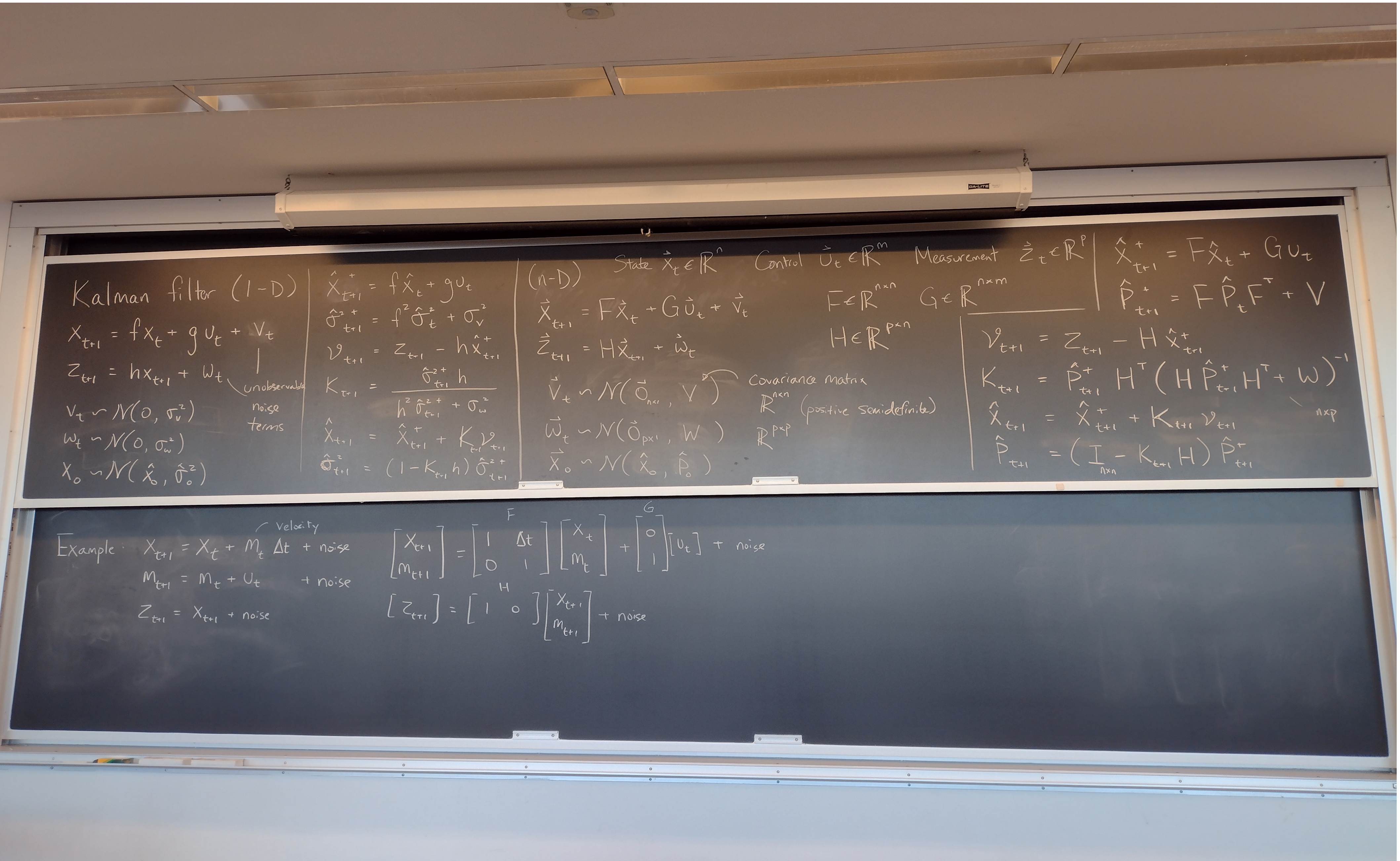
CS 4610/5335 – Lecture 12
Localization and Mapping

