ARTICLE IN PRESS
JID: AMC [m3Gsc;March 11, 2017;2:22]

Applied Mathematics and Computation 000 (2017) 1–9

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Multisensor data fusion using Elman neural networks

Krzysztof Kolanowski*, Aleksandra Świetlicka, Rafał Kapela, Janusz Pochmara, Andrzej Rybarczyk

Poznan University of Technology, Faculty of Computing, Chair of Computer Engineering, Piotrowo 3A, Poznan 60-965, Poland

ARTICLE INFO

Article history:
Available online xxx

Keywords:
AHRS
IMU
Sensor fusion
Neural network
Inertial navigation

ABSTRACT

The paper presents a navigation system based on Elman Artificial Neural Network (ANN). The task of data fusion from different sensors is realized by trained ANN. Determining position in space is an issue of nonlinear hence. Not every type of ANN is used for such a task. Choice of Elman ANN was dictated by its construction and successfully applications to nonlinear problems requiring prediction. Elman network is composed of three layers. Comprises a layer of hidden layer units context which is connected to the hidden layer. Context-sensitive layer allows for store the values of previous hidden units. With this layer prediction is possible in sequential order. This is the effect of contextual memory where information is stored about what it was before. This kind of functionality is not able to provide any other standard neural network unidirectional. The system consists of MEMS (Micro Electro-Mechanical Systems) sensors, which are based on IMU (Inertial Measurement Unit). IMU is composed from gyroscopes, accelerometers and magnetometers which provide three dimensional linear accelerations and angular rates. This is a classic set of sensors for determining the position in space. The study presents the results of the implementation of algorithms for determining the position in space using trained Elman ANN. The data samples to train ANN were collected during the test flight of Quadrocopter. Paper presents the performance for different configurations of Elman ANN. Presented system provides easy addition of other sensors e.g. GPS/GLONASS receiver.

1. Introduction

The precise location in space plays an important role in many fields such as robotics [1,2], navigation [3,4], human motion analysis [5] and human-machine interface [6]. The sensors are used to accurately map the movement of the object in space. However, the sensors are not universal and not each of them is suitable for every task. Often, multiple sensors are used to measure various physical values such as: linear acceleration, angular acceleration and the magnetic field. Data collected from these sensors must be combined in order to obtain complete information on the position of the object in space.

Most of the values used in determining the position are expressed by a vector. The position is defined as a displacement between the two coordinate systems. The first one can be the ground or the starting point of the sensor calibration. The second one is the navigation object e.g. quadrocopter. Variables such as direction, speed, acceleration and movements can be easily calculated from based on the change of position [7,8]. Such calculations are performed as multiplying the vector representing the position of one coordinate system to another by a 3×3 rotation matrix. This representation is referred to

* Corresponding author.
E-mail address: krzysztof.kolanowski@put.poznan.pl (K. Kolanowski).

http://dx.doi.org/10.1016/j.amc.2017.02.031
0096-3003/© 2017 Elsevier Inc. All rights reserved.

as a representation of direction cosines or DCM (Direction Cosine Matrices). The conversion of the angular position between the Euler angles and DCM representation given by the rotation Matrix R

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

where:

ϕ - Roll; θ - Pitch; ψ - Yaw.

Another, often used method to define position of the object in space is to use a quaternion math. Quaternion is a four-dimensional complex number that can be used to represent the orientation of a rigid body or coordinate frame in three-dimensional space.

Data connections are made by using mathematical transformations describing the position in space using quaternions. Quaternion q can be written in the following form:

$$q = q_0 + iq_1 + jq_2 + kq_3 = (q_0, q_1, q_2, q_3) \tag{2}$$

$$i = \begin{bmatrix} i & 0 \\ 0 & -i \end{bmatrix} \quad j = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad k = \begin{bmatrix} 0 & i \\ i & 0 \end{bmatrix} \tag{3}$$

where:

i, j, k are special quaternion matrixes, q_0, q_1, q_2 are the vector components, q_3 is the scalar component of the quaternion, and:

$$\begin{aligned} q_0 &= e_x \sin\left(\frac{\vartheta}{2}\right) \quad q_1 = e_y \sin\left(\frac{\vartheta}{2}\right) \\ q_2 &= e_z \sin\left(\frac{\vartheta}{2}\right) \quad q_3 = \cos\left(\frac{\vartheta}{2}\right) \end{aligned} \tag{4}$$

where:

e_x, e_y, e_z are the principal axis ϑ is the principal angle.

Transforming the quaternion to the Euler representation of angles is simple and is done with the following formulas:

$$\phi = \tan^{-1}\left(\frac{2(q_1q_2 + q_0q_3)}{q_3^2 - q_2^2 - q_1^2 + q_0^2}\right) \tag{5}$$

$$\theta = \sin^{-1}\left(-2(q_0q_2 + q_1q_3)\right) \tag{6}$$

$$\psi = \tan^{-1}\left(\frac{2(q_0q_1 + q_3q_2)}{q_3^2 - q_2^2 - q_1^2 + q_0^2}\right) \tag{7}$$

2. Multisensor data fusion

To train the ANN we used Euler angles calculated with the AHRS algorithm. Designed ANN has nine inputs representing the various axes of each of the sensors, while at the output there are three signals corresponding to the description of the position in space of Euler angles (Roll, Pitch, Yaw).

