Dept of Industrial Engineering & Management

Intelligent Robotics Systems

Armin Biess

May 24, 2018

Ben-Gurion University of the Negev

Robot Mapping

Lecture Notes

- Lecture Notes: *Robot Mapping* by Cyrill Stachniss
- Probabilistic Robotics - Chapter 10: EKF SLAM

News from the robotics world

- RoboFly Is the First Wireless Insectoid Robot to Take Flight

  clip

- SpotMini

  clip

Introduction

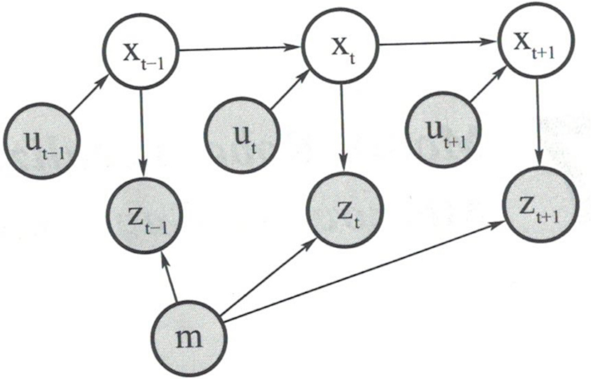
- only for specific environments maps are available
- many domains do not come with the luxury of a map
- even if blueprints of buildings are available, they do not contain furniture (e.g. vacuum cleaner)
- a real autonomous system should be able to learn the map

Challenges

- the space of all possibe maps is huge
- maps are defined over continuous space and thus have infinite dimensions
- even when discretizing space maps can be easily described by 10^5 variables and more
- full posteriors over maps are difficult to compute

Graphical model of mobile robot localization

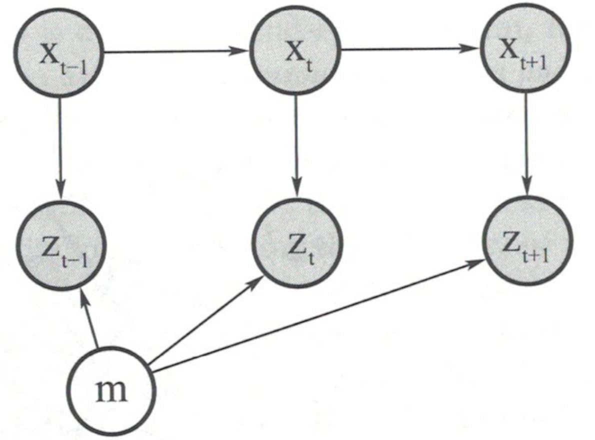
- map known, pose unknown



example: MCL

Graphical model of mapping with known poses

- pose known, map unknown



example: Occupancy grid mapping (chapter 9, Prob Rob)

Simultaneous Mapping and Localization (SLAM)

- pose unknown, map unknown
- building a map and locating the robot in the map at the same time
- chicken-and-egg problem

Problem formulation: SLAM

given

- robot's control

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

- robot's observations

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

wanted

- robot's path

$$x_{1:T} = (x_1, x_2, \ldots, x_T)$$

- map of the environment

$$m$$

What is SLAM?

Estimate the robot's path and map:

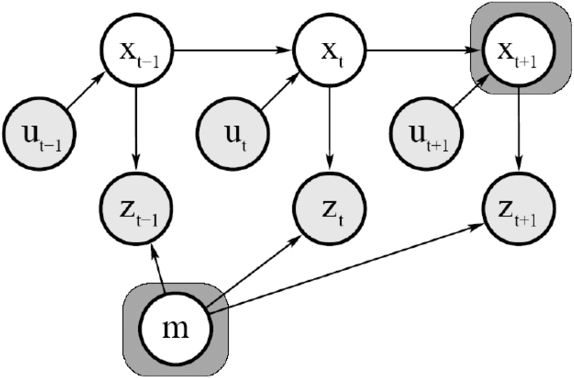
$$p(x_{0:T}, m \mid z_{1:T}, u_{1:T})$$

distribution path map given observations controls

Two main forms of SLAM

- **online SLAM** problem
- **full SLAM** problem

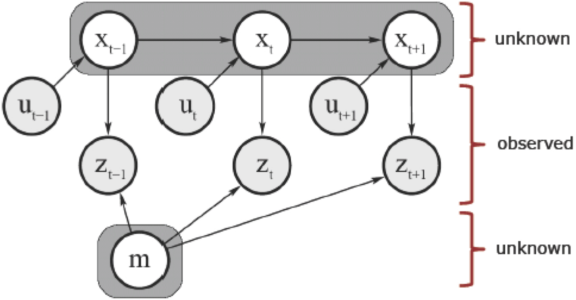
Graphical Model of online SLAM



Online SLAM problem: Estination of the posterior over the **momentary** pose x_t along with the map m:

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

Graphical Model of full SLAM



Full SLAM problem: Estination of the posterior over the **entire** path of poses $x_{1:t}$ along with the map m:

$$p(x_{1:t}, m | z_{1:t}, u_{1:t})$$

The online SLAM is obtained from the full SLAM by marginalization

$$p(x_t, m | z_{1:t}, u_{1:t}) = \int \int \cdots \int p(x_{1:t}, m | z_{1:t}, u_{1:t}) dx_1 dx_2 \ldots dx_{t-1}$$

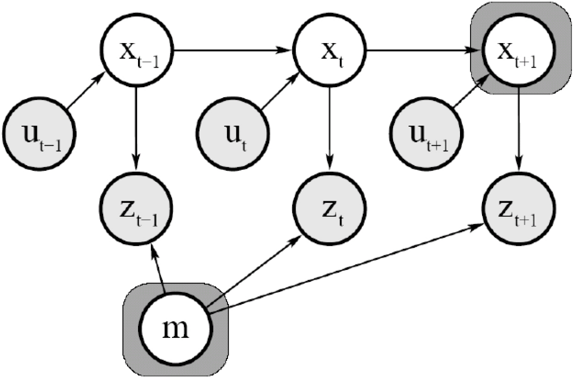
Three main paradigms

leading to different variants of SLAM algorithms

- Kalman: EKF-SLAM (this lecture - chapter 10, Prob Rob)
- Particle: Fast SLAM (chapter 13, Prob Rob)
- Graph: Graph SLAM (chapter 11, Prob Rob)

EKF for online SLAM

- The Kalman filter provides a solution to the online SLAM problem, i.e., $p(x_t, m | z_{1:T}, u_{1:T})$
- wer are interest in the **current** pose of the robot - not the full trajectory - and how the environment looks like.



EKF SLAM

- application of the EKF to SLAM
- estimate robots pose and location of features in the environment
- EKF SLAM uses feature-based maps (smaller than 1000 landmarks)
- assumption: known correspondence
- Gaussian noise assumption for robot motion and perception
- state space is

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

Extended Kalman filter algorithm

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

$\bar{\mu}_t = g(u_t, \mu_{t-1})$
$\bar{\Sigma}_t = G_t\, \Sigma_{t-1}\, G_t^T + R_t$

$K_t = \bar{\Sigma}_t\, H_t^T (H_t\, \bar{\Sigma}_t\, H_t^T + Q_t)^{-1}$
$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$
$\Sigma_t = (I - K_t\, H_t)\, \bar{\Sigma}_t$
return μ_t, Σ_t

EKF SLAM: State representation

- map with n landmarks: $(3+2n)$-dimensional Gaussian
- belief is represented by

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

EKF SLAM: State representation

- more compactly

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

EKF SLAM: State representation

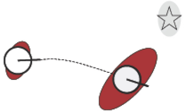
- even more compactly

$$\underbrace{\begin{pmatrix} x \\ m \end{pmatrix}}_{\mu} \quad \underbrace{\begin{pmatrix} \Sigma_{xx} & \Sigma_{xm} \\ \Sigma_{mx} & \Sigma_{mm} \end{pmatrix}}_{\Sigma}$$

EKF SLAM: Filter cycle

1. State prediction
2. Measurement prediction
3. Measurement
4. Data association
5. Update

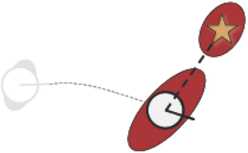
EKF SLAM: State prediction



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

- **state vector**
  motion commands only change the pose of the robot, not the locations of the landmarks (robot does not bump into the landmarks)
- **covariance**
  covariance of robot is changing
  correlations between robot pose and landmarks positions are changing

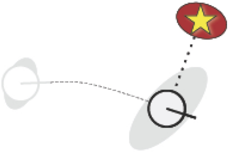
EKF SLAM: Measurement prediction



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

robot computes the predicted measurement (dashed line)

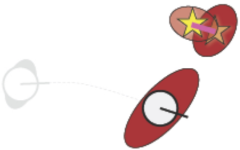
EKF SLAM: Obtained measurement



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

robot obtains real measurement

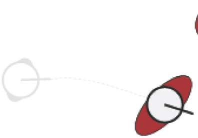
EKF SLAM: Data association and difference between predicted measurement $h(x)$ and obtained measurement z



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

robot needs to make a data association: to which landmark did the measurement belong to?
and then compare predicted and obtained measurement

