# CIE 482: Path Planning & Navigation for Mobile Robots

Lecture 18: Kalman Filtering III

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

Ramy Rashad
Assistant Professor
CIE Department

# Outline

- Recap last lecture
- Kalman Filter Performance
- Application: 2D robot position estimation
- Linearization of Nonlinear Models

# Outline

- Recap last lecture
- Kalman Filter Performance
- Application: 2D robot position estimation
- Linearization of Nonlinear Models

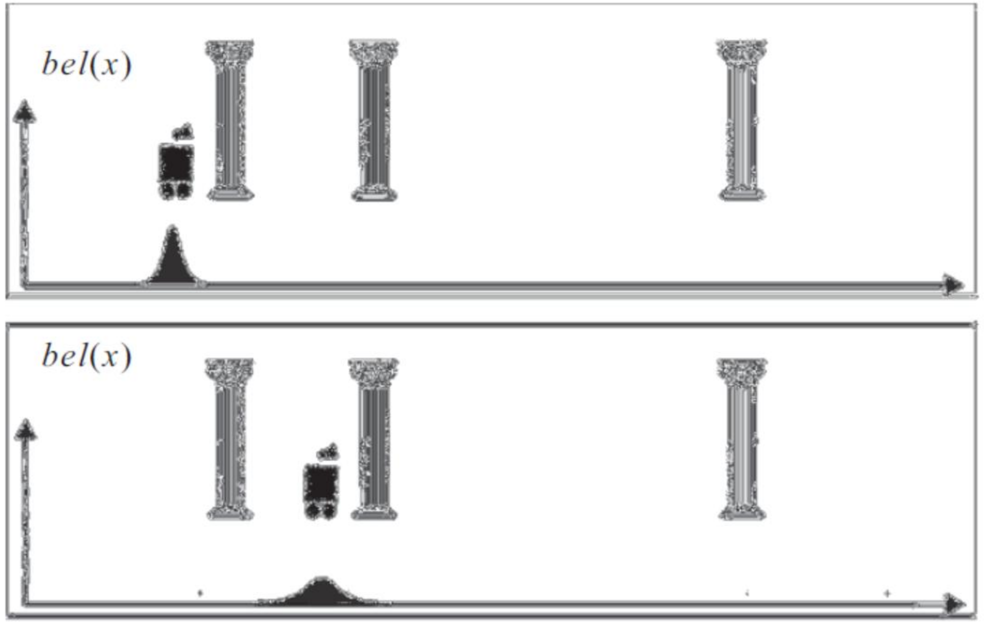
# Recap: Belief Representation

- 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: Kalman Filter

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

$u_k$

Linear

**State Transition Model**

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

1 Prediction update

Gaussian

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

Correction update 2

**Sensor Model**

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

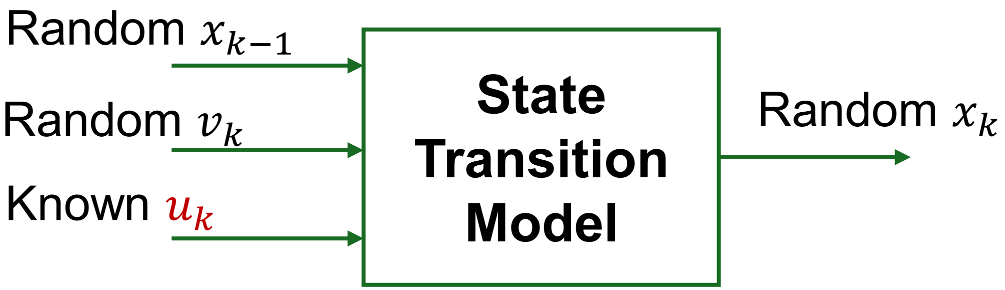
Linear

$z_k$

# Recap: 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.

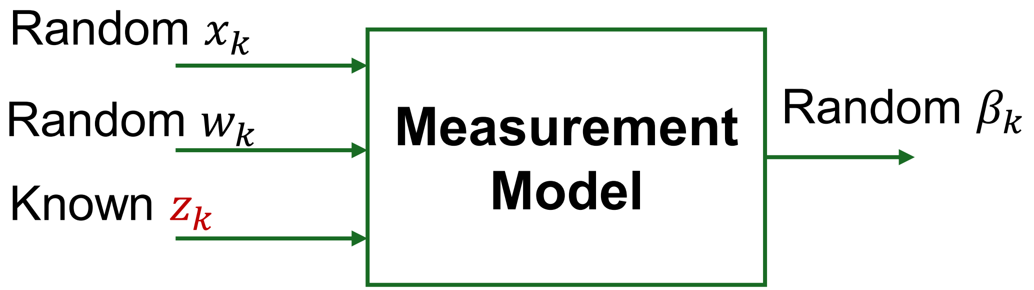



**Note** that $u_k$ is not a random variable but a known input.

# Recap: Measurement Probabilistic Model

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

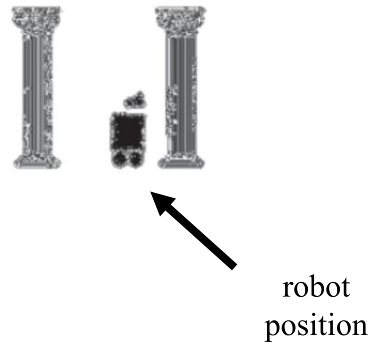
- If $x_k$ and $w_k$ are two random variables that are independent and have a Gaussian distribution, then $\beta_k$ will also have a Gaussian distribution.