To record samples from the sensors we have built a special equipment showed in Fig. 1. We used Atmel IMU module ATAVRSBIN1 features a three-axis accelerometer, three-axis gyroscope and three-axis magnetometer. To read and store the data we used development board STM32F4-Discovery.

The connections between components have been implemented using custom PCB containing an additional power block, communication via Bluetooth and SD card slot. Samples from the accelerometer and gyro were recorded at a frequency of 50 Hz while frequency of magnetometer was 10 Hz. Collected data was recorded on SD Card, then ripped to a PC where all data were processed with use MATLAB environment.

Attitude Heading and Reference Systems are able to provide a complete measurement of orientation relatively to the direction of gravity and the Earth's magnetic field. An orientation estimation algorithm is a fundamental component of any IMU system. It is required to join together the separate sensor data into a single, optimal estimation of orientation.

Typical sensors used to form an AHRS are accelerometers to measure the proper acceleration along the three axes of the UAV body coordinate, gyroscopes to provide the three-axis angular rates and magnetometers to capture the magnet values of the three axes (Fig. 2) [9,10].

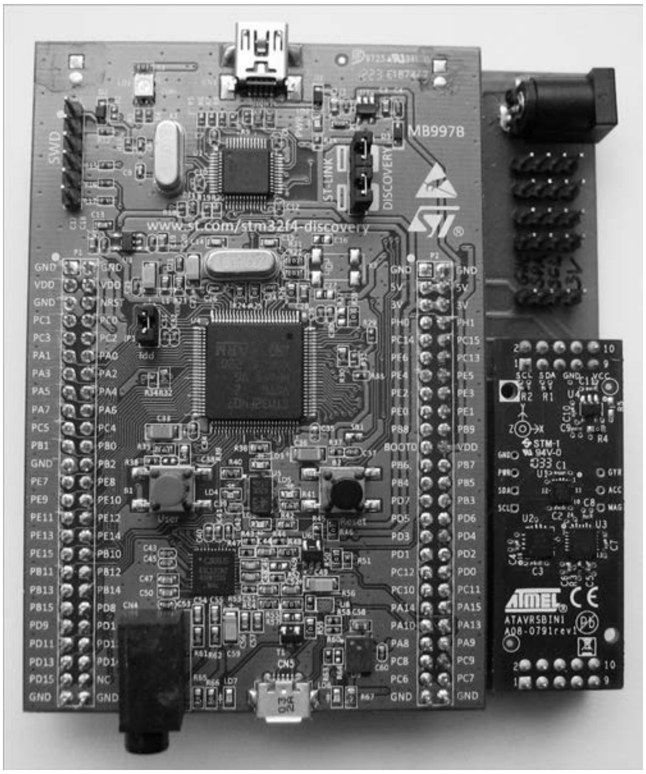

Fig. 1. Samples recorder device.

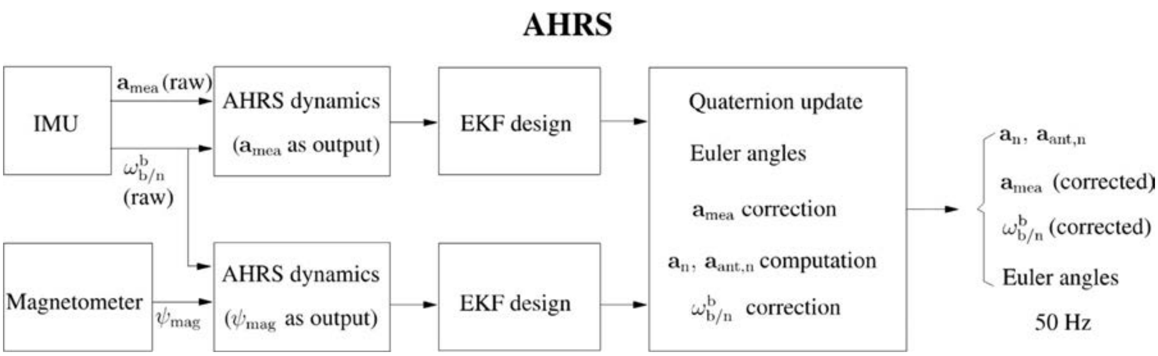

Fig. 2. Scheme of AHRS [10].

