CIE 482: Path Planning & Navigation for Mobile Robots

Lecture 19: Extended Kalman Filter

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

Ramy Rashad
Assistant Professor
CIE Department

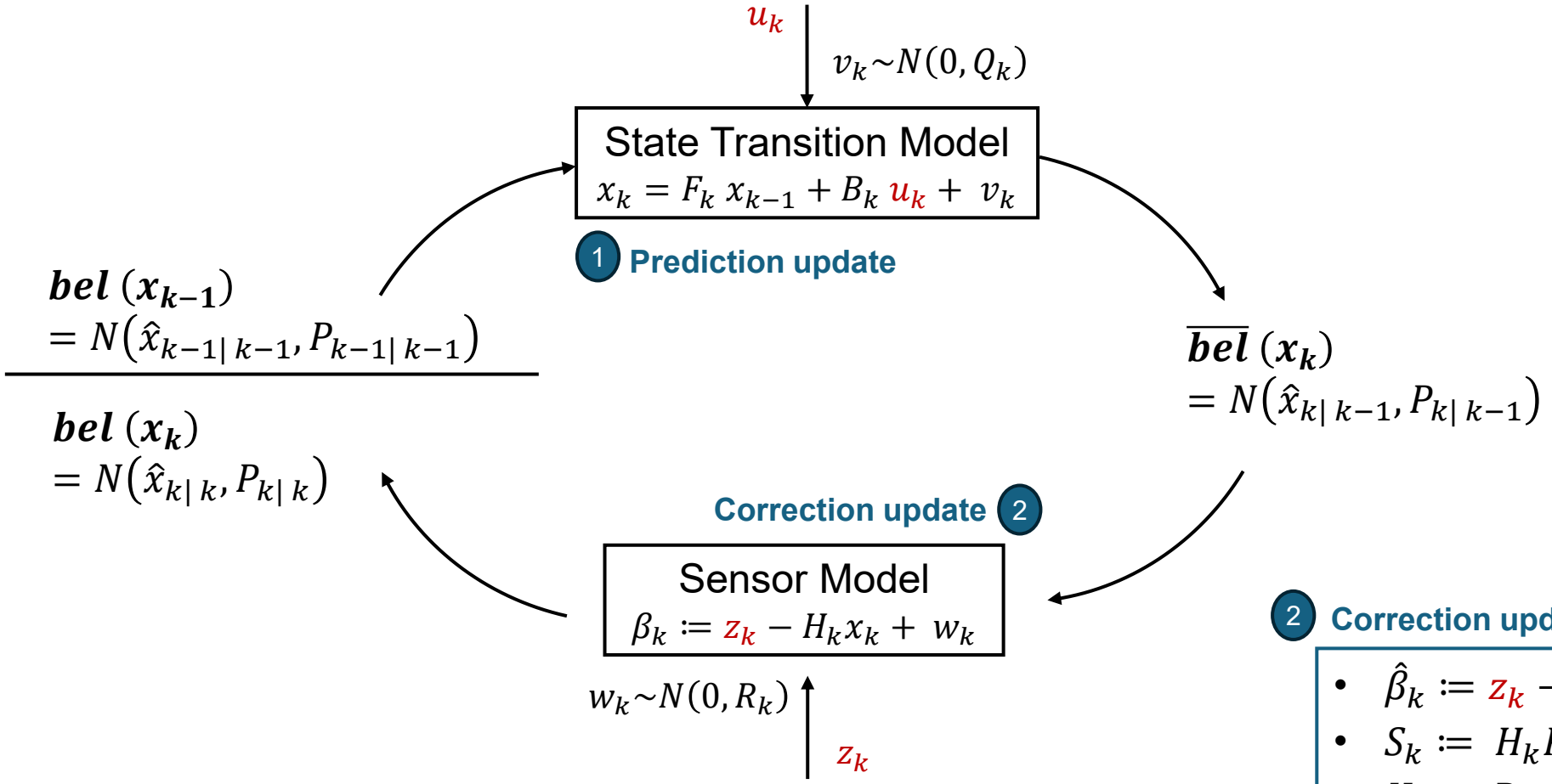
Outline

- Recap last lectures
- Extended Kalman Filter Algorithm
- IMU Orientation Estimation