Lawson L.S. Wong
Northeastern University
3/2/22

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

On the board: n-D Kalman Filter



Recap: Extended Kalman filter

H.2 Nonlinear Systems – Extended Kalman Filter

For the case where the system is not linear it can be described generally by two functions: the state transition (the motion model in robotics) and the sensor model

$$\boxed{\begin{aligned} \boldsymbol{x}\langle k{+}1\rangle &= \boldsymbol{f}\big(\boldsymbol{x}\langle k\rangle, \boldsymbol{u}\langle k\rangle, \boldsymbol{v}\langle k\rangle\big) && \text{(H.9)} \\ \boldsymbol{z}\langle k\rangle &= \boldsymbol{h}\big(\boldsymbol{x}\langle k\rangle, \boldsymbol{w}\langle k\rangle\big) && \text{(H.10)} \end{aligned}}$$

and as before we represent model uncertainty, external disturbances and sensor noise by Gaussian random variables $\boldsymbol{v}$ and $\boldsymbol{w}$.

We linearize the state transition function about the current state estimate $\hat{\boldsymbol{x}}_k$ as shown in Fig. H.2 resulting in

$$\boldsymbol{x}'\langle k{+}1\rangle \approx \boldsymbol{F}_x \boldsymbol{x}'\langle k\rangle + \boldsymbol{F}_u \boldsymbol{u}\langle k\rangle + \boldsymbol{F}_v \boldsymbol{v}\langle k\rangle \tag{H.11}$$

$$\boldsymbol{z}'\langle k\rangle \approx \boldsymbol{H}_x \boldsymbol{x}'\langle k\rangle + \boldsymbol{H}_w \boldsymbol{w}\langle k\rangle \tag{H.12}$$

where $\boldsymbol{F}_x = \partial \boldsymbol{f} / \partial \boldsymbol{x} \in \mathbb{R}^{n\times n}$, $\boldsymbol{F}_u = \partial \boldsymbol{f} / \partial \boldsymbol{u} \in \mathbb{R}^{n\times m}$, $\boldsymbol{F}_v = \partial \boldsymbol{f} / \partial \boldsymbol{v} \in \mathbb{R}^{n\times n}$, $\boldsymbol{H}_x = \partial \boldsymbol{h} / \partial \boldsymbol{x} \in \mathbb{R}^{p\times n}$ and $\boldsymbol{H}_w = \partial \boldsymbol{h} / \partial \boldsymbol{w} \in \mathbb{R}^{p\times p}$ are Jacobians of the functions $\boldsymbol{f}(\cdot)$ and $\boldsymbol{h}(\cdot)$.

Recap: Extended Kalman filter

H.2 Nonlinear Systems – Extended Kalman Filter

For the case where the system is not linear it can be described generally by two functions: the state transition (the motion model in robotics) and the sensor model

$$\boldsymbol{x}\langle k+1\rangle = \boldsymbol{f}\big(\boldsymbol{x}\langle k\rangle, \boldsymbol{u}\langle k\rangle, \boldsymbol{v}\langle k\rangle\big) \tag{H.9}$$

$$\boldsymbol{z}\langle k\rangle = \boldsymbol{h}\big(\boldsymbol{x}\langle k\rangle, \boldsymbol{w}\langle k\rangle\big) \tag{H.10}$$

and as before we represent model uncertainty, external disturbances and sensor noise by Gaussian random variables $\boldsymbol{v}$ and $\boldsymbol{w}$.

We linearize the state transition function about the current state estimate $\hat{\boldsymbol{x}}_k$ as shown in Fig. H.2 resulting in

$$\boxed{\boldsymbol{x}'\langle k+1\rangle \approx \boldsymbol{F}_x\boldsymbol{x}'\langle k\rangle + \boldsymbol{F}_u\boldsymbol{u}\langle k\rangle + \boldsymbol{F}_v\boldsymbol{v}\langle k\rangle} \tag{H.11}$$

$$\boxed{\boldsymbol{z}'\langle k\rangle \approx \boldsymbol{H}_x\boldsymbol{x}'\langle k\rangle + \boldsymbol{H}_w\boldsymbol{w}\langle k\rangle} \tag{H.12}$$

where $\boldsymbol{F}_x = \partial\boldsymbol{f}/\partial\boldsymbol{x} \in \mathbb{R}^{n\times n}$, $\boldsymbol{F}_u = \partial\boldsymbol{f}/\partial\boldsymbol{u} \in \mathbb{R}^{n\times m}$, $\boldsymbol{F}_v = \partial\boldsymbol{f}/\partial\boldsymbol{v} \in \mathbb{R}^{n\times n}$, $\boldsymbol{H}_x = \partial\boldsymbol{h}/\partial\boldsymbol{x} \in \mathbb{R}^{p\times n}$ and $\boldsymbol{H}_w = \partial\boldsymbol{h}/\partial\boldsymbol{w} \in \mathbb{R}^{p\times p}$ are Jacobians of the functions $\boldsymbol{f}(\cdot)$ and $\boldsymbol{h}(\cdot)$.

