BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 68, No. 6, 2020
DOI: 10.24425/bpasts.2020.135389

Robot sensor failure detection system based on convolutional neural networks for calculation of Euler angles

A. ŚWIETLICKA* and K. KOLANOWSKI

Institute of Automation and Robotics, Poznan University of Technology, Piotrowo 3A, 60-965 Poznań, Poland

Abstract. In this work, we present a failure detection system in sensors of any robot. It is based on the k-fold cross-validation approach and built from N neural networks, where N is the number of signals read from sensors. Our tests were carried out using an unmanned aerial vehicle (UAV, quadrocopter), where signals were read from three sensors: accelerometer, magnetometer and gyroscope. Artificial neural network was used to determine Euler angles, based on signals from these sensors. The presented system is an extension of the system that we proposed in one of our previous papers. The improvement shown in this work took place on two levels. The first one was related to improvement of a neural network's reproduction quality – we have replaced a recurrent neural network with a convolutional one. The second level was associated with the improvement of the validation process, i.e. with adding some new criteria to check the values of Euler's angles determined by the convolutional neural network in subsequent time steps. To highlight the proposed system improvement we present a number of indicators such as RMSE, NRMSE and NDR (Normalized Detection Ratio).

Key words: quadrocopter, convolutional neural network, AHRS, Attitude and Heading Reference System.

1. Introduction

Nowadays, scientifically, the issue of failure detection is very interesting due to, among others, a wide use of multi-sensor systems. It is interesting not only from a UAV (Unmanned Aerial Vehicle) point of view but also from other devices [1] and systems like maritime and underground navigation or augmented reality systems.

Presented in this paper work describes the improvement of the previously presented system [2], which is based on k-fold cross-validation approach. Our new failure detection system uses a convolutional neural network (CNN) instead of recurrent one, as an approximator of the Euler angles. Another improvement is related with introduction of additional conditions strengthening the system of failure detection. An undoubted advantage of our failure detection system is that sensors can be of any type, i.e. giving readings from different ranges.

The research presented in this article is a part of a larger project that includes issues related to determining the position of an object in space. In work [3], we started the research presentation by showing how to determine Euler angles using the Elman recurrent neural network. As a continuation, in [4] the Elman network structure was expanded with kinetic models of a biological neuron and the network thus created was tested on new data. The work [2] concerned the use of recurrent networks to detect damage in unmanned aerial vehicle sensors.

The aim of our work was to build a fast failure detection system (or rather to improve the existing one). The first problem faced in building this system was calculating the position of the considered object (in our case UAV – unmanned aerial vehicle) in space. Most of the solutions require computationally expensive matrix calculations, which impede either software or hardware implementation. That was one of the reasons why we have reached towards artificial neural networks (ANN), which are known from their many advantages. Among many other ANNs automatically detect the important features without any human supervision, which means that a designer does not have to know the specific relation between inputs and outputs of the system. Additionally we have decided to consider convolutional neural networks because of their computational simplicity, which is important in terms of real-time calculation either of position in space or of the sensor failure.

The final sensor failure detection system is built on the idea of k-fold cross-validation approach, composed from N neural networks, where N is a number of collected from sensors signals. Each of the neural network takes as an input $N-1$ signals. When all but one neural networks show incorrect values of Euler angles, than it is assumed that the sensor excluded in this very one sensor is somehow damaged. To confirm our assumptions about the quality of the proposed system we propose various measures and visual characteristics, like root-mean-square error (RMSE), normalized root-mean-square error (NRMSE) and normalized detection ratio (NDR). Additionally we show standard summary of the convolutional neural network training overview.

The paper is organized as follows: Section 2 shows research related with determining an object's position in space. Section 3 presents the mathematical background for the quaternions,

*e-mail: aleksandra.swietlicka@put.poznan.pl

Manuscript submitted 2020-07-08, revised 2020-09-10, initially accepted for publication 2020-10-02, published in December 2020

which are commonly used for determining an object's position. Section 4 focuses on convolutional neural network for calculation of Euler angles, while Section 5 shows failure detection system based on these neural networks with some results of system's operation. Finally, in Section 6, a short summary is provided.

2. Related work

We are, even unconsciously, used to the sensors that surround us. They are everywhere, in cars, smartphones, watches, home appliances [5], industrial machines, even clothes are equipped with sensors [6], etc. Sensors usually determine measurable physical parameters such as temperature, pressure, acceleration, speed, direction and many others.

To determine the position of any object relative to the start point of the reference system, it is necessary to choose one of the measuring techniques used for this. The most commonly used measurement techniques are global navigation satellite systems (GNSS), heterogeneous wireless sensors [7], inertial sensors, radiolocation, vision systems and many others. The most commonly used techniques for determining the object's position are combining these systems, which results in accuracy improvement and weaknesses elimination of a particular measurement method [8]. For example a navigation system based on a radio signal from GNSS will not work properly in a tunnel or underground [9].

The latest works on determining an object's position in space are still based on the technique that uses small and cheap inertial sensors made in Micro Electro-Mechanical Systems (MEMS) technology and GNSS system receivers [10, 11].

Because in general the computational and process complexity of industrial machines elimination, reliability and security become more and more important. Diagnostics of industrial processes deals with the recognition of changes in the states of these processes, where industrial processes are understood as a series of intentional activities implemented at a set time by a specific set of machines and devices with specific available resources. Diagnosis is treated as a process of detecting and distinguishing object damage as a result of collecting, processing, analyzing and evaluating diagnostic signals. The diagnosis can be carried out at different levels of detail. Depending on the type of object and knowledge available about it, the result of the diagnosis may be a detailed identification of the damage or only a general definition of the condition class [12].

It is possible to list a lot of papers on failure detection in sensor systems [13–16], as many researchers look for a perfect solution for this problem – usually for a specific device. Many of the presented solutions assume that the device is equipped with at least several sensors, which is an undoubted problem due to universality of the proposed failure detection systems. A great comparison of available solutions of this problem is analysed in [17].

