Yaw/Heading optimization by Machine learning model based on MEMS magnetometer under harsh conditions

Available online at www.sciencedirect.com

Measurement

Journal homepage: www.elsevier.com/locate/measurement

# Yaw/Heading Optimization by Machine Learning Model based on MEMS Magnetometer under Harsh Conditions

**Minh Long Hoang, Antonio Pietrosanto**

*Department of Industrial Engineering, University of Salerno, Via Giovanni Paolo II, 132 Fisciano (SA), ITALY*

ARTICLE INFO

*Article history:*

Keywords:
Magnetometer
Machine Learning
Random Forest Regression
MEMS
IMU
Accelerometer
Gyroscope
Sensor fusion

ABSTRACT

The paper's main goal is to accomplish a high accuracy of yaw/heading by Machine Learning approach when the motion range of vehicle/device calibration is limited. The nonlinear Random Forest (RF) Regression with proper training has a high potential to deal with the magnetometer uncertainty before calibration and during iron distortion cases. The proposed solution solely requires the magnetometer without other sensor's support. A Pan Tilt Unit-C46 (PTU-C46) with high precise positioning was used as a reference heading value to label the corresponding magnetic features in the learning model. The proposed approach helps yaw estimation to be carried out under harsh conditions, which resolve many difficulties in orientation tracking since the magnetometer is susceptible to hard iron and soft iron in the environment. In addition, many mechanical devices work only within the specific range and waste their dynamic motion around two axes or more just for calibration. Thus, the research focuses on the level rotation calibration around Z-axis within the restricted range of motion for practical application. The experiment was carried out using a low-cost platform equipped with Micro-Electro-Mechanical System (MEMS) sensors as gyroscope, accelerometer, and magnetometer. The 9 Degree of Freedom (DoF) Madgwick was implemented into the Microcontroller to compare with the proposed model. The sensor fusion can track the yaw value after the level calibration despite various error conduction. The RF model accomplishes a superior result with more stability and more minor error. Under iron disturbance or calibration absence, the ML model still maintains the good tracking command with maximum Mean Square Error of about 0.3$^{o}$, while the Madgwick is unsuccessful in heading measurement due to huge error in these circumstances.

## 1. Introduction

Nowadays, yaw/heading determination has played a key element in wide range of applications such as localization, position detection, and orientation tracking [1],[2]. The Micro-electro-mechanical-system inertial measurement units' sensors (MEMS-IMUs) [3],[4] have been utilized to detect heading popularly due to its small-size, high flexibility, and low cost, which are suitable in robotic control, automation [5], [6], etc.

Among Euler angles (roll, pitch and yaw) [7], [8] yaw is considered as the most challenging parameter to measure because the accelerometer is failed to give the yaw information when the sensor frame is aligned with the Earth frame. The Z-axis acceleration does not change its reading value when the sensor rotates to the right or left. The gyroscope conducts significant drift components during the integration process. On the other hand, GPS works less effectively in the indoor environment that becomes the problem for indoor applications. Environmental problems like Ionosphere's delay because of the distance and delay in time caused the low accuracy in result [9]. Thus, the magnetometer is an indispensable sensor to bring the true heading value by measuring the strength of the Earth's magnetic field vector in body coordinates. However, the iron distortion usually interferes the magnetometer performance. Hard iron from the magnetized material or external magnet moves the magnetic sphere away from original center coordinate. Soft iron from metallic material such as nickel battery deforms the magnetic

* Corresponding author.
E-mail address: mhoang@unisa.it.

field's shape, which results in the wrong value in data acquisition.

The magnetometer calibration is required to remove the bias and offset the measured data [10]. The most common method is to rotate the sensor following figure 8 shape or surrounding the axes to collect the maximum and minimum value of Mx, My, and Mz, which are the magnetic field of the X-axis, Y-axis, and Z-axis. After noisy data calculation, they will be used to correct the magnetic sphere position as well as its form scale [11], [12]. This method is quite effective and already employed in many real applications. Nevertheless, it is difficult for the entire rotation motion to occur since the platform is attached to a vehicle or machine arm with limited spin capability. There are various types of calibration to enhance the signal quality. A real-time approach for compass calibration using an extended Kalman filter and an unscented filter. This approach converts the magnetometer-body and geomagnetic reference vectors into an attitude-independent observation using scalar checking [13]. However, the common drawback of this kind of calibration method is that the algorithm will diverge or fail while large amounts of noise or blunders deteriorate the measurements. This issue prohibits compass calibration from many practical applications.

A particle swarm optimization system is used to calibrate the magnetometer, as it depends on a more accurate nonlinear model and does not need to consider the initial estimation parameters [14], [15]. This algorithm has the disadvantage of high computational cost, which is difficult to implement.

One method that can guarantee high precision and be used by many industrial companies is Madgwick- 9 Degree of Freedom (DOF) [16]. This sensor fusion filter uses the acceleration and magnetic data to correct the drift of angular velocity. With the quaternion propagation and gradient descent algorithm, the Madgwick filter achieves high reliability and has been considered one of the most popular techniques for heading evaluation. However, its accuracy still requires the good calibration of the magnetometer to compensate for gyroscope drift and the absence of information of acceleration on yaw spin.

Generally, all calibration solutions involve rotation around two or more axes which is challenging to attain. It is limited by the range of movement of vehicles or mechanical device. A paper proposes an integrated magnetometer and gyroscope calibration method with level rotation to support a cubature Kalman filter for calibration parameters estimate [17]. This method only needs level rotation for real-time calibration, but it still needs gyroscope support with complicated mathematics. Moreover, it is not guaranteed that it can handle the sudden interference after level rotation calibration.

The artificial intelligent approach has high potential for the concerned problem. A neural network was proposed to work on the nonlinear relationship between the compass heading and the proper heading, then converted the compass heading into the correct heading [18]. This technique minimizes the calibration motion and shows the potential in the heading correction during the magnetic disturbance. However, the input of the neural networks is only compass heading, putting a high burden on the training data and the network construction complexity. There is only one input parameter, and the model is vulnerable to the new disturbance. Moreover, training neural networks needs a lot of training data for orientation tracking, much more than that required for regular machine learning algorithms. In addition, complicated compotation is required to run variations of stochastic gradient descent over hundreds of epochs on multiple mini batches.

