Enhancement of INS Accuracy Using Artificial Neural Networks For Trajectory Tracking

Mona Belaid
Aeronautical Engineering Departmement
University of Tripoli
Tripoli, Libya
m.belaid@aerodept.edu.ly

Othman Maklouf
Aeronautical Engineering Department
University of Tripoli
Tripoli, Libya
o.maklouf@aerodept.edu.ly

Saleh Basha
Aeronautical Engineering Departmement
University of Tripoli
Tripoli, Libya
amsb97@yahoo.com

***Abstract*— Inertial navigation system (INS) is a common choice for the purpose of position determination and trajectory tracking. However, standalone INS suffering from the problem of error accumulation with the time. This paper represents a methodology of enhancing of INS using Artificial Neural Networks (ANN). This will include the method of designing a navigation algorithm based on ANN in order to increase the accuracy and reliability of INS for trajectory tracking and fault detection. INS algorithm data is used to create an ANN algorithm. Several ANN structures are generated using neural network toolbox in MATLAB. The two algorithms are modelled and then a simulation study is carried out using MATLAB/Simulink tool box. Finally, the results of the INS algorithm and the ANN algorithm are compared and discussed.**

Keywords—INS; ANN; DR; Trajectory Tracking.

I. INTRODUCTION

INS is the implementation of inertial sensors to determine the position, velocity and attitude of aircraft, UAV, ship, submarines, cars, robots etc. INS is a self-contained navigation system because there is no information needed outside the system. The principle of dead Reckoning (DR) technique is used by INS to provide information about the required navigation states. In DR the current position of the vehicle is determined by the information about the previous position and inertial sensors measurements. The navigation state errors expand over time and are unlimited unless they are restrained by aid navigation systems, due to the constant accumulation of sensor errors caused by this DR technique. All self-contained navigation systems have this characteristic, which is a serious disadvantage. INS typically comprises an Inertial Measurement Unit (IMU), which is made up of three accelerometers and three rate gyros, to track a moving vehicle in three dimensions. The accelerometers are used to measure the acceleration along the axes of motion, and the gyro to measure the rotation attitude angles (pitch, roll and yaw).

There are two main categories of INS: stabilized platform system and strapdown system. In a stabilized system, the platform is mounted using gimbals, allowing the platform freedom in all three axes, to keep the IMU's axes aligned with the navigation frame. In strap down systems the IMU is mounted rigidly onto the moving object, IMU output measurements are thus made in the body frame rather than the navigation frame. To estimate the velocity and position states, it is therefore necessary to transform the acceleration measurements from the body frame to the navigation frame before performing the double integration. The strapdown inertial navigation algorithm is illustrated in Fig. 1. Recently, The INS has been widely used in many fields such as the intelligent systems due to its capability to provide self-contained and high rate of the output motion information. Aartificial intelligence (AI) has become a powerful tool that has been successfully applied in many applications. One of the components of AI is an ANN that is meant to simulate the functioning of a human brain. It has a self-learning capability that enable them to produce better results as more data becomes available. It can provide an analytical alternative to conventional techniques [1].

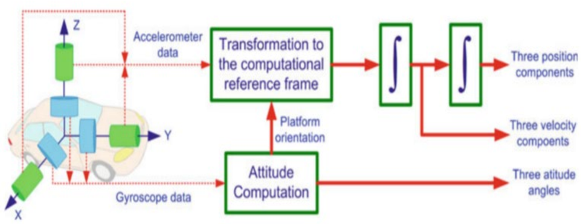

Fig. 1. *Strapdown inertial navigation algorithm.*

II. ANN:AN OVERVIEW

ANN plays a significant tools in solving different complicated problems in different areas. It was resemble the human brain in order to solve such problems. ANN is made up of a number of interconnected processing units called neurons. Input, hidden, and output layers are among the layers in which neurons are arranged. ANN learns the relation between inputs and outputs of the system through an iterative process called training [3].

