1. Paweł TOMIŁO[1], 2. Joanna MICHAŁOWSKA[2], 3. Jan LASKOWSKI[1], 4. Jarosław PYTKA[4], 5. Orest KOCHAN[5], 6. Ernest GNAPOWSKI[6]

Lublin University of Technology, Faculty of Management (1), Faculty of Electrical Engineering and Computer Science (2), Faculty of Mechanical Engineering (4), Faculty of Mathematics and Information Technology (6), Lviv Polytechnic National University, Department of Measuring-Information Technologies (5)
ORCID: 1. 0000-0003-4461-3194; 2. 0000-0002-2353-6305; 3. 0000-0002-4951-8674; 4. 0000-0002-5474-3585; 5. 0000-0002-3164-3821; 6. 0000-0001-6633-8395

doi:10.15199/48.2025.02.28

Artificial neural network model for estimation of relative airplane altitude

***Abstract**. The main purpose of the publication is to develop an artificial neural network model capable of estimating aircraft altitude based on accelerometer and gyroscope readings. The developed network structure uses LSTM, SE and transform encoder layers. Measurement data for training, validation and testing of the neural network were obtained from tests using an original measurement system developed. The developed device set was placed on the Koliber 150 aircraft. In order to verify the correctness of the model, the values estimated by the model were compared with those estimated by the Kalman filter algorithm using the double integration algorithm. The developed artificial neural network model has an overall estimation error of 7.43m, while the error for the Kalman filter was 20.27m. It has been shown that the use of the proposed model allows achieving satisfactory accuracy in aircraft altitude estimation due to the model's ability to adapt to the drift in the Inertial Measurement Unit (IMU).*

***Streszczenie.** Głównym celem publikacji jest opracowanie modelu sztucznej sieci neuronowej, który będzie w stanie oszacować wysokość samolotu na podstawie wskazań akcelerometru i żyroskopu. Opracowana struktura sieci wykorzystuje warstwy LSTM, SE i enkodera transformatorowego. Dane pomiarowe do uczenia, walidacji i testowania sieci neuronowej uzyskano z badan za pomocą opracowanego autorskiego system pomiarowego. Opracowane urządzenie umieszczono na samolocie Koliber 150. W celu weryfikacji poprawności modelu wartości oszacowane zostały porównane z wartościami, które oszacowano na podstawie algorytmu filtrem Kalmana z zastosowaniem algorytmu podwójnego całkowania. Opracowany model sztucznej sieci neuronowej charakteryzuje się ogólnym błędem estymacji wynoszącym 7.43m, zaś błąd dla filtru Kalmana wyniósł 20.27m. Wykazano że zastosowanie zaproponowanego modelu pozwala na osiągnięcie zadowalającej dokładności przy estymacji wysokości samolotu dzięki zdolności modelu do adaptacji do dryftu w Inercyjnej jednostce pomiarowej (IMU). (**Model sztucznej sieci neuronowej do szacowania względnej wysokości samolotu**)*

Słowa kluczowe: Sztuczna sieć neuronowa, szacowanie wysokości, statek powietrzny, inercyjna jednostka pomiarowa
Keywords: Artificial neural network, altitude estimation, aircraft, inertial measurement unit

Introduction

Satellite navigation is a world-renowned system that allows to determine the position of a given object in space using satellites. GNSS (Global Navigation Satellite Systems) receivers are characterized by low refresh rates in the range of 1-10 Hz. At higher refresh rates, there is a decrease in the position accuracy of the receiver [1]. Higher refresh rate is required for high-precision applications with fast-moving objects. Inertial navigation systems (INS) are used for this purpose. They are also used in the event of problems with the satellite navigation system. INS uses data from the Inertial Measurement Unit (IMU) systems to determine the position of a given object. Devices of this type are an independent system that ensures uninterrupted determination of the position of a given object in space, therefore they have been used in areas such as: autonomous navigation, aviation applications and dynamic vehicle control [2].

A frequently used method of determining the position of an object in time is the double integration method [3]. Sensor data is subject to dynamic noise and bias, therefore filters such as the Kalman filter (KF) or the alpha-beta filter are often used. They are responsible for compensation of the noise on the sensor reading. The accuracy of these solutions is significantly affected by the quality of the sensor [3].

