Z}_1(T) = 1$

$$\Rightarrow a_{12} + 2a_{13}\ T + 3a_{14}\ T^2 = 1$$

For seventh row

$$Z_2(T) = a_{21} + a_{22}\ T + a_{23}\ T^2 + a_{24}\ T^2$$

For eigth row

$$\dot{Z}_2(T) = X_{2t} = a_{22} + 2a_{23}\ T + 3a_{24}\ T^2$$

$\therefore$ Finally $A\ a = b$

$$a = A^{+}\ b \quad \text{or} \quad A^{-1}\ b$$

Theme: EEE494 07/03/2023 Date: / /

Sat Sun Mon Tue wed Thu Fri

1. Traj. Generation and Tracking
2. Motion Planning
3. Computer Vision
4. Localization

Computer Vision → Pixels

Picture compression

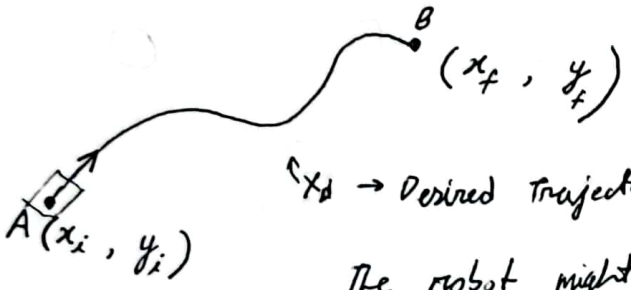
5MB → 1MB

100×100 matrix

100 eigenvalues

I remove 50 eigenvalues

Picture size compresses but becomes blurry



The robot might face disturbances

Disturbances

- sensor noise
- Actuator noise
- External disturbance

X_d = Desired Trajectory → In a perfect world we ge this trajectory with no disturbances

The ..... line represents the True Path → $x(t)$

due to disturbance

covers

$\therefore e(t) =$ [illegible] $x_d(t)$

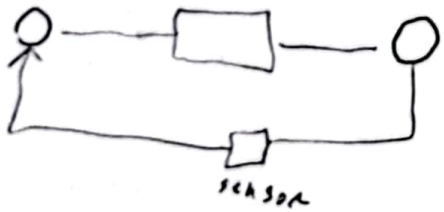
$$\therefore \quad e(t) = x(t) - x_d(t)$$

Modify controls, so that $e(t) \to 0$; $x(t) \cong x_d(t)$

Trajectory tracking controller.

$e(t)$ → error, also called drift

closed loop →



Key note: $\dot{x} = f(x, u)$ ⟶ non-linear system

⟶ Differentially flat ~~system~~ trajectory

Differentially flat trajectory can be linearized

Differentially flat system

↓

Linearize

↓

PD controller → PD controller (Sec 342)

$$w_1 = \ddot{x}_d + k_{px}(x_d - x) + k_{dx}(\dot{x}_d - \dot{x})$$

$$w_2 = \ddot{y}_d + k_{py}(y_d - y) + k_{dy}(\dot{y}_d - \dot{y})$$

x_d and y_d coming from DF system matrix

x and y " " sensors

If k_{px}, k_{py}, k_{dx}, k_{dy} gain are > 0

Then $e(t) \to 0$

Traj. Tracking → PD controller

Now, Posture regulation (how the robot is oriented)

Liaponov based controllers

Given a system $\dot{x} = f(x, u)$,

if there exists a function $V(x, u)$

The system can be stabilized if, (i) $V(x, u) = 0$ for $x = 0$

(ii) $V(x, u) > 0$ for $x \neq 0$

(iii) $\frac{d\,V(x,u)}{dt} < 0$

EEE 494

12/03/2023

[Continuation of Module 1]

From EEE 342

Linear systems

$\dot{x} = Ax + Bu$

$u = kx$

↑ vector of gains

$\dot{x} = Ax + Bkx$

$\Rightarrow \dot{x} = \underbrace{(A + Bk)}_{\text{closed loop poles}} x$

k is controller gains

* Closed loop poles are eigenvalues of $A+Bk$

Pick k so that the eigenvalues of $A+Bk$ are negative

Non-linear systems

$\dot{x} = f(x, u)$ • → $\dot{x} = f(x(t), u(t), t)$

Lyapunov's stability theorem : If there exists a function

(i) $V(x, u)$ such that $V(x, u) = 0$ for $x = 0$

(ii) $V(x, u)$ such that $V(x, u) > 0$ for $x \neq 0$

(iii) $\frac{dV(x,u)}{dt} < 0$ for $x \neq 0$

Lyapunov's Stability Theorem

If there exists a function $v(x,u)$ such that

(i) $v(x,u)=0$ for $x=0$

(ii) $v(x,u)>0$ for $x\neq 0$

(iii) $\frac{dv(x,u)}{dt}<0$ for $x\neq 0$

Then $\dot{x}=f(x,u)$ is stable

Examples v will be given, we will have to see whether the system is stable.

