Check for updates

TRANSACTIONS OF THE INSTITUTE OF MEASUREMENT & CONTROL

Article - Measurement

Transactions of the Institute of Measurement and Control
1–10
© The Author(s) 2023
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/01423312231214832
journals.sagepub.com/home/tim
Sage

Artificial neural network for tilt compensation in yaw estimation

Ali Mounir Halitim[1], Mounir Bouhedda[1], Sofiane Tchoketch-Kebir[1] and Samia Rebouh[2]

Abstract

Low-cost inertial measurement units (IMUs) are commonly used to determine the orientation of objects, such as unmanned aerial vehicles (UAVs) and smartphones. They calculate yaw by measuring Earth's magnetic field's horizontal components. However, in the presence of tilt (pitch or roll), a tilt-compensation operation is necessary. This is usually done by projecting measurements onto a horizontal plane. This method has limitations, particularly for large tilt angles and when the IMU is pointing toward the east or west directions. In this paper, we expose the shortcomings of this conventional approach and propose a novel machine learning–based solution employing an artificial neural network (ANN). This method eliminates the need to determine tilt angles and uses accelerometer and magnetometer measurements as its inputs. The dataset for training and testing the ANN was collected based on a 3D nonmagnetic scaled platform, using a low-cost IMU and a Raspberry Pi platform. On one hand, our method outperforms the conventional tilt-compensation technique and other complementary filters (Madgwick and Mahony) in terms of accuracy, as evidenced by the root mean square error (RMSE = 1.95°). However, this superiority comes at the expense of a more complex system that consumes more processing time.

Keywords

Artificial intelligence, inertial measurement, machine learning, micro–electro–mechanical systems, artificial neural network, tilt compensation, yaw estimation

Introduction

In numerous applications, such as tracking, navigation, and human activity recognition, it is imperative to obtain accurate estimates of orientation angles. Euler angles offer a means of representing the 3D orientation of an object through a combination of three rotations around the X, Y, and Z axes, conventionally referred to as Roll(ϕ), Pitch(θ), and Yaw(ψ), respectively.

Inertial measurement units (IMUs) have emerged as a compact and cost-effective alternative to conventional systems, thanks to advances in Micro–Electro–Mechanical Systems (MEMS) technology (Langfelder and Tocchio, 2013). However, the low-cost advantage comes with limitations, primarily manifested as reduced accuracy, especially when compared to more expensive, high-grade IMUs. Consequently, researchers have been actively working on improving the performance of solutions based on low-cost IMUs (Ru et al., 2022).

An IMU typically comprises a combination of tri-axis accelerometers, gyroscopes, and magnetometers. Accelerometers measure linear acceleration, enabling the tracking of an object's movement and changes in velocity. Gyroscopes, however, provides information about an object's angular velocity. Finally, magnetometers measure the strength and direction of the surrounding magnetic field (Carratu et al., 2019). Various approaches exist for combining these individual sensors to estimate orientation angles.

By fusing accelerometer and gyroscope measurements, precise estimations for pitch and roll have been achieved (Ferdinando et al., 2012; Madgwick et al., 2011). However, determining yaw, which represents the angle between the magnetic north and the forward axis X, poses a greater challenge. This is because, in addition to the gyroscope's drift, accelerometers cannot detect any rotation around the z-axis when the sensor frame aligns with the Earth frame (Hoang et al., 2020).

Yaw estimation can be achieved using magnetometers, which enable the measurement of the surrounding magnetic field. In the absence of magnetic disturbances, the magnetometer is capable of measuring the Earth's magnetic field which is used to determine the angle between the longitudinal axis of the sensor and the magnetic north. Meanwhile, the magnetometer measurements are usually susceptible to errors, primarily due to unwanted local magnetic fields surrounding the sensor (Renaudin et al., 2010). Therefore, the procedure

[1]Laboratory of Advanced Electronic Systems (LSEA), University of Medea, Algeria
[2]Laboratory of Experimental Biology and Pharmacology (LBPE), University of Medea, Algeria

Corresponding author:
Ali Mounir Halitim, Laboratory of Advanced Electronic Systems (LSEA), University of Medea, Medea, 26000, Algeria.
Email: mounir.halitim@gmail.com

of calibration is critical in achieving accurate readings from the magnetometer.

The Earth's magnetic field has a component parallel to the horizontal plane that always points toward the magnetic north. Therefore, it is feasible to calculate yaw using its horizontal components, as expressed by Relation 1

$$\psi = -\tan^{-1}\frac{H_y}{H_x} \tag{1}$$

where H_x and H_y are the horizontal projections of the Earth's magnetic field.

When the sensor is parallel to the ground, both pitch and roll angles are zero, and the calculation of yaw can be performed using the magnetic field's horizontal components. However, when magnetometers are deployed on a moving vehicle, they are most often not parallel to the Earth's surface. In such cases, a conventional tilt-compensation method is typically employed. It consists in projecting tri-axis magnetometer measurements onto a horizontal plane.

In Figure 1, the magnetometer axes are aligned with the body frame comprising three orthogonal axes. In this frame, the x-axis corresponds to the direction of forward motion, the z-axis points downward, and the y-axis is perpendicular to the plane formed by the x and z axes. The terms h_x, h_y, and h_z represent the magnetometer measurements along the body frame axes. With the knowledge of the sensor tilt angles, the projections of the magnetic field onto the horizontal plane formed by the X and Y axes can be determined using Relation 2

$$\begin{pmatrix} H_x \\ H_y \end{pmatrix} = \begin{pmatrix} cos\theta & sin\theta sin\phi & -sin\theta cos\phi \\ 0 & cos\phi & sin\phi \end{pmatrix} \begin{pmatrix} h_x \\ h_y \\ h_z \end{pmatrix} \tag{2}$$

Despite being widely used in the literature (Ahuja et al., 2021; Árvai and Homolya, 2020; Hanafi et al., 2019; Wang et al., 2022), this conventional method will be discussed in this paper to highlight its limitations, particularly for large tilt angles and when the IMU is pointing toward the east and west directions.