Outline

- Recap last lectures
- Extended Kalman Filter Algorithm
- IMU Orientation Estimation

Kalman Filter Overview



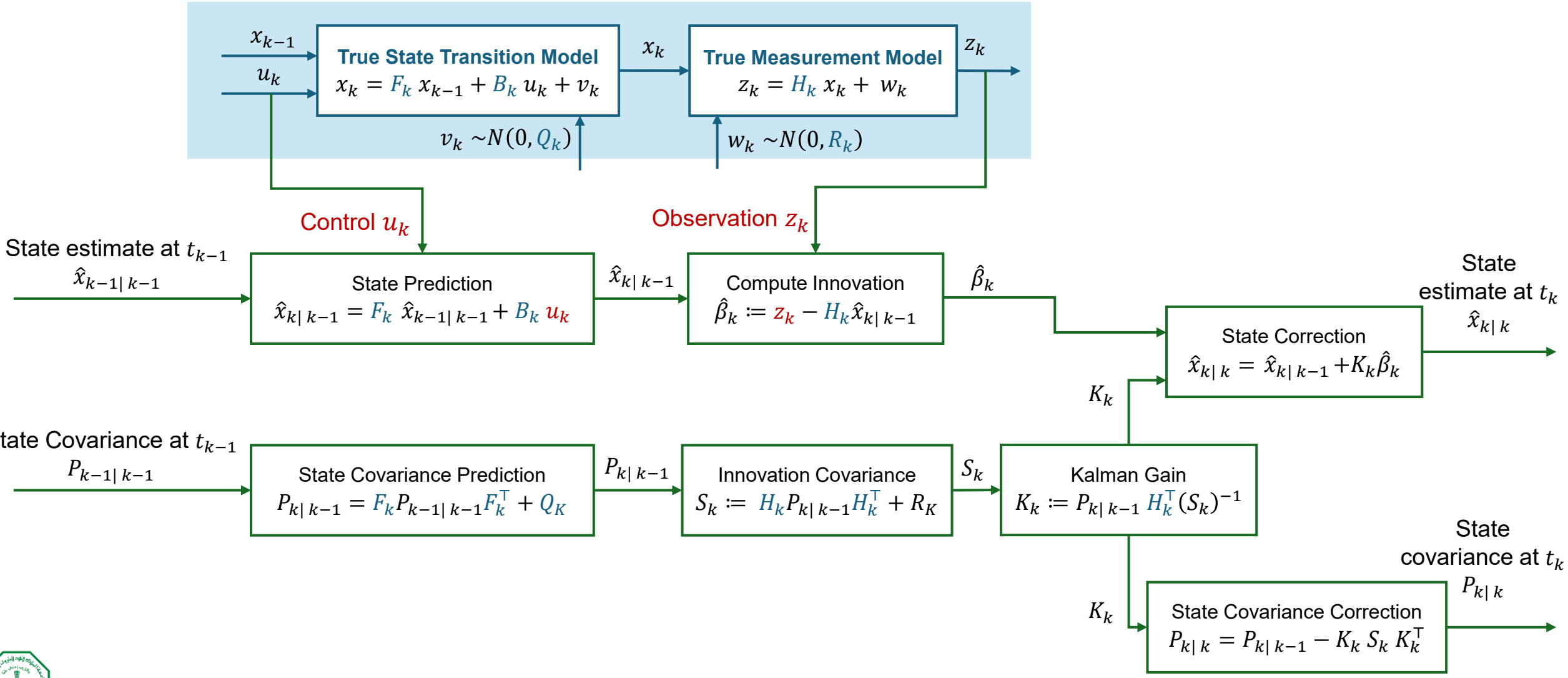
1 Prediction update

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

2 Correction update

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

Kalman Filter Block Diagram



Recap: 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

- 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:

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

$$\in \mathbb{R}^n \qquad \in \mathbb{R}^{n \times n} \qquad \in \mathbb{R}^n$$

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

Recap: Nonlinear State Transition Model

- The actual state transition model is nonlinear of the form

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

- 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 $x_{\mathrm{k-1}}$ evaluated at the latest estimate $\hat{x}_{k-1\,|\,k-1}$

