

University of Dayton

eCommons

Electrical and Computer Engineering Faculty
Publications

Department of Electrical and Computer
Engineering

4-2022

Towards Improved Inertial Navigation by Reducing Errors Using Deep Learning Methodology

Hua Chen

Tarek M. Taha

Vamsy P. Chodavarapu

Follow this and additional works at: https://ecommons.udayton.edu/ece_fac_pub



Part of the [Computer Engineering Commons](#), [Electrical and Electronics Commons](#), [Electromagnetics and Photonics Commons](#), [Optics Commons](#), [Other Electrical and Computer Engineering Commons](#), and the [Systems and Communications Commons](#)

Article

Towards Improved Inertial Navigation by Reducing Errors Using Deep Learning Methodology

Hua Chen *, Tarek M. Taha and Vamsy P. Chodavarapu

Electrical and Computer Engineering, University of Dayton, Dayton, OH 45469, USA;
ttaha1@udayton.edu (T.M.T.); vchodavarapu1@udayton.edu (V.P.C.)

* Correspondence: chenh6@udayton.edu

Abstract: Autonomous vehicles make use of an Inertial Navigation System (INS) as part of vehicular sensor fusion in many situations including GPS-denied environments such as dense urban places, multi-level parking structures, and areas with thick tree-coverage. The INS unit incorporates an Inertial Measurement Unit (IMU) to process the linear acceleration and angular velocity data to obtain orientation, position, and velocity information using mechanization equations. In this work, we describe a novel deep-learning-based methodology, using Convolutional Neural Networks (CNN), to reduce errors from MEMS IMU sensors. We develop a CNN-based approach that can learn from the responses of a particular inertial sensor while subject to inherent noise errors and provide near real-time error correction. We implement a time-division method to divide the IMU output data into small step sizes to make the IMU outputs fit the input format of the CNN. We optimize the CNN approach for higher performance and lower complexity that would allow its implementation on ultra-low power hardware such as microcontrollers. Our results show that we achieved up to 32.5% error improvement in straight-path motion and up to 38.69% error improvement in oval motion compared with the ground truth. We examined the performance of our CNN approach under various situations with IMUs of various performance grades, IMUs of the same type but different manufactured batch, and controlled, fixed, and uncontrolled vehicle motion paths.

Keywords: inertial navigation; inertial measurement unit; MEMS inertial sensors; deep learning



Citation: Chen, H.; Taha, T.M.; Chodavarapu, V.P. Towards Improved Inertial Navigation by Reducing Errors Using Deep Learning Methodology. *Appl. Sci.* **2022**, *12*, 3645. <https://doi.org/10.3390/app12073645>

Academic Editors: Jeng-Shyang Pan, Huanyu Liu, Jun-Bao Li, Meng Li and Shi-Huang Chen

Received: 18 March 2022

Accepted: 2 April 2022

Published: 5 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Micro-Electro-Mechanical-Systems (MEMS) Inertial Measurement Units (IMUs) are instrumental in many applications including smartphones, gaming devices, digital cameras, automobiles, wearable devices, structural health monitoring, energy exploration, and industrial manufacturing [1,2]. In recent years, MEMS IMUs have begun to enter the automotive market for high-precision navigation applications [3,4]. Autonomous vehicles rely on various sensors including cameras, ultrasound sensors, Radio Detection And Ranging (RADAR), Light Detection And Ranging (LIDAR), Inertial Navigation System (INS), and signals including radio frequency/cellular signals and satellite signals to sense and perceive their surroundings, safely navigate, and reach their destination [5–8]. The IMUs form the core component of an INS, which is a self-contained, dead-reckoning navigation system that attains essential motion parameters including position and velocity through the use of a Six-Degrees-of-Freedom (DoF) IMU comprising of a three-axis accelerometer and a three-axis gyroscope. Here, accelerometers measure linear motion along the x, y, and z axes (axial acceleration), while gyroscopes measure rotation (angular velocity) around these axes [9–11].

Global Navigation Satellite System (GNSS), which includes Global Positioning System (GPS), has been the primary means to obtain position and navigation information for most industrial and consumer applications [12–14]. The advantages of GNSS include its ability to provide absolute-navigation information, with long-term accuracy, anywhere in the

world. However, GNSS requires direct line-of-sight to four or more satellites for continuous operation, and thus leads to frequent GNSS signal blockages in many areas including indoor buildings, dense urban places, multi-level parking structures, and areas with thick tree-coverage [15,16]. In most safety critical applications, the INS works in conjunction with a GNSS. Here, the long-term stability of GNSS helps to bound the errors of an INS by using various filtering algorithms [17] (e.g., Complementary Filter [18], Extended Kalman Filter (EKF) [19–22], and Particle Filter (PF) [23,24]).

MEMS inertial sensors are prone to high noise and large uncertainties in their outputs, such as bias, scale factor, nonorthogonalities, drifts, and noise characteristics, etc., thereby limiting their stand-alone applications [25–27]. For instance, MEMS gyroscopes are prone to biases, scale factor and misalignment errors, and noises that result in quadratic errors in velocity and cubic errors in the position computations and thus do not allow for extended periods of navigation. These errors build up over time, thereby corrupting the accuracy of the measurements. Here, the deterministic error sources include the bias, scale factor, nonorthogonalities, etc. [28–30], which are typically removed by specific calibration procedures after experimentation. Stochastic errors occur due to random variations of bias or scale-factor errors over time and are known as bias or scale-factor drifts. There are other arguments that state that bias-scale-factor drift seems to be stochastic due to the lack of observability to disturbance. Drift may also occur because of inherent sensor noise that interferes with the output signals. These errors are nonsymmetric and cannot be compensated by deterministic models. The basic difference between deterministic and stochastic modeling is that in deterministic modeling a relationship must be established between one or more inputs and one or more outputs, whereas in stochastic modeling, there may not be any direct relationship between the inputs and outputs [31–35].

Currently, many academic research groups and companies are working on alternative navigation methods that can provide reliable and accurate aided inertial navigation within GNSS-denied environments, including fusion algorithms to bind the errors of an INS with radio signals, cameras, star-trackers, and Earth's magnetic field [36–38]. Zhang et al. [39] introduced a dual-model solution for GPS/INS during GPS outages, which integrates with Multiple-Decrease Factor Cubature Kalman filter (MDF-CKF) and Random Forest (RF) that can be used for modeling and compensating the velocity and positioning errors. Compared with traditional Artificial Neural Networks (ANNs), MDF-CKF with Random Forest algorithm dual mode has an overall 34.15% improvement in the position accuracy against conventional CKF. In 2019, Choi et al. [40] introduced an ANN model to estimate Center of Mass–Center of Pressure (CoM–CoP) inclination angle (IA) based on signals using an inertial sensor which included accelerometer, gyroscope, and magnetometer. Then, CoM–CoP IA was calculated to obtain horizontal distance which investigates gait stability. The team applied an ANN and a Long Short-Term Memory (LSTM) to improve the algorithm of CoM–CoP IA by using 3D motion analysis system.

Many research groups, including our group, have previously described the use of INS error reduction algorithms by using ANNs, such as Support Vector Machines (SVMs) [40–42] and deep learning techniques [41,43,44] to improve the performance of IMU by reducing its errors which would greatly improve the overall system accuracy and reduce cost when used as part of a sensor-fusion algorithm. Deep learning is a powerful machine learning technology based on ANNs, and expands traditional neural networks to a large scalable network, with a larger number of neurons and hidden layers. As a result, more complicated data-processing applications can be mapped into deep learning networks.

Many research groups have applied different kinds of deep learning networks to the field of autonomous driving including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Deep Reinforcement Learning (DRL) [45]. Varona et al. [46] used CNN to detect road anomalies by using accelerometers from IMU. The group tested with two window sizes for CNN data sampling: an 85 window size (1.9 s) for automatic pothole detection and a 100 window size (2 s) for road roughness classification. Following the work by Varona et al. [46], Baldini et al. [47] applied CNN on the raw data collected

directly from the accelerometers. The team proposed time-frequency representation-based inputs for their CNN approach and transformed the accelerometer data into a spectrogram, and then fed the spectral representation to a CNN. By using this approach, the research group achieved the accuracy of up to 97.2% to detect road anomalies.

