EKF

Recursive filter

Can be splited into 2 steps (Prediction & Correction)

Predictions step

it takes the current state of the robot, and the compute a new state (predicted state) based on the input commands

$$\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1})\, bel(x_{t-1})\, dx_{t-1}$$

If we're in state x_t, previous bel, we estimate how does the bel change after executing command u_t, where we're now $\Rightarrow x_t$

Correction

$$bel(x_t) = \eta\, p(z_t \mid x_t)\, \overline{bel}(x_t)$$

which takes sensor measurement

Prediction step

output of the prev step
previous mu, state

$$\bar{\mu}_t = g(u_t, \mu_{t-1})$$

control unit
control input

non linear function

new state

$$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t$$

process noise

prev covariance
uncertainty

update covariance

G_t = Jacobian of

non-linear function g which makes our robot how to move to new state.

~~Section (1.2) EKF~~

Uncertainty of sensor observation

Update

$$K_t = \bar{\Sigma}_t H_t^T \left(H_t \bar{\Sigma}_t H_t^T + Q_t \right)^{-1}$$

Kalman gain;

To compute a weighted gain,
how much certain is the robot about its predicted belief (robot model)

how much certain is the robot about its sensor measurement (measurement)

if you've got very noisy sensor, you should trust more on robot model.

if we have very noisy sensor data t, K will be very small.

$$\mu_t = \bar{\mu}_t + K_t \left(z_t - h(\bar{\mu}_t) \right)$$

$\bar{\mu}_t$: prev mean

z_t: actual measurement

$h(\bar{\mu}_t)$: expected measurement

if K_t is zero, new mean will be at the predicted mean (predicted belief)

Update the Covariance;

$$\Sigma_t = \underbrace{(I - K_t H_t)}_{\text{Jacobian of observation function}} \bar{\Sigma}_t \leftarrow \text{prev covariance uncertainty}$$

state space, $x_t = (x, y, \theta, m_{1,x}, m_{1,y}, \dots, m_{n,x}, m_{n,y})^T$

3 dimensional robot's pose
2 dimensional landmarks' pose

state vector & state covariance

3+2n Gaussian distribution

$$\underbrace{\begin{bmatrix} x_R \\ m_1 \\ \vdots \\ m_n \end{bmatrix}}_{\mu} \underbrace{\left[\begin{array}{c|ccc} \overset{A}{\Sigma_{x_R x_R}} & \Sigma_{x_R m_1} & \overset{B}{\dots} & \Sigma_{x_R m_n} \\ \hline \Sigma_{m_1 x_R} & \Sigma_{m_1 m_1} & \dots & \Sigma_{m_1 m_n} \\ \vdots & \vdots & \ddots & \\ \Sigma_{m_n x_R} & \Sigma_{m_n m_1} & & \Sigma_{m_n m_n} \\ C & & D & \end{array}\right]}_{\Sigma} — (1)$$

A $\Rightarrow$ Covariance on Robot pose (3×3)
θ, x, y

D $\Rightarrow$ Covariance on landmarks

B $\Rightarrow$ Covariance between landmark & robot

C $\Rightarrow$ Covariance between robot & landmark

The function g only affects ~~only~~ the robot's motion, not the landmarks.

Jacobian of the motion,

$$G_t = \begin{bmatrix} G_t^x & 0 \\ 0 & I \end{bmatrix} \quad \text{--- } (2)$$

identity, it affects only on first 3×3 block, robot's pose
$(2n\times2n)$

partial derivative of motion model w.r.t θ, x & y.

After update Covariance,

Block A, B & C will be updated.

Σ_{mm}, landmark Covariance is still the same.

Update Step

base on current estimate

Expected Observation (Section 1.2)

Robot pose

$$\begin{bmatrix} \Delta x = \bar{\mu}_{j,x} - \bar{\mu}_{t,x} \\ \Delta y = \bar{\mu}_{j,y} - \bar{\mu}_{t,y} \end{bmatrix}$$

landmarks

Difference in x & y between landmarks & Robot pose.

$$\hat{z}_t^i = \begin{bmatrix} \sqrt{q} \\ \operatorname{atan2}(\Delta y, \Delta x) - \bar{\mu}_{t,\theta} \end{bmatrix}$$