Example: $\dot{x}=-x$

Show this is stable using Lyapunov's Theorem

Solution

Choose $v(x)=x^2$

Example: $\dot{x} = -x_1$

$\dot{x}_2 = -x_2$

Show this system is stable using Lyapunov theorem

$\dot{x}_1 = -x_1$ ∴ state vector: $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

$\dot{x}_2 = -x_2$

Solution: choose $V(x) = x_1^2 + x_2^2$

(i) $V(0) = 0$

(ii) $V(x) > 0$ for $x \neq 0$

(iii) $\frac{dV}{dt} = 2x_1\dot{x}_1 + 2x_2\dot{x}_2$ [Implicit differentiation]

$= [illegible]$

$= 2x_1(-x_1) + 2x_2(-x_2)$ $\begin{bmatrix} \dot{x}_1 = x_1 \\ \dot{x}_2 = x_2 \end{bmatrix}$

$= -2x_1^2 - 2x_2^2$

$V = [x_1(t)]^2 + [x_2(t)]^2$

$= 2x_1(t) \frac{dx_1(t)}{dt} + 2x_2(t) \frac{dx_2(t)}{dt}$

x, y, θ plane এ Lyapunove function নাই

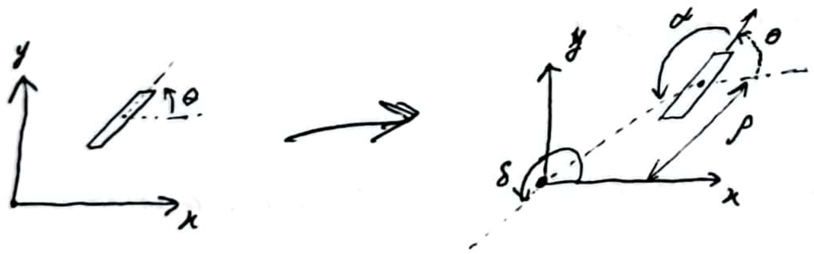
Example: Pose Stabilization

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \longrightarrow \begin{bmatrix} \rho \\ \alpha \\ \delta \end{bmatrix}$$

Co-ordinate transformation is done

Because Lyapunov function does not exist in x, y, θ plane

but it exists in ρ, α, δ.



$\rho = \sqrt{x^2 + y^2}$

$\alpha = a\tan 2(y, x) - \theta + \pi$

$\delta = \alpha + \theta$

$\dot{\rho}(t) = -v(t) \cos \alpha(t)$

$\therefore \dot{\alpha}(t) = \frac{v(t) \sin[\alpha(t)]}{\rho(t)} - \omega(t)$

$\dot{\delta}(t) =$

$\therefore V(\rho, \alpha, \theta) = \frac{1}{2}\rho^2 + \frac{1}{2}(\alpha^2 + k_3 \delta^2)$

$\therefore$ For $V(\rho, \alpha, \delta) = \frac{1}{2}\rho^2 + \frac{1}{2}(\alpha^2 + k_3\delta^2)$

$\therefore$ (i) $V(0,0,0) = 0$ for $\begin{bmatrix}\rho\\ \alpha\\ \delta\end{bmatrix} = \begin{bmatrix}0\\0\\0\end{bmatrix}$

(ii) $V(\rho, \alpha, \delta) > 0$ for $\begin{bmatrix}\rho\\ \alpha\\ \delta\end{bmatrix} > 0$

(iii) $\frac{dV}{dt} = \rho\dot{\rho} + \alpha\dot{\alpha} + k_3\delta\dot{\delta}$

$$= -\rho V\cos\alpha + \alpha\left[\frac{V\sin\alpha}{\rho} - \omega\right] + k_3\delta\left[\frac{V\sin\alpha}{\rho}\right]$$

Let, $V = k_1\rho\cos\alpha$

$$\omega = k_2\alpha + k_1\frac{\sin\alpha\cos\alpha}{\alpha}(\alpha + k_3\delta)$$

For $k_1, k_2, k_3 > 0$

Using the control law above, we can say $\frac{dV}{dt} < 0$.