Other tilt-compensation techniques have been documented in the existing literature. Xu et al. (2019) proposed a smartphone-based pedestrian dead reckoning solution that relies on an adaptive tilt-compensation method for precise yaw angle estimation. This method uses the terms of the direction cosine matrix (DCM), calculated using accelerometers and magnetometers. It is important to note that their study primarily focused on compensating for tilt related to pitch angle only. Duymaz and Işik (2021) introduced an alternative tilt-compensation technique, which employs a probabilistic approach using data from accelerometers and magnetometers. The proposed technique was validated exclusively through simulations.

Furthermore, other attitude estimation techniques can also address the tilt-compensation issue in yaw estimation. Two commonly used methods are Extended Kalman Filtering (EKF) and complementary filters (Cirillo et al., 2016). While EKF is based on a probabilistic approach, complementary filters employ frequency domain analysis to filter and fuse sensor data for orientation estimation. These filters are simple and effective, making them suitable for comparison with the proposed method. Two complementary filters, namely Madgwick et al. (2011) and Mahony et al. (2008), were employed in this study.

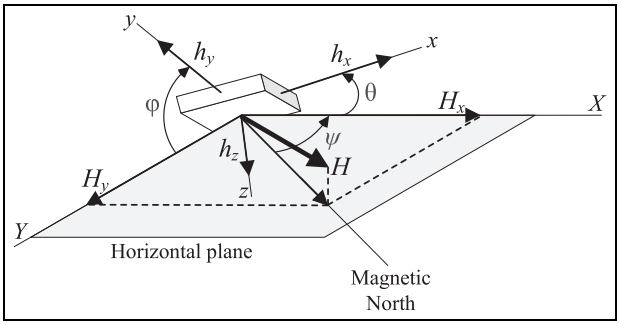


Figure 1. Yaw determination using a tri-axis magnetometer.

This paper proposes a novel machine learning (ML)–based method for tilt compensation in yaw estimation. It involves an artificial neural network (ANN) that uses accelerometer and magnetometer measurements as features and outputs an estimation of the yaw angle. This method is effective, regardless of whether tilt is present or not, and has the added advantage of not requiring the estimation of pitch and roll angles, nor the use of complex calibration approaches for accelerometer and magnetometer measurements. The proposed ANN adopts a supervised learning approach, in which input data and their corresponding outputs are used to adjust the ANN's parameters (weights and biases). The dataset used for training and testing the ANN is collected using a 3D nonmagnetic scaled platform.

Our paper presents two significant contributions. First, we conduct a series of experiments that highlight the limitations of the conventional tilt-compensation method. Second, and as an alternative, we propose an innovative ML-based solution that uses measurements from an IMU to compensate for tilt in yaw estimation. We compare our proposed approach not only to the conventional method but also to two well-established complementary filters.

The rest of this paper is structured as follows. In section "Magnetometer measurements calibration," we present the procedure of magnetometer measurements calibration. Section "Conventional tilt-compensation method assessment" details the conducted experiments used to reveal the limitations of the conventional tilt-compensation method. A brief description of Madgwick and Mahony filters is provided in section "Madgwick and Mahony algorithms." Section "Solution approach" introduces the dataset collection process along with the ANN's adopted model and its training process. In section "Results and discussion," an evaluation of the performance of the proposed method is conducted, and a comparison with the conventional tilt-compensation method, and Madgwick and Mahony filters, is presented. The last section "Conclusion" concludes the paper and outlines future work.

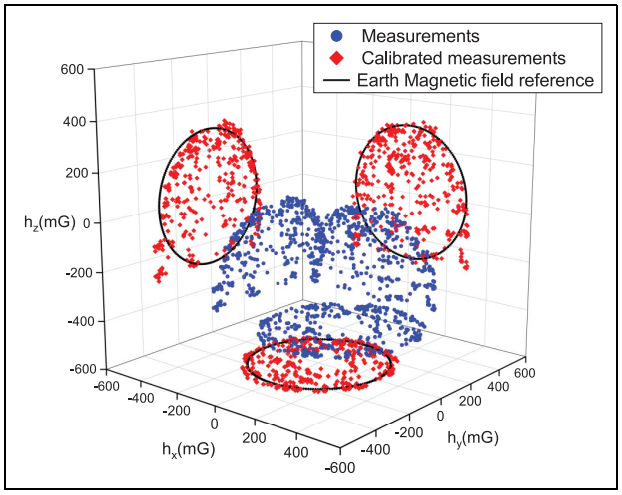


Figure 2. Magnetometer measurements calibration.

Magnetometer measurements calibration

As for a wide variety of low-cost sensors, magnetometers suffer from errors due to fabrication limitations (offset error, sensitivity error, cross-axis sensitivity errors, and sensor measurement noise). Nevertheless, the perturbations caused by ferromagnetic materials and electromagnetic systems in the vicinity of the sensor are known to be the major sources of errors affecting the magnetometer measurements (Renaudin et al., 2010). These perturbations can be categorized as hard and soft iron errors.

The ideal response for a tri-axis magnetometer is a sphere centered at the origin. Hard iron consists of constant or slowly time-varying fields generated by magnetized material or external magnet in the proximity of the sensor. This causes the magnetic sphere response to move away from the origin (Li et al., 2019). In contrast, soft iron is a changing effect that results from metallic material, such as nickel batteries, and typically manifests as a distortion of the ideal sphere into an ellipse (Long Hoang and Pietrosanto, 2022).

A widely used method to calibrate magnetometer readings involves moving the sensor in a figure eight pattern and recording the minimum and maximum readings along each of the three axes. These values are then used to calculate offsets and scale factors for each axis to re-center and rescale the magnetometer response (Poulose et al., 2019). The AK8963C magnetometer, which is integrated into the MPU9250 9 DoF motion sensor, was used to collect actual experimental data. The calibration procedure results are illustrated in Figure 2.