The failure detection in sensor signals was for the first time introduced in space mission projects, where small changes in the state variables significantly affected the subsequent stages in the object control [18, 19]. Soon after those first trials Kalman filters were proven to be an optimal solution for hidden variables state estimation in the space systems [20, 21]. In fact Kalman filters are still very successfully used in technical solutions [22–25].

Nowadays the use of non-conventional, non-deterministic methods in any real-life problem has become very popular. Finding solution of this problem without designing a deterministic model, i.e. without the precise knowledge of the relation between inputs and outputs of the model, has become very desirable, especially if this solution can work in real-time. For the problem of sensor failure detection it is possible to find a wide variety of papers that as a solution propose machine learning algorithms [26] and statistical methods [27, 28]. A novel approach for AHRS (Attitude Heading Reference System) based on artificial neural networks (a part of machine learning field) is presented in [29], which is focused on better estimation of the orientation of the mobile platform. It is also possible to find many solutions based on artificial neural networks, that focus mainly on maintenance of the flight direction and height [30–33].

3. Mathematical background

To make it possible to work with artificial neural networks, sample sets for learning, validation and testing data should be prepared. An algorithm was used to determine position in space based on a series of measurement samples from a multi-sensor system. For this purpose the extended Kalman filters and quaternions were used to simplify the matrix calculus.

The position of an navigated object can be calculated in few different ways. One of the common ways is designating a displacement between two coordinate systems (frames of reference), where the first one is centered on the Earth's surface or the starting point of the calibrated sensors, while the second one is the position of the navigated object, e.g. quadrocopter. The rotation of a vector in the Euclidean space is obtained by multiplying by the rotation matrix R (see Eq. (1)) where ϕ – Roll, θ – Pitch, ψ – Yaw, and represent rotations in longitudinal, transverse, and vertical axes, respectively.

$$R = \begin{pmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{pmatrix} \tag{1}$$

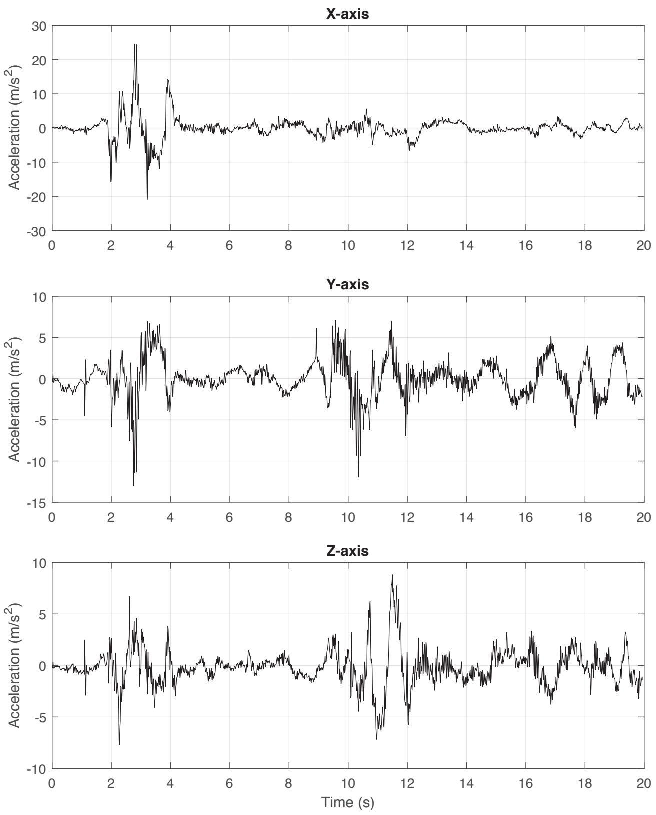

Fig. 1. Sample measurement signal from the accelerometer for 3 axes (X, Y and Z)

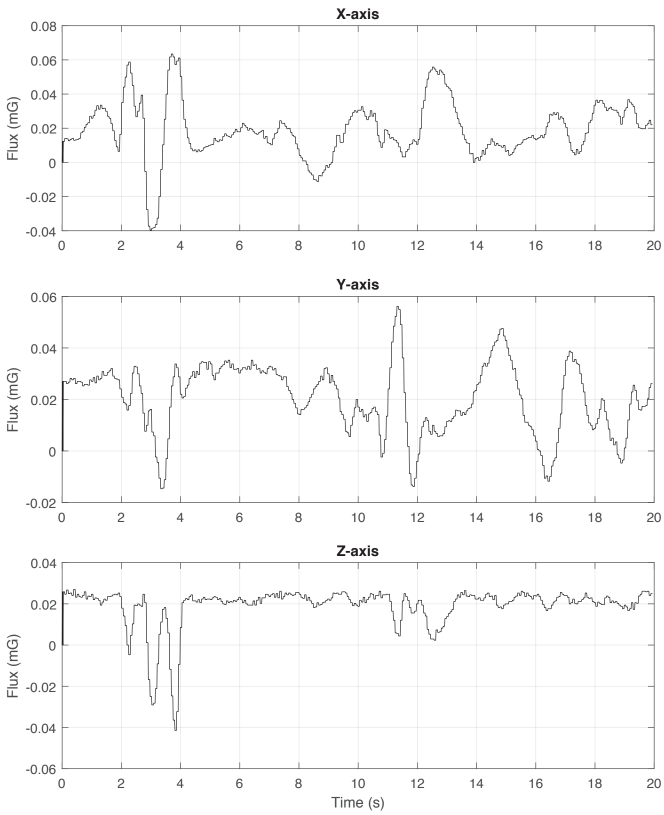

Fig. 2. Sample measurement signal from the magnetometer for 3 axes (X, Y and Z)

The AHRS (Attitude and Heading Reference System) makes it possible to calculate the position of an object in space. The system consists of sensors that provides information about the three degrees of freedom related with circular motion along axes x, y and z. It consists of the accelerometer, gyroscope and magnetometer [10] – sensors created in the MEMS (Micro Electro-Mechanical Systems) technology[1]. The system makes it possible to determine the position of an object in space based on linear and angular acceleration, direction and the magnitude of Earth's magnetic field [34]. An example data obtained from such set of sensors are presented in Figs. 1, 2 and 3 for accelerometer, magnetometer and gyroscope, respectively, collected during a random flight and hand maneuver of the quadrocopter. As we describe more precisely in the Section 4, various signals that we use to train and test our neural network are prepared not only during a standard flight but also by maneuvering in slow and fast motion of robot. Analysing these waveforms of few seconds of quadrocopter flight, we can see that between 2 and 4 seconds there was a change in flight altitude, while between 8 and 20 there was a displacement with rotation around its axis.