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

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

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

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

Recap: 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 $\hat{x}_{k-1 \mid k-1}$

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

Recap: 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_{k} evaluated at the prediction $\hat{x}_{k \mid k-1}$

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

Recap: 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 $L_k \in \mathbb{R}^{q \times l}$ is the Jacobian of h with respect to w_{k} evaluated at the prediction $\hat{x}_{k \mid k-1}$

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

Outline

- Recap last lectures
- Extended Kalman Filter Algorithm
- IMU Orientation Estimation

Prediction update: KF vs EKF

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

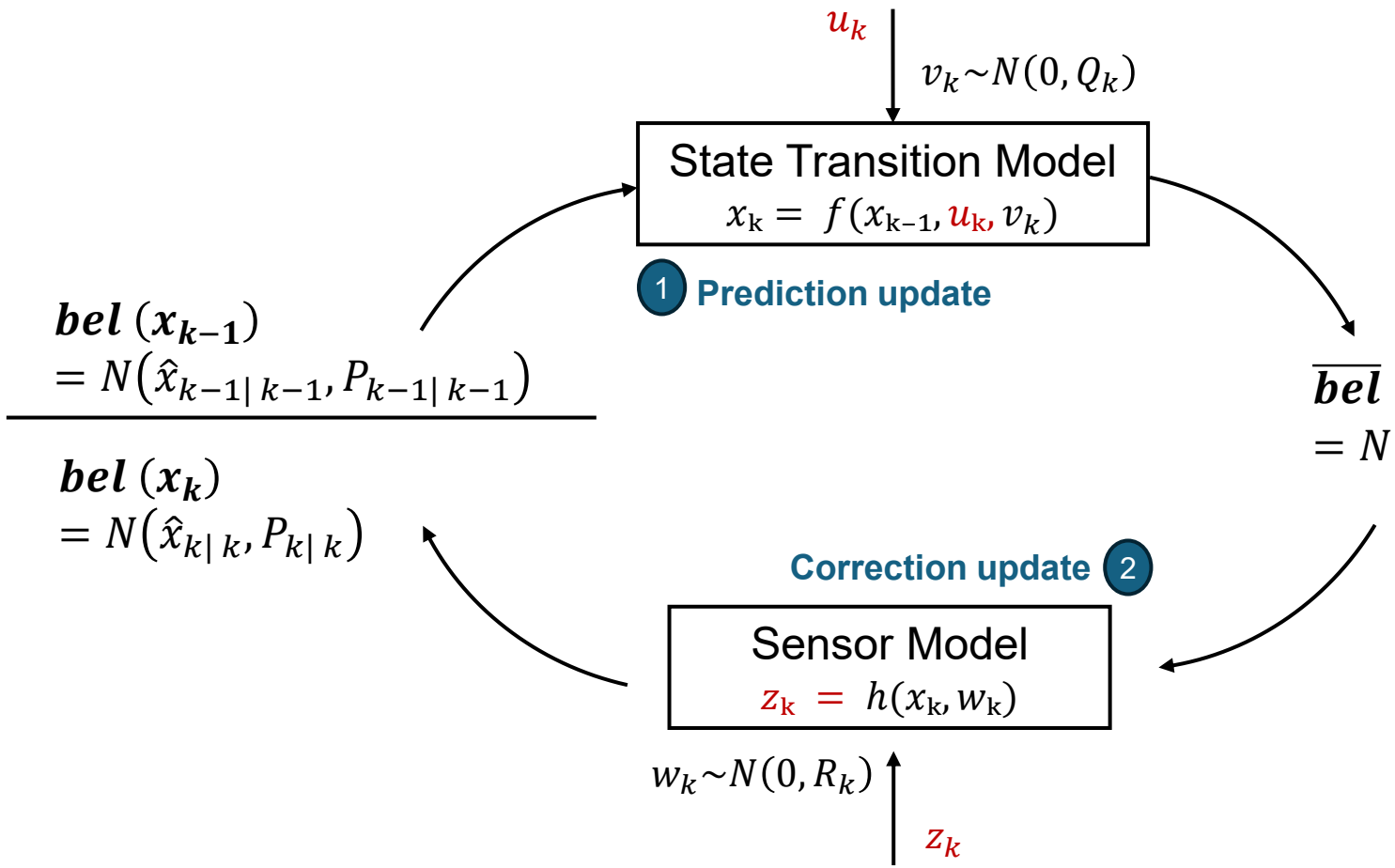
- Extended Kalman Filter
  - $\hat{x}_{k|\,k-1} = f(\,\hat{x}_{k-1|\,k-1}, u_k)$
  - $P_{k|\,k-1} = F_k P_{k-1|\,k-1} F_k^{\top} + G_k Q_K G_k^{\top}$
  - $F_k := \frac{\partial f}{\partial x_{k-1}}\left(\hat{x}_{k-1\,|\,k-1}\right)$
  - $G_k := \frac{\partial f}{\partial v_k}\left(\hat{x}_{k-1\,|\,k-1}\right)$