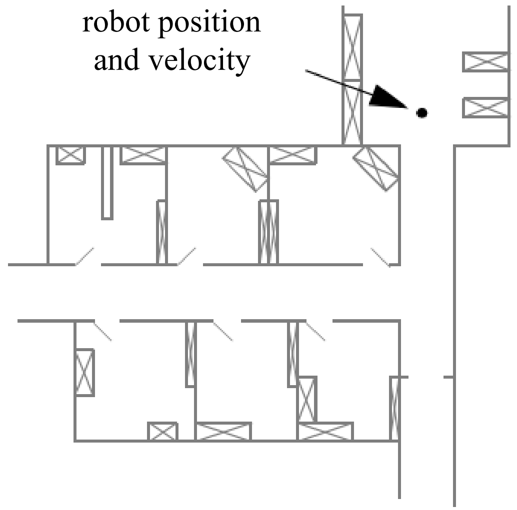



**Note** that $z_k$ is not a random variable but a known input.

# 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 $\mathrm{z}(t) \in \mathbb{R}^2$



# 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**

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

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

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

**Correction update** 2

**Sensor Model**

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

$w_k \sim N(0, R_k)$

$z_k$

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 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**

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

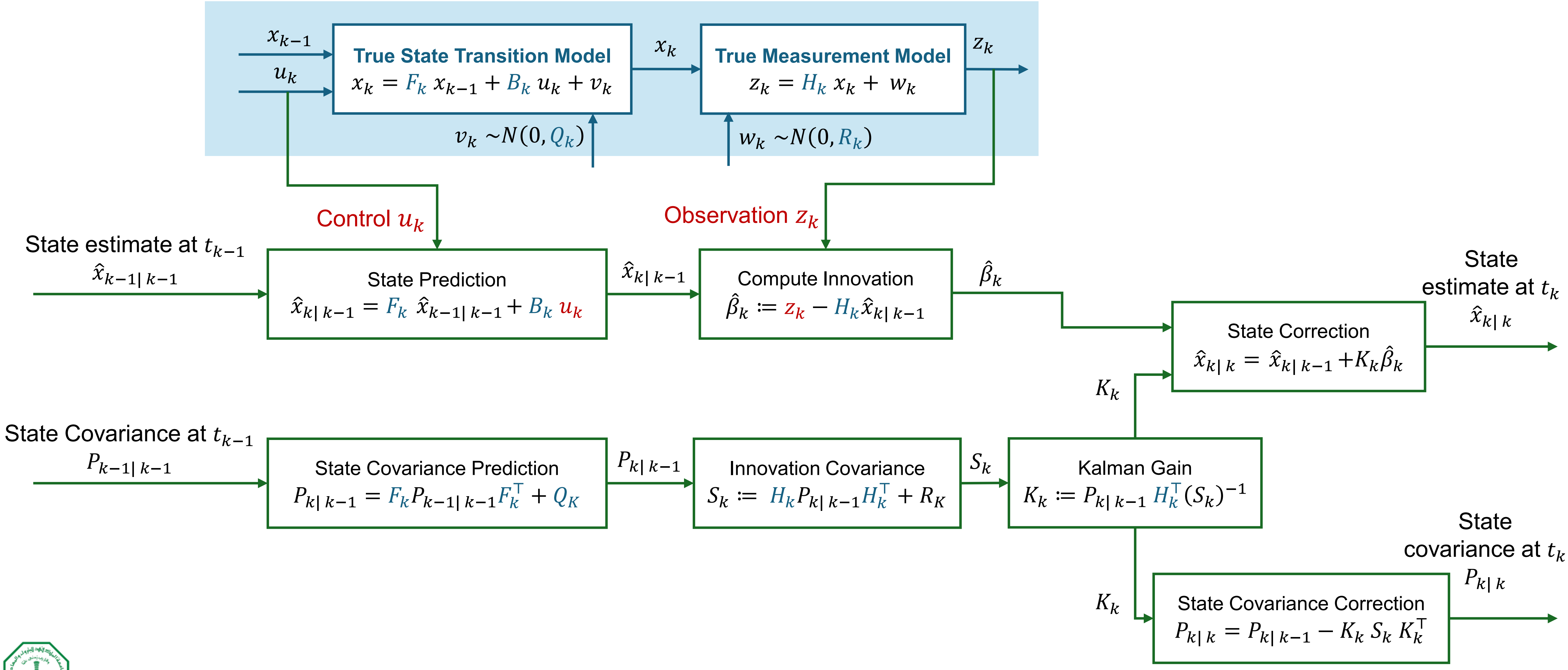
$w_k \sim N(0, R_k)$

$z_k$

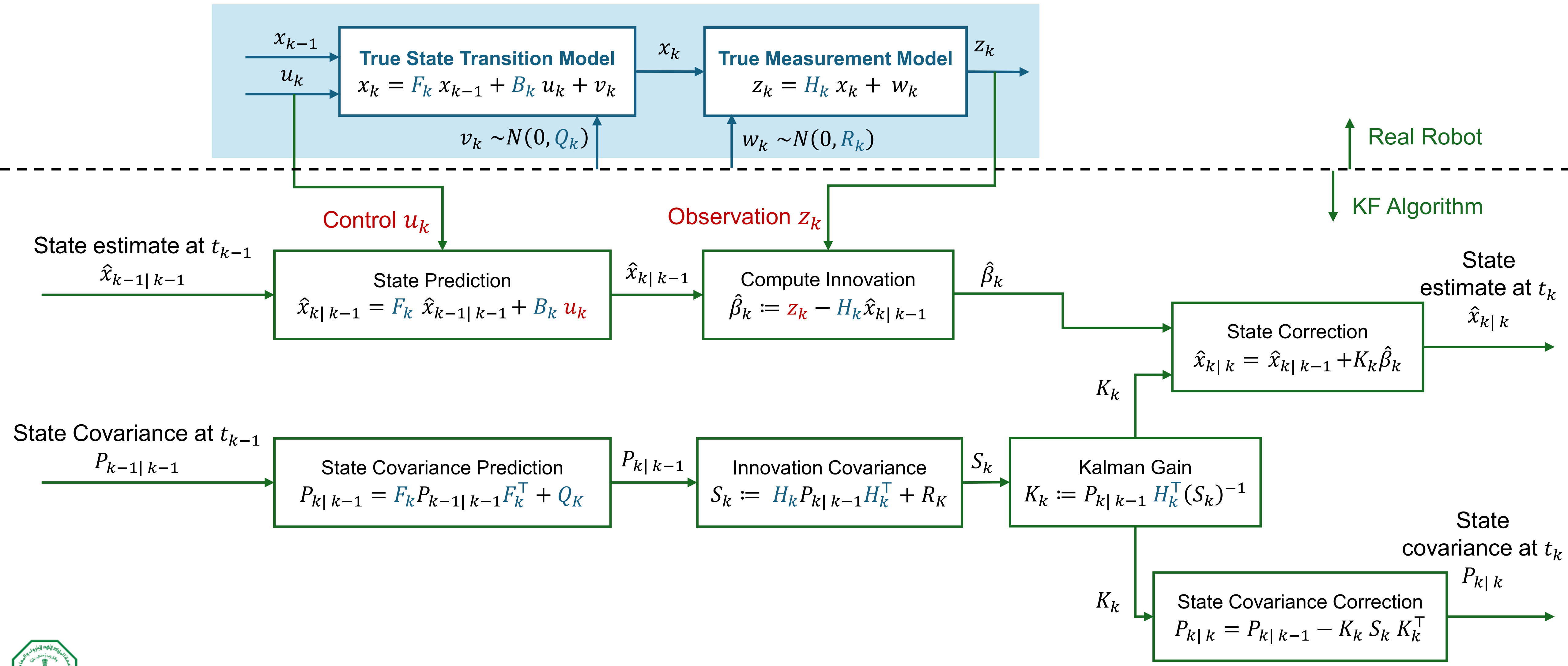
2 **Correction update**

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

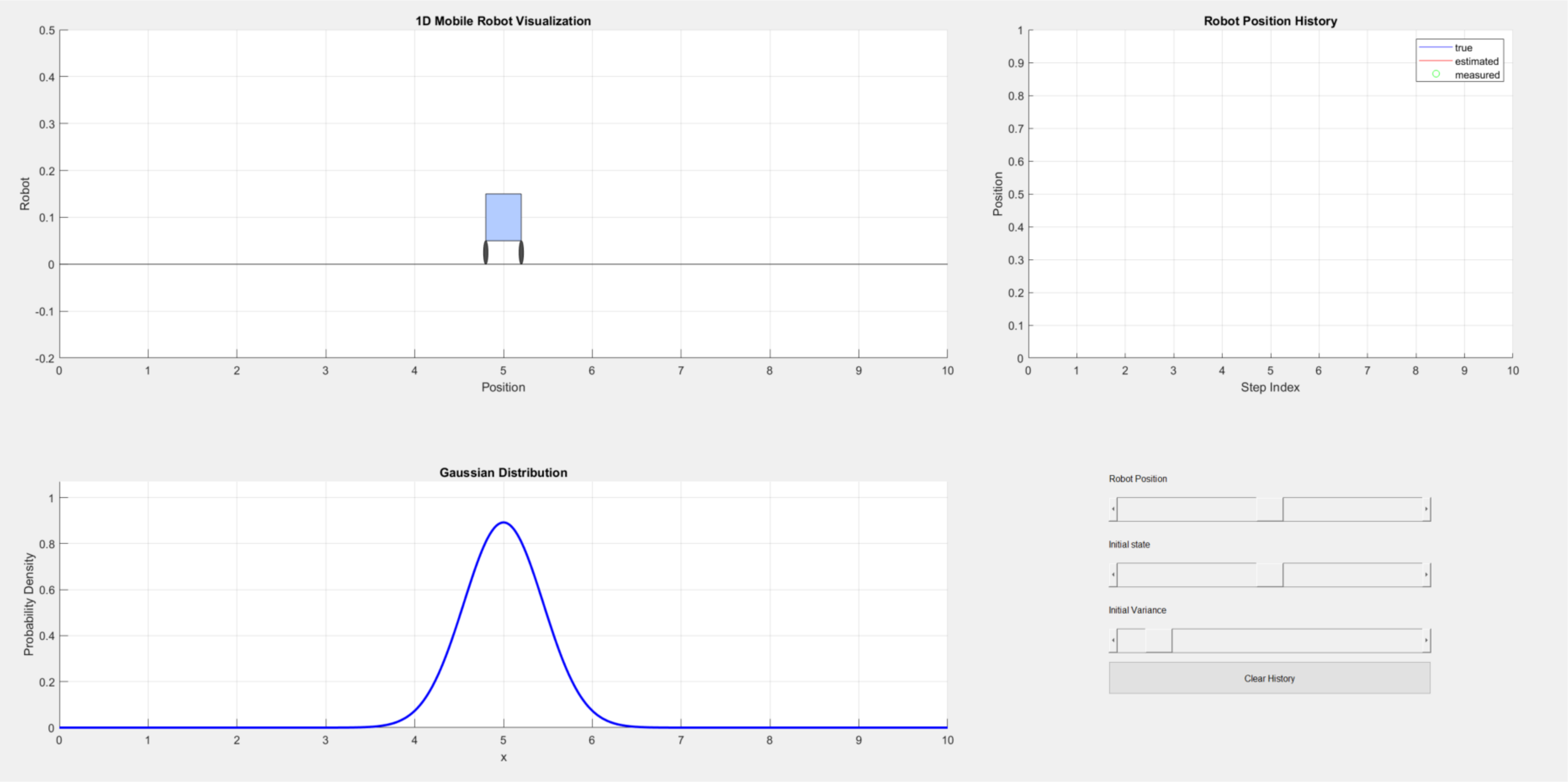
# Kalman Filter Block Diagram



# Kalman Filter Block Diagram



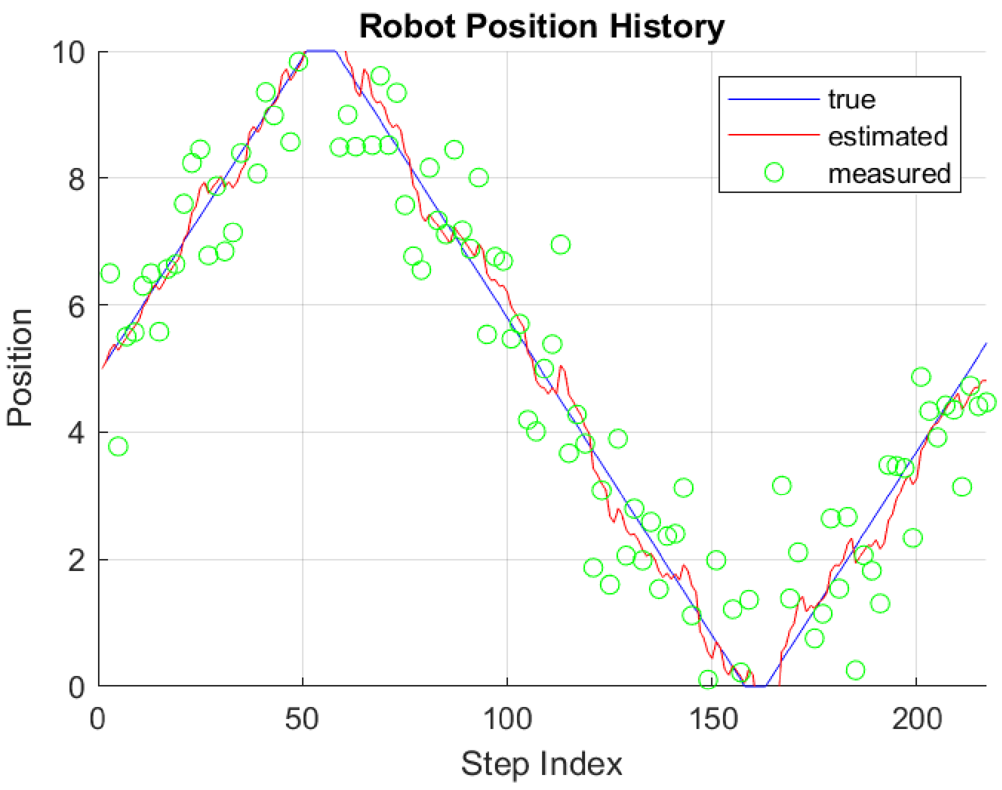
# 1D robot position estimation



# Outline

- Recap last lecture
- Kalman Filter Performance
- Application: 2D robot position estimation
- Linearization of Nonlinear Models

# Kalman Filter Performance

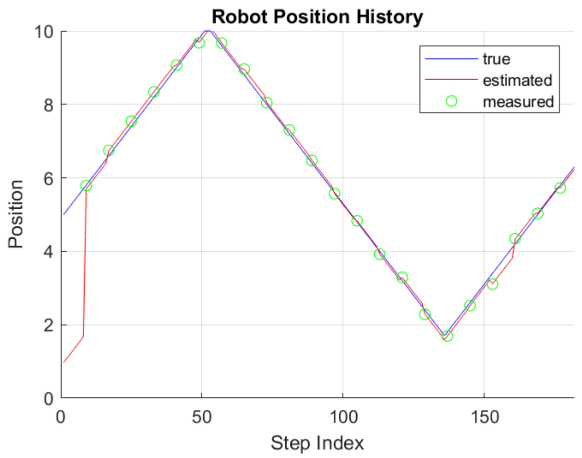




**Experiment a:** Initial Error in mean $\hat{x}_{0|0}$
Measurement frequency 1/8 Hz
Measurement variance $R_k = \sigma_z^2 = 0.01$
Control variance $Q_k = \sigma_u^2 = 0.03$

**Experiment b:** High measurement uncertainty
Measurement frequency 1/2 Hz
Measurement variance $R_k = \sigma_z^2 = 0.9$
Control variance $Q_k = \sigma_u^2 = 0.03$

# Observations

- Robustness to Initial Estimate Errors:
  - One of the key advantages of the Kalman filter is its ability to recover from poor initial guesses for the system state.
  - Even if the initial state estimate is far off, the measurement updates gradually correct the estimate.
  - This correction happens because the Kalman gain adjusts over time, increasing the weight on measurements when uncertainty in the estimate is high.