Matrix operators are often used in robotics transformations [35], without which it is impossible to imagine navigating the

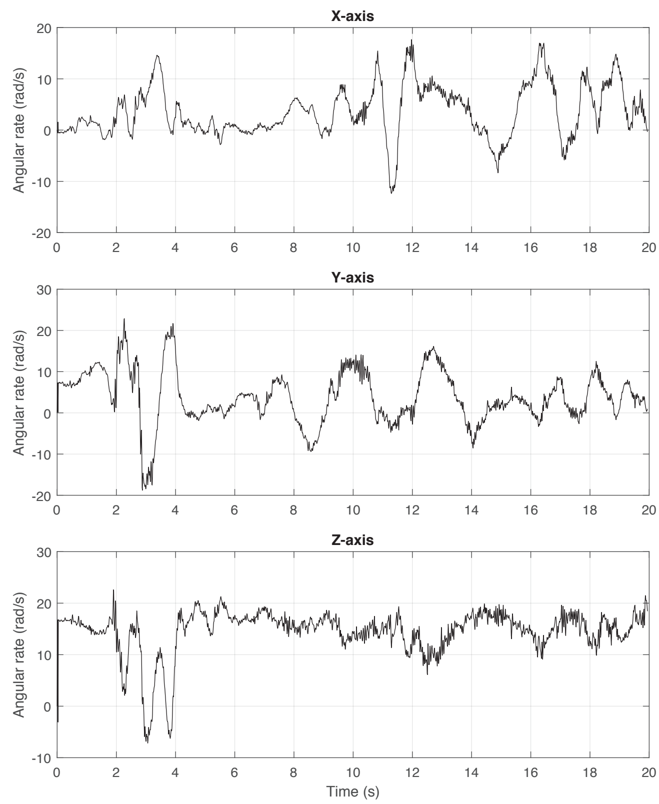

Fig. 3. Sample measurement signal from the gyroscope for 3 axes (X, Y and Z)

[1] As a sample recording device, a dedicated embedded system based on the STM32 Cortex M4 microcontroller was built. The system was built from Atmel ATAVRSBIN1 IMU (Inertial Measurement Unit), composed from accelerometer BMA150, gyroscope ITG3200 and magnetometer AK8975.

manipulator's kinematic chain or even to solve a simple kinematics problem. Quaternions are known since 1853, thanks to William R. Hamilton [36] and are commonly used for calculation of an object's position in space [10]. Many modern computer graphics systems work with quaternions, and they are used for making rotations in three-dimensional space. It is possible to mention many advantages of quaternions that make them so useful:

- they are not sensitive to the phenomenon of loss of freedom called "gimbal lock" but their normalization is recommended,
- they are represented by four numbers, not nine as in the case of a rotation matrix,
- they ensure easy transition from Euler angles to the axis of rotation and vice versa,
- interpolation of two quaternions is easier than of a matrix,
- normalization of quaternions is easier than orthogonalization of a matrix (the case of removing the accumulation of increasing errors of many calculations performed),
- turnover submission is done by multiplication (similar to matrices).

In order to simplify all the necessary matrix calculations we have decided to perform part of them by convolutional neural network (CNN). In our previous research [2] we focused on recurrent neural networks and now decided to improve the structure with convolutional layers. Computational simplicity of convolutional layers enables making more sophisticated, extensive structures and at the same time we can gain on speed of calculations.

4. Convolutional neural network for calculation of Euler angles

Based on the research presented in [2], the data obtained from the sensors underwent the normalization process using the function tanh (ax), where $a = 0.001$ for the accelerometer and $a = 0.01$ for other sensors.

For the calculation of Euler angles we have used a convolutional neural network (CNN), with the structure presented in Table 1, implemented in MATLAB 2018a environment. The CNN is built of one convolutional layer, after which we have used the ReLU activation layer and the MaxPooling layer, after which we have added the fully-connected layer after which we used the common method for regularization, namely dropout. The output layer predicting the Euler angles values is of the form of a regressor.

During training, we considered different sizes of the neural network's structure, we tried to add additional convolutional layers and increase the number of neurons in individual layers, but eventually we stayed with the structure with one hidden layer and 256 neurons, as larger structures did not improve results. In the training process, we examined three different training solvers: SGDM, RMSProp and Adam. An example training progress of the CNN is presented in Fig. 4, from which it is possible to notice that training lasted for about 1.5 minutes (hardware specification: 64-bit Windows operating system, 32 GB RAM, processor i7, 2.8 GHz)). Because we used a regression layer as the output of the CNN, in the training progress window we can observe the plot of the root mean square error (RMSE) calculated on each individual mini-batch. We can then observe that the RMSE drops below 0.05 while loss – below 0.002. The results are not satisfactory enough for Euler angles calculation itself, but as we mentioned in our previous work [2] it is not necessary for this neural network to be very precise as it will be subsequently used in the failure detection system.