Another article uses the Back Propagation neural network algorithm with the genetic algorithm as the compensation current control algorithm, based on a three-axis external coil magnetic field [19]. This technique shows the improvement in accuracy and effect for strong magnetic interference compensation. Nevertheless, this hardware compensation methods requires the additional hardware part: three-Axis Solenoid Coil with concerned parameters coil radius, coil thickness, coil height and number of turns for the construction of a coil compensation model and analysis the magnetic field characteristics of different coils.

Meanwhile, the necessity of an approach for magnetometer heading measurement has become critical that can solve the iron distortion problem directly without addition hardware with less complexity. Therefore, this paper proposes the Machine Learning (ML) models [20] with nonlinear regression algorithm as heading evaluation. Machine learning supports predictions using less computation cost, at the same time, makes it imperative to replicate manufacturing practices based on pattern recognition from the data [21], [22], [23]. The strong point of the model is listed below:

- Only magnetometer is required.
- It can work in 3 stages independently: without calibration, calibrated magnetometer and iron disturbance.
- In the calibration case, it only demands the vertical level rotation within the device's operating range.

The ML model contains 5 features: 3 axes magnetic fields, the sphere radius, and the ratio between $M_x$ and $M_y$. Among the regression algorithms, the Random Forest (RF) regression [24],[25] is selected with the highest score accuracy, based on multiple decision trees to output the most effective heading angle. Each decision tree generates its own prediction by comparing the features of the acquired value with the data of trained nodes. Each node separates the input data to more specific range. After multiple comparisons, the decision tree provides a prediction. The same progress occurs with other trees. Eventually, the last result comes from the average prediction of all decision trees in the forest. This technique can avoid variation and solve unexpected noise well because each decision tree can act as an analyst. The RF knows how to match the input data to the actual heading angle with the most relevant features and proper training data.

The paper focusses on general distortion such as magnet and nickel cadmium battery. Other specific types of distortion will be discussed in the future work.

A Pan Tilt Unit-C46 (PTU-C46) [26] with high accurate positioning was used as a reference heading for training data. Here, the level rotation around Z-axis (the upward vertical to the Earth frame) is proceeded by the PTU-C rotation from right to left and vice versa, as illustrated in Figure 1. The RF model is compared with the 9 DOF Madgwick filter to verify each technique's pros and cons in terms of error and signal behavior before-after calibration-iron interference.

The paper is organized as follows: the first part is about an analysis of magnetic field behavior in 3 stages of calibration in the restrained rotation range, then a brief description of the

Madgwick filter. After this part, the ML model and detail structure of the RF model will be analyzed. Finally, experimental results and signal analysis will be shown at the end of the paper.

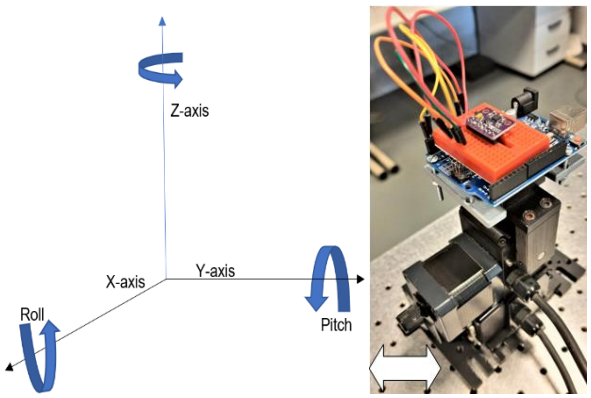


*Figure 1 PTU-C and mounted platform*

## 2. Magnetic field behavior

The examined angle is in the range of -150º to 150º due to the PTU-C rotation limit and also for research purpose, which is about the performance verification when the necessary range use of heading value is not fully 360º, only a smaller range like 90º, 300º, etc. In these cases, it will be optimal if the calibration motion just needs to occur at the same usage range and still provides effective measurement. The calibration was carried out by the PTU-C rotation of the Z axis from -150º to 150º, so the curve is not full circle, as shown in Figure 3. Figure 3 demonstrates the shape change of the $M_x$ and $M_y$ in 2D coordinate via three stages: before calibration - after calibration – during iron distortion. Ideally, the sphere must be centered around zero with a round shape. Nevertheless, the external interference moves the sphere away from the center or bend/attenuate the magnetic field due to hard iron and soft iron. In this case, the metallic material causes non-uniform of the geomagnetic field as no calibration subplot. After the calibration, the sphere has a round shape, centered around 0,0 coordinate.

Although after magnetometer calibration, if a new iron distortion such as strong magnet, the battery appears closely to the sensor, the sphere is still pushed away from the center point with the significant deformation. Regularly, the heading calculation is only carried out at the 2nd stage after the magnetometer calibration. The main task of the ML model is to effectively provide the heading value at the 1st and 3rd situations also.

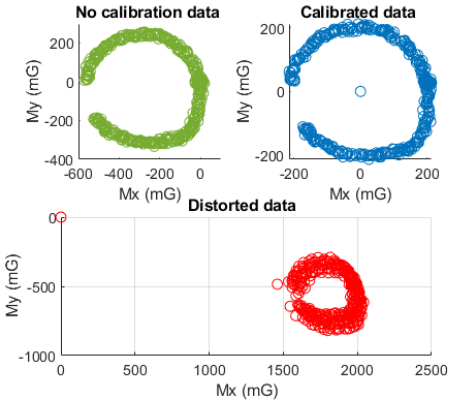


*Figure 3 Magnetic field behavior in 3 stages*

## 3. MADGIWCK-SENSOR FUSION FILTER

The Madgwick filter is an attitude and heading reference system (AHRS) algorithm, which is described clearly in [27] and [28]. It uses the measured acceleration and magnetic field to correct for gyroscopic drift. Figure 2 shows the block diagram of the Madgwick filter, which includes two main processes for the rigid body's orientation computation. In the first stage, a correction algorithm aligns the gyroscope measurements. The quaternion propagation is applied to the angular velocity to minimize the bias and the drift error.Then, the accelerometer and magnetometer measurements are fused together using an adjustable parameter, β, via the gradient descent algorithm to remove the noise in the gyroscope quaternion. Eventually, the yaw value is calculated by the fusion of the accelerometer, gyroscope, and magnetometer every update step based on the quaternion [29], [30].

