International Journal of Engineering Science Invention
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org ||Volume 4 Issue 6|| June 2015 || PP.62-71

Prediction and Estimation of Gyroscopic Couple by Analytical and Neural Network

[1,] Dilip Kumar Sonar, [2,] Subham SenGupta
[1,] *Asst.Prof. of Mechanical Engg.*
[1, 2,] *Dept of College of Engineering & Management Kolaghat. KTPP Township*

ABSTRACT: *Newton carried out studies on the motion of earth which rotates about its axis and at the same time it revolves around the sun. Those studies were further carried out by Euler and he laid down the mathematical foundation of the principals of gyroscopic motion. To understand gyroscopic effect, a body can be considered which rotates about its axis of symmetry. It will be found to offer a resistance to any change in direction of the axis of rotation. This is gyroscopic effect. A gyroscope is an instrument which consists up a heavy spinning rotor. One popular example is the spinning top. The use of gyroscopes was earlier limited to stabilization of seaborne ships and for gyroscopic compasses. After world war 2 it found application in bomb sights, aero planes , guided missiles etc. the gyroscopic effect generates forces and couples which act on vehicles and other means of transport like ships , airplanes etc. Neural network is used to minimize error or for predictions of those measures which are not possible to calibrate. Mainly the back propagation algorithm is used to minimize the error that causes the deviation in measurements or in calibration.*

I. INTRODUCTION

A Gyroscope is a device for measuring or maintaining orientation based on the principles of angular momentum. Mechanically, a gyroscope is a spinning wheel or disc in which the axle is free to assume any orientation. Although this orientation does not remain fixed, it changes in response to an external torque much less and in a different direction than it would without the large angular momentum associated with the disc's high rate of spin and moment of inertia. The device's orientation remains nearly fixed, regardless of the mounting platform's motion, because mounting the device in a gimbals minimizes external torque.
Gyroscopes based of other operating principles also exist, such as the electronic, microchip-packaged MEMS gyroscope devices found in consumer electronic devices, solid state ring lasers, fiber optic gyroscopes, and the extremely sensitive quantum gyroscope. Applications of gyroscope include inertial navigation systems where magnetic compasses would not work(as in the Hubble telescope) or would not be precise enough or for the stabilization of flying vehicles like radio controlled helicopters or unmanned aerial vehicles. Due to their precision gyroscopes are also used in gyrotheodolites to maintain direction in tunnel mining.

One type of network sees the nodes as 'artificial neurons'. These are called artificial neural networks (ANNs). An artificial neuron is a computational model inspired in the natural neurons. Natural neurons receive signals through *synapses* located on the dendrites or membrane of the neuron. When the signals received are strong enough (surpass a certain *threshold*), the neuron is *activated* and emits a signal though the *axon*. This signal might be sent to another synapse, and might activate other neurons.

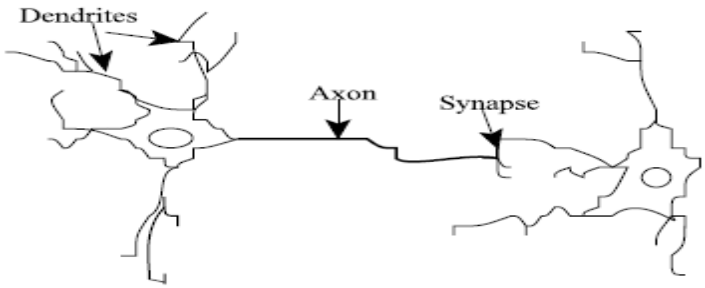


Figure 1. Natural Neurons (artist's conception).

The complexity of real neurons is highly abstracted when modelling artificial neurons. These basically

consist of *inputs* (like synapses), which are multiplied by *weights* (strength of the respective signals), and then computed by a mathematical function which determines the *activation* of the neuron. Another function (which may be the identity) computes the *output* of the artificial neuron (sometimes in dependance of a certain *threshold*). ANNs combine artificial neurons in order to process information.

Figure 2. An Artificial Neuron

The higher a weight of an artificial neuron is, the stronger the input which is multiplied by it will be. Weights can also be negative, so we can say that the signal is *inhibited* by the negative weight. Depending on the weights, the computation of the neuron will be different. By adjusting the weights of an artificial neuron we can obtain the output we want for specific inputs. But when we have an ANN of hundreds or thousands of neurons, it would be quite complicated to find by hand all the necessary weights. But we can find algorithms which can adjust the weights of the ANN in order to obtain the desired output from the network. This process of adjusting the weights is called *learning* or *training*. The number of types of ANNs and their uses is very high. Since the first neural model by McCulloch and Pitts (1943) there have been developed hundreds of different models considered as ANNs. The differences in them might be the functions, the accepted values, the topology, the learning algorithms, etc. Also there are many hybrid models where each neuron has more properties than the ones we are reviewing here. Because of matters of space, we will present only an ANN which learns using the back propagation algorithm (Rumelhart and McClelland, 1986) for learning the appropriate weights, since it is one of the most common models used in ANNs, and many others are based on it. Since the function of ANNs is to process information, they are used mainly in fields related with it. There are a wide variety of ANNs that are used to model real neural networks, and study behavior and control in animals and machines, but also there are ANNs who are used for engineering purposes, such as pattern recognition, forecasting, and data compression.