EKF SLAM: Update step



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

all state and covariance estimates are updated

EKF SLAM: Concrete example

Setup

- Robot moves in the plane
- Velocity-based motion model
- Robot observes point landmarks
- Range-bearing sensor
- Known data association
- Known number of landmarks

EKF SLAM: Initialization

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

$$\mu_0 = (0\ 0\ 0\ \ldots\ 0)^T$$

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

Extended Kalman Filter Algorithm

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

$\bar{\mu}_t = g(u_t, \mu_{t-1})$
$\bar{\Sigma}_t = G_t\, \Sigma_{t-1}\, G_t^T + R_t$

$K_t = \bar{\Sigma}_t\, H_t^T (H_t\, \bar{\Sigma}_t\, H_t^T + Q_t)^{-1}$
$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$
$\Sigma_t = (I - K_t\, H_t)\, \bar{\Sigma}_t$
return μ_t, Σ_t

EKF SLAM: Prediction Step (Motion)

- Goal: Update state space based on the robots motion
- Robot motion in the plane

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} \frac{r}{2}(u_l + u_r)\cos\theta\Delta t \\ \frac{r}{2}(u_l + u_r)\sin\theta\Delta t \\ \frac{r}{L}(u_l - u_r)\ \Delta t \end{pmatrix}$$

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

EKF SLAM: Augment the state space

- from the motion in the plane

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} \frac{r}{2}(u_l + u_r)\cos\theta\Delta t \\ \frac{r}{2}(u_l + u_r)\sin\theta\Delta t \\ \frac{r}{L}(u_l - u_r)\ \Delta t \end{pmatrix}$$

- to the $2N+3$ dimensional space

$$\begin{pmatrix} x' \\ y' \\ \theta' \\ \vdots \end{pmatrix} = \underbrace{\begin{pmatrix} x \\ y \\ \theta \\ \vdots \end{pmatrix} + \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ & & & & \underbrace{}_{2N\ cols} & \end{pmatrix}^T}_{F_x^T} \begin{pmatrix} \frac{r}{2}(u_l + u_r)\cos\theta\Delta t \\ \frac{r}{2}(u_l + u_r)\sin\theta\Delta t \\ \frac{r}{L}(u_l - u_r)\ \Delta t \end{pmatrix}}_{g(u_t, x_{t-1})}$$

matrix sizes:
$((2N+3)\times 1) = ((2N+3)\times 1) + ((2N+3)\times 3)\times(3\times 1)$

EKF SLAM: Full motion model

- full motion model with noise

$$\begin{pmatrix} x' \\ y' \\ \theta' \\ \vdots \end{pmatrix} = \underbrace{\begin{pmatrix} x \\ y \\ \theta \\ \vdots \end{pmatrix} + \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & \underbrace{\dots}_{2N\ cols} & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \end{pmatrix}^T}_{F_x^T} \begin{pmatrix} \frac{r}{2}(u_l + u_r)\cos\theta\Delta t \\ \frac{r}{2}(u_l + u_r)\sin\theta\Delta t \\ \frac{r}{L}(u_l - u_r)\Delta t \end{pmatrix}}_{g(u_t, x_{t-1})}$$

$$+ \quad \mathcal{N}(0, R_t)$$

where $R_t = F_x^T R_t^x F_x \in \mathbb{R}^{(2N+3)\times(2N+3)}$ extends the covariance matrix to the dimension of the full state vector squared. The prediction covariance of the pose is $R_t^x \in \mathbb{R}^{3\times 3}$.

landmarks are not affected: $m'_{1,x} = m_{1,x}, \dots$

Extended Kalman Filter Algorithm

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

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

EKF SLAM: Update covariance step

Jacobian:

$$G_t := \frac{\partial g(u_t, x_{t-1})}{\partial x_{t-1}}\Big|_{x_{t-1}=\mu_{t-1}}$$

The Jacobian is a $(2N+3) \times (2N+3)$ matrix:

$$G_t \quad = \quad I + F_x^T G_t^x F_x$$

with

$$G_t^x \quad = \quad \begin{pmatrix} 0 & 0 & -\frac{r}{2}(u_l + u_r)\sin \mu_{t-1,\theta}\Delta t \\ 0 & 0 & \frac{r}{2}(u_l + u_r)\cos \mu_{t-1,\theta}\Delta t \\ 0 & 0 & 0 \end{pmatrix},$$

where $F_x \in \mathbb{R}^{3\times(2N+3)}$ and $I \in \mathbb{R}^{3\times(2N+3)}$ is the identity matrix and $\mu_{t-1,\theta}$ defines the θ-component of the vector μ_{t-1}.

