The Simplest SLAM

I wish to recover both state of robot
AND ~~to~~ a map of the world.

CASE:

2D translating robot,

N Beacons at ${}^{i}\underline{m}$ (in World coords)

- State of robot $\underline{x}_i$ (")
- robot measures location of beacons in its frame

State

$$\begin{pmatrix} \underline{x} \\ {}^{1}\underline{m} \\ {}^{2}\underline{m} \\ \vdots \\ \vdots \\ \vdots \end{pmatrix}$$

Dynamics

$$\underline{x}_{i+1} = \underline{x}_i + \underline{\xi}_i$$

$${}^{k}m_{i+1} = {}^{k}m_i$$

Measurement

$${}^{k}\underline{y}_i = ({}^{k}\underline{m}_i - \underline{x}_i) + \eta_i$$

This is all linear, so sling it in a KF

Issues:

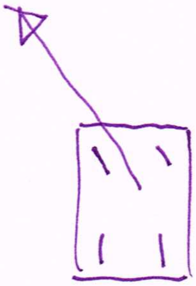
- Data association
  - you need to know which measurement comes from which beacon

- Strategies
  - $\log P({}^{k}g_i \mid x_i ; {}^{r}m_i)$ and matching (Bipartite ; Greedy)
  - mark the beacons

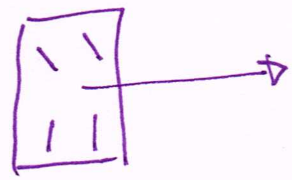
- Missing data:
  - What happens if a beacon ~~doe~~ isn't found?
  - Easy drop from measurement matrix, proceed.

- How accurate?
  - should be very good, with more than 2 Beacons
  - 2 is a problem.

- Rotating + translating robot
  - like our car.
  - motion of robot depends on state



OK



NOT OK

(Formally - Non-holonomic).

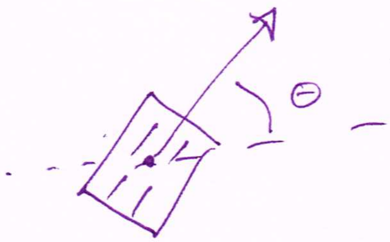
State of car:

posn in W.C $\longrightarrow$ x

y

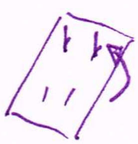
θ

angle in W.C



$$
\begin{array}{lcl}
x & \longrightarrow & x + v\Delta t \, \cancel{\sin}\ \cos\theta \\
y & & y + v\Delta t \sin\theta \\
\theta & & \theta
\end{array}
$$

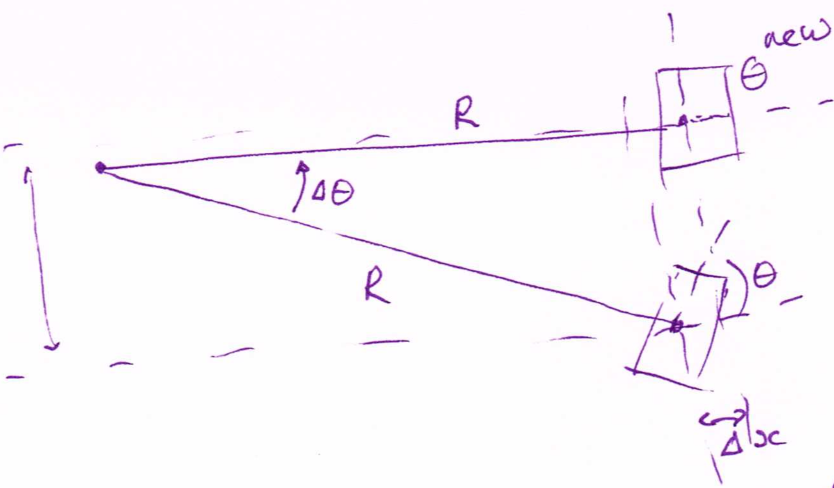
Pure rotation :



$$
\begin{array}{lcl}
x & \longrightarrow & x \\
y & & y \\
\theta & & \theta + \Delta\theta
\end{array}
$$

motion in a circle

$\theta^{new} = \theta + \Delta\theta$



$$
\begin{array}{lclcl}
x & \longrightarrow & x + \Delta x & = & x + R(\sin(\theta+\Delta\theta) - \sin\theta) \\
y & & y + \Delta y & = & y - R(\cos(\theta+\Delta\theta) - \cos\theta) \\
\theta & & \theta + \Delta\theta & &
\end{array}
$$

Notice:

pure rotation = motion in a circle, $R=0$

pure translation = " $R \to \infty = \frac{1}{\Delta\theta}$

$\Delta\theta \to 0$

(take derivatives).

So our model is

$$\begin{matrix} x \\ y \\ \theta \end{matrix} \to \begin{matrix} x + R(\sin(\theta+\Delta\theta) - \sin\theta) \\ y - R(\cos(\theta+\Delta\theta) - \cos\theta) \\ \theta + \Delta\theta \end{matrix} = \underline{f}(x, y, \theta; R, \ldots$$

where R comes from velocity etc.

Notice this isn't linear

Measurement

$${}^{k}\underline{g}_i = R^T(\theta)\left[{}^{k}\underline{M}_i - \begin{pmatrix} x \\ y \end{pmatrix}_i\right] + \eta_i$$

noise

depends on θ, nonlinear.

- **EKF** = extended Kalman filter

  - linearize dynamics, motion around current best estimate
  - apply KF

- **Particle filter**:

  - this needs to be done with care FOR OUR VEHICLE.
  - because dynamic uncertainty is quite large, AND meas. uncertainty is quite small.