Other research groups have proposed the use of RNN which is based on feedforward construction that makes use of sequential information from previous moments [48,49]. It is generally believed that RNN adds memory into the system solution and is useful in recognition of patterns in sequences of data such as text, genomes, the spoken word, and numerical-time-series data. Wang et al. [50] proposed a new CNN- and RNN-based Visual Odometry (VO) application to generate tracking trajectories for uncertain situations in a smart driving car.

Several research groups have also proposed LSTM algorithms for sensor error compensation and sensor fusion [51]. For example, Chen et al. [52] developed a new deep neural network based on VO by using low-cost IMUs. The team replaced the RNN with the LSTM and trained the combined networks with the raw measurements collected from IMUs. By applying LSTMs, the research group demonstrated that the new network achieved better estimated trajectories for nonperiodic motion in highly dynamic condition. Li et al. [53] proposed a sensor fusion architecture with IMU and 2D laser scanner by applying both CNNs and LSTMs. This system consisted of three networks: a CNN-based point cloud feature extraction from two laser scans, an LSTM-based IMU sequence registration, and an LSTM-based data fusion.

Besides sensor fusion application, several research groups have explored methods of IMU denoising using LSTM. Jian et al. [54] proposed an LSTM-based error-modelling methodology to identify the random errors in a gyroscope. The research team configured the LSTM network with five different lengths for input vectors. From the experimental results, the team demonstrated that a single-layer LSTM network can reduce the standard deviation by up to 42.4%, and the attitude errors by up to 52.0%.

Cho et al. [55] first introduced Gated Recurrent Unit (GRU), which is another variant of RNN, for gyroscope noise suppression. GRUs are similar to LSTM, with fewer components than LSTM. With the benefit of fewer parameters, GRUs have fast training speed and relatively better performance for certain tasks. Jiang et al. [56] proposed several LSTM and GRU hybrid network architectures and compared denoising performances along with LSTM-only and GRU-only networks. From the demonstrated results, the team achieved up to 72% decrease in attitude errors by using LSTM-GRU hybrid network. The above-described prior works are based on supervised learning, by which the networks are typically trained by labelled datasets to perform classification and regression tasks. By contrast, with the environment of nonprovided labelled data, an unsupervised learning algorithm is needed. Reinforcement learning (RL) is applied to learn the features from collected data through estimations of trial and error [57–59]. One recent work has focused on applying deep RL to the field of inertial sensors and autonomous driving. Yang and Kuo [60] presented a new sensor fusion system with GPS, IMU, and wheel odometry by applying deep RL. The team proposed an unscented Kalman filter (UKF) which is suitable for solving nonlinear conditions. In addition, the team utilized Model Predictive Control (MPC) in the dynamic vehicle control system. The parameters of MPC were produced by the data collected from GPS, IMU, and wheel odometry, which were trained by using deep RL. The developed deep-RL-based MPC framework achieved estimated travel distance error of 0.82% and a root mean square error of 0.227m in the path tracking. It also demonstrated up to 32.64% of error improvement for the distance by using RL-based MPC compared with traditional MPC.

In this work, we describe a methodology of using a deep learning algorithm that is optimized for implementation on size-, weight-, and power- (SWaP) efficient hardware for low-cost and portable systems. Specifically, the developed algorithm has a simple network structure and helps to minimize the errors of IMU signals. Compared with EKF approaches [19–22], the proposed system does not need external measuring sources such

as GPS or vehicle sensors. EKF approaches require external measuring sources to achieve high accuracy for the processing of IMU signals. Compared with prior PF approaches [23], we applied the method for both high-grade and low-grade IMUs. Compared with an RF [39] approach, our work can reduce the errors of IMU signals from both accelerometers and gyroscopes. In the current work, we were able to apply our algorithm with a higher data-sampling rate compared with prior ANN, CNN, or LSTM approaches [40,45–47,52–56].

The algorithm can learn from the responses of a particular inertial sensor while subject to inherent noise errors and provide near real-time error correction. We implement a time-division method to divide the IMU data into small step sizes. By using this method, we make the IMU outputs fit the input format of CNN. We set a total of 121 levels of accelerations from accelerometers (operational to $\pm 3 \text{ m/s}^2$) and 91 levels of angular velocities from gyroscopes (operational to $\pm 45^\circ/\text{S}$) as outputs to train CNN. Raw datasets are collected from various grades of IMUs and several IMUs of the same grade. We configure three data sizes for the input formations of the networks to examine the performances of CNN for various performance grades of IMUs and IMUs of the same type but different manufacturing batches. The primary objective of this methodology is developing algorithms with higher performance and lower complexity that would allow implementation on ultra-low power artificial intelligence microcontrollers such as the Analog Devices MAX78000.

2. Methods

2.1. IMU Data Sampling

The typical format of an IMU output is a digital data series in the time domain as shown in Figure 1. To make the IMU output fit the input format of a deep learning network, such as a CNN, the IMU data is divided into small step sizes by time division. Within each time step, the IMU output data is approximated as having a constant acceleration for the accelerometer and angular velocity value for the gyroscope. This approximation would hold as long as the time divisions are small enough. Figure 1 shows that the smaller the time divisions, the less change in the acceleration and angular velocity would occur; therefore, the more accurate representation of the acceleration and the angular velocity would be obtained. The IMU data is sampled by using Mathworks MATLAB.

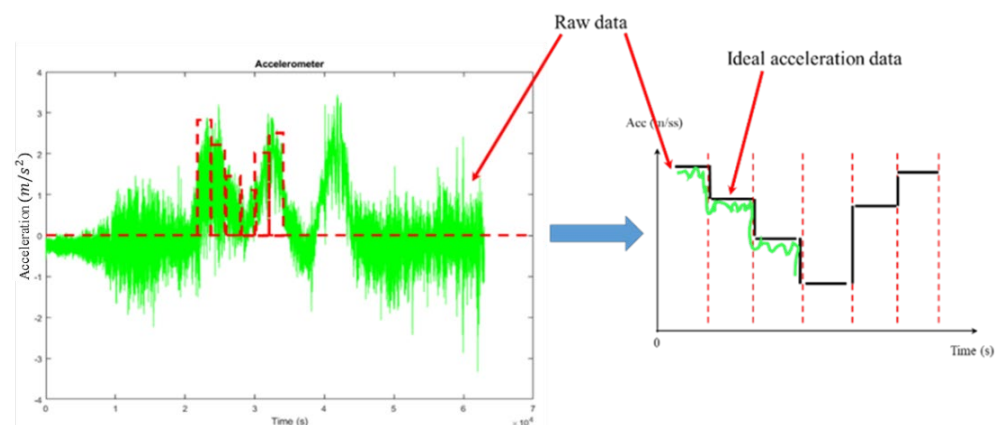


Figure 1. Time–divided IMU outputs.

2.2. CNN Architectures

We developed three CNNs in this work. After converting the sampled IMU data to a vector and normalizing it, the vector was fed to convolutional layers and max-pooling layers. The activation function applied after convolutional layers were ReLU. After extracting the features by convolutional and pooling layers, the vector was flattened and fed to fully connected layers. The network configurations are shown in Figure 2. We used TensorFlow to program, train, and test the CNNs.

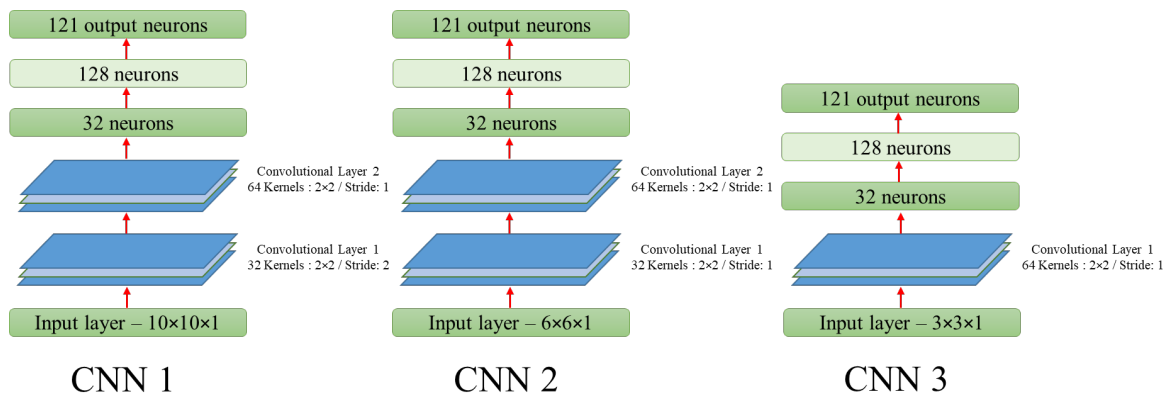


Figure 2. Three network configurations.