An alternative way of writing the Jacobian is

$$G_t = \begin{pmatrix} \tilde{G}_t^x & 0 \\ 0 & I_{2N\times 2N} \end{pmatrix}$$

with $\tilde{G}_t^x = \frac{\partial g}{\partial(x\ y\ \theta)} = \begin{pmatrix} 1 & 0 & -\frac{r}{2}(u_l + u_r)\sin \mu_{t-1,\theta}\Delta t \\ 0 & 1 & \frac{r}{2}(u_l + u_r)\cos \mu_{t-1,\theta}\Delta t \\ 0 & 0 & 1 \end{pmatrix}$,.

Extended Kalman Filter Algorithm

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

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

Extended Kalman Filter Algorithm

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

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

EKF-SLAM: Prediction of covariance matrix

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

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

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

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

EKF-SLAM: Prediction of covariance matrix

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

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

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

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

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

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

return μ_t, Σ_t

EKF-SLAM: Prediction-step

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

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

3: $$\bar{\mu}_t = \mu_{t-1} + F_x^T \begin{pmatrix} \frac{r}{2}(u_l + u_r) \cos \mu_{t-1,\theta} \Delta t \\ \frac{r}{2}(u_l + u_r) \sin \mu_{t-1,\theta} \Delta t \\ \frac{r}{L}(u_l - u_r)\ \Delta t \end{pmatrix}$$

4: $$G_t = I + F_x^T \begin{pmatrix} 0 & 0 & -\frac{r}{2}(u_l + u_r) \sin \mu_{t-1,\theta} \Delta t \\ 0 & 0 & \frac{r}{2}(u_l + u_r) \cos \mu_{t-1,\theta} \Delta t \\ 0 & 0 & 0 \end{pmatrix} F_x$$

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

Extended Kalman Filter Algorithm

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

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

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

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

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

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

return μ_t, Σ_t

EKF SLAM: Correction

- next we need to analyze the measurement model $z_t = h(x_t)$
- compute expected observation $h(\bar{\mu}_t)$
- compute Jacobian H_t of $h(x_t)$
- then, proceed with computing the Kalman gain K_t

EKF SLAM: Sensor = Range-Bearing Observation

- we assume sensor = laser range finder
- range-bearing observation $z_t^i = (r_t^i, \phi_t^i)^T$
- new initialization: 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

EKF SLAM: Expected Observation

- Remember: the measurement model is

$$z_t^i = \underbrace{\begin{pmatrix} \sqrt{(m_{j,x}-x)^2+(m_{j,y}-y)^2} \\ \mathrm{atan2}(m_{j,y}-y, m_{j,x}-x)-\theta \end{pmatrix}}_{h(x_t,j)} + \mathcal{N}(0, \underbrace{\begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\phi^2 \end{pmatrix}}_{Q_t})$$

  This expression is approxmiated by the linear function

$$h(x_t, j) \approx h(\bar{\mu}_t, j) + H_t^i \cdot (x_t - \bar{\mu}_t),$$

  where H_t^i is the derivative of h with respect to the **full** state vector x_t evaluated at the estimate $\bar{\mu}_t$:

$$H_t^i = \frac{\partial h(x_t, j)}{\partial x_t}_{|x_t=\bar{\mu}_t}$$

  and $j = c_t^i$ is the landmark that corresponds to measurement z_t^i.

- now we can compute expected observation $h(\bar{\mu}_t, j)$ and Jacobian of observation H_t^i

EKF SLAM: Expected Observation $\hat{z}_t = h(\bar{\mu}_t)$

- compute expected observation according to the current estimate:

$$\begin{aligned}
\hat{z}_t^i &= \left.\underbrace{\begin{pmatrix} \sqrt{(m_{j,x}-x)^2+(m_{j,y}-y)^2} \\ \text{atan2}(m_{j,y}-y, m_{j,x}-x)-\theta \end{pmatrix}}_{h(x_t,j)}\right|_{x_t=\bar{\mu}_t} \\
&= \underbrace{\begin{pmatrix} \sqrt{q} \\ \text{atan2}(\delta_y, \delta_x)-\bar{\mu}_{t,\theta} \end{pmatrix}}_{h(\bar{\mu}_t,j)}
\end{aligned}$$

with

$$\begin{aligned}
\delta &= \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{j,x}-\bar{\mu}_{t,x} \\ \bar{\mu}_{j,y}-\bar{\mu}_{t,y} \end{pmatrix} \\
q &= \delta^T \delta
\end{aligned}$$

EKF SLAM: Jacobian for the Observation H_t^i

- the Jacobian factors in a low-dimensional Jacobian ${}^{low}H_t^i$ and a matrix $F_{x,j}$ that maps ${}^{low}H_t^i$ into the high dimensional space of the full state vector:

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

EKF SLAM: Jacobian for the Observation H_t^i

- compute the Jacobian ${}^{low}H_t^i$

$$
\begin{aligned}
{}^{low}H_t^i &= \frac{\partial}{\partial(x\ y\ \theta\ m_{j,x}\ m_{j,y})}\left[\begin{pmatrix}\sqrt{(m_{j,x}-x)^2+(m_{j,y}-y)^2}\\ \text{atan2}(m_{j,y}-y, m_{j,x}-x)-\theta\end{pmatrix}\right]_{|x_t=\bar{\mu}_t} \\
&= \begin{pmatrix}\frac{\partial d}{\partial x} & \frac{\partial d}{\partial y} & \frac{\partial d}{\partial \theta} & \frac{\partial d}{\partial m_{j,x}} & \frac{\partial d}{\partial m_{j,y}} \\ \frac{\partial \text{atan2}(,)-\theta}{\partial x} & \frac{\partial \text{atan2}(,)-\theta}{\partial y} & \frac{\partial \text{atan2}(,)-\theta}{\partial \theta} & \frac{\partial \text{atan2}(,)-\theta}{\partial m_{j,x}} & \frac{\partial \text{atan2}(,)-\theta}{\partial m_{j,y}}\end{pmatrix}_{|x_t=\bar{\mu}_t} \\
&= \begin{pmatrix}\frac{x-m_{j,x}}{d} & \frac{y-m_{j,y}}{d} & 0 & \frac{m_{j,x}-x}{d} & \frac{m_{j,y}-y}{d} \\ \frac{m_{j,y}-y}{d^2} & \frac{x-m_{j,x}}{d^2} & -1 & \frac{y-m_{j,y}}{d^2} & \frac{m_{j,x}-x}{d^2}\end{pmatrix}_{|x_t=\bar{\mu}_t} \\
&= \frac{1}{q}\begin{pmatrix}-\sqrt{q}\delta_x & -\sqrt{q}\delta_y & 0 & \sqrt{q}\delta_x & \sqrt{q}\delta_y \\ \delta_y & -\delta_x & -q & -\delta_y & \delta_x\end{pmatrix},
\end{aligned}
$$

$$
\delta = \begin{pmatrix}\delta_x\\ \delta_y\end{pmatrix} = \begin{pmatrix}\bar{\mu}_{j,x}-\bar{\mu}_{t,x}\\ \bar{\mu}_{j,y}-\bar{\mu}_{t,y}\end{pmatrix}, \quad q=\delta^T\delta
$$

$\frac{d}{dx}\text{arctan}\, x = \frac{1}{1+x^2}$, $\quad d := \sqrt{(m_{j,x}-x)^2+(m_{j,y}-y)^2}$

EKF SLAM: Jacobian for the Observation H_t^i

- map Jacobian to the high dimensional space

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

with

$$F_{x,j} = \begin{pmatrix} 1 & 0 & 0 & 0\cdots0 & 0 & 0 & 0\cdots0 \\ 0 & 1 & 0 & 0\cdots0 & 0 & 0 & 0\cdots0 \\ 0 & 0 & 1 & 0\cdots0 & 0 & 0 & 0\cdots0 \\ 0 & 0 & 0 & 0\cdots0 & 1 & 0 & 0\cdots0 \\ 0 & 0 & 0 & \underbrace{0\cdots0}_{2j-2} & 0 & 1 & \underbrace{0\cdots0}_{2N-2j} \end{pmatrix}$$

$$H_t^i \in \mathbb{R}^{2\times(2N+3)}\,,\, {}^{low}H_t \in \mathbb{R}^{2\times5}\,,\, F_{x,j} \in \mathbb{R}^{5\times(2N+3)}$$

Extended Kalman Filter Algorithm

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

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

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

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

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

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

return μ_t, Σ_t

Extended Kalman Filter Algorithm

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

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

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

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

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

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

7: ➡ return μ_t, Σ_t

EKF-SLAM: Correction (1/2)

EKF_SLAM_Correction

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

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

$j = c_t^i$

if landmark j *never seen before*

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

endif

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

$q = \delta^T \delta$

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

EKF-SLAM: Correction (2/2)

$$F_{x,j} = \begin{pmatrix} 1 & 0 & 0 & 0\cdots0 & 0 & 0 & 0\cdots0 \\ 0 & 1 & 0 & 0\cdots0 & 0 & 0 & 0\cdots0 \\ 0 & 0 & 1 & 0\cdots0 & 0 & 0 & 0\cdots0 \\ 0 & 0 & 0 & 0\cdots0 & 1 & 0 & 0\cdots0 \\ 0 & 0 & 0 & \underbrace{0\cdots0}_{2j-2} & 0 & 1 & \underbrace{0\cdots0}_{2N-2j} \end{pmatrix}$$

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

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

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

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

endfor

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

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

return μ_t, Σ_t

EKF-SLAM

DONE!

