# 

**FInal Project Report**

**Instructor: Dr. Naser El-Sheimy  
Date: 27th April, 2023**

**Naman Agarwal**

**UCID: 30138256**

**ENGO 623**

Contents

[1](#_Toc133479243)

[Introduction 2](#_Toc133479244)

[Mechanization Module Implementation 3](#_Toc133479245)

[Measurements Correction 4](#_Toc133479246)

[Initial Alignment 5](#_Toc133479247)

[Mechanization Equations in Local-Level NED Frame 6](#_Toc133479248)

[Attitude Update: 6](#_Toc133479249)

[Specific-Force Frame Transformation: 6](#_Toc133479250)

[Velocity Update: 6](#_Toc133479251)

[Position Update: 7](#_Toc133479252)

[Results and Discussion 7](#_Toc133479253)

[Error-State Kalman Filter 14](#_Toc133479254)

[Error-State Dynamic Equations 14](#_Toc133479255)

[Results and Discussion 17](#_Toc133479256)

[Conclusion And Future Scope 24](#_Toc133479257)

[Bibliography 25](#_Toc133479258)

# 

# Introduction

This Report will present the results of implementing (i) an INS mechanization module in the local-level frame (LLF) and (ii) Mechanization integrated with an Error-State Kalman Filter to emulate Loosely-Coupled INS/GNSS system or simultaneous ZUPT and CUPT. The first section will cover a step-by-step description of the mechanization equations implementation, which have been primarily referenced from [1]. It will also include the position and error results/plots obtained using the implemented mechanization module. The next section will then detail the working of implemented Error-State Kalman Filter and exhibit the results/plots obtained from it. At the end conclusions regarding the obtained results are discussed.

Table : Summary of the dataset, the implemented mechanization process and Error-State Kalman Filter

|  |  |
| --- | --- |
| Item | Description |
| Dataset | Static/Stationary dataset obtained after leaving an IMU running without any motion for a period of time. |
| Data Rate | 64 Hz |
| Duration | 1000 seconds |
| Mechanization frame | Local-Level Frame(LLF), where LLF is described in North-East-Down(NED) direction. |
| Attitude representation | Direction Cosine Matrix(DCM) |
| Modelled IMU noise parameters | Random Walk Errors e.g. ARW & VRW +  Bias Instability( modelled as 1st order GM process) |
| IMU noise parameters value | |  |  |  | | --- | --- | --- | | **Parameter** | **Accelerometer** | **Gyroscope** | | Bias(Residual + Turn-On) |  | 0.01 | | Bias Instability/In-Run Bias | *50* | 0.015 | | Correlation Time | 1 hour | 1 hour | | VRW/ARW | 0.003 | 0.01 | |
| Frequency of Measurement Update in the Error State Kalman Filter | Approx 1 minute |

# 

# Mechanization Module Implementation

The Mechanization module is designed referencing [1], therefore instead of using ENU fame and Quaternion attitude representation which is followed in the [2]. NED frame along with DCM attitude representation is adopted. Mechanization is the process of converting the output of an IMU into position, velocity, and attitude information[3]. The outputs include rotation rates about three body axes measured by the gyroscopes triad and three specific forces along the body axes measured by the accelerometer triad, all of which are with respect to the inertial frame. Mechanization is a recursive process that starts with a specified set of initial values and iterates on the output. A general diagram of INS mechanization is shown in Figure 1.

Transformation into Navigational Reference frame

Attitude Computation

Accelerometers

Gyroscope

Platform Linear Acceleration

Platform Rotation

Correction

Gravity Model

Coriolis Correction

Attitude

Position

IMU

Figure 1: A block diagram depicting the mechanization process of an INS (SOURCE ADAPTED FROM [3] )

Inertial Measurement Unit

Velocity

The following list summarizes the general steps of the LLF mechanization

1. Initialize the INS with the initial position, velocity, and attitude estimates.
2. Perform initial static alignment of the platform to determine the initial attitude.
3. Apply corrections to the raw measurements to remove bias and scale factor errors.
4. Determine the specific force and angular rate measurements in the LLF frame, by transforming them from Body-Frame.
5. Update the attitude estimate using the angular rate measurements.
6. Apply gravity correction and compensate for Earth's rotation to determine the acceleration in the LLF frame.
7. Double integrate the acceleration to estimate velocity and position in the ECEF frame.
8. Update the INS output at the measurement rate by repeating steps 4-8 for each new set of measurements.
9. Optional: Perform sensor fusion with additional sensors, such as GNSS, to improve the INS accuracy.