Neural networks are used in many fields, such as quality management, mechanical engineering and automotive application [4,5]. They allow to perform such tasks as classification, detection, segmentation, regression and signal filtering. Methods have been developed that allow supporting the Kalman filter with the use of artificial neural networks (ANN) such as [6,7]. These methods integrate artificial neural networks with the Kalman filter. Adjusting the filtration parameters consists in finding the relationship between the processed data, which is why ANNs are a great solution that allows to find this relationship. Artificial neural networks, based on the output data, try to reproduce the output data with the smallest possible error, so the performance of fitting the nonlinear model with use of ANN is excellent in many cases. The performance of ANN is highly dependent on the data as well as the structure of the network itself. This approach allows solving non-linear time-varying problems without the need for an external mechanism model. For this reason, ANNs are a helpful tool for process and noise modeling, and also allow the reconstitution of unknown elements in the Kalman filter [8].

The authors used the prototype IMU measurement system to develop an algorithm using an artificial neural network to estimate Koliber 150 aircraft flight altitude during climb phase. The prototype device is characterized by small dimensions and cost effectiveness [9]. Training the network will take place in the process of supervised learning by applying input data in the form of IMU indication and position data using a high-accuracy sensor fusion GPS module. The Koliber 150 is a low-wing aircraft for training and sports purposes.

Periods of time in which the aircraft performed the following maneuvers were used for data analysis:

- takeoff/climb;
- cruise;
- descent/approach;

Acceleration in 3 axes as well as pitch and roll angles were used to reconstruct the change in aircraft altitude. The ZED-F9R GNSS receiver was used to collect the data. This module uses concurrent GNSS receivers and is capable of tracking multiple constellations. It has a multi-band front-end architecture that allows it to receive four major constellations – GPS (Global Positioning System), GLONASS, Galileo and

BeiDou, as well as SBAS (Satellite Based Augmentation System) and QZSS (Quasi-Zenith Satellite System) satellites simultaneously.

Structure of the neural network

The basic assumption of the developed algorithm is to create a light neural network that will allow to determine the height of the aircraft based on the IMU indication. During the development of the structure, the following layers were considered:

- LSTM – Long-Short Term Memory;
- Transformer encoder;
- SE – Squeeze and Excitation.

The LSTM (Long-Short Term Memory) module uses loops that allow information to flow from previous states to subsequent states. The LSTM mechanism has gates that allow information to be stored for many iterations. The input data for the current moment together with the output data from the previous state are used to determine the output data for the current state. The LSTM module consists of 3 gates [10,11]:

- forget gate;
- input gate;
- output gate.

Weights and biases are assigned to the values before passing through the gate. These values go through the activation function. The next step is to determine what new information will be stored in the LSTM cell. The vector created by activation functions is a candidate to be remembered. The input gate determines how much influence the candidate will have on changing the state of the given LSTM cell. The output value is based on the output information from the previous state and the weights, this information goes through the sigmoidal function. The resulting value is multiplied by the current state of the cell, which has passed through the hyperbolic tangent function [12].

The encoder transformer consists of two modules with residual connections. The first module is made up of a summation layer, normalization layer and multi-head attention (MHA). The second module is similar to the first one except that it has a feed-forward layer and no MHA The three learning-related matrices W^Q,W^K and W^V are used to project the input values in order to determine self-attention. Q, K and V are processed values. For a particular output value Q, the attention mechanism determines the weights, which establish the relative importance of each value in a specific series - K. The input data V is multiplied by the calculated weights [13].

The SE module is responsible for feature recalibration. The use of this module is possible for any transformation that maps the input values X to U such that $U \in R^{(W \times C)}$, where $W \times C$ is dimension of input data. The U features first go to the descriptor, which is responsible for their aggregation along the selected dimension. The descriptor creates global embeddings of the channel distribution function. The data after aggregation serve as input data for the self-gate mechanism (a layer of densely connected neurons). This mechanism is responsible for creating a set of weights that will be used to modulate the values for each channel [12]. In order to develop the structure, 3 models were created:

- model using a transformer encoder;
- model using LSTM;
- model consisting of transformer encoder, LSTM and SE.