$h(\bar{\mu}_t)$

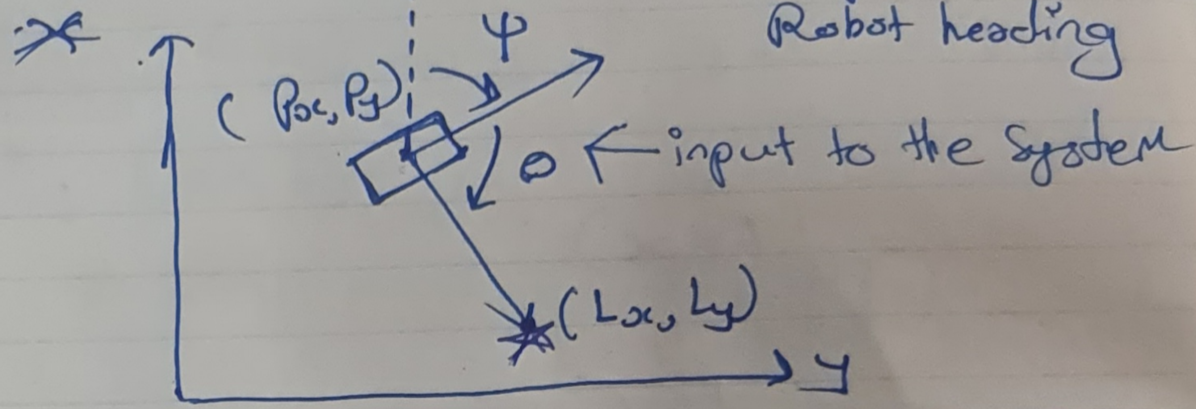
Robot current Heading

q = Euclidean distance

Bearing angle;

$$\text{atan2}(\Delta y, \Delta x) \Longrightarrow \boxed{\psi + \theta} - \psi$$

Robot heading



Jacobian for the Observation, H matrix (Section 1.3)

$$\text{low}\, H_t^i = \frac{\partial h(\bar{\mu}_t)}{\partial \bar{\mu}_t} \quad (2 \times 5 \text{ dimensional})$$

low dimensional space => (θ, x, y, $m_{j,x}$, $m_{j,y}$)

$$\doteq \begin{bmatrix} \frac{\partial \sqrt{q}}{\partial \theta} & \frac{\partial \sqrt{q}}{\partial x} & \frac{\partial \sqrt{q}}{\partial y} & \frac{\partial \sqrt{q}}{\partial m_{j,x}} & \frac{\partial \sqrt{q}}{\partial m_{j,y}} \\ \frac{\partial\, \text{atan2}(\ldots)}{\partial \theta} & \frac{\partial\, \text{atan2}(\ldots)}{\partial x} & \ldots & & \end{bmatrix}$$

$$(x_l - x_R)^2 + (y_l - y_R)^2$$

By Applying Chain Rule;

$$\frac{\partial \sqrt{q}}{\partial x} = \frac{1}{2} q^{-1/2} \cdot 2 \underset{(x_l - x_R)}{\Delta x} (-1)$$

$$= \frac{1}{\sqrt{q}} (-\Delta x)$$

$$= \frac{1}{q} (-\sqrt{q}\, \Delta x)$$

$$^{low}H_t^i = \frac{1}{q} \begin{bmatrix} 0 & -\sqrt{q}\,\Delta x & -\sqrt{q}\,\Delta y & \sqrt{q}\,\Delta x & \sqrt{q}\,\Delta y \\ -q & \Delta y & -\Delta x & -\Delta y & \Delta x \end{bmatrix}$$

Map to High Dimensional Space

$$H_t^i = {}^{low}H_t^i \, (F_{x,j}) \longrightarrow [5 \times (3+2n)] \text{ matrix}$$

Topleft block is identity (3×3 block)

$$F_{x,j} \Rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 \dots 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \dots 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \dots 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \dots 0 & 1 & 0 \\ 0 & 0 & 0 & \underbrace{0 \dots 0}_{2j-2} & 0 & 1 \end{bmatrix}$$

End Section 1.3

Summary;

① $\bar{\mu}_t = g(u_t, u_{t-1})$

② $\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t$

③ $K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$

④ $\mu_t = \bar{\mu}_t + K_t(z_t - \underbrace{h(\bar{\mu}_t)}_{\hat{z}})$

⑤ $\Sigma_t = (I - K_t H_t) \bar{\Sigma}_t$

Assume Unknown Data Association.