EKF-SLAM: Full Algorithm

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

$$F_x = \begin{pmatrix} 1 & 0 & 0 & 0\cdots0 \\ 0 & 1 & 0 & 0\cdots0 \\ 0 & 0 & 1 & 0\cdots0 \end{pmatrix}$$

$$\bar{\mu}_t = \mu_{t-1} + F_x^T \begin{pmatrix} \frac{r}{2}(u_l + u_r)\cos\mu_{t-1,\theta}\Delta t \\ \frac{r}{2}(u_l + u_r)\sin\mu_{t-1,\theta}\Delta t \\ \frac{r}{L}(u_l - u_r)\ \Delta t \end{pmatrix}$$

$$G_t = I + F_x^T \begin{pmatrix} 0 & 0 & -\frac{r}{2}(u_l + u_r)\sin\mu_{t-1,\theta}\Delta t \\ 0 & 0 & \frac{r}{2}(u_l + u_r)\cos\mu_{t-1,\theta}\Delta t \\ 0 & 0 & 0 \end{pmatrix} F_x$$

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

Initialization

$$\mu_0 = (0\ 0\ 0\ \ldots\ 0)^T$$

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

EKF_SLAM_Correction

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

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

$\quad j = c_t^i$

$\quad$ *if landmark* j *never seen before*

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

$\quad$ *endif*

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

$$q = \delta^T \delta$$

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

$$F_{x,j} = \begin{pmatrix} 1 & 0 & 0 & 0\cdots0 & 0 & 0 & 0\cdots0 \\ 0 & 1 & 0 & 0\cdots0 & 0 & 0 & 0\cdots0 \\ 0 & 0 & 1 & 0\cdots0 & 0 & 0 & 0\cdots0 \\ 0 & 0 & 0 & 0\cdots0 & 1 & 0 & 0\cdots0 \\ 0 & 0 & 0 & \underbrace{0\cdots0}_{2j-2} & 0 & 1 & \underbrace{0\cdots0}_{2N-2j} \end{pmatrix}$$

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

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

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

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

endfor

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

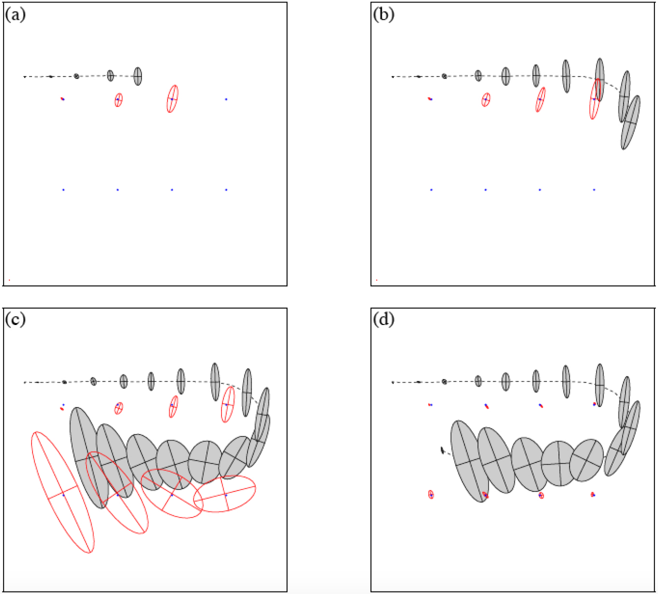
$\Sigma_t = \bar{\Sigma}_t$

return μ_t, Σ_t

Implementation notes

- measurement update in a single step possible (no for loop) $\Rightarrow$ large H matrix $\Rightarrow$ only one full belief update
- normalize the angular components

EKF-SLAM: Simulated example

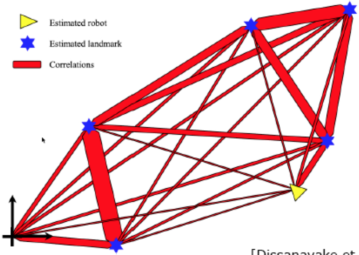


SLAM: Loop closure

- 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**

EKF-SLAM properties

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



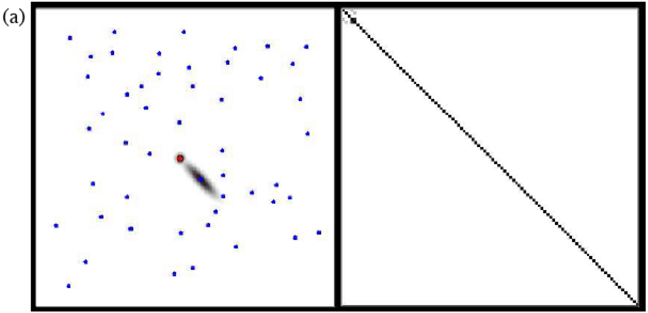
[Dissanayake et al., 2001]