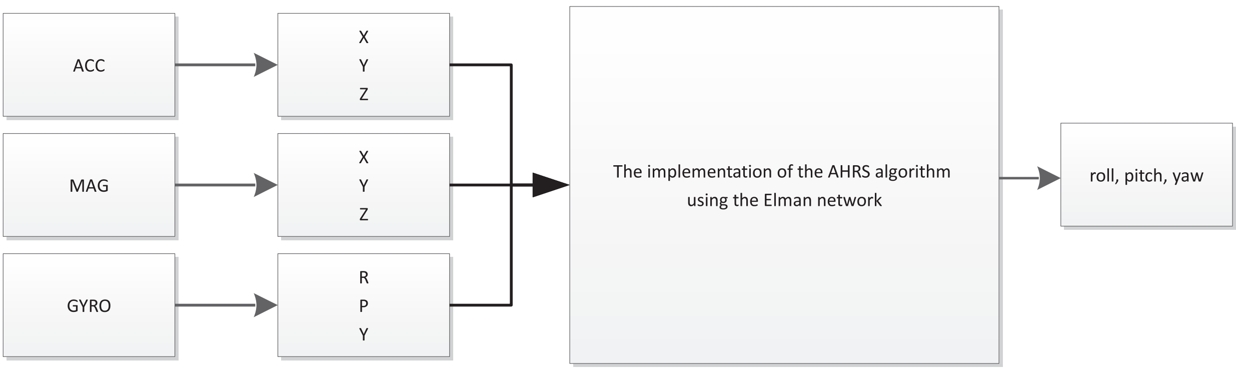

Fig. 3. Block diagram of the system.

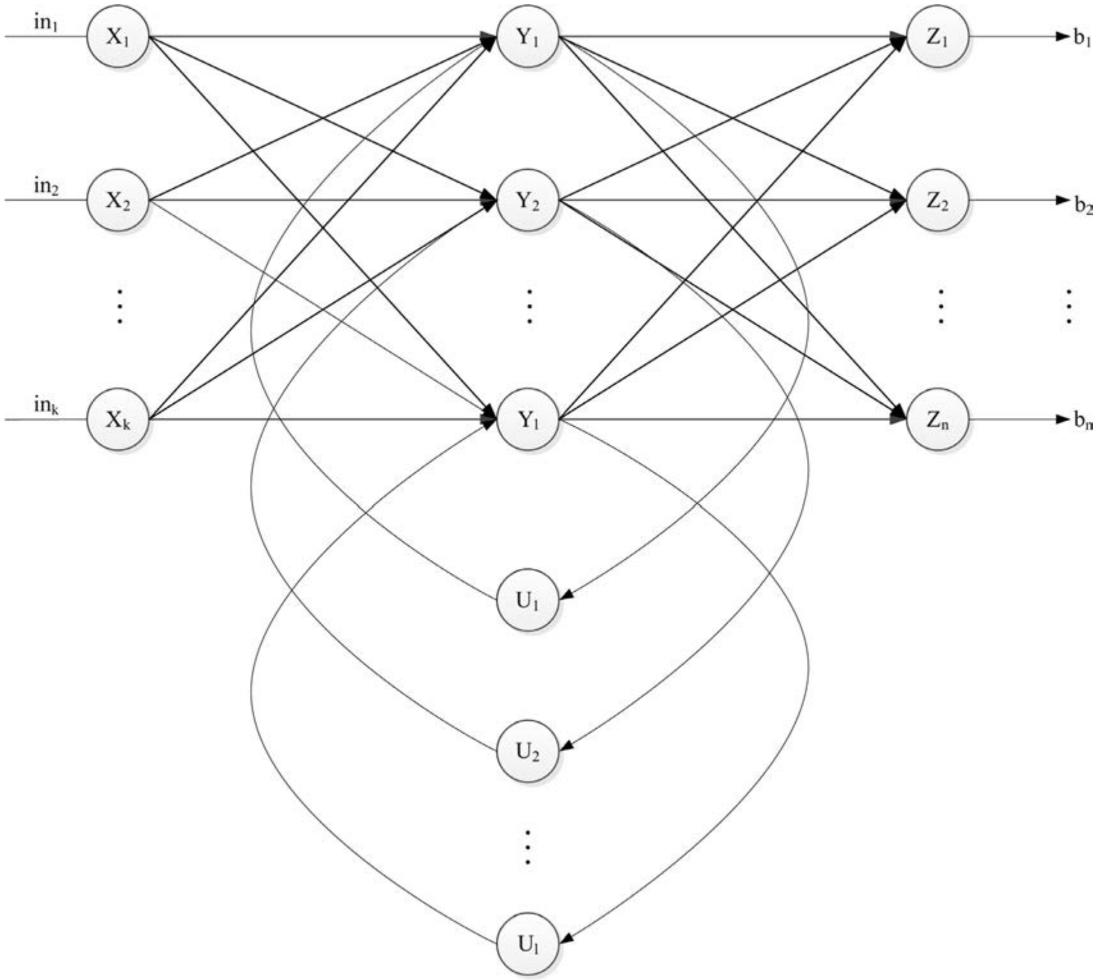

Fig. 4. Elman Neural Network.

