9th International Conference on Computer and Knowledge Engineering (ICCKE 2019), October 24-25 2019, Ferdowsi University of Mashhad

Robust Real-time Magnetic-based Object Localization to Sensor's Fault using Recurrent Neural Networks

[1]Sara Naseri-Golestani, [1]Hamed Rafiei, [1]Mohammad-R. Akbarzadeh-T., [2]Alireza Akbarzadeh, [2]Amirmohammad Naddafshargh, [3]Sadra Naddafshargh
[1]Dearment of electrical engineering, Center of Excellence in Soft Computing and Intelligent Information Processing (SCIIP),
[2]Department of mechanical engineering, Center of Excellence in Soft Computing and Intelligent Information Processing (SCIIP),
[3]Department of computer engineering
[123]Ferdowsi University of Mashhad.
Emails: Naserigolestani.sara, rafiei.hamed, akbazar, ali_akbarzadeh, amirmohammad.nadaffshargh, sadranaddafshargh@mail.um.ac.ir

Abstract—**Magnetic sensors often experience faults such as no-response, noisy signal, and saturation. Yet, they have considerable object localization applications that require high precision, such as in medical operations. Conventionally, Dipole Magnetic (DM) position tracking is used for magnetic localization, even while a sensory fault occurs. But DM position tracking is not sufficiently accurate, and its computational cost is a matter of concern. Accordingly, the proposed approach here is in three folds. First, we propose to use a heuristic to detect faulty sensors and to stop the propagation of faulty reading by setting their readings to zero. Second is using a nonlinear modeling platform, Recurrent Neural Network (RNN) for the actual nonlinear mapping of the magnet sensory readings and placement due to its' accurate outputs. And third is to prepare a sufficiently rich data set for training the network that is prepared under no sensory fault. The experimental study here confirms that the faulty sensory reading is successfully identified and set to zero by the proposed heuristic, and the nonlinear mapping of the neural network provides a good assessment of magnet localization even when the corresponding inputs from faulty sensors are set to zero. The experimental setup here consists of a network of eight magnetic sensors, one of which becomes faulty during the experimentation process. More specifically, results show that the accuracy of our method has improved up to 444.3% to DM method and its robustness enhanced to 105.3% to an RNN which is trained without our rich data set.**

Keywords—Sensor fault detection; ANN; Sensor fault compensation; Real-time; Permanent magnet.

I. INTRODUCTION

Nowadays, the use of permanent magnets is growing in many applications, such as in medical and military service operations. Permanent magnets have a significant place in such uses due to their magnetic field, which is capable of being tracked by magnetic sensors. The magnetic-based localization is based on the magnetic sensor's data, which are a measurement of the magnetic field in three different axes. Although all of these localizations are based on the data which come from magnetic sensors, sometimes they might send us the wrong data.

However, because of the importance of their use, it is so much important to realize which data are not damaged and which are intact. Therefore, we require to make our RNN, which is a precise network, robust to such impairments using the help of a rich data set for the training process and also a procedure to detect faults in the test phase.

Magnetic sensors are such devices which are being used in places where there is a need for remote controlling such as medical surgeries, medical prosthesis. These sensors can sense the location of strong magnetic fields such as the magnetic field in a permanent magnet and send back data which show the power of that field. An example of how much this data sending system is important is provided in [1], which is using magnetic sensors' data to localize a magnet in human's nose and esophagus and lastly its stomach. Zhenglong Sun et al.[1] has aimed to improve the speed and accuracy of certain actions, such as Ventriculostomy and NG-Tube, which use remote control devices to insert some pieces into the body, which is a risky action. Therefore, magnetic localization is suggested to improve these actions and to reduce or even prevent errors which lead to serious problems. In this work, they have tried two methods for magnetic localization: DM method and Artificial Neural Networks (ANNs). The NN method shows a better result in the accuracy and speed of calculation. In [2] the authors also concentrate on magnet localization with DM method. In this paper, they locate four magnets separately and simultaneously, which faced them with problems. they report time delay because of the large number of calculations and low accuracy due to other magnetic fields around. In electronic devices which are controlled by sensors' data, if a fault happens in one of the sensors, it will probably cause controlling faults in the master device which might make insufferable damages.