Unknow Data Association

1. Find the existing landmarks that generates a measurement closest to the current measurement
2. If the closest one is farther than threshold, this is new landmark.

To find Mahalnobis distance, for the correlation between the variables.

inverse of Covariance

$$MD = \sqrt{(\vec{x} - \vec{\mu}) S^{-1} (\vec{x} - \vec{\mu})}$$

landmark data points

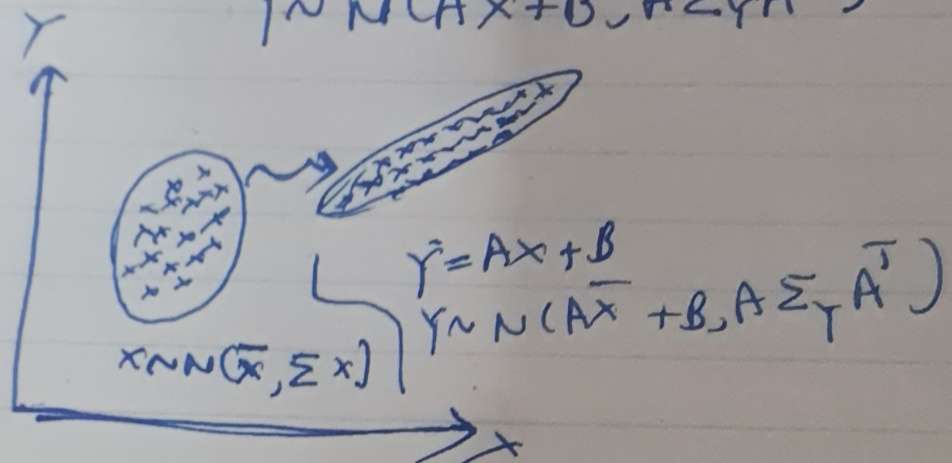
mean vector (or) Centroid

effects of:

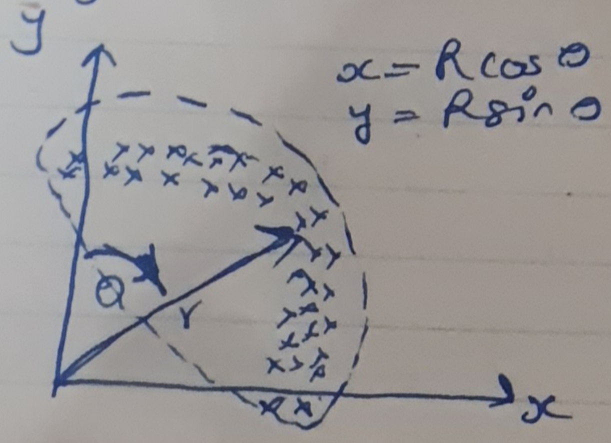
=> Linear Approx of non-linear Uncertain transform

=> Large init condition (or) estimation errors

$$X \sim N(\bar{X}, \Sigma_X)$$
$$\downarrow$$
$$Y = AX + B$$
$$Y \sim N(A\bar{X} + B, A\Sigma_Y A^T)$$



Not always the best;



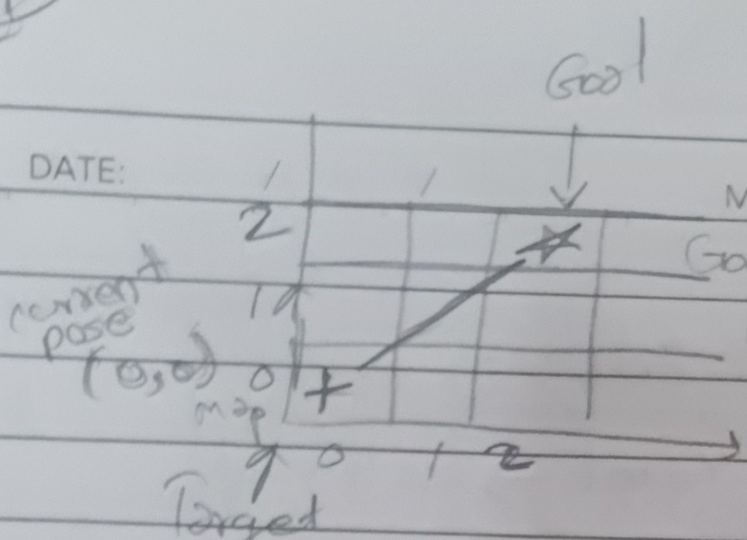
mean & covariance for all these data points