In Table 2 there are shown the RMSE (root-mean-squared error) and NRMSE (normalized root-mean-squared error) calculated between the values of Euler angles obtained from the prediction carried out by the CNN and the values of these angles obtained by means of quaternion calculations (i.e. AHRS algorithm). In the comparison we took into account the recurrent neural network (`layrecnet`) described in the article [2] as the best one and a new, convolutional neural network with three types of solver. To calculate the RMSE and NRMSE values, we used samples from eight different tests:

test_1: test
test_2: test
test_3: oscillations along the X axis
test_4: oscillations along the Y axis
test_5: oscillations along the Z axis
test_6: maneuvering in slow motion of robot in air
test_7: maneuvering in fast motion of robot in air
test_8: shaking a robot

Table 1
Structure of the convolutional neural networked used in calculation of Euler angles

1	'input_layer'	Image Input	9×1×1 images
2	'convolution_layer'	Convolution	256 1×1 convolutions with stride [1 1] and padding [0 0 0 0]
3	'RELU_activation'	ReLU	ReLU
4	'maxpooling_layer'	Max Pooling	1×1 max pooling with stride [1 1] and padding [0 0 0 0]
5	'dense_layer'	Fully Connected	3 fully connected layer
6	'dropout_layer'	Dropout	10% dropout
7	'regression_output_layer'	Regression Output	mean-squared-error

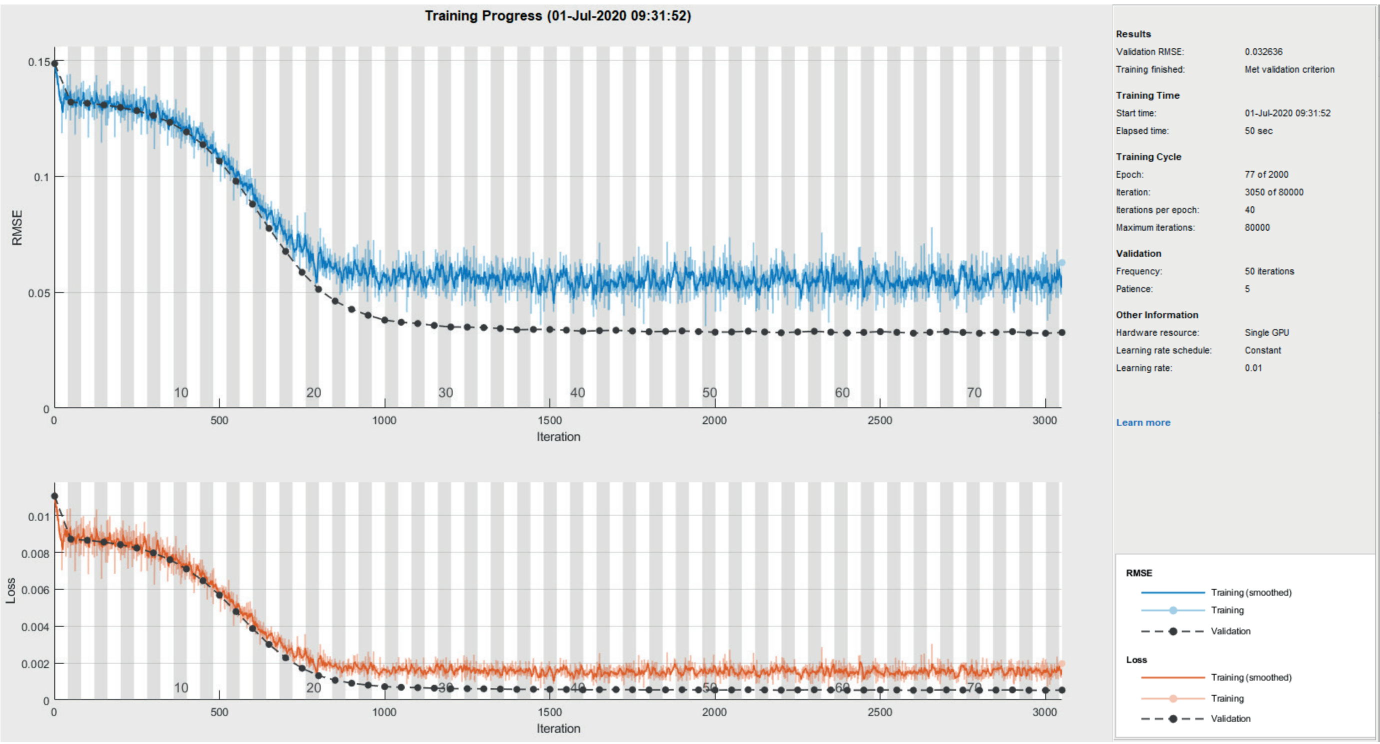

Fig. 4. Training progress of the convolutional neural network for Euler angles calculation

Table 2
RMSE & NRMSE calculated between the values of Euler angles obtained from the prediction carried out by the CNN & RNN and the values of these angles obtained by means of quaternion calculations