The following subsections describe the mechanization steps in more detail.

## Measurements Correction

The general form of the measurement correction is given by the following equations

Where , The gyroscope and accelerometer corrected the readings

Three-by-Three identity matrix

,The gyroscope and accelerometer readings

*Sg, Sa* The gyroscope and accelerometer scale factor

*Ng, Na* The gyroscope and accelerometer non-orthogonality matrix

*d, b* The gyroscope drift and the accelerometer bias

For this report, the bias and drift terms only are corrected. The scale factor and the non-orthogonality matrix are considered error-free.

## Initial Alignment

The Initial alignment technique for the project has been referred and adopted from [4]. The technique performs accelerometer leveling and gyro compassing in a single step to compute the initial attitude in the form of a direction cosine matrix(DCM). The Rotation matrix/DCM converts the body frame to a Local-Level NED frame.   
  
The initial alignment process performs averaging of the first few epochs of IMU data when the IMU device is considered to be stationary or static and uses the averaged value(,) to compute the attitude. For this project first 65 seconds of data is averaged.

Following is the methodology to perform the initial alignment to obtain the initial attitude .

If is defined as , we also have

Since = = , these three vector relations can be written:

Where, & L = latitude

## Mechanization Equations in Local-Level NED Frame

The local-level NED frame mechanization equations are adopted from [1]. In the local-navigation-frame implementation of the inertial navigation equations, the ECEF frame is used as the reference frame, while the local navigation frame (north, east, down) comprises the resolving axes. Thus, attitude is expressed as the body-to-navigation-frame coordinate transformation matrix, , and velocity is Earth-referenced in local navigation frame axes, . Position is expressed in the curvilinear form (i.e., as geodetic latitude, , longitude, , and geodetic height, ) and is commonly integrated directly from the velocity rather than converted from its Cartesian form.  
  
A brief summary of the four steps required in this mechanization is presented below:

### Attitude Update:

*Please note the following symbols: and represents Skew-Symmetric Matrix and time-interval between consecutive IMU measurements.* *And (+) and (-) represents current/new and previous estimate.*

Where, ,

It is important to preserve the orthonormal nature of the DCM or rotation matrix and to ensure that reorthogonalization and renormalization need to be performed at regular intervals. Intuitively it is exactly the same as Quaternion normalization technique discussed in [2]. Refer [1] to learn more about DCM re-ortho-normalization technique.

### Specific-Force Frame Transformation:

### Velocity Update:

=

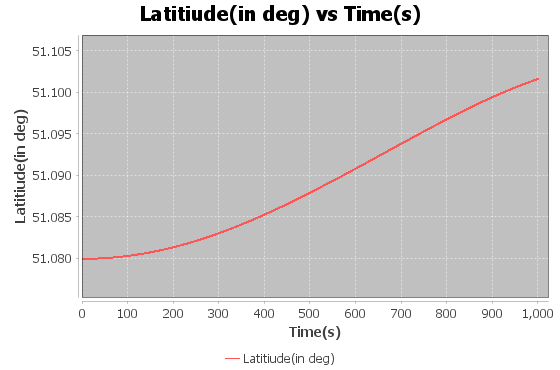
Where is the acceleration due to gravity, modeled as a function of latitude and

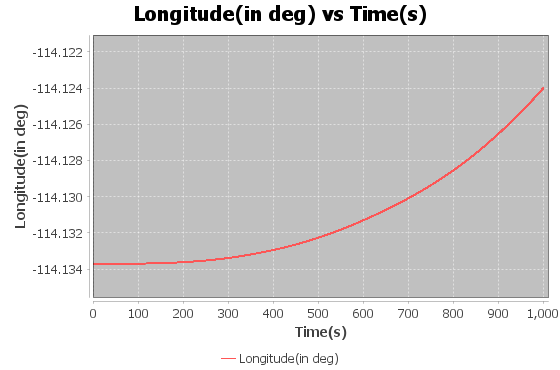
Height in the local level frame (NED).