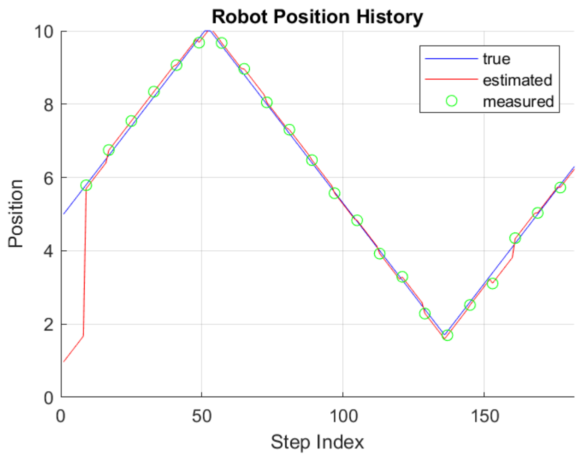



**Experiment a**

# Observations

- Effect of $Q_k$ and $R_k$:
  - $R_k$ represents how much you *trust* the measurements.
  - $Q_k$ represents how much you *trust the* motion model.
  - Case $R_k < Q_k$:
    - Filter gives more weight to measurements.
    - Estimates will follow measurements closely.

$R_k < Q_k$




**Experiment a**

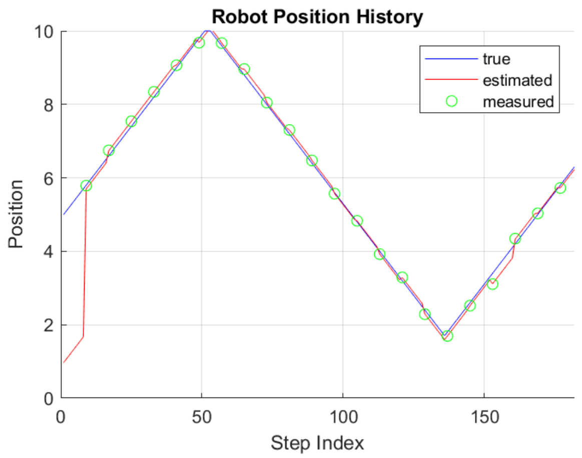
$R_k > Q_k$


**Experiment b**

# Observations

- Effect of $Q_k$ and $R_k$:
  - $R_k$ represents how much you *trust* the measurements.
  - $Q_k$ represents how much you *trust the* motion model.
  - Case $R_k > Q_k$:
    - Filter trusts the model more than the measurements.
    - Estimates become smoother and less noisy

$R_k < Q_k$




**Experiment a**

$R_k > Q_k$


**Experiment b**

# Outline

- Recap last lecture
- Kalman Filter Performance
- Application: 2D robot position estimation
- Linearization of Nonlinear Models

# 2D robot position estimation

- Robot State: 2D position and velocity
  - $x(t) = (\xi(t), v(t)) \in \mathbb{R}^4$
- Control input: PC acceleration sensor at 1 Hz
  - $u(t) = a(t) + \eta(t), \qquad \eta(t) \sim N(0, Q)$

$Q = \mathrm{diag}\,(\sigma_{u1}^2, \sigma_{u2}^2)$

- Observation input: EC position sensor at 1/Ts Hz
  - $z(t) = \xi(t) + w(t), \qquad w(t) \sim N(0, R)$

$R = \mathrm{diag}\,(\sigma_{z1}^2, \sigma_{z2}^2)$

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

**PC**: proprioceptive
**EC**: exteroceptive

# 2D robot position estimation

- 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:

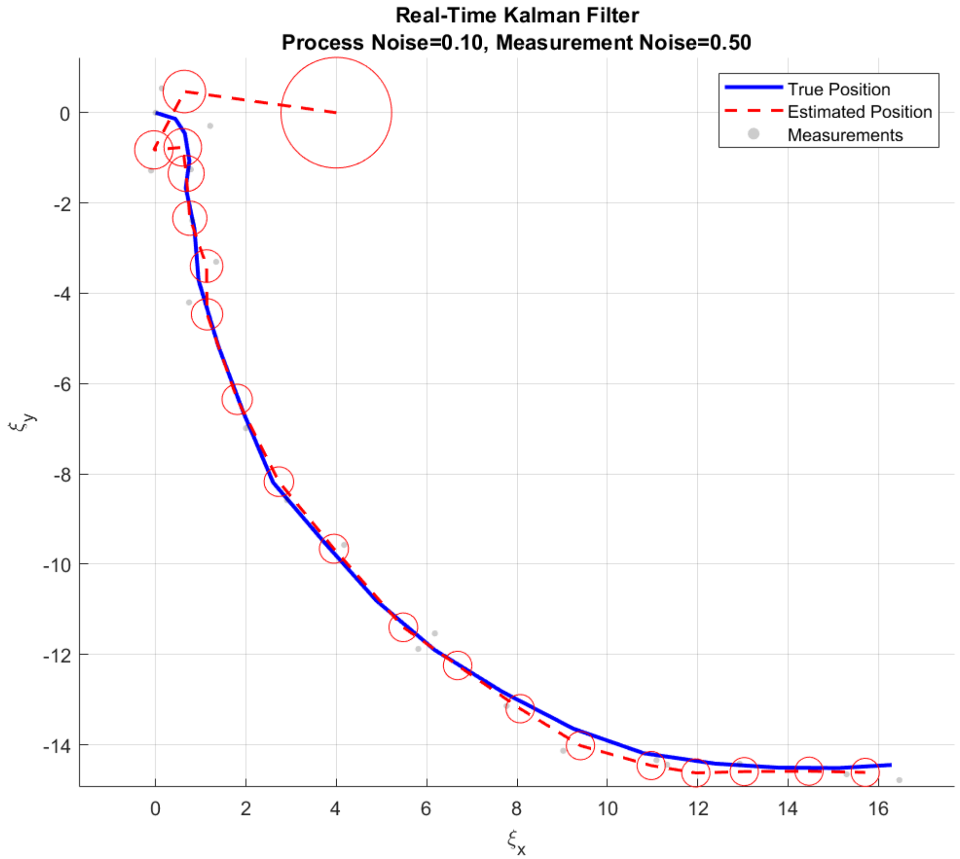
$$z_k = H_k x_k + w_k$$

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

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

# Results of Kalman Filter

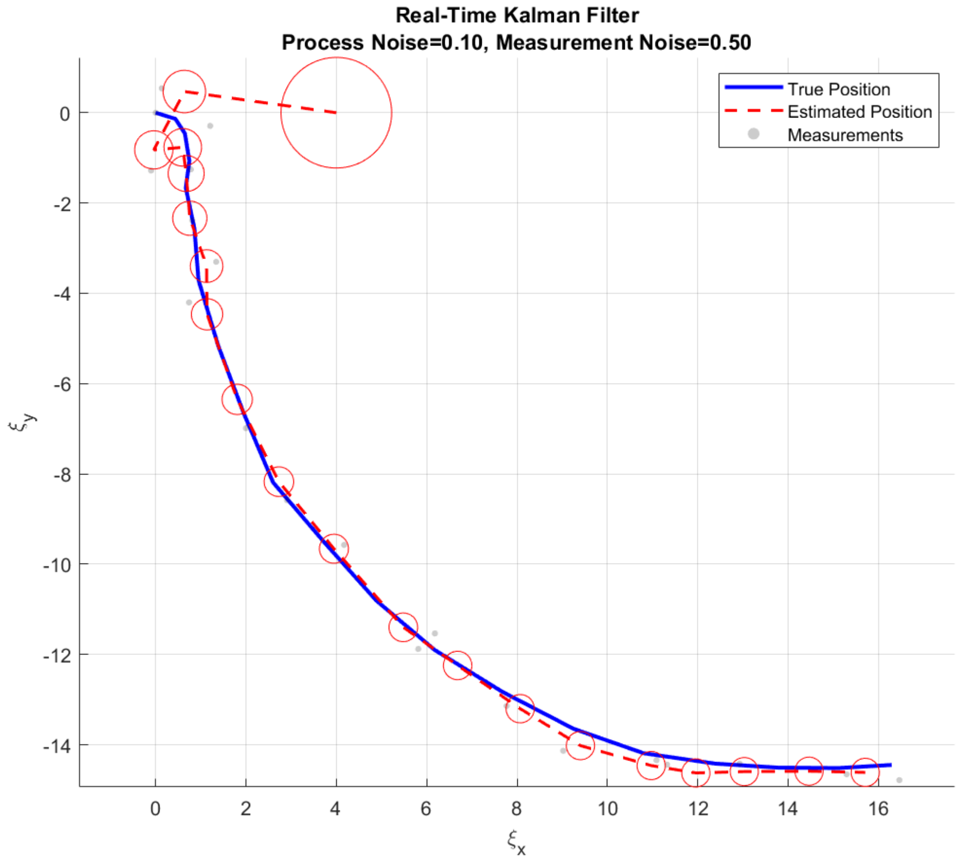
- The **uncertainty ellipses** illustrate the **covariance** around the filter's estimated position $\hat{x}_{k|k} = \begin{pmatrix} \hat{\xi}_{k|k} \\ \hat{v}_{k|k} \end{pmatrix}$ at each timestep.



# Results of Kalman Filter

- In two dimensions, the **covariance matrix** for the position state can be visualized as an ellipse whose shape and orientation show how uncertain or "spread out" the estimate is.

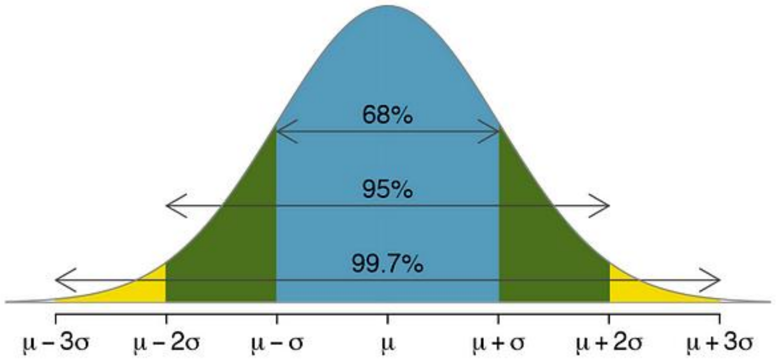
1. A **larger ellipse** means the filter is less certain about the position, e.g., if measurement noise is high or few measurements have arrived.
2. A **smaller ellipse** means the filter has converged to a more precise belief about the current position.



# Confidence Intervals

- 1D Case (Single State Variable):
  - 68.3% CI -> $\hat{x}_{k|k} \ \pm 1\sigma$
  - 95.4% CI -> $\hat{x}_{k|k} \ \pm 2\sigma$
  - 99.7% CI -> $\hat{x}_{k|k} \ \pm 3\sigma$

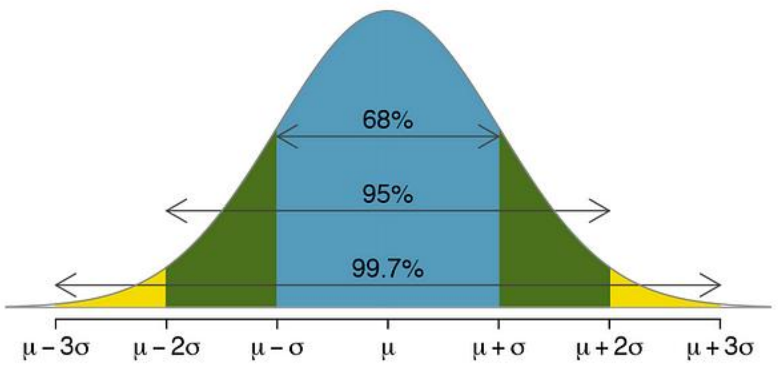
  where $\sigma = \sqrt{P_{k|k}}$ , the standard deviation.