EEE494

14/03/23

Given an n-order differential equation, we can write it as a system of 1st order differential equations. This is called state-space representation

$\dot{X} = Ax + Bu$ → linear system

$\dot{X} = f(x, u)$ → non-linear system

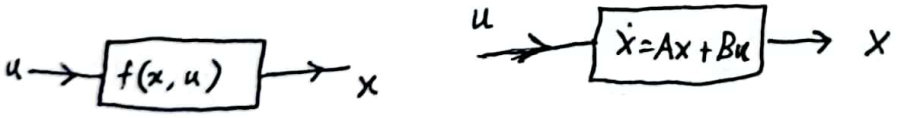
X is the state vector
u is the control vector or input vector

Linear system: A is the $n \times n$ system matrix

B is the $n \times m$ input matrix

↘ $n \times 1$ for single input

Block Diagram representation



Note: An eigenvalue is associated with one eigenvector.

Example: $\dddot{y} + 6\ddot{y} + 11\dot{y} + 6y = u$

Define the state-vector $x = \begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \end{bmatrix}$

solution:

$$\dddot{y} = -6y - 11\dot{y} - 6\ddot{y} + u$$

$n = 3$, $m = 1$ for $x = \begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \end{bmatrix}$

$$\therefore \dot{x} = \begin{bmatrix} \dot{y} \\ \ddot{y} \\ \dddot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6 & -11 & -6 \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u$$

* The poles are defined as the eigenvalues of the system.

for practice try $x = \begin{bmatrix} \ddot{y} \\ \dot{y} \\ y \end{bmatrix}$

* We can find eigenvalues from eigenvalues - eigenvector equation

$$Av = \lambda v$$

v is the eigenvector associated with the eigenvalue λ

$$Av = \lambda v$$

$$Av - \lambda v = 0$$

$$(A - \lambda I)\, v = 0$$

For this equation to hold we must have

$$\det(A - \lambda I) = 0$$

Using the above equation we can calculate the eigenvalues

For a $n \times n$ <u>full rank</u> matrix

we have n eigenvalues and n eigenvectors

$A v_1 = \lambda_1 v_1 \longrightarrow$ Eigenvector v_1 is associated to λ_1

$A v_2 = \lambda_2 v_2 \longrightarrow$ " v_2 " " " λ_2

$A v_3 = \lambda_3 v_3 \longrightarrow$ " v_3 " " " λ_3

$\vdots$

$A v_n = \lambda_n v_n \longrightarrow$ " v_n " " " λ_n

square matrix

$$A \underbrace{\begin{bmatrix} v_1 & v_2 & v_3 & \cdots & v_n \end{bmatrix}}_{\text{because this is column vector}} = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{bmatrix} \begin{bmatrix} v_1 & v_2 & v_3 & \cdots & v_n \end{bmatrix}$$

$$\therefore A V = V \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \lambda_3 & \\ 0 & & & \lambda_n \end{bmatrix}$$

Multiply V^{-1} from right

$$AVV^{-1} = V \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \lambda_3 & \\ & & & \lambda_4 \end{bmatrix} V^{-1}$$

$$A = V \begin{bmatrix} \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 \end{bmatrix}$$

Multiply V^{-1} from left

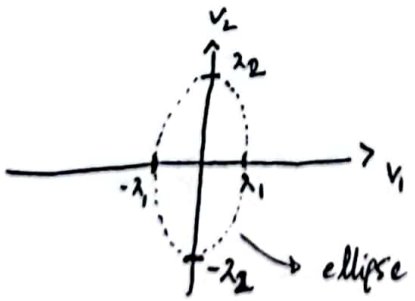
$$A = V \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \lambda_3 & \\ & & & \ddots \lambda_n \end{bmatrix}$$

$$V^{-1} A V A = V^{-1} V \begin{bmatrix} \lambda_1 & 0 & \\ 0 & \lambda_2 & \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

$$A = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

For λ_1 v_1

λ_2 v_2



when $\lambda_1 = \lambda_2$

shape → circle

The smaller the ellipse, the lesser the uncertainty

Used for localization.

Theme:

Date: / /

Sat Sun Mon Tue wed Thu Fri

EEE494

21/03/23

$$\dot{X} = Ax + Bu$$

A is the system matrix

The eigenvalues of A are the poles of the system.