### Position Update:

## Results and Discussion

The following graphs/plots show the results obtained from the implemented Mechanized module.   
  
**Please note the attitude errors roll, pitch, and yaw are computed in the NED frame, therefore follow the NED frame sign convention.**





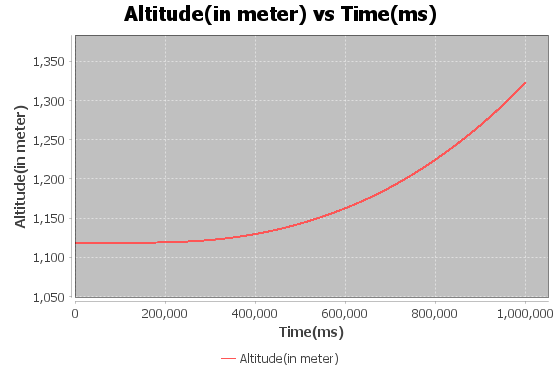
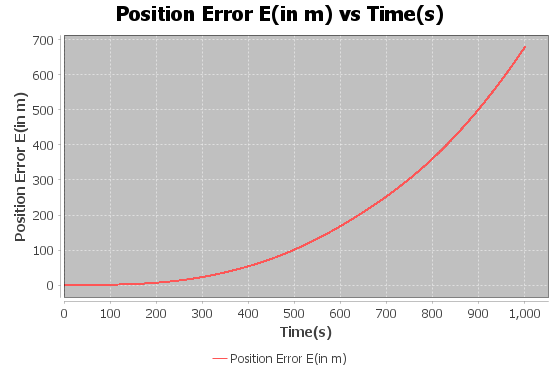
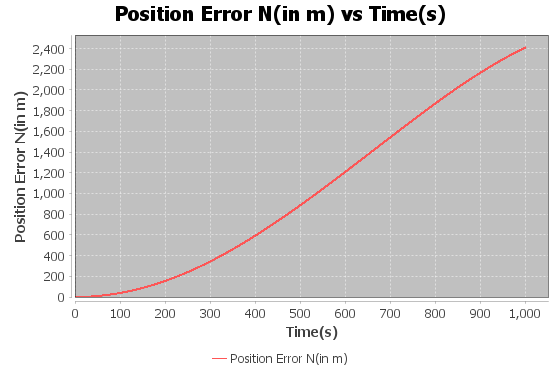


Figure 2: Calculated Position





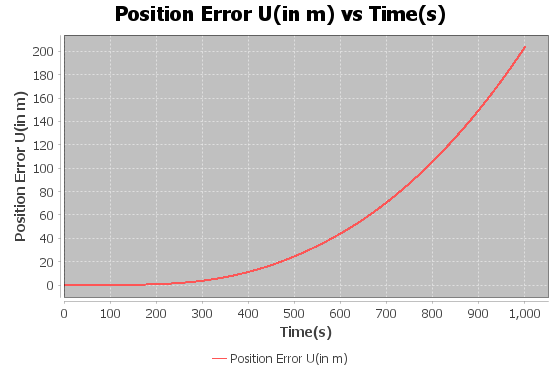
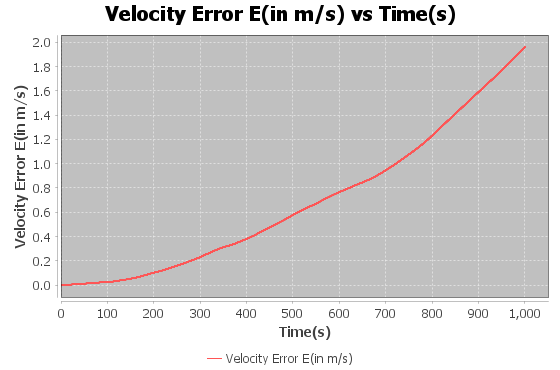
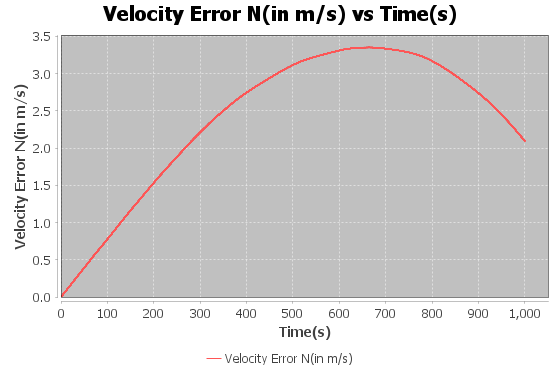