The AHRS algorithm can be evaluated according to the following steps:

- Read samples

$$\begin{aligned} \mathbf{a} &= (a_x,\ a_y,\ a_z) \\ \omega &= (\omega_x,\ \omega_y,\ \omega_z) \\ \mathbf{H} &= (H_x,\ H_y,\ H_z) \end{aligned} \tag{8}$$

- Normalize measurements

$$\tilde{\mathbf{H}} = (0,\ \mathbf{H}) \tag{9}$$

$$\mathbf{b} = (0,\ \|(h_2,\ h_3)\|,\ 0,\ h_4) \tag{10}$$

$$\mathbf{q} = \mathbf{q}/\|\mathbf{q}\| \tag{11}$$

- Reference direction of Earth's magnetic field

$$\mathbf{h} = \mathbf{q} \otimes \left(\tilde{\mathbf{H}} \otimes \bar{\mathbf{q}}\right) \tag{12}$$

- Estimated direction of gravity and magnetic field

$$\mathbf{v} = \begin{pmatrix} 2(q_2q_4 - q_1q_3) \\ 2(q_1q_2 + q_3q_4) \\ q_1^2 - q_2^2 - q_3^2 + q_4^2 \end{pmatrix}^T \tag{13}$$

$$\mathbf{w} = \begin{pmatrix} 2b_2\left(0.5 - q_3^2 - q_4^2\right) + 2b_4(q_2q_4 - q_1q_3) \\ 2b_2(q_2q_3 - q_1q_4) + 2b_4(q_1q_2 + q_3q_4) \\ 2b_2(q_1q_3 + q_2q_4) + 2b_4\left(0.5 - q_2^2 - q_3^2\right) \end{pmatrix}^T \tag{14}$$

- Compute error as sum of cross product between estimated direction and measured direction of fields

$$\mathbf{e} = \mathbf{a} \times \mathbf{v} + \mathbf{H} \times \mathbf{w} \tag{15}$$

- Apply feedback terms

$$\omega = \omega + K_p \cdot \mathbf{e} \tag{16}$$

- Compute rate of change of quaternion

$$\dot{\mathbf{q}} = 0.5(\mathbf{q} \otimes (0,\ \omega)) \tag{17}$$

- Integrate to yield quaternion

$$\mathbf{q} = \mathbf{q} + \dot{\mathbf{q}}/256 \tag{18}$$

- Determine the position using the rotation matrix

$$R = \begin{pmatrix} 2q_1^2 + 2q_2^2 - 1 & & \\ 2(q_2q_3 - q_1q_4) & & \\ 2(q_2q_4 + q_1q_3) & 2(q_3q_4 - q_1q_2) & 2q_1^2 + 2q_4^2 - 1 \end{pmatrix} \tag{19}$$

where:

- **a** - linear acceleration,
- **ω** - angular acceleration,
- **H** - magnetic flux,
- **q** - quaternion,
- $\tilde{\mathbf{H}}$ - magnetic flux quaternion,
- **h** - reference direction of Earth's magnetic field,
- **b** - normalized **h**,
- **v** - direction of gravity,
- **w** - direction of magnetic field,
- **e** - error,
- $\dot{\mathbf{q}}$ - rate of change of quaternion.

ANNs as a great nonlinear function approximator give a great perspective for determining the object's position. The advantage of ANN is not only the possibility of faster response but also avoiding a number of advanced trigonometric calculations.

3. Artificial Neural Network

Artificial Neural Network can be an alternative to complicated calculations which occur in the AHRS algorithm. ANN that has good training data and appropriate achieves training algorithm positive results as the standard calculation in shorter time to response. A large number of trigonometric operations and matrix operations are computationally and time expensive. For implementation of AHRS we have chosen an Elman type of ANN.

Elman ANN is a simple example of a recurrent neural network. Elman ANN is composed of three layers. Its structure contains at least one hidden layer from which the feedback is led. Includes a layer of hidden layer units context which is connected to the hidden layer. Context-sensitive layer allows for store the values of previous hidden units. With this layer prediction is possible in sequential order. This is the effect of contextual memory where information is stored about what it was before. This kind of functionality is not able to provide by any other standard unidirectional ANN.

4. Experimental results

Fig. 5 shows the data collected from the three axes of three sensors. The samples were then analyzed with the help of the algorithm described in Section 2, so processed data is shown in Fig. 6. That data was used to train ANN.

In all experiments we used Levenberg–Marquardt algorithm of training for different number of neurons in the hidden layer and different sizes of the feedback. In Table 1 we summarize some errors obtained during these experiments and training time of ANN in MATLAB (Intel Core i5 M520 2.4 GHz, 8 GB RAM).

Each of the configurations had nine inputs and three outputs. The shape of the neural network for last configurations is shown in Fig. 7.

Each configuration has been checked by a test sequence to obtain samples from the output of Elman ANN to estimated values of the Euler angles of the position in space.