9871-7281-5075-8/19/$31.00

RNNs are such networks which contains a memory inside when there is a need of information from last steps. These networks are good in solving high computational problems like in [3]. There is also a usage of RNN for object tracking in [4].

In this work due to the low accuracy and time-consuming calculation process of DM model, we are trying to detect and compensate fault occurrence in magnetic sensors using an embedded ANN in the proposed algorithm. After the training part of ANNs, the process will get so fast and its computing time will be efficient.

In this paper, Section II provides related work to our method. Section III is the preliminary concepts needed for the proposed algorithm. Our algorithm is given in Section IV; and the results are presented in Section V. Conclusion of our method's efficiency is presented in the last Section VI.

II. RELATED WORKS

In 2016, an investigation on sensors' fault has been done to check if these faults can be detected or make no problem for localizing in [5]. They presented two methods for passive magnetic localization based on inverse optimization method and the direct ANN. Based on the information of this work for the ANN method when a fault of a single sensor is detected because of too many connections it is impossible to reduce the inputs each time a sensor fault is detected. Therefore, the network is using the last observation before this fault happens and for the next localizations, it will localize with the inverse optimization method. But in DM position tracking they will ignore the data of detected sensor and do the rest of localization. Although the DM method's computational cost is in a matter of concern, it does not need to learn before operations start.

Moreover, some work has used Kalman filters to detect and identify the fault [6, 7]. The work of Bayesian Belief Network (BNN) in [6] use Kalman filters for stochastic state estimator and also Levenberg observers to identify the fault that has been detected with Kalman filters. In [6] each fault detector is a Kalman filter with a specific embedded failure model. With this method, two separated faults can be identified. Combination of neural networks with fuzzy logic and signal processing has made a knowledge-based model which in [8] they call it model-free method for detecting faults.

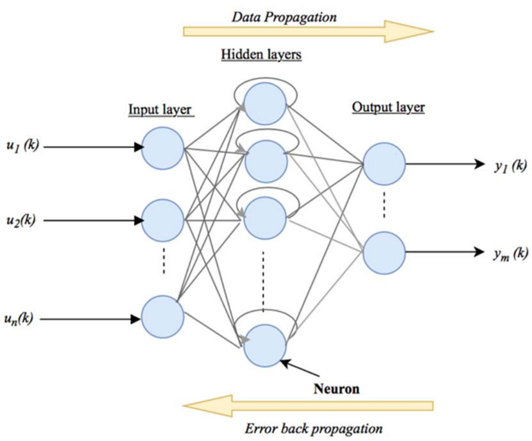


Fig. 1: RNN inside the architecture

In addition, to detect faults with the help of Kalman filters, to make robustness to the problem of fault detection, some ideas are presented [9, 10]. The first opinion is based on Non-linear Unknown Input Observer (NUIO) which is on the basis of capture rule to make robustness to sensor fault detection [9]. The result in this work [9] expresses the idea that this method has superiority to methods of extended Kalman filter and previous NUIOs. Ligang Wu et al. [8] have set filter of dependent and independent fuzzy rules to make the challenge of fault detection more robust.

III. PRELIMINARIES

In this section, the essential materials we require is introduce. First, the neural networks we use and the next material is the kind of faults we detect.

A. Neural network

The network we use in this paper is a RNN with error back-propagation algorithm. The architecture of the inside of this network is present in Fig. 1.

Another exhibition of RNN's inside architecture is placed in Fig. 2 which can be modify with the diagrammatic rule to Fig. 3.

