Exam
- Section (Theory)
- Section - Modeling (Filtering + Least Squares)

INDOOR LOCALIZATION (exercise)

indoor GPS = set of beacons at known locations in the environment → transmit data pkg PERIODICALLY

data sent by the i-th beacon = [time when pkg was transmitted
location p_i of the beacon]

→ SYNCHRONIZED, no clock skew = (all beacons speak in the same clock)

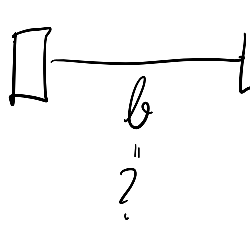
radio → speed of light!

IF ROBOT HAS SAME CLOCK AS BEACONS
straight forward: we know time when pkg sent
+
$v_{transmission}$
+
$t_{current}$
→ we know the distance travelled between robot and beacon

However, clock robot NOT SYNCHRONIZED with beacons

Differential robots → controlled by rotational velocity of the wheels

we do not know



ID = ?

ΔT control interval + velocity assumed to be constant within interval control

$n_x \sim \Sigma(\emptyset, n_x)$ → $x_t = f(x_{t-1}, u_{t-1}) + n_x$
(in addition to control noise)

3.1. Model Dynamic System

- State
- Control
- Measurements

- Transition Function
- Observation Function

State
- robot pose $x_R \in SE(2)$, $x_R \in \mathbb{R}^3 = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$
- intrinsic parameters $X_K = (K_r, K_\ell, b) \in \mathbb{R}^3$
  - coefficient to convert rad of rotation to distance
  - distance between wheels
- time delay between receiver and the beacons X_τ ⊕ assume ~ constant (offset clock robot vs GPS)

Control – $u = (u_\ell, u_r)^T \in \mathbb{R}^2$

left and right wheel velocity (rad/s)

⇓

state $X = (X_R, X_K, X_\tau)^T \in \mathbb{R}^7$

measurements
- t_{tx} – clock time when pkg transmitted
- t_{rx} – " " " received

$\Rightarrow Z = t_{rx} - t_{tx}$ (time difference)

p_i is not part of the measurement (does not change when robot moves through environment)

transitional model:

$$\Delta_{x,t} \cong \frac{u_r \cdot K_r + u_\ell \cdot K_\ell}{2} \cdot \Delta T$$

$$\Delta_{y,t} \cong \emptyset$$

$$\Delta_{\theta,t} = \frac{u_r \cdot K_r - u_\ell \cdot K_\ell}{b} \cdot \Delta T$$

robot relative motion

frame (world → robot)

$$\Rightarrow X_{R,t} = X_{R,t} + \begin{pmatrix} \cos(\theta_t) \cdot \Delta_{x,t} \\ \sin(\theta_t) \cdot \Delta_{y,t} \\ \Delta_{\theta,t} \end{pmatrix} \text{ — robot pose}$$

$X_{K,t} = X_{K,t-1}$ — kinematic parameters

$X_{\tau,t} = X_{\tau,t-1}$ — time skew

measurement function:

$$h(x) = \frac{\| X_R^{-1} \cdot p_i \|}{C_0} - X_\tau$$

distance between beacon and the robot

not dependent on orientation of robot

expected time of transmission between beacon i and robot

compensation for the clock skew

is the orientation observable?
- not from the measurements
- but I can infer it through the transition of the state!

only because our transition model correlates translation with orientation

Question 2 and 3

will it work? - depends on initial guess! → IF TOO FAR OFF + NOT SENSING TO MANY BEACONS → problem may be under constrained...

