CS 4610/5335 – Lecture 13
Simultaneous Localization & Mapping

Lawson L.S. Wong
Northeastern University
3/7/22

Material adapted from:
1. Robert Platt, CS 4610/5335
2. Peter Corke, Robotics, Vision and Control
3. Sebastian Thrun, Wolfram Burgard, & Dieter Fox, Probabilistic Robotics
4. Marc Toussaint, U. Stuttgart Robotics Course

Announcements

Ex3 due Friday (3/11)

- Update: Only hand in KF, E0, E1, E2
  (E3 and E4 to be handed in with Ex4)

Announcements

Ex3 due Friday (3/11)

- Update: Only hand in KF, E0, E1, E2
  (E3 and E4 to be handed in with Ex4)

Ex2: Debrief?

Announcements

Ex3 due Friday (3/11)
- Update: Only hand in KF, E0, E1, E2
  (E3 and E4 to be handed in with Ex4)

Ex2: Debrief?

Project:
- Making progress on robot platforms!
- Will post initial hardware assignments / comments soon

Announcements

Ex3 due Friday (3/11)

- Update: Only hand in KF, E0, E1, E2
  (E3 and E4 to be handed in with Ex4)

Ex2: Debrief?

Project:

- Making progress on robot platforms!
- Will post initial hardware assignments / comments soon

- Some equipment may be available by Friday
  - RGB-D cameras (some D435s)
  - Turtlebots and Duckiebots
- Let us know if you want to borrow them this week
  (post unresolved followup on your project thread)

Outline

Recap: Localization with a known map (landmarks)

Mapping (landmarks) with known location

Simultaneous localization and mapping (EKF-SLAM)

EKF-based mobile robot localization

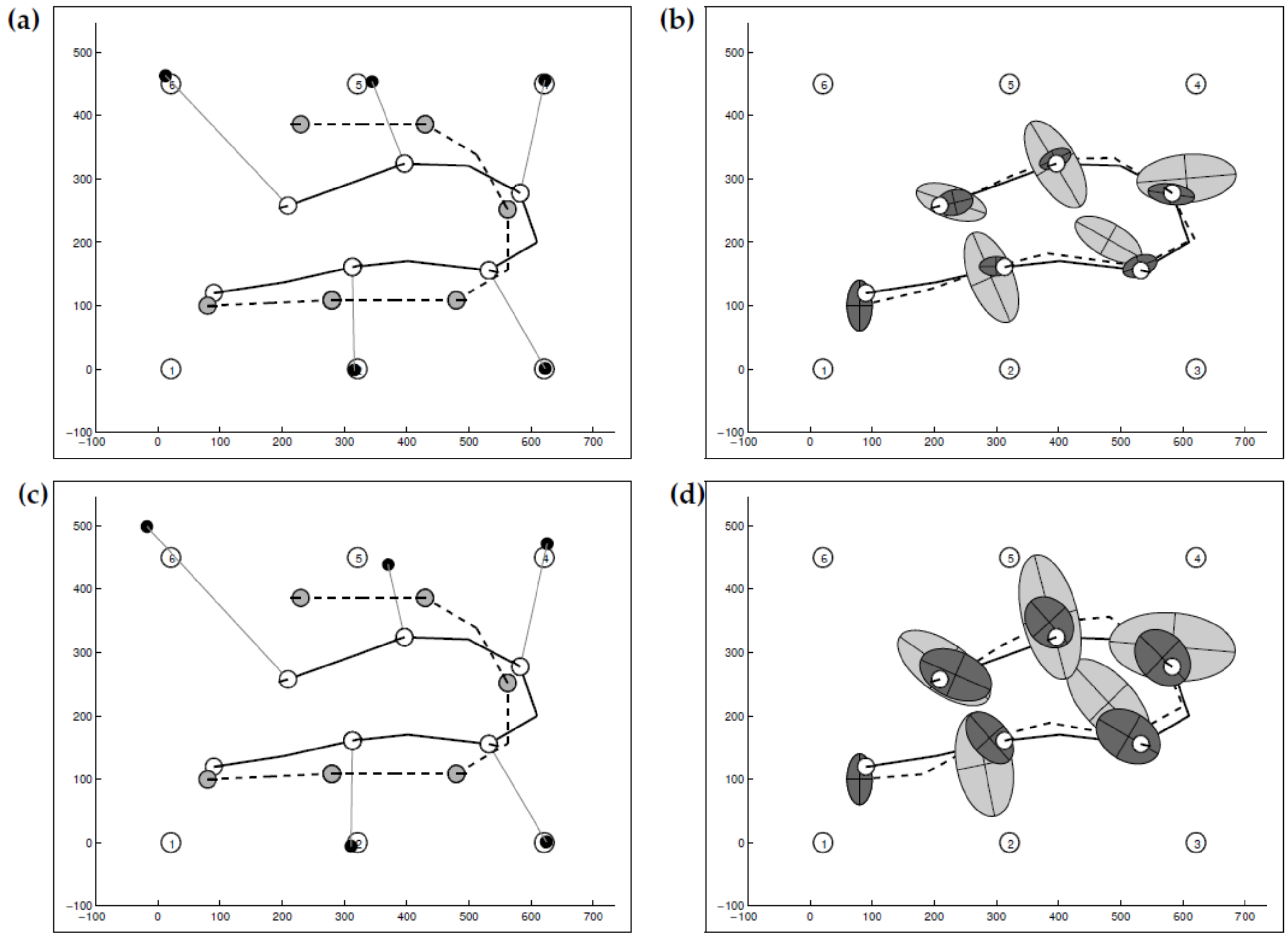
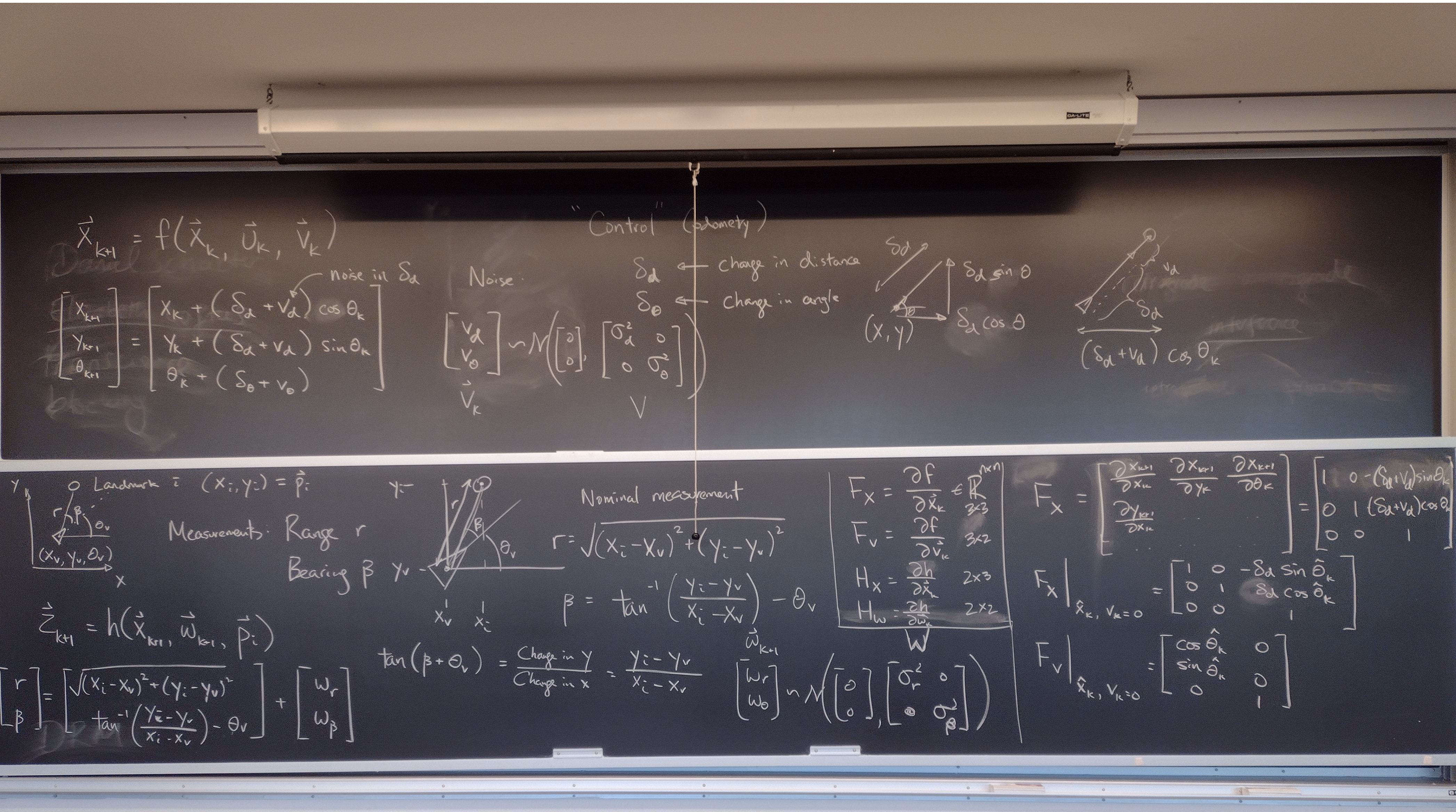


Figure 7.11 EKF-based localization with an accurate (upper row) and a less accurate (lower row) landmark detection sensor. The dashed lines in the left panel indicate the robot trajectories as estimated from the motion controls. The solid lines represent the true robot motion resulting from these controls. Landmark detections at five locations are indicated by the thin lines. The dashed lines in the right panels show the corrected robot trajectories, along with uncertainty before (light gray, $\bar{\Sigma}_t$) and after (dark gray, Σ_t) incorporating a landmark detection.