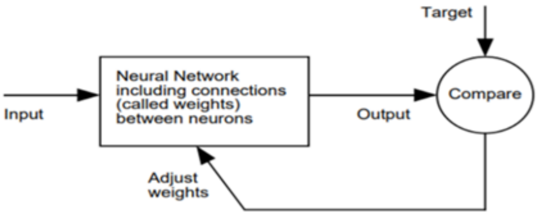

Fig. 2. ANN-Training Structure

Training of ANN can be adjusted based upon certain criteria Fig.2 shows the idea of supervised learning [4]. For an ANN model to be accurate and reliable, the input and output parameters must be carefully chosen.

A. *Topology of Artificial Neural Network*

The topology or architecture of ANN determines the results to be obtained. The simplest architecture of a neural network can be described as given below:

1) *Artificial Neuron*

An artificial neuron tries to replicate the structure and behavior of the natural neuron.

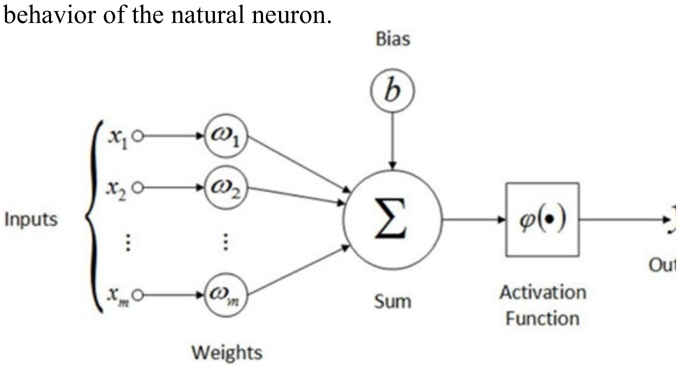

Fig. 3. *simple model of an artificial neuron*

A modified simple model of an artificial neuron is shown in Fig.3 with inputs $(x1, x2, \dots xi)$ being connected to neuron with weights $(w1, w2, \dots wi)$ on each connection. The output (y) from the neuron is computed as given in "(1)". The neuron sums all the signals it receives, with each signal being multiplied by its associated weights on the connection, A bias is also added, every activation function (φ) takes a single number and performs a certain fixed mathematical operation on it [5].

$$y = \varphi(\textstyle\sum_{i=1}^{n}(xi)(wi) + b) \qquad (1)$$

2) *Weight*

The importance of the input value is reduced by a weight. Weights close to zero indicate that changing this input will not alter the result. The degree to which the input will affect the output is determined by a weight [6].

3) *Activation Process (Transfer Function)*

It used to introduce neural network nonlinearity. The Neural Network Tool-Box software contains a number of transfer functions. Tan-Sigmoid and Log-Sigmoid are the two commonly used transfer functions for multi-layer neural networks [3].

B. *Neural Network Types*

ANNs can be categorized based on their structure, data flow, number of neurons used, layer density, etc. Neural network types based on data flow

1) *Forward Propagation*

It is the process of predicting the output value by feeding input values to ANN. This is called the forward propagation.

2) *Back Propagation*

In this type of propagation, the difference in targeted output, and the output obtained, is propagated back to the layers and adjusting the weights. It is one of the most frequently utilized neural network techniques for classification and prediction. As Fig.4 shows, the outputs of hidden layers are propagated to the output layer where the output is calculated. This output is compared with the desired output for the given input [1].