EKF

$p(x_{0,0}) \sim \mathcal{N}(x, \mu_{00}, \Sigma_{00})$

$p(u) \sim u_t + n_u$

$A_t \in \mathbb{R}^{7\times 7}$

$B_t \in \mathbb{R}^{7\times 2}$

$C_t \in \mathbb{R}^{1\times 7}$

PREDICT:

$$\mu_{t|t-1} = A_t \mu_{t-1|t-1} + B_t u_{t-1}$$

$$\Sigma_{t|t-1} = A_t \Sigma_{t-1|t-1} A_t^T + B_t \Sigma_n B_t^T + \Sigma_x$$

CORRECT:

$$\mu_z = C_t \mu_{t|t-1}$$

$$K_t = \Sigma_{t|t-1} C_t^T \left(\Sigma_z + C_t \Sigma_{t|t-1} C_t^T\right)^{-1}$$

$$\mu_{t|t} = \mu_{t|t-1} + K_t (z_t - \mu_z)$$

$$\Sigma_{t|t} = (I - K_t C_t) \Sigma_{t|t-1}$$

UKF

$p(x_{0,0}) \sim \mathcal{N}(x, \mu_{00}, \Sigma_{00})$

$p(u) \sim u_t + n_u$

$$p(x_{t-1|t-1}, u_{t-1}) \sim \mathcal{N}\left(\begin{pmatrix} \mu_{t-1|t-1} \\ u_{t-1} \end{pmatrix}, \begin{pmatrix} \Sigma_{t-1|t-1} & 0 \\ 0 & \Sigma_n \end{pmatrix}\right)$$

- Compute 19 sigma points $\chi_{t-1|t-1}$
- apply transition to obtain
- apply transition to obtain $\chi_{t|t-1}$

CORRECTION:

- $z_t^{(i)} = h\left(\chi_{t|t-1}^{(i)}\right)$
- $\mu_z = \sum_i w_m^{(i)} z_t^{(i)}$
- $\mu_{t|t-1} = \sum_i w_m^{(i)} \chi_{t|t-1}^{(i)}$

⇓

- reconstruct required block of covariance

$$\Sigma_{x,z} = \sum_i w_c^{(i)} \left(\chi_{t|t-1}^{(i)} - \mu_{t|t-1}\right)\left(z_t^{(i)} - \mu_z\right)^T$$

$$\Sigma_{z,z} = \sum_i w_c^{(i)} \left(z_t^{(i)} - \mu_z\right)\left(z_t^{(i)} - \mu_z\right)^T$$

⇓

- apply ~~talking~~ CONDITIONING equation

$$\mu_{t|t} = \mu_{t|t-1} + \Sigma_{x,z}\left(\Sigma_{zh} + \Sigma_{z,z}\right)^{-1}(z_t - \mu_z)$$

$$\Sigma_{t|t} = \left(\Sigma_{t|t-1} + \Sigma_x\right) - \Sigma_{x,z}\left(\Sigma_{zh} + \Sigma_{z,z}\right)^{-1} \Sigma_{zx}$$

$\boxed{\text{PARTICLE FILTER}}$ → not the best → give $\mathbb{R}^7$ (the state) ⊕ high dim. of state

or mahalanobis distance (or 1gxt) if I want a smoother profile... EXAMPLE

initial particle set

$P_0 = \{ (x_0^{(i)}, w_0^{(i)}) \mid i \in [1, 2, ..., N] \}$ →

control noise

$p(u) \sim u_t + n_u$

our perturbation that alters the controls

but not the time skew

TRANSITION:

apply transitional model to each element of P_{t-1}

for each particle, draw a different sample from control and precess noise

$(u_{t-1}^{(i)}, \delta_x^{(i)})$

$X_t^{(i)} = f(x_{t-1}^{(i)}, u_{t-1}^{(i)}) + \delta_x^{(i)}$

given that n_u does not influence the translation of / THAT'S WHY the time offset

inject noise in the state to avoid overconvergence of the time skew

CORRECTION:

→ $p(z_t \mid x_t^{(i)}) = \frac{1}{2} e^{-|h(x_t^{(i)}) - z_t|}$

likelihood function → must be HIGH if predicted matched vs low no

black magic?...

