Course 05: EKF SLAM

Saturday, November 2, 2019 10:23 PM

slam05-ekf-slam

Robot Mapping

EKF SLAM

Cyrill Stachniss

UNI FREIBURG

AiS Autonomous Intelligent Systems

1

Simultaneous Localization and Mapping (SLAM)

- Building a map and locating the robot in the map at the same time
- Chicken-or-egg problem

2

Definition of the SLAM Problem

Given

- The robot's controls

$$u_{1:T} = \{u_1, u_2, u_3, \ldots, u_T\}$$

① raw commands (e.g. speed)
② (OR) odometry readings [Yes! We treat odom read as control commands].
↳ counting the revolutions of the wheels and estimate where the vehicle goes.

- Observations

$$z_{1:T} = \{z_1, z_2, z_3, \ldots, z_T\}$$

① laser range finder
② camera
[range, steer info of the env]

Wanted

- Map of the environment

$$m$$

- Path of the robot

$$x_{0:T} = \{x_0, x_1, x_2, \ldots, x_T\}$$

3

Three Main Paradigms

Kalman filter | Particle filter | Graph-based

4

Bayes Filter

- Recursive filter with prediction and correction step

- **Prediction**

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

- **Correction**

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

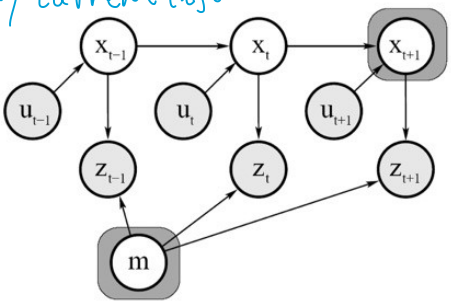
5

EKF for Online SLAM

- We consider here the Kalman filter as a solution to the online SLAM problem

$$p(x_t, m \mid z_{1:t}, u_{1:t})$$

only current info



6

Extended Kalman Filter Algorithm

1: **Extended_Kalman_filter**$(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$:

① pred step

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

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

jacobian of g

for specific app:
① define g
② compute G_t

② corr step

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

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

6: $\Sigma_t = (I - K_t\, H_t)\, \bar{\Sigma}_t$

7: return μ_t, Σ_t

obtain observ

pred observ

About Kalman gain K_t: to compute a weight
1) how much the robot is certain about its predicted belief ($\bar{\mu}_t$ & $\bar{\Sigma}_t$)
2) how much the robot is certain about its sensor properties

7

Intuition here is : WHAT ARE GOING TO ESTIMATE ?

↳ pose of ROBOT + LANDMARKS

EKF SLAM

- Application of the EKF to SLAM
- Estimate robot's pose and locations of landmarks in the environment
- Assumption: known correspondences
- State space (for the 2D plane) is

$$x_t = (\underbrace{x, y, \theta}_{\text{robot's pose}}, \underbrace{m_{1,x}, m_{1,y}}_{\text{landmark 1}}, \ldots, \underbrace{m_{n,x}, m_{n,y}}_{\text{landmark n}})^T$$

8

EKF SLAM: State Representation

- Map with n landmarks: (3+2n)-dimensional Gaussian
- Belief is represented by

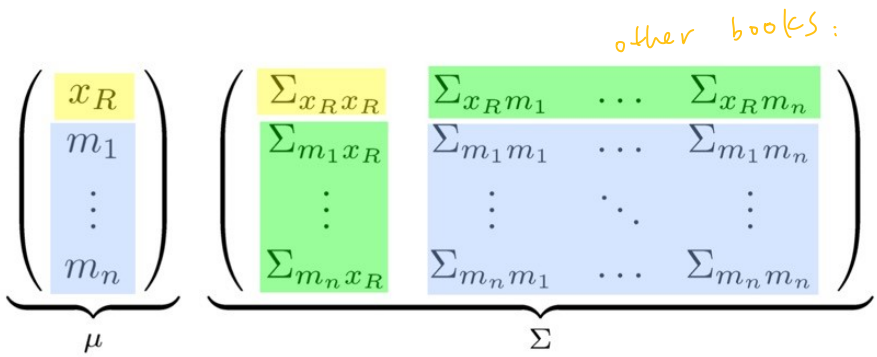
some books :

$$\underbrace{\begin{pmatrix} x \\ y \\ \theta \\ m_{1,x} \\ m_{1,y} \\ \vdots \\ m_{n,x} \\ m_{n,y} \end{pmatrix}}_{\mu} \quad \underbrace{\begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{x\theta} & \sigma_{xm_{1,x}} & \sigma_{xm_{1,y}} & \cdots & \sigma_{xm_{n,x}} & \sigma_{xm_{n,y}} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{y\theta} & \sigma_{ym_{1,x}} & \sigma_{ym_{1,y}} & \cdots & \sigma_{m_{n,x}} & \sigma_{m_{n,y}} \\ \sigma_{\theta x} & \sigma_{\theta y} & \sigma_{\theta\theta} & \sigma_{\theta m_{1,x}} & \sigma_{\theta m_{1,y}} & \cdots & \sigma_{\theta m_{n,x}} & \sigma_{\theta m_{n,y}} \\ \sigma_{m_{1,x}x} & \sigma_{m_{1,x}y} & \sigma_{\theta} & \sigma_{m_{1,x}m_{1,x}} & \sigma_{m_{1,x}m_{1,y}} & \cdots & \sigma_{m_{1,x}m_{n,x}} & \sigma_{m_{1,x}m_{n,y}} \\ \sigma_{m_{1,y}x} & \sigma_{m_{1,y}y} & \sigma_{\theta} & \sigma_{m_{1,y}m_{1,x}} & \sigma_{m_{1,y}m_{1,y}} & \cdots & \sigma_{m_{1,y}m_{n,x}} & \sigma_{m_{1,y}m_{n,y}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \sigma_{m_{n,x}x} & \sigma_{m_{n,x}y} & \sigma_{\theta} & \sigma_{m_{n,x}m_{1,x}} & \sigma_{m_{n,x}m_{1,y}} & \cdots & \sigma_{m_{n,x}m_{n,x}} & \sigma_{m_{n,x}m_{n,y}} \\ \sigma_{m_{n,y}x} & \sigma_{m_{n,y}y} & \sigma_{\theta} & \sigma_{m_{n,y}m_{1,x}} & \sigma_{m_{n,y}m_{1,y}} & \cdots & \sigma_{m_{n,y}m_{n,x}} & \sigma_{m_{n,y}m_{n,y}} \end{pmatrix}}_{\Sigma}$$