Correction update: KF vs EKF

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

- Extended Kalman Filter
  - $\hat{x}_{k|\,k} = \hat{x}_{k|\,k-1} + K_k \hat{\beta}_k$
  - $P_{k|\,k} = P_{k|\,k-1} - K_k\, S_k\, K_k^{\top}$
  - $\hat{\beta}_k \coloneqq z_k - h(\hat{x}_{k|\,k-1})$
  - $S_k \coloneqq H_k P_{k|\,k-1} H_k^{\top} + L_k R_K L_k^{\top}$
  - $K_k \coloneqq P_{k|\,k-1}\, H_k^{\top} (S_k)^{-1}$
  - $H_k \coloneqq \frac{\partial h}{\partial x_k}\left(\hat{x}_{k\,|\,k-1}\right)$
  - $L_k \coloneqq \frac{\partial h}{\partial w_k}\left(\hat{x}_{k\,|\,k-1}\right)$

Extended Kalman Filter



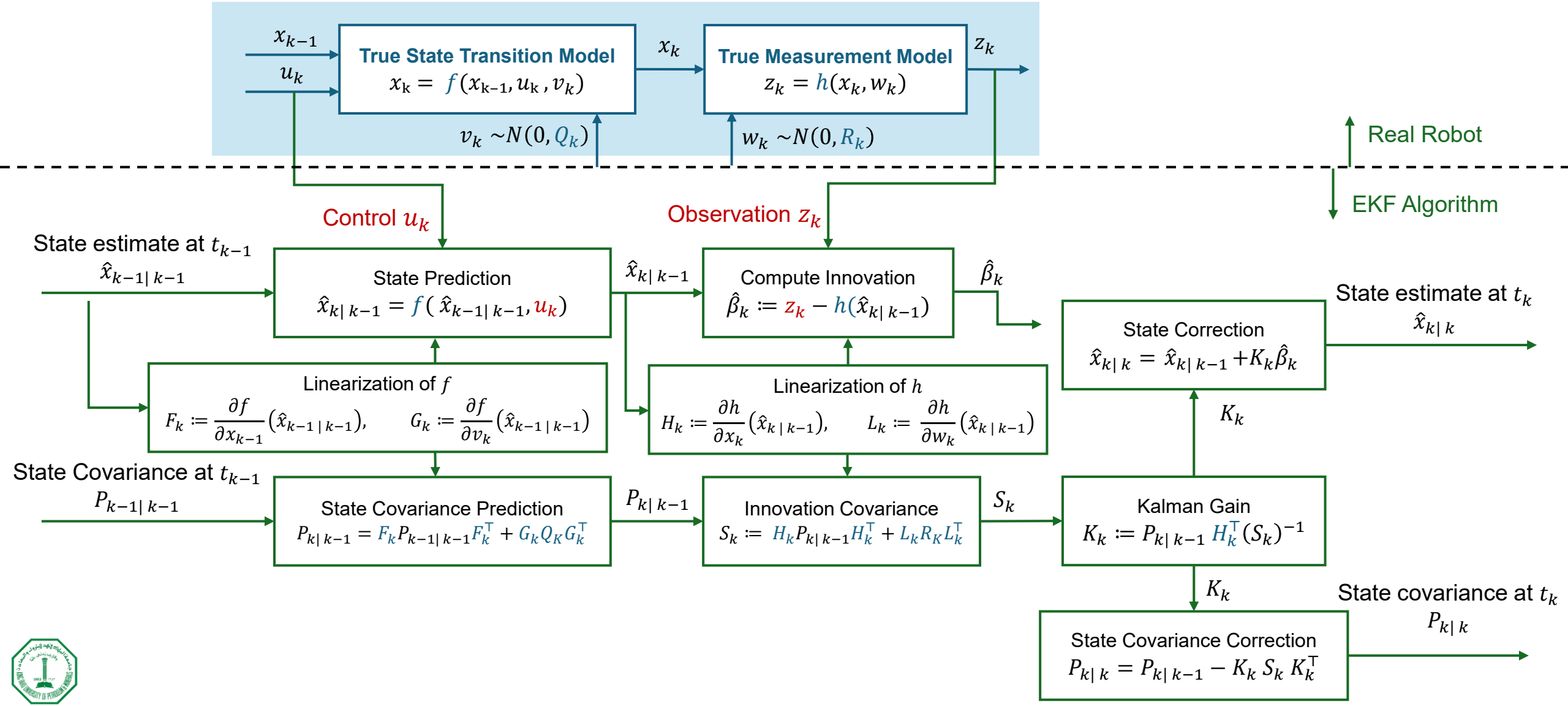
1 Prediction update

- $\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_k)$
- $P_{k|k-1} = F_k P_{k-1|k-1} F_k^\top + G_k Q_K G_k^\top$
- $F_k := \frac{\partial f}{\partial x_{k-1}}(\hat{x}_{k-1|k-1})$
- $G_k := \frac{\partial f}{\partial v_k}(\hat{x}_{k-1|k-1})$