1

A-star Path

DATE: / /

MO TU WE TH FR SA SU



Goal ($x = 2, y = 2$)

Based on the resolution

0.1 => resolution

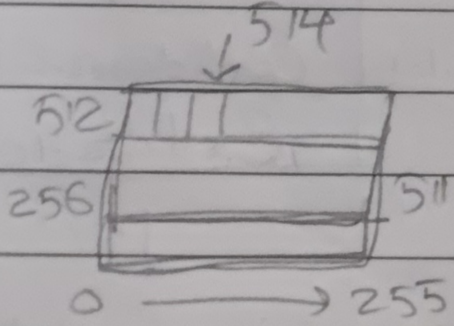
Note => Only for Quadrant 1 of map frame

if Goal is $x = 2, y = 2$, index is => 514

512 → 514

256 → 511

0 ⟹ 255



G cost => the cost to reach the cell from the start, along the previous path till current pose

H cost => cost from cell which is currently observed to the goal, ignoring obstacle <may be manhattan, Euclidean>

F = G + H

②

DATE: / / MO TU WE TH FR SA SU

step ① create two empty lists "open" & "closed"

step ② create cell obj "current" which is equal to the start cell (robot's position)

step ③ Add "current" to "open"

step ④ loop until find the Goal

4.1 => Check the size of open list

4.2 => Exploring the Neighbours

4.2.1 if cell is already in open list calculate new F cost and compare if new is lower, make the current cell new parent

4.2.2 => if not in either open or closed, Add to open if it's not obstacle.

~~step~~ 4.3. After checking all the neighbours, Add current to closed & remove from open list.

4.4. Find the cell in the open list with the lowest F cost and make new current.

4.5. We've found the goal and set the last cell as goal's parent. Add the goal to closed.

x = -1, x <= 1, x++

y = -1, y <= 1, y++

DATE: / /

MO TU WE TH FR SA SU

y = -1, x = -1 (outside)
y = 0, x = -1 (outside)
y = 1, x = -1 (outside)
x = 0, y = -1 (outside)
x = 0, y = 0 (inside)
x = 0, y = 1 (inside)
x = 1, y = -1 (outside)
x = 1, y = 0 (inside)
x = 1, y = 1 (inside)

for all the neighbours of start pose

~~x = 0, y = 0~~
x = 0, y = 0
x = 1, y = 0
~~x = 1, y = 1~~

All the parents are current x & y (0, 0)

After first iteration

Add the current to closed

open closed
~~(0, 0)~~ removed (0, 0)
(0, 1)
(1, 0)
(1, 1) —

Set lowest F cost to current of next

x = 0; x <= 2, x++

y = 0; y <= 2; y++

0, 0
0, 1
0, 2
1, 0
1, 1
1, 2
2, 0
2, 1
2, 1

4

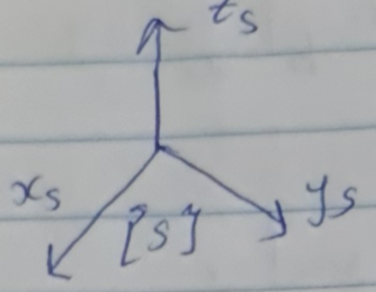
DATE: /　　　　MO TU WE TH FR SA SU

Open	closed
~~(0,0)~~	(0,0)
	(1,1) → Parent (0,0)
(0,1)	
(1,0)	(2,2) → Parent (1,1)
~~(1,1)~~	Push the to parent of goal
(0,2)	$\boxed{\text{Path} \Leftarrow (2,2)(1,1)(0,0)}$
(1,2)	
~~(2,0)~~	1st step
(2,1)	Path ← (2,2)(1,1)
~~(2,2)~~	2nd step
	Path ⇐ (2,2)(1,1)(0,0)

After comparing that path.back().index is equal to the start.index, path trace back is completed.