9

EKF SLAM: State Representation

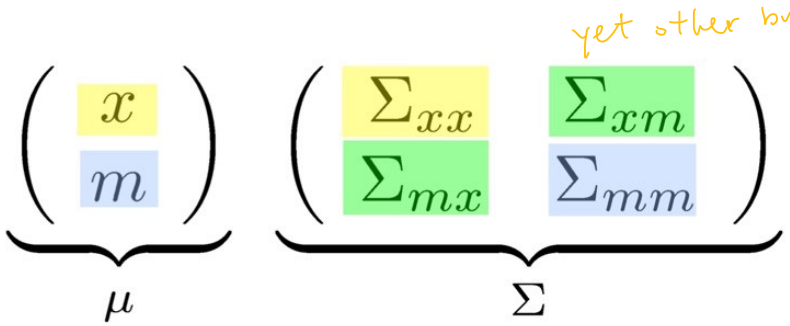
- More compactly



10

EKF SLAM: State Representation

- Even more compactly (note: $x_R \rightarrow x$)



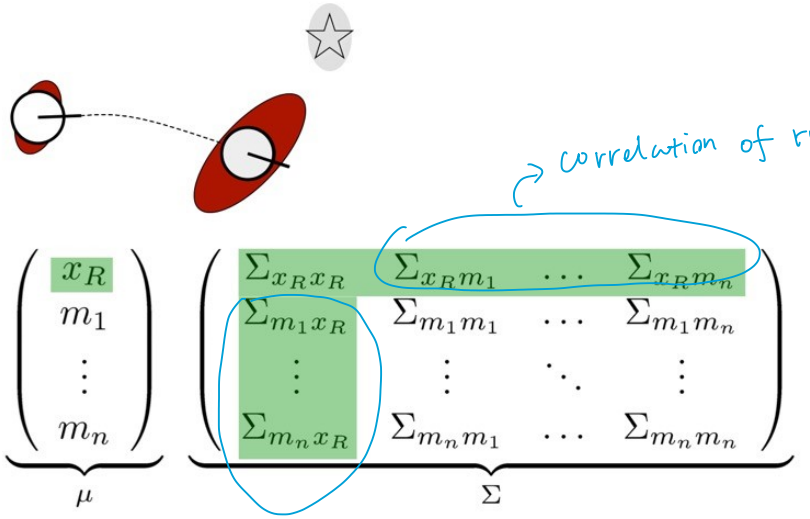
11

EKF SLAM: Filter Cycle

1. State prediction $\bar{\mu}_t, \bar{\Sigma}_t$
2. Measurement prediction $h(\bar{\mu}_t)$
3. Measurement
4. Data association
5. Update

12

EKF SLAM: State Prediction



correlation of robot pose and landmarks

Question: what is the TC

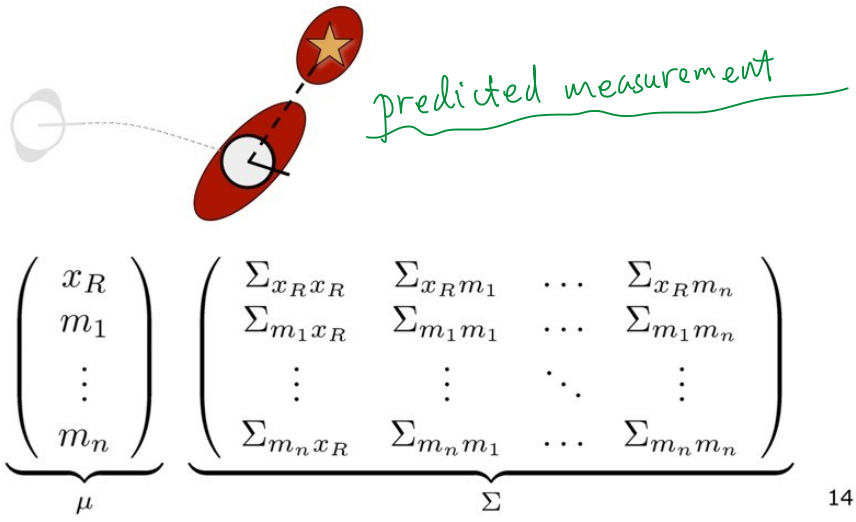
i.e: how many entrie

O(n)

↳ n is -

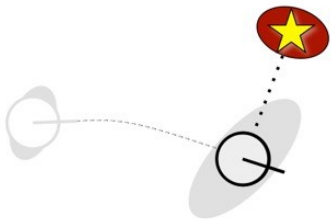
13

EKF SLAM: Measurement Prediction



14

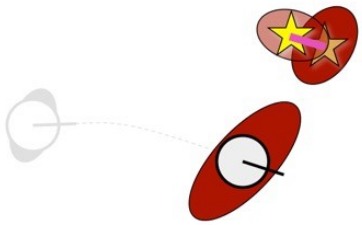
EKF SLAM: Obtained Measurement



$$\underbrace{\begin{pmatrix} x_R \\ m_1 \\ \vdots \\ m_n \end{pmatrix}}_{\mu} \quad \underbrace{\begin{pmatrix} \Sigma_{x_R x_R} & \Sigma_{x_R m_1} & \cdots & \Sigma_{x_R m_n} \\ \Sigma_{m_1 x_R} & \Sigma_{m_1 m_1} & \cdots & \Sigma_{m_1 m_n} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{m_n x_R} & \Sigma_{m_n m_1} & \cdots & \Sigma_{m_n m_n} \end{pmatrix}}_{\Sigma}$$

15

EKF SLAM: Data Association and Difference Between h(x) and z



leads to:
$(\Rightarrow K_t \Rightarrow \mu_t \& \Sigma_t)$

$$\underbrace{\begin{pmatrix} x_R \\ m_1 \\ \vdots \\ m_n \end{pmatrix}}_{\mu} \quad \underbrace{\begin{pmatrix} \Sigma_{x_R x_R} & \Sigma_{x_R m_1} & \cdots & \Sigma_{x_R m_n} \\ \Sigma_{m_1 x_R} & \Sigma_{m_1 m_1} & \cdots & \Sigma_{m_1 m_n} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{m_n x_R} & \Sigma_{m_n m_1} & \cdots & \Sigma_{m_n m_n} \end{pmatrix}}_{\Sigma}$$

16

EKF SLAM: Update Step

$$\underbrace{\begin{pmatrix} x_R \\ m_1 \\ \vdots \\ m_n \end{pmatrix}}_{\mu} \quad \underbrace{\begin{pmatrix} \Sigma_{x_R x_R} & \Sigma_{x_R m_1} & \cdots & \Sigma_{x_R m_n} \\ \Sigma_{m_1 x_R} & \Sigma_{m_1 m_1} & \cdots & \Sigma_{m_1 m_n} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{m_n x_R} & \Sigma_{m_n m_1} & \cdots & \Sigma_{m_n m_n} \end{pmatrix}}_{\Sigma}$$