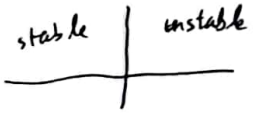
Google Map blue ball

The smaller the ball the accurate the map is tracking me

(Localization)

- The smaller the eigen values the better the accuracy

For system stability, $\lambda < 0$



$Av = \lambda v$

v is the eigenvector associated with eigen value λ

$$\therefore \quad Av_2 = \lambda_1 v_1$$
$$Av_2 = \lambda_2 v_2$$
$$Av_3 = \lambda_2 v_3$$
$$\vdots$$
$$Av_n = \lambda_n v_n$$

More generalized

$$A\Psi = \Psi\Lambda$$

Ψ is matrix of eigenvectors

$$\therefore \Psi = [v_1 \; v_2 \; v_3 \; v_3 \; \ldots \; v_n]$$

Λ is matrix of eigenvalues

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 & \cdots \\ 0 & \lambda_2 & & & \\ 0 & & \lambda_3 & & 0 \\ 0 & & & \ddots & \\ \vdots & & 0 & & \lambda_n \end{bmatrix}$$

Date: / /

Theme: Sat Sun Mon Tue wed Thu Fri

$$\therefore \quad V\Lambda = [v_1 \; v_2 \; v_3 \; \dots \; v_n] \begin{bmatrix} \lambda_1 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & & \\ 0 & 0 & \lambda_3 & & 0 \\ \vdots & & & \ddots & \\ 0 & 0 & & & \lambda_n \end{bmatrix}$$

If I want λ_1

~~$AV_1 = [\quad][v_1][\quad]$~~

$$Av_1 = [v_1 \; v_2 \; v_3 \; \dots \; v_n] \begin{bmatrix} \lambda_1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\therefore Av_1 = [v_1][\lambda_1]$$

$\therefore$ In general $AV = V\Lambda = v_1\lambda_1 + v_2\lambda_2 + v_3\lambda_3 + \dots + v_n\lambda_n$

If matrix A is full rank then we have eigenvalue decomposition

If matrix A is not full rank then we have single value decomposition

Multiply from the left by V^{-1}

$$V^{-1}AV = V^{-1}V\Lambda = \Lambda$$

Theme:

Date: / /

Sat Sun Mon Tue wed Thu Fri

Multiply from the right by V^{-1}

$$A V V^{-1} = V \Lambda V^{-1}$$

$$\therefore \Rightarrow A = V \Lambda V^{-1}$$

When V and Λ are given we can multiply from right by V^{-1}

$$A = V \Lambda^{-1} V^{-1}$$

$$\Lambda^{-1} = \begin{bmatrix} \frac{1}{\lambda_1} & 0 & 0 & 0 \\ 0 & \frac{1}{\lambda_2} & & \\ 0 & & \frac{1}{\lambda_3} & \\ 0 & & & \ddots \\ 0 & & & \frac{1}{\lambda_n} \end{bmatrix}$$

$$A A^{-1} = I$$

$$V \Lambda V^{-1} A^{-1} = I$$

$\Lambda V^{-1} A^{-1} = V^{-1}$ [Multiply both side from the left with V^{-1}]

$V^{-1} A^{-1} = \Lambda^{-1} V^{-1}$ [Multiplying both side from the left with Λ^{-1}]

For Stability $\longrightarrow$ more negative eigenvalues

For Accuracy $\rightarrow$ value of eigenvalue is small

$$\dot{X} = Ax + Bu$$

control law $u = kx$

$$\dot{x} = Ax + Bk\,x$$

$$\Rightarrow \dot{x} = (A + Bk)\,x$$

closed loop system

Choose k so that $A+Bk$ is closed loop system matrix

For this, eigenvalues of $A+Bk$ must be negative for stability

$\therefore$ Choose k so that $A+Bk$ is stable.