Recap: Extended Kalman filter

prediction phase

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

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

update 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

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

Recap: Extended Kalman filter

Procedure EKF

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

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*

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

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

– *the update step*

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

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

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

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

Algorithm H.1.
Procedure EKF

Recap: 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*

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

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

– *the update step*

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

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

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

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

Algorithm H.1. Procedure EKF

Recap: 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*

$$\boxed{\begin{aligned}&\hat{\boldsymbol{x}}^+\langle k+1\rangle = \boldsymbol{f}\left(\hat{\boldsymbol{x}}\langle k\rangle, \boldsymbol{u}\langle k\rangle\right) \quad \textit{// predict state at next time step}\\ &\hat{\boldsymbol{P}}^+\langle k+1\rangle = \boldsymbol{F}_x\hat{\boldsymbol{P}}\langle k\rangle\boldsymbol{F}_x^T + \boldsymbol{F}_v\hat{\boldsymbol{V}}\boldsymbol{F}_v^T \quad \textit{// predict cov ariance at next time step}\end{aligned}}$$

– *the update step*

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

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

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

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

Algorithm H.1. Procedure EKF

Recap: 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*

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

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

$-$ *the update step*

$$\boxed{\begin{aligned} &\nu = \boldsymbol{z}\langle k+1\rangle - \boldsymbol{h}\left(\hat{\boldsymbol{x}}^+\langle k+1\rangle\right) \quad \text{// innovation : measured} - \text{predicted sensor value} \\ &\boldsymbol{K} = \boldsymbol{P}^+\langle k+1\rangle\boldsymbol{H}_x^T\left[\boldsymbol{H}_x\boldsymbol{P}^+\langle k+1\rangle\boldsymbol{H}_x^T + \boldsymbol{H}_w\hat{\boldsymbol{W}}\boldsymbol{H}_w^T\right]^{-1} \quad \text{// Kalman gain}\end{aligned}}$$

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

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

Algorithm H.1.
Procedure EKF

Recap: 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*

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

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

– *the update step*

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

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

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

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

Algorithm H.1.
Procedure EKF

Outline

✔ Recap: EKF

Localization with a known map (landmarks)