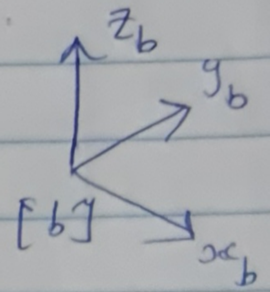
Calculate every angle for all waypoints before publishing.

Rotation Matrices Section 1.0



{s} ⇒ space frame, fixed frame
world frame

{b} ⇒ body frame

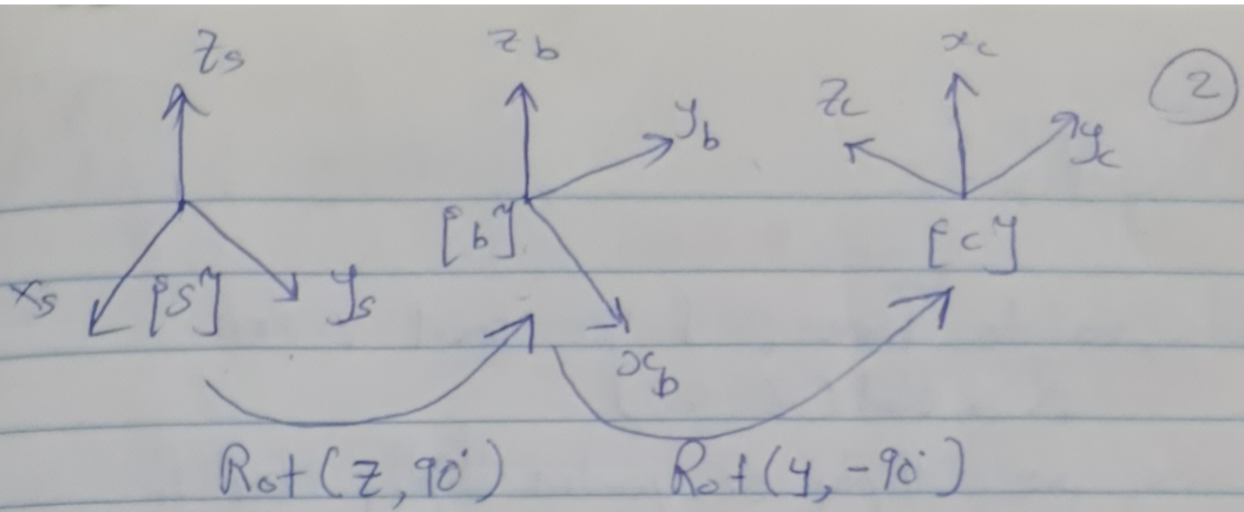


We can write the orientation of frame {b} with respect to frame {s}

$$R_{sb} = [x_b \; y_b \; z_b] = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = R$$

frame of orientation
frame of reference

Note: Right Hand Rule & Every counter clockwise rotation is positive.

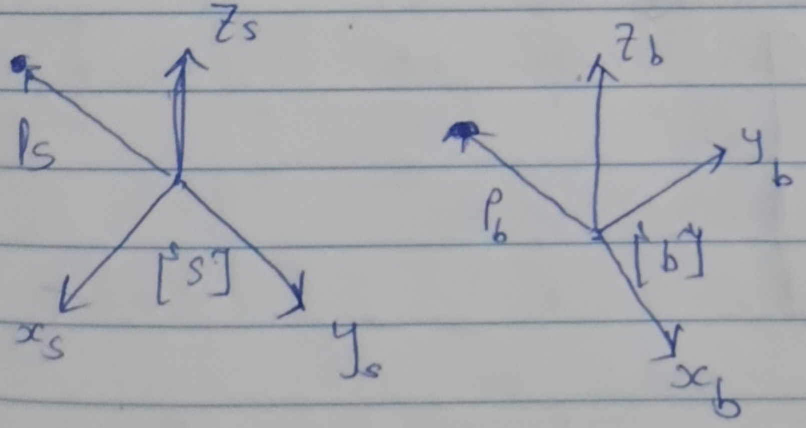


$$R_{sc} = R_{sb} R_{bc} = R_{sc}$$

$$= \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix}$$

Changing the frame [c] w.r.t frame [s]

Changing the frame of reference of a vector



$$P_b = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$$

$$P_s = R_{sb} P_b \quad \text{—— (1)}$$

All these examples are only for orientation.