		layer recurrent neural network		convolutional neural network					
				SGDM solver		RMSProp solver		ADAM solver	
		RMSE	NRMSE	RMSE	NRMSE	RMSE	NRMSE	RMSE	NRMSE
test_1	Roll	0.5319	0.0029	0.2616e-03	0.1454e-05	0.2578e-03	0.1433e-05	0.2654e-03	0.1475e-05
	Pitch	0.1449	0.0017	0.0992e-03	0.1154e-05	0.1042e-03	0.1212e-05	0.1112e-03	0.1294e-05
	Yaw	1.0837	0.0060	0.3863e-03	0.2147e-05	0.4001e-03	0.2223e-05	0.3625e-03	0.2014e-05
test_2	Roll	0.6335	0.00352	0.3968e-03	0.2206e-05	0.4029e-03	0.2240e-05	0.3891e-03	0.2163e-05
	Pitch	0.1691	0.0019	0.1216e-03	0.1415e-05	0.1350e-03	0.1571e-05	0.1408e-03	0.1638e-05
	Yaw	1.1938	0.0066	0.4102e-03	0.2279e-05	0.4348e-03	0.2416e-05	0.4062e-03	0.2257e-05
test_3	Roll	1.0824	0.0060	0.5575e-03	0.3099e-05	0.5706e-03	0.3172e-05	0.5622e-03	0.3125e-05
	Pitch	0.2068	0.0024	0.2318e-03	0.2696e-05	0.2420e-03	0.2816e-05	0.2323e-03	0.2703e-05
	Yaw	1.0309	0.0057	0.4977e-03	0.2765e-05	0.5323e-03	0.2958e-05	0.4975e-03	0.2764e-05
test_4	Roll	1.1144	0.0061	0.4977e-03	0.2767e-05	0.4935e-03	0.2743e-05	0.5417e-03	0.3011e-05
	Pitch	0.2437	0.0028	0.0662e-03	0.0771e-05	0.0746e-03	0.0869e-05	0.0501e-03	0.0584e-05
	Yaw	0.9198	0.0051	0.6684e-03	0.3714e-05	0.7162e-03	0.3979e-05	0.6201e-03	0.3446e-05
test_5	Roll	1.4184	0.0078	0.0678e-03	0.0377e-05	0.0742e-03	0.0413e-0	0.0904e-03	0.0503e-05
	Pitch	0.1915	0.0022	0.0497e-03	0.0579e-05	0.0780e-03	0.0908e-05	0.0735e-03	0.0855e-05
	Yaw	0.6165	0.0034	0.4565e-03	0.2537e-05	0.4708e-03	0.2616e-05	0.3852e-03	0.2141e-05
test_6	Roll	0.4357	0.0024	0.1520e-03	0.0845e-05	0.1478e-03	0.0822e-05	0.1562e-03	0.0869e-05
	Pitch	0.2569	0.0029	0.0972e-03	0.1132e-05	0.0856e-03	0.0996e-05	0.0899e-03	0.1047e-05
	Yaw	1.5926	0.0088	0.2682e-03	0.1490e-05	0.2701e-03	0.1501e-05	0.2789e-03	0.1550e-05
test_7	Roll	9.5361	0.0530	0.3262e-03	0.1814e-05	0.3155e-03	0.1754e-05	0.3446e-03	0.1916e-05
	Pitch	2.0287	0.0235	0.1642e-03	0.1910e-05	0.1504e-03	0.1750e-05	0.1596e-03	0.1857e-05
	Yaw	9.2753	0.0515	0.2950e-03	0.1639e-05	0.3035e-03	0.1687e-05	0.2885e-03	0.1603e-05
test_8	Roll	9.8419	0.0547	0.4382e-03	0.2436e-05	0.4423e-03	0.2459e-05	0.4370e-03	0.2429e-05
	Pitch	2.0391	0.0237	0.2167e-03	0.2521e-05	0.1947e-03	0.2265e-05	0.1988e-03	0.2313e-05
	Yaw	9.7087	0.0539	0.4983e-03	0.2769e-05	0.4956e-03	0.2754e-05	0.5131e-03	0.2851e-05

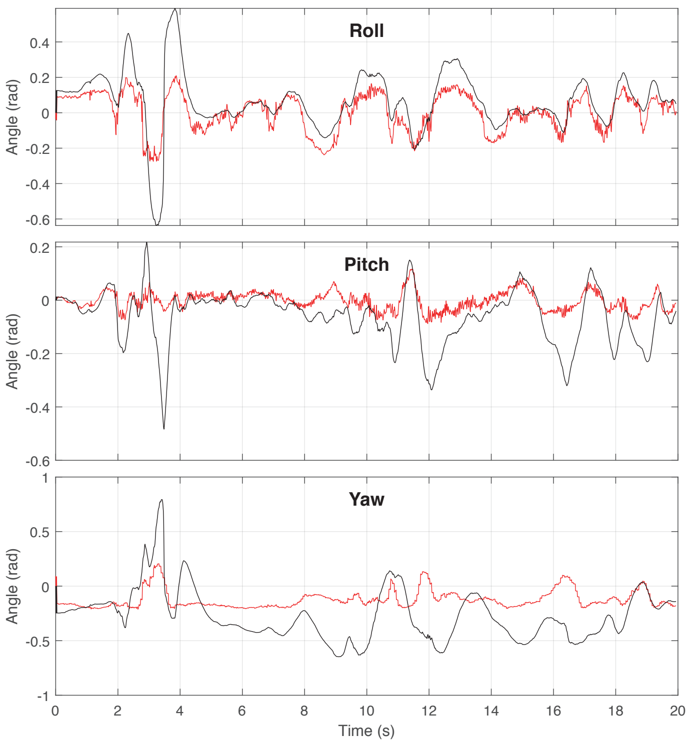

Fig. 5. Euler angles (Roll, Pitch, Yaw) calculated with use of the AHRS system (black line) and obtained with prediction function of the CNN (red line)

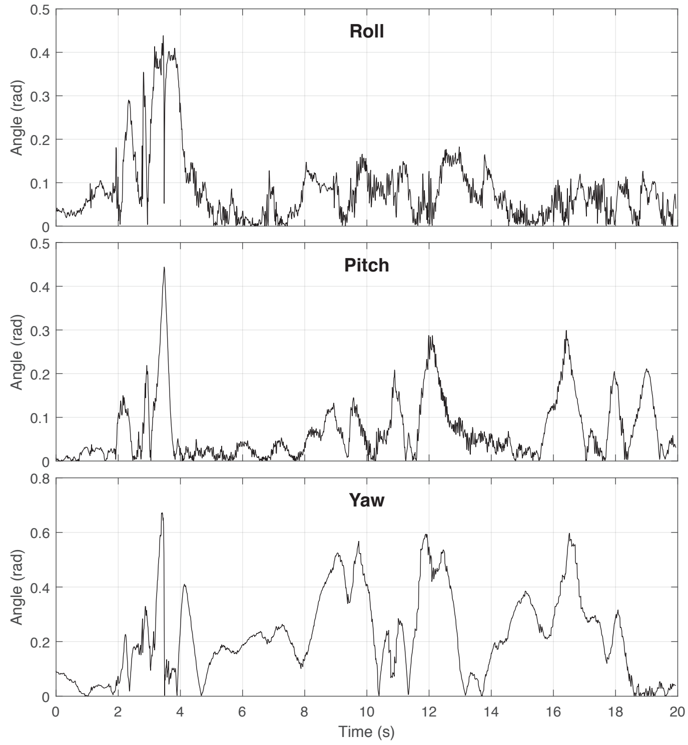

Fig. 6. The difference between Euler angles (Roll, Pitch, Yaw) calculated with use of the AHRS system and the ones obtained with prediction function of the CNN

Test_1 and test_2 were collected during a standard check up of sensors, which were performed by moving the set of sensors to observe their correct work.

Reviewing Table 2 we can see that replacing the recurrent neural network with the convolutional one significantly improved the values of the RMSE and NRMSE coefficients – in both cases we can observe improvement of these coefficients by three orders, regardless of the chosen solver.