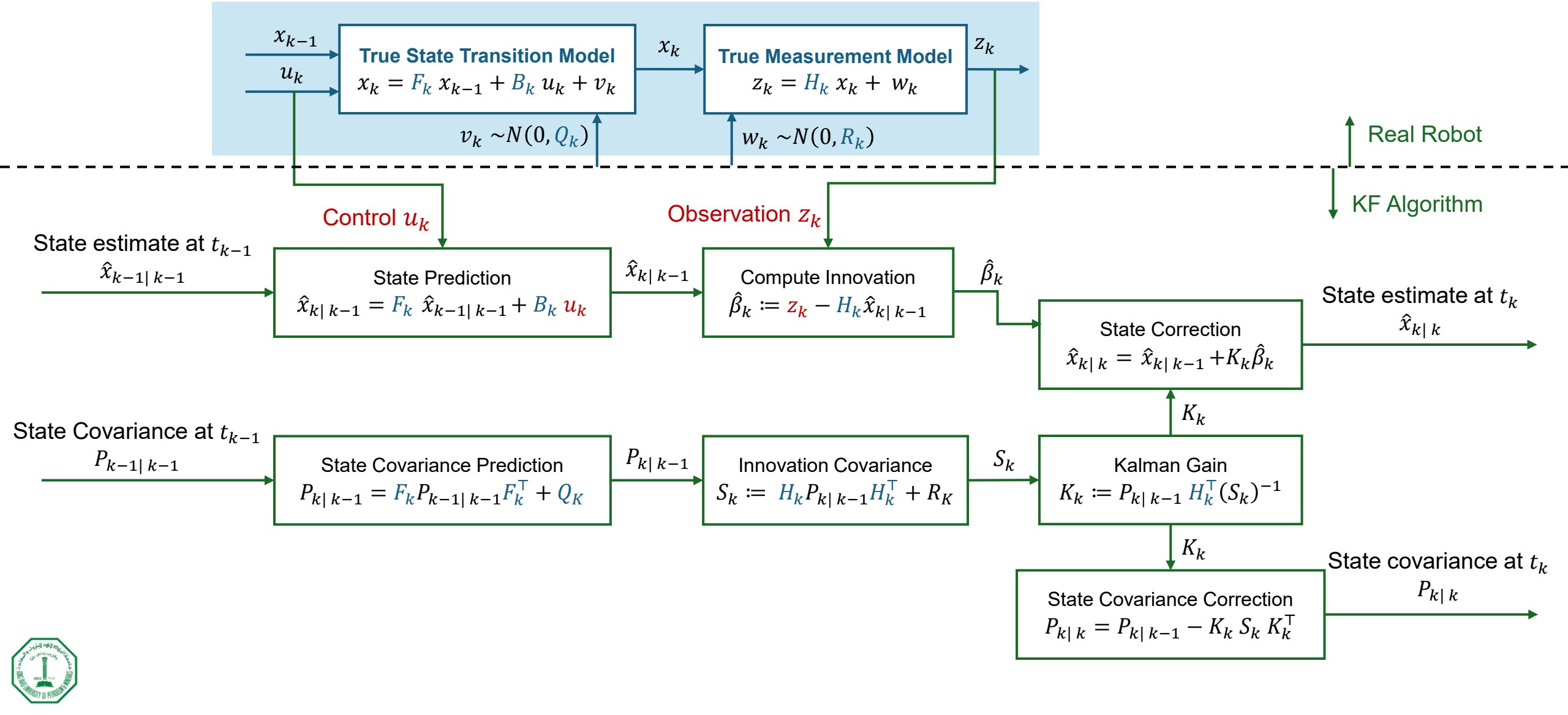
2 Correction update

- $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \hat{\beta}_k$
- $P_{k|k} = P_{k|k-1} - K_k S_k K_k^\top$
- $\hat{\beta}_k := z_k - h(\hat{x}_{k|k-1})$
- $S_k := H_k P_{k|k-1} H_k^\top + L_k R_K L_k^\top$
- $K_k := P_{k|k-1} H_k^\top (S_k)^{-1}$
- $H_k := \frac{\partial h}{\partial x_k}(\hat{x}_{k|k-1})$
- $L_k := \frac{\partial h}{\partial w_k}(\hat{x}_{k|k-1})$

EKF Block Diagram



Standard KF Block Diagram

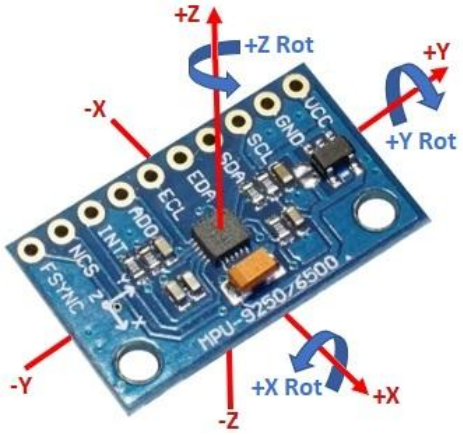


Outline

- Recap last lectures
- Extended Kalman Filter Algorithm
- IMU Orientation Estimation

Problem formulation

- How to fuse the measurements of a 9DOF IMU to estimate the orientation of a robot $R \in SO(3)$?
  - Kinematic Model:
    - $\dot{R}_b^s = R_b^s\,[\omega]$
  - Gyroscope Model:
    - $\omega_{\mathrm{m}} = \omega + b_g + \eta_g$
  - Accelerometer Model:
    - $a_{\mathrm{m}} = R_s^b g + b_a + w_a$
  - Magnetometer Model:
    - $m_{\mathrm{m}} = R_s^b \beta + b_m + w_m$



We assume the IMU is calibrated and non-accelerating.

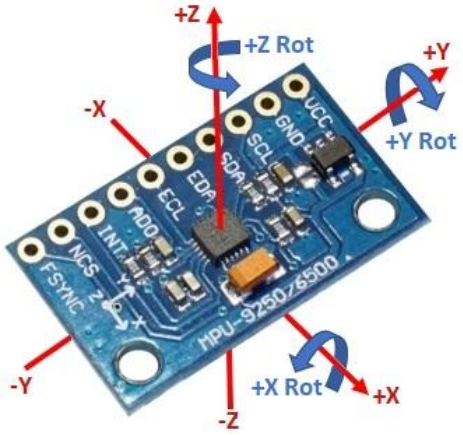
Problem formulation

- How to fuse the measurements of a 9DOF IMU to estimate the orientation of a robot $R \in SO(3)$?
  - Kinematic Model:
    - $\dot{R}_b^s = R_b^s\,[\omega]$
  - Gyroscope Model:
    - $\omega_{\mathrm{m}} = \omega + b_g + \eta_g$
  - Accelerometer Model:
    - $a_{\mathrm{m}} = R_s^b g + b_a + w_a$
  - Magnetometer Model:
    - $m_{\mathrm{m}} = R_s^b \beta + b_m + w_m$

