IMPLEMENTATION OF RECURSIVE KALMAN FILTER ON SYNTHETIC IMU DATA

ABSTRACT

This paper describes the application of synthetic Inertial Measurement Unit (IMU) data to attitude control utilizing a Recursive Kalman Filtering technique. In many applications, such as robots, spacecraft, and unmanned aerial vehicles, where exact orientation knowledge is crucial, attitude control is an indispensable component. IMUs are frequently employed to estimate the orientation of these systems, and a common option to increase attitude estimation accuracy is the Kalman Filter. The Recursive Kalman Filtering method discussed in this work is designed to continuously update and refine the estimated attitude of a device based on measurements obtained from any IMU. The IMU provides data related to acceleration and angular velocity, which are processed by the Kalman Filter to estimate the device's orientation in three-dimensional space. The Recursive Kalman Filter iteratively fuses the sensor measurements with a mathematical model of the system, providing an accurate and real-time estimation of the device's attitude.

Keywords: Attitude control, Recursive Kalman Filtering, IMU Data, Inertial Measurement Unit, Orientation estimation, Roll, Yaw, Pitch.

INTRODUCTION

Numerous applications, such as robots, satellites, and unmanned aerial vehicles, demand precise and instantaneous assessment of a device's orientation, or attitude. For a gadget to be stable, navigate, and controllable, it must be able to correctly detect its direction in three dimensions. In response to this pressing requirement, inertial measurement units, or IMUs, have gained popularity as a means of gathering acceleration and angular velocity data that can be used to estimate attitude. In order to provide a reliable and effective method of attitude control, this study applies Recursive Kalman Filtering on data synthetically generated for an IMU. Based on generated sensor readings, the Recursive Kalman Filter is a recursive, state-based estimating technique that constantly updates and improves the estimated attitude of a device.

The purpose of this study is to investigate the theoretical foundations of the Recursive Kalman Filtering technique and how it is used for attitude estimation. It also explores how to include an IMU into the system and covers important topics like sensor calibration and data preparation. The study's experimental findings and performance assessments confirm that Recursive Kalman Filtering is a reliable method for precisely measuring a device's attitude.

The capacity to precisely manage and move objects in robotics, aerospace, and other fields is critical in an ever-evolving technological context. An IMU in conjunction with the Recursive Kalman Filtering technique presents a viable means of accomplishing these objectives. With applications ranging from advanced robotics to autonomous flying, this research advances navigation and control systems by improving the accuracy and real-time capabilities of attitude estimation.

This paper's later sections will go further into the application of Recursive Kalman Filtering for attitude control with an IMU, providing insightful information about the technique's usefulness across a range of applications.

THE INERTIAL MEASUREMENT UNIT

An Inertial Measurement Unit (IMU) is a device that combines multiple sensors to measure and report information about the linear and angular motion of an object. IMUs are widely employed in many different fields, such as robotics, autonomous cars, aerospace, and virtual reality. They are useful instruments for monitoring and comprehending the acceleration, velocity, and direction of an item.

Key components of an IMU typically include:

Accelerometers: These sensors measure linear acceleration in three axes (X, Y, and Z). They detect changes in velocity and are often used to estimate an object's position and velocity.

Gyroscopes: Gyroscopes measure angular velocity (rotation rate) around the same three axes. They are crucial for tracking changes in orientation and rotation.

Magnetometers (optional): Some IMUs include magnetometers to provide information about the object's orientation relative to the Earth's magnetic field. This is especially useful for heading or compass direction information.

IMU systems come in two varieties: six degrees of freedom (6 DoF) and nine degrees of freedom (9 DoF). Accelerometers and gyroscopes, which provide information on linear acceleration and angular velocity, are commonly found in a 6 DoF IMU. With the addition of a magnetometer to provide orientation data, a 9 DoF IMU can estimate an object's whole three-dimensional orientation in space. IMUs are frequently used in conjunction with sensor fusion methods, like complementary or Kalman filtering, to accurately estimate orientation, position, and velocity by combining input from several sensors. These devices are especially useful for applications like motion capture, interior navigation, and drone stabilization where GPS signals could be erratic or non-existent.

