CIE 482: Path Planning & Navigation for Mobile Robots

Lecture 17: Kalman Filtering II

جامعة الملك فهد للبترول والمعادن
King Fahd University of Petroleum & Minerals

Ramy Rashad
Assistant Professor
CIE Department

Outline

- Recap last lecture
- Examples
- Kalman Filter Algorithm
- Application: 1D robot position estimation

Outline

- Recap last lecture
- Examples
- Kalman Filter Algorithm
- Application: 1D robot position estimation

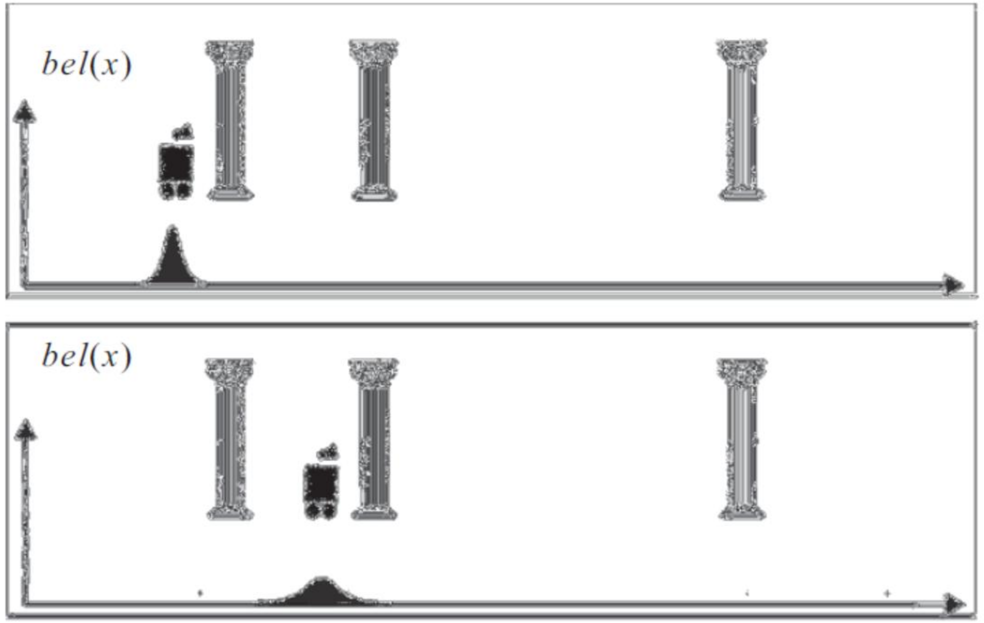
Recap: Belief Representation

- We represent the robot not just using a single possible position but also a possibly infinite set of positions.
- In this approach, the belief is modeled by a probability distribution function $bel(x)$.
- In this approach, beliefs are

represented by conditional

probability distributions:

$$bel(x) = p(x|y)$$



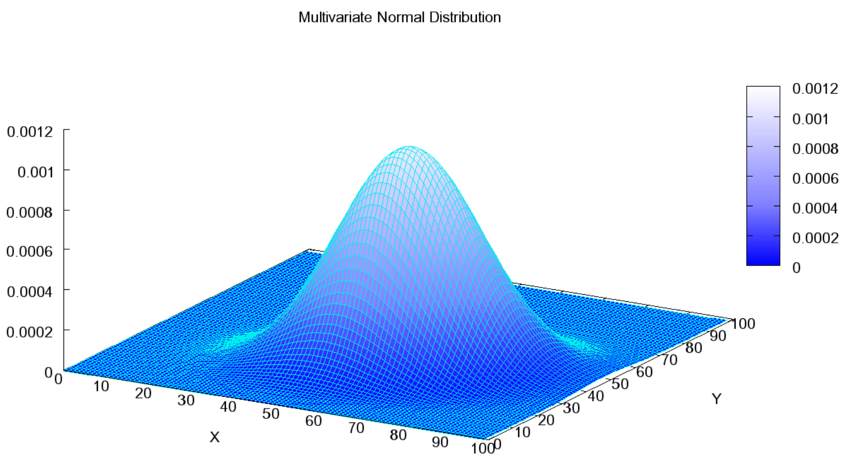
The belief function $bel(x)$ is our best guess of the robot's state x.