The target number of epochs for each model was 30, the iterations in which the network reached the smallest value of the loss function were selected. The networks were compared due to the average height estimation error. On this basis, the final structure of the network was selected. Comparison of the discussed models is presented in table 1.

Table 1. Comparison of ANN models

ANN TYPE	Transformer encoder	LSTM	Transformer encoder + LSTM + SE
Estimation ERROR [m]	39,61	19,87	5,67
Trainable parameters	4 751	20 129	24 942
Non-trainable parameters	0	0	10
Total parameters	4 751	20 129	24 952

The final structure of the network consists of 2 branches. In the first branch, the input data goes to the transformer encoder, in the next phase, the information goes to the CBT module (convolution 1D, normalization of batch, activation function - hyperbolic tangent). The next step is two layers of bidirectional LSTM. Output from this branch is a 1D convolution with shape (100, 1). In the second branch input data goes to 1D convolution layer and SE module. Data from both branches is concatenated – shape (100,6). Output layer is a 1d convolution. Figure 1 shows the structure of the network, taking into account the shape of the layers.

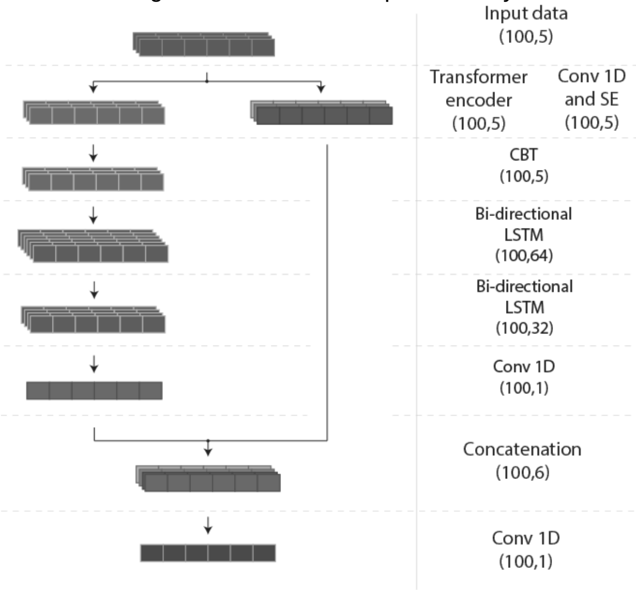

Fig. 1. Structure of the neural network

Training

The input data to the network were time courses of accelerations along the x, y, z axes, as well as pitch and roll angles. GPS indications - the altitude of the aircraft were used as the truth data for the training process.

For the training process, the dataset of 10 hours of flight was divided into training, validation and test sets. An additional 100 samples for every single set of measurement are stored to reduce the impact of the overfitting phenomenon, a random data shift was applied for each of the measurements. The shift value changes when the epoch ends during the training process.

Pre-processing takes place before data is fed into the neural network. For this purpose a standardization method was used. This process normalizes input data to have zero mean and unity variance.

Normalization methods help to gradually stabilize the gradient descent by placing different features on a similar scale. This helps in faster convergence of the model while maintaining the current value of the learning rate parameter or allows the use of a higher value of this parameter.

In the training process, the ADAM algorithm was adopted as the optimizer. The ADAM algorithm is responsible for adaptive moment estimation. It is a combination of two descent gradient methodologies: momentum and RMSP (Root Mean Square Propagation). The ADAM algorithm with the following parameters was used $\alpha = 0{,}001; \beta_1 = 0{,}9, \beta_2 = 0{,}999;\ \epsilon = 10^{-8}$, values were adopted according to [13].

Logarithm of the hyperbolic cosine of the prediction error (log-cosh) was chosen as the loss function. As discussed by [16] this function belongs to the class of robust estimators. This means that the function prefers solutions close to the median rather than the mean value and is more tolerant to outliers.

In the process of training the neural network, the target number of epochs was 150. Due to the lack of decrease in the value of the function for the loss validation set for 30 epochs, the weights obtained for epoch 43 were selected.

After the training network reached mean squared error of 0,3571 for the test set and 0,3235 for the set intended for training. In the case of the value of the loss function 0,1494 was achieved for the test set and 0,1185 for the training data set.