2.3. Error Reduction Method by CNN

The raw-data output of an IMU (with both deterministic and random errors) is fed as input to the CNN. Figure 3 shows a block diagram of the new methodology introduced in this work. The IMU raw data is divided into step pieces by time divisions. At each time division, the IMU output data can be approximated as having a near constant acceleration and angular velocity value, as long as the time divisions are small enough. The data piece from each time division is then fed to a trained CNN. By this strategy, the output data plots can be mapped into different acceleration and angular velocity classes. Finally, the network sends out the filtered acceleration and angular velocity classes and those classes are used to compute the actual system position and orientation information.

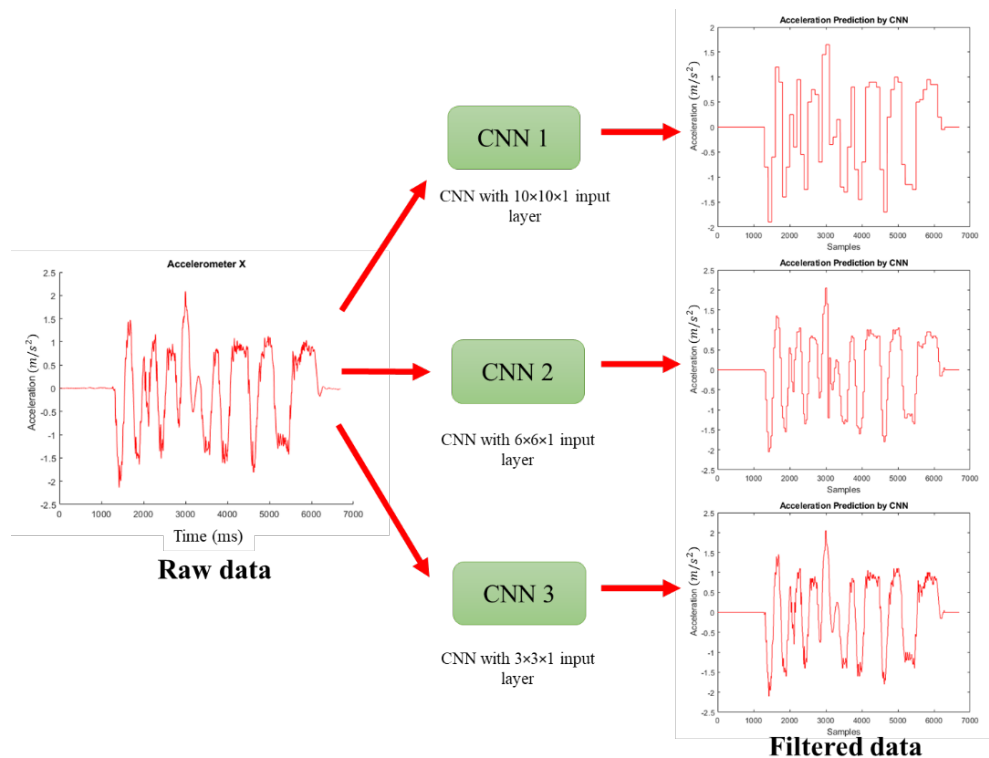


Figure 3. Sampled IMU outputs filtered by Deep Learning implementation.

2.4. Strapdown INS Mechanization

The mechanization process is the step of converting the IMU outputs into navigation information which includes velocity, position, and attitude. In this project, position information is calculated and compared with the ground truth measured by a tape ruler. The navigation information is computed by using the following pseudo steps [1,61].

First, the dynamic transformation matrix is calculated by the equation given as:

$$\dot{R}_b^l = R_b^l \Omega_b^l, \tag{1}$$

where Ω_b^l is the angular velocity matrix measured by the gyroscope and R_b^l is the transformation matrix from the last state. The latest transformation matrix is updated by the dynamic \dot{R}_b^l .

Next, the specific force f^b measured from an accelerometer is transferred into f^l where f^b is the specific force on the body frame and f^l is on the local frame. The computation equation is given as:

$$\dot{v}^l = f^l = R_b^l f^b - (2\Omega_{ie}^l + \Omega_{el}^l)v^l + g^l, \tag{2}$$

where v^l is the dynamic velocity information in the local frame. g^l is the earth's gravity field. Ω_{ie}^l and Ω_{el}^l are the skew-symmetric matrices corresponding to w_{ie}^l and w_{el}^l , where w_{ie}^l is the rotation of the Earth in the local-level frame and w_{el}^l is the transportation rate in terms of the position information from the Earth frame to the local frame. Then, the velocity information is updated by the dynamic \dot{v}^l .

After the velocity information is updated, the position is updated in terms of the displacements along the three axes. The equations are given as:

$$\dot{R}^l = v^l, \tag{3}$$

$$R = \sqrt{R_N + R_E}, \tag{4}$$

where R_N is the displacement on the north axis and R_E is the displacement on the east axis. R is the trajectory result of the motion, as shown in Figure 4.

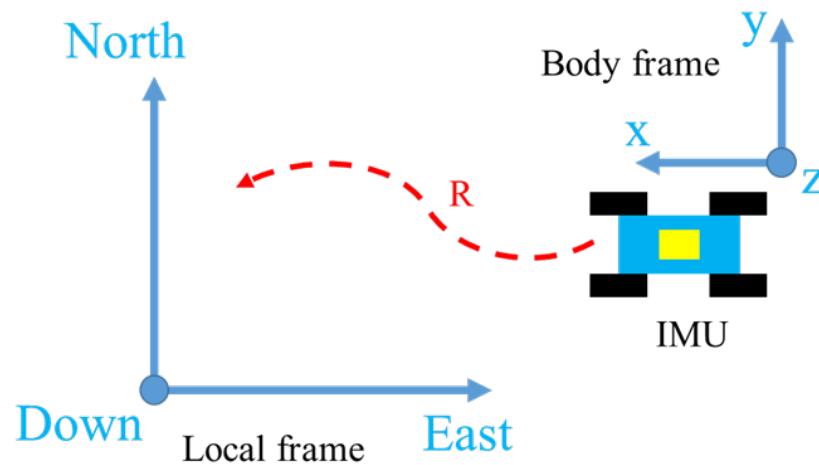


Figure 4. Body frame and local frame of strapdown INS mechanization.

3. Results

To obtain CNN-training and -testing datasets, a linear motion stage and a rotary motor capable of generating fixed, precise accelerations and angular velocities, are incorporated in this work. These IMU units are mounted on the stage (Newmark Systems Inc. Model CS-500-1, Rancho Santa Margarita, CA, USA) or motor (Newmark Systems Inc. Model RM-3-110) and data are collected from the accelerometers and gyroscopes. Two IMU units are used in the project, a high-grade MEMS IMU (Epson Inc. Model M-G364PD, Suwa, Nagano, Japanese) and a low-grade MEMS IMU (TDK InvenSense Inc. Model ICM-20648, San Jose, CA, USA). Table 1 shows the specifications of Epson and TDK IMUs. The breakout boards have USB connectors, and the raw data is collected using software on the computer attached by a USB cable.

Table 1. Specifications EPSON and TDK IMUs.