Mapping (landmarks) with known location

EKF-based mobile robot localization

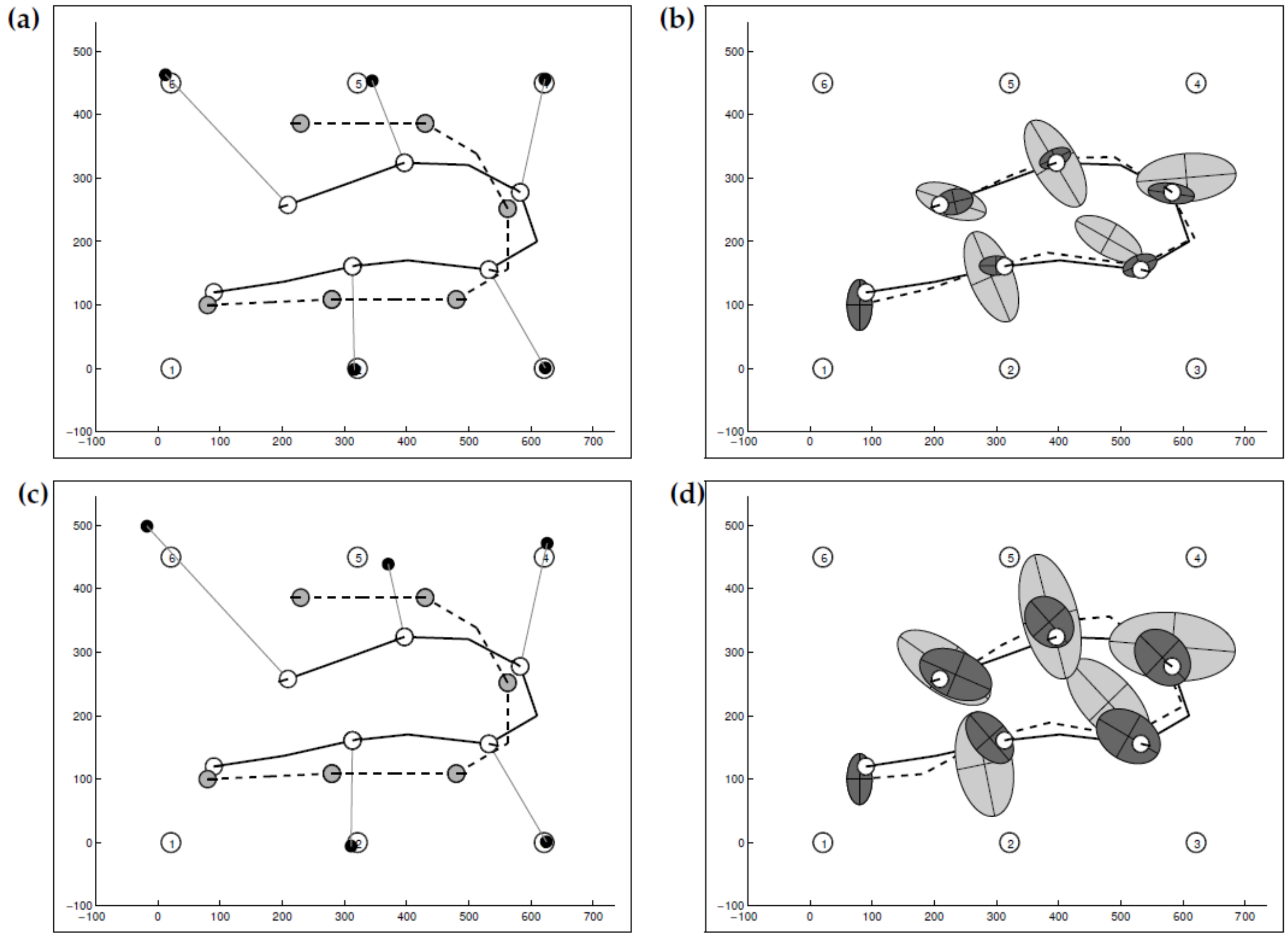


Figure 7.11 EKF-based localization with an accurate (upper row) and a less accurate (lower row) landmark detection sensor. The dashed lines in the left panel indicate the robot trajectories as estimated from the motion controls. The solid lines represent the true robot motion resulting from these controls. 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.

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 onboard 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

