Ai-Driven Signal Filtering For Enhanced Kalman Filter Performance in High-Noise Environment

Abbas Ogal
Faculty Of Engineering And Architecture
Department of Aeronautical Engineering
Gelisim University
Istanbul, Turkey
abbas.ogal@ogr.gelisim.edu.tr

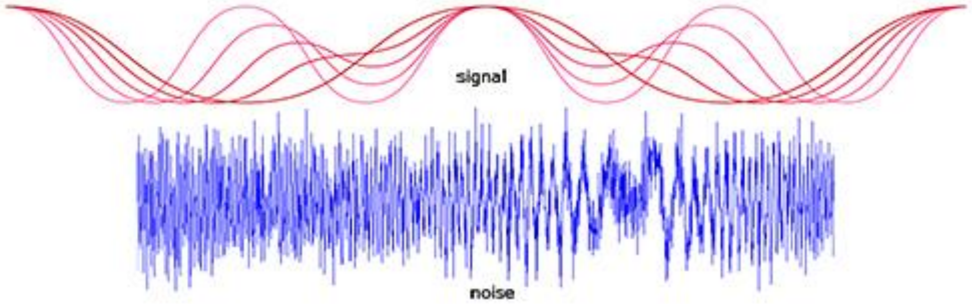
Asst. Prof. PERİ GÜNEŞ
Faculty Of Engineering And Architecture
Department of Aeronautical Engineering
Gelisim University
Istanbul, Turkey
pgunes@gelisim.edu.tr

Overview

- Kalman filter is **widely used in signal estimation**.
- **The Extended Kalman Filter** (EKF) face challenges in dynamic noisy environments.
- This research proposes using **artificial intelligent(AI)** to improve EKF performance.
- Experimental comparison with IMU sensor data.

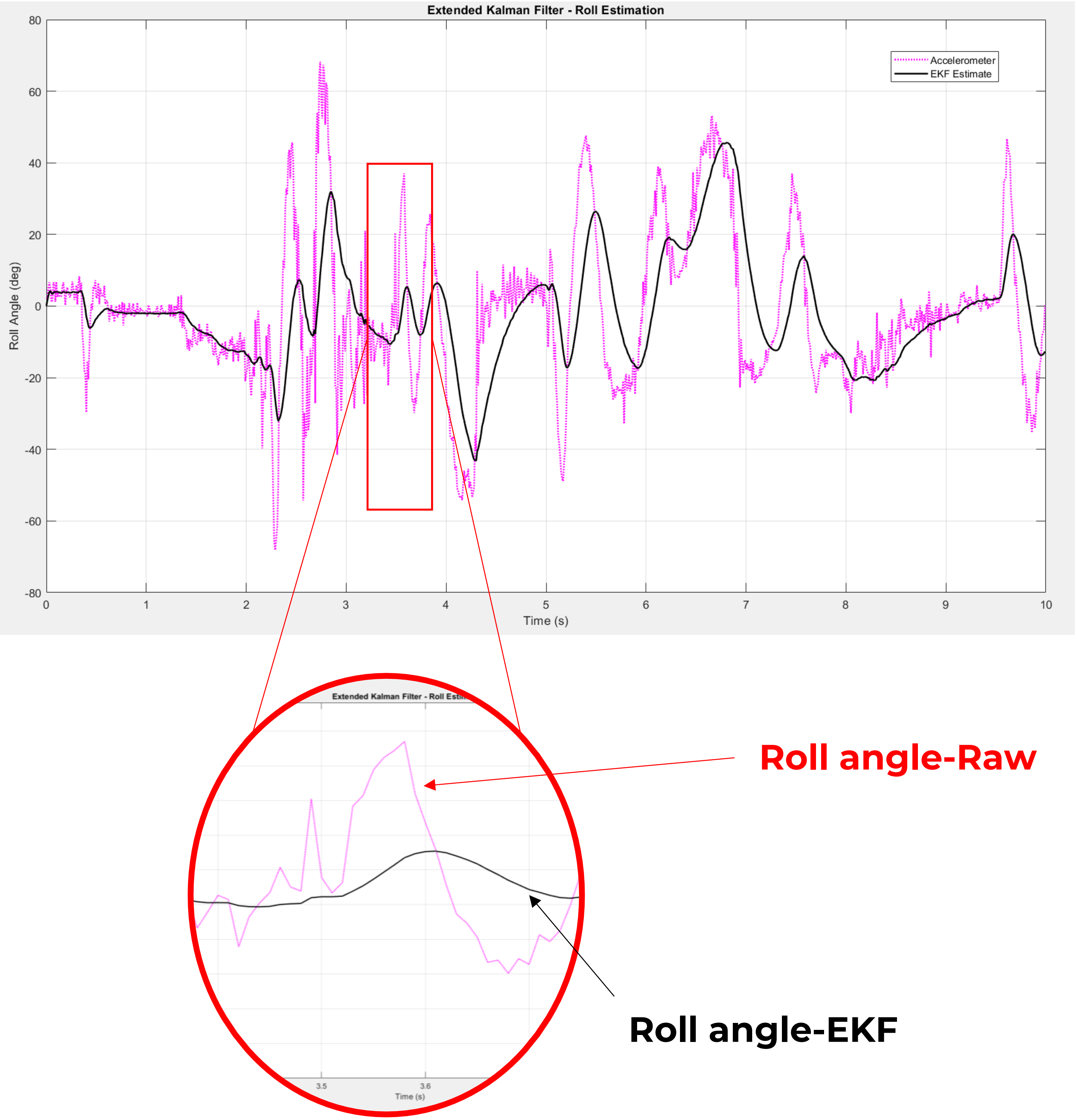
Introduction

- Signal **noise** is a key problem in the Research.
- **Extended Kalman Filter** is popular but struggles under noise.
- **Artificial Intelligent** can help enhance EKF performance by ***Parameter tuning*** Method (Q, R).



Research Problem

- Extended KF has limitations with non-linear data.
- Noise distorts data → high error rate.
- AI can dynamically predict/correct signal errors.