Rotating Frame

let say, rotate frame [c] 90° about z axis

$$R = Rot(z, 90°)$$

By using premultiply or postmultiply by R

X

Changing frame of a reference for both translation & orientation

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} P_s = \overbrace{\begin{bmatrix} \cos\theta & -\sin\theta & 0 & x \\ \sin\theta & \cos\theta & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}}^{T_{sb}} \overbrace{\begin{bmatrix} x_v \\ y_v \\ 0 \\ 1 \end{bmatrix}}^{P_b} \quad \text{—— } (2)$$

2D twist can be written as

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

Adjoint

2D adjoint converts a twist in one frame to a twist in another frame.

2) Ad_j for 2d planner

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

$$V_a = \begin{bmatrix} Ad_{T_{ab}} \end{bmatrix} V_b \quad \text{--- (3) Eqn}$$

V_b: twist in frame b

V_a: twist in frame a

$$T_{sc} = T_{sb} T_{bc} = T_{sc}$$

$$= \begin{bmatrix} c1 & -s1 & 0 \\ s1 & c1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c2 & -s2 & 0 \\ s1 & c2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T_{sc} = \begin{bmatrix} c1 & -s1 & 0 & x_1 \\ s1 & c1 & 0 & y_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c2 & -s2 & 0 & x_2 \\ s2 & c2 & 0 & y_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eqn --- (4)}$$

Section 10.1

Odometry

6 dimensional twist of chassis,

$$V_{b6} = \begin{bmatrix} \omega_{bx} \\ \omega_{by} \\ \omega_{bz} \\ v_{bx} \\ v_{by} \\ v_{bz} \end{bmatrix}$$, Since the model is diff drive, 2d planar

$$V_{b6} = \begin{bmatrix} 0 \\ 0 \\ \omega_{bz} \\ v_{bx} \\ v_{by} \\ 0 \end{bmatrix}, \quad v_{by} = 0$$

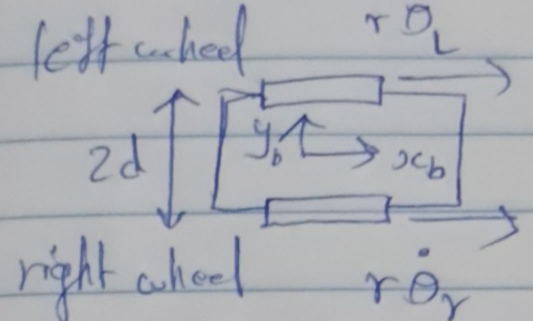
Odom process;

1. Wheel displacement, $\Delta\theta$ (from encoder)
2. Assume dt is 1; $\dot{\theta} = \Delta\theta / \Delta t$
3. Find twist (chassis) from $\Delta\theta$, wheelToTwist
4. Integrate twist (linear & angular):
   To find the configuration of chassis (Tf) at $k+1$ ~~with respe~~ relative to k

Note: $\Delta t = 1$, $T_{b_k b_{k+1}} = e^{[V_{b6}]}$

5. find new chassis tf relative to space (world, fixed) frame.

$$T_{sb_{k+1}} = T_{sb_k} T_{b_k b_{k+1}}$$



Wheel To Twist

chassis twist $\rightarrow$ $$V_b = r \begin{bmatrix} -1/2d & 1/2d \\ 1/2 & 1/2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\theta_L \\ \Delta\theta_r \end{bmatrix} \quad \text{—— Equ 5}$$

Velocity of base w.r.t world frame.

$$\begin{bmatrix} \dot{\theta} \\ \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \frac{r}{2d} & \frac{r}{2d} \\ \frac{r}{2}c\theta & \frac{r}{2}c\theta \\ \frac{r}{2}s\theta & \frac{r}{2}s\theta \end{bmatrix} \begin{bmatrix} \Delta\theta_L \\ \Delta\theta_R \end{bmatrix} \quad \text{—— Equ 6}$$

wheel to twist;

$$\omega_z = r\left[\frac{1}{2d}\Delta\theta_R - \frac{1}{2d}\Delta\theta_L\right] \quad \text{—— Equ 7}$$

radius $\uparrow r$; wheel separation $\uparrow 2d$

Equ 5 & 7 are drive from Equ 6