- **correlation matrix = normalized covariance matrix**:

$$\rho = (\rho_{ij}) = \frac{\Sigma_{ij}}{\sqrt{\sigma_{ii} \cdot \sigma_{jj}}}\,, \quad -1 \leq \rho_{ij} \leq 1\,, \quad \rho_{ij} = \rho_{ji}$$

example: X = length of fish [in cm], Y = weight of fish [in kg]
$\rho_{xy} = E[(\frac{x-\mu_x}{\sqrt{\sigma_{xx}}})(\frac{y-\mu_y}{\sqrt{\sigma_{yy}}})]$
$\rho_{xy} > 0$ means if $(x - \mu_x) \lessgtr 0$ then $(y - \mu_y) \lessgtr 0$

EKF SLAM

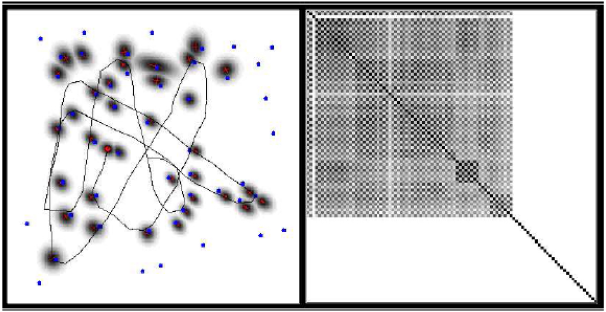


map correlation matrix

- correlation matrix of augmented state x_t:
  $$x_t = (\underbrace{x, y, \theta}_{\text{robot's pose}}, \underbrace{m_{1,x}, m_{1,y}}_{\text{landmark 1}}, \ldots, \underbrace{m_{n,x}, m_{n,y}}_{\text{landmark n}})^T$$
- at the beginning each variable is correlated with itself

EKF SLAM

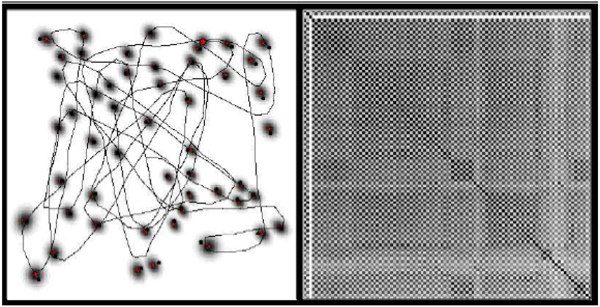
(b)



map correlation matrix

EKF SLAM

(c)

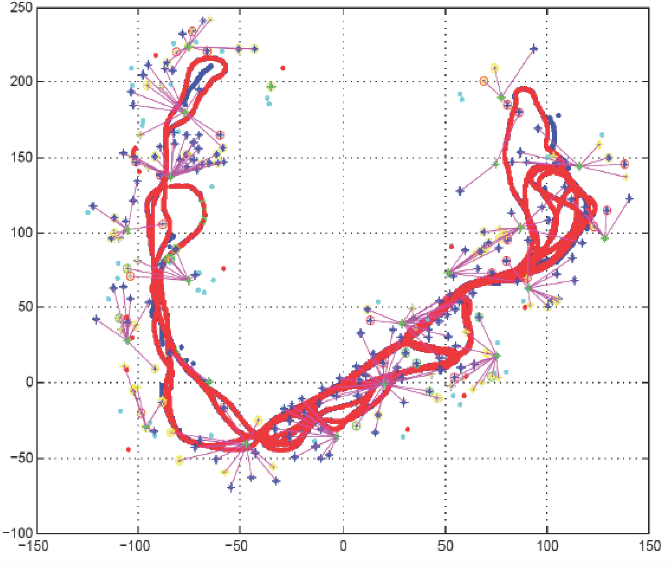


map correlation matrix

Example 1: Victoria Park Dataset

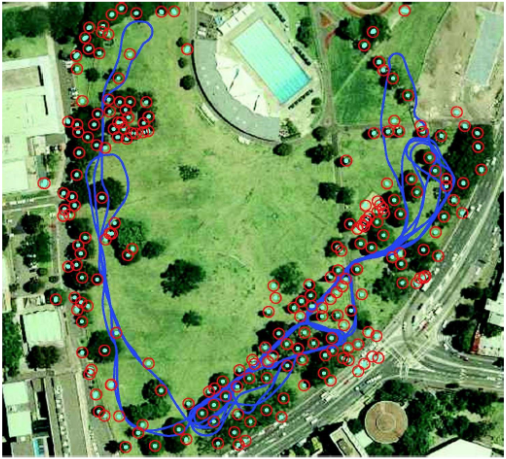
- Victoria Park @ University of Sidney
- landmarks = trees
- standard data set
- one of the first implementation