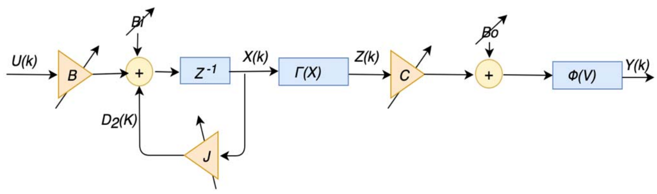


Figure 2:Topology of RNN

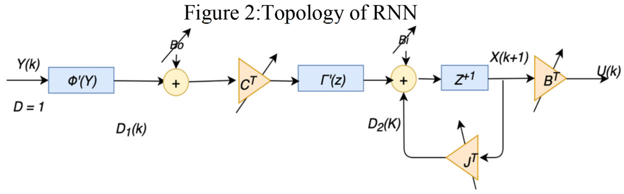


Figure 3:The adjoint topology of RNN

The basic RNN topology has an input $U(k)=[u_1, u_2, \ldots, u_n]$ and the estimated output of $Y(k)=[y_1, y_2, \ldots, y_m]$ that the $X(k)$ is presented as the internal state, $Y_P(k)$ is the real output and $Z(k)$ as hidden layers state vectors. J, B, and C are real-valued weight matrices and finally Bi and Bo are matrices of biases values which have random value at first.

The formulation of this architecture (Fig. 3) is illustrated below:

$$X(k+1)=J(k)*X(k)+B(k)*U(k)+Bi(k) \tag{1}$$

$$E(k)=Y_P(k)-Y(k) \tag{2}$$

$$Z(k)=\Gamma[X(k)] \tag{3}$$

$$V(k)=CZ(k)+Bo \tag{4}$$

$$Y(k) = \Phi[V(k)] \tag{5}$$

Where $E(k)$ is the estimated error of the k_{th} data $\Gamma[.]$ and $\Phi[.]$ are activation functions and $V(k)$ is the output of the last layer's summation.

The formulation of the Back-Propagation algorithm is also as below:

$$D_1(k)=\Phi'[Y(k)] \tag{6}$$

$$D_2(k)=\Gamma'[Z(k)]CD_1(k) \tag{7}$$

$$DY[C(k)]=\partial Y(k)/\partial C(k)=D_1(k)Z^T(k) \tag{8}$$

$$DY[J(k)]=\partial Y(k)/\partial J(k)=D_2(k)X^T(k) \tag{9}$$

$$DY[B(k)]=\partial Y(k)/\partial B(k)=D_1(k)U^T(k) \tag{10}$$

$$DY[Bi(k)]=D_1(k) \tag{11}$$

$$DY[Bo(k)]=D_2(k) \tag{12}$$

$$C(k+1)=C(k)+DY[C(k)] \tag{13}$$

$$J(k+1)=J(k)+DY[J(k)] \tag{14}$$

$$B(k+1)=B(k)+DY[B(k)] \tag{15}$$

$$Bi(k+1)=Bi(k)+DY[Bi(k)] \tag{16}$$

$$Bo(k+1)=Bo(k)+DY[Bo(k)] \tag{17}$$

Where $\Phi'[Y(k)]$ is the derivation of Φ to $Y(k)$, $\Gamma'[Z(k)]$ is the derivation of Γ to $Z(k)$ and $DY[C(k)]$, $DY[J(k)]$, $DY[B(k)]$, $DY[Bi(k)]$ and $DY[Bo(k)]$ are the derivation of $Y(k)$ to $C(k)$, $J(k)$, $B(k)$, $Bi(k)$ and $Bo(k)$ respectively.

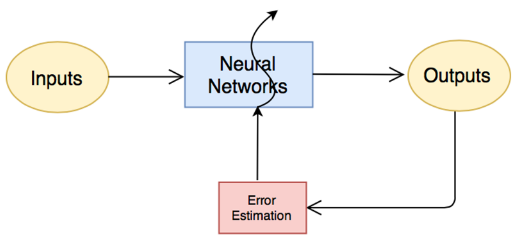


Fig. 4: Training Process

In Fig. 4 the training step of our RNN is proposed, in which is shown we first trained our network to set its weights.

B. Candidate faults

In [1] the authors present three different faults that happen mostly in magnetic sensors. These electronic materials are such devices we are now controlling our setup with. These three faults are:

- No-Response: before the localization starts, all sensors should be calibrated in one starting point such as zero due to the remaining data in them from their last localization. After calibration, localization of our permanent magnets begins. If any of the sensors send the same data as their starting point after the beginning of localization, this sensor has got a fault and in [1] they have named it No-Response mode.