Literature Survey:

There are lots of literatures published before on the same task; systems, some inspired by biological neural networks. Researchers from many scientific disciplines are designing artificial neural networks (An"s) to solve a variety of problems in pattern recognition, prediction, optimization, associative memory, and control (see the "Challenging problems" sidebar). Conventional approaches have been proposed for solving these problems.

Although successful applications can be found in certain well-constrained environments, none is flexible enough to perform well outside its domain. ANNs provide exciting alternatives, and many applications could benefit from using them.' This article is for those readers with little or no knowledge of ANNs to help them understand the other articles in this issue of *Computer.* We discuss the motivations behind the development of *A " s ,* describe the basic biological neuron and the artificial computational model, outline network architectures and learning processes, and present some of the most commonly used ANN models. We conclude with character recognition, a successful ANN application.

A first wave of interest in neural networks also known as connectionist models or parallel distributed processing emerged after the introduction of simplified neurons by **McCulloch and Pitts in McCulloch Pitts [1]**. Neurons were presented as models of biological neurons and as conceptual components for circuits that could perform computational tasks.

When **Minsky and Papert published Perceptrons in [2]** in which they showed the deficiencies of perceptron models. Most neural network funding was redirected and researchers left the field, only a few researchers continued their efforts. Most Notably Teuvo Kohonen, Stephen Grossberg_,James Anderson and Kunihiko Fukushima. The interest in neural networks re-emerged only after some important theoretical results were attained in the early eighties. Most notably the discovery of error back propagation and new hardware developments increased the processing capacities. This renewed interest is reflected in the number of scientists. The amounts of funding the number of large conferences and the number of journals associated with neural networks. Nowadays most universities have a neural networks group within their psychology, physics, and

computer science or biology department's artificial neural networks can be most adequately characterized as "computational models" with particular properties such as the ability to adapt or learn to generalize or to cluster or organize data and which operation is based on parallel processing. However, many of the above mentioned properties can be attributed to existing non neural models. The intriguing question is to which extent the neural approach proves to be better suited for certain applications than existing models. To date an equivocal answer to this question is not found. Often parallels with biological systems are described. However, there is still so little known even at the lowest cell level about biological systems that the models we are using for our artificial neural systems seem to introduce an oversimplification of the biological models

Neural Network Concept:

BIOLOGICAL INSPIRATION

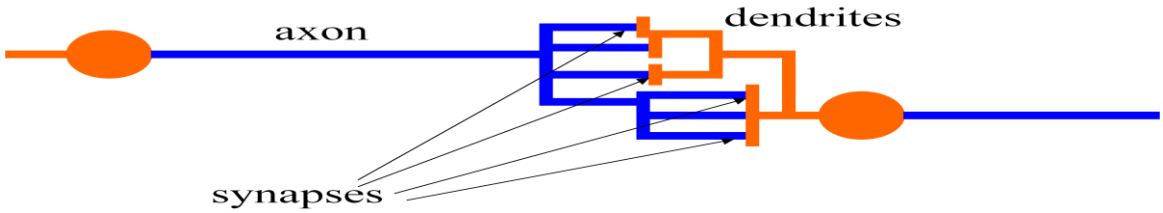


Figure 3. Biological Neurons

The spikes travelling along the axon of the pre-synaptic neuron trigger the release of neurotransmitter substances at the synapse. The neurotransmitters cause excitation or inhibition in the dendrite of the post-synaptic neuron. The integration of the excitatory and inhibitory signals may produce spikes in the post-synaptic neuron. The contribution of the signals depends on the strength of the synaptic connection. Artificial neural network Neurons work by processing information. They receive and provide information in form of spikes

- Artificial neural network (ANN) is a machine learning approach that models human brain and consists of a number of artificial neurons.
- Neuron in ANNs tend to have fewer connections than biological neurons.
- Each neuron in ANN receives a number of inputs.
- An activation function is applied to these inputs which results in activation level of neuron (output value of the neuron).
- Knowledge about the learning task is given in the form of examples called training examples.
- An Artificial Neural Network is specified by:
  neuron model: the information processing unit of the NN,
  Architecture: a set of neurons and links connecting neurons. Each link has a weight,
  A learning algorithm: used for training the NN by modifying the weights in order to model a particular learning task correctly on the training examples.
- The aim is to obtain a NN that is trained and generalizes well.
- It should behave correctly on new instances of the learning task.
- The neuron is the basic information processing unit of a NN. It consists of:

1 A set of links, describing the neuron inputs, with weights $W_1, W_2, \ldots, W_m$

2 An adder function (linear combiner) for computing the weighted sum of the inputs: (Real numbers) $u = \sum_{j=1}^{m} w_j x_j$

3 Activation functions for limiting the amplitude of the neuron output. Here 'b' denotes bias.

$$y = \varphi(u + b)$$

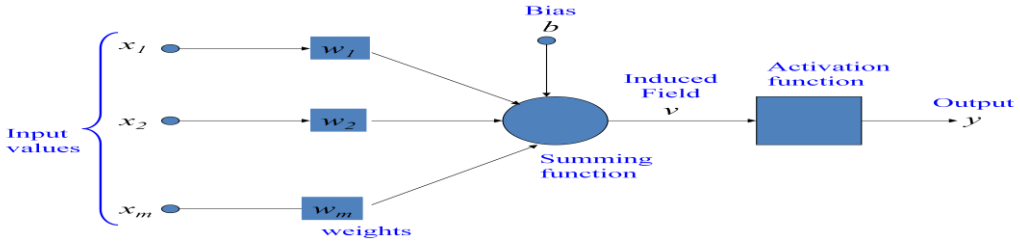


Figure 4. Neuron Diagram

BIAS OF A NEURON:

- The bias ***b*** has the effect of applying a transformation to the weighted sum ***u***

$$v = u + b$$

- The bias is an external parameter of the neuron. It can be modeled by adding an extra input.
- ***v*** is called **induced field** of the neuron

$$v = \sum_{j=0}^{m} w_j x_j$$

$$w_0 = b$$

neuron model

- The choice of activation function determines the neuron model.

Examples:

- step function:

$$\varphi(v) = \begin{cases} a & \text{if } v < c \\ b & \text{if } v > c \end{cases}$$

- ramp function:

$$\varphi(v) = \begin{cases} a & \text{if } v < c \\ b & \text{if } v > d \\ a + ((v - c)(b - a)/(d - c)) & \text{otherwise} \end{cases}$$

-

- sigmoid function with z,x,y parameters

$$\varphi(v) = z + \frac{1}{1 + \exp(-xv + y)}$$

- Gaussian function:

$$\varphi(v) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\left(\frac{v - \mu}{\sigma}\right)^2\right)$$

Network architecture

- Three different classes of network architectures - single-layer feed-forward ,multi-layer feed-forward ,recurrent and The architecture of a neural network is linked with the learning algorithm used to train

Single layer feed forward

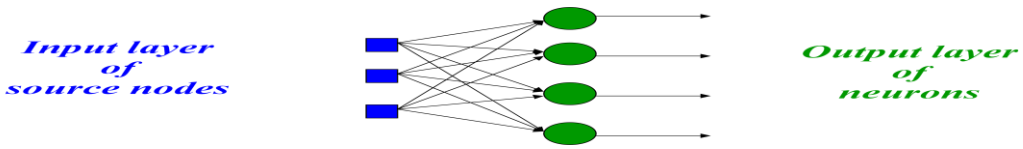


Figure 5. Single Layer Feedforward

Perceptron : neuron model
(a special form of single layer feed forward network)

- The perceptron was first proposed by Rosenblatt (1958) is a simple neuron that is used to classify its input into one of two categories.
- A perceptron uses a **step function** that returns +1 if weighted sum of its input ≥ 0 and -1 otherwise

$$\varphi(v) = \begin{cases} +1 \text{ if } v \geq 0 \\ -1 \text{ if } v < 0 \end{cases}$$

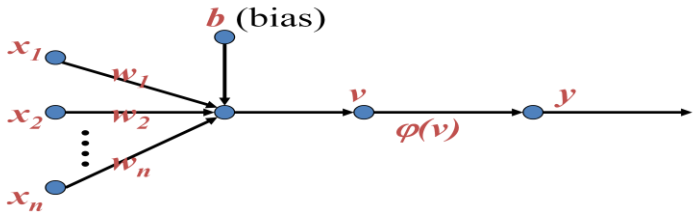


Figure 6. A Perceptron Model

Multilayer feedforward neural network (FFNN)

- FFNN is a more general network architecture, where there are hidden layers between input and output layers.
- Hidden nodes do not directly receive inputs nor send outputs to the external environment.
- FFNNs overcome the limitation of single-layer NN.