Last time: EKF localization



2-D mobile robot model

Process dynamics: $\boldsymbol{x}\langle k{+}1\rangle = \boldsymbol{f}\big(\boldsymbol{x}\langle k\rangle,\, \delta\langle k\rangle,\, \boldsymbol{v}\langle k\rangle\big)$

The robot's configuration at the next time step, including the odometry error, is

$$\boldsymbol{x}\langle k{+}1\rangle = \boldsymbol{f}\big(\boldsymbol{x}\langle k\rangle,\, \delta\langle k\rangle,\, \boldsymbol{v}\langle k\rangle\big) = \begin{pmatrix} x\langle k\rangle + (\delta_d + v_d)\cos\theta\langle k\rangle \\ y\langle k\rangle + (\delta_d + v_d)\sin\theta\langle k\rangle \\ \theta\langle k\rangle + \delta_\theta + v_\theta \end{pmatrix} \tag{6.2}$$

where k is the time step, $\delta\langle k\rangle = (\delta_d, \delta_\theta)^T \in \mathbb{R}^{2\times 1}$ is the odometry measurement and $\boldsymbol{v}\langle k\rangle = (v_d, v_\theta)^T \in \mathbb{R}^{2\times 1}$ is the random measurement noise over the preceding interval.

$$\boldsymbol{v} = (v_d, v_\theta)^T \sim N(0, V)$$

$$\boldsymbol{V} = \begin{pmatrix} \sigma_d^2 & 0 \\ 0 & \sigma_\theta^2 \end{pmatrix}$$

2-D mobile robot model

To make this tangible we will consider a common type of sensor that measures the range and bearing angle to a landmark in the environment, for instance a radar or a scanning-laser rangefinder such as shown in Fig. 6.22a. The sensor is mounted on-board the robot so the observation of the i^{th} landmark is

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{p}_i) = \begin{pmatrix} \sqrt{(y_i - y_v)^2 + (x_i - x_v)^2} \\ \tan^{-1}(y_i - y_v)/(x_i - x_v) - \theta_v \end{pmatrix} + \begin{pmatrix} w_r \\ w_\beta \end{pmatrix} \tag{6.8}$$

where $\boldsymbol{z} = (r, \beta)^T$ and r is the range, β the bearing angle, and $\boldsymbol{w} = (w_r, w_\beta)^T$ is a zero-mean Gaussian random variable that models errors in the sensor

$$\begin{pmatrix} w_r \\ w_\beta \end{pmatrix} \sim N(0, W), \quad W = \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\beta^2 \end{pmatrix}$$

Extended Kalman filter: Prediction step

$$\hat{\boldsymbol{x}}^{+}\langle k+1\rangle = \boldsymbol{f}\big(\hat{\boldsymbol{x}}\langle k\rangle, \hat{\boldsymbol{u}}\langle k\rangle\big)$$

Predicted mean

$$\hat{\boldsymbol{P}}^{+}\langle k+1\rangle = \boldsymbol{F}_x \hat{\boldsymbol{P}}\langle k\rangle \boldsymbol{F}_x^T + \boldsymbol{F}_\nu \hat{\boldsymbol{V}} \boldsymbol{F}_\nu^T$$

Predicted covariance

EKF localization: Prediction step

$$\hat{\boldsymbol{x}}^{+}\langle k+1\rangle = \boldsymbol{f}\left(\hat{\boldsymbol{x}}\langle k\rangle, \hat{\boldsymbol{u}}\langle k\rangle\right)$$

$$\hat{\boldsymbol{P}}^{+}\langle k+1\rangle = \boldsymbol{F}_x \hat{\boldsymbol{P}}\langle k\rangle \boldsymbol{F}_x^T + \boldsymbol{F}_v \hat{\boldsymbol{V}} \boldsymbol{F}_v^T$$

For our motion model:

$$\boldsymbol{F}_x = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{v}=0} = \begin{pmatrix} 1 & 0 & -\delta_d \sin\theta_v \\ 0 & 1 & \delta_d \cos\theta_v \\ 0 & 0 & 1 \end{pmatrix}$$

$$\boldsymbol{F}_v = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{v}}\right|_{\boldsymbol{v}=0} = \begin{pmatrix} \cos\theta_v & 0 \\ \sin\theta_v & 0 \\ 0 & 1 \end{pmatrix}$$

$$\boldsymbol{x}\langle k+1\rangle = \boldsymbol{f}\left(\boldsymbol{x}\langle k\rangle, \delta\langle k\rangle, \boldsymbol{v}\langle k\rangle\right) = \begin{pmatrix} x\langle k\rangle + \left(\delta_d + v_d\right)\cos\theta\langle k\rangle \\ y\langle k\rangle + \left(\delta_d + v_d\right)\sin\theta\langle k\rangle \\ \theta\langle k\rangle + \delta_\theta + v_\theta \end{pmatrix}$$

Extended Kalman filter: Update step

$$\boldsymbol{\nu} = \boldsymbol{z}^{\#}\langle k+1\rangle - \boldsymbol{h}\left(\hat{\boldsymbol{x}}^{+}\langle k+1\rangle, \boldsymbol{p}_i\right)$$

Innovation

$$\boldsymbol{K} = \boldsymbol{P}^{+}\langle k+1\rangle \boldsymbol{H}_x^T \boldsymbol{S}^{-1}$$

$$\boldsymbol{S} = \boldsymbol{H}_x \boldsymbol{P}^{+}\langle k+1\rangle \boldsymbol{H}_x^T + \boldsymbol{H}_w \hat{\boldsymbol{W}} \boldsymbol{H}_w^T$$

Kalman gain

$$\hat{\boldsymbol{x}}\langle k+1\rangle = \hat{\boldsymbol{x}}^{+}\langle k+1\rangle + \boldsymbol{K}\boldsymbol{\nu}$$

Posterior mean

$$\hat{\boldsymbol{P}}\langle k+1\rangle = \hat{\boldsymbol{P}}^{+}\langle k+1\rangle - \boldsymbol{K}\boldsymbol{H}_x \hat{\boldsymbol{P}}^{+}\langle k+1\rangle$$

Posterior cov.

EKF localization: Update step

$$\boldsymbol{\nu} = \boldsymbol{z}^{\#}\langle k+1\rangle - \boldsymbol{h}\left(\hat{\boldsymbol{x}}^{+}\langle k+1\rangle, \boldsymbol{p}_i\right)$$

$$\boldsymbol{K} = \boldsymbol{P}^{+}\langle k+1\rangle \boldsymbol{H}_x^T \boldsymbol{S}^{-1}$$

$$\boldsymbol{S} = \boldsymbol{H}_x \boldsymbol{P}^{+}\langle k+1\rangle \boldsymbol{H}_x^T + \boldsymbol{H}_w \hat{\boldsymbol{W}} \boldsymbol{H}_w^T$$

$$\hat{\boldsymbol{x}}\langle k+1\rangle = \hat{\boldsymbol{x}}^{+}\langle k+1\rangle + \boldsymbol{K}\boldsymbol{\nu}$$

$$\hat{\boldsymbol{P}}\langle k+1\rangle = \hat{\boldsymbol{P}}^{+}\langle k+1\rangle - \boldsymbol{K}\boldsymbol{H}_x \hat{\boldsymbol{P}}^{+}\langle k+1\rangle$$

$$\boldsymbol{H}_x = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{w}=0} =$$

$$\begin{pmatrix} -\dfrac{x_i - x_v}{r} & -\dfrac{y_i - y_v}{r} & 0 \\ \dfrac{y_i - y_v}{r^2} & -\dfrac{x_i - x_v}{r^2} & -1 \end{pmatrix}$$

$$\boldsymbol{H}_w = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{w}}\right|_{\boldsymbol{w}=0} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

For our measurement model:

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{p}_i) = \begin{pmatrix} \sqrt{(y_i - y_v)^2 + (x_i - x_v)^2} \\ \tan^{-1}(y_i - y_v)/(x_i - x_v) - \theta_v \end{pmatrix} + \begin{pmatrix} w_r \\ w_\beta \end{pmatrix}$$

Extended Kalman filter

$$\hat{\boldsymbol{x}}^{+}\langle k+1\rangle = \boldsymbol{f}(\hat{\boldsymbol{x}}\langle k\rangle, \hat{\boldsymbol{u}}\langle k\rangle)$$ predict state one step ahead

$$\hat{P}^{+}\langle k+1\rangle = F_x \hat{P}\langle k\rangle F_x^T + F_v V F_v^T$$ project covariance one step ahead

prediction phase

$$\boldsymbol{v} = \boldsymbol{z}^{\#}\langle k+1\rangle - h(\hat{\boldsymbol{x}}^{+}\langle k+1\rangle, \boldsymbol{p}_i)$$ new information – innovation

$$K = P^{+}\langle k+1\rangle H_x^T \left(H_x P^{+}\langle k+1\rangle H_x^T + H_w W H_w^T\right)^{-1}$$ how to distribute the innovation to states – Kalman gain

$$\hat{\boldsymbol{x}}\langle k+1\rangle = \hat{\boldsymbol{x}}^{+}\langle k+1\rangle + K\boldsymbol{v}$$ state updated with innovation

$$\hat{P}\langle k+1\rangle = \hat{P}^{+}\langle k+1\rangle - K H_x \hat{P}^{+}\langle k+1\rangle$$ updated covariance

update phase

Fig. 6.6.
Summary of extended Kalman filter algorithm showing the prediction and update phases

F_x, F_v, H_x, H_w are Jacobian matrices

Extended Kalman filter

$$\hat{\boldsymbol{x}}^{+}\langle k{+}1\rangle = \boldsymbol{f}(\hat{\boldsymbol{x}}\langle k\rangle, \hat{\boldsymbol{u}}\langle k\rangle)$$ predict state one step ahead

$$\hat{P}^{+}\langle k{+}1\rangle = F_x \hat{P}\langle k\rangle F_x^T + F_v V F_v^T$$ project covariance one step ahead

prediction phase

$$\boldsymbol{\nu} = \boldsymbol{z}^{\#}\langle k{+}1\rangle - h(\hat{\boldsymbol{x}}^{+}\langle k{+}1\rangle, \boldsymbol{p}_i)$$ new information – innovation

$$K = P^{+}\langle k{+}1\rangle H_x^T \left(H_x P^{+}\langle k{+}1\rangle H_x^T + H_w W H_w^T\right)^{-1}$$ how to distribute the innovation to states – Kalman gain

$$\hat{\boldsymbol{x}}\langle k{+}1\rangle = \hat{\boldsymbol{x}}^{+}\langle k{+}1\rangle + K\boldsymbol{\nu}$$ state updated with innovation

$$\hat{P}\langle k{+}1\rangle = \hat{P}^{+}\langle k{+}1\rangle - K H_x \hat{P}^{+}\langle k{+}1\rangle$$ updated covariance

update phase

Fig. 6.6. Summary of extended Kalman filter algorithm showing the prediction and update phases

F_x , F_v , H_x , H_w are Jacobian matrices

Piazza @145:
If no landmark is detected, should anything be updated?
If multiple landmarks are detected,
what should be updated, and how?