- Noise: when sensors are sending their data, which show the amount of magnetic field they sense. In this data transmission, if one of the sensors suddenly starts to send data which are out of our normal range (more than ε) and is too different with its neighbor sensors' data, this sensor is sensing noise and its data should not be considered in our localization.

- Saturation: Sometimes, after a while, magnetic sensors data would not change and it will be stuck at a high amount, which is a phase of saturation, and when a saturation happens, these sensors will not send correct data till we calibrate them all again.

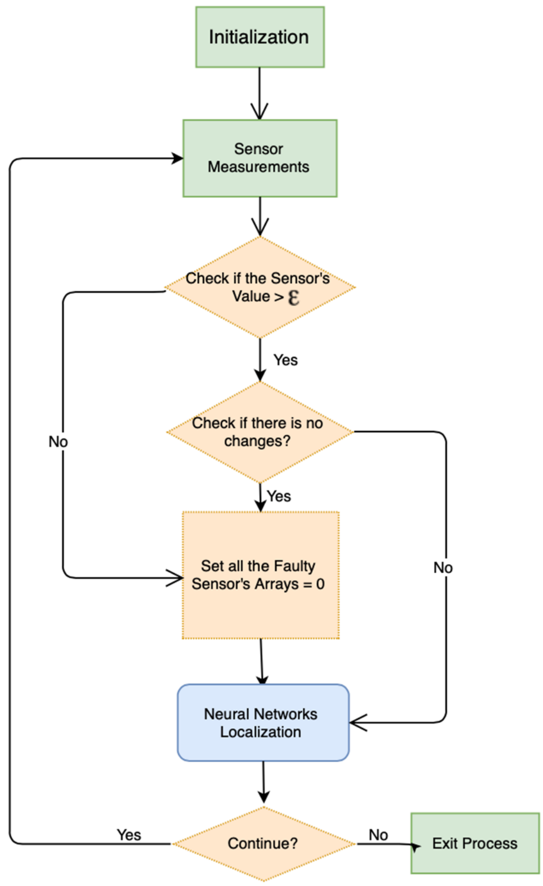


Fig. 5: Proposed method algorithm

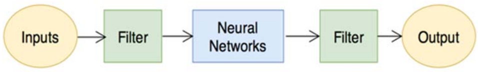


Fig. 6: Test Process (filters is before input and output of the procedure are rloess filters to make the results smooth.)

III. Proposed Method

Fig. 5 expresses the method we are using to solve the localization problem when a fault happens in one or even more that one sensor. This algorithm is robust to sensors faults.

After initialization, sensors' data would be sent with SPI communication with an STM32 micro-controller to our system and a table on Matlab receives it that each row shows the measurements of the magnetic field from a specific sensor in three axis X, Y, Z. After receiving data from our magnetic sensors, they should be checked if they have an amount more than a normal point (ε) and if their measurements did not change at all. In the first place, we are facing with Noise detected the fault and in the next step, we are having either saturation or no-response phase. The solution to all of the faults named above, after this detection, is to set the data of the row which belongs to the detected sensor into zero and send them to RNN which we use in this algorithm.

To make our RNN respond to this detection, a well-combined dataset is needed. These data are a combination of datasets that in each of them there is one of the fault possibilities, for example for N sensors there are (2^N -2) different states that sensors might be affected with faults, except the state that all sensors affect with fault. So to make a rich dataset with combining all these possible states and the non-affected data to one rich dataset. Lastly, this rich dataset will be sent directly to our training process input. Our neural network localization test phase diagram is presented in (Fig. 6) which contains two filters in the input and output of our network. After passing data through the rloess filter to get smoother, they will be sent to an RNN to give us the location of our magnets.

IV. Simulation Results

The RNN that is used in this work has 3 hidden layers with 20, 10 and 5 neurons respectively from 24 neurons input to 2 neurons of output. First, this network will be trained with 2000 epochs on a zigzag movement rich dataset. After setting all weights in this phase, and also after passing input data through the rloess filter, which is a filter for damping noises in Matlab, our algorithm will be tested on a curve movement. The output data would also get a pass to the rloess filter for better results. Despite the fact that the training phase is computationally time-consuming, the test time would be very fast.