Figure 3: Position Errors





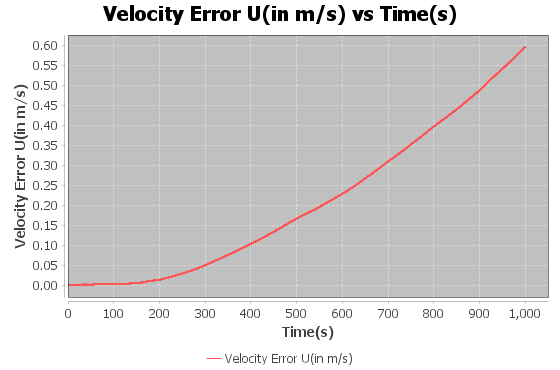
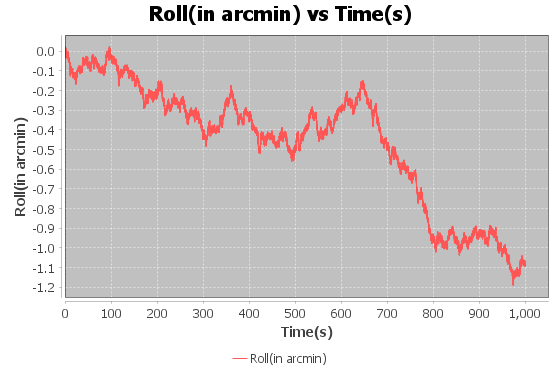
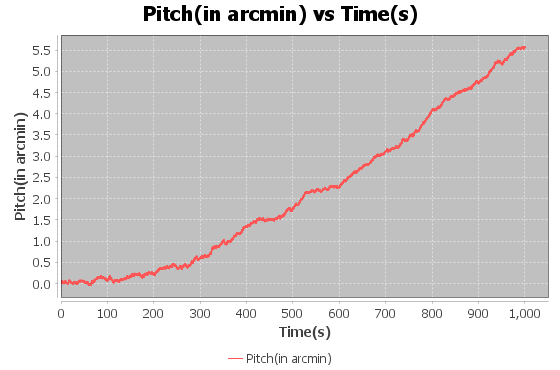


Figure 4: Velocity Errors





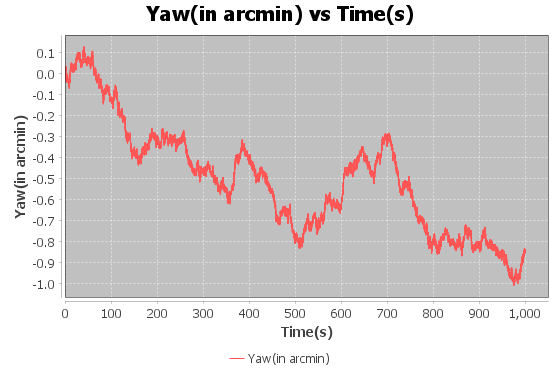


Figure 5: Attitude Error

Table 2: Estimate of position and errors obtained from the Mechanization module at the final epoch

|  |  |  |  |
| --- | --- | --- | --- |
| Position( Lat, Lon and Alt) | 51.10161 degree | -114.12400 degree | 1322.699 meter |
| Position Error in ENU | 679.680 meter | 2410.116 meter | 203.7061 meter |
| Velocity Error in ENU | 1.962 meter/sec | 2.094 meter/sec | 0.595 meter/sec |
| Attitude Error(Roll, Pitch and Yaw) | -1.092 arcmin | 5.560 arcmin | -0.853 arcmin |

It is evident from observing Figures 2, 3, 4, 5, and Table 1 that without any external aiding the position, velocity, and attitude errors continue to drift and deteriorate.   
  