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

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

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

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

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

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

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

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

– the update step

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

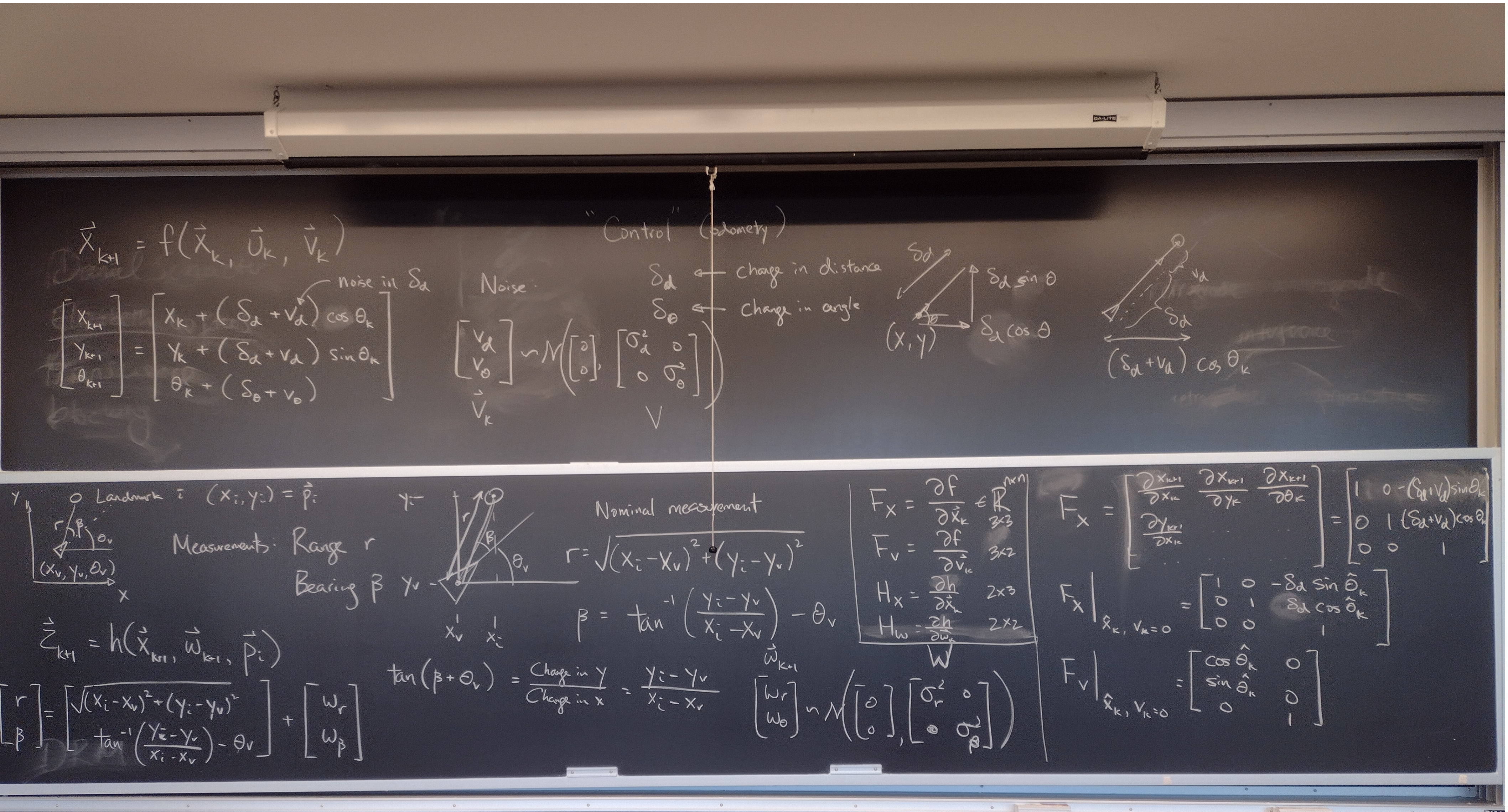
$$\boldsymbol{K} = \boldsymbol{P}^+\langle k+1\rangle\boldsymbol{H}_x^T\left[\boldsymbol{H}_x\boldsymbol{P}^+\langle k+1\rangle\boldsymbol{H}_x^T + \boldsymbol{H}_w\hat{\boldsymbol{W}}\boldsymbol{H}_w^T\right]^{-1} \quad \textit{// Kalman gain}$$