Kalman Filter

Kalman filter is a state estimator, which is able to extract information from noisy data. The principle of operation of the algorithm consists of 2 steps: prediction and updating. In the prediction step, the algorithm is estimating the current states variables and their uncertainties. Previous states of the system are used to estimate subsequent states. In update step, the Kalman filter uses the measurement data in the current time step to estimate the current state of the system. An example of using the Kalman filter on the collected data is shown in Figure 2.

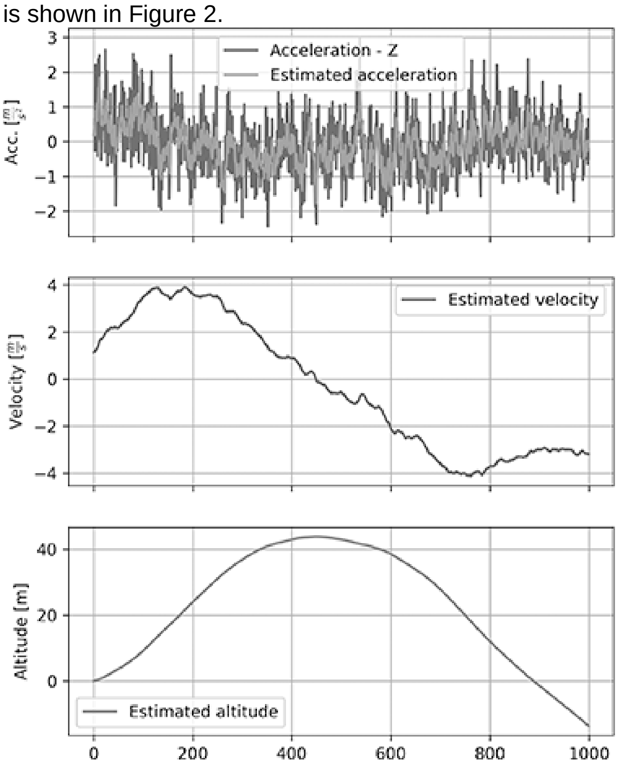

Fig. 2. Example of using the Kalman filter on the collected data

Results

For analysis 10 segments were designated for the measurements for each of the 3 groups - climb, cruise - level flight, and descent. For the purpose of comparison, 9 characteristic points were used, which determine the position of the aircraft with the use of a GNSS receiver. The data processed using the artificial neural network was compared with those processed using the Kalman filter. Comparison of data for an example measurement for is shown in figure 3.

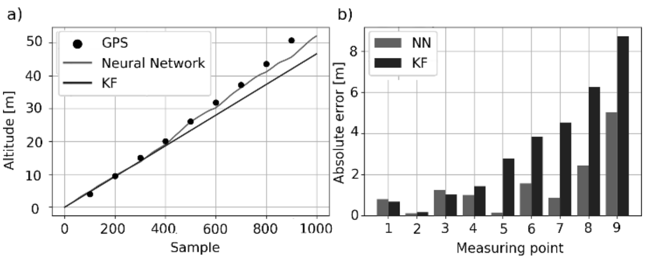

Fig. 3. a) Flight altitude estimation b) Absolute error

In the case of the climb phase, for the results processed using the neural network, there was no increasing error as in the case of the Kalman filter. For the first two measurement points, the network showed a larger average error than the Kalman filter. However, for subsequent measurements, the Kalman filter had a much larger altitude estimation error than data processed with the use of artificial neural network. For the entire measurement set, the mean absolute error of the neural network was 5.48m, and for the Kalman filter it was 11.24m. Only for the first measurement section, the neural network showed a greater average error than the Kalman filter. The mean absolute error for each of the measurement sections for the climb phase is shown in Figure 4.