Extended Kalman filter

Procedure EKF

Input : $\hat{\boldsymbol{x}}\langle k\rangle \in \mathbb{R}^n$, $\hat{\boldsymbol{P}}\langle k\rangle \in \mathbb{R}^{n\times n}$, $\boldsymbol{u}\langle k\rangle \in \mathbb{R}^m$, $\boldsymbol{z}\langle k+1\rangle \in \mathbb{R}^p$, $\hat{\boldsymbol{V}} \in \mathbb{R}^{n\times n}$, $\hat{\boldsymbol{W}} \in \mathbb{R}^{p\times p}$

Output: $\hat{\boldsymbol{x}}\langle k+1\rangle \in \mathbb{R}^n$, $\hat{\boldsymbol{P}}\langle k+1\rangle \in \mathbb{R}^{n\times n}$

– *linearize about* $\boldsymbol{x} = \hat{\boldsymbol{x}}\langle k\rangle$

compute Jacobians: $\boldsymbol{F}_x \in \mathbb{R}^{n\times n}$, $\boldsymbol{F}_v \in \mathbb{R}^{n\times n}$, $\boldsymbol{H}_x \in \mathbb{R}^{p\times n}$, $\boldsymbol{H}_w \in \mathbb{R}^{p\times p}$

– *the prediction step*

$\hat{\boldsymbol{x}}^+\langle k+1\rangle = \boldsymbol{f}\left(\hat{\boldsymbol{x}}\langle k\rangle, \boldsymbol{u}\langle k\rangle\right)$ *// predict state at next time step*

$\hat{\boldsymbol{P}}^+\langle k+1\rangle = \boldsymbol{F}_x \hat{\boldsymbol{P}}\langle k\rangle \boldsymbol{F}_x^T + \boldsymbol{F}_v \hat{\boldsymbol{V}} \boldsymbol{F}_v^T$ *// predict cov ariance at next time step*

– *the update step*

$\nu = \boldsymbol{z}\langle k+1\rangle - \boldsymbol{h}\left(\hat{\boldsymbol{x}}^+\langle k+1\rangle\right)$ *// innovation : measured – predicted sensor value*

$\boldsymbol{K} = \boldsymbol{P}^+\langle k+1\rangle \boldsymbol{H}_x^T \left[\boldsymbol{H}_x \boldsymbol{P}^+\langle k+1\rangle \boldsymbol{H}_x^T + \boldsymbol{H}_w \hat{\boldsymbol{W}} \boldsymbol{H}_w^T\right]^{-1}$ *// Kalman gain*

$\hat{\boldsymbol{x}}\langle k+1\rangle = \hat{\boldsymbol{x}}^+\langle k+1\rangle + \boldsymbol{K}\nu$ *// update state estimate*

$\hat{\boldsymbol{P}}\langle k+1\rangle = \hat{\boldsymbol{P}}^+\langle k+1\rangle - \boldsymbol{K}\boldsymbol{H}_x \hat{\boldsymbol{P}}^+\langle k+1\rangle$ *// update covariance estimate*

Algorithm H.1.
Procedure EKF

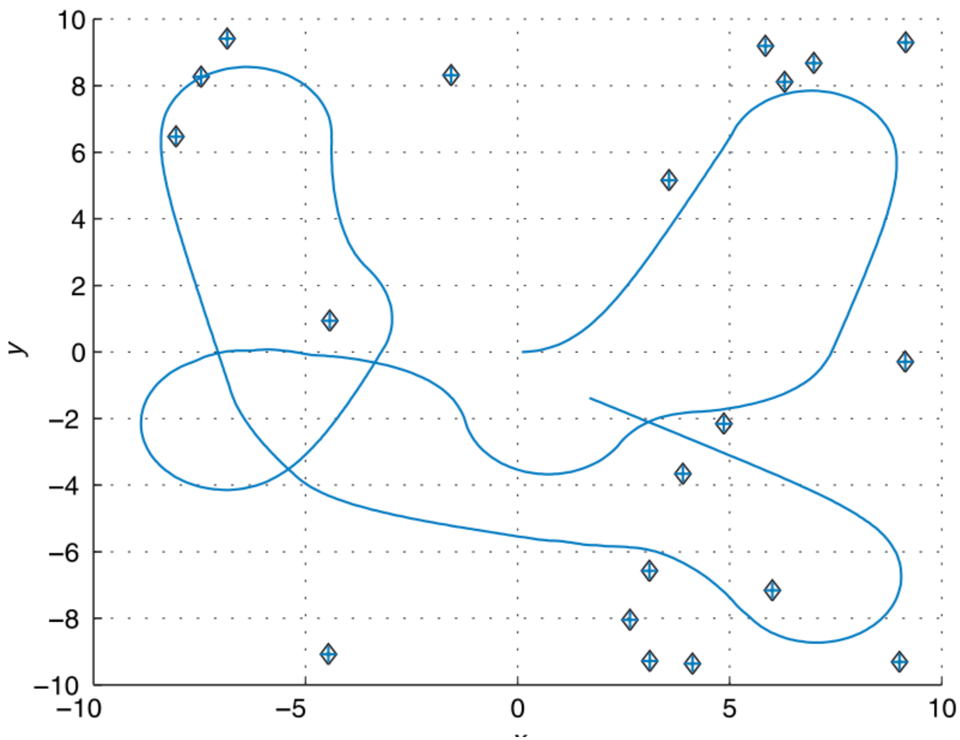
Outline

- ✔ Recap: Localization with a known map (landmarks)

Mapping (landmarks) with known location

Simultaneous localization and mapping (EKF-SLAM)

Mapping using the EKF



How to use EKF to estimate landmark positions?

State: $\hat{\boldsymbol{x}} = \left(x_1, y_1, x_2, y_2, \cdots x_M, y_M\right)^T \in \mathbb{R}^{2M\times 1}$

Positions of the *M* landmarks (in world frame)

Mapping using the EKF

$$\hat{\boldsymbol{x}}^{+}\langle k+1\rangle = \boldsymbol{f}(\hat{\boldsymbol{x}}\langle k\rangle, \hat{\boldsymbol{u}}\langle k\rangle)$$ predict state one step ahead

$$\hat{P}^{+}\langle k+1\rangle = F_x \hat{P}\langle k\rangle F_x^T + F_v V F_v^T$$ project covariance one step ahead

prediction phase

$$\boldsymbol{\nu} = \boldsymbol{z}^{\#}\langle k+1\rangle - \boldsymbol{h}(\hat{\boldsymbol{x}}^{+}\langle k+1\rangle, \boldsymbol{p}_i)$$ new information – innovation

$$K = P^{+}\langle k+1\rangle H_x^T \left(H_x P^{+}\langle k+1\rangle H_x^T + H_w W H_w^T\right)^{-1}$$ how to distribute the innovation to states – Kalman gain

$$\hat{\boldsymbol{x}}\langle k+1\rangle = \hat{\boldsymbol{x}}^{+}\langle k+1\rangle + K\boldsymbol{\nu}$$ state updated with innovation

$$\hat{P}\langle k+1\rangle = \hat{P}^{+}\langle k+1\rangle - K H_x \hat{P}^{+}\langle k+1\rangle$$ updated covariance

update phase

Fig. 6.6. Summary of extended Kalman filter algorithm showing the prediction and update phases

Landmark positions are now our states (assume known vehicle pose):

$$\boxed{\hat{\boldsymbol{x}} = (x_1, y_1, x_2, y_2, \cdots x_M, y_M)^T \in \mathbb{R}^{2M\times 1}}$$

High level idea:
1. Landmark positions do not move

2a. If new landmark detected, expand the state

2b. Else, update the appropriate landmark

Mapping using the EKF

Landmark positions are now our states
(assume known vehicle pose):

$$\hat{x} = (x_1, y_1, x_2, y_2, \cdots x_M, y_M)^T \in \mathbb{R}^{2M\times 1}$$

High level idea:

1. Landmark positions do not move
   Nothing to do in prediction step!

2a. If new landmark detected, expand the state
    State mean, covariance grow by two dimensions
    Covariance update requires “insertion” Jacobian

2b. Else, update the appropriate landmark
    Very similar to previous EKF update step
    Jacobian is now 2 * (2M) matrix instead of 2 * 3