17

EKF SLAM: Concrete Example

Setup

- Robot moves in the 2D plane
- Velocity-based motion model i.e. no odom, take pure vel cmd. (trans, ro
- Robot observes point landmarks
- Range-bearing sensor ⇒ laser range finder (bearing info).
- Known data association
- Known number of landmarks

18

Initialization

- Robot starts in its own reference frame (all landmarks unknown)
- 2N+3 dimensions

landmarks, robot pose

$$\mu_0 = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \end{pmatrix}^T$$

$$\Sigma_0 = \begin{pmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \infty & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \infty \end{pmatrix}$$

We don't know anything about landmarks

19

Extended Kalman Filter Algorithm

1: **Extended_Kalman_filter**$(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$:

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

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

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

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

6: $\Sigma_t = (I - K_t \, H_t) \, \bar{\Sigma}_t$

7: *return* μ_t, Σ_t

20

Prediction Step (Motion)

- Goal: Update state space based on the robot's motion [vel based motion model]
- Robot motion in the plane

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \underbrace{\begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} -\frac{v_t}{\omega_t}\sin\theta + \frac{v_t}{\omega_t}\sin(\theta + \omega_t \Delta t) \\ \frac{v_t}{\omega_t}\cos\theta - \frac{v_t}{\omega_t}\cos(\theta + \omega_t \Delta t) \\ \omega_t \, \Delta t \end{pmatrix}}_{g_{x,y,\theta}(u_t, (x,y,\theta)^T)}$$

↳ it's a truncatated version of g, as g itself is full version considering landmarks as well.

- How to map that to the 2N+3 dim space?

21

Update the State Space

- From the motion in the plane

From the motion in the plane

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} -\frac{v_t}{\omega_t}\sin\theta + \frac{v_t}{\omega_t}\sin(\theta+\omega_t\Delta t) \\ \frac{v_t}{\omega_t}\cos\theta - \frac{v_t}{\omega_t}\cos(\theta+\omega_t\Delta t) \\ \omega_t\,\Delta t \end{pmatrix}$$

- to the 2N+3 dimensional space

$$\begin{pmatrix} x' \\ y' \\ \theta' \\ \vdots \end{pmatrix} = \underbrace{\begin{pmatrix} x \\ y \\ \theta \\ \vdots \end{pmatrix} + \underbrace{\begin{pmatrix} 1 & 0 & 0 & \underbrace{0\ldots0}_{2N cols} \\ 0 & 1 & 0 & 0\ldots0 \\ 0 & 0 & 1 & 0\ldots0 \end{pmatrix}^T}_{F_x^T} \begin{pmatrix} -\frac{v_t}{\omega_t}\sin\theta + \frac{v_t}{\omega_t}\sin(\theta+\omega_t\Delta t) \\ \frac{v_t}{\omega_t}\cos\theta - \frac{v_t}{\omega_t}\cos(\theta+\omega_t\Delta t) \\ \omega_t\,\Delta t \end{pmatrix}}_{g(u_t,x_t)}$$

22

Extended Kalman Filter Algorithm

1: **Extended_Kalman_filter**$(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$**:**

2: ~~$\bar{\mu}_t = g(u_t, \mu_{t-1})$~~ **DONE**

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

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

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

6: $\Sigma_t = (I - K_t\,H_t)\,\bar{\Sigma}_t$

7: $return\ \mu_t, \Sigma_t$

23

Update Covariance

- The function g only affects the robot's motion and not the landmarks

Jacobian of the motion (3x3)

$$G_t = \begin{pmatrix} G_t^x & 0 \\ 0 & I \end{pmatrix}$$

The function doesn't

affect the part of landmarks.

Identity (2N x 2N)

Note 1: Based on the calculation below, G_t only has two elem non-zero, others are Identity matrix. (which takes consideration of linearization of heading)

Note 2: If it's linear func. the G_t should just be Identity Mat. as they are nonlinear funcs, those two elements are nonzero.

24

Jacobian of the Motion

$$G_t^x = \frac{\partial}{\partial(x,y,\theta)^T}\left[\begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} -\frac{v_t}{\omega_t}\sin\theta + \frac{v_t}{\omega_t}\sin(\theta+\omega_t\Delta t) \\ \frac{v_t}{\omega_t}\cos\theta - \frac{v_t}{\omega_t}\cos(\theta+\omega_t\Delta t) \\ \omega_t\,\Delta t \end{pmatrix}\right]$$

$$= I + \frac{\partial}{\partial(x,y,\theta)^T}\begin{pmatrix} \cdots \\ \cdots \\ \cdots \end{pmatrix}$$

$$= I + \begin{pmatrix} 0 & 0 & -\frac{v_t}{\omega_t}\cos\theta + \frac{v_t}{\omega_t}\cos(\theta+\omega_t\Delta t) \\ 0 & 0 & -\frac{v_t}{\omega_t}\sin\theta + \frac{v_t}{\omega_t}\sin(\theta+\omega_t\Delta t) \\ 0 & 0 & 0 \end{pmatrix}$$

nothing to do with $(x,y,\theta)^T$

25

Jacobian of the Motion

$$G_t^x = \frac{\partial}{\partial(x,y,\theta)^T}\left[\begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} -\frac{v_t}{\omega_t}\sin\theta + \frac{v_t}{\omega_t}\sin(\theta+\omega_t\Delta t) \\ \frac{v_t}{\omega_t}\cos\theta - \frac{v_t}{\omega_t}\cos(\theta+\omega_t\Delta t) \\ \omega_t\,\Delta t \end{pmatrix}\right]$$

$$= I + \frac{\partial}{\partial(x,y,\theta)^T}\begin{pmatrix} -\frac{v_t}{\omega_t}\sin\theta + \frac{v_t}{\omega_t}\sin(\theta+\omega_t\Delta t) \\ \frac{v_t}{\omega_t}\cos\theta - \frac{v_t}{\omega_t}\cos(\theta+\omega_t\Delta t) \\ \omega_t\,\Delta t \end{pmatrix}$$