Recap: Multivariate Gaussian Distribution

- In case the random variable $x \in \mathbb{R}^k$ is a k-dimensional vector, we will have a multivariate normal distribution characterized by the density function:

$$p(x) = \frac{1}{(2\pi)^{k/2}\sqrt{\det(\Sigma)}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right)$$

where $\mu \in \mathbb{R}^k$ & $\Sigma \in \mathbb{R}^{k\times k}$ denote the mean & covariance matrix.



Recap: Summary of Bayesian Filtering

Propreoceptive sensor measurement u_k

State Transition Model
$p(x_k | x_{k-1}, Z_{k-1}, U_k)$

1 **Prediction update**

$\boldsymbol{bel}\,(\boldsymbol{x_{k-1}})$
$= p(x_{k-1} | Z_{k-1}, U_{k-1})$

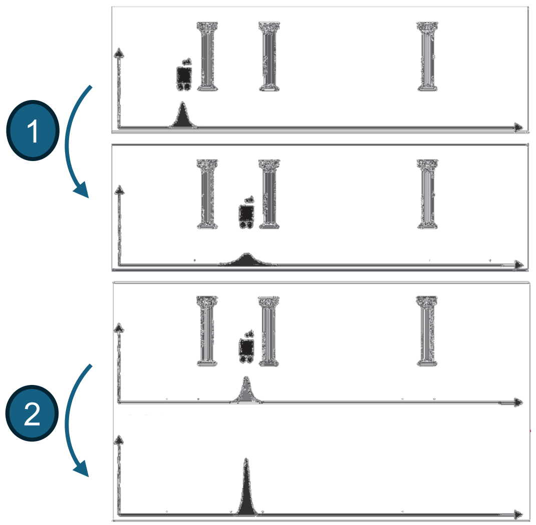
$\boldsymbol{bel}\,(\boldsymbol{x_k})$
$= p(x_k | Z_k, U_k)$

$\overline{\boldsymbol{bel}}\,(\boldsymbol{x_k})$
$= p(x_k | Z_{k-1}, U_k)$

Correction update 2

Sensor Model
$p(z_k | x_k)$

Exteroceptive sensor measurement z_k



Recap: Kalman Filter

u_k

Linear

State Transition Model

$x_k = F_k\, x_{k-1} + B_k\, u_k + v_k$

1 **Prediction update**

Gaussian

$\boldsymbol{bel}\,(\boldsymbol{x_{k-1}}) = p(x_{k-1}|Z_{k-1}, U_{k-1})$

$\boldsymbol{bel}\,(\boldsymbol{x_k}) = p(x_k|Z_k, U_k)$

Gaussian

Gaussian

$\overline{\boldsymbol{bel}}\,(\boldsymbol{x_k}) = p(x_k|Z_{k-1}, U_k)$

Correction update 2

Sensor Model

$\beta_k \coloneqq z_k - H_k x_k + w_k$

Linear

z_k

Recap: Probabilistic Models

- The state transition model and measurement models are not deterministic, but probabilistic with Gaussian distributions.

	State transition model	**Measurement model**
Difference Equation	$x_k = F_k\, x_{k-1} + B_k\, u_k + v_k$	$z_k = H_k\, x_k + w_k$
Probability Distribution Function	$p(x_k \mid x_{k-1}, u_k)$	$p(z_k \mid x_k)$
Mean of Gaussian PDF	$\hat{x}_k$	$\hat{z}_k$
Covariance of Gaussian PDF	P_k	R_k

