EE585 Intermediate Report
Visual-Inertial Extended Kalman Filter Navigation
FALL 2025

Çağdaş Güven 2738938
Ege Uğur Aguş 2096063
Umut Can Gülmez 2099067

I. Problem Definition

Autonomous navigation is one of the most important research areas in robotics, requiring precise state estimation. Inertial Measurement Units, (IMUs), are standard in autonomous systems thanks to their high data rate and independence from environmental factors. However, IMUs do not measure position directly, instead they measure linear accelerations and angular velocities. The computation of position from linear accelerations requires double integration, which causes sensor noises to grow quadratically. Moreover, like most of the other sensors IMUs suffer from initial/time-varying biases. Thus, an external reference is required to correct these errors, otherwise, the position estimate diverges over time.

Monocular cameras offer a complementary solution. They are lightweight, low-cost, and capable of observing environmental features that may be processed to serve as references to bound the inertial drift.

The objective of this study is to design and implement a **Visual-Inertial Navigation System (VINS)** for a ground vehicle operating in a planar environment. The system aims to estimate the state of a vehicle,

- Position ($\mathbf{p}$)
- Velocity ($\mathbf{v}$)
- Orientation ($\mathbf{q}$)

by using measurements from two sensors,

1) **Inertial Measurement Unit (IMU):** Provides high-frequency measurements of the linear accelerations and angular velocities of the vehicle.
2) **Monocular Camera:** Provides low-frequency observations of known landmarks, which can be processed to obtain information about the state of the vehicle.

The problem defined for a planar robot, however, in this study the implementation of a **6-Degrees-of-Freedom (6-DoF)** estimation framework is selected. This decision is made to ensure the scalability of this study to general 3D robotic applications.

II. Literature Survey

The main purpose of the project is the integration of inertial and visual data for robust state estimation. In this section, we will explain the state-of-the-art analysis for the Visual-Inertial Navigation.

- **Extended Kalman Filter:** The Extended Kalman Filter (EKF) is one of the mainstream algorithms used in state estimation for robotics applications, where system dynamics are generally nonlinear. This algorithm is an updated version of the Kalman filter that uses linearization (Taylor series expansion) to estimate nonlinear system. Prediction and measurement are the two main steps of the algorithm. [1]
- **Robust Visual Inertial Odometry Using a Direct EKF-Based Approach:** This work introduced a Direct EKF-based Visual-Inertial Odometry (VIO) method. After feature detection they focus on intensity differences between pixels and feed them into EKF as an innovation term. They calculated the landmark positions according to the camera position all the time. Their approach gives successful results in dynamic environments. According to their results, the system is robust to motion blur and feature scarcity. [5]
- **Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft MAVs:** This paper designed an efficient, robust, tightly-coupled monocular VIO system for MAVs. It used the EKF as the main fusion method. This study used IMU state and correcting it using the error from visual features. The system achieved a low complexity and high stability for autonomous vehicles with limited resources. [6]
- **Semi-Direct Visual Odometry for Monocular, Wide-angle, and Multi-Camera Systems (S-DVO):** S-DVO introduced a semi-direct approach to visual odometry, which has both the benefits of feature-based and direct methods. This system uses the visual features for robust tracking while creating a direct image alignment technique to improve pose estimates. They tested their system on quadrotors and 3d scanning applications with smartphones. [7]
- **Indirect Kalman Filter for 3D Attitude Estimation:** This work focuses on the implementation of the Indirect Kalman Filter (IKF), the

initial paper of the Error State Kalman Filter (ES-EKF), for estimating 3D attitude. The IKF maintains a separate nominal state, like the current quaternion attitude, and a small-scale error state, such as the difference between the true and nominal attitude. The EKF linearization is applied only to the smaller error state, which results in improved accuracy and stability when integrating high-rate sensor data like gyroscopes. [9]