26

Jacobian of the Motion

$$
\begin{aligned}
G_t^x &= \frac{\partial}{\partial (x,y,\theta)^T} \left[\begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} -\frac{v_t}{\omega_t}\sin\theta + \frac{v_t}{\omega_t}\sin(\theta+\omega_t\Delta t) \\ \frac{v_t}{\omega_t}\cos\theta - \frac{v_t}{\omega_t}\cos(\theta+\omega_t\Delta t) \\ \omega_t \Delta t \end{pmatrix} \right] \\
&= I + \frac{\partial}{\partial (x,y,\theta)^T} \begin{pmatrix} -\frac{v_t}{\omega_t}\sin\theta + \frac{v_t}{\omega_t}\sin(\theta+\omega_t\Delta t) \\ \frac{v_t}{\omega_t}\cos\theta - \frac{v_t}{\omega_t}\cos(\theta+\omega_t\Delta t) \\ \omega_t \Delta t \end{pmatrix} \\
&= I + \begin{pmatrix} 0 & 0 & -\frac{v_t}{\omega_t}\cos\theta + \frac{v_t}{\omega_t}\cos(\theta+\omega_t\Delta t) \\ 0 & 0 & -\frac{v_t}{\omega_t}\sin\theta + \frac{v_t}{\omega_t}\sin(\theta+\omega_t\Delta t) \\ 0 & 0 & 0 \end{pmatrix}
\end{aligned}
$$

27

Jacobian of the Motion

$$
\begin{aligned}
G_t^x &= \frac{\partial}{\partial (x,y,\theta)^T} \left[\begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} -\frac{v_t}{\omega_t}\sin\theta + \frac{v_t}{\omega_t}\sin(\theta+\omega_t\Delta t) \\ \frac{v_t}{\omega_t}\cos\theta - \frac{v_t}{\omega_t}\cos(\theta+\omega_t\Delta t) \\ \omega_t \Delta t \end{pmatrix} \right] \\
&= I + \frac{\partial}{\partial (x,y,\theta)^T} \begin{pmatrix} -\frac{v_t}{\omega_t}\sin\theta + \frac{v_t}{\omega_t}\sin(\theta+\omega_t\Delta t) \\ \frac{v_t}{\omega_t}\cos\theta - \frac{v_t}{\omega_t}\cos(\theta+\omega_t\Delta t) \\ \omega_t \Delta t \end{pmatrix} \\
&= I + \begin{pmatrix} 0 & 0 & -\frac{v_t}{\omega_t}\cos\theta + \frac{v_t}{\omega_t}\cos(\theta+\omega_t\Delta t) \\ 0 & 0 & -\frac{v_t}{\omega_t}\sin\theta + \frac{v_t}{\omega_t}\sin(\theta+\omega_t\Delta t) \\ 0 & 0 & 0 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & -\frac{v_t}{\omega_t}\cos\theta + \frac{v_t}{\omega_t}\cos(\theta+\omega_t\Delta t) \\ 0 & 1 & -\frac{v_t}{\omega_t}\sin\theta + \frac{v_t}{\omega_t}\sin(\theta+\omega_t\Delta t) \\ 0 & 0 & 1 \end{pmatrix}
\end{aligned}
$$

28

This Leads to the Update

1: **Extended_Kalman_filter**$(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$**:**

2: ~~$\bar{\mu}_t = g(u_t, \mu_{t-1})$~~ **Apply & DONE**

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

$$
\begin{aligned}
\bar{\Sigma}_t &= G_t \Sigma_{t-1} G_t^T + R_t \\
&= \begin{pmatrix} G_t^x & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} \Sigma_{xx} & \Sigma_{xm} \\ \Sigma_{mx} & \Sigma_{mm} \end{pmatrix} \begin{pmatrix} (G_t^x)^T & 0 \\ 0 & I \end{pmatrix} + R_t \\
&= \begin{pmatrix} G_t^x \Sigma_{xx} (G_t^x)^T & G_t^x \Sigma_{xm} \\ (G_t^x \Sigma_{xm})^T & \Sigma_{mm} \end{pmatrix} + R_t
\end{aligned}
$$

Note: we can perfectly see WHICH PART of the matrix are UPDATED.

this is large block and it's NOT UPDATED.

29

Extended Kalman Filter Algorithm

1: **Extended_Kalman_filter**$(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$**:**

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

3: $\bar{\Sigma}_t = G_t \, \Sigma_{t-1} \, G_t^T + R_t$ **DONE**

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

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

6: $\Sigma_t = (I - K_t \, H_t) \, \bar{\Sigma}_t$

7: $return \; \mu_t, \Sigma_t$

30

EKF SLAM:Prediction Step

EKF_SLAM_Prediction$(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, c_t, R_t)$**:**

2: $F_x = \begin{pmatrix} 1 & 0 & 0 & 0 \cdots 0 \\ 0 & 1 & 0 & 0 \cdots 0 \\ 0 & 0 & 1 & 0 \cdots 0 \end{pmatrix}$

projection matrix s.t. 3 → 2N+3

3: $\bar{\mu}_t = \mu_{t-1} + F_x^T \begin{pmatrix} -\frac{v_t}{\omega_t} \sin \mu_{t-1,\theta} + \frac{v_t}{\omega_t} \sin(\mu_{t-1,\theta} + \omega_t \Delta t) \\ \frac{v_t}{\omega_t} \cos \mu_{t-1,\theta} - \frac{v_t}{\omega_t} \cos(\mu_{t-1,\theta} + \omega_t \Delta t) \\ \omega_t \Delta t \end{pmatrix}$

pred MEAN, old MEAN

4: $G_t = I + F_x^T \begin{pmatrix} 0 & 0 & -\frac{v_t}{\omega_t} \cos \mu_{t-1,\theta} + \frac{v_t}{\omega_t} \cos(\mu_{t-1,\theta} + \omega_t \Delta t) \\ 0 & 0 & -\frac{v_t}{\omega_t} \sin \mu_{t-1,\theta} + \frac{v_t}{\omega_t} \sin(\mu_{t-1,\theta} + \omega_t \Delta t) \\ 0 & 0 & 0 \end{pmatrix} F_x$

5: $\bar{\Sigma}_t = G_t \, \Sigma_{t-1} \, G_t^T + \underbrace{F_x^T R_t^x F_x}_{R_t}$

only the 3×3 part changes, all the rest remain zero.