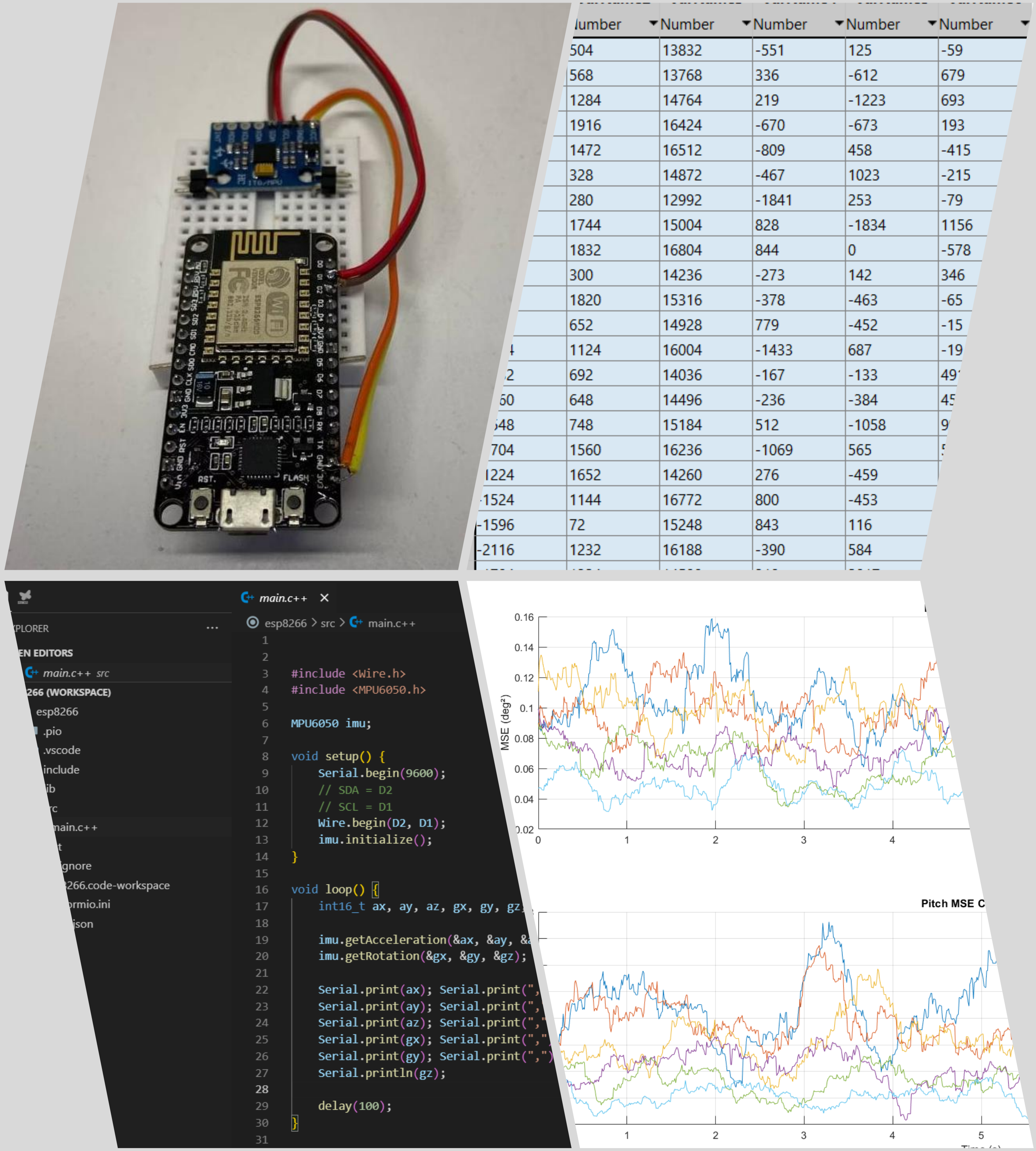


Objectives

- Improve EKF by integrating AI-Models.
- Predict noise **Parameter**(Q,R) before filtering.
- Reduce Meas Square Error (MSE).
- Enhance filter stability.

Implementation Details

- Data Collection.
- Data training by using Different typer of AI.
- Comparing Data
- Environment System



Data Collection Process

Connect ESP32 With Computer

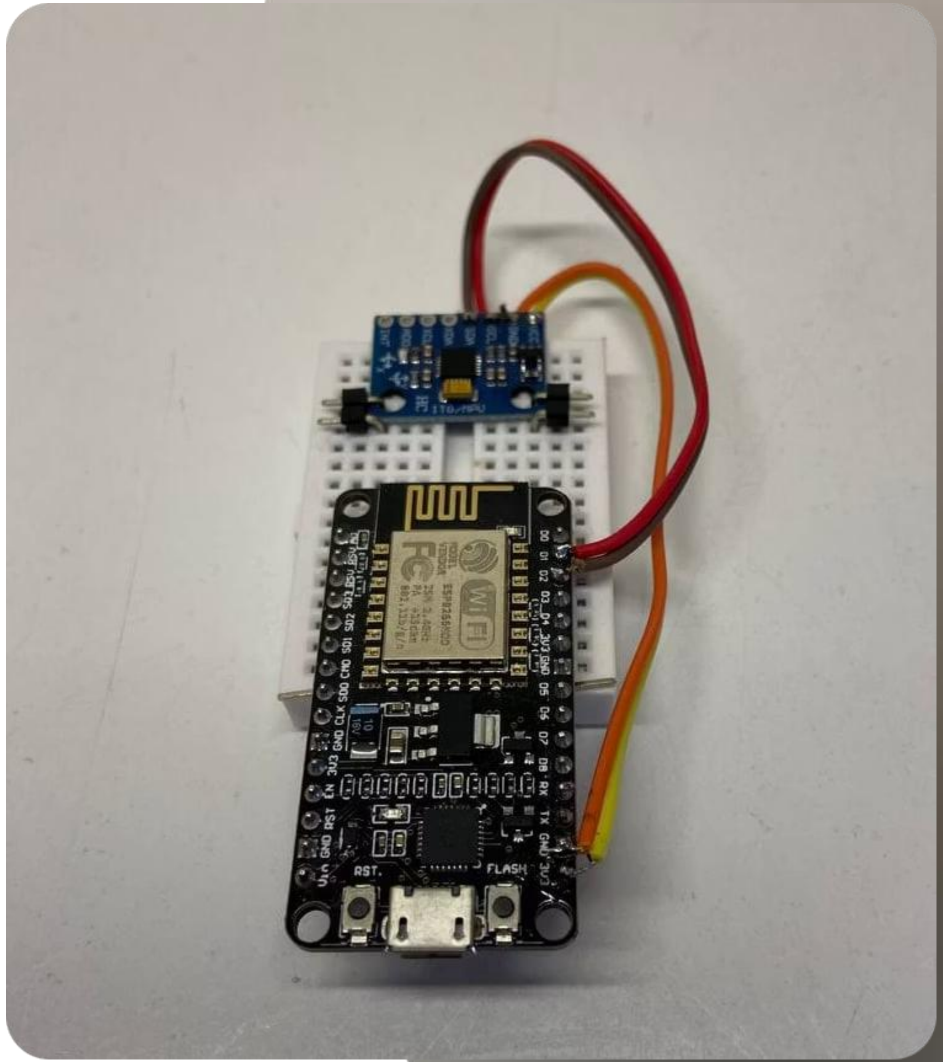
Use USB to connection **ESP32** with a computer.

Connect ESP32 with IMU

Interface the ESP32 microcontroller with the IMU sensor for real-time data capture by using I2C protocol.

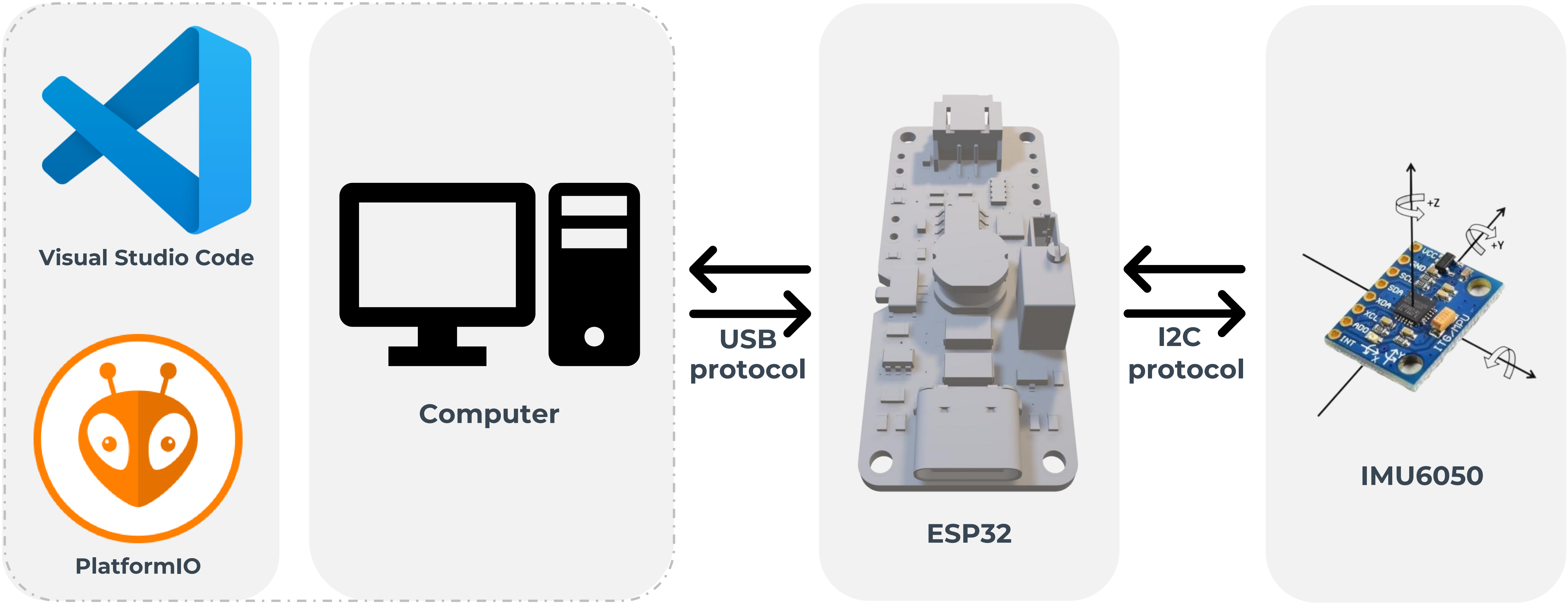
Record Sensor Data

Capture Data from IMU conditions for analysis.