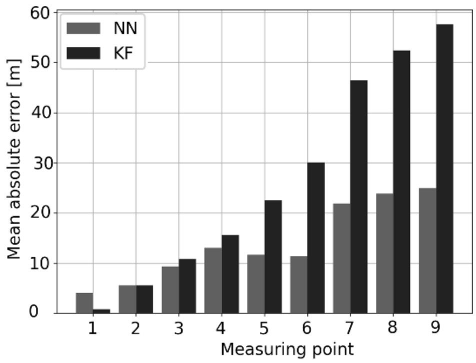

Fig. 4. Mean absolute error per measuring point for climb phase

During the cruise phase, the Kalman filter tended to overestimate the relative change in altitude. Therefore, its indications significantly differed from the measurement points. The artificial neural network maintained a similar value of the mean absolute error for the measurement points. The mean absolute error for the measurement points is shown in Figure 5. For the measurement segments in which the aircraft cruised - small changes in altitude, the neural network was characterized by an average absolute error of 3.77m, and the error for the Kalman filter was 22.69m. There were two measurements in the measurement set, for which the Kalman filter significantly overestimated the relative height change. The mean absolute error for each of the measurement sections for the climb phase is shown in Figure 9.

The last flight phase analyzed was the descent/approach. Again with the Kalman filter, there was a cumulative error.

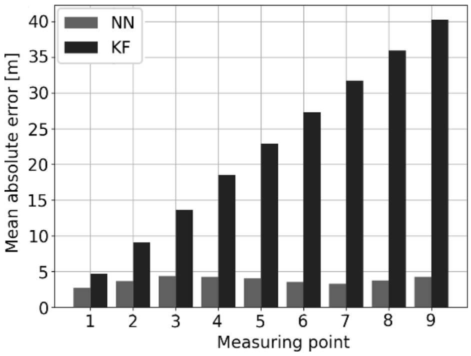

Fig. 5. Mean absolute error per measuring point for cruise phase

The neural network for the first two measurement points showed a greater error than the Kalman filter. The mean absolute error for the measurement points is shown in Figure 6.

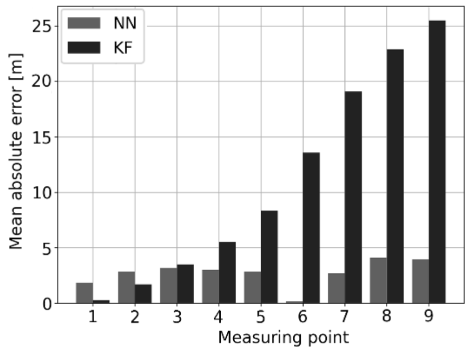

Fig. 6. Mean absolute error per measurement for descent/approach

The average absolute altitude estimation error for the discussed flight phase in the case of the neural network was 13.98 m, and for the Kalman filter the error was 26.87m. In this phase of the flight, both in the case of the neural network and the Kalman filter, the error was the largest.

Conclusion

In order to demonstrate the correctness of the selected model, the values obtained using the Kalman filter method and the developed algorithm using the neural network were compared. The Kalman filter is characterized by a much larger error in the estimation of the relative change in the altitude of the aircraft, and thus in the overall estimation of the altitude. A cost-effective prototype IMU set was used for the measurements, the parameters of which allowed obtaining satisfactory results in obtaining an artificial neural network.

The average error for all measurements, for the ANN model, was 7.43m, while for the Kalman filter using double integration, an error of 20.27m was obtained. During the tests in the aircraft, quite large vibrations from the engine were observed, which were transferred to the entire structure of the aircraft. The mentioned vibrations affected the reading from the IMU, which in the case of the Kalman filter translated into a larger error. The neural network learned the features of the signal from the tested IMU system and in the training process it got "used" to the occurring vibrations from the engine.

The use of a transformer encoder allowed the creation of attention maps for the input data. The maps were used to emphasis the importance of the selected features of the input data. The network's input data, in the form of a tensor of dimensions (100,5), is processed to obtain a relative change in the height of the aircraft in the form of a tensor (100,1). The network estimates the height change as a sequence of successive values for the entire input data tensor. LSTM modules store information from previous states, which allows the network to refer to the previous state.