In the scope of this paper, the Kalman filter is made to test the Roll, Yaw, and Pitch data of an IMU. Sensor fusion techniques are used to collect roll, pitch, and yaw data, also known as Euler angles or Tait-Bryan angles, from an Inertial Measurement Unit (IMU). X, Y, and Z are the three orthogonal axes in which linear acceleration and rotational velocity are commonly provided by IMUs. To determine the roll, pitch, and yaw angles that characterize the object's orientation in three dimensions, these raw readings must be processed. These angles are obtained as follows:

Roll (ϕ):

- Roll is the rotation about the X-axis. It represents the tilt of an object from side to side.
- To calculate roll, you typically use the following equation:
  Where a_y and a_z are the linear acceleration values in the Y and Z axes, respectively.

$$\phi = \arctan\left(\frac{a_y}{a_z}\right)$$

Pitch (θ):

- Pitch is the rotation about the Y-axis. It represents the tilt of an object from front to back.
- The pitch angle can be calculated as follows:
  Here, a_x, a_y, and a_z are the linear acceleration values in the X, Y, and Z axes

$$\theta = \arctan\left(\frac{-a_x}{\sqrt{a_y^2 + a_z^2}}\right)$$

Yaw (ψ):

- Yaw is the rotation about the Z-axis. It represents the object's heading in the horizontal plane.

- Calculating yaw is more complex and often requires additional sensor data, such as magnetometer readings. Yaw can be found by fusing data from a magnetometer, gyroscopes, and accelerometers using sensor fusion techniques like the complementary filter or the Kalman filter

Similarly, the respective data for Yaw and Pitch can be generated using Python and plotted as follows.

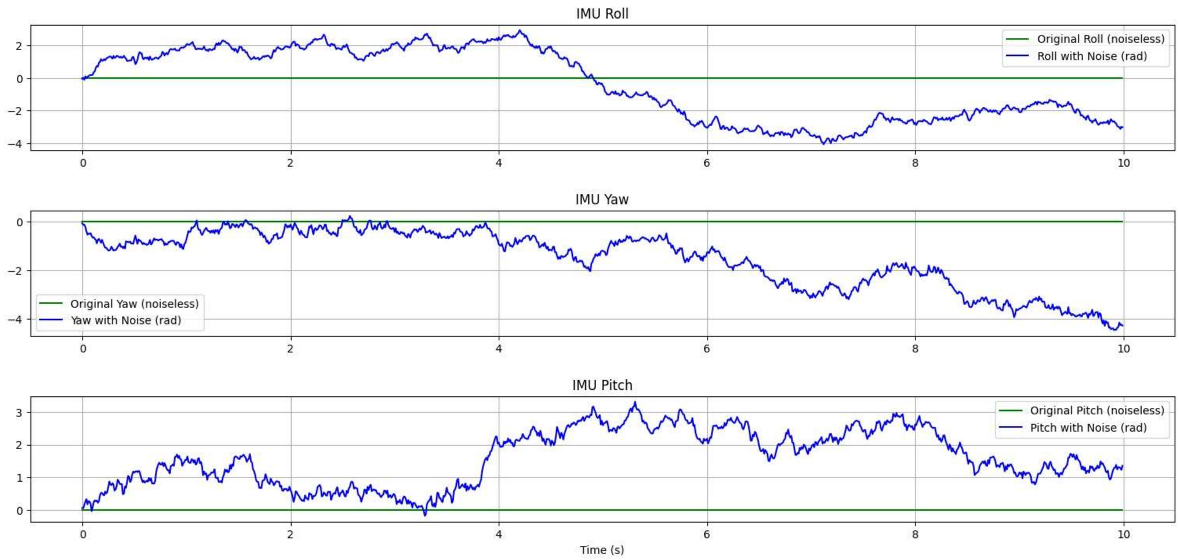

Fig: 1 Roll, Yaw, Pitch Data

THE KALMAN FILTER

To estimate the state of a dynamic system, which may contain variables like position, velocity, and orientation, the Recursive Kalman Filter employs a set of procedures. The Recursive Kalman Filter involves the following essential steps:

1. Initialization:
   - Initialize the initial state estimate (usually a vector) and the associated error covariance matrix.
2. Prediction:
   - Predict the system's state at the next time step based on the known system dynamics.
   - Update the error covariance matrix to account for the prediction error.
3. Update (or Correction):
   - Obtain a measurement of the system's state from sensors. These measurements are often noisy and contain errors.
   - Calculate the Kalman Gain, which determines the weight given to the prediction and measurement in the state update.
   - Update the state estimate using a weighted combination of the predicted state and the measurement, resulting in a more accurate state estimate.
   - Update the error covariance matrix to reflect the reduced uncertainty in the state estimate.
4. Return to Prediction:

- The updated state estimate and error covariance matrix are used as the new initial conditions for the next iteration.

Finally, the data is made more accurate by curve smoothing, also known as data smoothing, which is a data analysis technique used to remove noise and irregularities from a dataset, resulting in a cleaner and more interpretable representation of the underlying trend or pattern. It is commonly applied to time series data, where the goal is to reduce the impact of random variations, measurement errors, and outliers while retaining the essential characteristics of the data.

SIMULATION AND RESULTS

Simulating Inertial Measurement Unit (IMU) data in Python typically involves generating synthetic sensor readings for acceleration, angular velocity, and, optionally, magnetic field strength to mimic the behaviour of a real IMU. Real IMUs provide more complex and noisy data, and simulating realistic sensor noise and biases is an essential part of creating accurate simulations. Additionally, apply sensor fusion techniques, such as the Kalman filter, to integrate the simulated sensor data and estimate orientation and motion.