The position and velocity error is the biggest in the North(N) direction, which can mean that the Rotation matrix computed is unable to properly transform body frame to LLF frame. This causes a horizontal leveling error that leads to the wrongful introduction of some portion of the gravity component in North-Frame. Since North Velocity error is tightly coupled with Pitch error due to Schuler effect, the estimation accuracy of one is interdependent on the other. The attitude error results obtained from the experiment confirm the aforementioned hypothesis, as the Pitch error is the biggest out of all the attitude errors.

Apart from slow constant drift, the large inaccuracy which is observed in Horizontal axes, especially N-axis can be primarily due to imperfect coarse/static alignment performed at the beginning to compute initial attitude. Even a misalignment of a few arcminutes can lead to accumulation of large errors as time passes. Therefore, Accelerometer leveling/Initial alignment needs to be accurate and robust.

Overall the drift observed in all the position and error plots can be credited to various uncalibrated deterministic and stochastic noise components in the inertial sensor e.g. White noise(ARW/VRW), Bias Instability, Scale Factor, Non-Orthogonality, Turn-on Bias, etc. To eliminate stochastic noise components, online estimation is required which includes introducing extra states for error estimation in the estimation filter. External aiding via GNSS, Odometer, Magnetometer, CUPTs, ZUPTs, etc greatly improves the error estimation capability of the filter. Even techniques which utilizes certain constraints like non-holonomic constraints on land vehicle, or height constraints using barometer can improve the inertial sensor's performance

# Error-State Kalman Filter

## Error-State Dynamic Equations

The theory and mathematics concerning the design and implementation of Error-State KF is referred from [5]. Similar to Mechanization module discussed above the Error State Dynamic equations are also derived in NED aligned LLF frame. The first order differential equation() of the 15- state error model is provided below, the notation used in the equations are defined in the Figure 6.

+

( 1 )

Where,

is the correlation time of the bias instability noise param, which is modelled as 1st order Gaussian Markov(GM) process

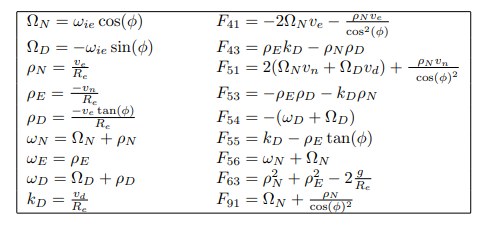


Figure : Definition of Notation for INS error equations (SOURCE ADAPTED FROM [5])

In the implemented error-state KF, only Random walk Errors and Bias Instabilities are considered and modelled as part of IMU sensor's stochastic errors. Therefore andare both identity or unity and and are inverse of bias instability parameter’s correlation time.

The IMU noise parameter values are provided in Table 1. To suitably model and translate the noise parameters in the state model, [6] was taken as a reference. The Residual or Turn-On bias is used to initialize the bias parameters and .

From the first-order differential equation ( 1 ), the dynamics matrix F and matrix G can be utilized to derive transition matrix and process covariance matrix . Since the parameters in and are variable and matrices themselves are large and complex, the numerical methods should be utilized instead of analytical methods to estimate the required matrices. C. F. van Loan[7], [8] method is therefore used in the project.

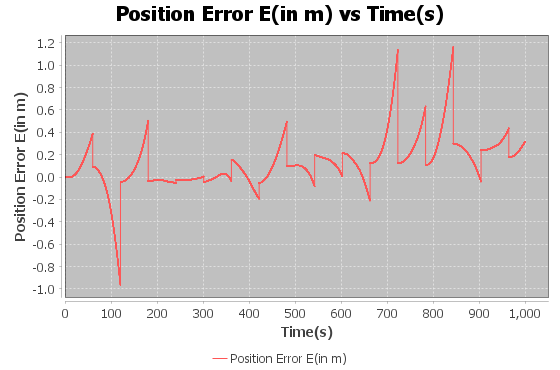
For the measurement update step of the Kalman filter, as the device is stationary, the known position estimate along with zero velocity vector is provided each time at the update step. This manner of update, resembles Loosely-Coupled(LC) INS/GNSS integration or can also be viewed as performing Zero-Velocity update(ZUPT) and Coordinate Update(CUPT) simultaneously. But in reality, though the proposed update step method shares similarities with the two mentioned procedures, it cannot be strictly categorized under either of these terms because the GNSS system was never utilized to estimate the position solution.