Conventional tilt-compensation method assessment

In this section, we begin by presenting the experimental evaluation of the conventional tilt-compensation method, conducted using the MPU9250 9 DoF motion sensor. To perform these experiments, we make use of Relations 1 and 2, and employ a 3D nonmagnetic scaled platform, depicted in Figure 3. We set various yaw angles ranging from 0° to 355°,

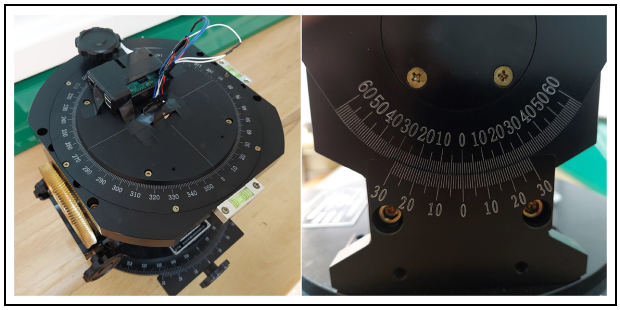


Figure 3. 3D nonmagnetic scaled platform.

in increments of 5°, and for each of these yaw values, we set different pitch angles spanning from −60° to 60°.

The obtained results are illustrated in Figure 4(a) revealing that the yaw estimation error can reach up to ± 30°. These results prompted us to consider the possibility that the inferior outcomes were due to a faulty implementation of the tilt-compensation method. To gain further confirmation, we opted to conduct similar tests using a smartphone equipped with a magnetic compass. The obtained results using a Samsung Galaxy M51 smartphone are shown in Figure 4(b).

Both the MPU9250 and Samsung Galaxy M51 smartphone experiments yielded similar outcomes. Specifically, the yaw estimation error was found to be dependent on the actual yaw, with smaller errors when the sensors are oriented toward the north or south, and more significant errors when pointing toward the east or west. Moreover, the errors were found to be less critical for small tilt values, while they became more significant for greater tilt angles.

Madgwick and Mahony algorithms

To compare the performance of the proposed method, we will also consider two commonly used Attitude and Heading Reference System (AHRS) algorithms: Madgwick and Mahony. These algorithms use measurements from the accelerometer, gyroscope, and magnetometer to estimate the orientation of an object.

Unlike Euler Angles, which can suffer from Gimbal lock issues and ambiguous representation, these algorithms use quaternions as the representation form (Hoang et al., 2021). It is important to note that it is possible to convert between Euler angles and quaternions using appropriate mathematical transformations (Diebel, 2006).

The Madgwick algorithm is a widely used method that uses accelerometer and magnetometer measurements to compensate for errors in the gyroscope measurements. It includes two main stages. First, a correction algorithm is used to align the gyroscope measurements. A quaternion propagation is then applied to the angular velocity to minimize bias and drift errors, starting from the orientation estimated at the previous step.

In the second stage, accelerometer and magnetometer measurements are fused together through the use of a gradient descent algorithm. This fusion process corrects the orientation estimated solely from gyroscope measurements. To

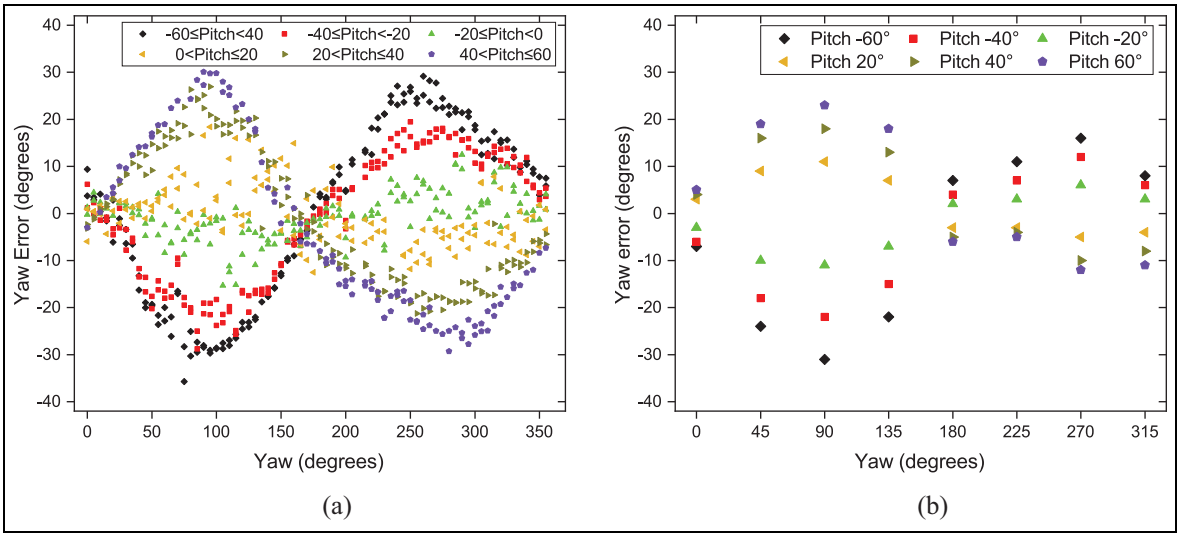


Figure 4. Conventional tilt-compensation method assessment. (a) With MPU 9250. (b) With Samsung Galaxy M51 smartphone.