IMU Noise:
- $\eta_g \sim N(0, C_g)$
- $w_a \sim N(0, C_a)$
- $w_m \sim N(0, C_m)$

IMU Biases:
- $b_g, b_a, b_m \in \mathbb{R}^3$

Gravity and Environment Magnetic Field:
- $g = (0,0,-9.81) \in \mathbb{R}^3$
- $\beta = (B_x, B_y, B_z) \in \mathbb{R}^3$

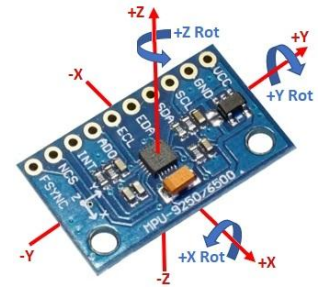


We assume the IMU is calibrated and non-accelerating.

Design Choices

- There are several choices that could be made leading to different Kalman Filter formulations:

1. Orientation Parametrization:
   - **Rotation Matrices** (orthonormality constraints, more complex to update)
   - **Euler Angles** (intuitive but prone to gimbal lock and singularities)
   - **Quaternions** (most common, avoids singularities and allows easy normalization)
2. Bias Handling:
   - **Exclude bias** and assume external calibration or stationarity
   - **Include gyroscope bias** in the state and model it as a random walk
3. Discretization Method



Random Walk

- A **random walk** is a process where a variable changes over time by adding small, random increments.
- Mathematically, you can express it as:

$$\dot{b}(t) = 0 + w(t), \qquad w \sim N(0, Q)$$

- Or equivalently in discrete time as:

$$b_k = b_{k-1} + w_k$$

- This means the bias doesn't stay constant but also doesn't evolve in a predictable way, it just drifts slowly due to noise.

Why Use Random Walk for Bias Estimation?

- In many systems (e.g., inertial sensors like gyros or accelerometers), biases are not truly constant, but they:
  - Change slowly over time,
  - Cannot be measured directly,
  - Corrupt the measurements.
- Modeling them as random walks allows the Kalman filter to:
  - Adapt to changing biases,
  - Estimate them alongside the true state.

- The process noise covariance Q is a tuning parameter that controls how fast the bias is allowed to drift.

EKF Formulation 1

- Let's design an EKF based on Euler angles excluding biases
  - State vector:
    - (Roll, pitch, yaw) angles + Magnetic Field
    - $x = (\alpha, \beta) = (\phi, \theta, \psi, B_x, B_y, B_z) \in \mathbb{R}^6$
  - Kinematic Model:
    - $\dot{\alpha} = T(\alpha)\ \omega$
  - Unbiased Gyroscope Model:
    - $\omega_{\mathrm{m}} = \omega + \eta_g$
  - Unbiased Accelerometer Model:
    - $a_{\mathrm{m}} = R(\alpha)^{\top} g + w_a$
  - Unbiased Magnetometer Model:
    - $m_{\mathrm{m}} = R(\alpha)^{\top} \beta + w_m$

$$T(\alpha) = \begin{pmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & \frac{s_\phi}{c_\theta} & \frac{c_\phi}{c_\theta} \end{pmatrix}$$

$$R(\alpha) = \begin{pmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{pmatrix}$$