Comparison of the results obtained by direct calculation using the AHRS algorithm and by the implementation of the ANN for Roll axis is shown in Fig. 8.

Comparing the output values obtained with the artificial neural network, and the value obtained from the calculation of the angular position in space is presented in Fig. 9. We can observe that the increased number of neurons in the hidden layer has no significant effect on the accuracy of estimated values. Increasing the number of neurons in the feedback loop allows the parameters noted improvement obtained by using of artificial neural network.

Please cite this article as: K. Kolanowski et al., Multisensor data fusion using Elman neural networks, Applied Mathematics and Computation (2017), http://dx.doi.org/10.1016/j.amc.2017.02.031

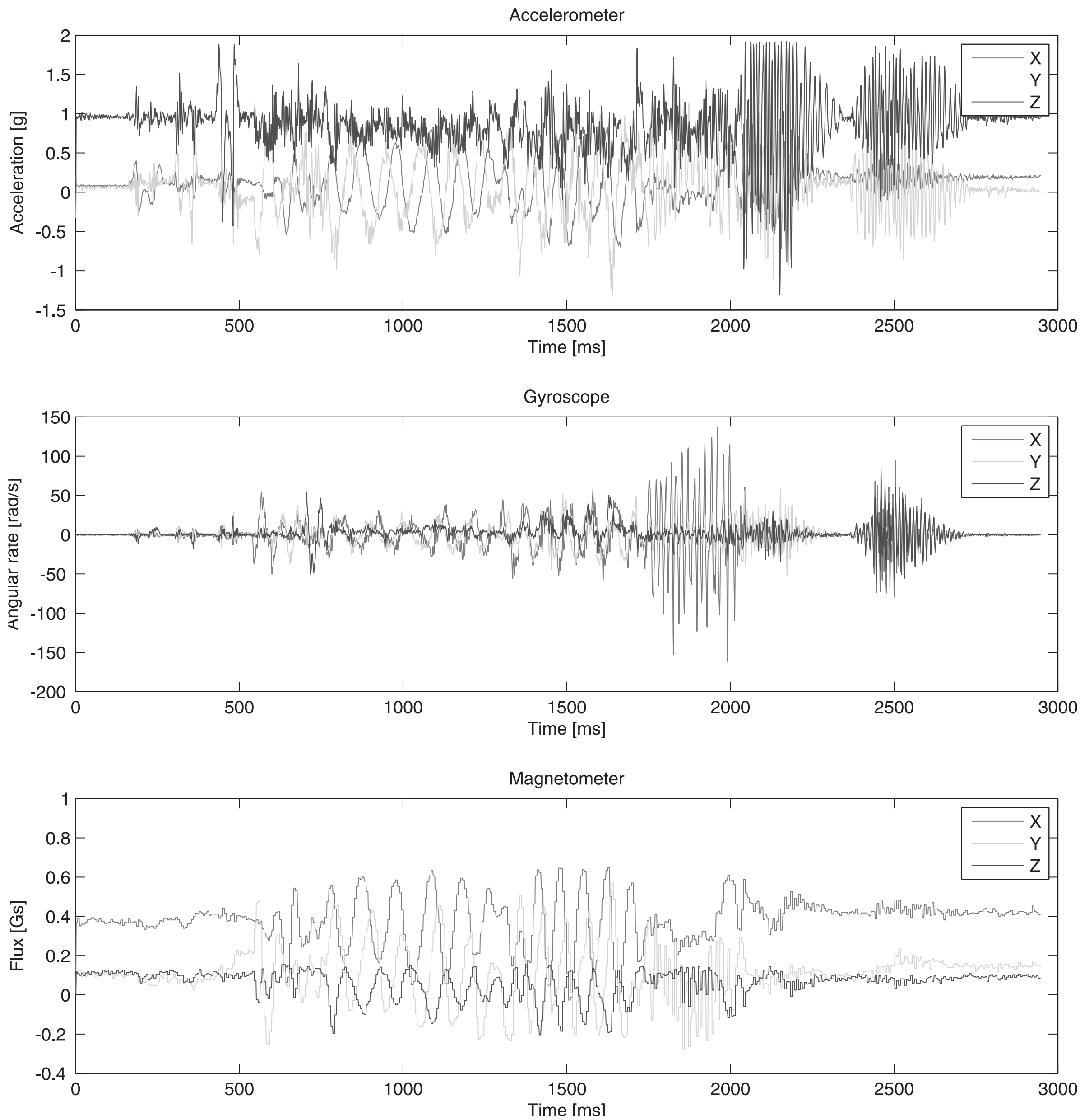

Fig. 5. Collected data samples.

Table 1
Error value depending on the configuration of an ANN.