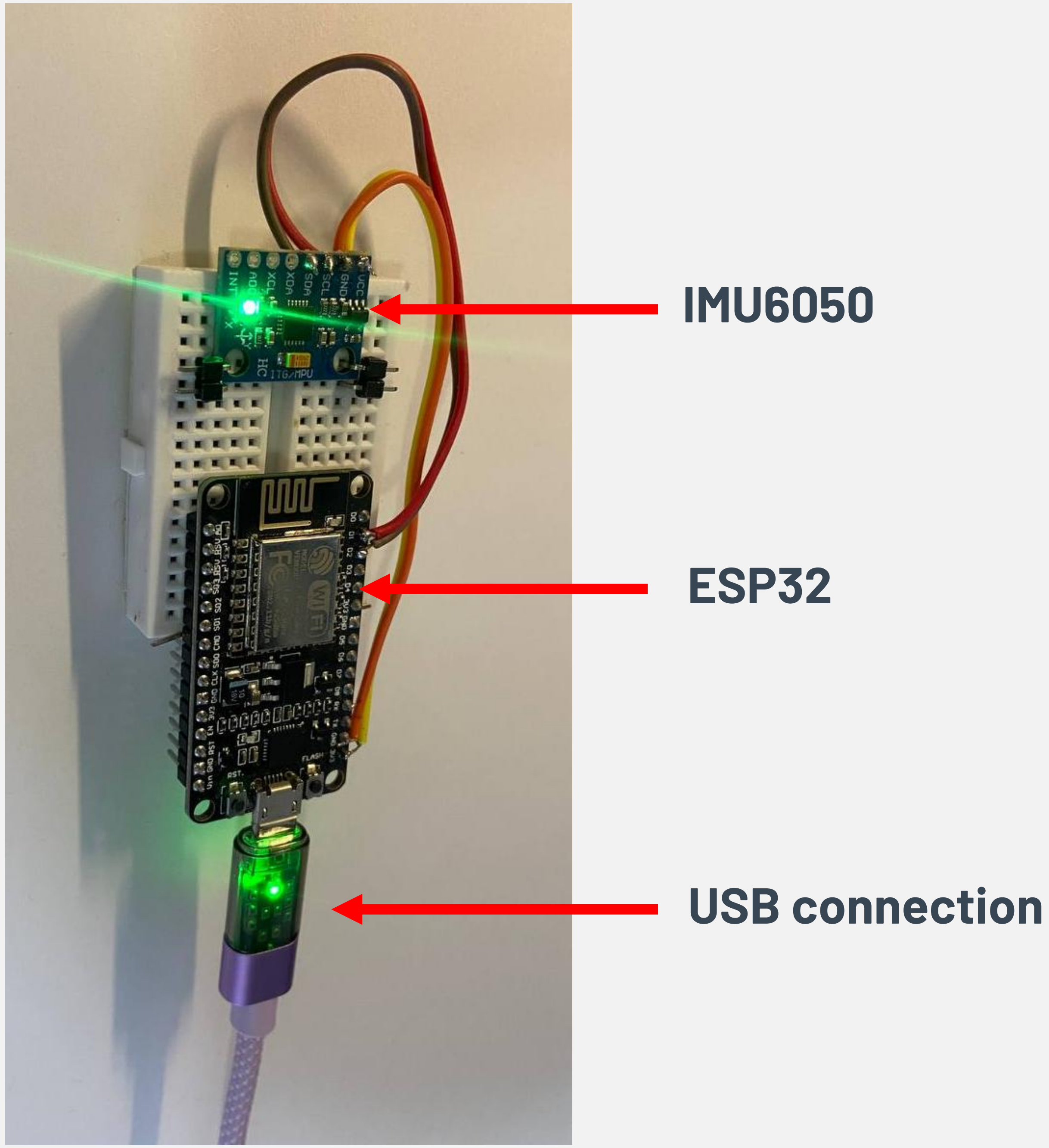
Data Collection Process
Connection system

use **Visual Studio Code** and **PlatformIO** Extinction for streamlined data capture this setup enables real-time IMU sensor data recording for analysis.



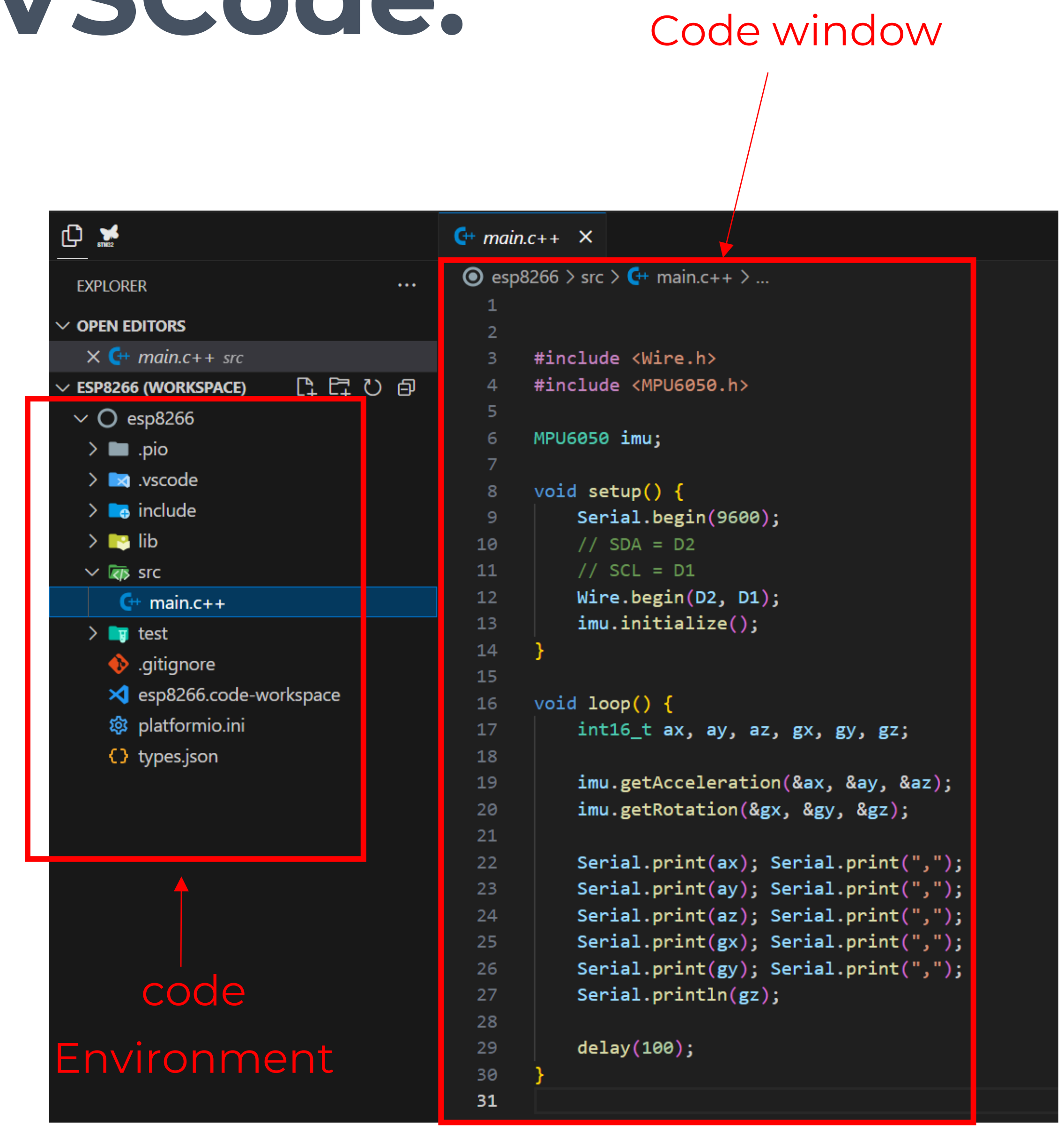
Data Collection Process
Hardware Setup

- The **IMU6050 sensor** is connected to the ESP32 board using I2C. The ESP32 is powered and programmed through a USB cable.