EKF mapping: Prediction step

The prediction equation is straightforward in this case since the landmarks are assumed to be stationary

$$\hat{\boldsymbol{x}}^{+}\langle k{+}1\rangle = \hat{\boldsymbol{x}}\langle k\rangle \tag{6.16}$$

$$\hat{\boldsymbol{P}}^{+}\langle k{+}1\rangle = \hat{\boldsymbol{P}}\langle k\rangle \tag{6.17}$$

$$\cancel{\hat{\boldsymbol{x}}^{+}\langle k{+}1\rangle = \boldsymbol{f}\big(\hat{\boldsymbol{x}}\langle k\rangle, \hat{\boldsymbol{u}}\langle k\rangle\big)}$$

$$\cancel{\hat{\boldsymbol{P}}^{+}\langle k{+}1\rangle = \boldsymbol{F}_x \hat{\boldsymbol{P}}\langle k\rangle \boldsymbol{F}_x^T + \boldsymbol{F}_v \hat{\boldsymbol{V}} \boldsymbol{F}_v^T}$$

Mapping using the EKF

Landmark positions are now our states
(assume known vehicle pose):

$$\hat{x} = (x_1, y_1, x_2, y_2, \cdots x_M, y_M)^T \in \mathbb{R}^{2M \times 1}$$

High level idea:

1. Landmark positions do not move
   Nothing to do in prediction step!

2a. If new landmark detected, expand the state
State mean, covariance grow by two dimensions
Covariance update requires “insertion” Jacobian

2b. Else, update the appropriate landmark
Very similar to previous EKF update step
Jacobian is now 2 * (2M) matrix instead of 2 * 3

EKF Mapping: Update Step (existing landmark)

For the mapping case the Jacobian $\boldsymbol{H}_x$ used in Eq. 6.11 describes how the landmark observation changes with respect to the full state vector. However the observation depends only on the position of that landmark so this Jacobian is mostly zeros

$$\boldsymbol{H}_x = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right|_{w=0} = \left(0 \cdots \boldsymbol{H}_{p_i} \cdots 0\right) \in \mathbb{R}^{2\times 2M} \qquad (6.24)$$

where $\boldsymbol{H}_{p_i}$ is at the location in the vector corresponding to the state $\boldsymbol{p}_i$. This structure

Assumption: We know which landmark we observed:
Landmark i with position p_i

EKF Mapping: Update Step (existing landmark)

For the mapping case the Jacobian $\boldsymbol{H}_x$ used in Eq. 6.11 describes how the landmark observation changes with respect to the full state vector. However the observation depends only on the position of that landmark so this Jacobian is mostly zeros

$$\boldsymbol{H}_x = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{w}=0} = \left(0 \cdots \boldsymbol{H}_{p_i} \cdots 0\right) \in \mathbb{R}^{2\times 2M} \tag{6.24}$$

where $\boldsymbol{H}_{p_i}$ is at the location in the vector corresponding to the state $\boldsymbol{p}_i$. This structure

Assumption: We know which landmark we observed:
Landmark i with position $p_i = (x_i , y_i)$

Jacobian of (r, β) with respect to landmark position (x_i , y_i):

$$\boldsymbol{H}_{p_i} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{p}_i} = \begin{pmatrix} \frac{x_i - x_v}{r} & \frac{y_i - y_v}{r} \\ -\frac{y_i - y_v}{r^2} & \frac{x_i - x_v}{r^2} \end{pmatrix}$$

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{p}_i) = \begin{pmatrix} \sqrt{(y_i - y_v)^2 + (x_i - x_v)^2} \\ \tan^{-1}(y_i - y_v)/(x_i - x_v) - \theta_v \end{pmatrix} + \begin{pmatrix} w_r \\ w_\beta \end{pmatrix}$$

EKF Mapping: Update Step (existing landmark)

$$\boldsymbol{\nu} = \boldsymbol{z}^{\#}\langle k+1\rangle - \boxed{\boldsymbol{h}\left(\boldsymbol{x}_v\langle k+1\rangle, \hat{\boldsymbol{x}}^{+}\langle k+1\rangle\right)}$$

$$\boldsymbol{K} = \boldsymbol{P}^{+}\langle k+1\rangle \boldsymbol{H}_x^T \boldsymbol{S}^{-1}$$

$$\boldsymbol{S} = \boldsymbol{H}_x \boldsymbol{P}^{+}\langle k+1\rangle \boldsymbol{H}_x^T + \boldsymbol{H}_w \hat{\boldsymbol{W}} \boldsymbol{H}_w^T$$

$$\hat{\boldsymbol{x}}\langle k+1\rangle = \hat{\boldsymbol{x}}^{+}\langle k+1\rangle + \boldsymbol{K}\boldsymbol{\nu}$$

$$\hat{\boldsymbol{P}}\langle k+1\rangle = \hat{\boldsymbol{P}}^{+}\langle k+1\rangle - \boldsymbol{K}\boldsymbol{H}_x \hat{\boldsymbol{P}}^{+}\langle k+1\rangle$$

Known x_v

Unknown p_i

$$\boldsymbol{H}_x = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right|_{w=0} = \left(0 \cdots \boldsymbol{H}_{p_i} \cdots 0\right) \in \mathbb{R}^{2\times 2M}$$

$$\boldsymbol{H}_{p_i} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{p}_i} = \begin{pmatrix} \frac{x_i - x_v}{r} & \frac{y_i - y_v}{r} \\ -\frac{y_i - y_v}{r^2} & \frac{x_i - x_v}{r^2} \end{pmatrix}$$

$$\boldsymbol{H}_w = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{w}}\right|_{w=0} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

For our measurement model:

$$\boldsymbol{z} = \boxed{\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{p}_i)} = \begin{pmatrix} \sqrt{(y_i - y_v)^2 + (x_i - x_v)^2} \\ \tan^{-1}(y_i - y_v)/(x_i - x_v) - \theta_v \end{pmatrix} + \begin{pmatrix} w_r \\ w_\beta \end{pmatrix}$$

Mapping using the EKF

Landmark positions are now our states
(assume known vehicle pose):

$$\hat{x} = (x_1, y_1, x_2, y_2, \cdots x_M, y_M)^T \in \mathbb{R}^{2M \times 1}$$

High level idea:

1. Landmark positions do not move
   Nothing to do in prediction step!

2a. If new landmark detected, expand the state
State mean, covariance grow by two dimensions
Covariance update requires "insertion" Jacobian

2b. Else, update the appropriate landmark
Very similar to previous EKF update step
Jacobian is now 2 * (2M) matrix instead of 2 * 3

EKF mapping: Update step (new landmark)

We introduce the function $\boldsymbol{g}(\cdot)$ which is the inverse of $\boldsymbol{h}(\cdot)$ and gives the coordinates of the observed landmark based on the known vehicle pose and the sensor observation

$$\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{z}) = \begin{pmatrix} x_v + r\cos(\theta_v + \beta) \\ y_v + r\sin(\theta_v + \beta) \end{pmatrix}$$

Given a measurement (r, β), where is the new landmark?

EKF mapping: Update step (new landmark)

We introduce the function $\boldsymbol{g}(\cdot)$ which is the inverse of $\boldsymbol{h}(\cdot)$ and gives the coordinates of the observed landmark based on the known vehicle pose and the sensor observation

$$\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{z}) = \begin{pmatrix} x_v + r\cos(\theta_v + \beta) \\ y_v + r\sin(\theta_v + \beta) \end{pmatrix}$$

Given a measurement (r, β), where is the new landmark?

Expand the state with "insertion" operator:

$$\boldsymbol{x}\langle k\rangle' = \boldsymbol{y}\left(\boldsymbol{x}\langle k\rangle, \boldsymbol{z}\langle k\rangle, \boldsymbol{x}_v\langle k\rangle\right)$$

$$= \begin{pmatrix} \boldsymbol{x}\langle k\rangle \\ \boldsymbol{g}\left(\boldsymbol{x}_v\langle k\rangle, \boldsymbol{z}\langle k\rangle\right) \end{pmatrix}$$

This is our new posterior mean $\hat{\boldsymbol{x}}\langle k+1\rangle$

EKF mapping: Update step (new landmark)

Expand the state with "insertion" operator:

$$\boldsymbol{x}\langle k\rangle' = \boldsymbol{y}(\boldsymbol{x}\langle k\rangle, \boldsymbol{z}\langle k\rangle, \boldsymbol{x}_v\langle k\rangle)$$

$$= \begin{pmatrix} \boldsymbol{x}\langle k\rangle \\ \boldsymbol{g}(\boldsymbol{x}_v\langle k\rangle, \boldsymbol{z}\langle k\rangle) \end{pmatrix}$$

This is our new posterior mean $\hat{\boldsymbol{x}}\langle k+1\rangle$

Covariance update:

$$\hat{\boldsymbol{P}}\langle k\rangle' = \boldsymbol{Y}_z \begin{pmatrix} \hat{\boldsymbol{P}}\langle k\rangle & 0 \\ 0 & \hat{\boldsymbol{W}} \end{pmatrix} \boldsymbol{Y}_z^T \qquad \hat{\boldsymbol{P}}\langle k+1\rangle$$

where $\boldsymbol{Y}_z$ is the insertion Jacobian

$$\boldsymbol{Y}_z = \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{z}} = \begin{pmatrix} \boldsymbol{I}_{n\times n} & & \boldsymbol{0}_{n\times 2} \\ \boldsymbol{G}_x & \boldsymbol{0}_{2\times n-3} & \boldsymbol{G}_z \end{pmatrix}$$

$$\boldsymbol{G}_x = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$\boldsymbol{G}_z = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{z}} = \begin{pmatrix} \cos(\theta_v + \beta) & -r\sin(\theta_v + \beta) \\ \sin(\theta_v + \beta) & r\cos(\theta_v + \beta) \end{pmatrix}$$

