



CIE482: Path Planning & Navigation for Mobile Robots
Assignment 5

Instructions:

- This is a **graded** assignment of **6%**.
- This assignment involves both a written part and a programming part with MATLAB.
- You solve the written part by writing on the pdf directly with a tablet & electronic pen **or** print the pdf and answer with an ink-pen.
- Upload your solution as a **single .zip file** including the **MATLAB code, Plots of the results, and pdf** on Blackboard by **12:00 pm Tuesday 28 April 2026 (Before the lecture)**.
- You may use ChatGpt to explain for you the given MATLAB files. However, **implement the EKF code yourself**.
- Plagiarism will **not be tolerated** at all.

Student Name:	
Student ID:	

**Question 1: Implementing an Extended Kalman Filter (EKF) for IMU Sensor Fusion.**

You are provided with the system model and sensor data for a rigid body. Your task is to implement the Extended Kalman Filter (EKF) prediction and correction steps to estimate:

The body's orientation (Euler angles: roll, pitch, yaw)

The magnetic field components (B_x , B_y , B_z) expressed in the body frame.

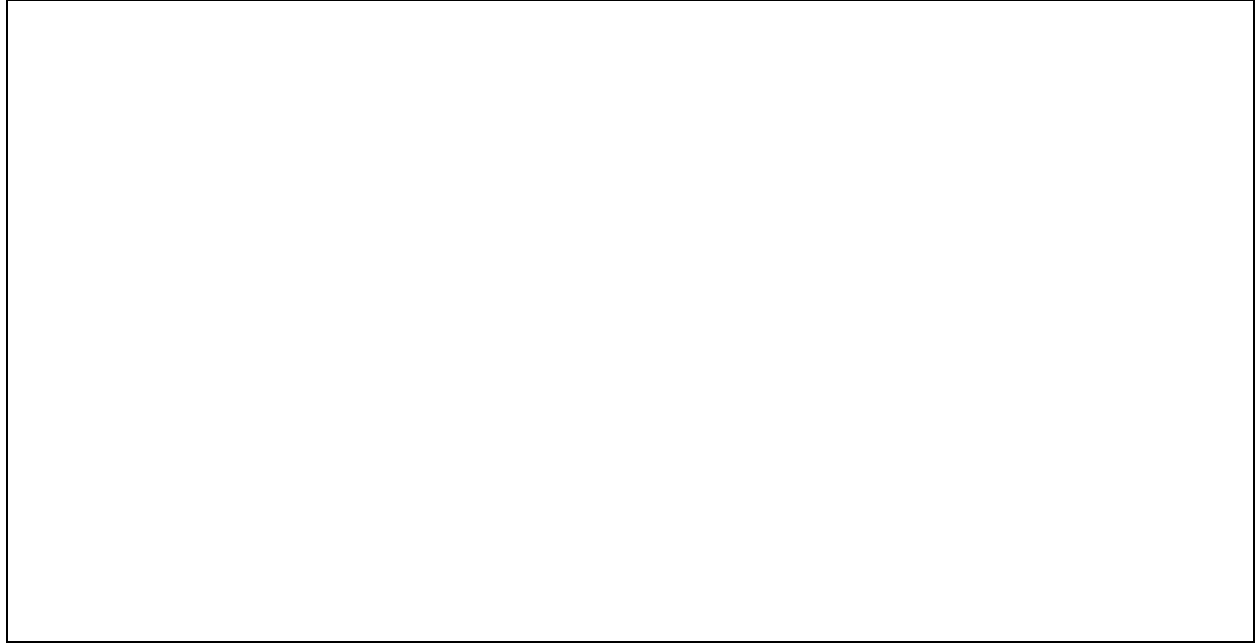
Files Provided

- `system_model.m`: Defines the discrete-time system dynamics $f(x,u)$, measurement model $h(x)$, and their Jacobians $F(x,u)$, $G(x,u)$, and $H(x)$.
- `ekf_imu_fusion_simulation.m`: Sets up the simulation, loads measurement data, and provides necessary functions for system modeling.

Note: The provided `ekf_imu_fusion_simulation.m` has the EKF loop **removed** — you will implement it yourself.

Your tasks:

- a) Write a short paragraph (3–5 sentences) for each file (`system_model.m` and `ekf_imu_fusion_simulation.m`) explaining:
 - What the file provides.
 - What key functions or variables are available.
 - How it fits into the full EKF implementation.
 - This paragraph should be written in your own words to ensure you understand the structure of the code before you begin coding your EKF loop.



- b) Implement an EKF inside the main simulation loop (over time k):
- Use $f_fun(x,u)$, $F_fun(x,u)$, $G_fun(x,u)$ for the prediction model.
 - Use $h_fun(x)$, $H_fun(x)$ for the measurement model.
- c) Once your EKF is working, experiment with tuning the filter by adjusting Q_k and R_k .
- a. Try increasing or decreasing the gyro noise and magnetic field entries in Q_k .
 - b. Try increasing or decreasing the accelerometer & magnetometer noise entries in R_k .
 - c. Find a set of Q_k and R_k values that balance fast convergence and smooth estimates.
 - d. Observe how the filter performance changes when you vary these parameters.
 - e. Save plots of your results to be included with your submission
 - f. Summarize below your observations on these plots discussing how different tuning parameters affected the estimates.