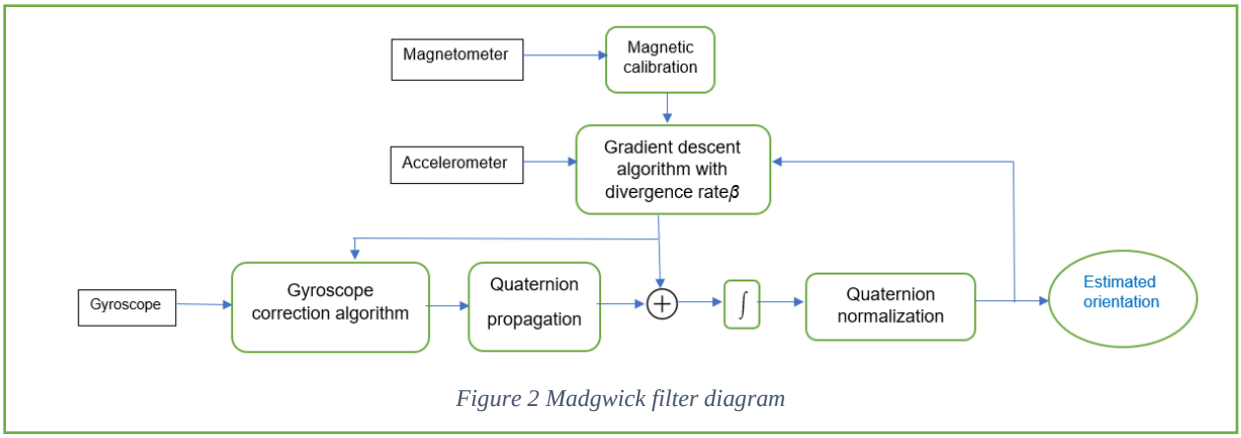


*Figure 2 Madgwick filter diagram*

## 4. MACHINE LEARNING MODEL ANALYSIS

### *4.1. ML features and output*

Ideally, the total intensity of Earth's magnetic field, measured by an ideal magnetometer should be a constant, so the sphere radius is supposed to have a fixed value. Practically, even after the calibration, the magnetic radius still has variation. Moreover, the fundamental heading calculation is based on the one unique arc tangent value from two variables $M_x$ and $M_y$, as described in [31]. Thus, together with magnetic field, two of these parameters are added into the ML model as detail features in order to enhance the specification during training process.

$$\text{Rad} = \sqrt{(M_x - X_0)^2 + (M_y - Y_0)^2 + (M_z - Z_0)^2} \quad (1)$$

$$\text{Ratio} = \frac{M_x}{M_y} \quad (2)$$

Where Rad is sphere radius. $X_0$, $Y_0$, $Z_0$ are center points with zero value.

The PTU-C plays the role of angle reference, labeled to the ML heading in the training model.

Numerous instances are fed into the ML model. The data types are listed as below:

- $M_x$, $M_y$, $M_z$, Rad, ratio as inputs and corresponding heading angle as output
- The Z-level calibrated mx, my, mz ,R, ratio as inputs and corresponding heading angle as output
- With the same parameters, the distorted data when the strong magnet interruption presence after the Z-axis calibration.

Table 1 shows 5 features and 1 output as heading value.

*Table 1ML Inputs and output*

| Inputs | | | | | Output |
|---|---|---|---|---|---|
| $M_x$ | $M_y$ | $M_z$ | Rad | Ratio | Yaw |

### *4.2. ML model evaluation*

The ML regression is a powerful tool to predict the continuous output variable in term of real value. The first critical stage is to select the most appropriate model for the concerned case by validating all the potential models via $R^2$ score (R squared), which indicates the fit accuracy of a set of predictions to the actual values. The value of $R^2$ will be between 0 and 1, 0 being no fit and 1 being a perfect fit.

To avoid overfitting, the ML algorithms must be evaluated on data that is not used to train the algorithm. For this case, more than 5000 instances are fed into the model. Cross-validation is utilized to estimate the performance of each ML algorithm with more minor variance than a single train-test set to split. This method splits the dataset into k-parts. Each part of the data is called a fold. The algorithm is trained on k - 1 folds with one held back and tested on the held back fold. This is repeated so that each fold of the dataset is given a chance to be the held back test set. After running cross-validation, the result is accomplished with k different performance scores, summarized by using a mean and a standard deviation. The choice of k should allow the size of each test partition to be large enough reasonably and enough repetitions of the test evaluation of the algorithm to provide a fair estimate of the algorithm's performance on unseen data. 10-fold cross-validation were selected for the magnetic field and heading estimate.

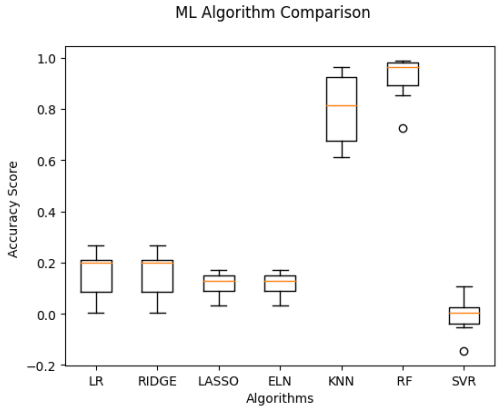


*Figure 4 ML Algorithm Comparison*

The relationship between the magnetic field and the heading angle is nonlinear which needs the nonlinear ML algorithms such as K-Nearest Neighbors (KNN), Random Forest (RF), and Support Vector Regression (SVR). Other linear ML Algorithms have the low capability to proceed with the data, especially during the iron distortion situation, as demonstrated in Figure 4.

As reported in Table 2, only KNN and RF algorithms have the highest score while other algorithms have low scores. The SVR has negative score, which means that this model fits the data poorly for this case. Fundamentally, KNN works based on finding the closest similar points via Euclidean distance. To predict a point, the k number of the closest point to the concerned point are taken into account and then track points by majority vote of its k neighbors. A lower number of neighbors can reduce the bias but causes higher variance because the system will model more noisy data. Here, the KNN number of neighbors is set as 3 to compromise the bias and variance in the model. In this case, KNN has the disadvantage with high dimensions because it becomes difficult for the algorithm to calculate the distance in each dimension.