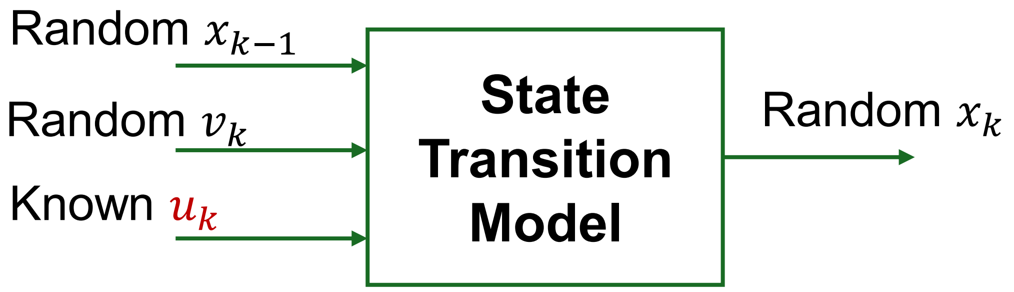
In summary, Kalman Filter assumes:

- $p(x_k \mid x_{k-1}, u_k) = N(\hat{x}_k, P_k)$
- $p(z_k \mid x_k) = N(\hat{z}_k, R_k)$

Recap: State Transition Probabilistic Model

$$\boxed{x_k = F_k\, x_{k-1} + B_k\, u_k + v_k}$$

- If x_{k-1} and v_k are two random variables that are independent and have a Gaussian distribution, then x_k will also have a Gaussian distribution.
- Question: If we know the mean and covariance of x_{k-1} and v_k, how can we compute the mean and covariance of x_k?



Note that u_k is not a random variable but a known input.

Recap: State Transition Probabilistic Model

$$x_k = F_k\, x_{k-1} + B_k\, u_k + v_k$$

- If x_{k-1} and v_k are two random variables that are independent and have a Gaussian distribution, then x_k will also have a Gaussian distribution.
- If $x_{k-1} \sim N(\hat{x}_{k-1}, P_{k-1})$ and $v_k \sim N(0, Q_k)$ then $x_k \sim N(\hat{x}_k, P_k)$ with

Mean

$$\hat{x}_k = F_k \hat{x}_{k-1} + B_k\, u_k$$

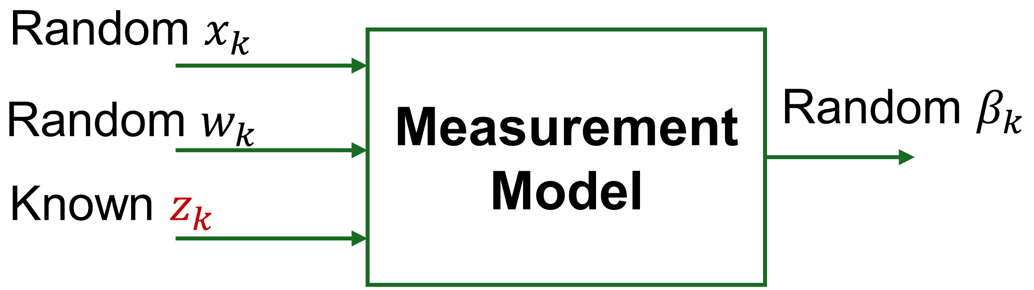
Covariance

$$P_k = F_k P_{k-1} F_k^{\top} + Q_K$$

Recap: Measurement Probabilistic Model

$$\boxed{\beta_k := z_k - H_k x_k + w_k}$$

- If x_k and w_k are two random variables that are independent and have a Gaussian distribution, then β_k will also have a Gaussian distribution.
- Question: If we know the mean and covariance of x_k and w_k, how can we compute the mean and covariance of β_k?



Note that z_k is not a random variable but a known input.