Example

$$X = \begin{bmatrix} -3 & -2 \\ 1 & 0 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u$$

Find the control law $u = kx$ so that closed loop eigen values are at $[-3, -2]$

solution

$$u = [k_1 \quad k_2]x$$

$$A + Bk = \begin{bmatrix} -3 & -2 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} [k_1 \quad k_2]$$

$$= \begin{bmatrix} -3 & -2 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} k_1 & k_2 \\ 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} -3+k_1 & -2+k_2 \\ 1 & 0 \end{bmatrix}$$

Now finding the eigenvalues of $A+Bk$

$$(A+Bk)v = \lambda v = 0$$

$$(A+Bk-\lambda)v = 0$$

For this equation to hold

$$\det[A+Bk-\lambda I] = 0$$

Date: / /

Sat Sun Mon Tue wed Thu Fri

me:

$$A + Bk - \lambda I = 0$$

$$= \begin{bmatrix} -3+k_1 & -2+k_2 \\ 1 & 0 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}$$

$$= \begin{bmatrix} -3+k_1-\lambda & -2+k_2 \\ 1 & -\lambda \end{bmatrix}$$

~~$\det(A+Bk-\lambda I) \Rightarrow (-3+k_1-\lambda)(-\lambda)$~~

$$\det(A+Bk-\lambda I) \Rightarrow (-3+k_1-\lambda)(-\lambda) - (-2+k_2) = 0$$

$$\Rightarrow \lambda^2 + (3-k_1)\lambda + (2-k_2) = 0 \quad \text{— (i)}$$

For $[-3, -2]$

$$(\lambda+3)(\lambda+2) = 0$$

$$\lambda^2 + 5\lambda + 6 = 0 \quad \text{— (ii)}$$

Comparing (i) and (ii)

$$3-k_1 = 5 \qquad 2-k_2 = 6$$

$$k_1 = -2 \qquad k_2 = -4$$

$\therefore$ Desired control law : $\boxed{u = [-2 \ -4]x}$

Ans

Theme: EEE494

Date: / /

Sat Sun Mon Tue wed Thu Fri

28/03/23

- Robot Takes info from the environment
  - Sensors take input which is the state vector
    - Makes Decision: $\dot{x} = f(x, u)$
      - Takes action on the environment using actuators

Trajectory generation, tracking, motion planning

dynamical, state-space methods

- optimal control
- Differential flatness
- Lyapunov controller
- PID controller

configuration space methods

- ~~Graph~~ search methods
- Sampling based method
  - A*, PRM, RRT } Prerequisite: Data structure & Algorithms

Next class → Computer Vision

Date: /

Theme:

Sat Sun Mon Tue

EEE 494

04/04/23

Trajectory generation, tracking

- state space based control techniques
- configuration space (c-space) based techniques

C - space based methods

① search based methods

② sampling based methods

C - space

- $C_{Forbidden}$
- C_{free}

$C_{Forbidden}$ → maroon regions

C_{free} → white regions

See lecture 5 pg: 2

Example

① Search based methods : A^*, Dijkstra's

② Sampling based methods : PRM (Probabilistic Road Map)

RRT (Rapidly exploring random tree)

Theme:

Date: / /

Sat Sun Mon Tue wed Thu Fri

In c-space techniques

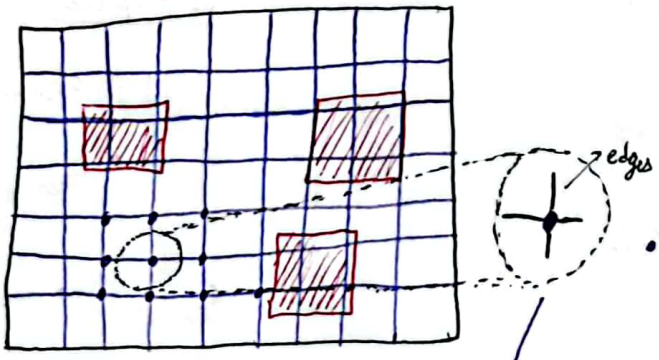
we create the geometric

path of the

robot

we don't consider, the dynamics of the robot.

Search based method



— discritization

— forbidden space

• blue points are called 'vertex' or nodes

For discritization, if no. of square increases, accuracy increases but computation also increase

The black lines connect the 'nodes' and are called 'edges'

∴ Using the nodes of edges for the 2D grid, we can set up a graph of vertices and edges

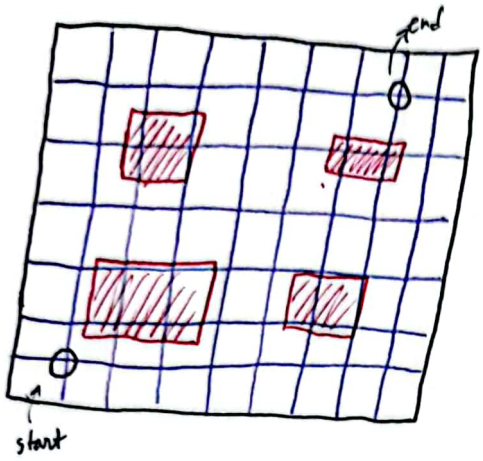
$G = (V, E)$

→ List of node/vertices of free spaces are ready

Theme:

Date: /

Sat Sun Mon Tue



Graph of free space • vertices/nodes created

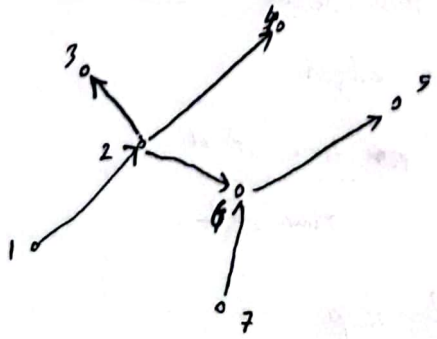
Now we want to send the robot to end point from start.

See lecture 5 pg: 5-6

(Cost-to Arrival) See lecture • 5

A*

Use (Cost-to-Go)



To go from 1 to 5

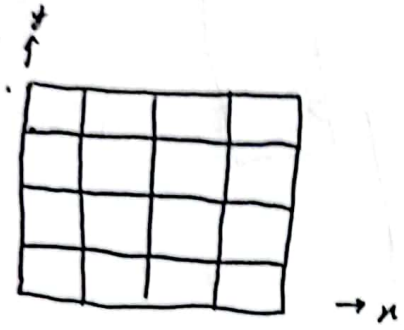
1 → 2 → 6 → 5

Theme:

Date: / /

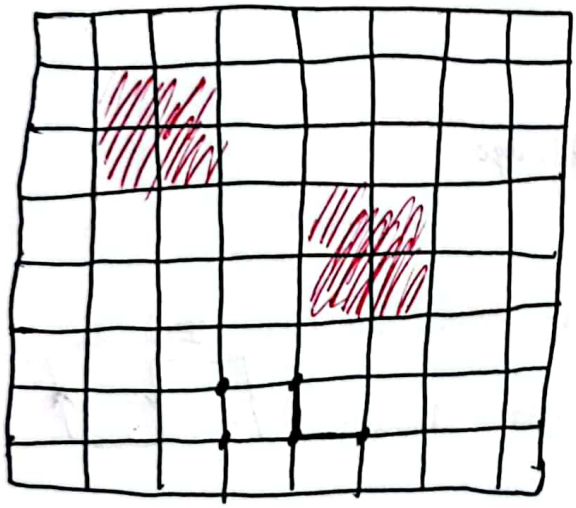
Sat Sun Mon Tue wed Thu Fri

EEE494



Discretization ছোট করলে no. of nodes বাড়ে

Grid Based Search Method



C_{obs} = forbidden zone

C_f → free space (white space)

Steps

① Create graph $G = (V, E)$

from 2d grid

V → Vertex

E → Edge

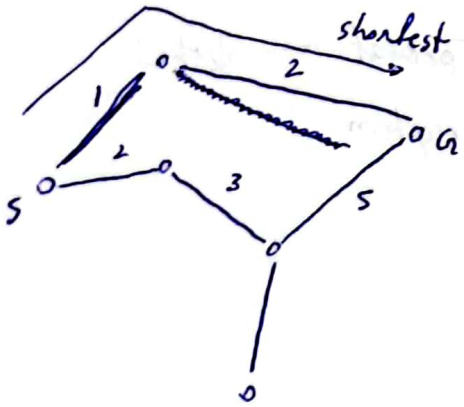
Note

② Intermediate step

a) Collision checker

Exclude node in C_{obs}

② Search for shortest path using A^*



Dijkstra's algorithm

minimize cost of arrival

A^* algorithm

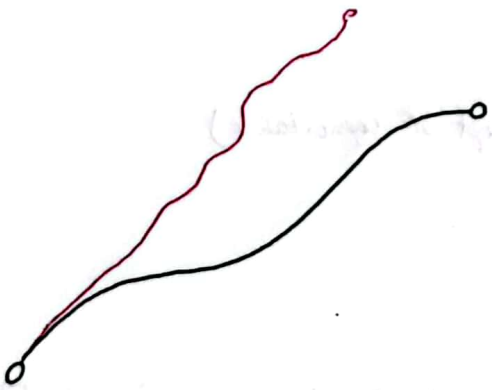
minimize cost of arrival

+

cost to go

cons: Computationally intensive

A lot of computation required to generate graph



x_d desired location

x current position

$e = x_d - x$