$w_t^{(i)} = w_{t-1}^{(i)} \cdot p(z_t \mid x_t^{(i)})$

⊕ normalization of state (angle)

RESAMPLE → do not forget!

Additional exercises → 07_EXERCISES_KALMAN_FILTER!

$\boxed{\text{LEAST SQUARES BEARING-ONLY}}$ → only observable the angle when observing a point (from a certain unknown position)

relative angle between robot and point

$p = \begin{pmatrix} p_x \\ p_y \end{pmatrix}$ — point (in the world)

$T = [R \mid t]$ → position world relative to the robot

$T_{inv} = [R_{inv}, t_{inv}]$

$p\_robot = R^T \cdot p + t_{inv}$ → not our measurement

$z_t = atan2(p\_robot.y, p\_robot.x)$ → observable bearing angle

1° - start from mocked data → to evaluate if data obtained is consistent to what we see!

generate random points

transform points

get bearing

atan2 has 2 values → $\partial atan2 = [\ \ ,\ \ ]$

$$= \frac{1}{\underbrace{p^T \cdot p}_{x^2+y^2}} \cdot [-p(2),\ p(1)]$$

PREDICTION:

h → compute the bearing

assuming I have a robot position

$[e, J] = \text{ERROR AND JACOBIAN}(X, P, Z)$:

BEARING

POSITION WORLD → ROBOT

$t = X(1:2)$; // translation

$\theta = X(3)$

$R = R(\theta)$

$p\_in\_robot = R \cdot p + t$;

$Z\_hat = atan2(p\_in\_robot(2), p\_in\_robot(1))$

$e = z\_hat - z$ $\longrightarrow$ MISSING NORMALIZATION !

$e = atan2(\sin(e), \cos(e))$

$$\frac{\partial e_{new}}{\partial x} = \frac{\partial\, atan2}{\partial x} \cdot \frac{\partial e_{old}}{\partial x}$$

$J\_icp = rotation2D\_gradient(\theta) \cdot p$ → kernel part

$J = J\_atan2(p\_in\_robot) * J\_icp$ → COMPOSITIVE FUNCTION

ENDFUCTION

↓

ICP → Only requires to change the

error and Jacobian Building

Least Squares Application – Odometry Calibration

In literature sometimes

$H \rightarrow$ approximation?

b → gradient vector

$J_{M \times N}$

total measurement → long as the state vector

1° - identify state (dimension) ⊕ Euclidean parametrization

2° - identify measurements (what can we observe about environment or system)

3° - identify prediction function

$$u_i' = f_i(x) = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix} \cdot u_i$$

We assume we do not have access to the intrinsics!

for each u_i', we have u_i^* → ground-truth by external sensor

Chi-squared is an exact PARABOLA!

$$e_i(x) = u_i^* - \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix} \cdot u_i \longrightarrow \text{LINEAR FUNCTION}$$

DOES NOT DEPEND ON THE STATE

$$A_i = \frac{\partial e_i(x)}{\partial x} = -\begin{pmatrix} u_{i,x} & u_{i,y} & u_{i,\theta} & & \emptyset & \\ & \emptyset & & u_{i,x} \; u_{i,y} \; u_{i,\theta} & & \emptyset \\ & & & \emptyset & & u_{i,x} \; u_{i,y} \; u_{i,\theta} \end{pmatrix}$$

$\frac{\partial}{\partial x_{11}} \quad \frac{\partial}{\partial x_{12}} \quad \frac{\partial}{\partial x_{13}} \quad \ldots$

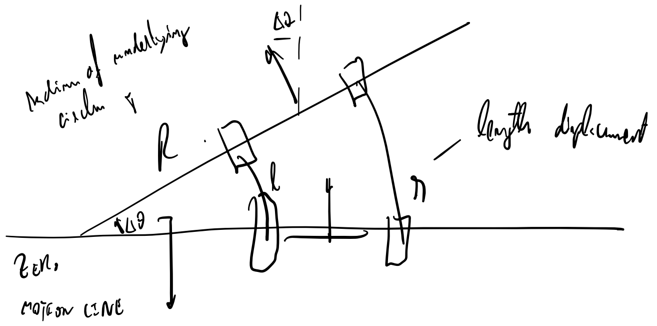
CONSTANT (in each iteration of the algorithm is the state estimation that changes)

Will converge in single iteration!