In Fig. 5 we present Euler angles calculated with use of the AHRS system (black line) and obtained with prediction function of the CNN for data from test_6. Additionally in Fig. 6 we present the difference between plots presented in Fig. 5 for each of the Euler angles.

5. Sensor failure detection system

After designing the convolutional neural network for calculation of Euler angles it was then possible to design the failure detector for any number of the control signals with use of the *k*-fold cross-validation approach. The system is built from N approximators with $N - 1$ inputs each, where N is the number of signals obtained from robot's sensors. Each approximator – artificial neural network – has to exclude a different input signal. The value obtained with each approximator is then analysed in an additional block VB (Validation Block), which checks whether the output signal is outside of the assumed range. If one of the approximators does not report an error, while others do, then it is obvious that the excluded in the non-reporting error approximator signal comes from a damaged sensor. The whole Failure Detector is shown in Fig. 7.

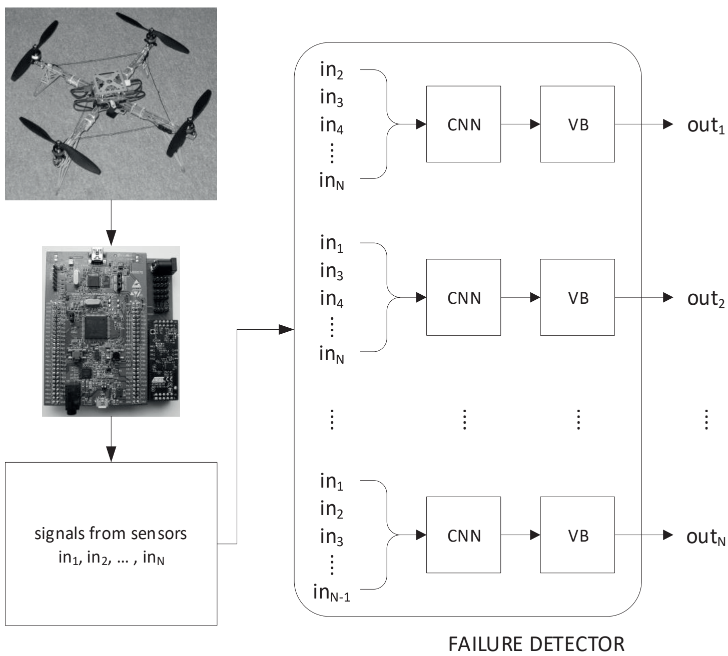

Fig. 7. Basic failure detector system – based on convolutional neural network

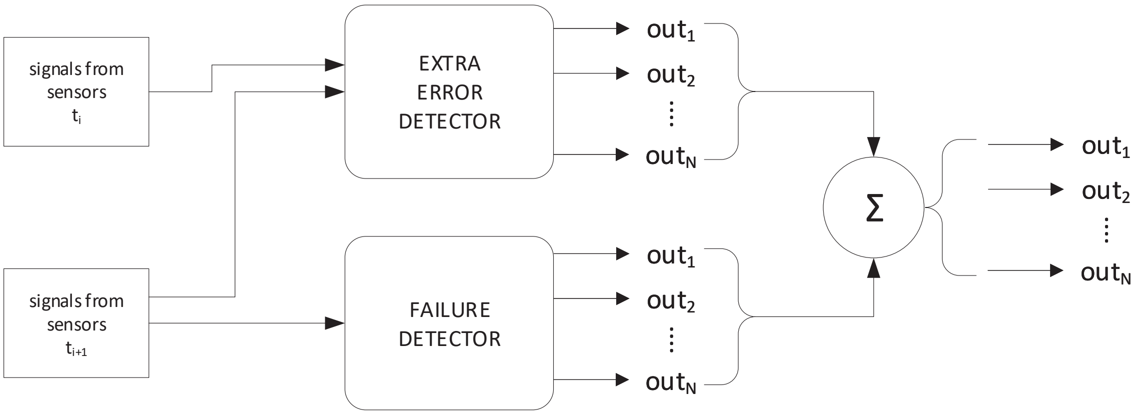

Fig. 8. A full scheme of the new failure detecting system

To add a new functionality to the sensor failure detection system we propose to include a simple block which checks the changes between Euler angles in subsequent time steps. At this moment we propose a simple extension which monitors changes between two subsequent readings from sensors. The difference between two subsequent readings from sensors can be very high for example when the object reaches 16g acceleration but it can also mean that something is wrong with one of the sensors or with the object itself. To make it clear our new failure detection system is presented in Fig. 8, where in block Extra Error Detector (EED) the comparison of two subsequent readings takes place.

To present how well does the sensor failure detection system perform we have developed a measure NDR (Normalized Detection Ratio), which is calculated with respect to the variance of the noise added to the original value of the considered signal. For a noise signal we have decided to use Gaussian one. NDRs showed in Figs. 9 and 10 converges quite quickly to

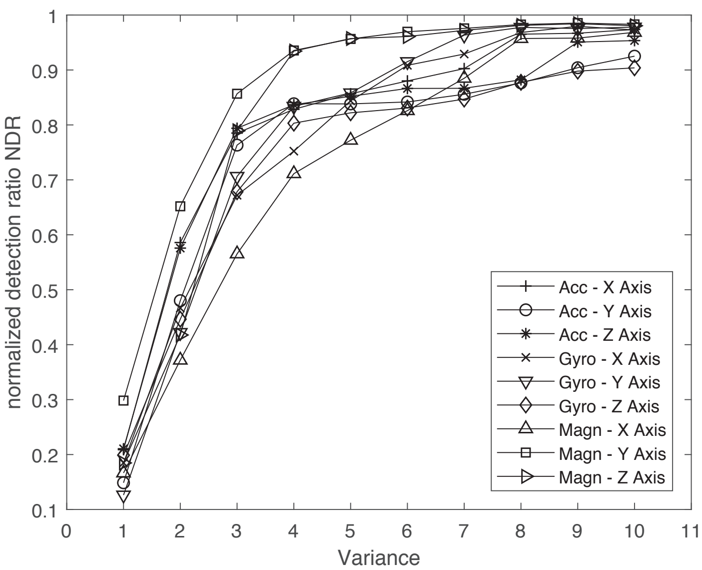

Fig. 9. Normalized Detection Ratio for failure detection system based on convolutional neural networks