No.	No. of neurons in the hidden layer	No. of neurons in the feedback	Error (roll)	Error (pitch)	Error (yaw)	Training time (hh:mm:ss)
1	10	6	04206	03299	04256	00:04:10
2	20	8	17965 E-4	20845 E-4	17270 E-4	01:59:11
3	30	12	14272 E-4	11195 E-4	14154 E-4	22:45:29
4	40	16	16044 E-4	19055 E-4	18762 E-4	38:41:32
5	50	20	20893 E-4	27618 E-4	27481 E-4	62:14:20

Please cite this article as: K. Kolanowski et al., Multisensor data fusion using Elman neural networks, Applied Mathematics and Computation (2017), http://dx.doi.org/10.1016/j.amc.2017.02.031

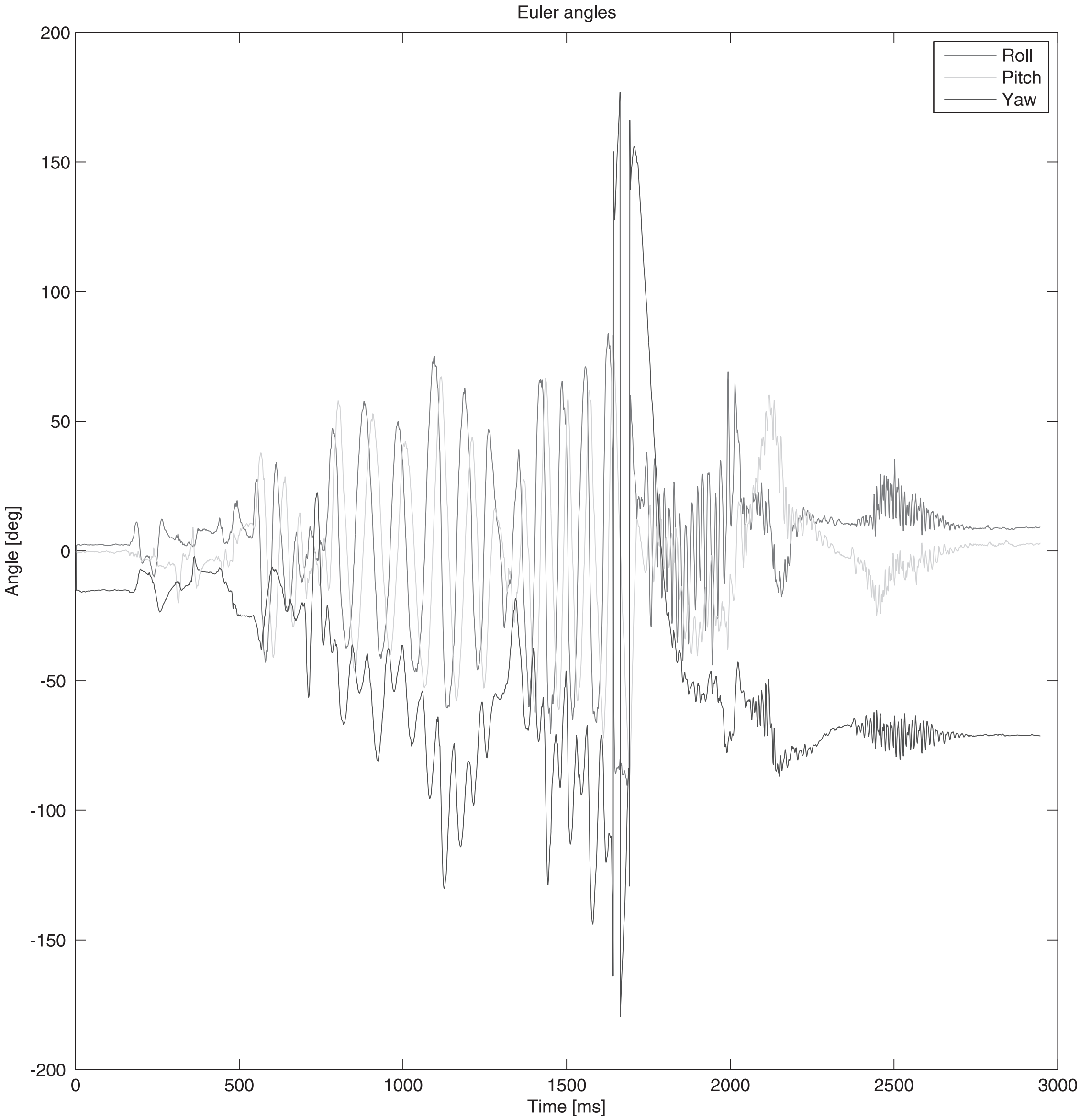

Fig. 6. Data after the fusion.

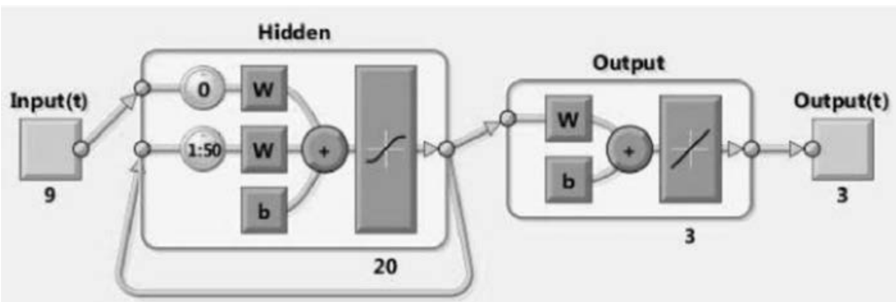

Fig. 7. Visualization of the last configuration of ANN.