Authors: *dr inż. Paweł Tomiło, Faculty of Management, Lublin University of Technology, Lublin, Nadbystrzycka 38, 20-618 Lublin, e-mail:p.tomilo@pollub.pl; dr inż. Joanna Michałowska, Faculty of Electrical Engineering and Computer Science, Lublin University of Technology, Lublin, Nadbystrzycka 38A, 20-618 Lublin, e-mail:j.michalowska@pollub.pl; dr inż. Jan Laskowski, Faculty of Management, Lublin University of Technology, Lublin, Nadbystrzycka 38, 20-618 Lublin, e-mail:j.laskowski@pollub.pl; dr. hab. inż. Jarosław Pytka, Faculty of Mechanical Engineering, Lublin University of Technology, Nadbystrzycka 36, 20-618 Lublin, e-mail:j.pytka@pollub.pl; prof. Orest Kochan, Department of Measuring-Information Technologies, Lviv Polytechnic National University, 79013 Lviv, Ukraine, e-mail:orest.v.kochan@lpnu.ua; dr hab. Ernest Gnapowski, Faculty of Mathematics and Information Technology, Lublin University of Technology, Lublin, Nadbystrzycka 38, 20-618 Lublin, e-mail:e.gnapowski@pollub.pl*

REFERENCES

[1] Rychlicki M., Kasprzyk Z., Rosiński A., Analysis of Accuracy and Reliability of Different Types of GPS Receivers. *Sensors*, 20, (2020), 6498, doi:10.3390/S20226498
[2] Noureldin A., Karamat T.B., Georgy J., Fundamentals of Inertial Navigation Satellite-Based Positioning and Their Integration. Fundamentals of Inertial Navigation Satellite-Based Positioning and their Integration, *Springer,* (2013), 1–313, doi:10.1007/978-3-642-30466-8/COVER
[3] Filippeschi A., Schmitz N., Miezal M., Bleser G., Ruffaldi E., Stricker D., Fortino G., Ghasemzadeh H., Li W., Zhang Y., et al., Survey of Motion Tracking Methods Based on Inertial Sensors: A Focus on Upper Limb Human Motion, *Sensors*, Vol. 17, (2017) 1257, doi:10.3390/S17061257
[4] Jamil F., Kim D., Improving Accuracy of the Alpha–Beta Filter Algorithm Using an ANN-Based Learning Mechanism in Indoor Navigation System, *Sensors*, Vol. 19, (2019) 3946, doi:10.3390/S19183946
[5] Kulisz M., Zagórski I., Józwik J., Korpysa J., Research Modelling and Prediction of the Influence of Technological Parameters on the Selected 3D Roughness Parameters as Well as Temperature Shape and Geometry of Chips in Milling AZ91D Alloy, *Material*, Vol. 15, (2022), 4277, doi:10.3390/MA15124277
[6] Tomiło P., Classification of the Condition of Pavement with the Use of Machine Learning Methods. *Transport and Telecommunicatio*, Vol. 24, (2023), 158–166, doi:10.2478/TTJ-2023-0014,
[7] Revach G., Shlezinger N., Ni X., Escoriza A.L., Van Sloun R.J.G., Eldar Y.C. KalmanNet, Neural Network Aided Kalman Filtering for Partially Known Dynamics, *IEEE Transactions on Signal Processing*, Vol. 70, (2022) 1532–1547, doi:10.1109/TSP.2022.3158588
[8] Jiang, K., Zhang, C., Wei, B., Li, Z., & Kochan, O., Fault diagnosis of RV reducer based on denoising time–frequency attention neural network, *Expert Systems with Applications*, (2024), 238, 121762
[9] Pytka J., Budzyński P., Tomiło P., Michałowska J., Błażejczak D., Gnapowski E., Pytka J., Gierczak K., Measurement of Aircraft Ground Roll Distance during Takeoff and Landing on a Grass Runway, *Measurement*, Vol. 195, (2022), 111130 ,doi:10.1016/J.MEASUREMENT.2022.111130
[10] Graves A. Supervised Sequence Labelling with Recurrent Neural Networks. Springer (2012), doi:10.1007/978-3-642-24797-2
[11] Hochreiter S., Schmidhuber J., Long Short-Term Memory. *Neural Comput*, Vol. 9, (1997), 1735–1780 doi:10.1162/NECO.1997.9.8.1735.
[12] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser Ł., Polosukhin I. Attention Is All You Need. *Adv Neural Inf Process Syst,* (2017), 5999–6009,
[13] Hu J., Shen L., Albanie S., Sun G., Wu E., Squeeze-and-Excitation Networks, *IEEE Trans Pattern Anal Mach Intell*, Vol. 42,(2020) 2011–2023, doi:10.1109/TPAMI.2019.2913372