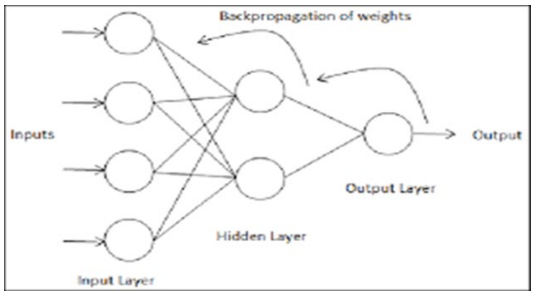

Fig. 4. *Backpropagation Algorithm*

III. INS Mechanization and Simulation

By resolving the differential equations describing the system motion, INS mechanization is the process of extracting the navigation states (position, velocity, and attitude) from the raw inertial measurements. The navigation frame is typically where mechanization is carried out. From selected raw data inertial measurements, a reference trajectory is produced using the INS algorithm using the MATLAB with Simulink block diagram shown in Fig. 5. Fig. 6 displays the resulting reference trajectory.

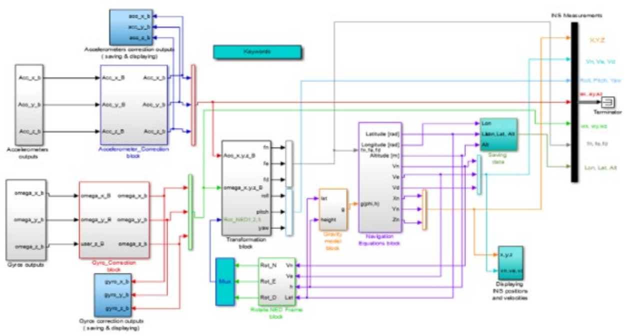

Fig. 5. *The Simulink block diagram.*

Fig. 6. *INS Simulink model*

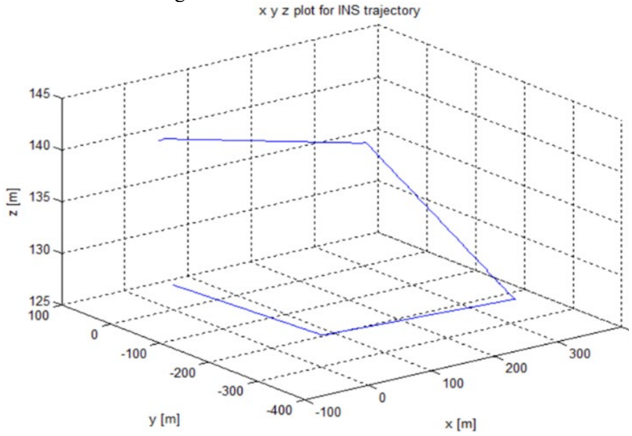

Fig. 7. *3D Reference INS Trajectory*

IV. IMPLEMATATION OF ANN

The use of ANN is described in this section with the goal of assisting the INS in trajectory tracking. There will be data generation and ANN training.

A. Data Generation

The required data for the ANN modelling were generated using the INS Simulink block diagram given in previous section. The simulation time for all cases is 18 seconds. In this work the data obtained from INS sub blocks inputs and outputs (targets) which about 1800 samples, almost 80% from the data will be employed here for creating and training an accurate ANN model.

B. Creating and trainig process

The training process of the ANN is employed here to obtain an accurate network structure and to ensure a good generalization characteristic of the INS algorithm. Neural network MATLAB toolbox (nntool) is used to create, view, and train the network and export the final results to the workspace. In this work five neural networks are designed with different types of neural networks, number of neurons and activation functions. One of these ANN is assigned for NED Frame Sub Block shown in Fig.7. In nntool, the data set used for creating network is divide in to training, testing and validation. Table I and Table II shows the inputs and outputs (targets) of the model.