Recap: Measurement Probabilistic Model

$$\beta_k := z_k - H_k x_k + w_k$$

- If x_k and w_k are two random variables that are independent and have a Gaussian distribution, then β_k will also have a Gaussian distribution.
- If $x_k \sim N(\hat{x}_k, P_k)$ and $w_k \sim N(0, R_k)$ then $\beta_k \sim N\left(\hat{\beta}_k, S_k\right)$ with

Mean

$$\hat{\beta}_k = z_k - H_k \hat{x}_k$$

Covariance

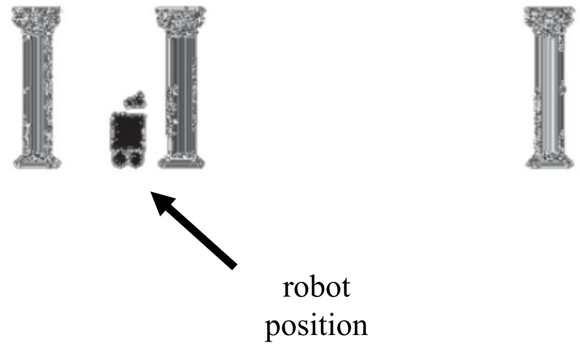
$$S_k = H_k P_k H_k^\top + R_K$$

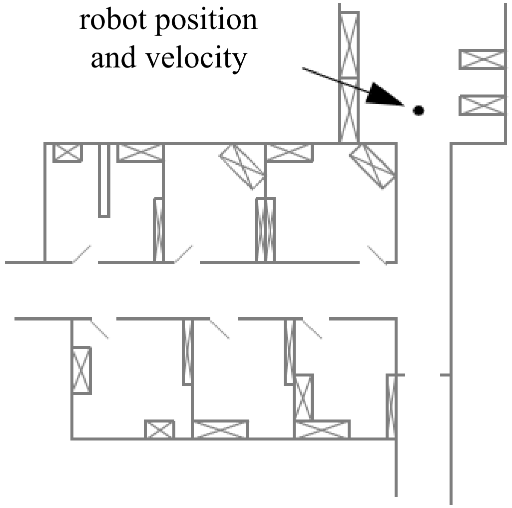
Outline

- Recap last lecture
- Examples
- Kalman Filter Algorithm
- Application: 1D robot position estimation

Example Estimation Problems

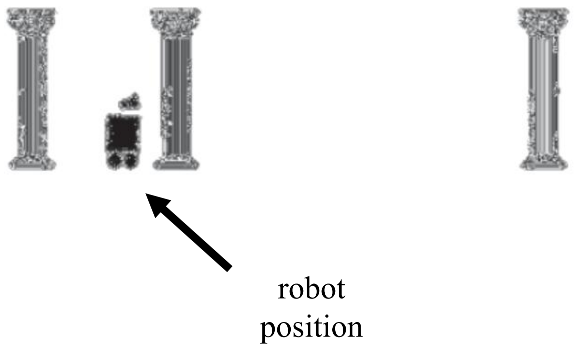
- 1D robot position:
  - State $x(t) = \xi(t) \in \mathbb{R}$
  - Velocity sensor $u(t) \in \mathbb{R}$
  - Position sensor $\mathrm{z}(t) \in \mathbb{R}$
- 2D robot position and velocity
  - State $x(t) = (\xi(t), v(t)) \in \mathbb{R}^4$
  - Acceleration sensor $u(t) \in \mathbb{R}^2$
  - Position sensor $\mathrm{z}(t) \in \mathbb{R}^2$





Example 1:

- Robot State: 1D position
  - $x(t) = \xi(t) \in \mathbb{R}$
- Control input: PC velocity sensor at 1 Hz
  - $u(t) = v(t) + \eta(t), \qquad \eta(t) \sim N(0, \sigma_u^2)$
- Observation input: EC position sensor at 1/Ts Hz
  - $z(t) = \xi(t) + w(t), \qquad w(t) \sim N(0, \sigma_z^2)$
- Kinematic model:
  - $\dot{x}(t) = \dot{\xi}(t) = v(t)$



PC: proprioceptive
EC: exteroceptive

Example 1:

- Kinematic model:
  - $\dot{x}(t) = \dot{\xi}(t) = v(t)$

Discretization rule:

$$\dot{x}_k \approx \frac{x_k - x_{k-1}}{\Delta t}$$

- State Transition model:

$$x_k = x_{k-1} + \Delta t\, u_k + \eta_k$$

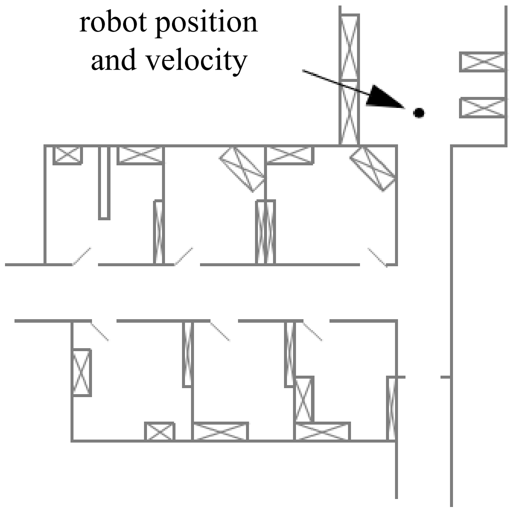
- Measurement model:

$$z_k = x_k + w_k$$

$$F_k = H_k = 1, \qquad B_k = \Delta t, \qquad Q_k = \sigma_u^2, \qquad R_k = \sigma_z^2$$

Example 2:

- Robot State: 2D position and velocity
  - $x(t) = (\xi(t), v(t)) \in \mathbb{R}^4$
- Control input: PC acceleration sensor at 1 Hz
  - $u(t) = a(t) + \eta(t), \qquad \eta(t) \sim N(0, Q)$
- Observation input: EC position sensor at 1/Ts Hz
  - $z(t) = \xi(t) + w(t), \qquad w(t) \sim N(0, R)$
- Kinematic model:
  - $\dot{x}(t) = \begin{pmatrix} \dot{\xi}(t) \\ \dot{v}(t) \end{pmatrix} = \begin{pmatrix} v(t) \\ a(t) \end{pmatrix}$



Example 2:

- Kinematic model:
  - $\dot{x}(t) = \begin{pmatrix} \dot{\xi}(t) \\ \dot{v}(t) \end{pmatrix} = \begin{pmatrix} v(t) \\ a(t) \end{pmatrix}$

Discretization:

$$\dot{x}_k \approx \frac{x_k - x_{k-1}}{\Delta t}$$

- State Transition model:

$$x_k = F_k x_{k-1} + \mathrm{B}_k\, u_k + \eta_k$$

- Measurement model:

$$z_k = H_k x_k + w_k$$

$$F_k = \begin{pmatrix} I_2 & \Delta t\, I_2 \\ 0_{2\times 2} & I_2 \end{pmatrix}, \qquad H_k = (I_2 \quad 0_{2\times 2}), \qquad B_k = \begin{pmatrix} (\Delta t)^2 I_2 \\ \Delta t\, I_2 \end{pmatrix},$$

$$Q_k = \mathrm{diag}\,(\sigma_{u1}^2, \sigma_{u2}^2), \qquad R_k = \mathrm{diag}\,(\sigma_{z1}^2, \sigma_{z2}^2)$$

Outline

- Recap last lecture
- Examples
- Kalman Filter Algorithm
- Application: 1D robot position estimation

Kalman Filter Overview

Gaussian

$\boldsymbol{N}\left(\boldsymbol{\hat{x}_{k-1|\,k-1}}, \boldsymbol{P_{k-1|\,k-1}}\right)$

$\boldsymbol{bel}\ (\boldsymbol{x_{k-1}})$
$= p(x_{k-1}|Z_{k-1}, U_{k-1})$

$\boldsymbol{bel}\ (\boldsymbol{x_k})$
$= p(x_k|Z_k, U_k)$

Gaussian

$\boldsymbol{N}\left(\boldsymbol{\hat{x}_{k|k}}, \boldsymbol{P_{k|\,k}}\right)$

u_k

$v_k \sim N(0, Q_k)$

State Transition Model
$x_k = F_k\, x_{k-1} + B_k\, u_k + v_k$

1 **Prediction update**

Gaussian

$\boldsymbol{N}\left(\boldsymbol{\hat{x}_{k|\,k-1}}, \boldsymbol{P_{k|\,k-1}}\right)$