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

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

Algorithm H.1.
Procedure EKF

On the board: EKF localization



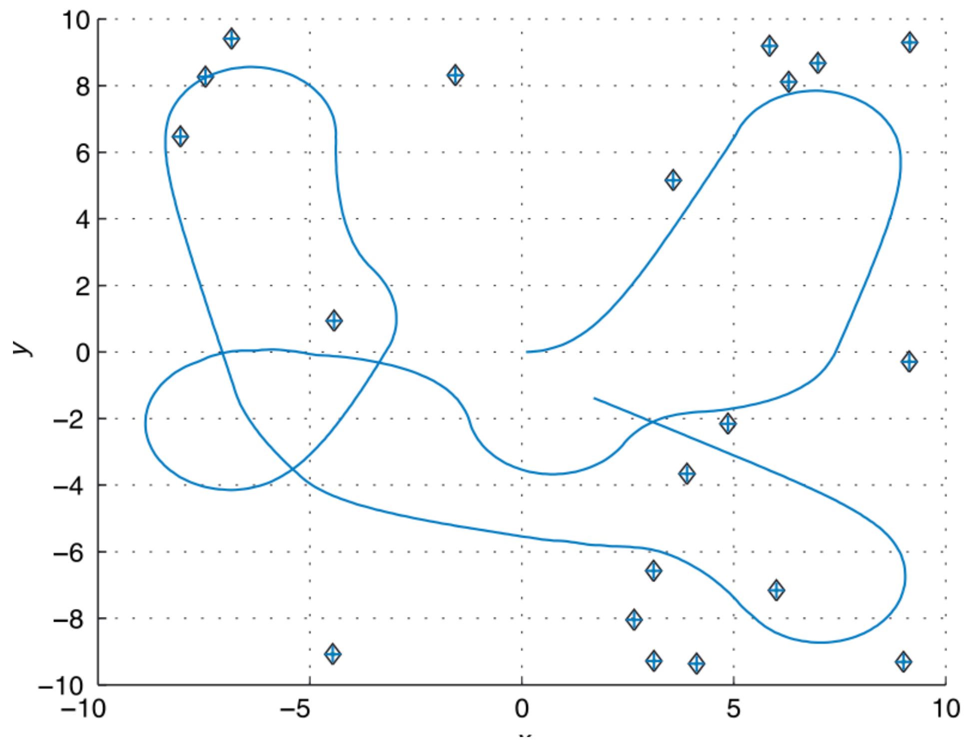
Outline

✔ Recap: EKF

✔ Localization with a known map (landmarks)

Mapping (landmarks) with known location

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

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

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

EKF mapping: Prediction step

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

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

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

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

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

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:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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):

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

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

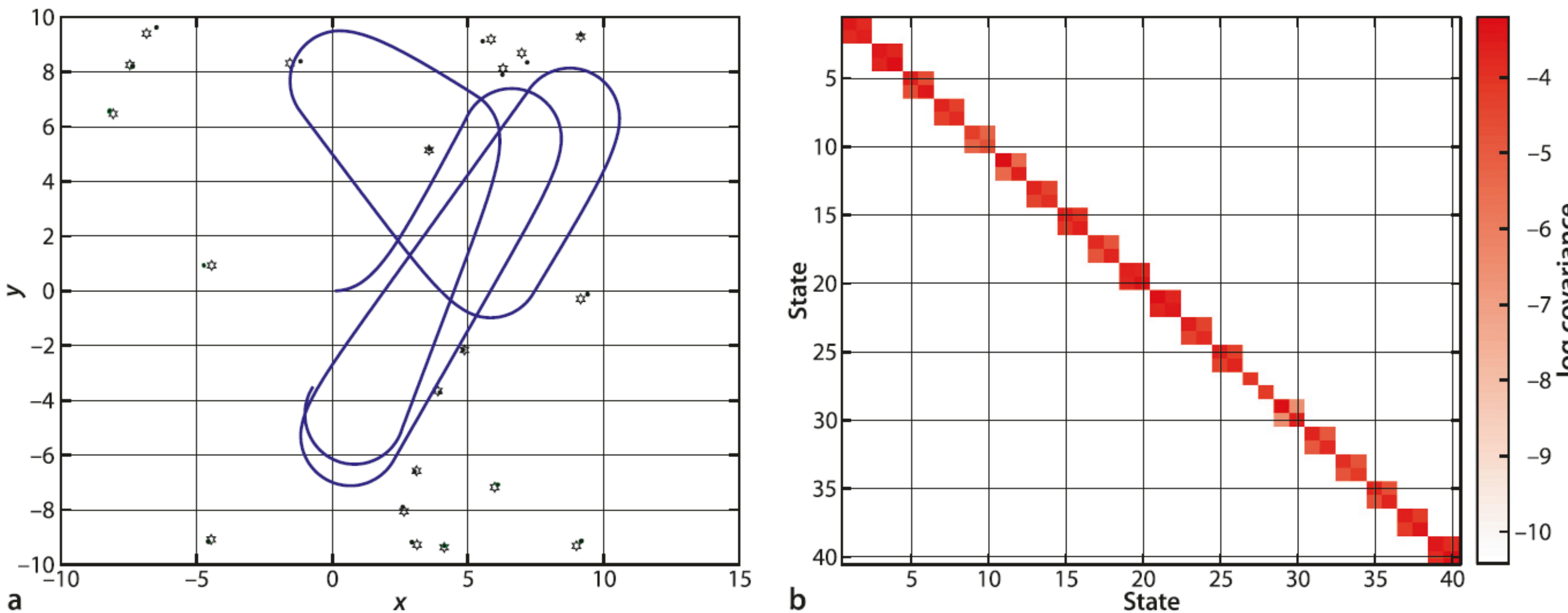
The Toolbox implementation is

```
>> 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, []);
```