- **Extended Kalman Filter vs. Error State Kalman Filter for Aircraft Attitude Estimation:** This paper compares the Extended Kalman Filter (EKF) with the Error State Kalman Filter (ES-EKF) for aircraft attitude estimation. The study shows that the ES-EKF offers advantages in stability, consistency, and computational efficiency, especially when rotations are represented by quaternions. In regular EKF, you linearize the whole state (position, velocity, orientation), which can be very nonlinear and cause problems. In ES-EKF, you keep a nominal state that you propagate normally, and only estimate the small error state between your estimate and the truth. Since this error is small, when you linearize it, the approximation is much better. [8]

III. Methodology

The system will be implemented in *ROS 2 Humble* using a high-level programming language, *C++* or *Python*. These are selected to create a code base, which can be utilized, for both simulated and real-world environments. The primary objective is the validation of the system with the *Gazebo Fortress*, using the available ground truth information to validate the performance.

The main deliverable decided in the methodology of the project are,

- ES-EKF implementation and plots of pose estimates vs. ground truth
- Bias initialization for the IMU
- Quaternion-based orientation representation

A. Filter Architecture

The planned filter architecture is an **Error-State Extended Kalman Filter (ES-EKF)** rather than a standard *Extended Kalman Filter*. This selection is made for being able to work with quaternions for orientation, instead of Euler angles.

Standard EKF applies additive corrections ($\mathbf{x}_{new} = \mathbf{x}_{old} + \Delta\mathbf{x}$). However, the unit quaternions are subject to unit-norm constraint, ($||\mathbf{q}|| = 1$). Thus, adding corrections directly to a quaternion violates the unit-norm constraint. Moreover, a quaternion has 4 parameters with only 3 degrees of freedom, *DoF*. Modeling its uncertainty with a 4×4 covariance matrix leads to rank deficiency (singularity), causing numerical instability during the matrix inversion required for the Kalman Gain.

The ES-EKF resolves these issues by defining the state as two distinct parts,

1) **Nominal State:** Holds the large values, handles non-linearities/constraints.
2) **Error State:** Holds the low values, handles noise from motion and sensor measurements and linear.

The true state is obtained with sum (multiplication for orientation) of nominal and error states.

B. State Vector Definitions

The **Nominal State** vector $\mathbf{x} \in \mathbb{R}^{16}$ comprises the kinematic properties and sensor biases,

$$\mathbf{x} = \begin{bmatrix}\mathbf{p}_{wb}^T & \mathbf{v}_{wb}^T & \mathbf{q}_{wb}^T & \mathbf{b}_a^T & \mathbf{b}_g^T\end{bmatrix}^T \tag{1}$$

where $\mathbf{q}_{wb} \in \mathbb{R}^4$ is the unit quaternion.

The **Error State** vector $\delta\mathbf{x} \in \mathbb{R}^{15}$ is defined as,

$$\delta\mathbf{x} = \begin{bmatrix}\delta\mathbf{p}^T & \delta\mathbf{v}^T & \delta\boldsymbol{\theta}^T & \delta\mathbf{b}_a^T & \delta\mathbf{b}_g^T\end{bmatrix}^T \tag{2}$$

$\delta\boldsymbol{\theta} \in \mathbb{R}^3$ represents the 3-axis angular error. The relationship between the true orientation $\mathbf{q}_{true}$, nominal $\mathbf{q}_{nom}$, and error $\delta\boldsymbol{\theta}$ is multiplicative,

$$\mathbf{q}_{true} \approx \mathbf{q}_{nom} \otimes \begin{bmatrix}1 \\ \frac{1}{2}\delta\boldsymbol{\theta}\end{bmatrix} \tag{3}$$

C. Prediction

The nominal state is propagated using kinematics with the IMU measurement $(\mathbf{a}_m, \boldsymbol{\omega}_m)$ at time k,

$$\dot{\mathbf{p}} = \mathbf{v} \tag{4}$$

$$\dot{\mathbf{v}} = \mathbf{R}(\mathbf{q})(\mathbf{a}_m - \mathbf{b}_a) - \mathbf{g} \tag{5}$$