31

Extended Kalman Filter Algorithm

1: **Extended_Kalman_filter**$(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$:

2: ~~$\bar{\mu}_t = g(u_t, \mu_{t-1})$~~ **DONE**

3: ~~$\bar{\Sigma}_t = G_t \, \Sigma_{t-1} \, G_t^T + R_t$~~ **Apply & DONE**

Correction steps:

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

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

6: $\Sigma_t = (I - K_t \, H_t) \, \bar{\Sigma}_t$

7: *return* μ_t, Σ_t

32

EKF SLAM: Correction Step

- Known data association
- $c_t^i = j$: i-th measurement at time t observes the landmark with index j
- Initialize landmark if unobserved
- Compute the expected observation
- Compute the Jacobian of h
- Proceed with computing the Kalman gain

33

Range-Bearing Observation

[obtained observation]

- Range-Bearing observation $z_t^i = (r_t^i, \phi_t^i)^T$

- If landmark has not been observed

$$\begin{pmatrix} \bar{\mu}_{j,x} \\ \bar{\mu}_{j,y} \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{t,x} \\ \bar{\mu}_{t,y} \end{pmatrix} + \begin{pmatrix} r_t^i \cos(\phi_t^i + \bar{\mu}_{t,\theta}) \\ r_t^i \sin(\phi_t^i + \bar{\mu}_{t,\theta}) \end{pmatrix}$$

observed location of landmark j | estimated robot's location | relative measurement

34

Expected Observation

- Compute expected observation according to the current estimate

$$\begin{aligned} \delta &= \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{j,x} - \bar{\mu}_{t,x} \\ \bar{\mu}_{j,y} - \bar{\mu}_{t,y} \end{pmatrix} \\ q &= \delta^T \delta \\ \hat{z}_t^i &= \begin{pmatrix} \sqrt{q} \\ \operatorname{atan2}(\delta_y, \delta_x) - \bar{\mu}_{t,\theta} \end{pmatrix} \\ &= h(\bar{\mu}_t) \end{aligned}$$

(diff bet rob pose and landmark) (square of dist diff)

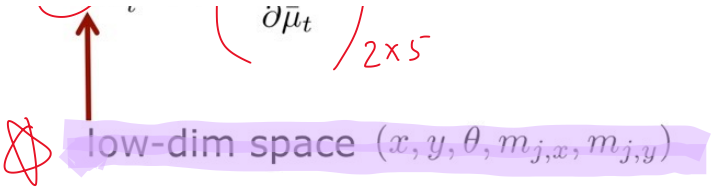
35

Jacobian for the Observation

- Based on $\delta = \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{j,x} - \bar{\mu}_{t,x} \\ \bar{\mu}_{j,y} - \bar{\mu}_{t,y} \end{pmatrix}$, $q = \delta^T \delta$, $\hat{z}_t^i = \begin{pmatrix} \sqrt{q} \\ \operatorname{atan2}(\delta_y, \delta_x) - \bar{\mu}_{t,\theta} \end{pmatrix}$

- Compute the Jacobian

low $H_t^i = \left(\frac{\partial h(\bar{\mu}_t)}{} \right.$



36

Jacobian for the Observation

- Based on

$$\begin{aligned} \delta &= \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{j,x} - \bar{\mu}_{t,x} \\ \bar{\mu}_{j,y} - \bar{\mu}_{t,y} \end{pmatrix} \\ q &= \delta^T \delta \\ \hat{z}_t^i &= \begin{pmatrix} \sqrt{q} \\ \text{atan2}(\delta_y, \delta_x) - \bar{\mu}_{t,\theta} \end{pmatrix} \end{aligned}$$

- Compute the Jacobian

state vector dim: $3+2N$

$$\begin{aligned} {}^{\text{low}}H_t^i &= \frac{\partial h(\bar{\mu}_t)}{\partial \bar{\mu}_t} \\ &= \begin{pmatrix} \frac{\partial \sqrt{q}}{\partial x} & \frac{\partial \sqrt{q}}{\partial y} & \cdots \\ \frac{\partial \text{atan2}(\ldots)}{\partial x} & \frac{\partial \text{atan2}(\ldots)}{\partial y} & \cdots \end{pmatrix}_{2\times 5} \end{aligned}$$

only affects x, y; everything else is 0

37

The First Component

- Based on

$$\begin{aligned} \delta &= \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{j,x} - \bar{\mu}_{t,x} \\ \bar{\mu}_{j,y} - \bar{\mu}_{t,y} \end{pmatrix} \\ q &= \delta^T \delta \\ \hat{z}_t^i &= \begin{pmatrix} \sqrt{q} \\ \text{atan2}(\delta_y, \delta_x) - \bar{\mu}_{t,\theta} \end{pmatrix} \end{aligned}$$

- We obtain (by applying the chain rule)

$$\begin{aligned} \frac{\partial \sqrt{q}}{\partial x} &= \frac{1}{2} \frac{1}{\sqrt{q}} 2\,\delta_x\,(-1) \\ &= \frac{1}{q}\,(-\sqrt{q}\delta_x) \end{aligned}$$

38

Jacobian for the Observation

- Based on

$$\begin{aligned} \delta &= \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{j,x} - \bar{\mu}_{t,x} \\ \bar{\mu}_{j,y} - \bar{\mu}_{t,y} \end{pmatrix} \\ q &= \delta^T \delta \\ \hat{z}_t^i &= \begin{pmatrix} \sqrt{q} \\ \text{atan2}(\delta_y, \delta_x) - \bar{\mu}_{t,\theta} \end{pmatrix} \end{aligned}$$

- Compute the Jacobian

$$\begin{aligned} {}^{\text{low}}H_t^i &= \frac{\partial h(\bar{\mu}_t)}{\partial \bar{\mu}_t} \\ &= \frac{1}{q} \begin{pmatrix} -\sqrt{q}\delta_x & -\sqrt{q}\delta_y & 0 & +\sqrt{q}\delta_x & \sqrt{q}\delta_y \\ \delta_y & -\delta_x & -q & -\delta_y & \delta_x \end{pmatrix} \end{aligned}$$

39

Jacobian for the Observation

- Use the computed Jacobian

$${}^{\text{low}}H_t^i = \frac{1}{q} \begin{pmatrix} -\sqrt{q}\delta_x & -\sqrt{q}\delta_y & 0 & +\sqrt{q}\delta_x & \sqrt{q}\delta_y \\ \delta_y & -\delta_x & -q & -\delta_y & \delta_x \end{pmatrix}$$

- map it to the high dimensional space

$$H_t^i = {}^{\text{low}}H_t^i \, F_{x,j}$$

$$F_{x,j} = \begin{pmatrix} 1 & 0 & 0 & \underbrace{0 \cdots 0}_{2j-2} & 0 & 0 & \underbrace{0 \cdots 0}_{2N-2j} \\ 0 & 1 & 0 & & 0 & 0 & \\ 0 & 0 & 1 & & 0 & 0 & \\ 0 & 0 & 0 & & 1 & 0 & \\ 0 & 0 & 0 & & 0 & 1 & \end{pmatrix}$$

rob state

mapping func from low dim to high dim.

jth landmark observed.

40

Next Steps as Specified...