$\overline{\boldsymbol{bel}}\ (\boldsymbol{x_k})$
$= p(x_k|Z_{k-1}, U_k)$

Correction update 2

Sensor Model
$\beta_k \coloneqq z_k - H_k x_k + w_k$

$w_k \sim N(0, R_k)$

z_k

Recall
$\boldsymbol{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$: Gaussian with mean $\boldsymbol{\mu} \in \mathbb{R}^{\boldsymbol{k}}$ and covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{\boldsymbol{k \times k}}$

Kalman Filter Overview

1 **Prediction update**

Compute $\hat{x}_{k|\,k-1}$ and $P_{k|\,k-1}$

- Given $\hat{x}_{k-1|\,k-1}, P_{k-1|\,k-1}, u_k$
- Using F_k, B_k, Q_k

u_k

$v_k \sim N(0, Q_k)$

State Transition Model

$x_k = F_k\, x_{k-1} + B_k\, u_k + v_k$

1 **Prediction update**

$\boldsymbol{bel}\,(\boldsymbol{x_{k-1}})$
$= N\left(\hat{x}_{k-1|\,k-1}, P_{k-1|\,k-1}\right)$

$\overline{\boldsymbol{bel}}\,(\boldsymbol{x_k})$
$= N\left(\hat{x}_{k|\,k-1}, P_{k|\,k-1}\right)$

$\boldsymbol{bel}\,(\boldsymbol{x_k})$
$= N\left(\hat{x}_{k|\,k}, P_{k|\,k}\right)$

Correction update 2

Sensor Model

$\beta_k \coloneqq z_k - H_k x_k + w_k$

$w_k \sim N(0, R_k)$

z_k

Kalman Filter Overview

u_k

$v_k \sim N(0, Q_k)$

State Transition Model

$$x_k = F_k\, x_{k-1} + B_k\, u_k + v_k$$

1 **Prediction update**

$\overline{\boldsymbol{bel}}\,(\boldsymbol{x_k}) = N\left(\hat{x}_{k|\,k-1}, P_{k|\,k-1}\right)$

Correction update 2

Sensor Model

$$\beta_k \coloneqq z_k - H_k x_k + w_k$$

$w_k \sim N(0, R_k)$

z_k

$\boldsymbol{bel}\,(\boldsymbol{x_{k-1}}) = N\left(\hat{x}_{k-1|\,k-1}, P_{k-1|\,k-1}\right)$

$\boldsymbol{bel}\,(\boldsymbol{x_k}) = N\left(\hat{x}_{k|\,k}, P_{k|\,k}\right)$