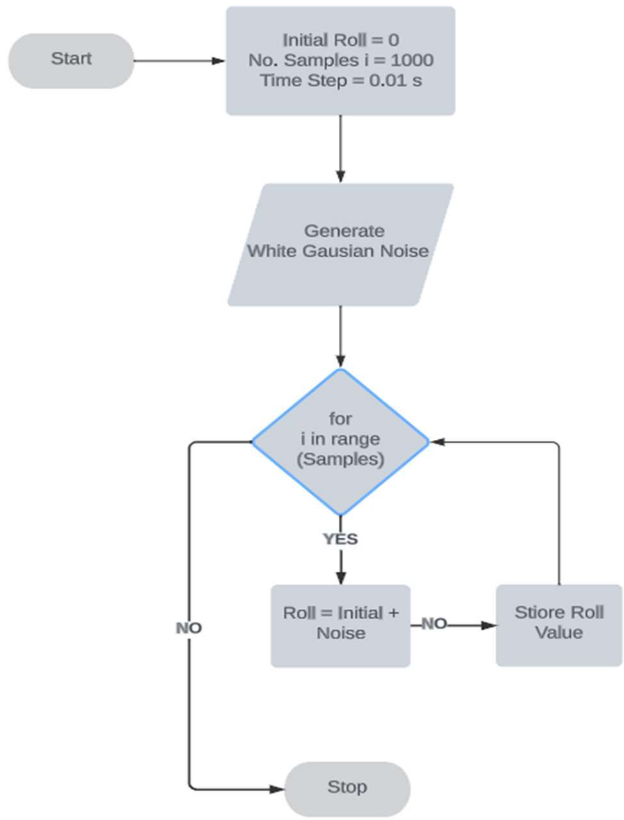

Fig: 2 Flowchart for Synthetic IMU data generation

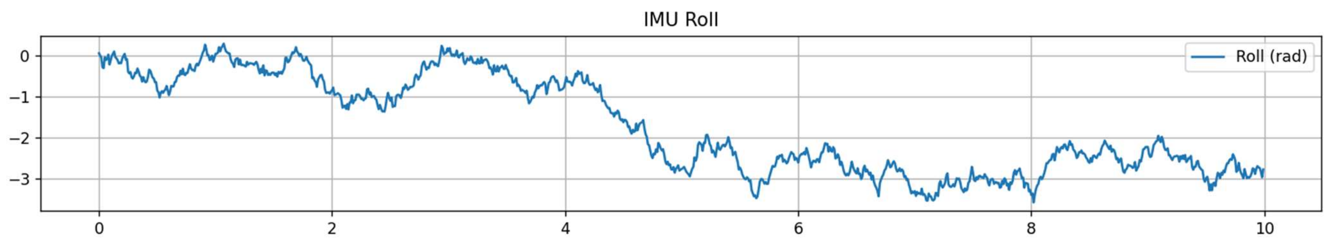

Fig: 3 Sample data with noise

Implementation of Kalman Filter

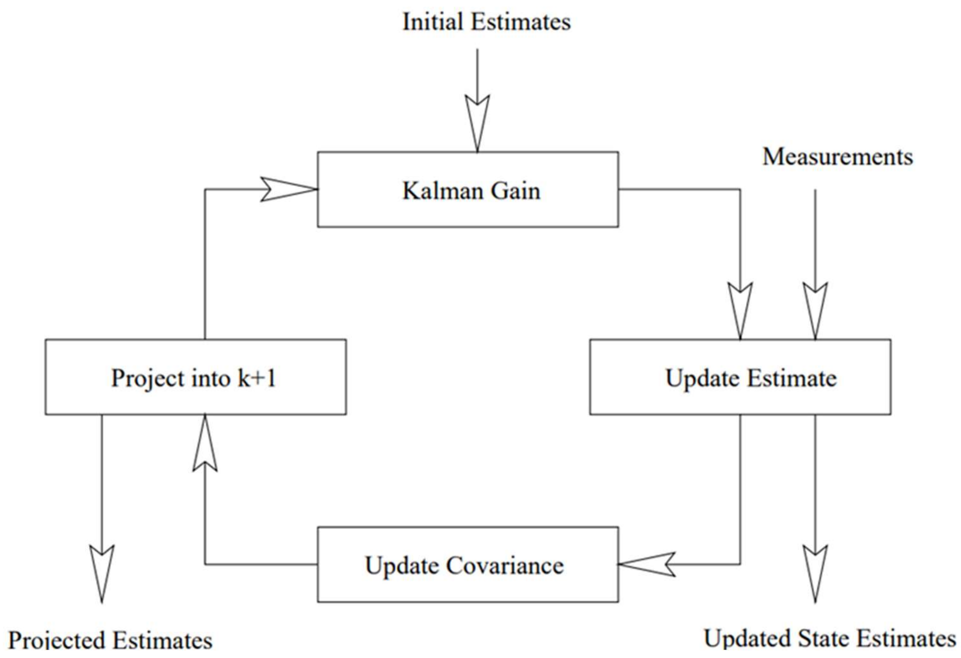

Fig: 4 Recursive Kalman Filter Flowchart

Variables:

1. **state_estimate**: The estimated state value at a given time step.
2. **state_covariance**: The covariance of the estimated state.

Constants:

1. **process_noise**: The process noise (Q) represents the system's uncertainty or error in the prediction step.
2. **measurement_noise**: The measurement noise (R) represents the uncertainty or error in the measurement step.

Matrices:

1. **F**: The state transition matrix represents the dynamics of the system. In this code, it is assumed to be a constant velocity model matrix.
2. **H**: The measurement matrix indicates what aspect of the state is being measured. In this code, it represents measuring the roll directly.

These variables and constants are crucial components of the Kalman filter, where the filter updates the state estimate based on the prediction step, the measurement step, and the associated uncertainties (process noise and measurement noise). The state estimate and state covariance are continually updated as new measurements become available, providing a filtered estimate of the system's state.