· **Options**

  · propagate samples thru dynamics reweight (as we did) DICEY

  · **alternative** propose particles from measurements, reweight.

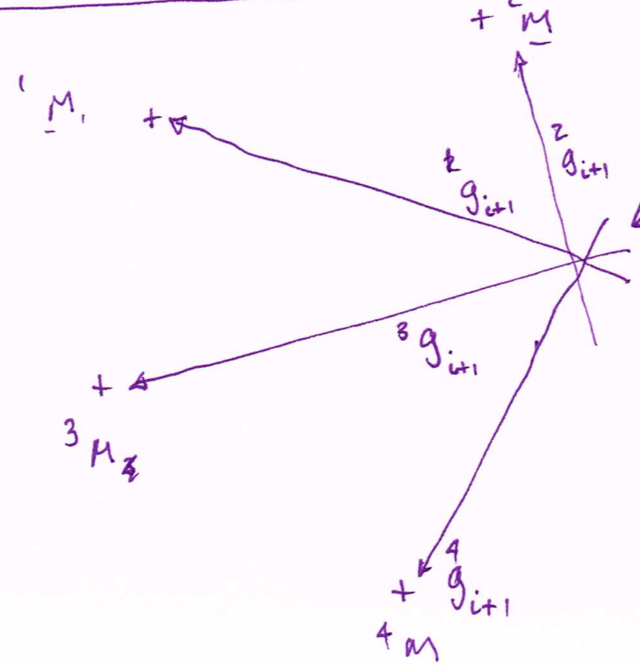
a particle is:

- $x, y, \theta, {}^1\underline{m}, \ldots {}^k\underline{m}$

- ~~the~~ particles will have slightly different ${}^i\underline{m}$ — PRIOR at start

strategy

- for some particle



find an $\hat{x}$ $\hat{y}$ $\hat{\theta}$

consistent with measurements, ~~and a~~

- now construct a proposal dist
$Q(\text{state}_{i+1} \mid M_{i+1})$
(eg normal, centered on mean).

- reweight w/ dynamics

the math:

~~Our p~~ Original P.F.

$$p_r \sim P(X_{i+1} \mid Y_1 \cdots Y_i)$$

weight: $\left(p_r, w_r\right) = \left(p_r, P(Y_{i+1} \mid X_{i+1} = p_r)\right)$

eval:

$$\frac{\sum f(p_r)\, w_r}{\sum w_r} \approx \frac{N \int f(u) \cdot P(Y_{i+1} \mid X_{i+1} = u) \times P(X_{i+1} = u \mid Y_1 \cdots Y_i)\, du}{N \int P(Y_{i+1} \mid X_{i+1} = u) \cdot P(X_{i+1} = u \mid Y_1 \cdots$$

$$\approx \int f(u)\, P(X_{i+1} = u \mid Y_1 \cdots Y_{i+1})\, du.$$

modified:

$$p_r \sim Q(X_{i+1} \mid meas)$$

weight:

$$p_r \, , \quad \frac{P(Y_{i+1} \mid X_{i+1} = p_r) \, P(X_{i+1} = p_r \mid Y_1 \cdots Y_i)}{Q(X_{i+1} = p_r \mid meas)}$$

check:

$$\sum f(p_r) . w_r \approx N \int f(u) . \left[\frac{P(Y_{i+1} \mid X_{i+1} = u) \, P(X_{i+1} = u \mid Y_1 \cdots Y_i)}{Q(X_{i+1} = u \mid meas)} \right]$$

$$\times \, Q(X_{i+1} = u \mid meas) . du$$

OK.

How to make Q.

- at particle, we have an est of old θ.
- so we can do linear least squares

$$R(\theta + \Delta\theta) = R(\theta)\left[I + \begin{bmatrix} 0 & -\Delta\theta \\ \Delta\theta & 0 \end{bmatrix}\right] \leftarrow \text{Taylor series}$$

$${}^{K}\underline{g}_i - \cancel{R(\theta)}\left[I + \begin{bmatrix} 0 & \Delta\theta \\ -\Delta\theta & 0 \end{bmatrix}\right] R^{T}(\theta)\left[{}^{K}M_i - \begin{pmatrix} x+\Delta x \\ y+\Delta y \end{pmatrix}\right]$$

ignore $O(\Delta^2)$; solve as linear least squares.

OR - est rotation, tx as before!

Simple Bundle adjustment

+ assume point csp's are known

$\overset{\text{coord frame}}{{}^{k}\underline{x}_{i}}$ point identity

(so we see point i in several frames)

+ we want best est. of ${}^{w}\underline{x}_i$ (world)

+ write $\underline{m}_i \leftarrow$ for ${}^{w}\underline{x}_i$

+ assume frame 1 is world

then

$$\sum_{k \in \text{frames}} \sum_{i \in \text{pts. vis in } k} \left[R(\theta_k)\,{}^{k}\underline{x}_i + {}^{k}t - \underline{m}_i \right]^2 = F(\theta_k, {}^{k}t, \underline{m}_i)$$

minimize this WRT θ_k, ${}^{k}t$, $\underline{m}$.

iterate two phases

$$\hat{\theta}_k, \hat{t}^k = \operatorname{argmin} F(\theta_k, t^k; \hat{m}_i)$$

here we fix world points, adjust θ, t;

notice ~~these~~ decomposes but isn't linear

$$\hat{m}_i = \operatorname{argmin} F(\hat{\theta}_k, \hat{t}^k; m_i)$$

least squares

You can extend this to ICP if you choose

- a fixed number of points in world coords
- initialize with map constructed above
- complicate w/ IRLS,