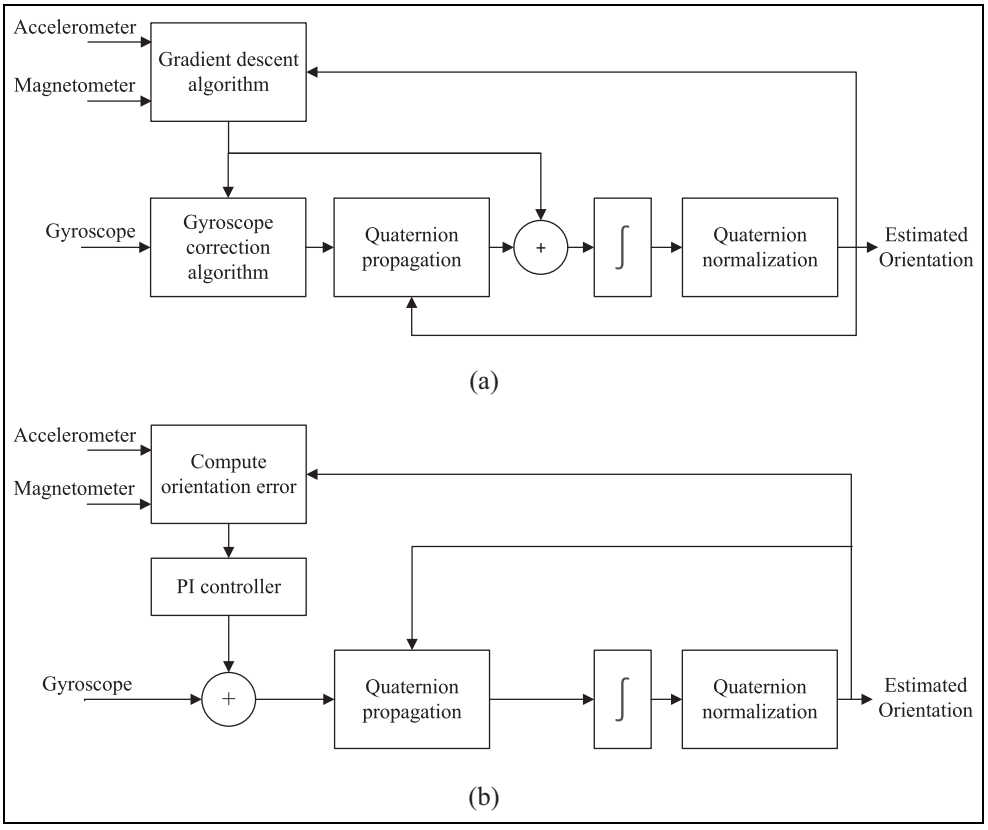


Figure 5. Block diagrams of two complementary algorithms. (a) Madgwick algorithm. (b) Mahony algorithm.

prevent numerical issues arising from finite resolution of numerical representations, the quaternion normalization technique is employed. Figure 5(a) shows the block diagram of the Madgwick algorithm. The detailed formulation of this algorithm can be found by Madgwick et al. (2011).

The Mahony algorithm, which is based on a proportional integral (PI) controller, uses accelerometer and magnetometer measurements in addition to the orientation computed at the previous step to calculate orientation error. This error is then used to correct the measured angular velocity. Similar to the

Madgwick algorithm, the Mahony algorithm also employs quaternion propagation and normalization steps to estimate the orientation. Figure 5(b) illustrates the block diagram of the Mahony algorithm and its detailed formulation is documented by Mahony et al. (2008).

Solution approach

Motivation

The estimation of the yaw angle is a regression problem that involves predicting a continuous response based on a set of inputs. In the literature, various regression techniques have been proposed for this purpose. A comparative study conducted by Zurada et al. (2011) attests that artificial intelligence (AI)–based algorithms outperform traditional methods in this context.

While the terms AI and ML are often used interchangeably, it is important to distinguish between the two. Specifically, ML is a subfield of AI that concerns the development and application of methods and algorithms that enable machines to identify patterns and make predictions. In contrast, AI refers more generally to the capability of machines to emulate human decision-making processes, as described by Ligeza (1995).

In many research fields, several nonlinear ML algorithms have been used to address regression problems. These include well-known techniques such as neural networks, support vector regression (SVR), random forest (RF), and K-nearest neighbor (KNN). Among these, neural networks stand out as primary tools for obtaining reasonable results as demonstrated by the comparative studies presented by Bratsas et al. (2019) and Sekeroglu et al. (2022). Hence, in our work, we have chosen to employ one of the neural network's variations, specifically the ANN.

ANNs are composed of interconnected nodes, also known as artificial neurons, that are designed to emulate the behavior of the human brain. In ANNs, signals are transmitted between neurons and the inputs are processed through a combination of weights and biases.

ANNs have a high potential to learn and model nonlinear and complex relationships between variables. ANNs are also capable of generalizing and making predictions on any set of new data. ML uses two primary learning techniques: unsupervised learning, which finds hidden features in input data, and supervised learning, which uses a known set of input data along with its corresponding output to train models, as noted by da Silva et al. (2016). With this in mind, this study proposes the use of a multilayer perceptron (MLP) ANN regressor for estimating the yaw angle, regardless of whether or not there is tilt. It consists of a feed-forward ANN that processes inputs only in the forward direction, using a supervised learning scheme.

The choice of input data for the proposed ANN was guided by the fundamental principles governing yaw angle measurement. In particular, as indicated by Relation 1, yaw angle depends on the horizontal magnetic field components. According to Relation 2, these components are linked to magnetometer readings (h_x, h_y, and h_z) and the current sensor's tilt, which can be estimated using accelerometer measurements (a_x, a_y, and a_z). Consequently, our proposed ANN uses calibrated magnetometer and accelerometer measurements along the three axes as features, as depicted in Figure 6.

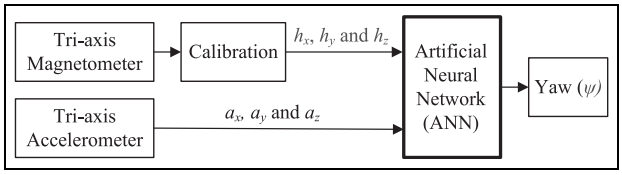


Figure 6. Inputs and outputs of the adopted ANN model.

Dataset collection

To gather the dataset used to train and test the proposed ANN, a cost-effective system was employed. It comprises the MPU9250 9 DoF motion sensor and the Raspberry Pi 3 compute module. This module incorporates a 64-bit Quad-Core processor running at 1.4 GHz and is equipped with 1GB of RAM. Its combination of compact size and powerful processing capabilities has made it a popular choice among developers and for educational applications. *Python* was employed as the programming language.