RESULT: The sample data, with Kalman filter has been plotted in Fig. 5

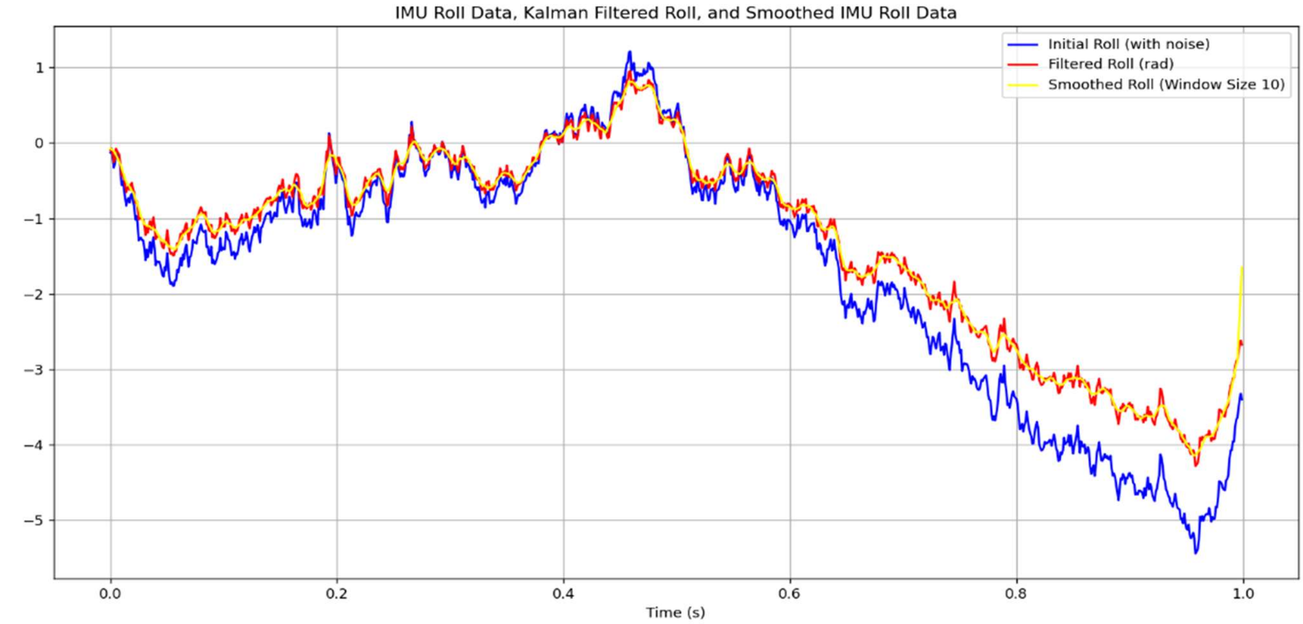

Fig: 5 Simulation Result with smoothed curve.

CONCLUSION

In this work, we investigated the use of an Inertial Measurement Unit (IMU) data to a Recursive Kalman filter for attitude control. This study aimed to improve attitude estimate accuracy and reliability for robotics, aerospace systems, and autonomous navigation applications. It has been shown that the Recursive Kalman filter improves attitude estimation. The filter reduces the problems caused by bias and noise in the sensors by combining data from the gyroscope and accelerometer sensors of the IMU to provide a more reliable and accurate solution. It is important to note that while the Recursive Kalman filter significantly enhances attitude estimation, it is not without its challenges. The accuracy of the filter is contingent on accurate sensor calibration, precise modelling of sensor noise, and the system's dynamic behaviour. Ongoing research and development are needed to address these challenges and further improve the filter's performance.

REFERENCES

1. **Candan, Batu. (2022). Design of Attitude Estimation Algorithms for Inertial Sensors only Measurement Scenarios.**
2. **Makni, Aida & Fourati, Hassen & Kibangou, Alain. (2014). Adaptive Kalman Filter for MEMS-IMU based Attitude Estimation under External Acceleration and Parsimonious use of Gyroscopes. 1379-1384. 10.1109/ECC.2014.6862535.**
3. **M. Pelka and H. Hellbrück, "Introduction, discussion and evaluation of recursive Bayesian filters for linear and nonlinear filtering problems in indoor localization," 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Alcala de Henares, Spain, 2016, pp. 1-8, doi: 10.1109/IPIN.2016.7743663.**
4. **Roberto S. Inoue , Marco H. Terra "Robust Recursive Kalman Filtering for Attitude Estimation" Department of Electrical Engineering, University of São Paulo at São Carlos, São Paulo, Brazil 10.3182/20110828-6-IT-1002.03677**