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \otimes (\boldsymbol{\omega}_m - \mathbf{b}_g) \tag{6}$$

Simultaneously, the error covariance matrix, $\mathbf{P}$, is propagated using the Jacobian of the error state transition function, $\mathbf{F}_x$,

$$\mathbf{P}_{k|k-1} = \mathbf{F}_x\mathbf{P}_{k-1|k-1}\mathbf{F}_x^T + \mathbf{Q}_{imu} \tag{7}$$

D. Measurement Update

When a camera observation $\mathbf{z}_{meas}$ is available, the update occurs in the error space,

1) **Residual:** Compute the difference between the observed pixel and the projected nominal state, $\mathbf{r} = \mathbf{z}_{meas} - h(\mathbf{x}_{nom})$.
2) **Kalman Gain:** Compute $\mathbf{K}$ using the Jacobian $\mathbf{H}$, note that it is w.r.t error state,

$$\mathbf{K} = \mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R}_{cam})^{-1} \tag{8}$$

3) **Estimate Error:** $\delta\hat{\mathbf{x}} = \mathbf{K}\mathbf{r}$.
4) **Injection:** The estimated error is injected into the nominal state. For position/velocity/bias, this is additive ($\mathbf{x} = \mathbf{x} + \delta\hat{\mathbf{x}}$). For orientation, it is multiplicative.
5) **Reset:** After injection, the error state $\delta\mathbf{x}$ is reset to zero, and the covariance is updated: $\mathbf{P} = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}$.

IV. Expected Outcome

Followings are the expected outcomes of this study,

1) **Estimation Package with Simulation Environment:** A functional ROS 2 package containing the *ES-EKF Node*, and a *Gazebo Simulation Environment*, where the *ES-EKF Node* can be tested.
2) **Trajectory Validation:** Analysis of the estimated vehicle path against the ground truth. The aim is to compare the following versions,
   - **Only IMU:** Using only IMU measurements to track the desired trajectory. It is expected to drift from the *desired* due to the double integration of sensor noises and biases.
   - **Visual-Inertial:** Using Visual measurement along with the IMU measurements to track the desired trajectory. It is expected to track the desired trajectory, with bounded positional error, validating the effect of the visual measurement updates.
3) **Filter Analysis:** Comparison of the estimation error ($\mathbf{x}_{est} - \mathbf{x}_{true}$) with the covariance matrix $\mathbf{P}$. The error is expected to remain in the $\pm 3\sigma$ interval of the $\mathbf{P}$, showing filter is consistent and correctly estimate the uncertainty.
4) **Bias Estimation:** The estimated accelerometer and gyroscope bias states ($\mathbf{b}_a, \mathbf{b}_g$) are expected to converge to the actual biases defined for the sensor in the *Gazebo*.

V. Performance Metrics

To evaluate the success of the proposed Visual-Inertial Navigation System project, we will need to use a set of metrics that compare the estimated states with the ground truths provided by Gazebo and ROS2 topics in our simulation environment.

- **Absolute Trajectory Error (ATE):** We will measure the trajectory accuracy using Root Mean Square Error between estimated and ground truth positions. At each time step k, we compare the estimated position p_{est} with ground truth p_{gt}:

  $$ATE_{rmse} = \sqrt{\frac{1}{N}\sum_{k=1}^{N} ||p_{est,k} - p_{gt,k}||^2} \quad (9)$$