RF is suitable for large data training. It predicts the result by averaging the numerous decision trees, so the model has high efficiency if it gains enough tree estimators. Another advantage of the RF model is to avoid the overfit issue from CART (Classification and Regression Trees), which generates the final outcome via a single decision tree only. After cross-validation test, KNN and RF models receive 20 percentages of the data, which are not included in cross-validation test for verification. And the collected score of KNN and RF for tested part are 0.72 and 0.96 respectively. The performance of these two algorithms will be compared further in the experimental part

*Table 2 Score accuracy for ML models*

| ML algorithm | $R^2$ score Mean | $R^2$ score Std |
|---|---|---|
| LR | 0.15 | 0.08 |
| RIDGE | 0.15 | 0.08 |
| LASSO | 0.11 | 0.04 |
| ELN | 0.04 | 0.04 |
| KNN | 0.79 | 0.12 |
| RF | 0.93 | 0.07 |
| SVR | -0.01 | 0.06 |

## 5. Random Forest Regression

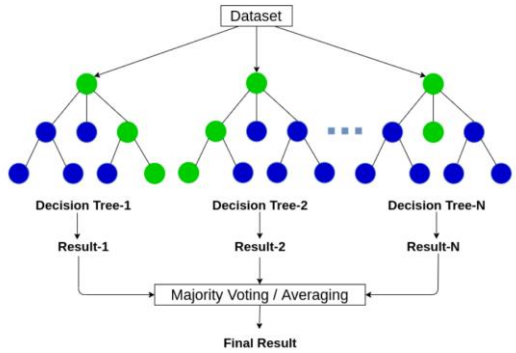


*Figure 5 RF structure [32]*

The most potential algorithm is RF which is the main model in this project and will be described in detail here. At this point, 80% of the total data are used for RF training and 20% left are employed for testing process. RF changes the algorithm for the way that the sub-trees are learned so that the resulting predictions from all of the subtrees have less correlation by using bootstrap method. In each bootstrap training set, about one-third of the instances are enter each decision tree [33].

RF is an ensemble of decision trees that includes 3 main features for data processing:

- Every single tree is constructed of a different sample of rows. At each node, a different sample of features is selected for splitting to the following stages.
- Each of the trees provides its prediction.
- These predictions are averaged to supply the last decision value by the trees in the forest.

There are some core parameters that impact the RF performance:

- Number of estimators represents the number of trees in the forest. Typically, a higher number of trees leads to better data learning, but many trees can slow down the training process. To improve the training speed, this model engine is allowed to use no limit of processors. Here, 100 tree estimators are selected.
- The minimum number of samples required to be at a leaf node. A split point at any depth will only be considered if it leaves at least this minimum training samples in each of the left and right branches.
- The depth of each tree in the forest. The deeper the tree generates more splits and captures more information about the data, but more processing time consumption.

As shown in Figure 6, the rise of the decision tree number improves the $R^2$ score of the RF model. After 100 estimators, the score does not vary significantly in both training and test data. About the leaf minimum sample, 1 is a good value that simultaneously maintains the high score at training and testing evaluation. Finally, the trend of the maximum depth reaches saturation at 12 in the test part. Based on the analysis, the selected parameters are 100 estimators, 1 leaf minimum sample and 12 maximum depths.

Figure 7 shows the processing steps in a decision tree. Due to the resolution concern, the visualization only contains 3 depths for clear observation. The inner working of a Decision Tree operates following if-else conditions. The top node splits into left (True) and right nodes (False), which are decision nodes. These nodes then split into their respective right and left nodes. At the end of the leaf node, the average observation that occurs within that area is computed. Most bottom nodes are referred to as leaves or terminal nodes. The value in the leaves is usually the mean of the observations occurring within that specific region. For instance, in the rightmost leaf node below, $25^0$ is the angle value from the average of the 3 samples with zero MSE value.

For example, under the iron distortion condition, the acquired data has the Radius = 55000, My = 200, Ratio = 0.31. This data will enter the root node and the first answer is False because Radius > 53313.779, so it will move to the right. At this node, My <266.405, so it goes down to the left as True condition. At the 3rd node, its ratio > 0.24, so it comes to the right side as False case and take the final angle of $-100^o$. In this decision tree, this value has high accuracy because the MSE of this node is zero, but other values require more nodes to attain the more precise prediction.