Our data are 38433 magnetic field measurement samples from 30 different magnetic sensors in three axes at first. Due to this phase computationally time-consuming, we selected 1 from every 4 sensors as shown in selected yellow colors in Fig. 7. Then to make the rich dataset explained in the proposed method, we combine 254 different states of fault occurrence possibility with healthy data which makes 9800415 samples for the input of our training phase. After taking training epochs explained above, we tested our algorithm to detect and compensate no-response fault with 9566 data sample of a curved movement appeared in Fig. 7 in blue color. In which the healthy sensors are in yellow color with two grid squares and to simulate no-response fault, we set all data of sensors 17, which is selected randomly, in each axis to zero.

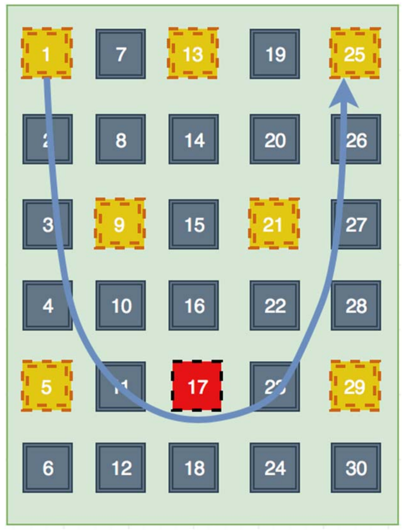
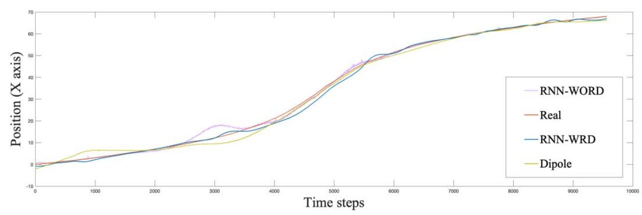
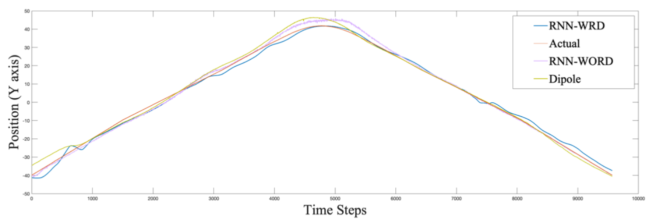


Fig. 7: Sensor board (Sensors in gray squares with simple double lines are not used, intact sensors are in yellow double grid squares and the sensor in the red grid around square is the faulty sensor that has been chosen randomly.)

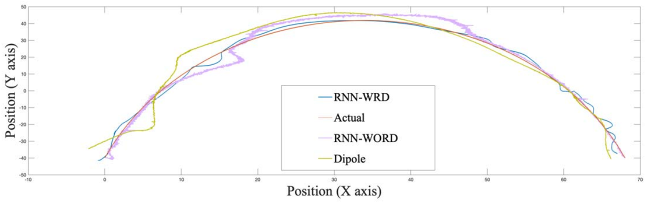
The results in Table I express that the RNN method is more accurate than the DM algorithm. Particularly; RNN-WRD (With Rich Dataset), the network with a rich training dataset, is 444.3% more accurate than the DM method and also it is 105.4% more robust than RNN-WORD (Without Rich Dataset), an RNN with simple training data. Moreover, the results shown in Fig. 8 confirms our expectation of the proposed algorithm that RNN-WRD have a more accurate result than the DM method. The real position of the magnet is presented in red color, the first network RNN-WRD in blue color expresses the best result among others as presented in Fig. 8. Also, the result of RNN-WORD in purple shows more accuracy to the DM method in green. In this DM method, the data of selected sensors are used only.