	Epson	TDK
Gyroscope		
Bias	± 0.1 deg/s	± 5 deg/s
Scale Factor	0.00375 deg/s	16.4 deg/s
Temperature Coefficient	± 0.0005 deg/s/ $^{\circ}$ C	± 0.05 deg/s/ $^{\circ}$ C
Noise Density	0.002 deg/s/ $\sqrt{\text{Hz}}$	0.015 deg/s/ $\sqrt{\text{Hz}}$
Angular Random Walk	0.09 deg/ $\sqrt{\text{Hr}}$	0.9 deg/ $\sqrt{\text{Hr}}$
Accelerometer		
Bias	± 5 mG	± 25 mG
Scale Factor	0.125 mG	2.048 mG
Temperature Coefficient	± 0.02 mG/ $^{\circ}$ C	± 0.80 deg/s/ $^{\circ}$ C
Noise Density	0.06 mG/ $\sqrt{\text{Hz}}$	0.23 mG/ $\sqrt{\text{Hz}}$
Velocity Random Walk	0.025 (m/s)/ $\sqrt{\text{Hr}}$	0.13 (m/s)/ $\sqrt{\text{Hr}}$
Output Data Rate	Up to 2 kHz	Up to 1 kHz

3.1. Train and Test Dataset Collection

For collecting acceleration data, a total of 121 levels of accelerations (0 to ± 3 m/s²) are applied to the linear motion stage, as shown in Figure 5a. These levels formulate 121 different classes for signal classification by the CNN algorithm. The data-collection rates of both EPSON and TDK IMUs are set to 500 Hz. To collect sufficient data samples, the linear stage runs a total of 100 rounds for each level of accelerations. For each round, the collected data is divided into two sets. We divided the collected IMU data, with 70% of the sets used for CNN training, 15% for validation, and 15% of the dataset for testing. The CNN is trained such that it takes raw IMU signals and learns its error patterns with respect to the IMU signals collected from the linear stage, since it generates precise accelerations. Then, the CNN maps the IMU signals into one of the 121 classes while removing these noises from learned knowledge. For collecting angular velocity data, a total of 91 levels of angular velocities (0 to ± 45 deg/s) are applied to the rotary motor, as shown in Figure 5b. The same procedure as the acceleration data is used for the gyroscope data.

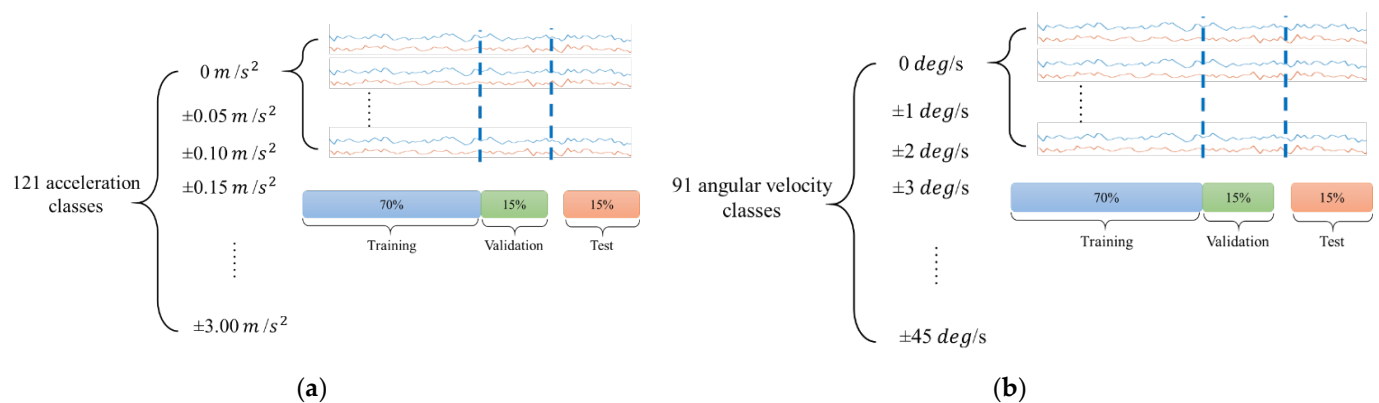


Figure 5. Collections of acceleration and angular velocity levels: (a) 121 acceleration levels collected from IMU; (b) 91 angular velocity levels collected from IMU.

3.2. Train and Test for Experimental Results

After the training and test data are collected, the data are divided into three small step sizes that contain 100, 36, and 9 data points each. To fit each step size of the data, we set different sizes of input layer for the CNN. In other words, we set input layer size of $10 \times 10 \times 1$ when feeding data with a step size of 100, size $6 \times 6 \times 1$ for a step size of 36, and $3 \times 3 \times 1$ for a step size of 9. Table 2 demonstrates the experimental results with different test accuracies by using acceleration data collected from EPSON M-G364PD IMU. From the table, 92.67% test accuracy is obtained with network configurations of

$10 \times 10 \times 1$ input size. Test accuracy decreases to 80.95% when reducing the input data size to $3 \times 3 \times 1$. Table 3 lists the experimental results with testing accuracies by using data collected from TDK ICM-20648 IMU. From the results in Table 3, TDK ICM-20648 IMU achieves relatively higher test accuracy for the $10 \times 10 \times 1$ input layer size, while $3 \times 3 \times 1$ has the lowest test accuracy. When comparing the test accuracies with the same input size of each network, TDK ICM-20648 IMU data shares similar test accuracies but requires more training epochs. The reason for such trends of test accuracies is that the smaller the input-size of the network, the less information in each step size is available to the network for training. By comparing the test accuracies with validation accuracies, the CNN with $6 \times 6 \times 1$ is considered a well-fitted model since both accuracies are closed with each other, while CNNs with $10 \times 10 \times 1$ and $3 \times 3 \times 1$ are over fitted since the test accuracies are less than validation.

Table 2. Test accuracy results for EPSON IMU accelerometer.

Input Step Size	Test Accuracy	Validation Accuracy	Training Epoch
$10 \times 10 \times 1$	92.67%	94.23%	320,000
$6 \times 6 \times 1$	88.88%	89.35%	870,000
$3 \times 3 \times 1$	80.95%	84.10%	880,000

Table 3. Test accuracy results for TDK IMU accelerometer.

Input Step Size	Test Accuracy	Validation Accuracy	Training Epoch
$10 \times 10 \times 1$	91.61%	93.46%	444,000
$6 \times 6 \times 1$	89.19%	89.26%	840,000
$3 \times 3 \times 1$	83.77%	85.93%	1,190,000

Tables 4 and 5 demonstrate all experimental results with different test accuracies by using angular velocity data collected from IMU units. From the table, up to 98.90% test accuracy is obtained with network configurations of $10 \times 10 \times 1$ input size. Test accuracies decrease much less than those test accuracies for acceleration. Most of the test accuracies for angular velocities are more than 90%, while some of the test accuracies for accelerations are less than 90% for accelerations.

Table 4. Test accuracy results for EPSON IMU gyroscope.

Input Size	Test Accuracy	Validation Accuracy	Training Epoch
$10 \times 10 \times 1$	98.90%	99.84%	240,000
$6 \times 6 \times 1$	96.13%	98.58%	650,000
$3 \times 3 \times 1$	95.72%	97.14%	755,000

Table 5. Test accuracy results for TDK IMU gyroscope.

Input Size	Test Accuracy	Validation Accuracy	Training Epoch
$10 \times 10 \times 1$	97.93%	99.70%	250,000
$6 \times 6 \times 1$	96.10%	97.35%	660,000
$3 \times 3 \times 1$	94.14%	96.20%	770,000

3.3. Network Application

To evaluate the performance of the proposed methodology, three CNNs (input sizes of $10 \times 10 \times 1$, $6 \times 6 \times 1$, and $3 \times 3 \times 1$) are applied to filter the data collected from two experimental setups: (1) a remote-control (RC) car (see Figure 6a) on a relatively straight path, but the motion path is uncontrolled; and (2) a toy train moving on an oval path, but using a controlled motion path (see Figure 6b). The IMU unit is mounted on the car body and the data-collection laptop is placed on a metal stand placed on top of the car

body. Cardboard barriers are placed to allow the car to run in a relatively straight path. The data-collection experiment is repeated at least ten times. The raw-data outputs from accelerometers in the IMU are collected and sent to the three networks mentioned above. Finally, the distances are computed by integrating the filtered acceleration results from each network. For the train, the IMU is mounted on top of one of the carriages driven by the battery-powered locomotive.