1: **Extended_Kalman_filter**$(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$**:**

2: ~~$\bar{\mu}_t = g(u_t, \mu_{t-1})$~~ **DONE**

3: ~~$\bar{\Sigma}_t = G_t \, \Sigma_{t-1} \, G_t^T + R_t$~~ **DONE**

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

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

6: $\Sigma_t = (I - K_t \, H_t) \, \bar{\Sigma}_t$

7: $return \; \mu_t, \Sigma_t$

Q_t: Tells us how certain we're about our sensor(s), based the specs of our sensor(s)

41

Extended Kalman Filter Algorithm

1: **Extended_Kalman_filter**$(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$**:**

2: ~~$\bar{\mu}_t = g(u_t, \mu_{t-1})$~~ **DONE**

3: ~~$\bar{\Sigma}_t = G_t \, \Sigma_{t-1} \, G_t^T + R_t$~~ **DONE**

4: ~~$K_t = \bar{\Sigma}_t \, H_t^T (H_t \, \bar{\Sigma}_t \, H_t^T + Q_t)^{-1}$~~ **Apply & DONE**

5: ~~$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$~~ **Apply & DONE**

6: ~~$\Sigma_t = (I - K_t \, H_t) \, \bar{\Sigma}_t$~~ **Apply & DONE**

7: $return \; \mu_t, \Sigma_t$

42

EKF SLAM – Correction (1/2)

EKF_SLAM_Correction

6: $Q_t = \begin{pmatrix} {\sigma_r}^2 & 0 \\ 0 & {\sigma_\phi}^2 \end{pmatrix}$

7: *for all observed features* $z_t^i = (r_t^i, \phi_t^i)^T$ *do*

8: $j = c_t^i$

9: *if landmark* j *never seen before*

10: $\begin{pmatrix} \bar{\mu}_{j,x} \\ \bar{\mu}_{j,y} \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{t,x} \\ \bar{\mu}_{t,y} \end{pmatrix} + \begin{pmatrix} r_t^i \cos(\phi_t^i + \bar{\mu}_{t,\theta}) \\ r_t^i \sin(\phi_t^i + \bar{\mu}_{t,\theta}) \end{pmatrix}$

11: *endif*

12: $\delta = \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{j,x} - \bar{\mu}_{t,x} \\ \bar{\mu}_{j,y} - \bar{\mu}_{t,y} \end{pmatrix}$

13: $q = \delta^T \delta$

14: $\hat{z}_t^i = \begin{pmatrix} \sqrt{q} \\ \text{atan2}(\delta_y, \delta_x) - \bar{\mu}_{t,\theta} \end{pmatrix}$

Note: ① We use a loop to carry out all the measurement outputs. ② We can do it in one shot

43

EKF SLAM – Correction (2/2)

15: $F_{x,j} = \begin{pmatrix} 1 & 0 & 0 & \underbrace{0 \cdots 0}_{2j-2} & 0 & 0 & \underbrace{0 \cdots 0}_{2N-2j} \\ 0 & 1 & 0 & & & & \\ 0 & 0 & 1 & & & & \\ 0 & 0 & 0 & & & & \\ 0 & 0 & 0 & & & & \end{pmatrix}$

16: $H_t^i = \frac{1}{q} \begin{pmatrix} -\sqrt{q}\delta_x & -\sqrt{q}\delta_y & 0 & +\sqrt{q}\delta_x & \sqrt{q}\delta_y \\ \delta_y & -\delta_x & -q & -\delta_y & +\delta_x \end{pmatrix} F_{x,j}$

17: $K_t^i = \bar{\Sigma}_t \, H_t^{iT} (H_t^i \, \bar{\Sigma}_t \, H_t^{iT} + Q_t)^{-1}$

18: $\bar{\mu}_t = \bar{\mu}_t + K_t^i (z_t^i - \hat{z}_t^i)$

19: $\bar{\Sigma}_t = (I - K_t^i \, H_t^i) \, \bar{\Sigma}_t$

20: *endfor*

21: $\mu_t = \bar{\mu}_t$

22: $\Sigma_t = \bar{\Sigma}_t$

23: *return* μ_t, Σ_t

44

Implementation Notes

- Measurement update in a single step requires only one full belief update
- Always normalize the angular components ⇒ otherwise, the angles will wrap around during iterations.
- You may not need to create the F matrices explicitly (e.g., in Octave)
  ↳ As a good practice in implementation.

⇒ disadvantage: ① dim changes dur
it can be subopti
② correlation bet
landmarks of s
(side note, i

VERY BAD!
↑ LEADS TO WIRED EFFECTS

45

Done!

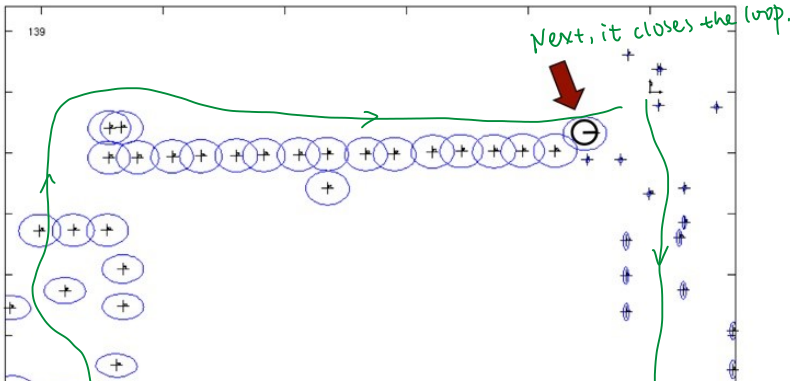
46

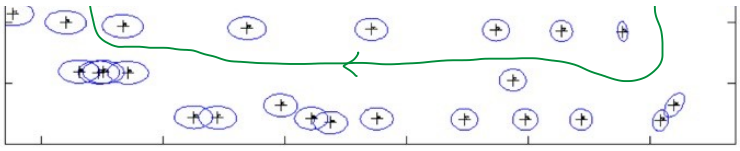
Loop Closing

- Loop closing means recognizing an already mapped area
- Data association under
  - high ambiguity
  - possible environment symmetries
- Uncertainties **collapse** after a loop closure (whether the closure was correct or not)