Mapping using the EKF

Landmark positions are now our states
(assume known vehicle pose):

$$\boxed{\hat{x} = \left(x_1, y_1, x_2, y_2, \cdots x_M, y_M\right)^T \in \mathbb{R}^{2M\times 1}}$$

High level idea:

1. Landmark positions do not move
   Nothing to do in prediction step!

2a. If new landmark detected, expand the state
   State mean, covariance grow by two dimensions
   Covariance update requires “insertion” Jacobian

2b. Else, update the appropriate landmark
   Very similar to previous EKF update step
   Jacobian is now 2 * (2M) matrix instead of 2 * 3

Mapping using the EKF

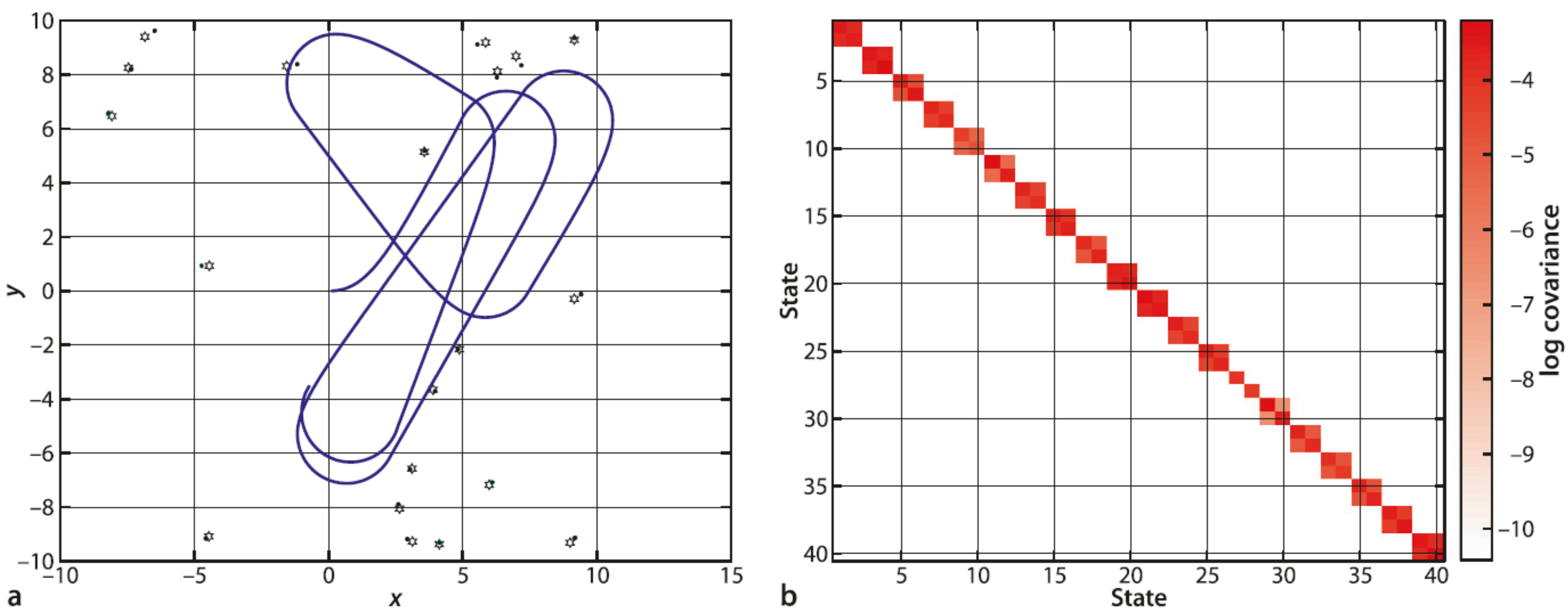
The Toolbox implementation is

```
>> map = LandmarkMap(20);
>> veh = Bicycle();  % error free vehicle
>> veh.add_driver( RandomPath(map.dim) );
>> W = diag([0.1, 1*pi/180].^2);
>> sensor = RangeBearingSensor(veh, map, 'covar', W);
>> ekf = EKF(veh, [], [], sensor, W, []);
```

the empty matrices passed to `EKF` indicate respectively that there is no estimated odometry covariance for the vehicle (the estimate is perfect), no initial vehicle state covariance, and the map is unknown. We run the simulation for 1 000 time steps

```
>> ekf.run(1000);
```

Fig. 6.9. EKF mapping results. **a** The estimated landmarks are indicated by *black dots* with 95% confidence ellipses (*green*), the true location (*black* ✡*-marker*) and the robot's path (*blue*). The landmark estimates have not fully converged on their true values and the estimated covariance ellipses can only be seen by zooming; **b** the nonzero elements of the final covariance matrix



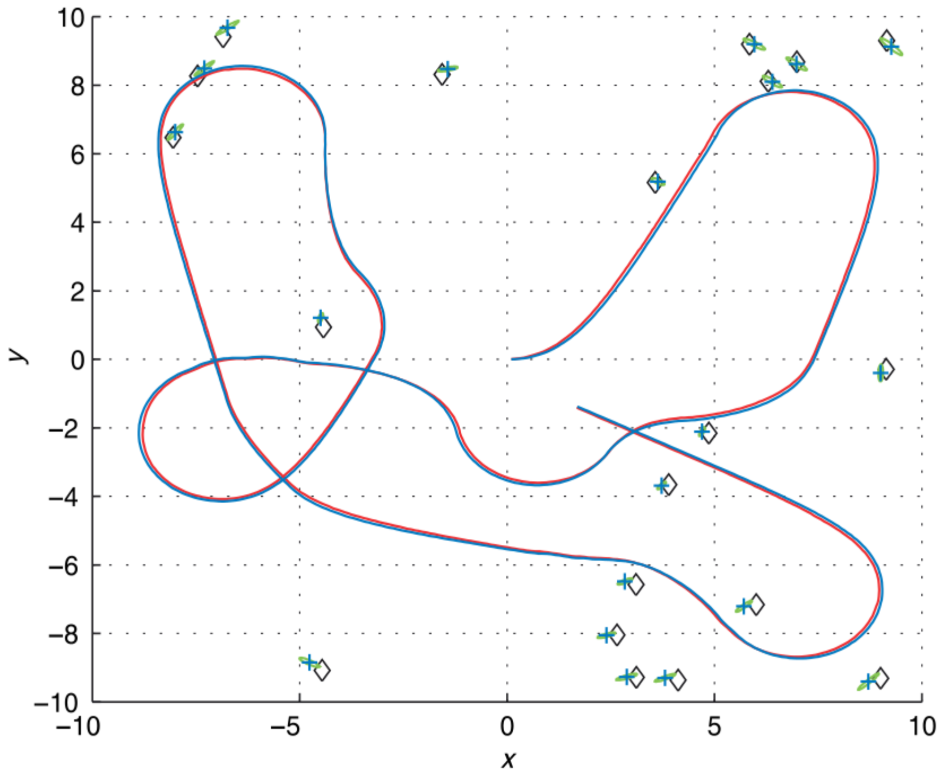
Outline

✔ Recap: Localization with a known map (landmarks)

✔ Mapping (landmarks) with known location

Simultaneous localization and mapping (EKF-SLAM)

SLAM using the EKF



Estimate both robot position and landmark positions:

$$\hat{\boldsymbol{x}} = (\underbrace{x_v, y_v, \theta_v}_{\text{robot position}}, \underbrace{x_1, y_1, x_2, y_2, \cdots x_M, y_M}_{\text{Landmark positions}})$$

Vehicle/landmark covariance

Vehicle covariance

$$\hat{\boldsymbol{P}} = \begin{pmatrix} \hat{\boldsymbol{P}}_{vv} & \hat{\boldsymbol{P}}_{vm} \\ \hat{\boldsymbol{P}}_{vm}^T & \hat{\boldsymbol{P}}_{mm} \end{pmatrix}$$

Landmark covariance

SLAM using the EKF

$$\hat{\boldsymbol{x}}^{+}\langle k+1\rangle = \boldsymbol{f}(\hat{\boldsymbol{x}}\langle k\rangle, \hat{\boldsymbol{u}}\langle k\rangle)$$

predict state one step ahead

$$\hat{P}^{+}\langle k+1\rangle = F_x \hat{P}\langle k\rangle F_x^T + F_v V F_v^T$$

project covariance one step ahead

prediction phase

$$\nu = \boldsymbol{z}^{\#}\langle k+1\rangle - h(\hat{\boldsymbol{x}}^{+}\langle k+1\rangle, \boldsymbol{p}_i)$$

new information – innovation

$$K = P^{+}\langle k+1\rangle H_x^T \left(H_x P^{+}\langle k+1\rangle H_x^T + H_w W H_w^T\right)^{-1}$$

how to distribute the innovation to states – Kalman gain

$$\hat{\boldsymbol{x}}\langle k+1\rangle = \hat{\boldsymbol{x}}^{+}\langle k+1\rangle + K\nu$$

state updated with innovation

$$\hat{P}\langle k+1\rangle = \hat{P}^{+}\langle k+1\rangle - K H_x \hat{P}^{+}\langle k+1\rangle$$

updated covariance

update phase

Fig. 6.6. Summary of extended Kalman filter algorithm showing the prediction and update phases

The state vector comprises the vehicle configuration *and* the coordinates of the M landmarks that have been observed so far

$$\boxed{\hat{\boldsymbol{x}} = (x_v, y_v, \theta_v, x_1, y_1, x_2, y_2, \cdots x_M, y_M)^T \in \mathbb{R}^{2M+3\times 1}}$$

The estimated covariance is a $(2M+3) \times (2M+3)$ matrix and has the structure

$$\hat{\boldsymbol{P}} = \begin{pmatrix} \hat{\boldsymbol{P}}_{vv} & \hat{\boldsymbol{P}}_{vm} \\ \hat{\boldsymbol{P}}_{vm}^T & \hat{\boldsymbol{P}}_{mm} \end{pmatrix}$$

where $\hat{\boldsymbol{P}}_{vv}$ is the covariance of the vehicle pose, $\hat{\boldsymbol{P}}_{mm}$ the covariance of the map landmark positions, and $\hat{\boldsymbol{P}}_{vm}$ is the correlation between vehicle and landmark states.