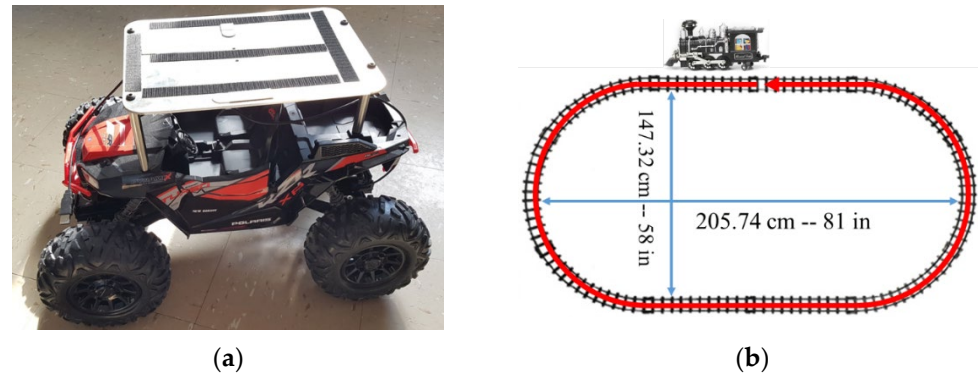


Figure 6. Experimental setups. (a) Application of the remote-control car. (b) Application of the toy train and track.

3.4. Experimental Results for the Remote-Control (RC) Car

The experiment with the RC car is repeated ten times and the IMU data are collected from both the ESPON and TDK IMUs. For each run, the RC car takes around 3.5 s along the straight path. For both the ESPON and TDK IMU, no calibration method such as a six-position static test is conducted prior to the experiments. Ground truth distances are measured by using a tape ruler. The error bars are the differences between computed distances and ground truths. IMU raw distances are computed by integrating the raw acceleration data collected from IMU. The other distances are computed by using the filtered acceleration data from the CNN algorithms. Tables 6 and 7 demonstrate all experimental results with ground truth distances and error percentages.

Table 6. Distance and percentage error results for EPSON IMU.

Trial	Ground Truth Distance (m)	IMU Raw Error vs. Ground Truth	$10 \times 10 \times 1$ CNN Error vs. Ground Truth	$6 \times 6 \times 1$ CNN Error vs. Ground Truth	$3 \times 3 \times 1$ CNN Error vs. Ground Truth
1	2.74	22.26%	12.04%	23.36%	23.36%
2	2.18	27.52%	0.92%	31.19%	31.65%
3	1.28	15.63%	24.22%	2.34%	15.63%
4	2.15	33.02%	13.49%	27.91%	31.63%
5	2.02	103.47%	38.12%	55.45%	70.30%
6	1.78	37.08%	3.93%	48.31%	41.01%
7	3.21	3.43%	3.74%	9.97%	6.23%
8	3.39	48.67%	21.53%	50.74%	57.82%
9	2.25	10.67%	14.67%	1.78%	1.78%
10	2.11	22.27%	9.00%	32.23%	28.91%
Average error		32.40%	14.17%	28.33%	30.83%

For the EPSON M-G364PD IMU, an average error of 14.17% is obtained for the CNN with $10 \times 10 \times 1$ input layer, while 28.33% is obtained for $6 \times 6 \times 1$ and 30.83% for $3 \times 3 \times 1$. The error percentages are computed by comparing with the ground truth distances which are measured by using tape ruler. From the results of average error percentages, the CNN with the $10 \times 10 \times 1$ input layer achieves the highest performance of error reduction. For the TDK ICM-20648 IMU, larger average error percentages are obtained for each CNN. Especially, the CNN with $10 \times 10 \times 1$ shows the average percentage error up to 169.86%, where the distance results are far more off than the ground truth. For the CNN with

$6 \times 6 \times 1$, we obtained the better average percentage of 62.30% while the $3 \times 3 \times 1$ CNN obtains 66.19%. For TDK ICM-20648 IMU, the results show that the CNN with the input layer of $6 \times 6 \times 1$ is the most optimized option, as there are more dynamic and systematic errors contained in the data collected from TDK IMU. In conclusion, we believe that the most optimized input-layer size for CNN is that of $6 \times 6 \times 1$, as it is best suited for many general applications. We achieved up to 32.5% percentage error improvement in the condition of straight-path motion for TDK ICM-20648 IMU.

Table 7. Distance and percentage error results for TDK IMU.

Trial	Ground Truth Distance (m)	IMU Raw Error vs. Ground Truth	$10 \times 10 \times 1$ CNN Error Vs. Ground Truth	$6 \times 6 \times 1$ CNN Error vs. Ground Truth	$3 \times 3 \times 1$ CNN Error vs. Ground Truth
1	2.59	3.86%	172.59%	19.69%	18.15%
2	2.78	87.41%	170.14%	39.21%	42.81%
3	2.9	113.10%	161.03%	61.38%	47.93%
4	2.67	88.39%	188.76%	64.04%	72.66%
5	2.77	61.73%	100.00%	46.93%	24.19%
6	2.62	152.29%	140.46%	92.37%	121.37%
7	2.61	124.52%	205.75%	52.49%	90.80%
8	2.96	137.84%	217.91%	127.03%	125.68%
9	2.83	100.71%	198.23%	71.02%	59.01%
10	2.97	78.11%	143.77%	48.82%	59.26%
Average error		94.80%	169.86%	62.30%	66.19%

3.5. Experimental Results for the Toy Train

The experiment with the train was conducted ten times. In each experimental run, the train makes three roundtrips and 60 s along the oval track to obtain better repeatability in data collection. No calibration method such as six-position static test is conducted prior to the experiments. The distance results of the motions and the percentage errors are calculated by using the same methods as with the RC car.

Train tracks are drawn using IMU raw data, $10 \times 10 \times 1$, $6 \times 6 \times 1$, and $3 \times 3 \times 1$ CNN, as shown in Figures 7 and 8. As shown in Figure 7, the train track drawn with EPSON IMU raw data (red trajectory) is much more erroneous. Compared with the red trajectory, train tracks by $10 \times 10 \times 1$ (pink), $6 \times 6 \times 1$ (green) CNN are much closer to the ground truth. As shown in Figure 8, the train track drawn with TDK IMU raw data (red trajectory) is significantly erroneous. Train tracks by $10 \times 10 \times 1$ (pink), $6 \times 6 \times 1$ (green) CNN are much closer to the ground truth.

For EPSON M-G364PD IMU, shown in Tables 8 and 9, 3.21% of the average error percentage is obtained for the CNN with $10 \times 10 \times 1$ input layer, while 5.84% is obtained for $6 \times 6 \times 1$ and 12.20% for $3 \times 3 \times 1$, while the average percentage error for IMU raw data is 44.53%. For TDK ICM-20648 IMU, 12.77% of the average error percentage is obtained for the CNN with $10 \times 10 \times 1$ input layer, while 6.75% is obtained for $6 \times 6 \times 1$ and 11.25% for $3 \times 3 \times 1$ and 42.69% for IMU raw data is 44.53%. Tables 8 and 9, show the more stable distance results and less average percentage errors are obtained for the controlled path. By comparing the average percentage errors between the RC car and the toy train, we find that the distances calculated by using CNN-filtered data are much closer to the ground truth, since fewer average percentage errors are achieved. Furthermore, the network input layer size of $6 \times 6 \times 1$ is still the best network configuration with the least average percentage error. We achieved up to 38.69% percentage error improvement in the condition of oval motion for TDK ICM-20648 IMU as well.