47

Before the Loop Closure

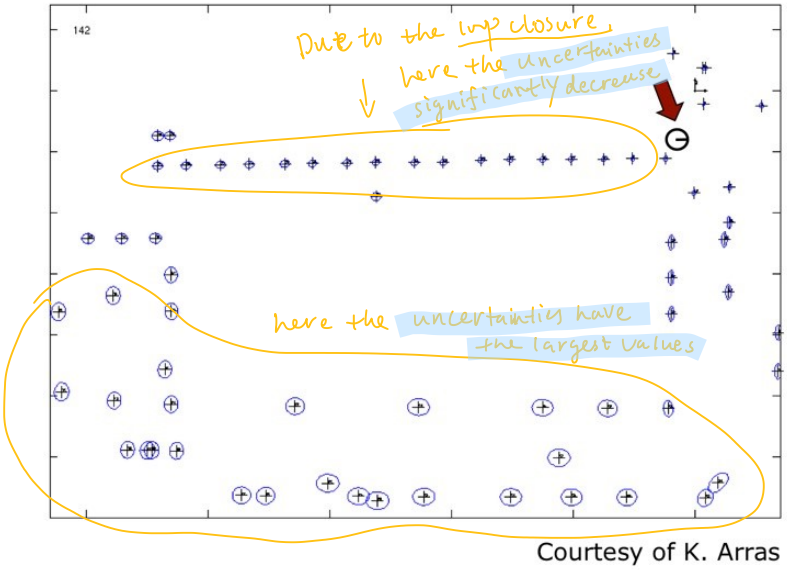




Courtesy of K. Arras

48

After the Loop Closure



Courtesy of K. Arras

49

Loop Closures in SLAM

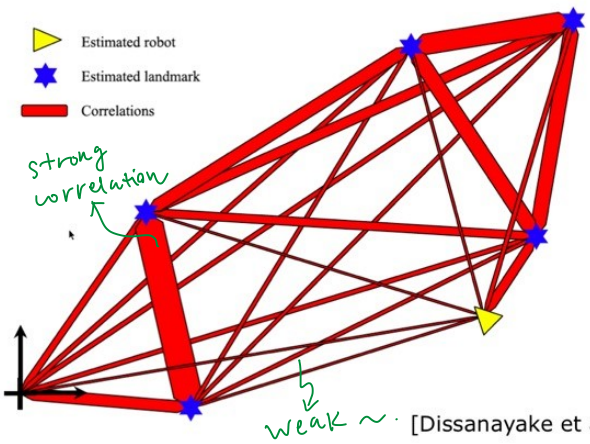
- Loop closing **reduces** the uncertainty in robot and landmark estimates
- This can be exploited when exploring an environment for the sake of better (e.g. more accurate) maps
- **Wrong loop closures lead to filter divergence**

VERY CRITICAL!

50

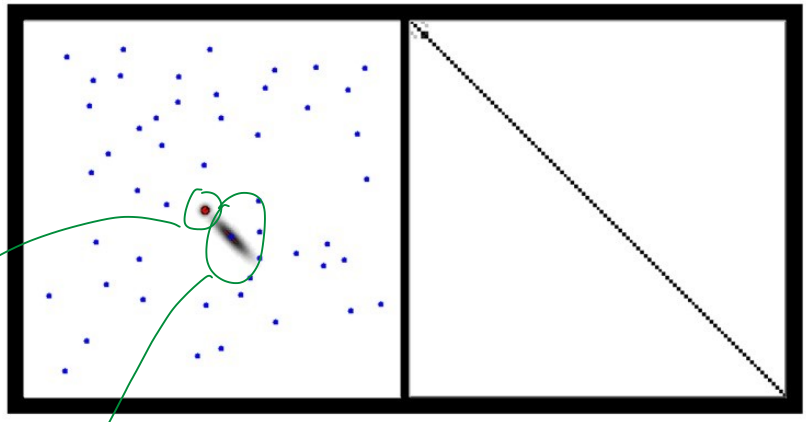
EKF SLAM Correlations

- In the limit, the landmark estimates become **fully correlated**



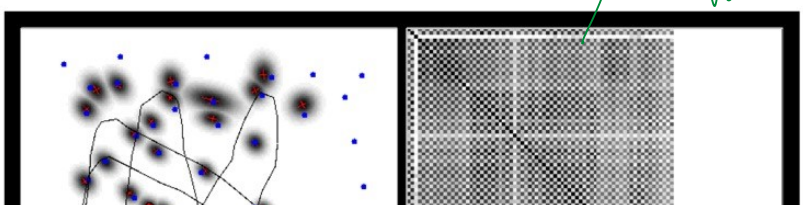
[Dissanayake et al., 2001] 51

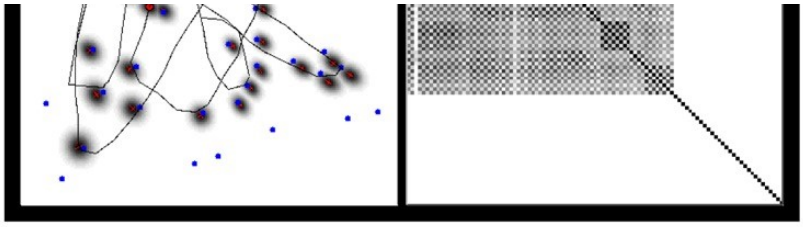
EKF SLAM Correlations



Courtesy of M. Montemerlo 52

EKF SLAM Correlations

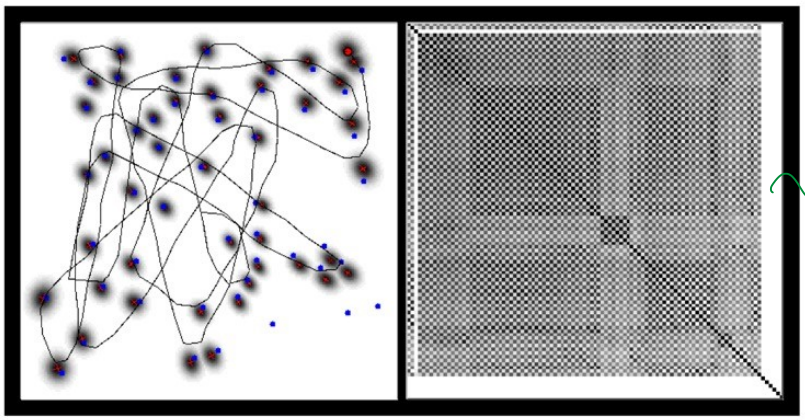




Map Correlation matrix

Courtesy of M. Montemerlo 53

EKF SLAM Correlations



Map Correlation matrix

Courtesy of M. Montemerlo 54

weird check board pattern.

WHY?