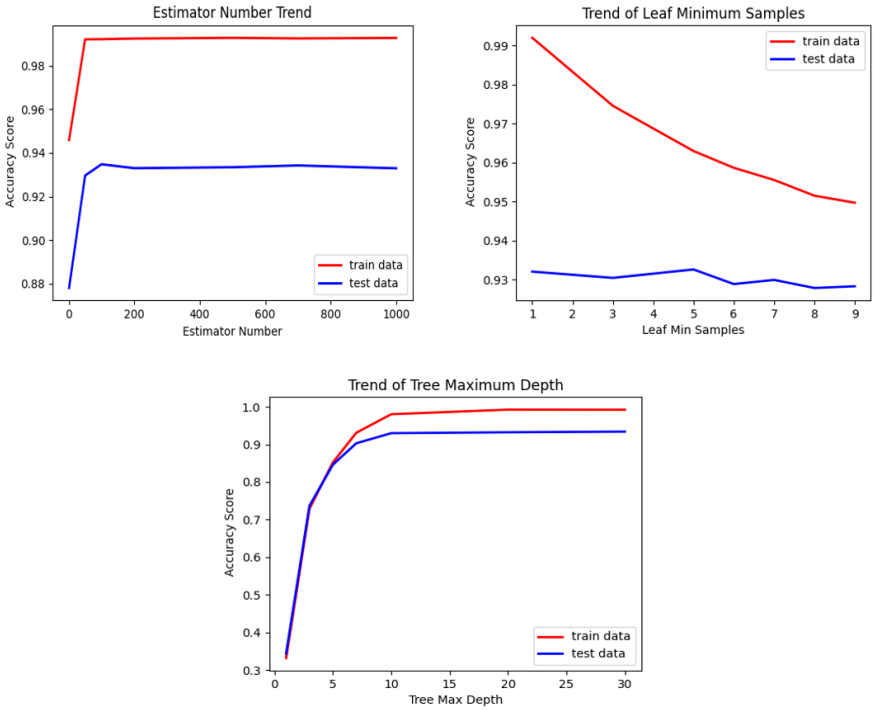


*Figure 6 Trend of RF accuracy respect to hypermeter*

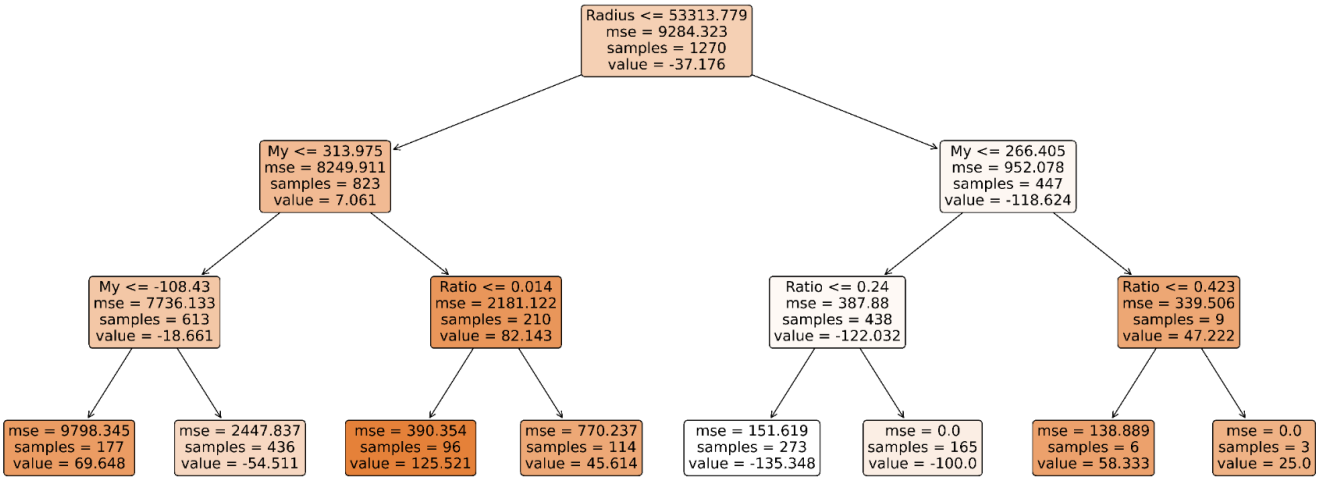


*Figure 7 A decision visualization with depth maximum of 3*

## 6. Experimental setup

### *6.1. Test bench*

The experimental sensors are the MPU-9250 [34], which is a 9-axis Motion Tracking device that combines a 3-axis gyroscope, 3-axis accelerometer, 3-axis magnetometer. The accelerometer works in the range of ± 2g; the gyroscope range is ± 245 °/s, full magnetic scale is ± 48 gauss. After the Low-pass filter, the output sample rate is about 92 Hz. The accelerometer and gyroscope are used only for the sensor fusion Madgwick. In this case, the fixed, parent coordinate system used is North-East-Down (NED) as the reference frame. In the NED reference frame, the X-axis points north, the Y-axis points east, and the Z-axis points downward.

The IMU acquisition and sensor fusion has been implemented into Arduino UNO [35] by Arduino Integrated Development Environment (IDE). The sensor is connected to the MCU development via an Inter-Integrated Circuit (I2C) communication line. A Pan-Tilt Uni Controller (PTU-C46) with resolution 0.051° preposition provides fast and accurate positioning of cameras that was manipulated to verify the algorithm performance. The sensor platform was assembled on PTU-C for tracking this device's orientation. The test bench was built up and mounted on the laboratory table. All the acquisition data were sent to the host computer for signal analysis via USB cable from the MCU-board. ML models were designed by Python environment based on scikit-learn [36], which are a powerful and easy-to-use free, open-source Python library for ML development and evaluation. The

magnetic value acquisitions enter the ML model as real-time input; then, yaw angles are detected practically.

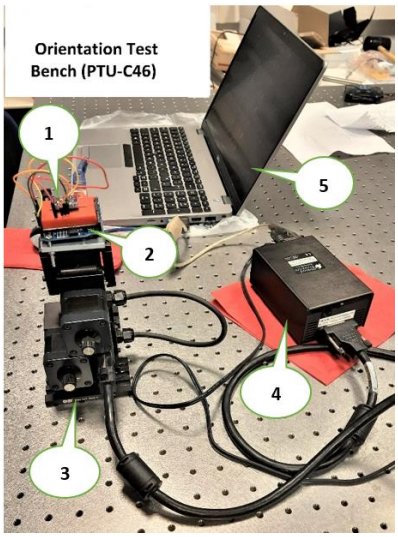


*Figure 8 Test bench for experiment.*

1. Sensor MPU-9250
2. Arduino Uno
3. PTU - C46 Pan Tilt Unit
4. PTU-C Controller
5. Computer

To test the iron distortion case, a strong magnet and a nickel cadmium battery were approached the sensor to evaluate the ML model performance in the interference circumstance.


*Figure 9 Strong magnet and battery for interference test*

### *6.2. ML model validation*

This part describes the validation of the ML and Dl model based on the training and test sets. The models were operated by a PC processor: Intel(R) Core (TM) i7-10850H CPU @ 2.70GHz.

With the rotation of PTU-C46, the magnetic field and the corresponding heading value were acquired to be the input data. From -150º to 150º with resolution of 25º, about 31 samples are attained for each reference angle. This process is applied to 3 cases: no calibration- after calibration and iron distortion case. To verify the output performance, the 10% of data is used as validation set to evaluate the trained model, and the rest of 10% data to test the final model. The validation data infuses new data into the model which hasn't evaluated before. Validation data provides the first test against unseen data, to evaluate how well the model makes predictions based on the new data. After the model is built, testing data once again validates that it can make accurate predictions

As reported in Table 3, the KNN require less time of model operation, while RF accomplishes the advantages of smaller error in both validation and test set.

*Table 3 Model validation*

| Parameters | KNN | RF |
|---|---|---|
| Data Number | 5580 | |
| Lunch time (s) | 2.05 | 4.35 |
| Validation MSE (º) | 1.95 | 0.24 |
| Test MSE (º) | 1.98 | 0.21 |

## 7. Experimental analysis

To validate the operation of the described models, the result analysis is divided into 3 major parts:

- The verification between KNN and RF models to select the best algorithm for ML operation.
- The comparison between the ML model and Madgwick system without calibration and after the magnetometer calibration.
- The impact of iron distortion on the tracking capability of both operating systems.

### *7.1. ML algorithm comparison*

The purpose of the 1st part is to characterize KNN and RF since both accomplish good $R^2$ scores, as described early.

After the magnetometer calibration, the rotation process was carried out from -150º to 150º continuously. This test also contains the sensor fusion Madgwick to observe its behavior during dynamic motion. As shown in Figure 10, the RF and Sensor Fusion Madgwick can achieve the desired points surround -150º and 150º in the proper trend without any strange spikes. The detail variation when the sensor finishes its motion and rests in a stationary angle will be analyzed in detail at the 3rd test.