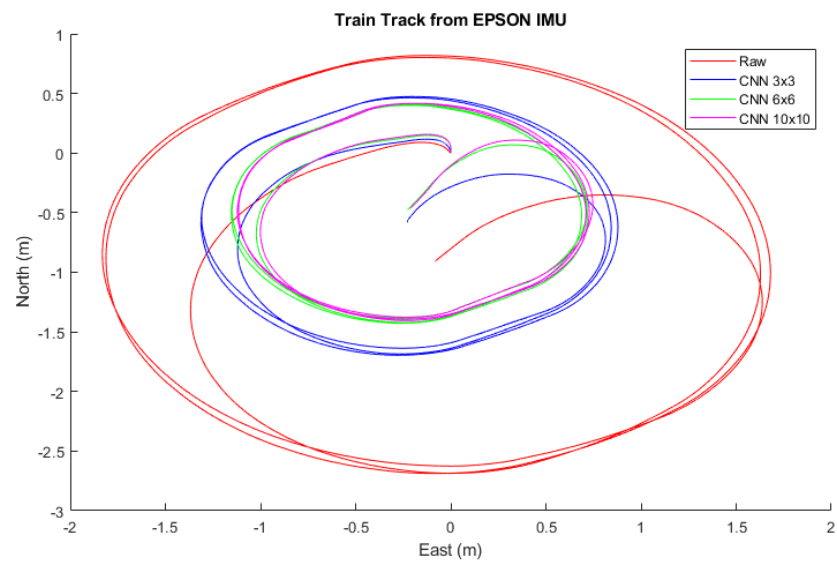


Figure 7. Two-Dimensional Train tracks drawn from EPSON IMU raw data, $10 \times 10 \times 1$, $6 \times 6 \times 1$, and $3 \times 3 \times 1$ CNN.

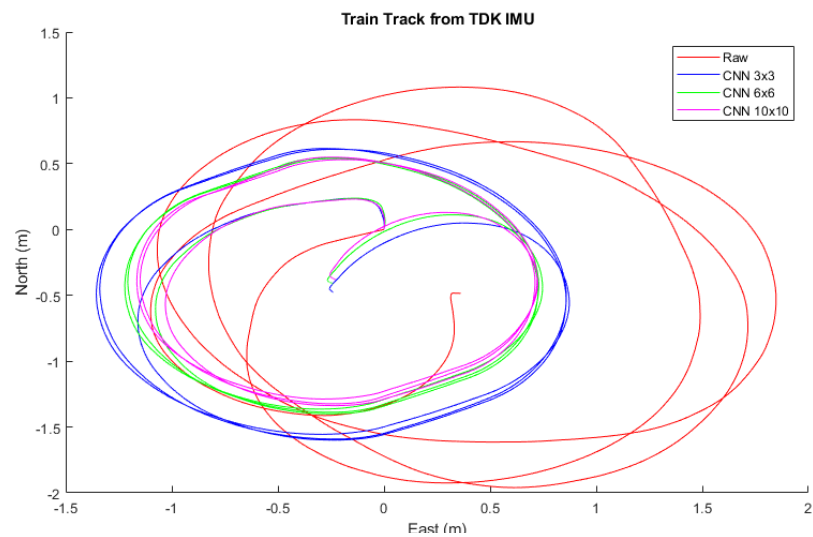


Figure 8. Two-Dimensional Train tracks drawn from TDK IMU raw data, $10 \times 10 \times 1$, $6 \times 6 \times 1$, and $3 \times 3 \times 1$ CNN.

Table 8. Distance and percentage error results for EPSON IMU.

Trial	Ground Truth Distance (m)	IMU Raw Error vs. Ground Truth	$10 \times 10 \times 1$ CNN Error vs. Ground Truth	$6 \times 6 \times 1$ CNN Error vs. Ground Truth	$3 \times 3 \times 1$ CNN Error vs. Ground Truth
1	15.89	43.80%	4.53%	7.11%	13.72%
2	15.92	43.28%	5.15%	6.16%	11.62%
3	16.20	40.37%	1.30%	1.60%	6.17%
4	15.22	43.36%	2.96%	5.19%	12.16%
5	15.91	45.76%	5.03%	8.49%	17.35%
6	16.15	47.12%	4.77%	6.13%	14.24%
7	15.78	47.85%	3.42%	6.59%	14.58%
8	16.22	43.34%	3.21%	2.10%	6.91%
9	15.76	44.86%	0.51%	7.80%	13.71%
10	15.68	45.54%	1.28%	7.21%	11.54%
Average error		44.53%	3.21%	5.84%	12.20%

Table 9. Distance and percentage error results for TDK IMU.

Trial	Ground Truth Distance (m)	IMU Raw Error vs. Ground Truth	10 × 10 × 1 CNN Error vs. Ground Truth	6 × 6 × 1 CNN Error vs. Ground Truth	3 × 3 × 1 CNN Error vs. Ground Truth
1	15.89	48.52%	16.43%	11.77%	13.78%
2	15.92	43.22%	11.56%	7.79%	12.19%
3	16.20	40.25%	12.84%	5.74%	7.53%
4	15.22	45.73%	18.00%	7.10%	8.94%
5	15.91	41.04%	13.45%	10.25%	13.95%
6	16.15	35.05%	10.90%	3.34%	10.65%
7	15.78	43.85%	9.82%	1.77%	13.05%
8	16.22	40.14%	12.33%	6.84%	8.57%
9	15.76	44.42%	11.48%	6.41%	12.31%
10	15.68	44.71%	10.84%	6.51%	11.54%
Average error		42.69%	12.77%	6.75%	11.25%

4. Discussion

By comparing our results with prior research, we found that a higher error-reduction improvement was obtained for the position calculation compared with PF [23], ANN [40], or RL [59] approaches (PF: 29% [23], ANN: 15% [40], and RL: 32.64% [59]). We achieved relatively closed error improvement compared with one RF [39] LSTM [52] approach (RF: 40.08% [39], LSTM: 40% [52]). However, the RF approach from Zhang et al. [39] works for only accelerometers, and the LSTM approach from Chen et al. [52] was applied to low-grade IMUs with a lower data-sampling frequency (10Hz) over a short range (2 m). Compared to EKF [22], SVM [42], RNN [50], and GRU [56], our error reduction was lower for the position (38.69%) than the improvements achieved with it (EKF: 65% [22], SVM: 96% [42], RNN: 64.8% [50], and GRU: 72.0% [56]). The EKF approach from Xu et al. [22] achieved up to 65% error reduction for the position, but the system needs external measuring sources such as GPS, which is not suitable for GPS-denied situations. The SVM approach from Xu et al. [42] achieved up to 96% error reduction, but their system also mostly relies on GPS. The RNN approach from Wang et al. [50] achieved up to 64.8% error reduction. However, their system requires aid from external cameras. The GRU approach from Jiang et al. [56] achieved up to 72.0% error improvement, but their system focuses on gyroscopes' denoising only. These results show that the proposed CNN methodology is a useful tool for the error-reduction method for MEMS IMUs. The CNN-training and -test results show the useful capability of data regression and classification.

5. Conclusions

This paper examined the extended design of an error-reduction method for MEMS IMUs. Compared with traditional error-removal processes, our approach succeeded in removing errors from various grades of IMUs applicable to consumer and industrial applications. We achieved a test accuracy of 92.67% in correctly identifying the accelerometers for the high-grade IMU and 91.61% for the low-grade IMU. Meanwhile we achieved a test accuracy of 98.90% in the high-grade IMU's gyroscopes and 97.93% in the low-grade IMU's gyroscopes. In addition, we achieved up to 32.5% percentage error improvement in the condition of straight-path motion and up to 38.69% percentage error improvement in the condition of oval motion compared with the ground truth. This study shows that CNNs are capable of reducing both systematic and stochastic errors for the IMUs without the aid of external measurement sources such as GNSS over short time durations. Furthermore, the results of this study could be implemented in an ultra-low power hardware to remove the errors of IMUs in real time.

Author Contributions: Conceptualization, H.C., T.M.T. and V.P.C.; methodology, H.C., T.M.T. and V.P.C.; software, H.C.; validation, H.C.; formal analysis, H.C. and V.P.C.; investigation, H.C. and V.P.C.; resources, T.M.T. and V.P.C.; data curation, H.C.; visualization, H.C.; supervision, V.P.C.; project administration, V.P.C.; writing—original draft preparation, H.C.; writing—review and editing, T.M.T. and V.P.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the School of Engineering at University of Dayton.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Research data presented in this article are available on request from the corresponding author with appropriate justification.