Humanoids? (more like a unicycle in 2D)

Any wheel drive

2D Calibrating a Unicycle (differential is a derivation of unicycle)



Change in the attitude of robot

$$r = \Delta\theta \left(R + b/2\right)$$

$$l = \Delta\theta \left(R - b/2\right)$$

$\Downarrow$ invert mapping

$$\frac{l}{\Delta\theta} + b/2 = R$$

$$\frac{r}{\Delta\theta} - b/2 = R$$

$$\longrightarrow \frac{l}{\Delta\theta} + b/2 = \frac{r}{\Delta\theta} - b/2$$

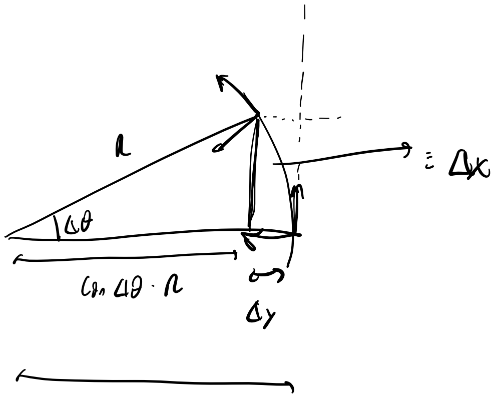
$$b = \frac{r-l}{\Delta\theta}$$

$$b = \frac{r-l}{\Delta\theta}$$

$$R = \frac{r+l}{2\Delta\theta} \longleftarrow \underset{\times 2}{\frac{l}{\Delta\theta}} + \frac{r-l}{2\Delta\theta} = R = \frac{l+r}{2\Delta\theta}$$

↓ Radius of circle of motion

$$\Delta x = R \cdot \text{sen}(\Delta\theta) \quad ; \quad \Delta y = R\,(1 - \cos(\Delta\theta))$$ → displacement from the … origin



$$r^{[i]} = K_r \cdot t_r^{[i]}$$
$$l^{[i]} = K_l \cdot t_l^{[i]}$$

encoder → meters (resemble wheel diameters) ↓ not exactly the radius !

encoder ticks

1° translate
2° rotate
→ introduce error

usually, we use Euler integration

REAL MODEL OF ROBOT = EXACT INTEGRATION

⇓

$$\Delta x^{[i]} = \frac{r^{[i]} + l^{[i]}}{2} \cdot \frac{\text{sen}(\Delta\theta^{[i]})}{\Delta\theta^{[i]}}$$

$$\Delta y^{[i]} = \frac{r^{[i]} + l^{[i]}}{2} \cdot \frac{1 - \cos(\Delta\theta^{[i]})}{\Delta\theta^{[i]}}$$

$$\Delta\theta^{[i]} = \frac{r^{[i]} - l^{[i]}}{b}$$

b → base line

what if $\Delta\theta \to \emptyset$?

⇓

PROBLEM

(undefined model when $\Delta\theta = \emptyset$)

1st alternative:

if $|\Delta\theta| \leq 1e\text{-}8$

$$\Delta x^{[i]} = \frac{r^{[i]} + l^{[i]}}{2}$$

$$\Delta y^{[i]} = \emptyset$$

ELSE

…

2° alternative:

if $\Delta\theta$ small → Taylor expansion !

MASKING THE SINGULARITY !

↓

NOT PHYSICAL → only due to the mathematical formulation of the model