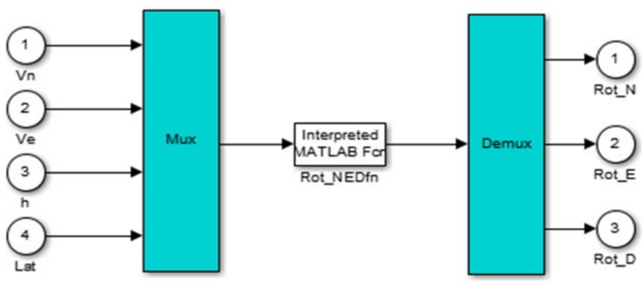

Fig. 8. *Detailed Rotate NED Frame block model*

Figure 8 shows that there are four input variables, one hidden layer, eight neurons, and three output variables. There are also four hidden layers and one hidden layer. The output layer neurons' nonlinear activation is determined by the "Pure line" activation function, while the hidden layer neurons' nonlinear activation is determined by the "Tan-Sigmoid" function, one of the most popular activation functions. Finding appropriate training and architectural parameters for an ANN remains a challenging task, even after assigning the data sets required for the network. Usually, these parameters are chosen through a process of trial and error. In order to find the best values for these parameters and achieve the best results for the models to train better and faster, many ANN models are developed and compared to one another. In fig. 9, these training parameters are displayed.

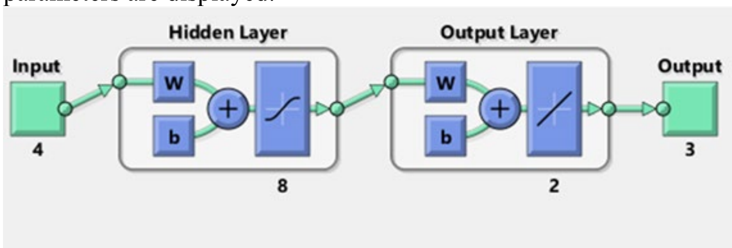

Fig. 9. *NN structure*

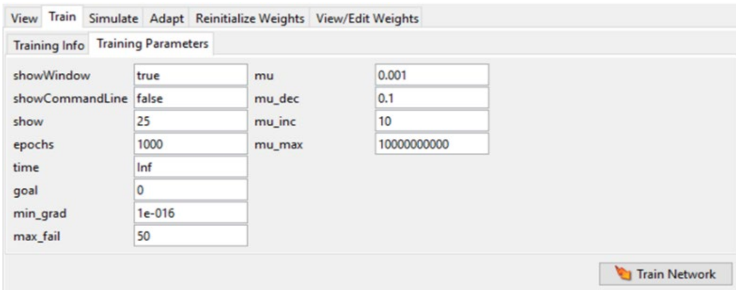

Fig. 10. *NN training parameters*

TABLE I NED Frame model input parameters for the ANN

N.o	***Vn[m/sec]***	***Ve [m/sec]***	***h[m]***	***Lat [rad]***
1	**0**	**0**	**127.4**	**0.79813906693700**
2	**-0.0001866**	**-0.0002009**	**127.4000**	**0.79813906693687**
. . .				
1500	**-204.8661**	**3.637980**	**115.9370**	**0.79816379391343**

TABLE II NED Frame output parameters for the ANN model

N.o	**ROT-N [rad/sec]**	**ROT-E [rad/sec]**	**ROT-D [rad/sec]**
1	**5.0901912642e-05**	**0**	**-5.22157965238e-05**
2	**5.0901881085e-05**	**0**	**-5.22157641519e-05**
. . .			
1500	**5.1471876843e-05**	**0**	**-5.280308478e-05**

Once sitting the parameters is done the training window appears as shown in Fig. 10 in which the data is divided using the "dividerand" function and "trainlm" training method were used. It also shows the epochs or iteration, the total time for training and the mean square error performance function.
After recording the training results, the ANN's performance is assessed by comparing the terms of their mean squared errors (MSE). The training, validation, and testing performance plots of the ANN are displayed in Fig. 11. According to the plots shown, it is evident that the training MSE was only about 2.629e-16, which is incredibly low. A lower MSE indicates a high level of ANN accuracy. At the iteration where the validation performance error grew to 637, the MSE was attained.