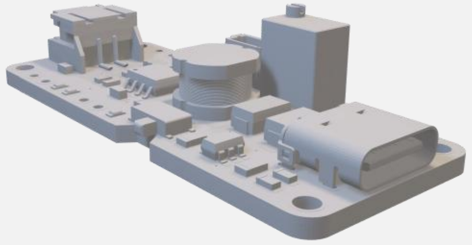
Data Collection Process
Environment PlatformIO with VSCode.

- **Code Environment**: This shows the folder structure of the **project**, including essential directories like (src, lib, and *configuration files*) such as **platformio.ini**. The main code file **main.cpp** is located in the src folder.
- **Code Window**: This displays the actual C++/C code used to collect data from the **MPU6050 sensor (IMU).** The code initializes the sensor and collects accelerometer and gyroscope data.



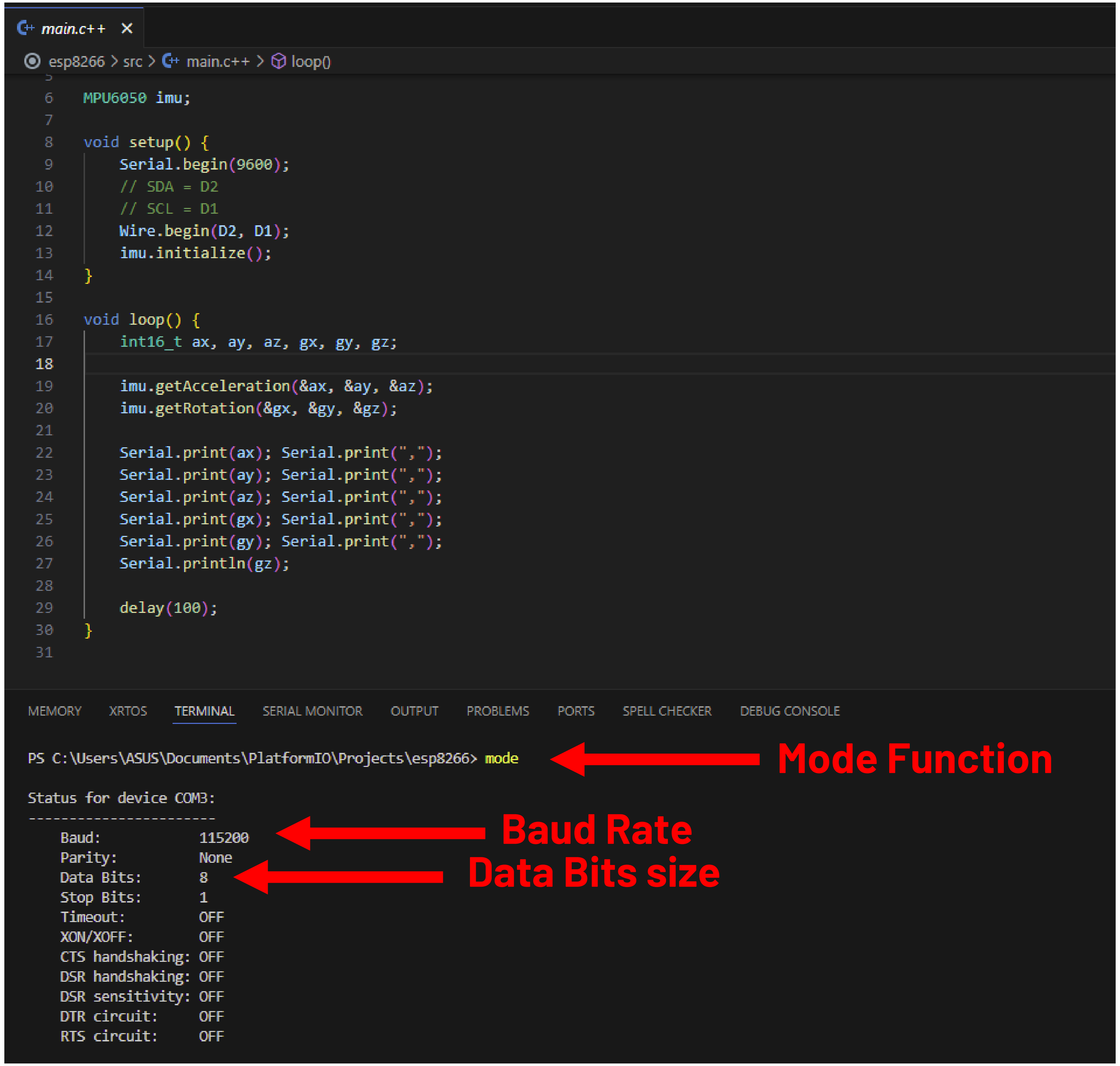
Data Collection Process

Visual Studio Code



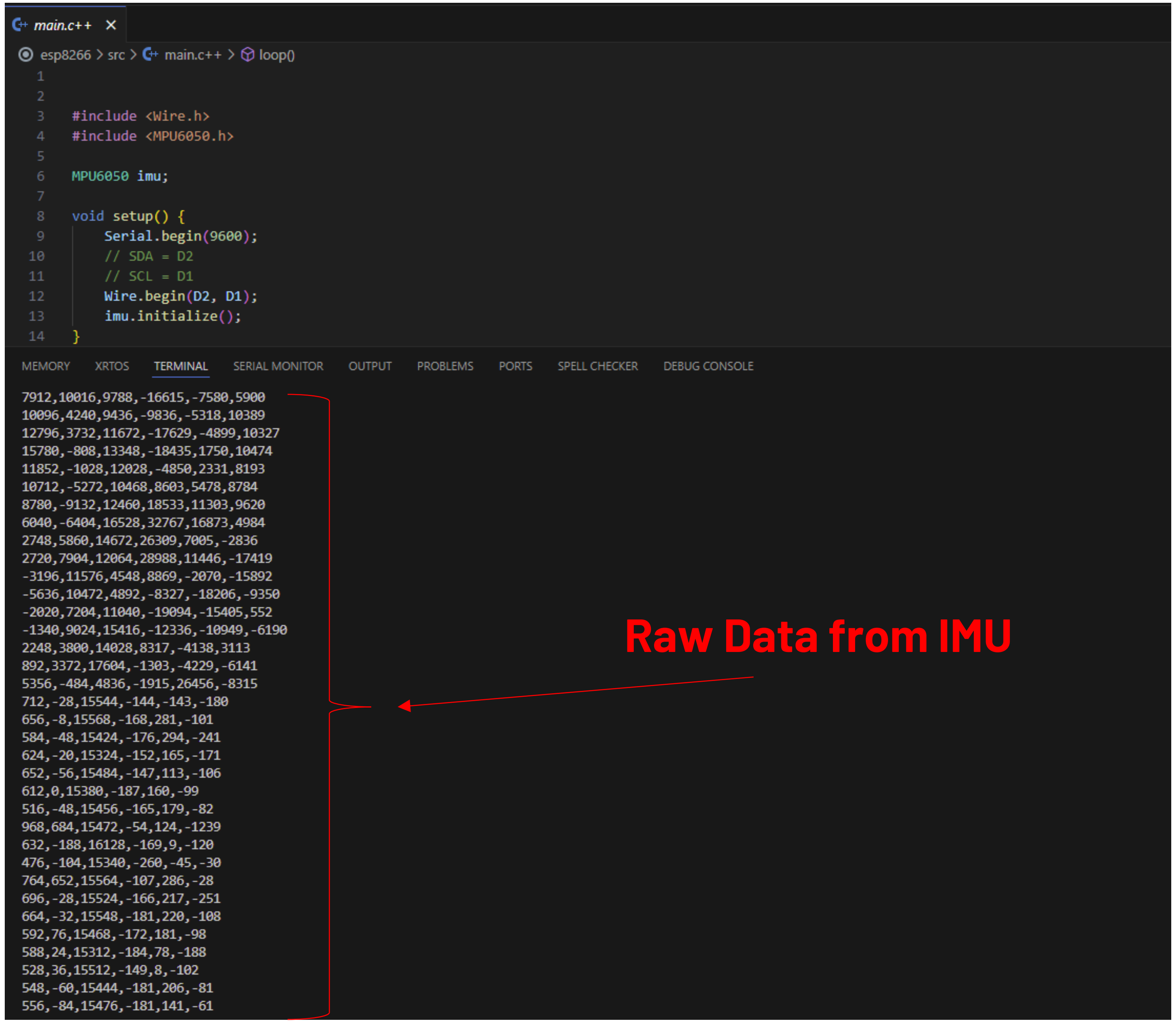
Connect ESP32 with IMU

- Using **Mode Function** inside terminal in Project **Workspace**
- To Checking Connection Between **system** (Computer) and **subsystem** (ESP32)



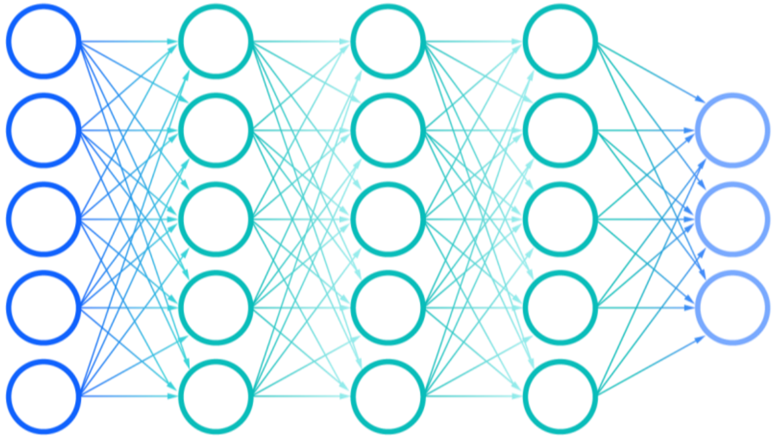
Data Collection

- **Raw Data** Output The ESP32 receives raw data from the IMU6050 and sends it via serial communication. The data includes acceleration and gyroscope values displayed in the **Serial Monitor**.