Acknowledgments: The authors would like to thank Priyanka Aggarwal for her help with explaining IMU mechanization and calibration methods.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Titterton, D.; Weston, J.L.; Weston, J. *Strapdown Inertial Navigation Technology*; IET: London, UK, 2004; ISBN 978-0-86341-358-2.
2. Zhao, W.; Cheng, Y.; Zhao, S.; Hu, X.; Rong, Y.; Duan, J.; Chen, J. Navigation Grade MEMS IMU for A Satellite. *Micromachines* **2021**, *12*, 151. [[CrossRef](#)] [[PubMed](#)]
3. Gonzalez, R.; Dabove, P. Performance Assessment of an Ultra Low-Cost Inertial Measurement Unit for Ground Vehicle Navigation. *Sensors* **2019**, *19*, 3865. [[CrossRef](#)] [[PubMed](#)]
4. Kwon, S.-G.; Kwon, O.-J.; Kwon, K.-R.; Lee, S.-H. UWB and MEMS IMU Integrated Positioning Algorithm for a Work-Tool Tracking System. *Appl. Sci.* **2021**, *11*, 8826. [[CrossRef](#)]
5. Lindner, L.; Sergiyenko, O.; Rivas-López, M.; Valdez-Salas, B.; Rodríguez-Quirón, J.C.; Hernández-Balbuena, D.; Flores-Fuentes, W.; Tyrsa, V.; Barrera, M.; Muerrieta-Rico, F.N.; et al. Machine Vision System for UAV Navigation. In Proceedings of the 2016 International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles International Transportation Electrification Conference (ESARS-ITEC), Toulouse, France, 2–4 November 2016; pp. 1–6.
6. Lindner, L.; Sergiyenko, O.; Rivas-López, M.; Ivanov, M.; Rodríguez-Quirón, J.C.; Hernández-Balbuena, D.; Flores-Fuentes, W.; Tyrsa, V.; Muerrieta-Rico, F.N.; Mercorelli, P. Machine Vision System Errors for Unmanned Aerial Vehicle Navigation. In Proceedings of the 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE), Edinburgh, UK, 19–21 June 2017; pp. 1615–1620.
7. Ivanov, M.; Sergiyenko, O.; Tyrsa, V.; Lindner, L.; Flores-Fuentes, W.; Rodríguez-Quirón, J.C.; Hernandez, W.; Mercorelli, P. Influence of Data Clouds Fusion from 3D Real-Time Vision System on Robotic Group Dead Reckoning in Unknown Terrain. *IEEE/CAA J. Autom. Sin.* **2020**, *7*, 368–385. [[CrossRef](#)]
8. Borodacz, K.; Szczepanski, C.; Popowski, S. Review and Selection of Commercially Available IMU for a Short Time Inertial Navigation. *Aircr. Eng. Aerosp. Technol.* **2022**, *94*, 45–59. [[CrossRef](#)]
9. Curey, R.K.; Ash, M.E.; Thielman, L.O.; Barker, C.H. Proposed IEEE Inertial Systems Terminology Standard and Other Inertial Sensor Standards. In Proceedings of the PLANS 2004. Position Location and Navigation Symposium (IEEE Cat. No.04CH37556), Monterey, CA, USA, 26–29 April 2004; pp. 83–90.
10. IEEE Standard for Inertial Sensor Terminology. *IEEE Std 528-2001*; IEEE: Piscataway, NJ, USA, 2001; pp. 1–26. [[CrossRef](#)]
11. Liu, J.; Liu, X.; Yang, W.; Pan, S. Investigating the Survey Instrument for the Underground Pipeline with Inertial Sensor and Dead Reckoning Method. *Rev. Sci. Instrum.* **2021**, *92*, 025112. [[CrossRef](#)]
12. Ashman, B.W.; Parker, J.J.K.; Bauer, F.H.; Esswein, M. Exploring the Limits of High Altitude Gps for Future Lunar Missions. In Proceedings of the Guidance, Navigation, and Control 2018, Pts I-II: Advances in the Astronautical Sciences; Walker, C.A.H., Ed.; Univelt Inc.: San Diego, CA, USA, 2018; Volume 164, pp. 491–504.
13. Capuano, V.; Basile, F.; Botteron, C.; Farine, P.-A. GNSS-Based Orbital Filter for Earth Moon Transfer Orbits. *J. Navig.* **2016**, *69*, 745–764. [[CrossRef](#)]
14. Yang, G.; Meng, W.; Lei, L.; Huan, C.; Qian, Z. GNSS receiver techniques based on high earth orbit spacecraft. *Chin. Space Sci. Technol.* **2017**, *37*, 101–109. [[CrossRef](#)]
15. Yang, J.; Wang, X.; Shen, L.; Chen, D. Availability Analysis of GNSS Signals above GNSSs Constellation. *J. Navig.* **2021**, *74*, 446–466. [[CrossRef](#)]
16. Zidan, J.; Adegoke, E.I.; Kampert, E.; Birrell, S.A.; Ford, C.R.; Higgins, M.D. GNSS Vulnerabilities and Existing Solutions: A Review of the Literature. *IEEE Access* **2021**, *9*, 153960–153976. [[CrossRef](#)]
17. Strachan, V.F. Inertial Measurement Technology in the Satellite Navigation Environment. *J. Navig.* **2000**, *53*, 247–260. [[CrossRef](#)]
18. Suh, Y.S. Attitude Estimation Using Inertial and Magnetic Sensors Based on Hybrid Four-Parameter Complementary Filter. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 5149–5156. [[CrossRef](#)]
19. Luo, J.; Fan, Y.; Jiang, P.; He, Z.; Xu, P.; Li, X.; Yang, W.; Zhou, W.; Ma, S. Vehicle Platform Attitude Estimation Method Based on Adaptive Kalman Filter and Sliding Window Least Squares. *Meas. Sci. Technol.* **2021**, *32*, 035007. [[CrossRef](#)]
20. Sabzevari, D.; Chatraei, A. INS/GPS Sensor Fusion Based on Adaptive Fuzzy EKF with Sensitivity to Disturbances. *IET Radar Sonar Navig.* **2021**, *15*, 1535–1549. [[CrossRef](#)]
21. Yang, C.; Shi, W.; Chen, W. Correlational Inference-Based Adaptive Unscented Kalman Filter with Application in GNSS/IMU-Integrated Navigation. *GPS Solut.* **2018**, *22*, 100. [[CrossRef](#)]