→ Strong correlation between x-x & y-location correlation, but weak ~ bet

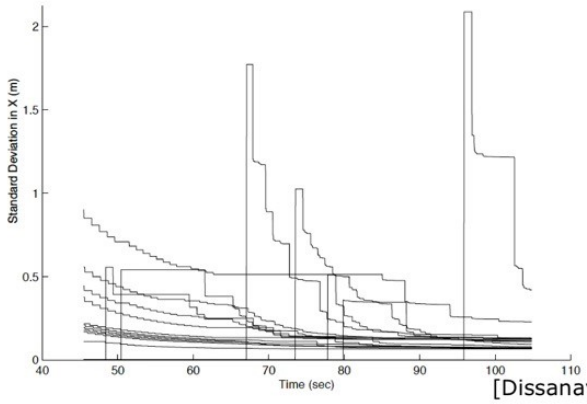
EKF SLAM Correlations

- The correlation between the robot's pose and the landmarks **cannot** be ignored → Any approaches that don't handle this very carefully is very likely to FAIL
- Assuming independence generates too optimistic estimates of the uncertainty

[Castellanos et al., 1997] 55

EKF SLAM Uncertainties

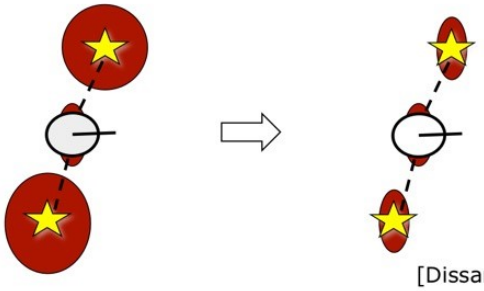
- The **determinant** of any sub-matrix of the map covariance matrix **decreases monotonically**
- New landmarks are initialized with **maximum uncertainty**



[Dissanayake et al., 2001] 56

EKF SLAM in the Limit

- In the limit, the covariance associated with any single landmark location estimate is determined only by the initial covariance in the vehicle location estimate. → You can't get more confidence than your initial uncertainties.



[Dissanayake et al., 2001] 57

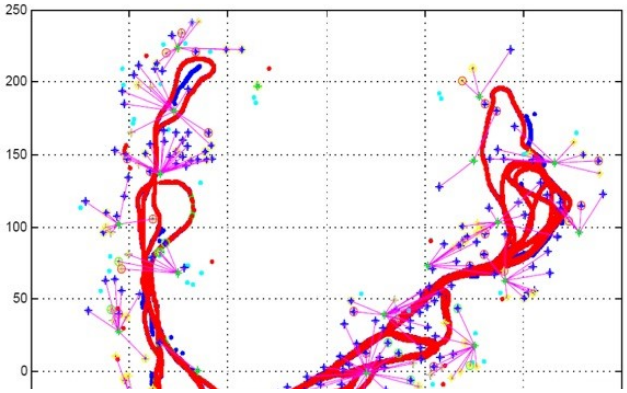
Example: Victoria Park Dataset

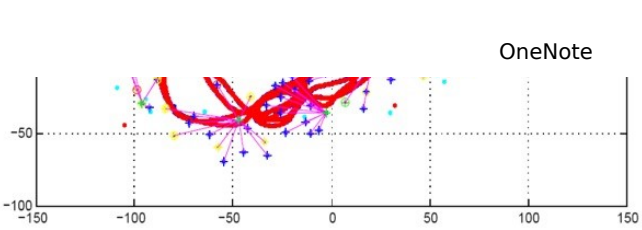
Courtesy of E. Nebot 58

Victoria Park: Data Acquisition

Courtesy of E. Nebot 59

Victoria Park: EKF Estimate

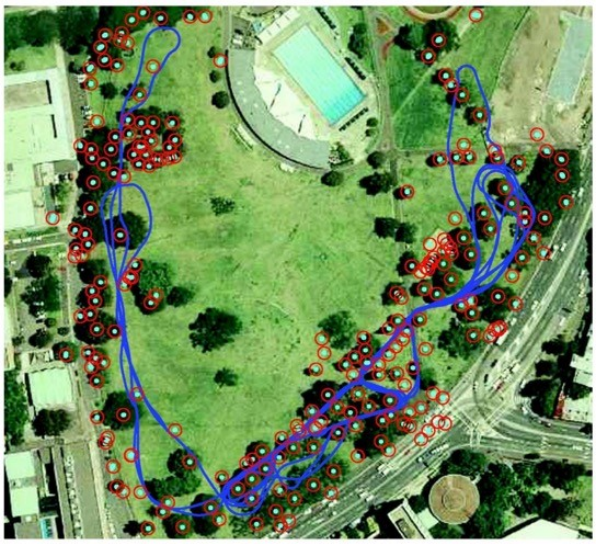




Courtesy of E. Nebot

60

Victoria Park: Landmarks



Courtesy of E. Nebot

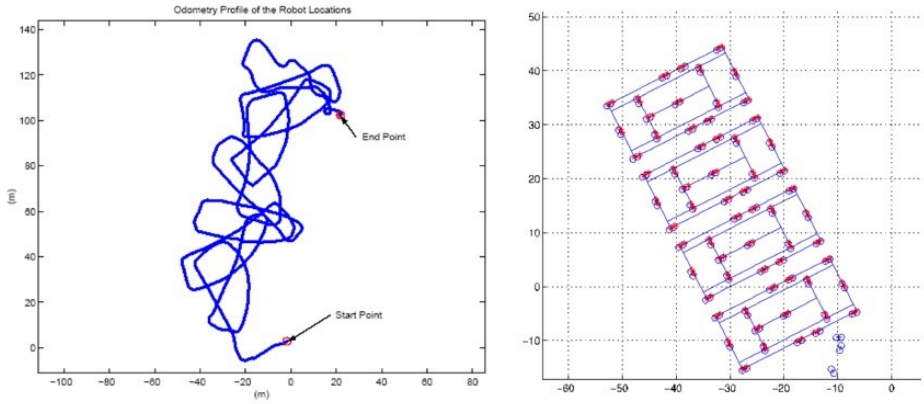
61

Example: Tennis Court Dataset

Courtesy of J. Leonard and M. Walter

62

EKF SLAM on a Tennis Court



odometry estimated trajectory

Courtesy of J. Leonard and M. Walter 63

EKF SLAM Complexity

- Cubic complexity depends only on the measurement dimensionality
- Cost per step: dominated by the number of landmarks: $O(n^2)$
- Memory consumption: $O(n^2)$
- The EKF becomes computationally intractable for large maps!

64

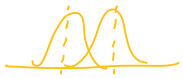
EKF SLAM Summary

- The first SLAM solution
- Convergence proof for the linear Gaussian case
- Can diverge if non-linearities are large

(and the reality is non-linear...)

- Can deal only with a single mode
- Successful in medium-scale scenes
- Approximations exists to reduce the computational complexity

bi-mode cannot be handled by



P.S. submap ... such as only maintain local maps and stitches to global map.

65

Literature

EKF SLAM

- Thrun et al.: "Probabilistic Robotics", Chapter 10

66