Processing Data

- Data Processing By using **MATLAB Software**.
- Using Raw Data to training **AI models** to prediction (Q,R)* value in Extend Kalman Filter **Formulation**.
- Compare Data from EKF with Data from EKF-AI models.



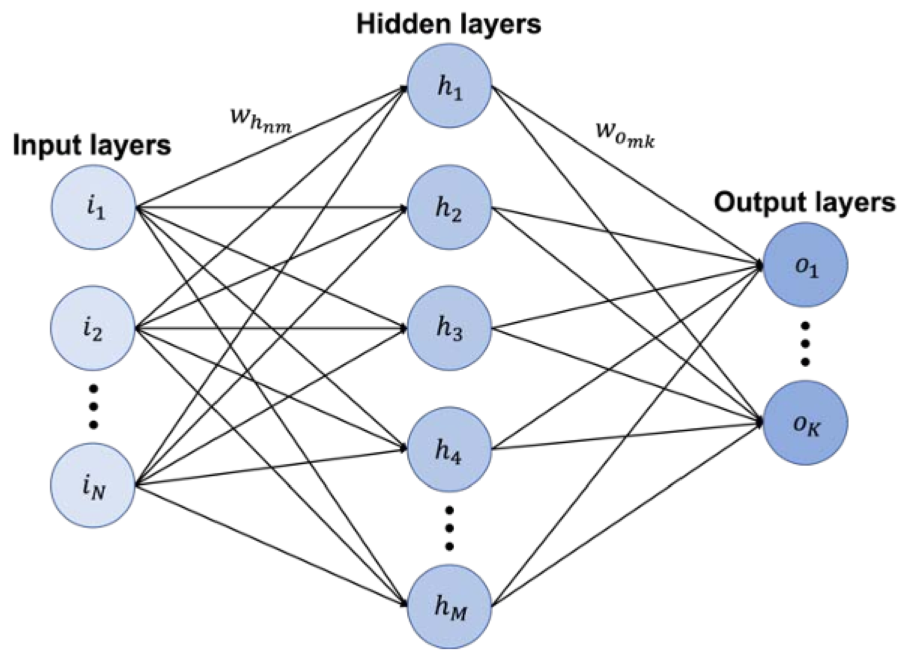
Artificial Intelligent (AI)

Several **AI models** were tested to improve the ***Parameter tuning*** filtering process and enhance **state estimation** accuracy. These models include:

1. **ANN (Artificial Neural Network):** Basic model used to **learn patterns** in noisy signals.
2. **LSTM (Long Short-Term Memory):** Handles time-series data and remembers long-term dependencies.
3. **GRU (Gated Recurrent Unit):** Similar to LSTM but **simpler** and **faster**, used for real-time estimation.
4. **BiLSTM (Bidirectional LSTM):** Reads data **forward** and **backward** for better context understanding.
5. **Fuzzy Logic Model:** Uses fuzzy rules to handle **uncertainty** and **smooth** out noisy signals.

Artificial Intelligent (AI)