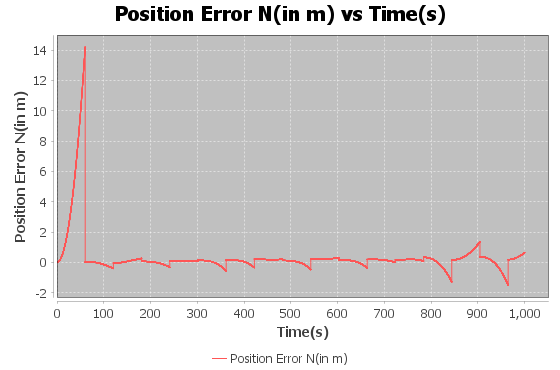
The following are the *Z* and *H* matrix, used as part of the measurement equation (

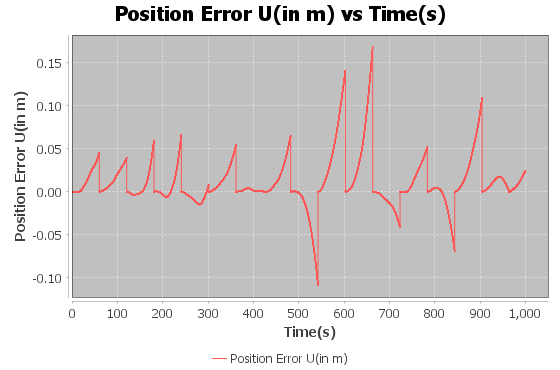
The update is performed approximately after every one minute. Since the Kalman Filter is implemented as closed loop, after the update step, the estimated error states are added to the total states/INS navigation solution which are being parallelly generated and propagated by Mechanization module. After correcting the total states, and improving the accuracy of the INS solution, the error estimates are reset to zero.

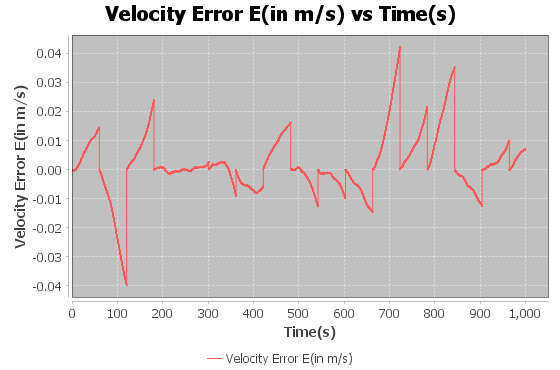
## Results and Discussion

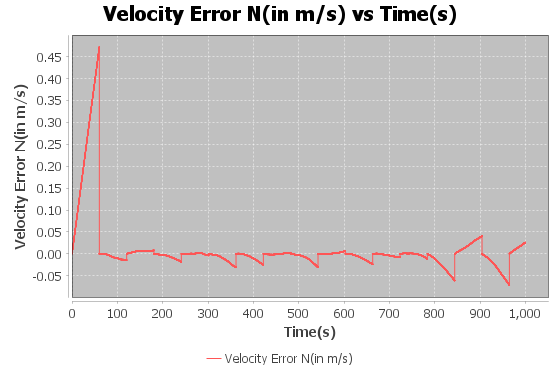
The following graphs/plots show the results obtained from implementing a Mechanized module integrated with an Error-State filter which periodically(~ every minute) performs an update step and corrects/adjusts the INS state estimates   
  
**Please note the attitude errors roll, pitch, and yaw are computed in the NED frame, therefore follow the NED frame sign convention.**