  This gives us a single scalar value showing the overall position accuracy of the system.
- **Normalized Estimation Error Squared (NEES) :** Validation of the EKF's consistency (For example whether predicted uncertainty matches the reality) will be computed using the NEES. For a state dimension n, the NEES at time k defined as:

  $$\epsilon_k = (x_k - \hat{x}_k)^T P_k^{-1} (x_k - \hat{x}_k) \quad (10)$$

  For a consistent filter the average NEES should be close to n. This is a sanity check for our covariance propagation. If errors follow the Gaussian behavior and the covariance matrix P is correct, this means the filter is consistent, and as a result calculated scalar value will follow a Chi-Square distribution with n degrees of freedom.
- **Sensor Comparison and Drift Rate:** We will do a comparison analysis to show if either of other sensors was superficial after sensor fusion. We will plot the cumulative error over distance traveled for three different modes.
  1) **Dead Reckoning (IMU Only):** build a baseline for drift. We should be seeing the IMU sensor inherent drift behavior here.
  2) **Visual-Only (Camera):** To check if either of the sensors is integrated successfully and the difference in the scale. Our Camera should correctly update the covariance matrix with the known landmark detection.
  3) **Tightly-Coupled Fusion:** To show the final system performance combining motion model (prediction) and measurement (correction).

  Results should show that the Fusion mode which is the full Kalman Filter (prediction and correction), will have a lower slope in the error-vs-distance plot compared to single sensor test modes.

VI. Work Allocation

Implementation of the project will be divided into the following parts:

1) **Simulation and Development Environment Setup:** We will develop a containerized environment for the Gazebo Fortress and ROS2 Humble for a standardized development environment. Umut Can will implement this part.
2) **Inertial and Visual Sensor Data Retrieval:** In this part, we will obtain and validate the sensor data from the simulation. Any necessary noise modeling or addition will be handled here. We will have RGB images, raw IMU data, and ground truth data from the simulation environment. Çağdaş will implement this part.
3) **Visual Data Processing:** Visual feature detection will be handled in this part. We will implement a simple landmark detection node for the project. This node will output the estimated pose. This output will later be used in measurement step of the EKF. Umut Can and Ege will implement this part.
4) **Error State Extended Kalman Filter Implementation:** In this part, we will synchronize the visual detection message and imu data that was calculated on the previous work packages and use this information as an input to our custom ES-EKF implementation. The prediction will be calculated as long as IMU data comes to the algorithm, and the measurement will be calculated when a visual detection message is retrieved. The

final output will be the fused odom message, and we will use the robot state in our evaluation. All team members is responsible for this task.

5) **Evaluation:** All members will collaborate on testing, and creating evaluation metrics of the project. The output topic fused odom and simulation's ground truth topic will be used to calculate metrics like the RMSE of the position and orientation, showing how well our implementation is working. We will try different noise levels and feature density on the map for our evaluation.

References

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[2] A. I. Mourikis and S. I. Roumeliotis, "A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation," in *Proc. 2007 IEEE International Conference on Robotics and Automation (ICRA)*, Rome, Italy, 2007, pp. 3565–3572, doi: 10.1109/ROBOT.2007.364024.

[3] C. Campos, R. Elvira, J. J. Gomez Rodriguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021, doi: 10.1109/TRO.2021.3075644.

[4] M. Bloesch, M. Burri, M. Hutter, and R. Siegwart, "Robust Visual Inertial Odometry Using a Direct EKF-Based Approach," in *Proc. 2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 4771–4778.

[5] M. Bloesch, M. Burri, P. Laube, M. Hutter, and R. Siegwart, "Kalman filter based visual-inertial odometry using direct photometric feedback," in *Proc. 2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 4763–4770.

[6] T. Qin, S. Shen, and F. Zeng, "Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft MAVs," in *Proc. 2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 5303–5310.

[7] C. Forster, Z. Zhang, M. Wuersch, and D. Scaramuzza, "Semi-Direct Visual Odometry for Monocular, Wide-angle, and Multi-Camera Systems," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 202–219, 2017.

[8] V. K. Madyastha and K. Ravindra, "Extended Kalman Filter vs. Error State Kalman Filter for Aircraft Attitude Estimation," in *AIAA Guidance, Navigation, and Control Conference*, 2011, p. 6615.

[9] N. Trawny and S. I. Roumeliotis, "Indirect Kalman Filter for 3D Attitude Estimation," Technical Report TR-2005-002, University of Minnesota, Department of Computer Science and Engineering, 2005.