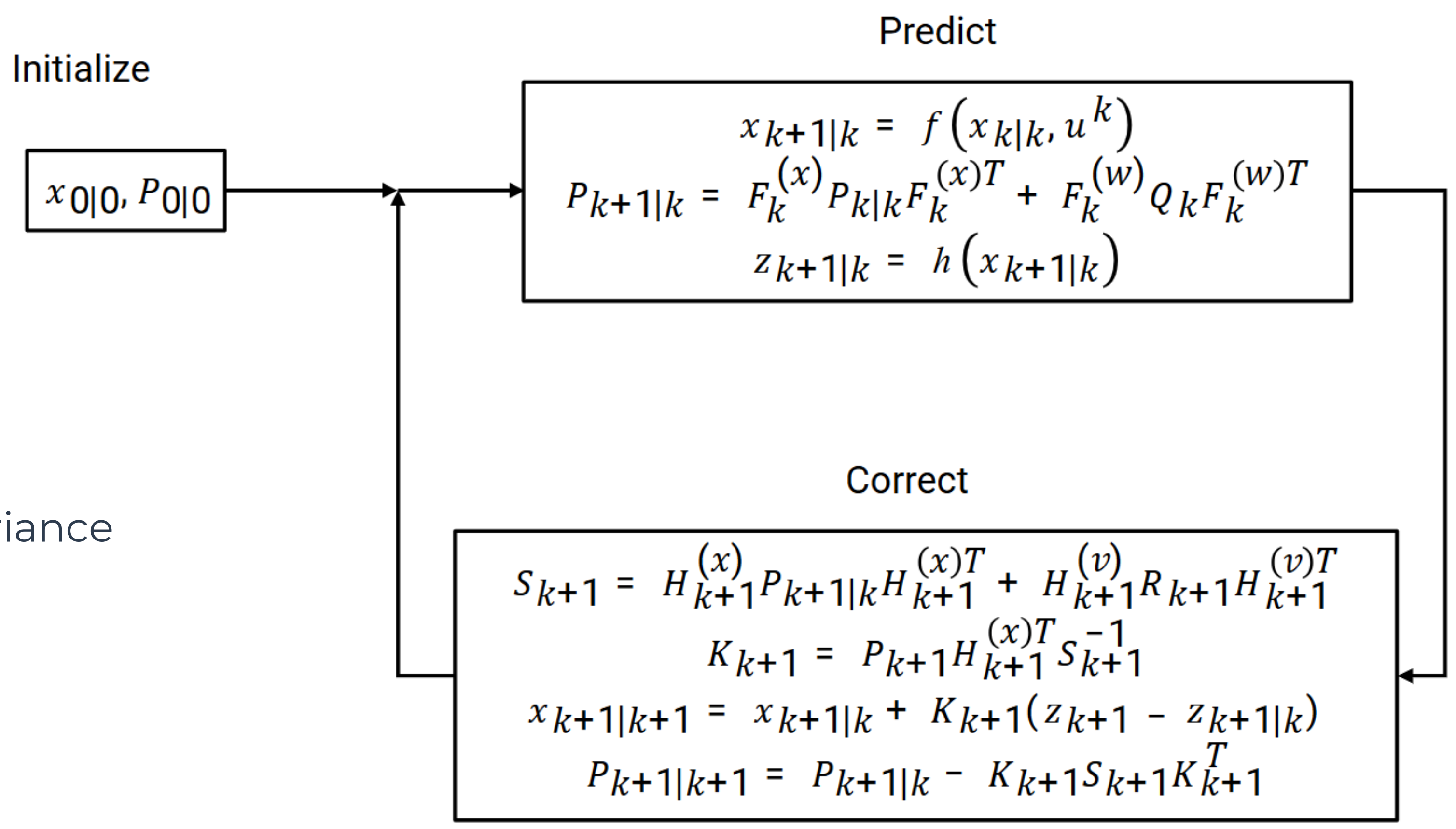
- General AI structure (Input → Hidden → Output).
- Learns patterns and noise.
- Adapts weights using gradient descent.



Extend Kalman Filter

- Recursive estimation method, Prediction & Update steps(correct).

- Equations :



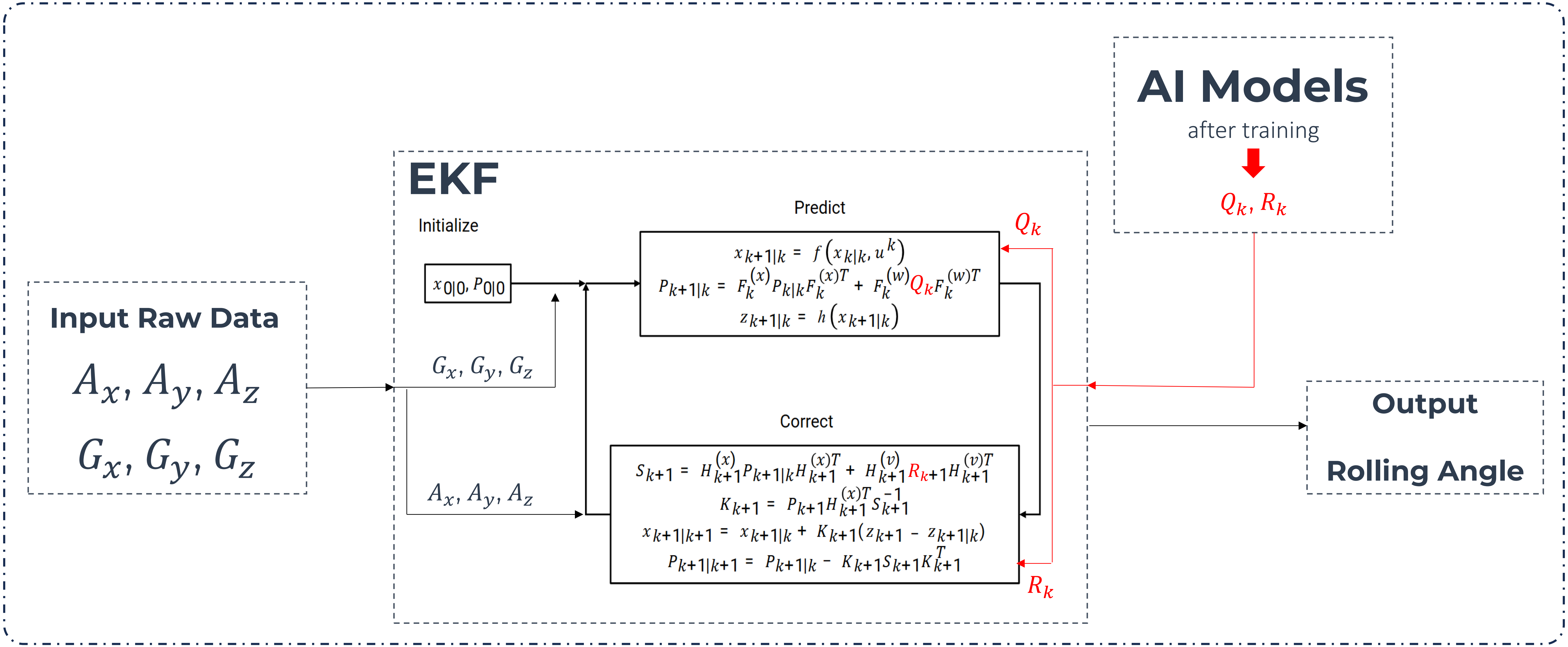
R_K = Measurement Noise Covariance

Q_K = Process Noise Covariance

H_K = Measurement Matrix

F_K = State Transition Matrix

Extend Kalman Filter With AI



A_x, A_y, A_z = Acceleration in Three Axis

G_x, G_y, G_z = gyroscope in Three Axis

Discussion – AI Model-EKF Advantages

- AI-Models improves sensor accuracy.
- Reduces sensor drift (Acceleration), sliding(gyroscope).
- decreases RMSE , MSE compared to standard EKF.
- Adapts to high-noise and dynamic Data.

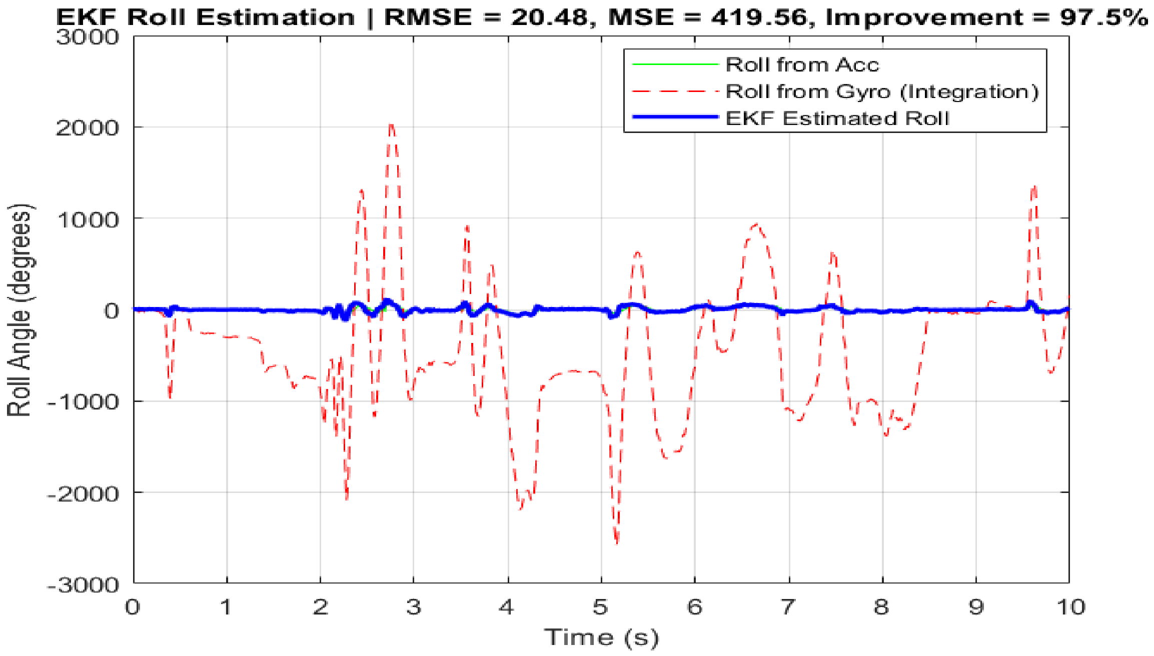
Results.