Using the 3D nonmagnetic scaled platform, we set different yaw values ranging from 0° to 359° with a 1° step. The platform was then arbitrarily rotated along the X and Y axes for each yaw value. The data were collected at a rate of 10 Hz. The collected dataset used for training and testing the proposed ANN consists of 206,300 samples, each of which includes the following data: the actual yaw value read from the 3D nonmagnetic scaled platform, which is used to label the collected features; the calibrated magnetometer measurements; and the accelerometer measurements along the three axes.

As previously referred, other attitude estimation filters can intrinsically solve the problem of tilt compensation in yaw estimation. The Madgwick and Mahony filters are also implemented and their results are recorded for comparison. It is worth mentioning that these two filters are constant gain filters. Therefore, their performance will vary based on the value of the chosen gain. To provide a fair qualitative comparison, we chose the gains that minimize the root mean square error (RMSE) of the estimation for each filter.

ANN architecture and training

The architecture of an ANN can have a significant impact on the network's performance. Input and output layers depend on the number of features and outputs which are in our Case 6 and 1, respectively. However, there is no one-size-fits-all solution for determining the optimal internal architecture, as it depends on the specific problem being addressed and the available data. One common approach is to start with a simple architecture and gradually increase its complexity and explore different combinations of parameters, such as the number of hidden layers and neurons, activation functions,

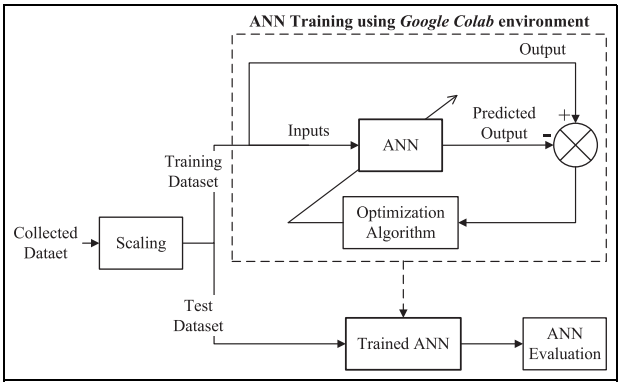


Figure 7. Scheme of ANN training process.

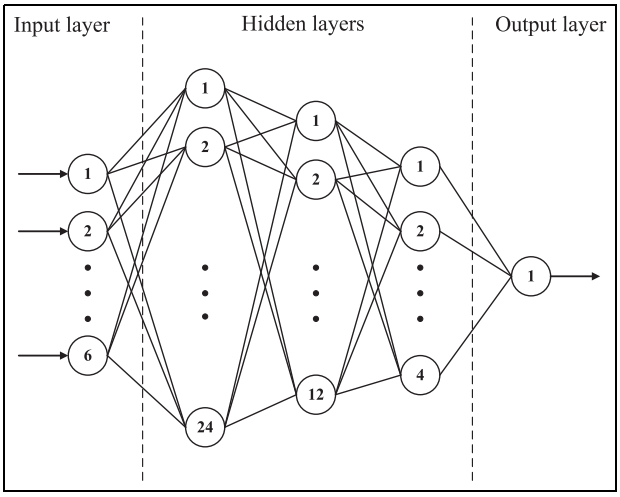


Figure 8. ANN's adopted architecture.

and optimization algorithms until satisfactory performance is achieved.

It is important to consider the trade-off between model complexity and generalization performance. A model that is too simple may underfit the data and results in poor performance. However, a model that is too complex may fit the training data well but may not generalize well to new, unseen data.

Before training the ANN, the collected dataset undergoes a pre-processing stage, where the output is scaled to match the range of the adopted activation function on the output layer (Lange et al., 1997). Also, since accelerometer and magnetometer measurements have different scales, it is preferrable to normalize input variables to ensure that each input feature contributes equally to the learning process, preventing features with larger scales from dominating the learning process (Kuhn and Johnson, 2019; Witten et al., 2016).

In our work, we start with a simple ANN with one hidden layer. Then, we seek for accuracy improvement by varying the size and depth of the ANN. As shown in Figure 7, and using *Google Colab* environment, an optimization algorithm is used to update the ANN's weights and biases according to a suitable metric, known as loss function, which is in our case selected to be the mean squared error (MSE) between the ANN predicted output and the actual output. Table 1 contains some of the obtained results using different ANN architectures.

To detect and avoid overfitting, we have used a common technique, known as k-fold cross-validation. The idea is to partition the training dataset into k subsets, or folds, where onefold is used for validation and the remaining $k-1$ folds are used for training the model. This process is repeated k times, with each fold serving as the validation set exactly once. The final evaluation metric is then computed as the average of the metrics obtained in each fold (da Silva et al., 2016).

We found that increasing the number of hidden layers beyond two did not result in significant improvements in MSE. Meanwhile, the maximum error on both training and test datasets is considerably reduced using three hidden layers. In general, using the same number of neurons for all hidden layers will suffice (da Silva et al., 2016), meanwhile, having a large first layer and following it up with smaller layers has led to better performance in our case. The optimal results were achieved using three hidden layers with 24, 12, and 4 neurons, as depicted in Figure 8. Figure 9 summarizes these results where all parameters have been normalized to be visualized in the same 3D graph.

The properties and performance of ANNs are significantly influenced by activation functions. They are responsible for ensuring the ANN nonlinearity. The most commonly used activation function is the sigmoid function (Bakr and Negm, 2012). Nevertheless, in our work, we have tested several activation functions and the best performance was given by the hyperbolic tangent (tanh) function for all layers.

Table 1. Some of the obtained results using different ANN architectures.

Architecture	Trainable parameters	Training MSE (deg^2)	Test MSE (deg^2)	Training max error (deg)	Test max error (deg)	Number of iterations
[254]	2033	7.97	8.29	64.41	64.54	1000
[12,6]	169	4.72	4.51	132.12	28.03	1000
[24,12]	481	3.98	4.01	33.95	20.00	830
[12,6,4]	195	3.69	3.79	13.79	11.98	610
[12,12,12]	409	3.87	3.90	18.32	8.20	450
[24,12,4]	**525**	**3.73**	**3.79**	**8.79**	**8.72**	**120**
[24,12,4,4]	545	3.68	3.71	9.12	7.58	110

Bold faced correspond to the optimal results achieved using the [24,12,4] architecture.