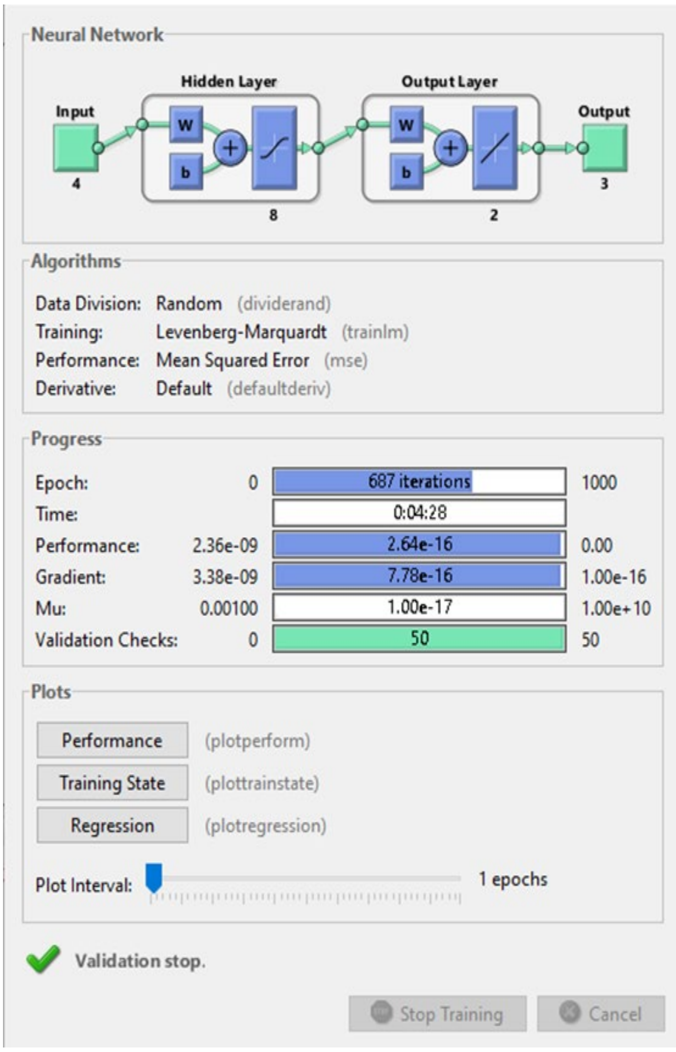

Fig. 11. *NN training windows*

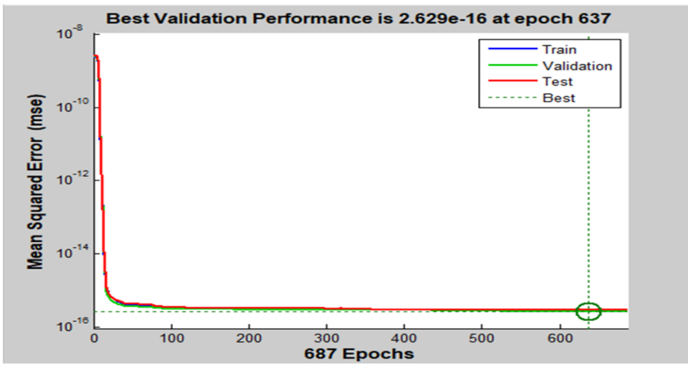

Fig. 12. *NN performance Plots*

Figure 12 depicts the connection between the targets' outputs and the ANN's outputs. The relationship between the outputs and the targets is shown by the R value. All of the graphs' R values are equal to 1. The results show a very good fit for each of the training, validation, and test data sets.