Model	RMSE (deg)	MSE (deg^2)	Improvement (%)
EKF	406.58	165309.96	51.10%
GRU_EKF	**9.2**	**84.59**	**98.90%**
FUZZY_EKF	15.44	238.27	98.10%
LSTM_EKF	20.48	419.56	97.50%
BILSTM_EKF	66.62	4438.6	92.00%
ANN_EKF	61.43	3773.77	92.60%

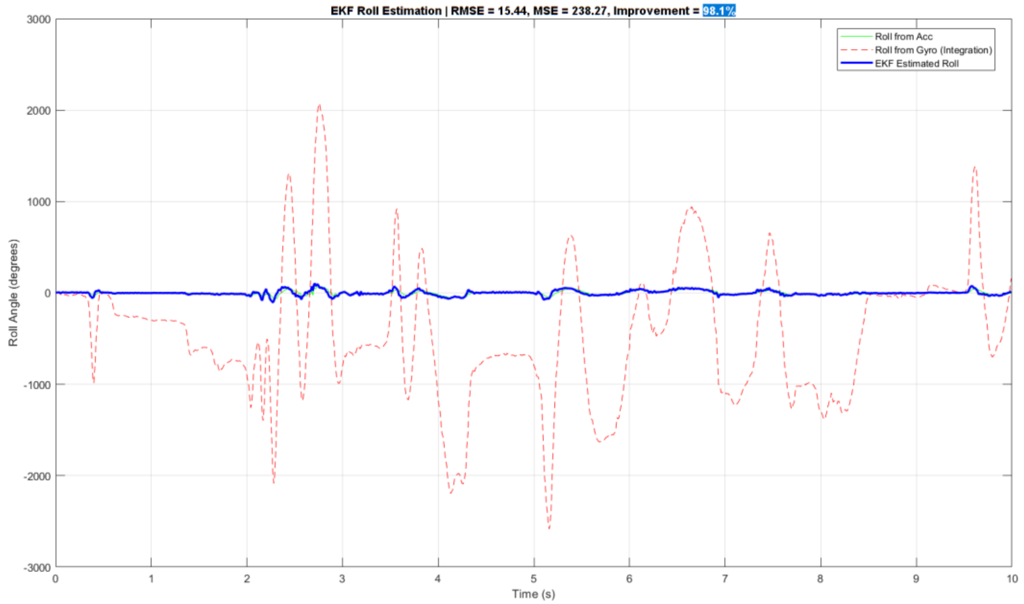
Several AI models were tested to enhance the performance of the Extended Kalman Filter (EKF) when dealing with noisy IMU data. As shown in the table, we evaluated each model using RMSE and MSE. Lower values mean better prediction accuracy.

Results.