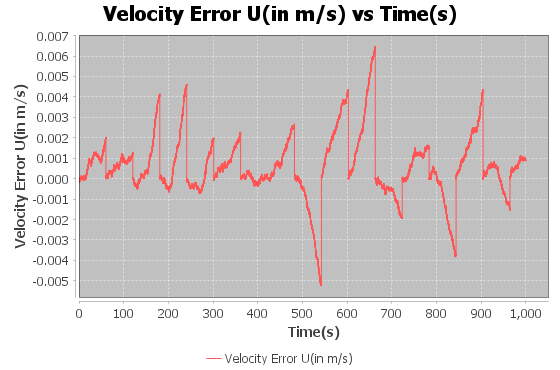


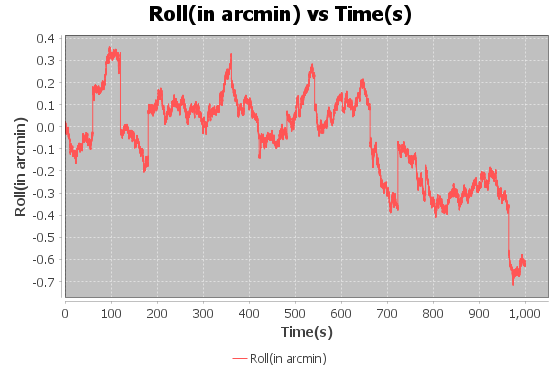


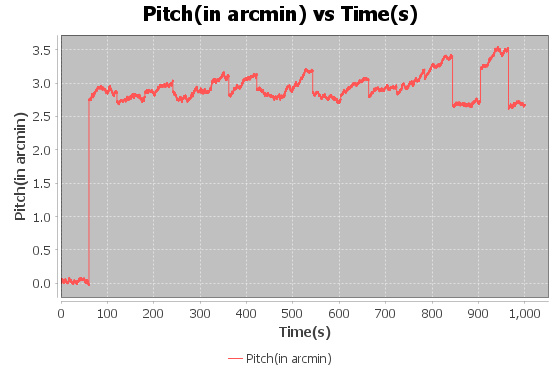


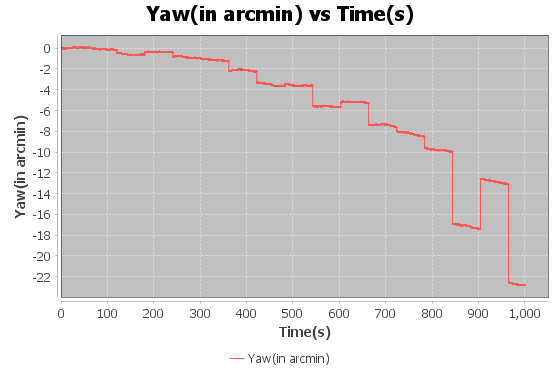


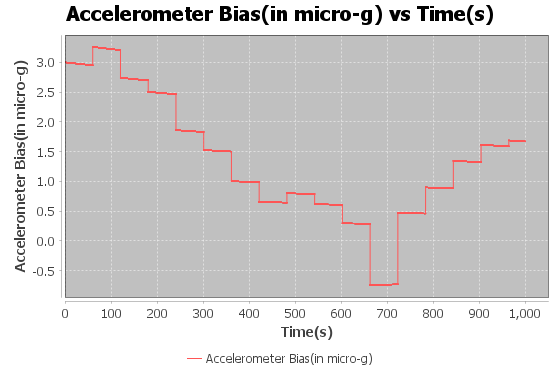












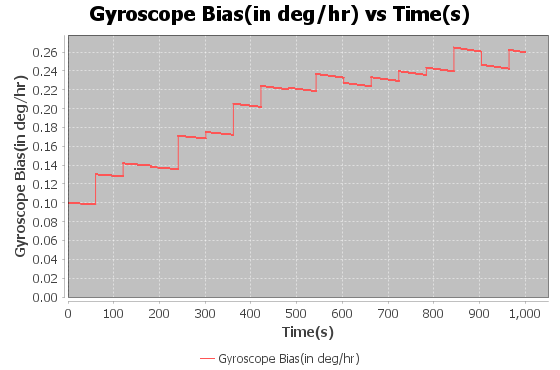


Table : Estimate of position and errors obtained from the Integrated Method at the final epoch

|  |  |  |  |
| --- | --- | --- | --- |
| Position( Lat, Lon and Alt) | 51.07995 degree | 114.13370 degree | 1118.52600 meter |
| Position Error in ENU | 0.31532 meter | 0.64537 meter | 0.02400 meter |
| Velocity Error in ENU | 0.00702 m/s | 0.02592 m/s | 8.9726e-4 m/s |
| Attitude Error(Roll, Pitch and Yaw) | -0.6321 arcmin | 2.6662 arcmin | -22.8204 arcmin |