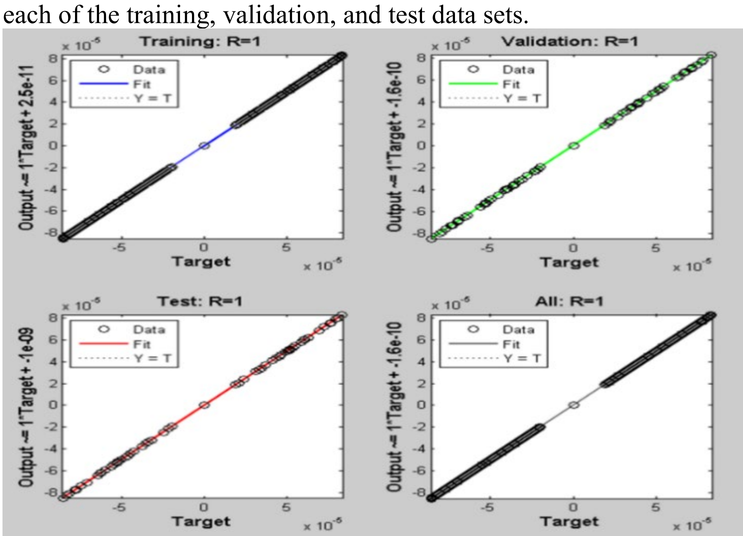

Fig. 13. *NN Regression Plot*

As seen in Fig.13, the chosen ANN is implemented using a Simulink model. With different numbers of neurons in the hidden layers and parameter values, the 4 additional neural networks are trained using feed forward BP and cascade neural networks.

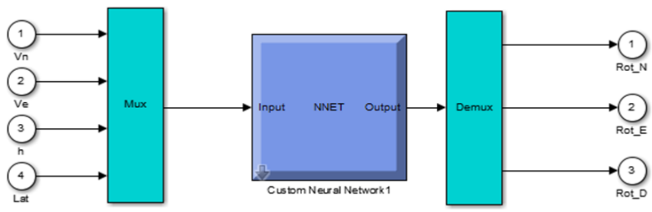

Fig. 14. *Rotate NED frame ANN block SIMULINK model*

C. *Testing Networks*

This section generates the identical reference INS trajectory shown in Fig. 6 by simulating the ANN algorithm; the resulting ANN trajectory is shown in Fig. 14.

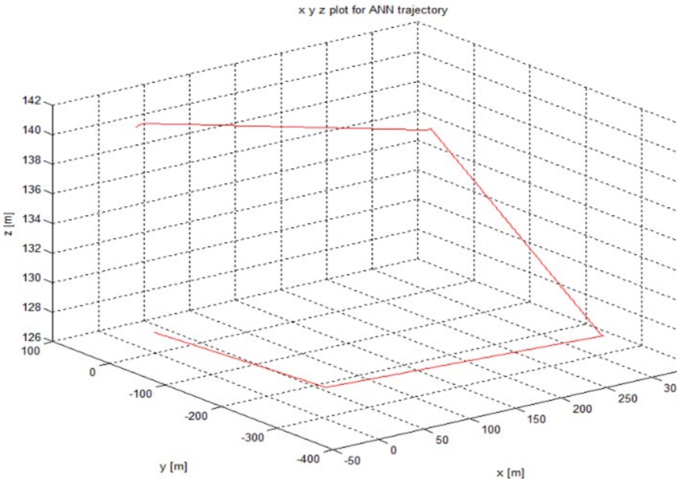

Fig. 15. *Trajectory of the ANN Navigation Model*

It should be noted that only 300 of the 1800 output data sets were not used in the training process. A compression study between the outputs of the ANN sub - blocks and the corresponding outputs of INS sub - blocks is presented in Figs. 15 through 17. The resulting neural network-based model can accurately predict how the system will respond to changes in input parameters.