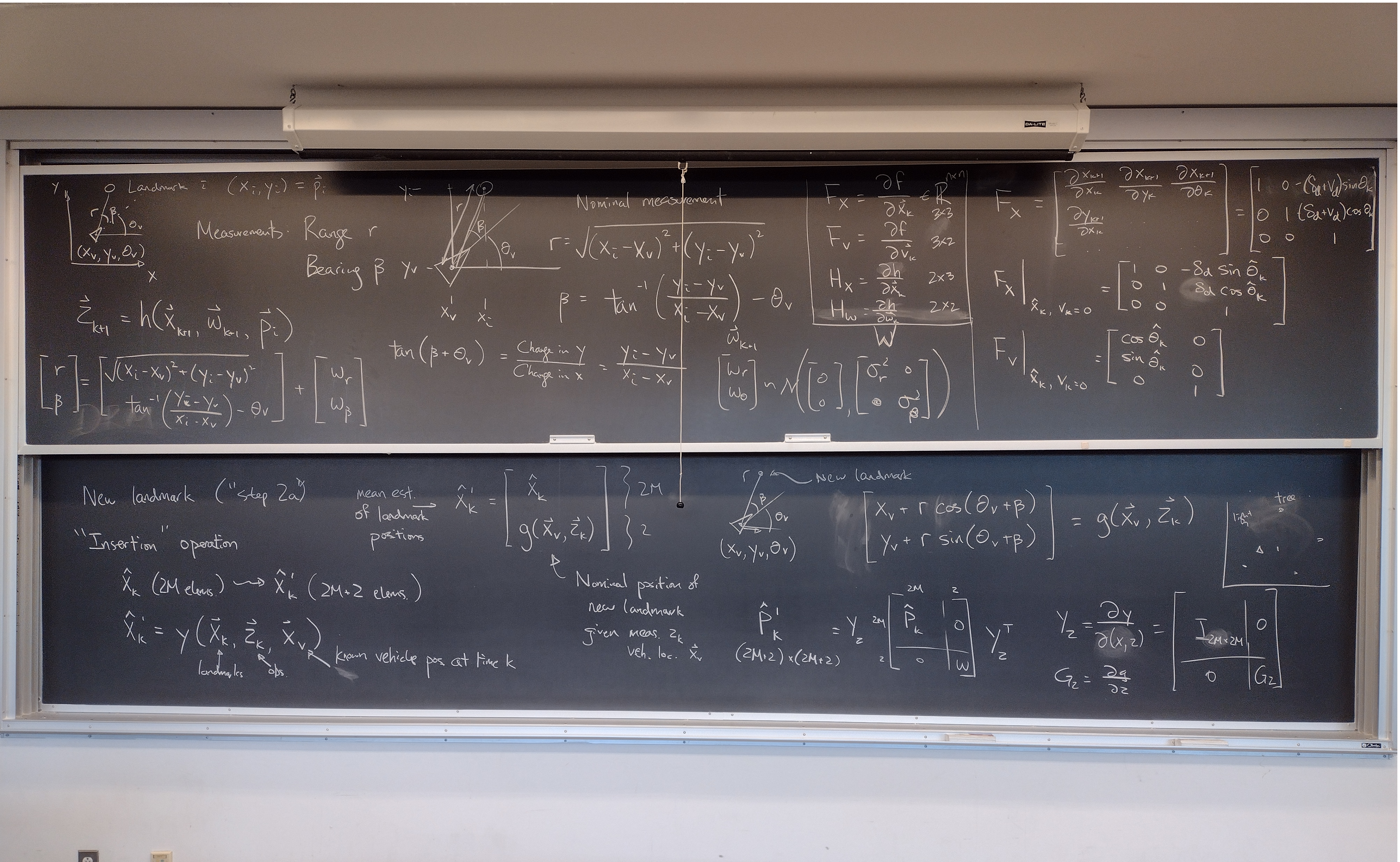
the empty matrices passed to `EKF` indicate respectively that there is no estimated odometry covariance for the vehicle (the estimate is perfect), no initial vehicle state covariance, and the map is unknown. We run the simulation for 1 000 time steps

```
>> 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



On the board: EKF mapping



Feedback

Piazza thread: 3/2 Lec 12 Feedback

Please post your answers to the following anonymously.

1. What did you like today?

2. What was unclear?

3. Any additional feedback / comments?