SLAM using the EKF

Landmark positions <u>and</u> vehicle pose are now our states:

$$\boxed{\hat{\boldsymbol{x}} = (x_v, y_v, \theta_v, x_1, y_1, x_2, y_2, \cdots x_M, y_M)^T \in \mathbb{R}^{2M+3\times 1}}$$

High level idea: Combine <u>localization</u> with known map
and <u>mapping</u> with known vehicle pose

SLAM using the EKF

Landmark positions <u>and</u> vehicle pose are now our states:

$$\boxed{\hat{\boldsymbol{x}} = \left(x_v, y_v, \theta_v, x_1, y_1, x_2, y_2, \cdots x_M, y_M\right)^T \in \mathbb{R}^{2M+3\times 1}}$$

High level idea: Combine <u>localization</u> with known map
and <u>mapping</u> with known vehicle pose

1. Write down <u>joint</u> motion model, derive Jacobian

2a. If new landmark detected, expand the state
"Insertion" Jacobian is slightly different now
Derive this from the <u>joint</u> "insertion" operator

2b. Else, update the appropriate landmark
Measurement Jacobian also slightly different now
Derive this from the <u>joint</u> measurement model

SLAM using the EKF

Landmark positions and vehicle pose are now our states:

$$\boxed{\hat{\boldsymbol{x}} = (x_v, y_v, \theta_v, x_1, y_1, x_2, y_2, \cdots x_M, y_M)^T \in \mathbb{R}^{2M+3\times 1}}$$

High level idea: Combine localization with known map
and mapping with known vehicle pose

1. Write down joint motion model, derive Jacobian

2a. If new landmark detected, expand the state
"Insertion" Jacobian is slightly different now
Derive this from the joint "insertion" operator

2b. Else, update the appropriate landmark
Measurement Jacobian also slightly different now
Derive this from the joint measurement model

… exercise to the reader (see ~~Ex3 Q3~~) – now in Ex4

Recall: Mapping using the EKF

$$\boldsymbol{H}_{p_i} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{p_i}} = \begin{pmatrix} \frac{x_i - x_v}{r} & \frac{y_i - y_v}{r} \\ -\frac{y_i - y_v}{r^2} & \frac{x_i - x_v}{r^2} \end{pmatrix}$$

$$\boldsymbol{x}\langle k\rangle' = \boldsymbol{y}(\boldsymbol{x}\langle k\rangle, \boldsymbol{z}\langle k\rangle, \boldsymbol{x}_v\langle k\rangle)$$

$$= \begin{pmatrix} \boldsymbol{x}\langle k\rangle \\ \boldsymbol{g}(\boldsymbol{x}_v\langle k\rangle, \boldsymbol{z}\langle k\rangle) \end{pmatrix}$$

$$\boldsymbol{H}_x = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right|_{w=0} = \left(0 \cdots \boldsymbol{H}_{p_i} \cdots 0\right) \in \mathbb{R}^{2\times 2M}$$

$$\hat{\boldsymbol{P}}\langle k\rangle' = \boldsymbol{Y}_z \begin{pmatrix} \hat{\boldsymbol{P}}\langle k\rangle & 0 \\ 0 & \hat{W} \end{pmatrix} \boldsymbol{Y}_z^T$$

where $\boldsymbol{Y}_z$ is the insertion Jacobian

$$\boldsymbol{Y}_z = \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{z}} = \begin{pmatrix} \boldsymbol{I}_{n\times n} & & \boldsymbol{0}_{n\times 2} \\ \boldsymbol{G}_x & \boldsymbol{0}_{2\times n-3} & \boldsymbol{G}_z \end{pmatrix}$$

$$\boldsymbol{G}_x = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$\boldsymbol{G}_z = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{z}} = \begin{pmatrix} \cos(\theta_v + \beta) & -r\sin(\theta_v + \beta) \\ \sin(\theta_v + \beta) & r\cos(\theta_v + \beta) \end{pmatrix}$$

New: SLAM using the EKF

$$\boxed{\boldsymbol{x}\langle k+1\rangle = \boldsymbol{f}\left(\boldsymbol{x}\langle k\rangle,\ \delta\langle k\rangle,\ \boldsymbol{v}\langle k\rangle\right)}$$

$$\boldsymbol{x}\langle k\rangle' = \boldsymbol{y}\left(\boldsymbol{x}\langle k\rangle, \boldsymbol{z}\langle k\rangle, \boldsymbol{x}_v\langle k\rangle\right)$$

$$= \begin{pmatrix} \boldsymbol{x}\langle k\rangle \\ \boldsymbol{g}\left(\boldsymbol{x}_v\langle k\rangle, \boldsymbol{z}\langle k\rangle\right) \end{pmatrix}$$

$$\hat{\boldsymbol{P}}\langle k\rangle' = \boldsymbol{Y}_z \begin{pmatrix} \hat{\boldsymbol{P}}\langle k\rangle & 0 \\ 0 & \hat{\boldsymbol{W}} \end{pmatrix} \boldsymbol{Y}_z^T$$

where $\boldsymbol{Y}_z$ is the insertion Jacobian

$$\boldsymbol{Y}_z = \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{z}} = \begin{pmatrix} \boldsymbol{I}_{n\times n} & & \boldsymbol{0}_{n\times 2} \\ \boldsymbol{G}_x & \boldsymbol{0}_{2\times n-3} & \boldsymbol{G}_z \end{pmatrix}$$

$$\boldsymbol{H}_{p_i} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{p_i}} = \begin{pmatrix} \dfrac{x_i - x_v}{r} & \dfrac{y_i - y_v}{r} \\ -\dfrac{y_i - y_v}{r^2} & \dfrac{x_i - x_v}{r^2} \end{pmatrix}$$

$$\boxed{\boldsymbol{H}_x = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right|_{w=0} = \left(\boldsymbol{H}_{x_v} \cdots 0 \cdots \boldsymbol{H}_{p_i} \cdots 0\right) \in \mathbb{R}^{2\times(2M+3)}}$$

$$\boxed{\boldsymbol{G}_x = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} = \begin{pmatrix} 1 & 0 & -r\sin\left(\theta_v + \beta\right) \\ 0 & 1 & r\cos\left(\theta_v + \beta\right) \end{pmatrix}}$$

$$\boldsymbol{G}_z = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{z}} = \begin{pmatrix} \cos\left(\theta_v + \beta\right) & -r\sin\left(\theta_v + \beta\right) \\ \sin\left(\theta_v + \beta\right) & r\cos\left(\theta_v + \beta\right) \end{pmatrix}$$