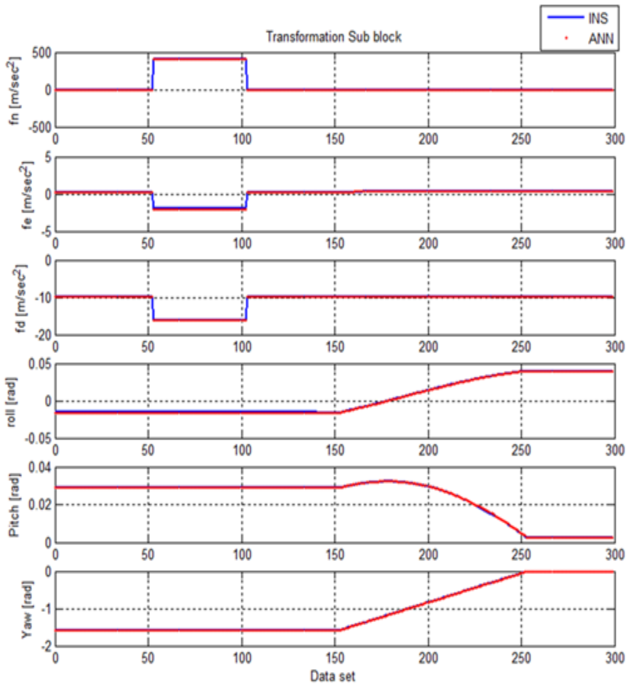

Fig. 16. *Comparison Between Outputs of The INS & ANN Transformation Sub Block*

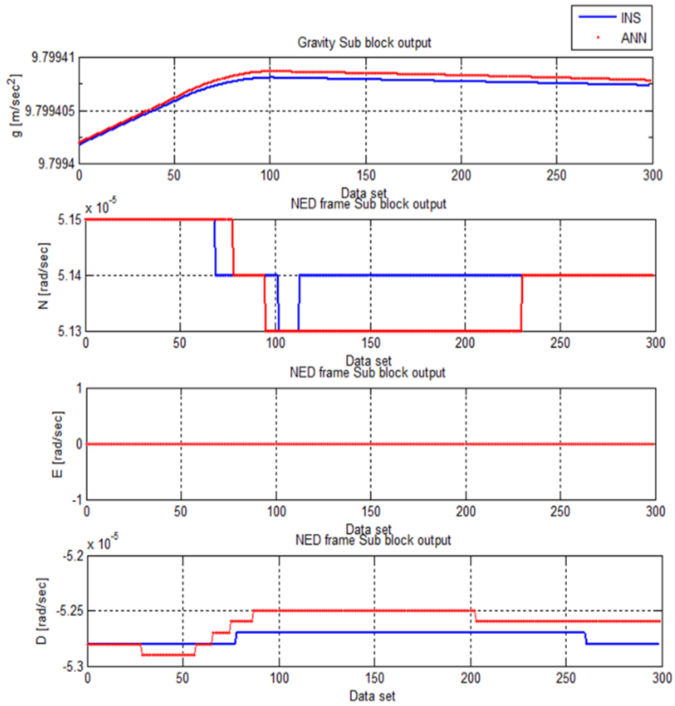

Fig. 17. *Comparison Between Outputs of The INS and ANN Gravity and NED Frame Sub Blocks*