The position and error plots and results obtained from the Error-State KF integrated method, expectedly provide a considerably better solution. It can be observed from the position and velocity error plots that the error values are corrected after each measurement update. Furthermore, a comparison between Table 2 and Table 3 demonstrates that the integrated Error-State KF method is highly effective in limiting error drifts.

As for the attitude errors, the Yaw/Azimuth error is relatively large at the final epoch for the Integrated Error-State KF compared to only Mechanization based solution. The reason probably is that Initial-Alignment, and specifically gyrocompassing produces erroneous Yaw estimates except for highly accurate/ high-end IMUs. And as a result, the Yaw angle estimated at initialization was incorrect and the final solution for the integrated method differs from it and leans more towards the true Yaw angle value.

It is important to note that in the final implementation of the project, all stochastic sensor noise parameters, except for correlation time and turn-on bias, were scaled by a factor of more than 10 (a factor of 20 was used for the results presented). Failing to scale these parameters can lead to failure, as the Kalman Filter may drift away and produce incomprehensible results. This is likely due to the underestimated noise parameter values provided in the manufacturer's specification sheet.

# Conclusion And Future Scope

The online estimation of error states using a Kalman Filter and their incorporation into the INS navigation solution can significantly enhance the accuracy of the system, as discussed in the Results and Discussion section of the Mechanization Module. This was verified through successful implementation of a similar filter scheme. Data recorded by the Stationary IMU device, which was used in this project served as a great learning source to develop and test Mechanization process as well as Error-State Kalman Filtering. The algorithm and program designed in this exercise can be used to test and compare the performance of other common IMU devices e.g. Smartphones, Wearables, etc. Additionally, by incorporating magnetometer data from other devices, one can improve the estimation of the Azimuth/Yaw angle, which is the least observable state during both initial alignment and rest of the device run, given the current setup/sensors used in this project.

*NOTE: The source code for the Java-based project created to obtain the aforementioned results can also be found on the Bitbucket – (link:* [*https://bitbucket.org/naman4u13/engo623final/*](https://bitbucket.org/naman4u13/engo623final/)*). User only needs to modify the output file path, before compiling the code to output the solution.*

# Bibliography

[1] P. D. Groves, “Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Second Edition.” https://ieeexplore.ieee.org/document/9101092 (accessed Apr. 17, 2023).

[2] N. El-Sheimy, “Lecture Notes for ENGO 623 - Inertial Suveying and INS/GPS Integration.” Jan. 2023.

[3] A. Noureldin, T. B. Karamat, and J. Georgy, *Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration*. Berlin, Heidelberg: Springer, 2013. doi: 10.1007/978-3-642-30466-8.

[4] K. R. Britting, *Inertial Navigation Systems Analysis*. Wiley-Interscience, 1971.

[5] J. A. Farrell, *Aided Navigation: GPS with High Rate Sensors*. McGraw Hill Professional, 2008.

[6] J. A. Farrell, F. O. Silva, F. Rahman, and J. Wendel, “IMU Error Modeling Tutorial: INS state estimation with real-time sensor calibration,” Oct. 2021, Accessed: Apr. 27, 2023. [Online]. Available: https://escholarship.org/uc/item/1vf7j52p

[7] C. Van Loan, “Computing integrals involving the matrix exponential,” *IEEE Trans. Autom. Control*, vol. 23, no. 3, pp. 395–404, Jun. 1978, doi: 10.1109/TAC.1978.1101743.

[8] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises*, 4th edition. Hoboken, NJ: Wiley, 2012.