The KNN algorithm failed to predict the goal at the end of the spinning period. Many sudden spikes appear, which pull down the KNN outputs from the right tracking way. This issue occurs because KNN does not work well with large dataset. The distance calculation between the new point and each existing points is huge which degrades the performance of the algorithm, especially when there are the mechanical vibrations in the real-time operation. Consequently, this algorithm is failed to provide the proper value in the dataset range from 130º to 150º.

Table 4 reports the evaluation metrics from the mentioned regression models: mean absolute error (MAE) and mean square error (MSE). As the evaluated data, the RF model accomplishes smaller error than the KNN model, so the RF fits the best magnetic field and yaw relationship. Therefore, the RF will be used to compare with the Sensor fusion filter.

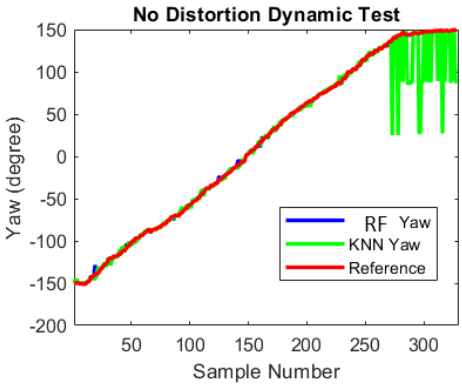


*Figure 10 Yaw test after calibration*

*Table 4 ML model error*

| ML model | MAE (°) | MSE (°) |
|---|---|---|
| KNN | 3.58 | 6.98 |
| RF | 0.21 | 0.30 |

### *7.2. Yaw value without magnetometer calibration*

In this test, PTU-C was controlled to move in stair-step of $25^0$. All the step angles were acquired to verify the yaw estimation from the Sensor fusion and RF model.

Theoretically, the yaw parameter cannot be detected due to the external interferences that cause the significant issue to magnetic measurement. As shown in Figure 11, the Madgwick filter is not able to track the yaw value appropriately, even with the support of the accelerometer and gyroscope. It still conducts the different value when the device moves and stays at new angle, but the precision is highly incoherent to the reference data. On the other hand, the RF model still haves good command of tracking the yaw value because it already trained the data in the no-calibrated case with the PTU-C angle as coherent yaw data with the MSE is about 0.19°.

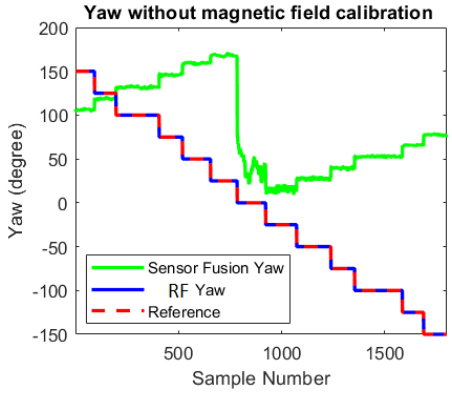


*Figure 11 Yaw measurement without magnetic calibration*

### *7.3. Sensor fusion vs RF algorithm*

After the magnetomer calibration, the yaw is detected by the 9DoF sensor fusion via Madgwick algorithm, which is compared with the RF to observe their tracking efficiency. The PTU-C spins from -150° to 150° with differential step 30°. About 100 samples are extracted from each indicated angle for the MSE validation

Figure 12 shows the data at static case of -60° and $150^0$. The sensor fusion generates considerable variation around the indicated angles which is understandable because the sensor is mounted on the metallic device PTU_C and the calibration rotation is only on the Z-level frame. The ML model demonstrates a better result with narrow variation. There are the samples which predicts exact the same value as reference that shows the well-being processing operation. Moreover, the RF also has good stability since the standard deviation (Std) is small as shown in Table 5. The MSE of the Madgwick filter needs to be minimized by more rotation around two or more axes during calibration. Otherwise, the yaw evaluation is less precise as reported data. About this point, the RF model successfully accomplishes inferior error with balanced execution.

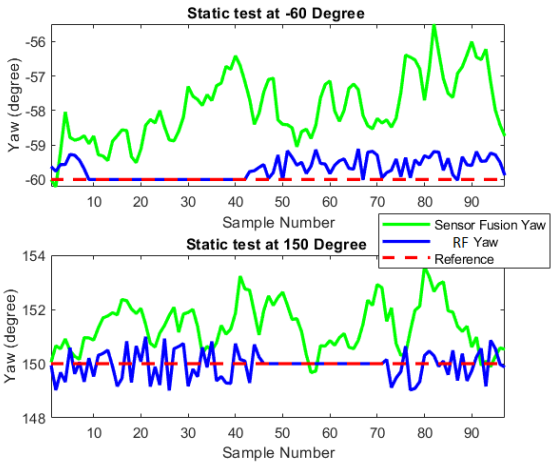


*Figure 12 Static test ant negative and positive case*

*Table 5 Error evaluation for RF and Madgwick filter*

| Yaw Reference (°) | RF Model (°) | | Sensor Fusion (°) | |
|---|---|---|---|---|
| | MSE | Std | MSE | Std |
| -150 | 0.19 | 0.04 | 2.10 | 1.47 |
| -120 | 0.18 | 0.09 | 0.72 | 0.62 |
| -90 | 0.18 | 0.04 | 4.01 | 1.13 |
| -60 | 0.17 | 0.07 | 4.22 | 0.95 |
| -30 | 0.23 | 0.10 | 1.37 | 1.04 |
| 0 | 0.15 | 0.09 | 1.86 | 0.86 |
| 30 | 0.27 | 0.04 | 1.51 | 0.85 |
| 60 | 0.21 | 0.05 | 0.88 | 0.94 |
| 90 | 0.15 | 0.04 | 1.27 | 1.03 |
| 120 | 0.10 | 0.08 | 3.45 | 0.99 |
| 150 | 0.25 | 0.07 | 2.70 | 0.87 |

*7.4. Iron distortion test*

The iron distortion is generated by a strong magnet and a nickel-cadmium battery, like Figure 9. Consequently, the interference pushes the Sensor Fusion yaw away from the original value, leading to a significant error in the acquisition period, as shown in Figure 14.