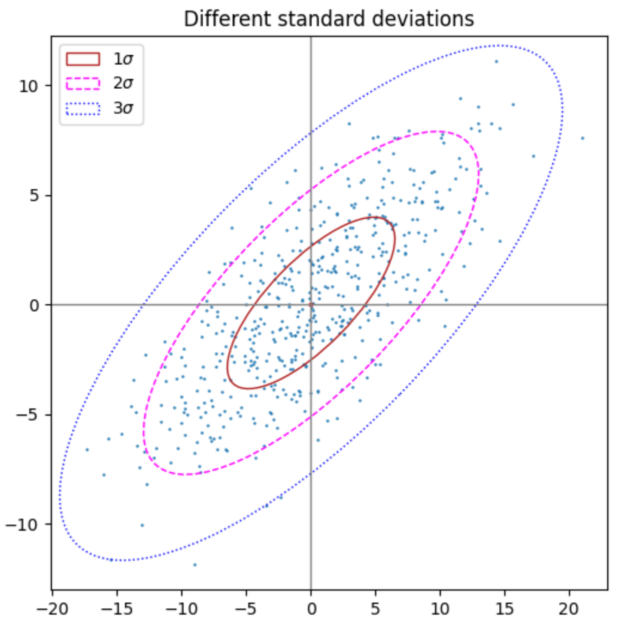
# Confidence Intervals

- 1D Case (Single State Variable):
  - 68.3% CI -> $\hat{x}_{k|k} \pm 1\sigma$
  - 95.4% CI -> $\hat{x}_{k|k} \pm 2\sigma$
  - 99.7% CI -> $\hat{x}_{k|k} \pm 3\sigma$

  where $\sigma = \sqrt{P_{k|k}}$ , the standard deviation.




- 2D Case (Two State Variables):
  - The state estimate $\hat{x}_{k|k}$ has a confidence ellipse, not an interval.
  - The ellipse is defined by the covariance matrix $P_{k|k}$.
  - A $2\sigma$ ellipsoid covers ~95% probability if the errors are Gaussian.



# Outline

- Recap last lecture
- Kalman Filter Performance
- Application: 2D robot position estimation
- Linearization of Nonlinear Models

# Drawbacks of standard Kalman filter

- KF works well when dynamics and sensors are actually linear.
- Problem:
  - Real-world systems (like mobile robots) often have nonlinear dynamics or nonlinear sensors.
- One of the possible extensions of the Kalman filter to nonlinear systems in the Extended Kalman Filter (EKF).
- The idea behind it is that it linearizes the nonlinear models around the latest estimate $\hat{x}_{k-1|k-1}$.

# Recall: Taylor series expansion

- Given a smooth function $f: \mathbb{R}^n \to \mathbb{R}$, the **Taylor series expansion of order $k$** around some given vector $x_0 \in \mathbb{R}^n$ expresses $f(x)$ as a sum of $k$ derivatives evaluated at $x_0$.

- Case $k = 1$:

$$f(x) \approx f(x_0) + df(x_0)(x - x_0),$$

where $df(x) = \left(\frac{\partial f}{\partial x_1}(x), \cdots, \frac{\partial f}{\partial x_n}(x)\right) \in \mathbb{R}^{1\times n}$ is called the gradient of the function $f$.

# Recall: Taylor series expansion

- Now suppose $f: \mathbb{R}^n \to \mathbb{R}^n$, the **Taylor series expansion** of order 1 around a vector $x_0 \in \mathbb{R}^n$ is given by

$$f(x) \approx f(x_0) + J_f(x_0)(x - x_0),$$

where $J_f(x_0) \in \mathbb{R}^{n\times n}$ is called the Jacobian matrix given by

$$J_f(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x) & \cdots & \frac{\partial f_1}{\partial x_n}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1}(x) & \cdots & \frac{\partial f_n}{\partial x_n}(x) \end{pmatrix}$$

$J_f(x)$ is usually denoted by $\frac{\partial f}{\partial x}(x)$

# Recall: Taylor series expansion

- Note that this allows us to approximate the function $f(x)$ near the vector $x_0 \in \mathbb{R}^n$ by the linear set of equations:

$$f(x) \approx \underset{\substack{\rotatebox{90}{$\in$} \\ \mathbb{R}^n}}{f(x_0)} + \underset{\substack{\rotatebox{90}{$\in$} \\ \mathbb{R}^{n\times n}}}{J_f(x_0)} \underset{\substack{\rotatebox{90}{$\in$} \\ \mathbb{R}^n}}{(x - x_0)},$$

$$\boxed{f(x) \approx \begin{pmatrix} f_1(x_0) \\ \vdots \\ f_n(x_0) \end{pmatrix} + \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x_0) & \cdots & \frac{\partial f_1}{\partial x_n}(x_0) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1}(x_0) & \cdots & \frac{\partial f_n}{\partial x_n}(x_0) \end{pmatrix} \begin{pmatrix} x_1 & - x_{0,1} \\ & \vdots \\ x_n & - x_{n,1} \end{pmatrix}}$$

# Nonlinear State Transition Model

- The actual state transition model is nonlinear of the form

$$x_k = f(x_{k-1}, u_k, v_k) \qquad f: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}^n$$