1 **Prediction update**

Compute $\hat{x}_{k|\,k-1}$ and $P_{k|\,k-1}$

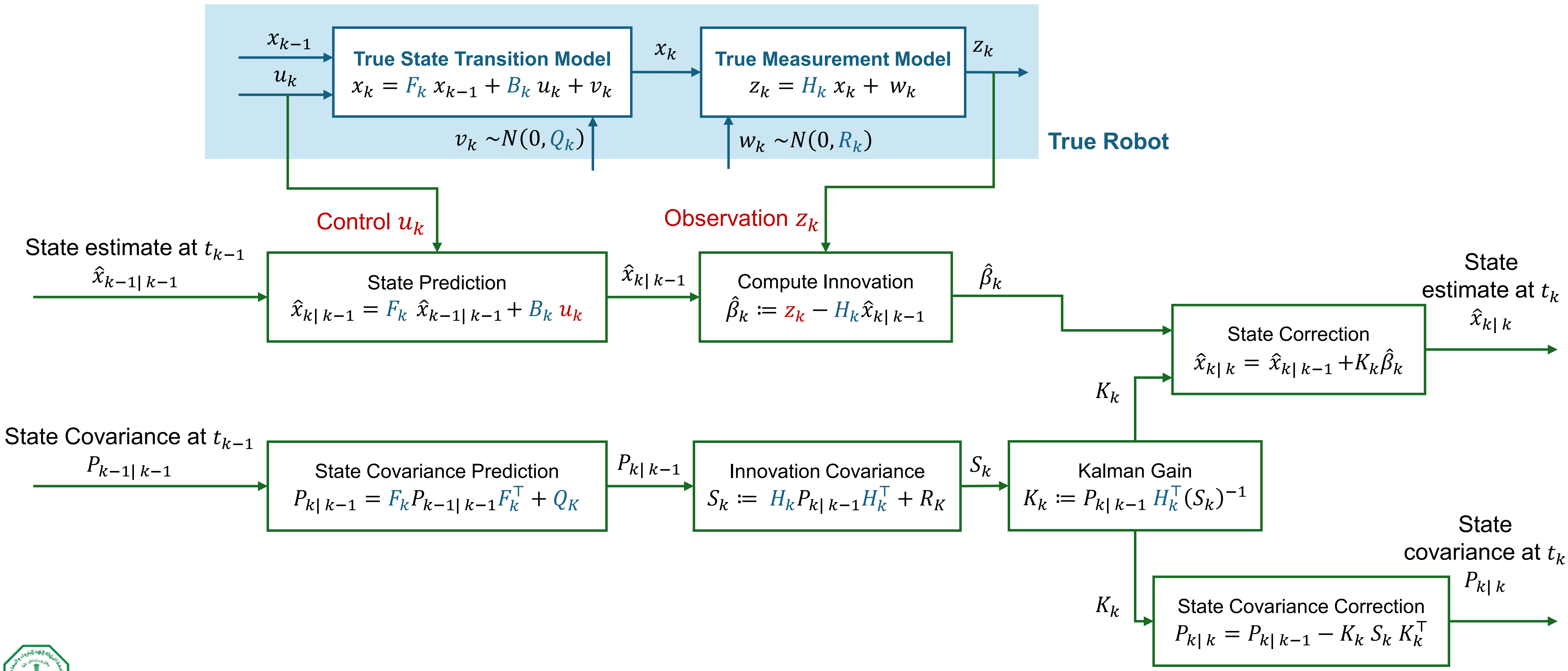
- Given $\hat{x}_{k-1|\,k-1}, P_{k-1|\,k-1}, u_k$
- Using F_k, B_k, Q_k