Figure 13 visualizes the yaw comportment in the RF model and Madgwick filter. The RF model still tracks the appropriate angle, while the sensor fusion cannot detect the correct value anymore due to the substantial interference. This feature shows the ML model over the traditional method. With multi-case training, the ML model can learn how to predict the results like they are supposed to be.

A decision tree produces the last decision via a series of questions to the data. Each question narrows our possible values via the lowest MSE in each subset until the model is precise enough to make a single prediction. Hence, after training with the distortion environment, the RF model adapts very well to the current situation with low error, as indicated in Table 6. The maximum MSE is only about 0.28º at 50º, which is a good result during the iron distortion.

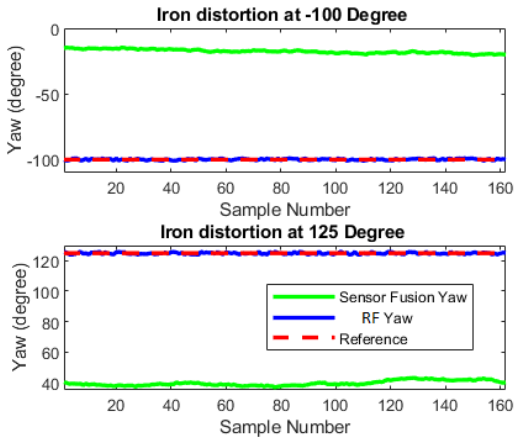


*Figure 13 Yaw measurement during iron interference*

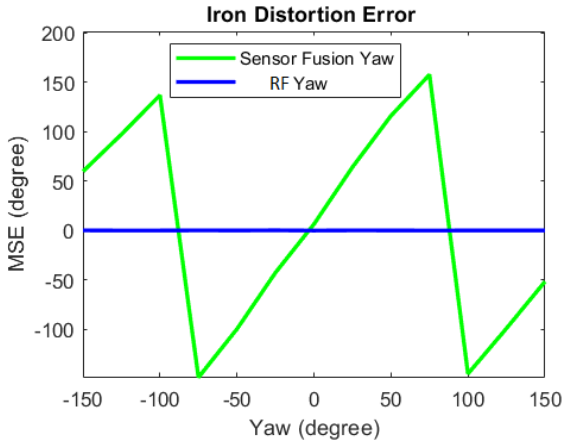


*Figure 14 MSE on the angle variation during iron distortion*

*Table 6 MSE during iron distortion*

| Yaw Reference (º) | Sensor Fusion MSE (º) | RF MSE (º) |
|---|---|---|
| -150 | 59.69 | 0.10 |
| -125 | 97.22 | 0.14 |
| -100 | 136.75 | 0.20 |
| -75 | -148.49 | 0.11 |
| -50 | -99.59 | 0.25 |
| -25 | -42.56 | 0.27 |
| 0 | 6.44 | 0.11 |
| 25 | 64.05 | 0.18 |
| 50 | 115.92 | 0.28 |
| 75 | 157.52 | 0.25 |
| 100 | -144.86 | 0.12 |
| 125 | -98.9 | 0.10 |
| 150 | -51.61 | 0.18 |

Another test was carried out for RF model to verify its performance at random dynamic motion. In this test, the device rotates from -150$^0$ to 150$^0$ and stops at a random point for various seconds during the iron distortion. Like previous analysis, the RF model follows the rotation trend of the device properly with an average MSE of approximately 0.24º, as illustrated in Figure 15.

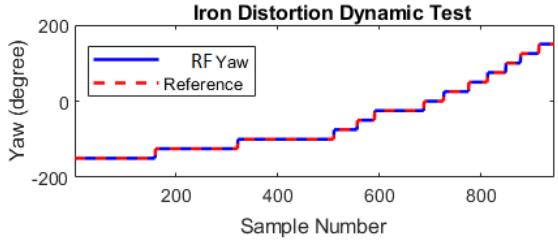


*Figure 15 RF performance during distortion*

## *8.* CONCLUSION

The paper described a practical ML approach for yaw/heading estimate that relief the calibration burden in magnetometer calibration. The RF model was selected among ML algorithms due to its superior accuracy compared

to the others. Furthermore, the RF model was compared with the 9DOF- Madgwick and demonstrated its pros.

In calibrated case, the sensor fusion technique generates bigger MSE due to the restraint of motion calibration. Thanks to training knowledge, the new approach is able to predict the proper value without magnetometer calibration and during iron interference. Meanwhile, the traditional sensor fusion conducts unacceptable error under these harsh conditions. To deal with more kinds of distorted material, more data are required to guarantee its execution.

The proposed method opens a new way for heading measurement, which is highly effective in the case of limited device motion. Hence, this technique potentially becomes a promising key in orientation with less complicated but effective processes. In the future, the work will be extended with more tests against various types of iron distortion to verify the model efficiency and its comportment.

## REFERENCES


[1] M. L. Hoang, A. Pietrosanto, S. D. Iacono and V. Paciello, "Pre-Processing Technique for Compass-less Madgwick in Heading Estimation for Industry 4.0," 2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), 2020, pp. 1-6, doi: 10.1109/I2MTC43012.2020.9128969.

[2] X. Zhang and X. Lu, "Yaw Robust Control of UAV Based on Linear Auto-Disturbance Rejection Compensation," 2018 3rd International Conference on Mechanical, Control and Computer Engineering (ICMCCE), 2018, pp. 80-84, doi: 10.1109/ICMCCE.2018.00024.

[3] M. L. Hoang and A. Pietrosanto, "An Effective Method on Vibration Immunity for Inclinometer based on MEMS Accelerometer," 2020 International Semiconductor Conference (CAS), 2020, pp. 105-108, doi: 10.1109/CAS50358.2020.9267997.

[4] S. Nihtianov and A. Luque, Eds., Smart sensors and MEMS: Intelligent sensing devices and microsystems for industrial applications, 2nd ed. Cambridge, England: Woodhead Publishing, 2018.

[5] M. F. Ruzaij, S. Neubert, N. Stoll and K. Thurow, "Auto calibrated head orientation controller for robotic-wheelchair using MEMS sensors and embedded technologies," 2016 IEEE Sensors Applications Symposium (SAS), 2016, pp. 1-6, doi: 10.1109/SAS.2016.7479886.