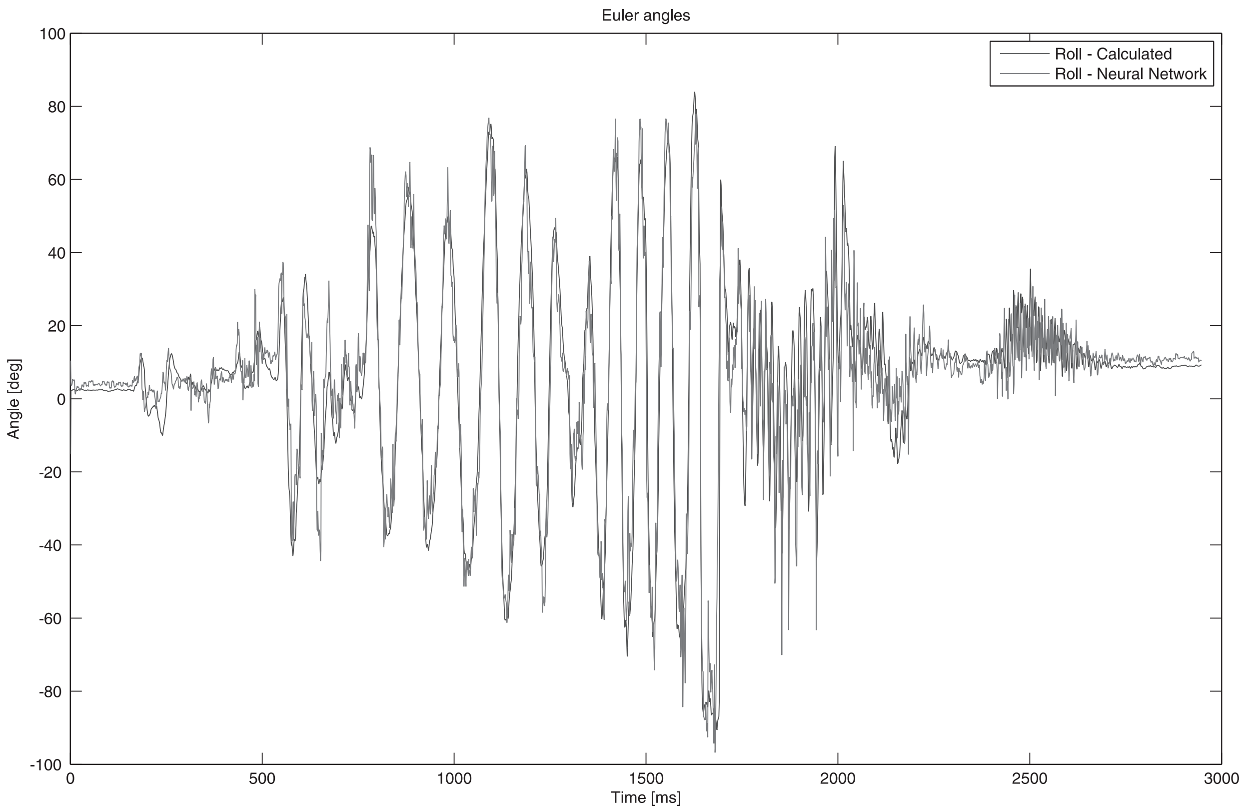

Fig. 8. Results of first configuration ANN vs calculated AHRS.