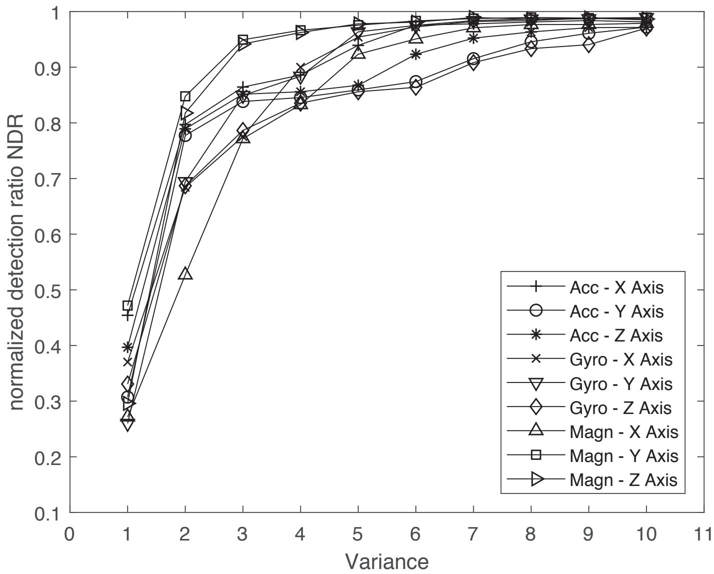

Fig. 10. Normalized Detection Ratio for failure detection system expanded with an extra error detector

a maximal value of normalized detection rate for all the sensors. The figures differ in that in the first case (Fig. 9) the system did not include the EED block, while in the second one – the EED block was taken into account. It is important to notice that NDR improves in comparison to the results presented in the previous work [2], but also adding an EED to the system improves the NDR's convergence twice – maximal value of normalized detection rate is reached twice as fast.

6. Conclusions

In this research we have presented a new failure detection system, which is built based on the system presented in one of our previous works [2]. As we already mentioned description of this system is a part of a bigger research project, which in general concerns the problem of unmanned aerial vehicles control.

Our system is based on *k*-fold cross-validation approach. It is built from convolutional neural networks, which arranged in bigger structure allows for failure detection in any sensor signal, not only in UAVs. By adding an additional block, which checks deviations in changes between subsequent signal readings, we achieved better metrics of the proposed system according to the one presented in [2].

As we already mentioned we have decided to change recurrent neural networks from our previous solution to convolutional ones because of their many advantages. One of them is simplicity of implementation (even in hardware), but surely the aspect of speed of training and obtaining values of Euler angles is one of the most important aspects. Training of the recurrent neural network can take 15–18 min and retrieving the values of Euler angles takes around 0.15 s, while convolutional neural network trains for about 60–80 s, and estimation of Euler angles takes around 0.01 s. This aspect in context of real time UAV control and fast failure detection gives a huge advantage of convolutional neural networks over the recurrent ones.

The area of the further research is very wide. It includes a comparison of our approach with other available solutions – which was not the subject of this paper and is considered as a part of our ongoing project. Another idea is related with testing our failure detection system on a completely different device. A good example can be the analysis of sensors associated with some production line, which is usually equipped with many sensors, e.g. extensometer, pressure or position sensors. A thorough analysis of these sensors working together with the possibility of detecting their malfunction can be even used to predict the exact moment of failure [37]. Another, very important aspect to be considered as the subsequent research is to focus on the aspect of anomalies that can be produced by sensors [38, 39].

References

[1] Z. Hendzel, "Collision free path planning and control of wheeled mobile robot using kohonen self-organising map", *Bull. Pol. Ac.: Tech.* 53(1), 39–47 (2005).

[2] R. Kapela, A. Świetlicka, K. Kolanowski, J. Pochmara, and A. Rybarczyk, "A set of dynamic artificial neural networks for robot sensor failure detection", in *2017 11th International Workshop on Robot Motion and Control (RoMoCo)*, 2017, pp. 199–204.

[3] K. Kolanowski, A. Świetlicka, R. Kapela, J. Pochmara, and A. Rybarczyk, "Multisensor data fusion using elman neural networks", *Appl. Math. Comput.* 319, 236 – 244 (2018).

[4] A. Świetlicka, K. Kolanowski, and R. Kapela, "Training the stochastic kinetic model of neuron for calculation of an object's position in space", *J. Intell. Robot. Syst.* 98, 615–626 (2020).

[5] M. Li and H. Lin, "Design and implementation of smart home control systems based on wireless sensor networks and power line communications", *IEEE Trans. Ind. Electron.* 62(7), 4430–4442 (2015).

[6] L. Manjakkal, M. Soni, N. Yogeswaran, and R. Dahiya, "Cloth based biocompatiable temperature sensor", in *2019 IEEE International Conference on Flexible and Printable Sensors and Systems (FLEPS)*, 2019, pp. 1–3.

[7] C.-H. Wu and Y.-C. Chung, "Heterogeneous wireless sensor network deployment and topology control based on irregular sensor model", in *Advances in Grid and Pervasive Computing* (C. Cérin and K.-C. Li, eds.), (Berlin, Heidelberg), pp. 78–88, Springer Berlin Heidelberg, 2007.

[8] W. Jiang, D. Liu, B. Cai, C. Rizos, J. Wang, and W. Shangguan, "A fault-tolerant tightly coupled gnss/ins/ovs integration vehicle navigation system based on an fdp algorithm", *IEEE Trans. Veh. Technol.* 68(7), 6365–6378 (2019).

[9] L.M.B. Winternitz, W.A. Bamford, and G.W. Heckler, "A gps receiver for high-altitude satellite navigation", *IEEE J. Sel. Top. Signal Process.* 3(4), 541–556 (2009).