SLAM using the EKF

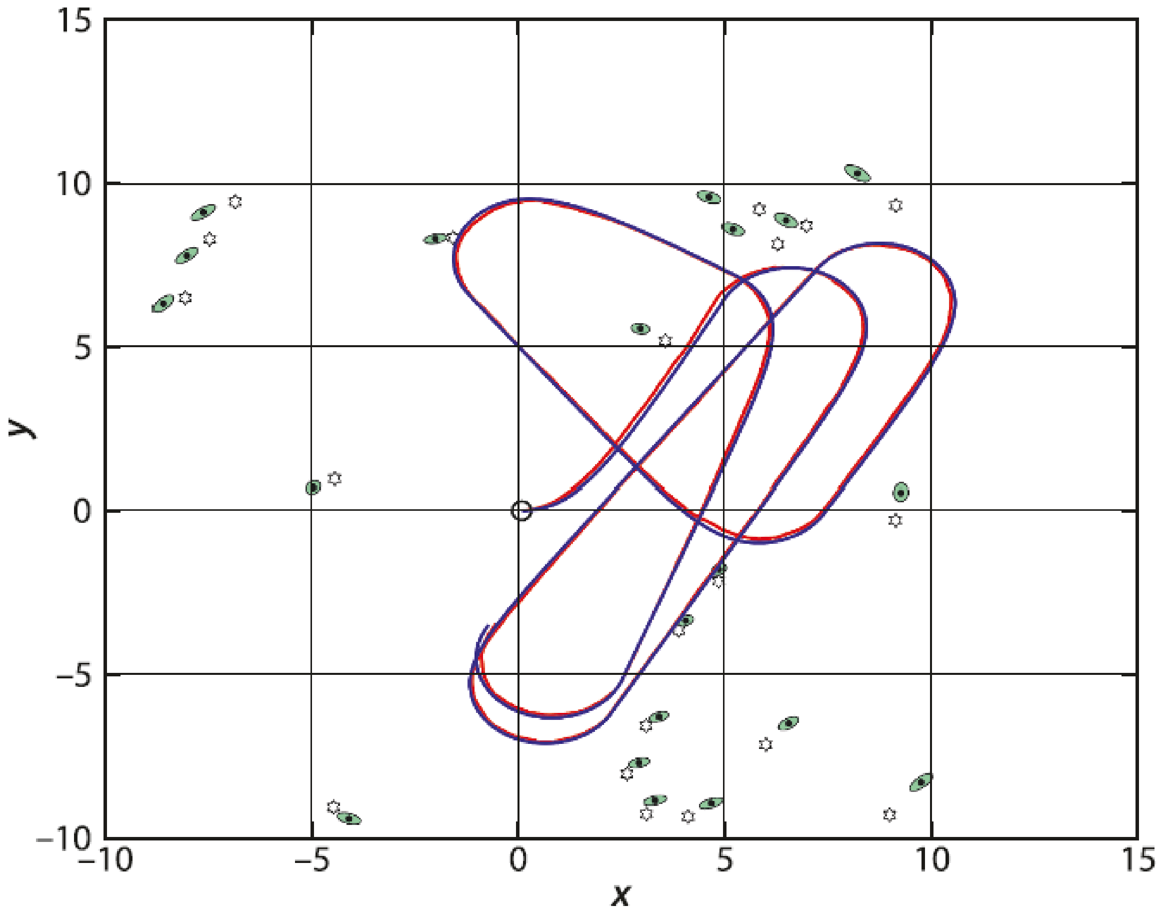
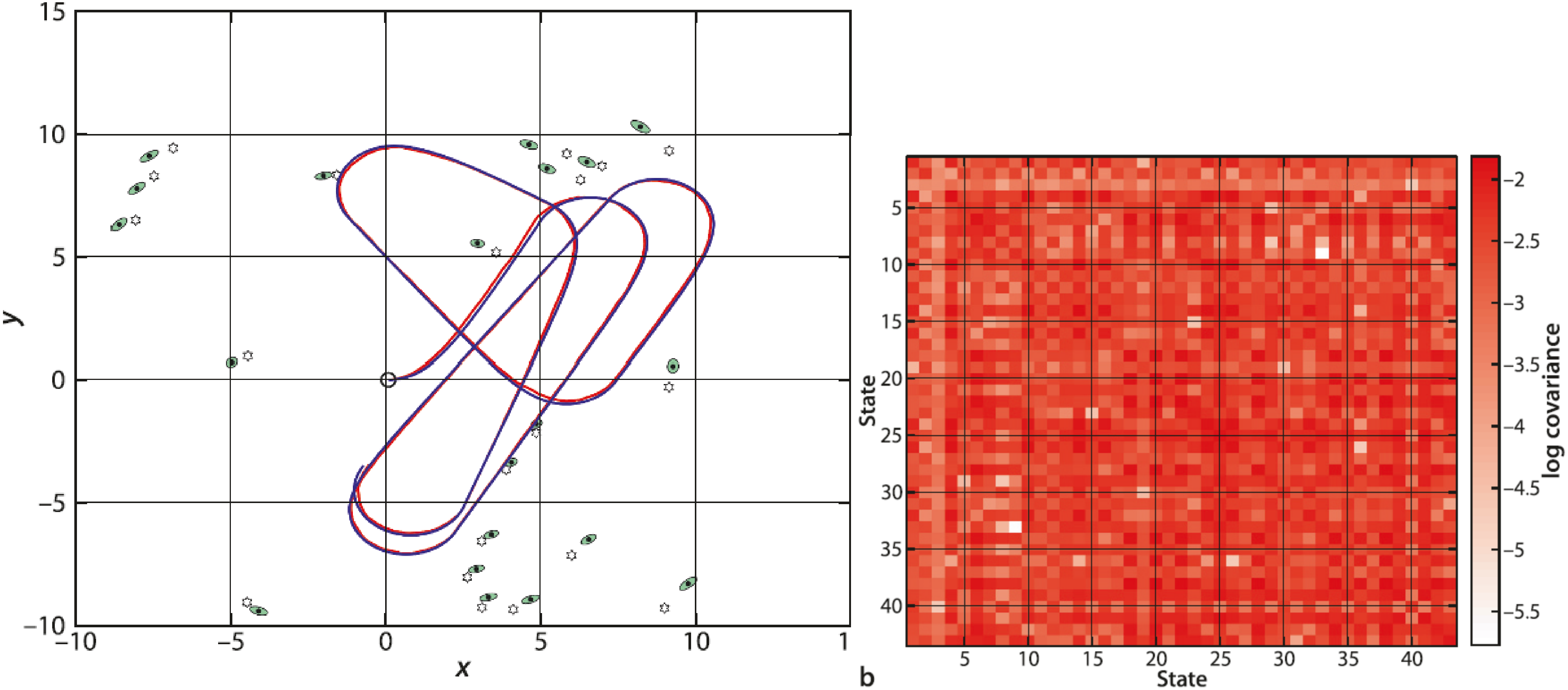


Fig. 6.11.
Simultaneous localization and mapping showing the true (*blue*) and estimated (*red*) robot path superimposed on the true map (*black* ✡*-marker*). The estimated map features are indicated by *black dots* with 95% confidence ellipses (*green*)

SLAM using the EKF



SLAM using the EKF

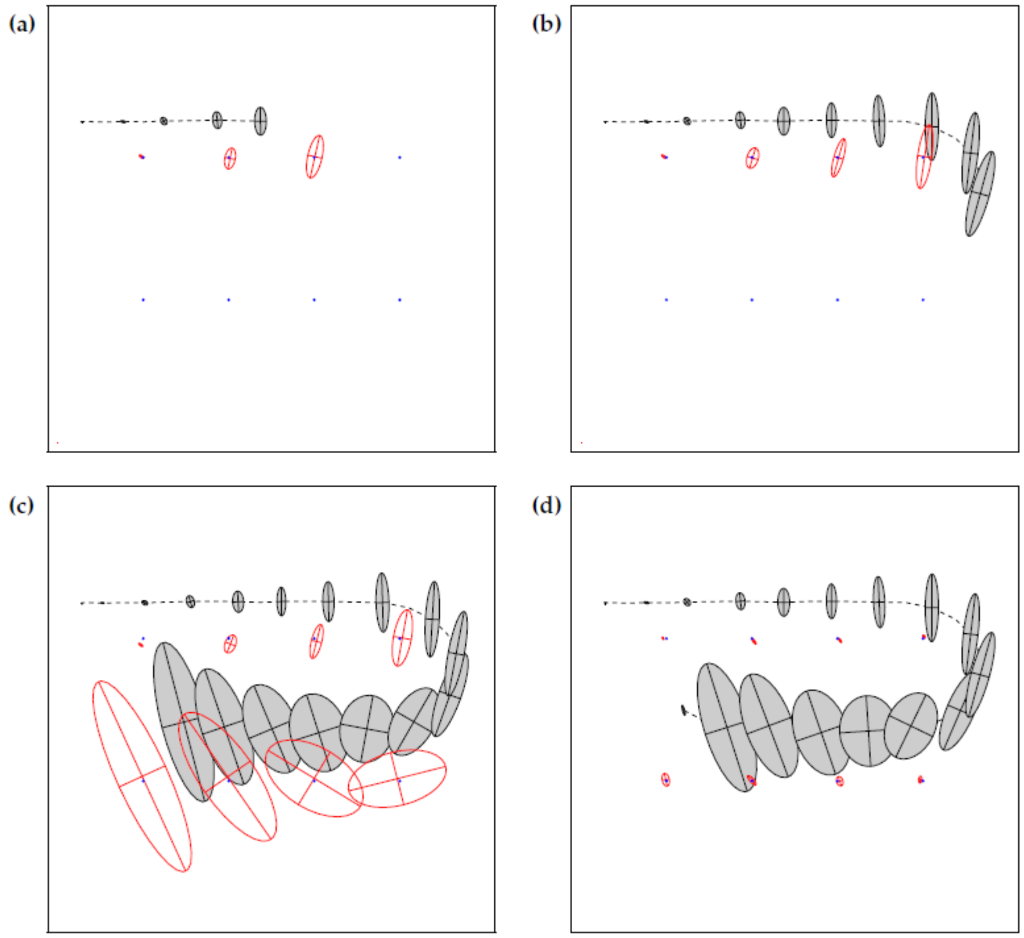
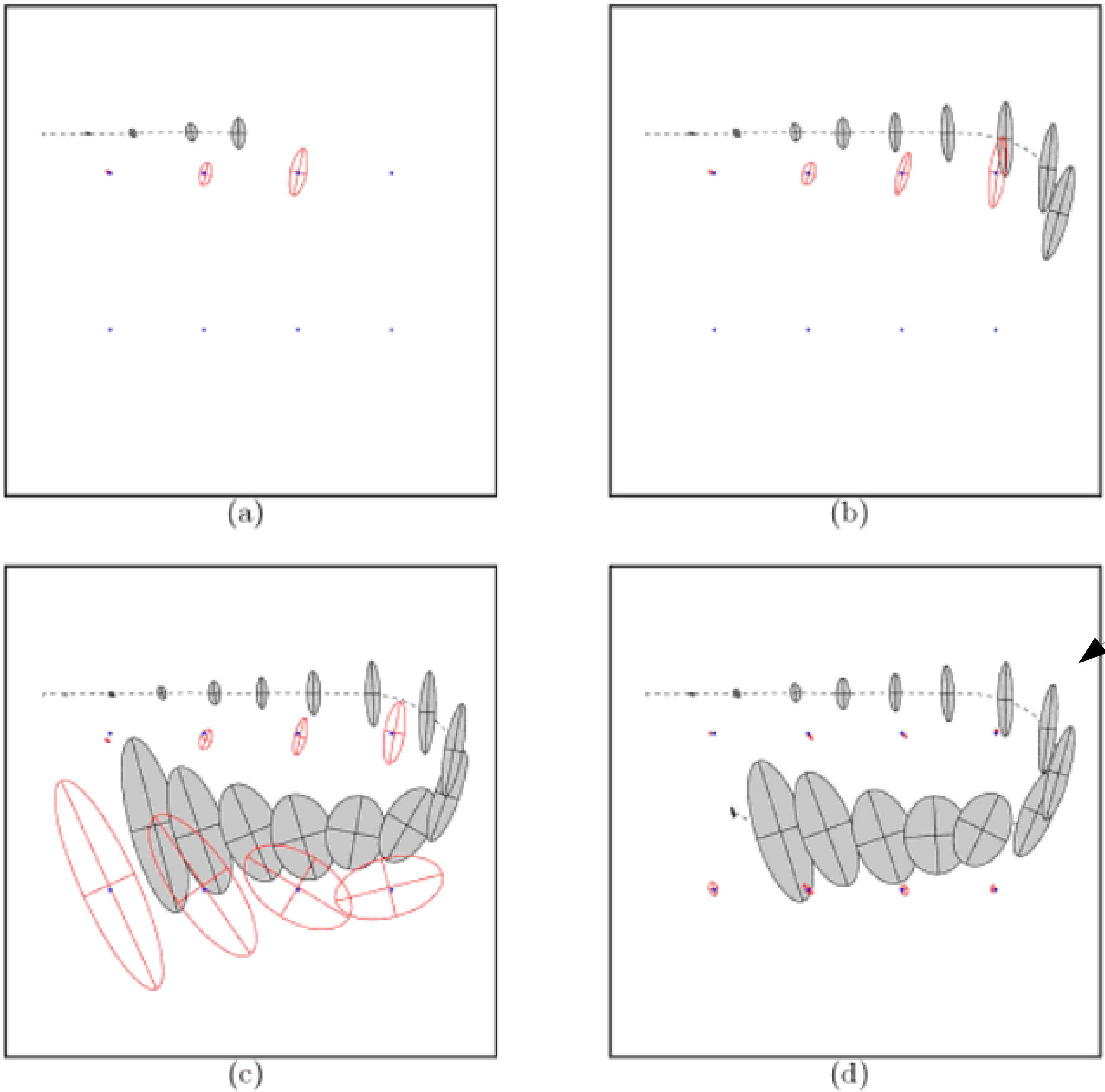