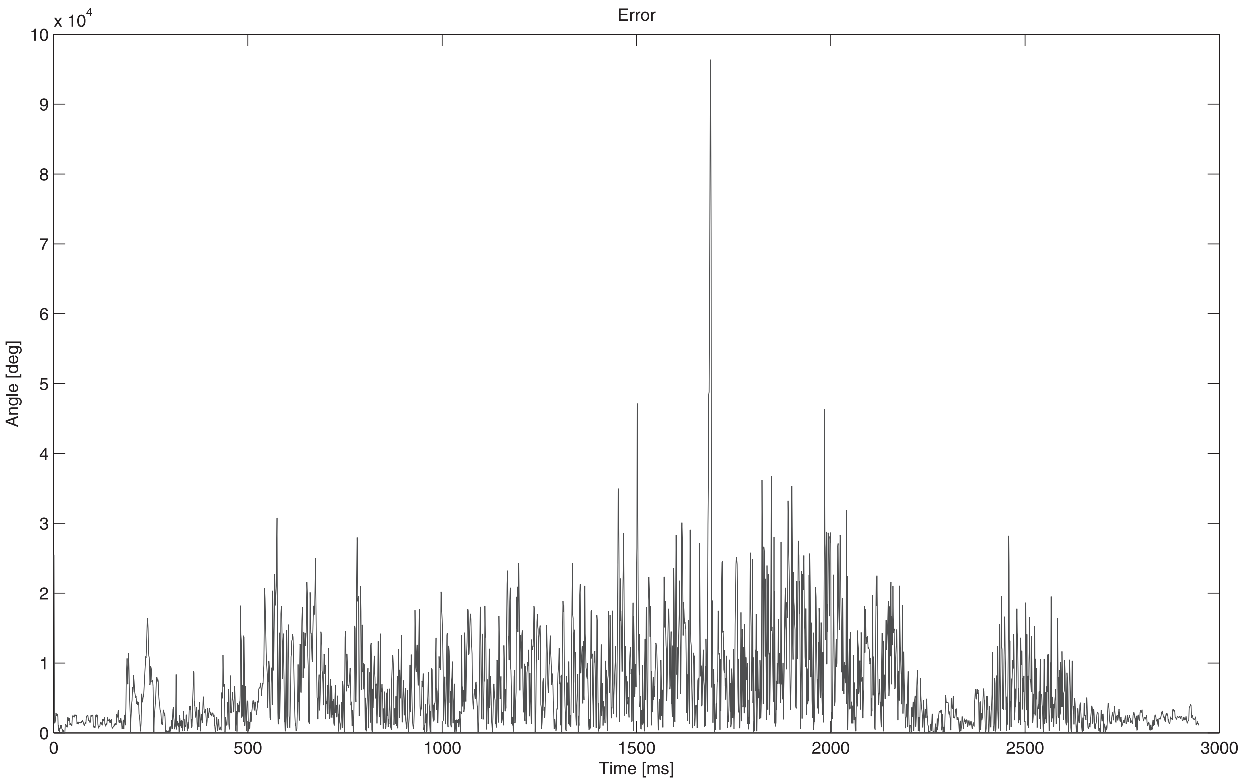

Fig. 9. Error of last configuration.

5. Conclusions

The obtained results allow to conclude that the assumptions were correct. Comparing the results it can be concluded that the values obtained using the ANN do not differ significantly from these calculated using the AHRS algorithm.

However, comparing the differences between the values of the ANN output and the calculated values can be seen a large error values. This is caused by the high dynamics of the neural network input data. Future studies need to prepare more data sets to learn the network.

This shows that we can successfully use the ANN as an alternative to complicated calculations. However, one should carry out a more thorough investigation of the selection of ANN parameters.

Future research should be enriched with additional sensors such as GPS receiver and barometer. These sensors will improve the positioning accuracy. And that will make the research will be more varied with more variables for analysis. It is also planned to compare the results with the use of other types of artificial neural networks.

References

[1] B. Barshan, H.F. Durrant-Whyte, Inertial navigation systems for mobile robots, in: Proceeding of the IEEE Transactions on Robotics and Automation, Vol. 11, 1995, pp. 328–342. June.
[2] L. Ojeda, J. Borenstein, Flexnav: fuzzy logic expert rule-based position estimation for mobile robots on rugged terrain, in: Proceeding of the IEEE International Conference on Robotics and Automation ICRA 02, Vol. 1, 2002, pp. 317–322. May 11–15.
[3] D.H. Titterton, J.L. Weston, Strapdown Inertial Navigation Technology, The Institution of Electrical Engineers, 2004.
[4] S. Beauregard, Omnidirectional pedestrian navigation for first responders, in: Proceeding of the 4th Workshop on Positioning, Navigation and Communication WPNC 07, 2007, pp. 33–36. Mar. 22.
[5] H.J. Luinge, P.H. Veltink, Inclination measurement of human movement using a 3-d accelerometer with autocalibration, in: Proceeding of the IEEE Transactions on Neural Systems and Rehabilitation Engineering, Vol. 12, 2004, pp. 112–121. Mar.
[6] H. Zhou, H. Hu, Human motion tracking for rehabilitation–a survey, Biomed. Signal Process. Control 3 (1) (2008) 118.
[7] J.B. Kuipers, Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality, Princeton University Press, 1999.
[8] Y.B. Itzhack, New method for extracting the quaternion from a rotation matrix, AIAA J. Guidance, Control Dyn. 23 (6) (2000) 1085–1087.
[9] F. Caron, E. Duflos, D. Pomorski, P. Vanheeghe, GPS/IMU data fusion using multisensor Kalman filtering, Inf. Fusion 7 (2006) 221–230.
[10] G. Cai, B. Chen, T. Lee, Unmanned Rotorcraft Systems, Springer, London, Dordrecht, Heidelberg, New York, 2011.

Please cite this article as: K. Kolanowski et al., Multisensor data fusion using Elman neural networks, Applied Mathematics and Computation (2017), http://dx.doi.org/10.1016/j.amc.2017.02.031