EKF Formulation 1

- State Transition Model $\dot{x} = f(x, u, \eta)$:
  - $\begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \end{pmatrix} = \begin{pmatrix} T(\alpha)\left[\omega_{\mathrm{m}} + \eta_g\right] \\ \eta_\beta \end{pmatrix}$
- Discretized Model $x_k = f(x_{k-1}, u_k, \eta_k)$:
  - $\begin{pmatrix} \alpha_k \\ \beta_k \end{pmatrix} = \begin{pmatrix} \alpha_{k-1} + \Delta t\, T(\alpha_{k-1})\left[\omega_{\mathrm{m},k} + \eta_{g,k}\right] \\ \beta_{k-1} + \eta_{\beta,k} \end{pmatrix}$
- Measurement Model $z_k = h(x_k) + w_k$:
  - $\begin{pmatrix} a_{\mathrm{m,k}} \\ m_{\mathrm{m,k}} \end{pmatrix} = \begin{pmatrix} R(\alpha_k)^\top g \\ R(\alpha_k)^\top \beta_k \end{pmatrix} + \begin{pmatrix} w_{a,k} \\ w_{m,k} \end{pmatrix}$

- $x = (\alpha, \beta) \in \mathbb{R}^6$
- $u = \omega_{\mathrm{m}} \in \mathbb{R}^3$
- $z = (a_{\mathrm{m}}, m_{\mathrm{m}}) \in \mathbb{R}^6$

Noise Variables:

- $\eta_{g,k} \sim N(0, C_g)$
- $\eta_{\beta,k} \sim N(0, C_\beta)$
- $w_{a,k} \sim N(0, C_a)$
- $w_{m,k} \sim N(0, C_m)$

Question: How do we compute the Jacobians F_k, G_k, H_k, L_k ?

We choose here the Backward Euler Discretization method

EKF Formulation 2

- Now let's design a more practical EKF that includes biases:
  - State vector:
    - (Roll, pitch, yaw) angles + Magnetic Field
    - $x = \left(\alpha, b_g, b_a, b_m, \beta\right) \in \mathbb{R}^{15}$
  - Kinematic Model:
    - $\dot{\alpha} = T(\alpha)\,\omega$
  - Gyroscope Model:
    - $\omega_{\mathrm{m}} = \omega + b_g + \eta_g$
  - Accelerometer Model:
    - $a_{\mathrm{m}} = R(\alpha)^{\mathsf{T}} g + b_a + w_a$
  - Magnetometer Model:
    - $m_{\mathrm{m}} = R(\alpha)^{\mathsf{T}} \beta + b_m + w_m$

EKF Formulation 2

- Discretized Model $x_k = f(x_{k-1}, u_k, \eta_k)$:
  - $$\begin{pmatrix} \alpha_k \\ b_{g,k} \\ b_{a,k} \\ b_{m,k} \\ \beta_k \end{pmatrix} = \begin{pmatrix} \alpha_{k-1} + \Delta t\, T(\alpha_{k-1})\left[\omega_{\mathrm{m},k} - b_{g,k-1} + \eta_{g,k}\right] \\ b_{g,k-1} + \eta_{b_g,k} \\ b_{a,k-1} + \eta_{b_a,k} \\ b_{m,k-1} + \eta_{b_m,k} \\ \beta_{k-1} + \eta_{\beta,k} \end{pmatrix}$$
- Measurement Model $z_k = h(x_k) + w_k$:
  - $$\begin{pmatrix} a_{\mathrm{m,k}} \\ m_{\mathrm{m,k}} \end{pmatrix} = \begin{pmatrix} R(\alpha_k)^\top g + b_{a,k} \\ R(\alpha_k)^\top \beta_k + b_{m,k} \end{pmatrix} + \begin{pmatrix} w_{a,k} \\ w_{m,k} \end{pmatrix}$$

- $x = (\alpha, b_g, b_a, b_m, \beta) \in \mathbb{R}^{15}$
- $u = \omega_{\mathrm{m}} \in \mathbb{R}^3$
- $z = (a_{\mathrm{m}}, m_{\mathrm{m}}) \in \mathbb{R}^6$

Noise Variables:

- $\eta_{g,k} \sim N(0, C_g)$
- $\eta_{\beta,k} \sim N(0, C_\beta)$
- $\eta_{b_g,k} \sim N\left(0, C_{b_g}\right)$
- $\eta_{b_a,k} \sim N(0, C_{b_a})$
- $\eta_{b_m,k} \sim N(0, C_{b_m})$

- $w_{a,k} \sim N(0, C_a)$
- $w_{m,k} \sim N(0, C_m)$

Question: How do we compute the Jacobians F_k, G_k, H_k, L_k ?