Table 2. Statistical results and computational time of the four methods.

Method	RMSE (*deg*)	Maximum error (*deg*)	Average execution time (μ**s**)
Conventional	10.42	54.28	5.99
Madgwick	6.10	40.93	130.83
Mahony	7.92	55.74	76.11
ANN	**1.95**	**8.72**	**3578.82**

Bold faced corresponds to the proposed ANN.

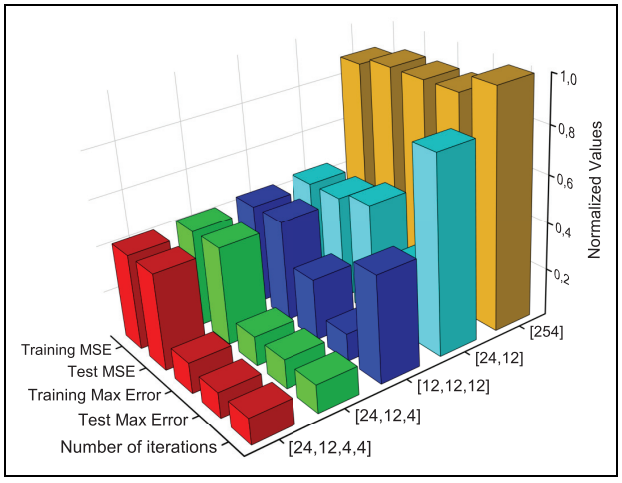


Figure 9. Performance comparison between different ANN architectures.

The tanh function is an activation function that has been shown to perform well in many applications, especially in image and speech recognition. It is a smooth, continuous, and differentiable function that maps its input to a range between -1 and 1. Unlike the sigmoid function, the tanh function has a zero-centered output, which makes it easier for the network to converge during the training process. The tanh function has been shown to be particularly effective in deep learning applications, where deep ANNs with multiple hidden layers are used to learn complex representations of the input data (LeCun et al., 2015). Therefore, the choice of the tanh activation function in our work is not surprising, as we are using an ANN with multiple hidden layers to learn a complex mapping between the sensor measurements and the yaw angle.

For the optimization algorithm, we have used two well-known algorithms: stochastic gradient descent (SGD) and adaptive moment estimation (Adam). In opposition to the classic gradient descent, SGD does not iterate over the full training set to update parameters at each iteration. Instead, it follows the negative gradient of the loss function after seeing only a single or a small set of samples. As such, SGD addresses the issue of the high computational cost. Nevertheless, the main challenge of SGD is the selection of a proper learning rate, where a small rate leads to slow

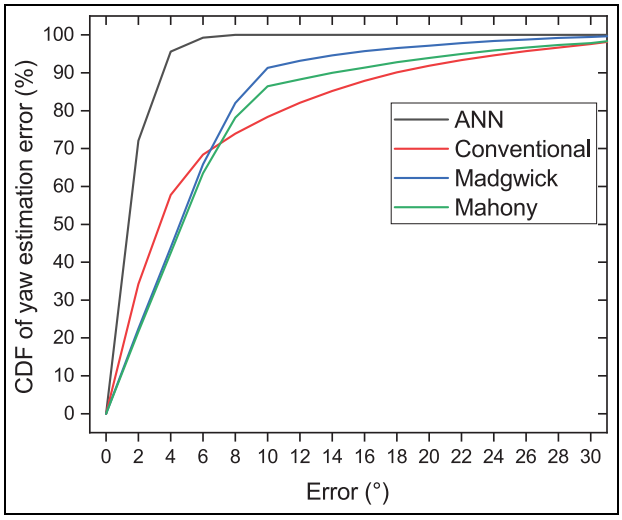


Figure 10. CDFs of yaw estimation error obtained by the four methods.

convergence and a large one causes the loss function to fluctuate around the minimum or even to diverge (Bottou, 2012).

Adam algorithm deals with this issue by computing adaptive learning rates for each parameter. To do so, it uses estimations of the first and second moments of the gradient, which enables Adam to use a larger effective step size (Kingma and Ba, 2015). In our case, both algorithms have reached the same optimum solution, but Adam optimizer's convergence occurred noticeably more quickly.

Results and discussion

Table 2 shows the statistical results for the evaluation of the trained ANN using the test dataset, which represents 20% (41,260 samples) of the collected dataset. The RMSE and the maximum error of the four methods were calculated to evaluate the performance of the trained model.

According to the results, the RMSE of the proposed ANN, which quantifies how spread out are the estimation errors, is relatively small compared to the other methods, indicating that the trained ANN can accurately predict the output values of new, unseen inputs. The maximum measured error is about 8° which means that there are some cases where the prediction is quite far from the actual value. Meanwhile, it represents a noticeable enhancement compared to other methods whose maximum error exceeds 40°.

Figure 10 illustrates the cumulative distribution functions (CDFs) of the yaw estimation error obtained by the different approaches using the testing dataset. Interestingly, more than 70% of error occurrences obtained by the ANN are less than 2°. Conventional tilt-compensation method, Madgwick, and Mahony filters present comparable performances with more than 30% of error occurrences exceeding 5°, as shown in Figure 11, which depicts the yaw estimation error statistics obtained by the four methods.