[6] S. Gang; L. Xisheng; Jiang, Zhengfu. 2018. "An Improved Yaw Estimation Algorithm for Land Vehicles Using MARG Sensors" Sensors 18, no. 10: 3251. https://doi.org/10.3390/s18103251

[7] M. Carratù, S. D. Iacono, M. Long Hoang and A. Pietrosanto, "Energy characterization of attitude algorithms," 2019 IEEE 17th International Conference on Industrial Informatics (INDIN), 2019, pp. 1585-1590, doi: 10.1109/INDIN41052.2019.8972300.

[8] M. L. Hoang and A. Pietrosanto, "A Robust Orientation System for Inclinometer With Full-Redundancy in Heavy Industry," in IEEE Sensors Journal, vol. 21, no. 5, pp. 5853-5860, 1 March1, 2021, doi: 10.1109/JSEN.2020.3040374.

[9] Abha Damani, Hardik Shah, Krishna Shah, "Global Positioning System for Object Tracking," International Journal of Computer Applications (0975 – 8887) vol. 109, no. 8, January 2015

[10] F. Hu, Y. Wu, Y. Yu, J. Nie, W. Li and Q. Gao, "An Improved Method for the Magnetometer Calibration Based on Ellipsoid Fitting," 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), 2019, pp. 1-5, doi: 10.1109/CISP-BMEI48845.2019.8965821.

[11] M. J. Caruso, "Applications of magnetoresistive sensors in navigation systems," SAE Transaction, vol. 106, pp. 1092–1098, 1997.

[12] V. Renaudin, M. H. Afzal, and G. Lachapelle, "Complete triaxis magnetometer calibration in the magnetic domain," J. Sens., vol. 2010, pp. 1–10, 2010.

[13] Crassidis J L , Lai K-L and Harman R R "Real-time attitude independent three-axis magnetometer calibration" J. Guid. Control Dyn. 28, pp. 115–20, 2005.

[14] Z. Wu, Y. Wu, X. Hu, and M. Wu, "Calibration of three-axis magnetometer using stretching particle swarm optimization algorithm," IEEE Trans. Instrum. Meas., vol. 62, no. 2, pp. 281–292, 2013.

[15] B. A. Riwanto, T. Tikka, A. Kestila, and J. Praks, "Particle swarm optimization with rotation axis fitting for magnetometer calibration," IEEE Trans. Aerosp. Electron. Syst., vol. 53, no. 2, pp. 1009–1022, 2017.

[16] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in Proc. IEEE Int. Conf. Rehabil. Robot., Jun. 2011, pp. 1–7.

[17] Z. Wu and W. Wang, "Magnetometer and gyroscope calibration method with level rotation," Sensors (Basel), vol. 18, no. 3, p. 748, 2018.

[18] J.-H. Wang and Y. Gao, "A new magnetic compass calibration algorithm using neural networks," Meas. Sci. Technol., vol. 17, no. 1, pp. 153–160, 2006.

[19] J. Fu, Z. Ning and Y. Chang, "Active Compensation Method for Strong Magnetic Interference of MEMS Electronic Compass," in IEEE Access, vol. 9, pp. 48860-48872, 2021, doi: 10.1109/ACCESS.2021.3067362.

[20] S. Mirzaei, T. Sidi, C. Keasar and S. Crivelli, "Purely Structural Protein Scoring Functions Using Support Vector Machine and Ensemble Learning," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 16, no. 5, pp. 1515-1523, 1 Sept.-Oct. 2019, doi: 10.1109/TCBB.2016.2602269.

[21] S. Ruhela and S. Riaz, "An intelligent combination: Assessing the impact of harmonized emotional and artificial intelligence for the success of industry 4.0," in 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2019, pp. 1–5

[22] L. Romeo, M. Paolanti, G. Bocchini, J. Loncarski, and E. Frontoni, "An innovative design support system for industry 4.0 based on machine learning approaches," in 2018 5th International Symposium on Environment-Friendly Energies and Applications (EFEA), 2018, pp. 1– 6.

[23] R. Shams, "Developing machine learning products better and faster at startups," IEEE Engineering Management Review, vol. 46, no. 3, pp. 36–39, 2018.

[24] H. S. Dhiman, D. Deb, and V. E. Balas, "Preface," in Supervised Machine Learning in Wind Forecasting and Ramp Event Prediction, Elsevier, 2020, pp. xix–xx.

[25] V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston, "Random forest: a classification and regression tool for compound classification and QSAR modeling," J. Chem. Inf. Comput. Sci., vol. 43, no. 6, pp. 1947–1958, 2003.

[26] Rollins Road Burlingame, "Pan-Tilt Unit (Model PTU) User's Manual", Version 1.14, 10/24/2000

[27] M. L. Hoang, S. D. Iacono, V. Paciello and A. Pietrosanto, "Measurement Optimization for Orientation Tracking Based on No Motion No Integration Technique," in IEEE Transactions on Instrumentation and Measurement, vol. 70, pp. 1-10, 2021, Art no. 9503010, doi: 10.1109/TIM.2020.3035571.

[28] O. Sarbishei, "On the accuracy improvement of low-power orientation filters using IMU and MARG sensor arrays," in 2016 IEEE International Symposium on Circuits and Systems (ISCAS), 2016.

[29] H. Xing, Z. Chen, C. Wang, M. Guo, and R. Zhang, "Quaternion-based complementary filter for aiding in the self-alignment of the MEMS IMU," in Proc. IEEE Int. Symp. Inertial Sensors Syst. (INERTIAL), Apr. 2019, pp. 1–4

[30] T. Brunner, J.-P. Lauffenburger, S. Changey, and M. Basset, "Quaternion-based IMU and stochastic error modeling for intelligent vehicles," in Proc. IEEE Intell. Vehicles Symp. (IV), Jun. 2015, pp. 877–882.

[31] M. L. Hoang and A. Pietrosanto, "Yaw/Heading optimization by drift elimination on MEMS gyroscope," Sens. Actuators A Phys., vol. 325, no. 112691, p. 112691, 2021.

[32] Abhishek Sharma, "Decision Tree vs. Random Forest - Which Algorithm Should You Use?" Analytics Vidhya, May 12, 2020.

https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/.

[33] L. Breiman, Random Forests. Machine Learning., vol. 45, no. 1, pp. 5–32, 2001.

[34] InvenSense, MPU-9250 Product Specification Revision 1.1, June 2016.

[35] ArduinoBoardUno. (n.d.). Retrieved May 28, 2021, from https://www.arduino.cc/en/Main/arduinoBoardUno>

[36] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.