22. Xu, Q.; Li, X.; Chan, C.-Y. Enhancing Localization Accuracy of MEMS-INS/GPS/In-Vehicle Sensors Integration During GPS Outages. *IEEE Trans. Instrum. Meas.* **2018**, *67*, 1966–1978. [[CrossRef](#)]
23. Luo, J.; Zhang, C.; Wang, C. Indoor Multi-Floor 3D Target Tracking Based on the Multi-Sensor Fusion. *IEEE Access* **2020**, *8*, 36836–36846. [[CrossRef](#)]
24. Ji, M.; Liu, J.; Xu, X.; Lu, Z. The Improved 3D Pedestrian Positioning System Based on Foot-Mounted Inertial Sensor. *IEEE Sens. J.* **2021**, *21*, 25051–25060. [[CrossRef](#)]
25. Passaro, V.M.N.; Cuccovillo, A.; Vaiani, L.; De Carlo, M.; Campanella, C.E. Gyroscope Technology and Applications: A Review in the Industrial Perspective. *Sensors* **2017**, *17*, 2284. [[CrossRef](#)]
26. Li, X.; Hu, J.; Liu, X. A High-Performance Digital Interface Circuit for a High-Q Micro-Electromechanical System Accelerometer. *Micromachines* **2018**, *9*, 675. [[CrossRef](#)]
27. Zhang, T.; Huang, Y.; Li, H.; Wang, S.; Guo, X.; Liu, X. An Iterative Optimization Method for Estimating Accelerometer Bias Based on Gravitational Apparent Motion with Excitation of Swinging Motion. *Review of Scientific Instruments* **2019**, *90*, 015102. [[CrossRef](#)] [[PubMed](#)]
28. Wu, Y.; Zhu, H.-B.; Du, Q.-X.; Tang, S.-M. A Survey of the Research Status of Pedestrian Dead Reckoning Systems Based on Inertial Sensors. *Int. J. Autom. Comput.* **2019**, *16*, 65–83. [[CrossRef](#)]
29. Han, S.; Meng, Z.; Omisore, O.; Akinyemi, T.; Yan, Y. Random Error Reduction Algorithms for MEMS Inertial Sensor Accuracy Improvement—A Review. *Micromachines* **2020**, *11*, 1021. [[CrossRef](#)] [[PubMed](#)]
30. Qureshi, U.; Golnaraghi, F. An Algorithm for the In-Field Calibration of a MEMS IMU. *IEEE Sens. J.* **2017**, *17*, 7479–7486. [[CrossRef](#)]
31. Narasimhappa, M.; Mahindrakar, A.D.; Guizilini, V.C.; Terra, M.H.; Sabat, S.L. MEMS-Based IMU Drift Minimization: Sage Husa Adaptive Robust Kalman Filtering. *IEEE Sens. J.* **2020**, *20*, 250–260. [[CrossRef](#)]
32. Sun, J.; Xu, X.; Liu, Y.; Zhang, T.; Li, Y. FOG Random Drift Signal Denoising Based on the Improved AR Model and Modified Sage-Husa Adaptive Kalman Filter. *Sensors* **2016**, *16*, 1073. [[CrossRef](#)]
33. Wang, D.; Dong, Y.; Li, Z.; Li, Q.; Wu, J. Constrained MEMS-Based GNSS/INS Tightly Coupled System With Robust Kalman Filter for Accurate Land Vehicular Navigation. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 5138–5148. [[CrossRef](#)]
34. Narasimhappa, M.; Mahindrakar, A.D.; Guizilini, V.C.; Terra, M.H.; Sabat, S.L. An Improved Sage Husa Adaptive Robust Kalman Filter for De-Noising the MEMS IMU Drift Signal. In Proceedings of the 2018 Indian Control Conference (ICC), Kanpur, India, 4–6 January 2018; pp. 229–234.
35. Cong, L.; Yue, S.; Qin, H.; Li, B.; Yao, J. Implementation of a MEMS-Based GNSS/INS Integrated Scheme Using Supported Vector Machine for Land Vehicle Navigation. *IEEE Sens. J.* **2020**, *20*, 14423–14435. [[CrossRef](#)]
36. Tian, Y.; Denby, B.; Ahriz, I.; Roussel, P.; Dreyfus, G. Hybrid Indoor Localization Using GSM Fingerprints, Embedded Sensors and a Particle Filter. In Proceedings of the 2014 11th International Symposium on Wireless Communications Systems (ISWCS), Barcelona, Spain, 26–29 August 2014; pp. 542–547.
37. Sabatini, A.M. Estimating Three-Dimensional Orientation of Human Body Parts by Inertial/Magnetic Sensing. *Sensors* **2011**, *11*, 1489–1525. [[CrossRef](#)]
38. Suh, Y.S. Simple-Structured Quaternion Estimator Separating Inertial and Magnetic Sensor Effects. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *55*, 2698–2706. [[CrossRef](#)]
39. Zhang, Y.; Shen, C.; Tang, J.; Liu, J. Hybrid Algorithm Based on MDF-CKF and RF for GPS/INS System During GPS Outages (April 2018). *IEEE Access* **2018**, *6*, 35343–35354. [[CrossRef](#)]
40. Choi, A.; Jung, H.; Mun, J.H. Single Inertial Sensor-Based Neural Networks to Estimate COM-COP Inclination Angle During Walking. *Sensors* **2019**, *19*, 2974. [[CrossRef](#)] [[PubMed](#)]
41. Chen, H.; Aggarwal, P.; Taha, T.M.; Chodavarapu, V.P. Improving Inertial Sensor by Reducing Errors Using Deep Learning Methodology. In Proceedings of the NAECON 2018-IEEE National Aerospace and Electronics Conference, Dayton, OH, USA, 23–26 July 2018; pp. 197–202.
42. Xu, Z.; Li, Y.; Rizos, C.; Xu, X. Novel Hybrid of LS-SVM and Kalman Filter for GPS/INS Integration. *J. Navig.* **2010**, *63*, 289–299. [[CrossRef](#)]
43. Gao, T.; Sheng, W.; Zhou, M.; Fang, B.; Zheng, L. MEMS Inertial Sensor Fault Diagnosis Using a CNN-Based Data-Driven Method. *Int. J. Pattern Recognit. Artif. Intell.* **2020**, *34*, 2059048. [[CrossRef](#)]
44. Liu, L.; Wang, Z.; Qiu, S. Driving Behavior Tracking and Recognition Based on Multisensors Data Fusion. *IEEE Sens. J.* **2020**, *20*, 10811–10823. [[CrossRef](#)]
45. Grigorescu, S.; Trasnea, B.; Cocias, T.; Macesanu, G. A Survey of Deep Learning Techniques for Autonomous Driving. *J. Field Robot.* **2020**, *37*, 362–386. [[CrossRef](#)]
46. Varona, B.; Monteserin, A.; Teyseyre, A. A Deep Learning Approach to Automatic Road Surface Monitoring and Pothole Detection. *Pers. Ubiquitous Comput.* **2020**, *24*, 519–534. [[CrossRef](#)]
47. Baldini, G.; Giuliani, R.; Geib, F. On the Application of Time Frequency Convolutional Neural Networks to Road Anomalies' Identification with Accelerometers and Gyroscopes. *Sensors* **2020**, *20*, 6425. [[CrossRef](#)]
48. Subathra, B.; Radhakrishnan, T.K. Recurrent Neuro Fuzzy and Fuzzy Neural Hybrid Networks: A Review. *Instrum. Sci. Technol.* **2012**, *40*, 29–50. [[CrossRef](#)]

49. Smagulova, K.; James, A.P. A Survey on LSTM Memristive Neural Network Architectures and Applications. *Eur. Phys. J. Spec. Top.* **2019**, *228*, 2313–2324. [[CrossRef](#)]
50. Wang, S.; Clark, R.; Wen, H.; Trigoni, N. End-to-End, Sequence-to-Sequence Probabilistic Visual Odometry through Deep Neural Networks. *Int. J. Robot. Res.* **2018**, *37*, 513–542. [[CrossRef](#)]
51. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
52. Chen, C.; Lu, C.X.; Wahlström, J.; Markham, A.; Trigoni, N. Deep Neural Network Based Inertial Odometry Using Low-Cost Inertial Measurement Units. *IEEE Trans. Mob. Comput.* **2021**, *20*, 1351–1364. [[CrossRef](#)]
53. Li, C.; Wang, S.; Zhuang, Y.; Yan, F. Deep Sensor Fusion Between 2D Laser Scanner and IMU for Mobile Robot Localization. *IEEE Sens. J.* **2021**, *21*, 8501–8509. [[CrossRef](#)]
54. Jiang, C.; Chen, S.; Chen, Y.; Zhang, B.; Feng, Z.; Zhou, H.; Bo, Y. A MEMS IMU De-Noiseing Method Using Long Short Term Memory Recurrent Neural Networks (LSTM-RNN). *Sensors* **2018**, *18*, 3470. [[CrossRef](#)]
55. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv Preprint arXiv:1406.1078*, 2014.
56. Jiang, C.; Chen, Y.; Chen, S.; Bo, Y.; Li, W.; Tian, W.; Guo, J. A Mixed Deep Recurrent Neural Network for MEMS Gyroscope Noise Suppressing. *Electronics* **2019**, *8*, 181. [[CrossRef](#)]
57. Kiran, B.R.; Sobh, I.; Talpaert, V.; Mannion, P.; Sallab, A.A.A.; Yogamani, S.; Pérez, P. Deep Reinforcement Learning for Autonomous Driving: A Survey. In *IEEE Transactions on Intelligent Transportation Systems*; IEEE: Piscataway, NJ, USA, 2021; pp. 1–18. [[CrossRef](#)]
58. Tesauro, G. TD-Gammon, a Self-Teaching Backgammon Program, Achieves Master-Level Play. *Neural Comput.* **1994**, *6*, 215–219. [[CrossRef](#)]
59. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016; ISBN 978-0-262-33737-3.
60. Yang, J.-A.; Kuo, C.-H. Integrating Vehicle Positioning and Path Tracking Practices for an Autonomous Vehicle Prototype in Campus Environment. *Electronics* **2021**, *10*, 2703. [[CrossRef](#)]
61. Noureldin, A.; Karamat, T.B.; Georgy, J. *Fundamentals of Inertial Navigation, Satellite-Based Positioning and Their Integration*; Springer: Berlin/Heidelberg, Germany, 2013; ISBN 978-3-642-30465-1.