[10] G. Cai, B.M. Chen, and T.H. Lee, *Unmanned Rotorcraft Systems*. Springer Publishing Company, Incorporated, 2016.

[11] Q. Fan, B. Sun, Y. Sun, and X. Zhuang, "Performance enhancement of mems-based ins/uwb integration for indoor navigation applications", *IEEE Sens. J.* 17(10), 3116–3130 (2017).

[12] H.G. Natke and C. Cempel, *Model-Aided Diagnosis of Mechanical Systems*. Springer-Verlag Berlin Heidelberg, 1997.

[13] S. Lu, M. Jiang, Q. Sui, H. Dong, Y. Sai, and L. Jia, "Multidamage identification system of cfrp by using fbg sensors and multi-classification rvm method", *IEEE Sens. J.* 15(11), 6287–6293 (2015).

[14] S.S.R. Patange, S. Raja, B. Aravindu, V.R. Ranganath, G.T. Suchithra, B.R. Kaveri, and B. Jyothi, "Wireless based sensor damage detection system for structural applications", in *2016 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT)*, 2016, pp. 312–317.

[15] T.Y. Wang, L.Y. Chang, D.R. Duh, and J.Y. Wu, "Distributed fault-tolerant detection via sensor fault detection in sensor networks", in *2007 10th International Conference on Information Fusion*, 2007, pp. 1–6.

[16] L. Yuqing, Y. Tianshe, L. Jian, F. Na, and W. Guan, "A fault diagnosis method by multi sensor fusion for spacecraft control system sensors", in *2016 IEEE International Conference on Mechatronics and Automation*, pp. 748–753, 2016.

[17] R. Isermann, *Fault-diagnosis systems. an introduction from fault detection to fault tolerance*, Springer-Verlag Berlin Heidelberg, 2006.

[18] E.C. Hall, *Reliability history of the apollo guidance computer*, MIT, Cambridge, 1972.

[19] F. Pirmoradi, F. Sassani, and C. de Silva, "Fault detection and diagnosis in a spacecraft attitude determination system", *Acta Astronaut.* 65(5), 710–729 (2009).

[20] *Breakthrough Air Force capabilities spawned by basic research*. CreateSpace Independent Publishing Platform, 2012.

[21] M.S. Grewal and A.P. Andrews, "Applications of Kalman filtering in aerospace 1960 to the present [historical perspectives]", *IEEE Control Syst. Mag.* 30(3), 69–78 (2010).

[22] E. Denti, R. Galatolo, and F. Schettini, "An attitude & heading reference system based on a kalman filter for the integration of inertial, magnetic and gps data", in *Proceedings of the 27th International Congress on Aeronautical Sciences*, (France), 2010, pp. 19–24.

[23] N. Lwin and H. Tun, "Implementation of flight control system based on kalman and pid controller for uav", *Int. J. Sci. Technol. Res.* 3, 309–312 (2014).

[24] M. Marmion, "Airborne attitude estimation using a kalman filter", Master's thesis, The University Centre of Svalbard, Longyearbyen, Norway, 2006.

[25] F. Orderud, *Comparison of kalman filter estimation approaches for state space models with nonlinear measurements*, 2005.

[26] R. Swischuk and D. Allaire, "A Machine Learning Approach to Aircraft Sensor Error Detection and Correction", *J. Comput. Inf. Sci. Eng.* 19, 041009 (2019).

[27] K. Thiyagarajan, S. Kodagoda, L. Van Nguyen, and R. Ranasinghe, "Sensor failure detection and faulty data accommodation approach for instrumented wastewater infrastructures", *IEEE Access* 6, 56562–56574 (2018).

[28] K. Thiyagarajan, S. Kodagoda, R. Ranasinghe, D. Vitanage, and G. Iori, "Robust sensor suite combined with predictive analytics enabled anomaly detection model for smart monitoring of concrete sewer pipe surface moisture conditions", *IEEE Sens. J.* 20(15), 8232–8243 (2020).

[29] M. Leccadito, T. Bakker, and R. Klenke, "A neural network approach to an attitude heading reference system", in *52nd Aerospace Sciences Meeting*, Maryland, USA, 2014.

[30] M. Battipede, M. Cassaro, P. Gili, and A. Lerro, "Novel neural architecture for air data angle estimation", in *Engineering Applications of Neural Networks* (L. Iliadis, H. Papadopoulos, and C. Jayne, eds.), (Berlin, Heidelberg), pp. 313–322, Springer Berlin Heidelberg, 2013.

[31] A.K. Ghosh, S.C. Raisinghani, and S. Khubchandani, "Estimation of aircraft lateral-directional parameters using neural networks", *J. Aircr.*35(6), 876–881 (1998).

[32] V. Kumar, O.SN.R. Ganguli, P. Sampath, and S. Suresh, "Identification of helicopter dynamics using recurrent neural networks and flight data", *J. Am. Helicopter Soc.*51, 164–174, 04 (2006).

[33] X. Kui Yue and J. Ping Yuan, "Neural network-based gps/ins integrated system for spacecraft attitude determination", *Chin. J. Aeronaut.*19(3), 233–238 (2006).

[34] S.O.H. Madgwick, "An efficient orientation filter for inertial and inertial / magnetic sensor arrays", tech. rep., 2010.

[35] J.J. Craig, *Introduction to Robotics: Mechanics and Control*. USA: Addison-Wesley Longman Publishing Co., Inc., 2nd ed., 1989.

[36] B.A. Rozenfeld, *The history of non-euclidean geometry: Evolution of the concept of a geometric space*. Springer, New York, NY, 1988.

[37] B. Jankowska and M. Szymkowiak, *Machine Ranking of 2-Uncertain Rules Acquired from Real Data*, pp. 198–222. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[38] K. Thiyagarajan, S. Kodagoda, and L. Van Nguyen, "Predictive analytics for detecting sensor failure using autoregressive integrated moving average model", in *2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2017, pp. 1926–1931.

[39] Y. Wang, N. Masoud, and A. Khojandi, "Real-time sensor anomaly detection and recovery in connected automated vehicle sensors", *IEEE Trans. Intell. Transp. Syst.* 1–11, (2020).