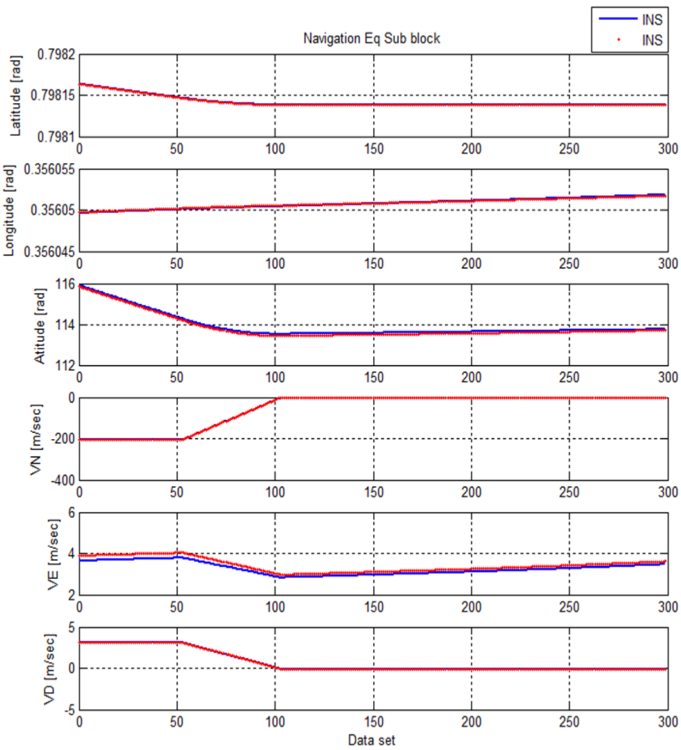

Fig. 18. *Comparison Between Outputs of The INS and ANN Navigation Eq Sub Block*

The INS trajectory presented in Fig.8 is used as a reference to make a comparison between the INS navigation algorithm model and the ANN model. As it can be seen through Fig.18 the ANN model has small drift compared to navigation algorithm model, these errors are due the fact that all the data are not included in the training process.

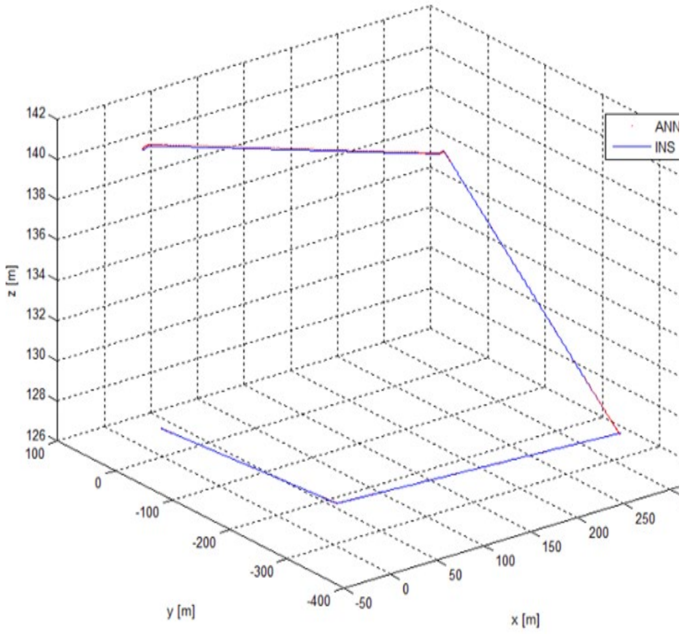

Fig. 19. *Comparison between ANN trajectory and INS trajectory*

V. CONCLUSIONS

This paper described the use of AI technology to improve the INS's trajectory tracking capabilities. This was accomplished by using the designed ANN in place of some of the INS algorithm's mathematical models. The results obtained demonstrate the ANN's capacity to generate a trajectory with a higher definition resolution than a standalone INS trajectory. The tested suggested ANN model was successfully verified, and the case study results were within acceptable bounds. The created model could be applied to navigation system fault detection, sensor validation, condition monitoring, and optimization. Traditional algorithms are largely derived from rigorous but occasionally challenging theoretical mathematical models. The use of AI can be implemented once there is enough and trustworthy training data.

VI. REFERENCESS

[1] Introduction to Neural Networks for Java (second edition). Jeff Heaton ,(published date:2005).

[2] Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration. Noureldin, A., Karamat, T. B., & Georgy, J. (2013). (last access date:1/2021)

[3] Modelling, Simulation and Control of Gas Turbines Using Artificial Neural Networks, Hamid Asgari. (last access date:4/2020)

[4] https://en.wikipedia.org/wiki/Artificial_neural_network. (last access date:4/2020)

[5] An Introduction to Artificial Neural Network. Harsh Kukreja, Bharath, Siddesh, Kuldeep. (last access date:10/2020)

[6] https://hackernoon.com/everything-you-need-to-know-about-neural-networks ,Kailash Ahirwar, (last access date:4/2020)