2 **Correction update**

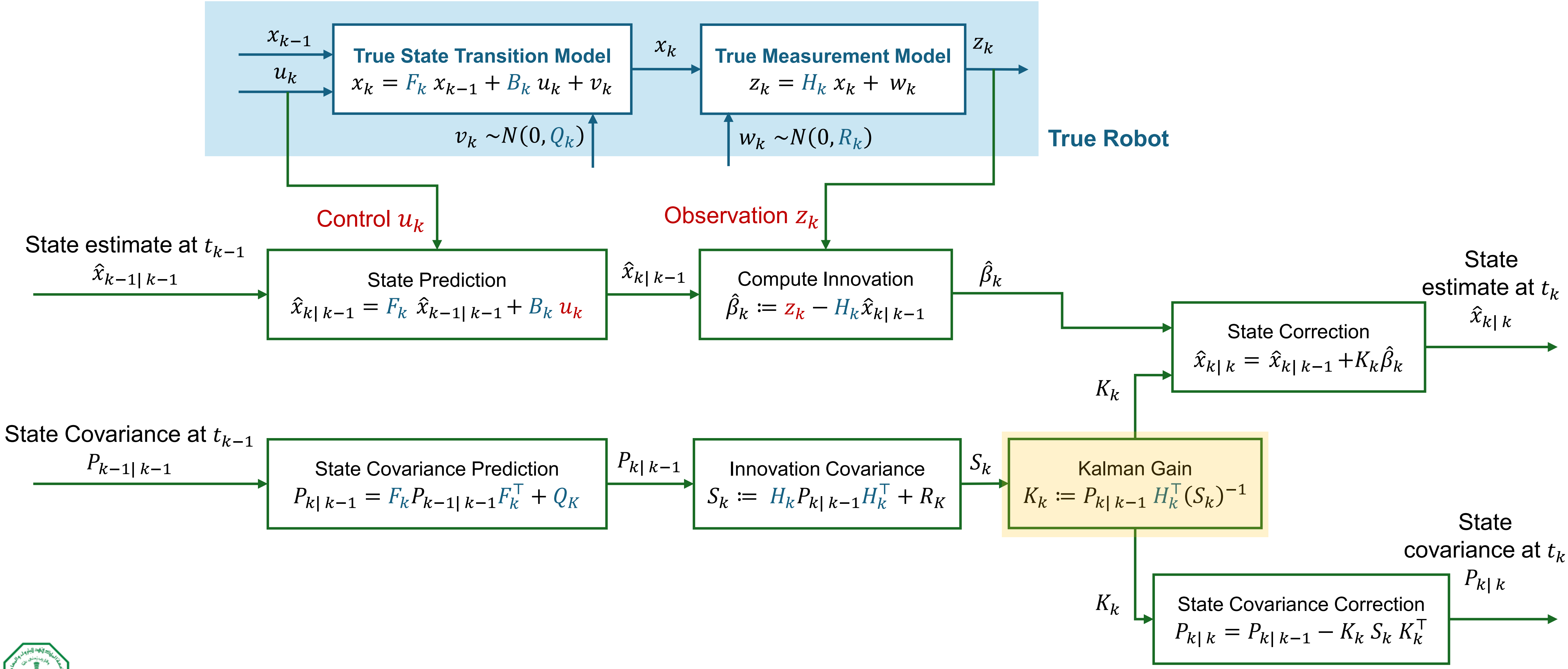
Compute $\hat{x}_{k|\,k}$ and $P_{k|\,k}$

- Given $\hat{x}_{k|\,k-1}, P_{k|\,k-1}, z_k$
- Using H_k, R_k

Kalman Filter Block Diagram



Kalman Filter Block Diagram



Kalman Filter Overview

① Prediction update

- $\hat{x}_{k|\,k-1} = F_k\ \hat{x}_{k-1|\,k-1} + B_k\, u_k$
- $P_{k|\,k-1} = F_k P_{k-1|\,k-1} F_k^{\top} + Q_K$

u_k

$v_k \sim N(0, Q_k)$

State Transition Model

$x_k = F_k\ x_{k-1} + B_k\, u_k + v_k$

① Prediction update

$\boldsymbol{bel}\,(\boldsymbol{x_{k-1}}) = N\left(\hat{x}_{k-1|\,k-1}, P_{k-1|\,k-1}\right)$

$\overline{\boldsymbol{bel}}\,(\boldsymbol{x_k}) = N\left(\hat{x}_{k|\,k-1}, P_{k|\,k-1}\right)$

$\boldsymbol{bel}\,(\boldsymbol{x_k}) = N\left(\hat{x}_{k|\,k}, P_{k|\,k}\right)$

Correction update ②

Sensor Model

$\beta_k \coloneqq z_k - H_k x_k + w_k$

$w_k \sim N(0, R_k)$

z_k

② Correction update

- $\hat{\beta}_k \coloneqq z_k - H_k \hat{x}_{k|\,k-1}$
- $S_k \coloneqq H_k P_{k|\,k-1} H_k^{\top} + R_K$
- $K_k \coloneqq P_{k|\,k-1}\ H_k^{\top} (S_k)^{-1}$
- $\hat{x}_{k|\,k} = \hat{x}_{k|\,k-1} + K_k \hat{\beta}_k$
- $P_{k|\,k} = P_{k|\,k-1} - K_k\ S_k\ K_k^{\top}$

Outline

- Recap last lecture
- Examples
- Kalman Filter Algorithm
- Application: 1D robot position estimation

1D robot position estimation