Ultimately, an analysis was conducted to determine the computational time of the four methods. The trained model

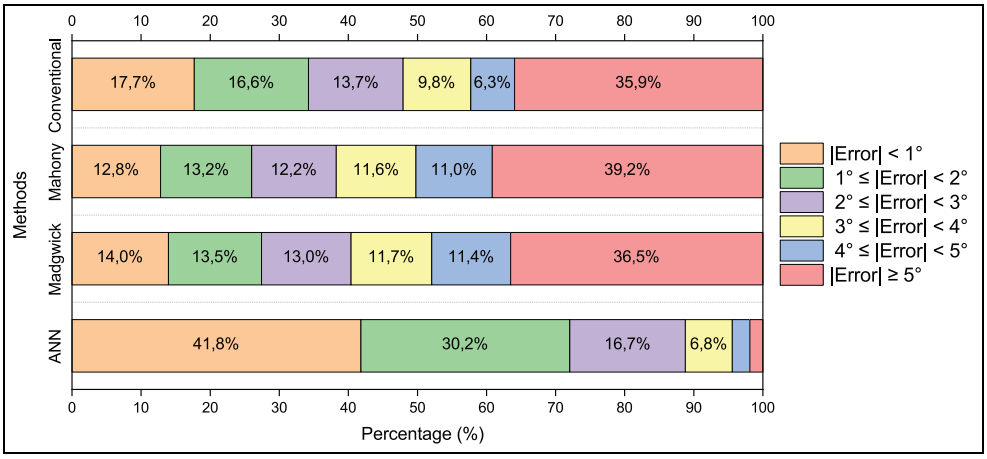


Figure 11. Histograms of the four methods' estimation errors.

was implemented on the Raspberry Pi 3 using *TensorFlow* library. The results of this analysis are presented in Table 2, which displays the corresponding execution times for each of the four methods. It was observed that the proposed ANN requires a significantly longer processing time in comparison to the other three methods. Specifically, the runtime of the ANN was found to be on the order of 3 ms, whereas the execution times for the two complementary filters and the conventional tilt-compensation method were observed to be around 100μs and 10μs, respectively.

To provide a visual representation of these findings, Figure 12 has been included. It illustrates a summary of comparisons among the four methods. Notably, all parameters have been normalized for better visualization in a single 3D graph.

In summary, our proposed ANN exhibits superior accuracy compared to the conventional tilt-compensation method, Madgwick, and Mahony filters. However, this superior performance comes at the cost of a more complex system, resulting in a longer computational time. Consequently, while our method excels in accuracy, it may not be the most efficient choice for applications prioritizing real-time processing or low computational requirements.

Nevertheless, there are still several limitations that need to be addressed to enhance the effectiveness and generalization capabilities of our proposed solution. One of the main challenges in ML-based methods is the availability of large and diverse datasets. While we used a large dataset for training, its lack of diversity may impact the model's performance on unseen scenarios. To tackle this issue, we intend to use a pan-tilt unit for accurate positioning in two axes, enabling precise labeling of data for training in both static and dynamic modes.

While our proposed ANN can estimate tilt based solely on accelerometer data, this approach can provide reliable results only when the sensor is in stationary position or has very low acceleration. However, in other situations, incorporating gyroscope data as inputs becomes necessary. In addition, our work employed a nonmagnetic platform, which does not introduce interference to magnetometer readings. Yet, in real-world applications, magnetic disturbances often arise from nearby materials and electromagnetic systems.

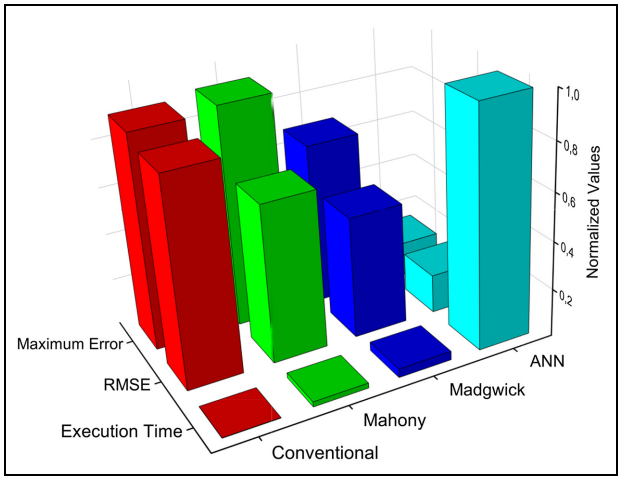


Figure 12. Performance comparison summary.

Conclusion

This paper proposes an ML-based method for tilt compensation in yaw estimation using a low-cost IMU and Raspberry Pi Platform. The paper begins by demonstrating the shortcomings of the conventional tilt-compensation method. The proposed solution consists of an ANN that uses accelerometer and magnetometer measurements as features, without the need for complicated calibration approaches. Moreover, the proposed solution does not require the estimation of pitch and roll angles.

The paper evaluates different ANN architectures and configurations and dedicates particular interest to the dataset collection process. The performance of the proposed solution is compared with that of the conventional tilt-compensation method and other complementary filters (Madgwick and

Mahony). The results show that the proposed solution outperforms the existing solutions in terms of accuracy, with RMSE of 1.95° and over 70% of error occurrences being less than 2°. However, the ANN's average execution time is higher than that of other methods.

As perspectives to this work, we plan to conduct further evaluations of the proposed solution using an UAV to perform dynamic tests. In addition, we plan to implement the proposed ANN on a platform with high-speed parallel data processing capabilities, such as field-programmable gate array (FPGA).

Overall, this paper presents a promising solution for tilt compensation in yaw estimation using low-cost IMU and Raspberry Pi Platform. The proposed solution's performance outperforms existing solutions in terms of accuracy, and further evaluations are planned to test the solution's performance in dynamic scenarios.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work is supported by Directorate General for Scientific Research and Technological Development (DGRSDT) and partially funded by the Thematic Research Agency on Science and Technology (ATRST) of the Algerian Ministry of Higher Education and Scientific Research (MESRS).

ORCID iD

Ali Mounir Halitim https://orcid.org/0000-0002-4780-7173

Data availability statement

The authors confirm that the data supporting the findings of this study are available within the article.

References

Ahuja A, Ferlini A and Mascolo C (2021) Pilotear: Enabling in-ear inertial navigation. In: *Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers*. New York: ACM, pp. 139–145.

Árvai L and Homolya S (2020) Filtering methods for indoor magnetic compass. In: *2020 21th International Carpathian Control Conference (ICCC)*. High Tatras, Slovakia: IEEE, pp. 1–5.