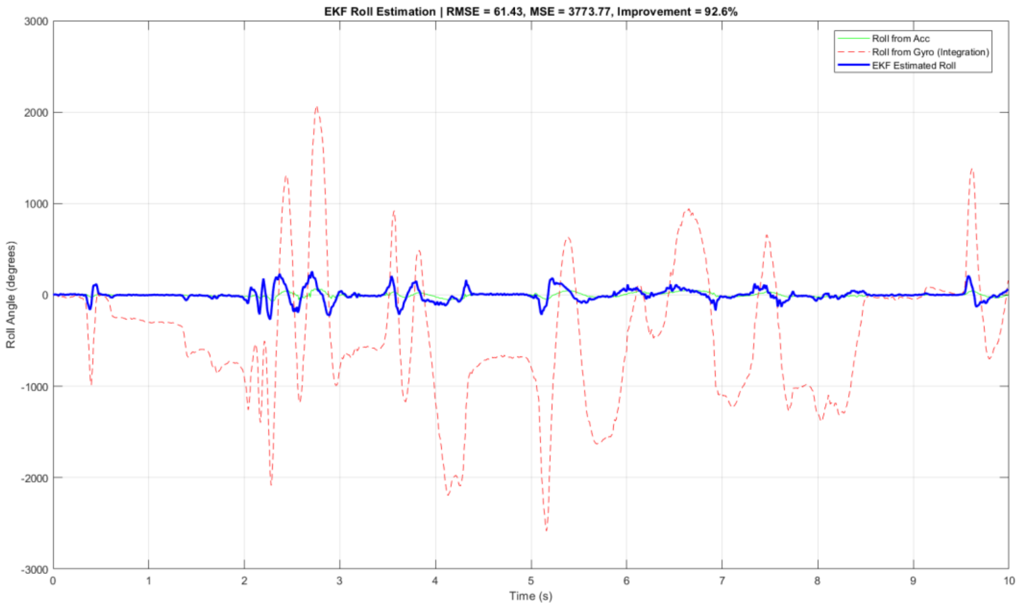
LSTM-EKF



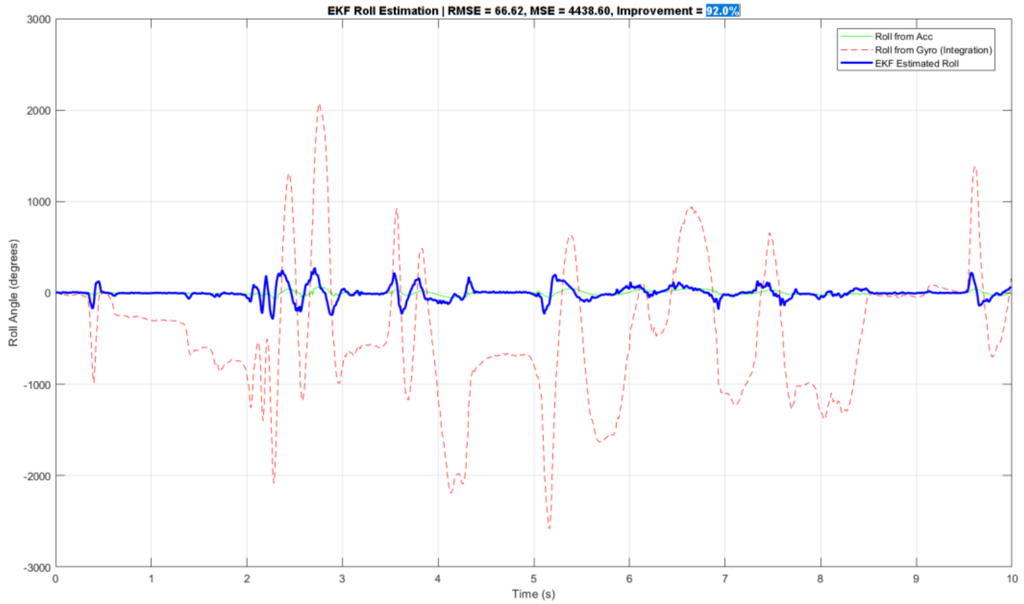
Fuzzy-EKF



ANN-EKF

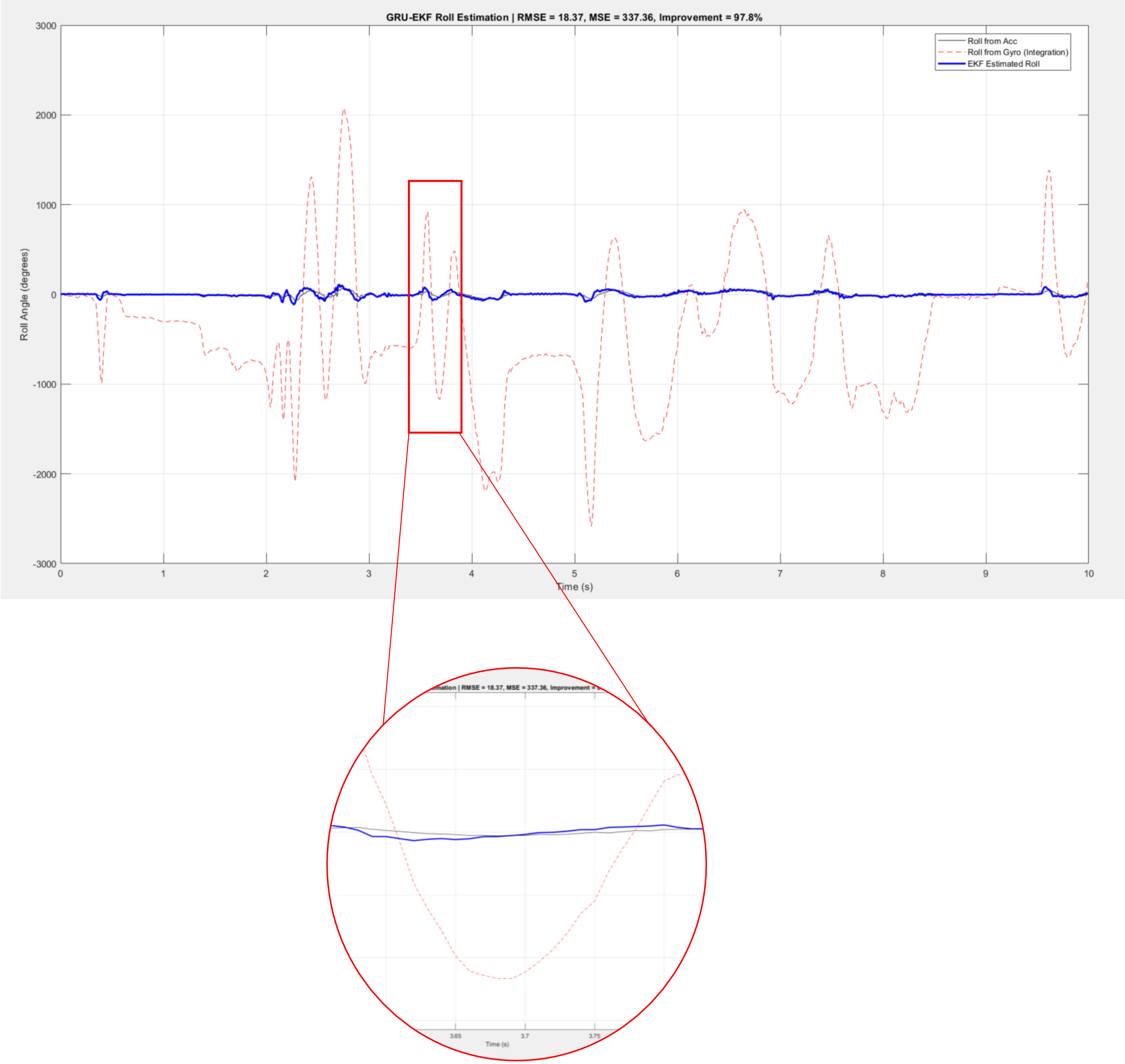


BiLSTM-EKF

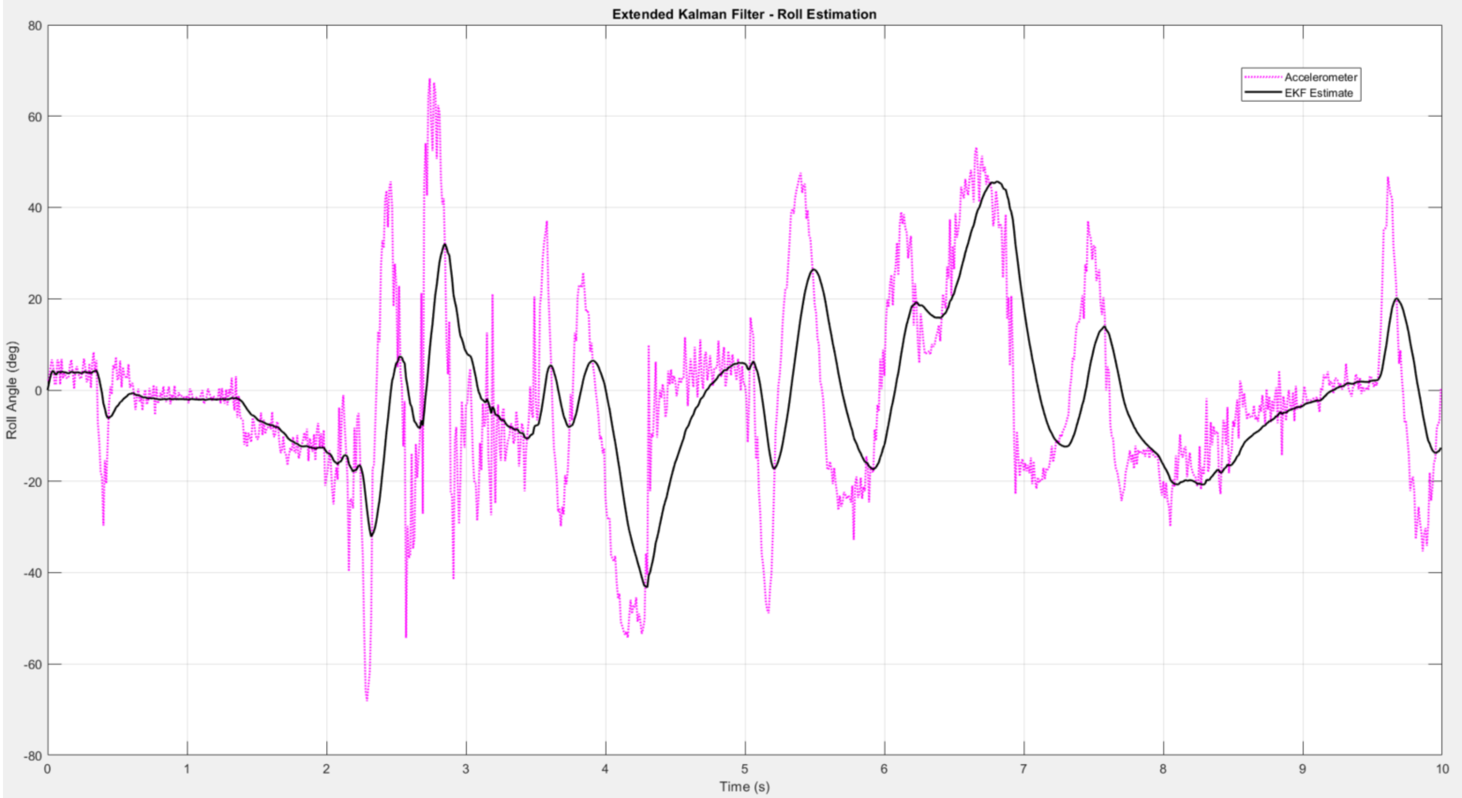


Results.

GRU-EKF



EKF



Conclusion

- GRU + EKF = Better Accuracy & Stability and Less Meas Square Error.
- Adaptive to real-time noise.
- Suitable for UAVs & navigation systems(GPS, IMU).
- The best performance was achieved by GRU_EKF, which significantly reduced the error and achieved 98.9% improvement compared to the standard EKF.
- Other models like FUZZY_EKF, LSTM_EKF, and ANN_EKF also showed strong improvements, but GRU had the lowest error overall.

Future Work & Challenges

- More advanced AI models.
- Reduce **computation cost**.
- Collecting data from a real aircraft

Thank You for **Listening.**