Example 1: Victoria Park Dataset



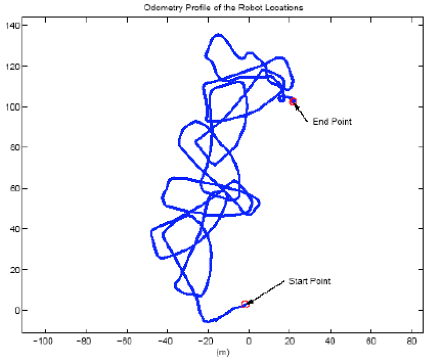
Example 1: Victoria Park Dataset

overlap of landmark postion estimates with satellite image

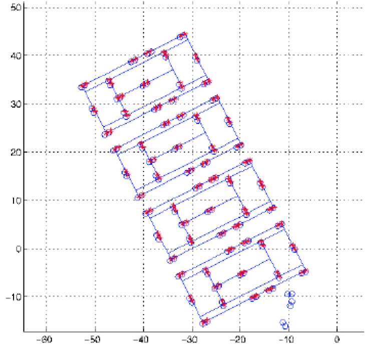


Example 2: Tennis Court Dataset, MIT

Example 2: Tennis Court Dataset, MIT



odometry



estimated trajectory

EKF SLAM - Complexity

- Cubic complexity depends only on the measurement dimensionality
- Cost per step: dominated by the number of landmarks: $O(n^3)$
- Memory consumption: $O(n^3)$
- The EKF becomes computationally intractable for large maps!

EKF SLAM - Summary

- The first SLAM solution in the robotics community (started beginning of 90's)
- 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 (no bimodal distributions)
- Successful in medium-scale scenes
- Approximations exists to reduce the computational complexity

Bonus: EKF-Localization algorithm

- we studied the **particle filter for localization**
- there is also an **EKF localization algorithm**, which is similar to EKF SLAM with the only difference that in EKF localization the landmark positions (map) are given.
- so you get the EKF localization algorithm for free :)

Bonus: EKF-Localization algorithm - Part 1/2

Algorithm EKF_localization_known_correspondences($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, c_t, m$):

$$\bar{\mu}_t = \mu_{t-1} + \begin{pmatrix} \frac{r}{2}(u_l + u_r)\cos\mu_{t-1,\theta}\Delta t \\ \frac{r}{2}(u_l + u_r)\sin\mu_{t-1,\theta}\Delta t \\ \frac{r}{L}(u_l - u_r)\ \Delta t \end{pmatrix}$$

$$G_t = \begin{pmatrix} 1 & 0 & -\frac{r}{2}(u_l + u_r)\sin\mu_{t-1,\theta}\Delta t \\ 0 & 1 & \frac{r}{2}(u_l + u_r)\cos\mu_{t-1,\theta}\Delta t \\ 0 & 0 & 1 \end{pmatrix}$$

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

$$Q_t = \begin{pmatrix} \sigma_r & 0 & 0 \\ 0 & \sigma_\phi & 0 \\ 0 & 0 & \sigma_s \end{pmatrix}$$

note: here the measurement model contains besides the range r and bearing ϕ also the signature s (e.g., color) of the landmark.

Bonus: EKF-Localization algorithm - Part 2/2

for all observed features $z_t^i = (r_t^i \;\; \phi_t^i \;\; s_t^i)^T$ do

$$j = c_t^i$$

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

$$q = \delta^T \delta$$

$$\hat{z}_t^i = \begin{pmatrix} \sqrt{q} \\ \operatorname{atan2}(\delta_y, \delta_x) - \bar{\mu}_{t,\theta} \\ m_{j,s} \end{pmatrix}$$

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

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

endfor

$$\mu_t = \bar{\mu}_t + \textstyle\sum_i K_t^i (z_t^i - \hat{z}_t^i)$$

$$\Sigma_t = (I - \textstyle\sum_i K_t^i \, H_t^i) \, \bar{\Sigma}_t$$

return μ_t, Σ_t

note: here the measurement model contains besides the range r and bearing ϕ also the signature s (e.g., color) of the landmark.