Bakr MH and Negm MH (2012) Modeling and design of high-frequency structures using artificial neural networks and space mapping. In: Deen MJ (ed.) *Advances in Imaging and Electron Physicsadvances in Imaging and Electron Physics: Volume 174*. The Netherlands, Elsevier, pp. 223–260.

Bottou L (2012) Stochastic gradient descent tricks. In: Montavon G, Orr GB and Müller K-R (eds) *Neural Networks: Tricks of the Trade. LECTU: Volume 7700*. Berlin: Springer, pp. 421–436.

Bratsas C, Koupidis K, Salanova JM, et al. (2019) A comparison of machine learning methods for the prediction of traffic speed in urban places. *Sustainability* 12(1): 142.

Carratu M, Iacono SD, LongHoang M, et al. (2019) Energy characterization of attitude algorithms. *2019 IEEE International Conference on Industrial Informatics (INDIN)*, Helsinki, Finland, 22–25 July 2019. IEEE, pp. 1585–1590.

Cirillo A, Cirillo P, De Maria G, et al. (2016) A comparison of multisensor attitude estimation algorithms. In: Fourati H and Belkhiat (eds) *Multisensor Attitude Estimation: Fundamental Concepts and Applications*. London: CRC Press, pp. 529–540.

da Silva IN, Spatti DH, Flauzino RA, et al. (2016) *Artificial Neural Networks: A Practical Course*. Cham: Springer International Publishing.

Diebel J (2006) Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix* 58(15–16): 1–35.

Duymaz E and Işik B (2021) A novel method for tilt compensation in inertial sensor systems. In: *2021 29th Signal Processing and Communications Applications Conference (SIU)*, 9–11 June 2021. IEEE, pp. 1–4.

Ferdinando H, Khoswanto H and Purwanto D (2012) Embedded Kalman filter for inertial measurement unit (IMU) on the ATMega8535. In: *INISTA 2012—International Symposium on INnovations in Intelligent SysTems and Applications*, 2–4 July 2012, pp. 1–5.

Hanafi ME, Abozied MA, Elhalwagy YZ, et al. (2019) Multi-sensory data fusion for high performance attitude estimation. In: *18th International Conference on Aerospace Sciences and Aviation Technology*, Cairo, Egypt, 9–11 April 2019. The Military Technical College, pp. 1–11.

Hoang ML, Iacono SD, Paciello V, et al. (2021) Measurement optimization for orientation tracking based on no motion no integration technique. *IEEE Transactions on Instrumentation and Measurement* 70: 1–10.

Hoang ML, Pietrosanto A, Iacono SD, et al. (2020) Pre-processing technique for compass-less madgwick in heading estimation for industry 4.0. In: *2020—International Instrumentation and Measurement Technology Conference (I2MTC), Proceedings*, Dubrovnik, Croatia, 25–28 May 2020, pp. 1–6. IEEE.

Kingma DP and Ba JL (2015) Adam: A method for stochastic optimization. In: *3rd International Conference on Learning Representations, ICLR—Conference Track Proceedings 2015*, San Diego, CA, 7–9 May.

Kuhn M and Johnson K (2019) *Feature Engineering and Selection: A Practical Approach for Predictive Models*. London: CRC Press.

Lange N, Bishop CM and Ripley BD (1997) *Neural Networks for Pattern Recognition* (Vol. 92). Oxford: Oxford University Press.

Langfelder G and Tocchio A (2013) MEMS integrating motion and displacement sensors. In: Nihtianov S and Luque A (eds) *Smart Sensors and MEMS: Intelligent Devices and Microsystems for Industrial Applications*. Sawston: Woodhead Publishing, pp. 366–401.

LeCun Y, Bengio Y and Hinton G (2015) Deep learning. *Nature* 521(7553): 436–444.

Li J, Shu K and Zhang H (2019) An efficient method for tri-axis magnetometer calibration. In: *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, 19–32 August 2019. IEEE, pp. 654–660.

Ligeza A (1995) *Artificial Intelligence: A Modern Approach* (Vol. 9). London: Pearson Education, Inc.

Long Hoang M and Pietrosanto A (2022) Yaw/heading optimization by machine learning model based on MEMS magnetometer under harsh conditions. *Measurement: Journal of the International Measurement Confederation* 193: 111013.

Madgwick SO, Harrison AJ and Vaidyanathan R (2011) Estimation of IMU and MARG orientation using a gradient descent algorithm. *IEEE International Conference on Rehabilitation Robotics* 25: 32.

Mahony R, Hamel T and Pflimlin JM (2008) Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on Automatic Control* 53(5): 1203–1218.

Poulose A, Kim J and Han DS (2019) Indoor localization with smartphones: Magnetometer calibration. In: *2019 IEEE International Conference on Consumer Electronics (ICCE)*, 11–13 January 2019. IEEE, pp. 1–3.

Renaudin V, Afzal MH and Lachapelle G (2010) Complete triaxis magnetometer calibration in the magnetic domain. *Journal of Sensors* 2010: 967245.

Ru X, Gu N, Shang H, et al. (2022) Mems inertial sensor calibration technology: Current status and future trends. *Micromachines* 13(6): 879.

Sekeroglu B, Ever YK, Dimililer K, et al. (2022) Comparative evaluation and comprehensive analysis of machine learning models for regression problems. *Data Intelligence* 4(3): 620–652.

Wang S, Wang A, Ran M, et al. (2022) Hand gesture recognition framework using a lie group based spatio-temporal recurrent network with multiple hand-worn motion sensors. *Information Sciences* 606: 722–741.

Witten IH, Frank E, Hall MA, et al. (2016) *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, pp. 1–621.

Xu S, Chen R, Yu Y, et al. (2019) Locating smartphones indoors using built-in sensors and wi-fi ranging with an enhanced particle filter. *IEEE Access* 7: 95140–95153.

Zurada J, Levitan AS and Guan J (2011) A comparison of regression and artificial intelligence methods in a mass appraisal context. *Journal of Real Estate Research* 33(3): 349–387.