Figure 10.3 EKF applied to the online SLAM problem. The robot's path is a dotted line, and its estimates of its own position are shaded ellipses. Eight distinguishable landmarks of unknown location are shown as small dots, and their location estimates are shown as white ellipses. In (a)–(c) the robot's positional uncertainty is increasing, as is its uncertainty about the landmarks it encounters. In (d) the robot senses the first landmark again, and the uncertainty of *all* landmarks decreases, as does the uncertainty of its current pose. Image courtesy of Michael Montemerlo, Stanford University.

SLAM using the EKF



Landmark covariance drops significantly as soon as “loop closure” occurs.

SLAM using the EKF

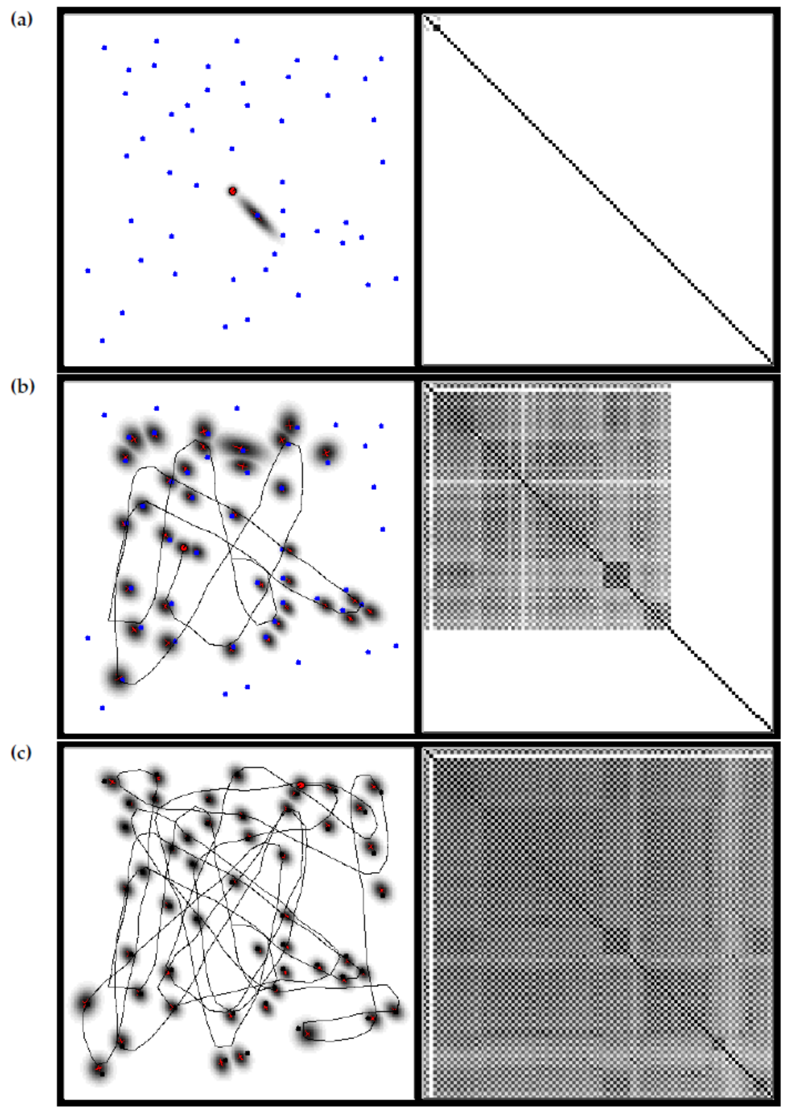


Figure 10.4 EKF SLAM with known data association in a simulated environment. The map is shown on the left, with the gray-level corresponding to the uncertainty of each landmark. The matrix on the right is the correlation matrix, which is the normalized covariance matrix of the posterior estimate. After some time, all x- and all y-coordinate estimates become fully correlated.

SLAM using the EKF

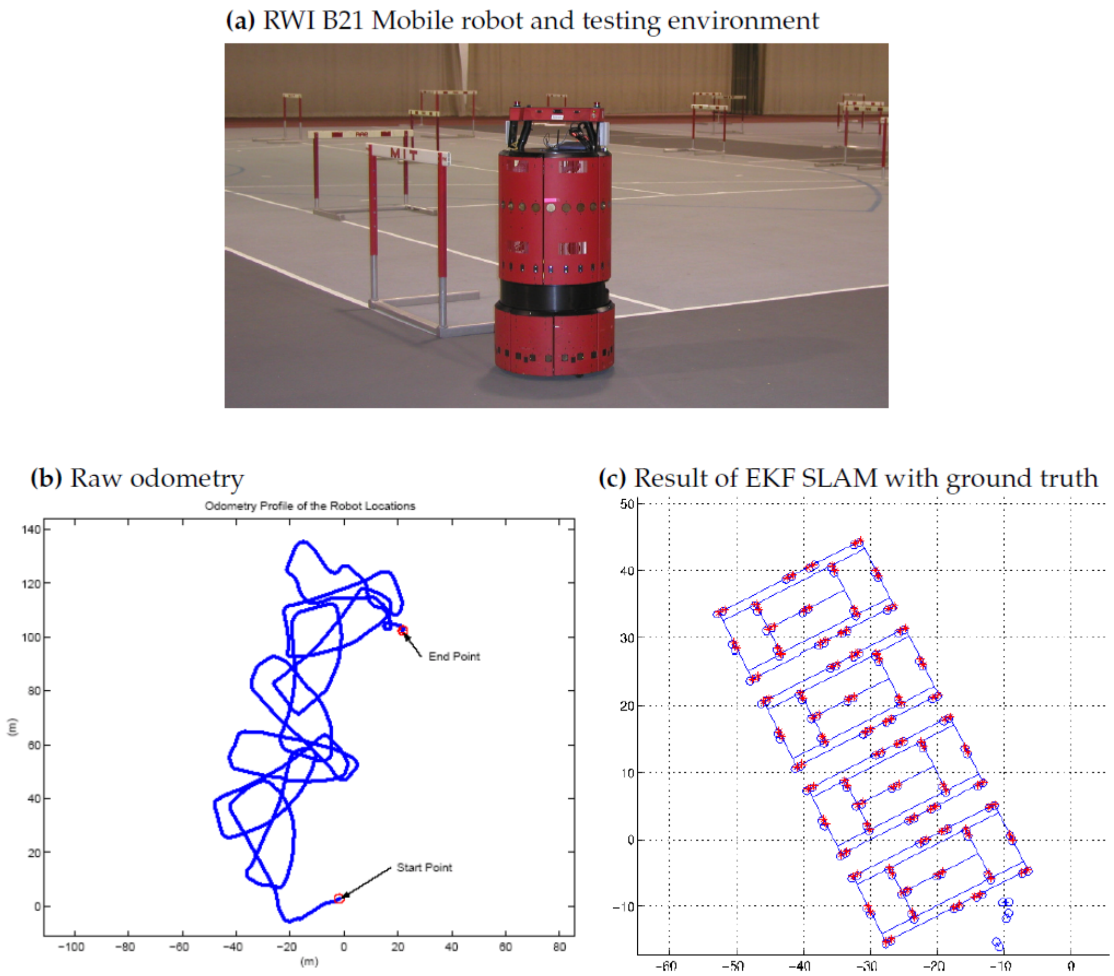


Figure 10.7 (a) The MIT B21 mobile robot in a calibrated testing facility. (b) Raw odometry of the robot, as it is manually driven through the environment. (c) The result of EKF SLAM is a highly accurate map. The image shows the estimated map overlayed on a manually constructed map. All images and results are courtesy of John Leonard and Matthew Walter, MIT.

Feedback

Piazza thread: 3/7 Lec 13 Feedback

Please post your answers to the following anonymously.

1. What did you like today?

2. What was unclear?

3. How is the semester going?
   Any changes we should aim for (after spring break)?

4. Any additional feedback / comments?