- Using Taylor series expansion, we can approximate it as the linear system

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

where $F_k \in \mathbb{R}^{n\times n}$ is the Jacobian of $f$ with respect to $x_{k-1}$ evaluated at the latest estimate $\hat{x}_{k-1\,|\,k-1}$

$$F_k \coloneqq \frac{\partial f}{\partial x_{k-1}}\left(\hat{x}_{k-1\,|\,k-1}\right) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}\left(\hat{x}_{k-1\,|\,k-1}\right) & \cdots & \frac{\partial f_1}{\partial x_n}\left(\hat{x}_{k-1\,|\,k-1}\right) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1}\left(\hat{x}_{k-1\,|\,k-1}\right) & \cdots & \frac{\partial f_n}{\partial x_n}\left(\hat{x}_{k-1\,|\,k-1}\right) \end{pmatrix} \in \mathbb{R}^{n\times n}$$

# Nonlinear State Transition Model

- The actual state transition model is nonlinear of the form

$$x_k = f(x_{k-1}, u_k, v_k) \qquad\qquad f: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}^n$$

- Using Taylor series expansion, we can approximate it as the linear system

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

where $B_k \in \mathbb{R}^{n\times m}$ is the Jacobian of $f$ with respect to $u_k$ evaluated at the latest estimate $\hat{x}_{k-1\,|\,k-1}$

$$B_k \coloneqq \frac{\partial f}{\partial u_k}\left(\hat{x}_{k-1\,|\,k-1}\right) = \begin{pmatrix} \frac{\partial f_1}{\partial u_1}\left(\hat{x}_{k-1\,|\,k-1}\right) & \cdots & \frac{\partial f_1}{\partial u_m}\left(\hat{x}_{k-1\,|\,k-1}\right) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1}\left(\hat{x}_{k-1\,|\,k-1}\right) & \cdots & \frac{\partial f_n}{\partial u_m}\left(\hat{x}_{k-1\,|\,k-1}\right) \end{pmatrix} \in \mathbb{R}^{n\times m}$$

**Note: we will not use the matrix $B_k$ at all !**

# Nonlinear State Transition Model

- The actual state transition model is nonlinear of the form

$$x_k = f(x_{k-1}, u_k, v_k) \qquad f: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}^n$$

- Using Taylor series expansion, we can approximate it as the linear system

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

where $G_k \in \mathbb{R}^{n\times p}$ is the Jacobian of $f$ with respect to $v_k$ evaluated at the latest estimate $\hat{x}_{k-1\,|\,k-1}$

$$G_k := \frac{\partial f}{\partial v_k}(\hat{x}_{k-1\,|\,k-1}) = \begin{pmatrix} \frac{\partial f_1}{\partial v_1}(\hat{x}_{k-1\,|\,k-1}) & \cdots & \frac{\partial f_1}{\partial v_p}(\hat{x}_{k-1\,|\,k-1}) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial v_1}(\hat{x}_{k-1\,|\,k-1}) & \cdots & \frac{\partial f_n}{\partial v_p}(\hat{x}_{k-1\,|\,k-1}) \end{pmatrix} \in \mathbb{R}^{n\times p}$$

# Nonlinear Measurement Model

- The actual measurement model is nonlinear of the form

$$z_{\mathrm{k}} = h(x_{\mathrm{k}}, w_{\mathrm{k}}) \qquad h: \mathbb{R}^n \times \mathbb{R}^l \to \mathbb{R}^q$$

- Using Taylor series expansion, we can approximate it as the linear system

$$z_{\mathrm{k}} = H_k x_{\mathrm{k}} + L_k w_{\mathrm{k}}$$

where $H_k \in \mathbb{R}^{q\times n}$ is the Jacobians of $h$ with respect to $x_{\mathrm{k}}$ evaluated at the prediction $\hat{x}_{k\,|\,k-1}$

$$H_k \coloneqq \frac{\partial h}{\partial x_k}(\hat{x}_{k\,|\,k-1}) = \begin{pmatrix} \frac{\partial h_1}{\partial x_1}(\hat{x}_{k\,|\,k-1}) & \cdots & \frac{\partial h_1}{\partial x_n}(\hat{x}_{k\,|\,k-1}) \\ \vdots & \ddots & \vdots \\ \frac{\partial h_q}{\partial x_1}(\hat{x}_{k\,|\,k-1}) & \cdots & \frac{\partial h_q}{\partial x_n}(\hat{x}_{k\,|\,k-1}) \end{pmatrix} \in \mathbb{R}^{q\times n}$$

# Nonlinear Measurement Model

- The actual measurement model is nonlinear of the form

$$z_k = h(x_k, w_k) \qquad h: \mathbb{R}^n \times \mathbb{R}^l \to \mathbb{R}^q$$

- Using Taylor series expansion, we can approximate it as the linear system

$$z_k = H_k x_k + L_k w_k$$

where $L_k \in \mathbb{R}^{q\times l}$ is the Jacobian of $h$ with respect to $w_k$ evaluated at the prediction $\hat{x}_{k\,|\,k-1}$

$$L_k := \frac{\partial h}{\partial w_k}(\hat{x}_{k\,|\,k-1}) = \begin{pmatrix} \frac{\partial h_1}{\partial w_1}(\hat{x}_{k\,|\,k-1}) & \cdots & \frac{\partial h_1}{\partial w_l}(\hat{x}_{k\,|\,k-1}) \\ \vdots & \ddots & \vdots \\ \frac{\partial h_q}{\partial w_1}(\hat{x}_{k\,|\,k-1}) & \cdots & \frac{\partial h_q}{\partial w_l}(\hat{x}_{k\,|\,k-1}) \end{pmatrix} \in \mathbb{R}^{q\times l}$$