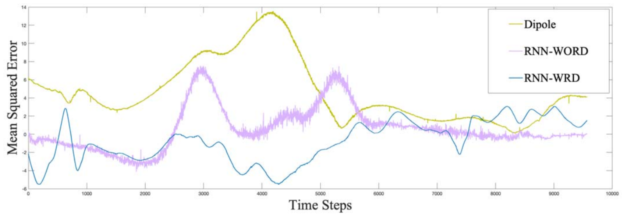
(a)



(b)



(c)



(d)

Fig. 8: Localization results after fault occurrence compared with the real location of the magnet a) X position estimation, b) Y position estimation, c) curve movement estimation, d) MSE in time steps of 42 steps per second frequency

V. CONCLUSION

In this paper, a robust magnetic localization system to sensor faults based on an RNN is proposed. This approach uses a heuristic to detect three different faults: noise, no-response and saturation fault. To make robustness, we train our network with a combination of 255 different data sets, one of which is the real data set of 8 sensors and in the rest 254 datasets there is a possibility of fault occurrence among eight sensors. Due to the RNNs complicated network and their use of each layer's output to make more accurate result, we select them for this localization. We compare the results of this method to an RNN which is not trained with the rich dataset and also with the output of the DM method in which our method shows the best result among all. The accuracy of RNN-WRD is 444.3% better than DM method and the robustness of this network is 105.4% better than an RNN-WORD.

For future works …

Although this method is accurate and robust to the appearance of a fault, it is responding to the test data which are taken from the plane of training data. In other words, to have the same result in different planes from the training set, we need to either collect a large data set for training phase or suggest a robust embedded ANN in an algorithm to the variation of the plane of interest.

Table I: Final MSEs and maximum errors of DM and RNN. The best results are shown in bold.

Table Head	Error		
	MSE	***MaxError in Amplitude***	***MaxError Components***
DM	21.6625	14.856	12.9652
RNN-WORD	5.1382	6.0967	5.9838
RNN-WRD	**4.8746**	**4.7091**	**4.6283**

VI. ACKNOWLEDGMENT

This project is supported by the National Institute for Medical Research Development (NIMAD), under grant number 962297. The authors would like to thank Dr. Ali Moradi for his contribution.

VII. REFERENCES

[1] Z. Sun, L. Maréchal, S. J. C. m. Foong, and p. i. biomedicine, "Passive magnetic-based localization for precise untethered medical instrument tracking," vol. 156, pp. 151-161, 2018.

[2] S. Tarantino, F. Clemente, D. Barone, M. Controzzi, and C. J. S. r. Cipriani, "The myokinetic control interface: tracking implanted magnets as a means for prosthetic control," vol. 7, no. 1, p. 17149, 2017.

[3] B. J. Radford, L. M. Apolonio, A. J. Trias, and J. A. J. a. p. a. Simpson, "Network traffic anomaly detection using recurrent neural networks," 2018.

[4] H. Al Hajj, M. Lamard, P.-H. Conze, B. Cochener, and G. Quellec, "Monitoring tool usage in surgery videos using boosted convolutional and recurrent neural networks," *Medical image analysis,* vol. 47, pp. 203-218, 2018.

[5] Z. Sun, S. Foong, L. Maréchal, T. H. Teo, U.-X. Tan, and A. Shabbir, "Real-time sensor fault detection and compensation in a passive magnetic field-based localization system," in *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, 2016, pp. 1040-1046: IEEE.

[6] S. I. Roumeliotis, G. S. Sukhatme, and G. A. Bekey, "Sensor fault detection and identification in a mobile robot," in *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No. 98CH36190)*, 1998, vol. 3, pp. 1383-1388: IEEE.

[7] N. Mehranbod, M. Soroush, and C. J. J. o. P. C. Panjapornpon, "A method of sensor fault detection and identification," vol. 15, no. 3, pp. 321-339, 2005.

[8] S. Postalcioglu, K. Erkan, and E. D. Bolat, "Intelligent sensor fault detection and identification for temperature control," in *Proceedings of the 11th WSEAS international conference on COMPUTERS*, 2007, pp. 131-134: World Scientific and Engineering Academy and Society (WSEAS).

[9] J. Zarei and E. J. M. Shokri, "Robust sensor fault detection based on nonlinear unknown input observer," vol. 48, pp. 355-367, 2014.

[10] L. Wu and D. W. J. I. T. o. F. S. Ho, "Fuzzy filter design for Itô stochastic systems with application to sensor fault detection," vol. 17, no. 1, pp. 233-242, 2008.