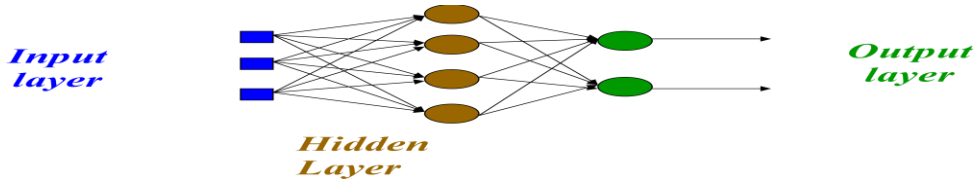


Figure 7. 3-4-2 Network

Multiyaler feedforward neural network model

- The classical learning algorithm of FFNN is based on the gradient descent method.
- For this reason the activation function used in FFNN are continuous functions of the weights, differentiable everywhere.
- The activation function for node i may be defined as a simple form of the **sigmoid function** in the following manner:

where $A > 0$, $V_i = \sum W_{ij} * Y_j$, such that W_{ij} is a weight of the link from node i to node j and Y_j is the output of node j.

Training algorithm back propagation method

- The Backpropagation algorithm learns in the same way as single perceptron.
- It searches for weight values that minimize the total error of the network over the set of training examples (training set).
- Backpropagation consists of the repeated application of the following two passes:
  - **Forward pass**: In this step, the network is activated on one example and the error of (each neuron of) the output layer is computed.
  - **Backward pass**: in this step the network error is used for updating the weights. The error is propagated backwards from the output layer through the network layer by layer. This is done by recursively computing the local gradient of each neuron.
  -

BACK PROPAGATION ALGORITHM

Back-propagation training algorithm

Figure 8. Back Propagation Training Algorithm

- Backpropagation adjusts the weights of the NN in order to minimize the network total mean squared error.
- Consider a network of three layers.
- Let us use i to represent nodes in input layer, j to represent nodes in hidden layer and k represent nodes in output layer.
- w_{ij} refers to weight of connection between a node in input layer and node in hidden layer.
- The following equation is used to derive the output value Yj of node j

$$Yj = \frac{1}{1+e^{-X_j}}$$

where, $X_j = \sum x_i \, . \, w_{ij} - \theta_j$, $1 \leq i \leq n$; n is the number of inputs to node j, and θ_j is threshold for node j

total mean squared error

- The error of output neuron *k* after the activation of the network on the *n-th* training example (x(n), d(n)) is:

$e_k(n) = d_k(n) - y_k(n)$

- The network error is the sum of the squared errors of the output neurons:

$$E(n) = \sum e_k^2(n)$$

- The total mean squared error is the average of the network errors of the training examples.

$$E_{AV} = \frac{1}{N} \sum_{n=1}^{N} E(n)$$

Figure 16. A Multilayer Network

An artificial neural network is composed of many artificial neurons that are linked together according to specific network architecture. The objective of the neural network is to transform the inputs into meaningful outputs.

EXPERIMENTAL ANALYSIS: Specifications of the gyroscope under observation:

MOTORISED GYROSCOPE -TECHNICAL TEACHING (D) EQUIPMENT -VOLTAGE = 230V CURRENT CAPACITY = 1.5A ,MAXIMUM SPEED = 8000 RPM, S.No : BBO 12B60 , Motor manufacturer: UP National MFRS. Ltd, Varanasi (India) INS: CLASS H.P./WATT 1/10/75 Type: AC/DC

The following variables have been taken under consideration:
m = mass of the rotating disc in kg
m_1 = mass of the weight in kg
r = distance of the weight (from its C.G.) to the centre of rotation of the disc in m

N_i = initial angular speed of the disc in rpm
N_f = final angular speed of the disc in rpm
N_{avg} = average angular speed of rotation

$$N_{avg} = \frac{N_i + N_f}{2}$$

ω = angular speed of rotation of the speed in radian

$$\omega = \frac{2\pi N_{avg}}{60}$$

θ_i= initial angle at which the pointer of the gyroscope lies
θ_f=final angle at which the pointer of the gyroscope lies
θ_r=actual angular displacement of the disc on the plane of the installation

$$\theta_r = \theta_f - \theta_i$$

t = time taken for the angular displacement
ω_p=angular velocity of precession

$$\omega_p = \frac{\theta_f - \theta_i}{t} = \frac{\theta_r}{t}$$

F=force acting due to the weight through its centre of gravity

$$F = m_1 \times g$$

Where $g = 9.81 m/s^2$ acceleration due to gravity
I = moment of inertia of the rotating disk

$$I = \frac{mr^2}{2}$$

$C_{experimental}$= experimental gyroscopic couple/torque

$$C_{experimental} = I\omega\omega_P$$

$C_{theoritical}$=theoretical gyroscopic couple/torque

$$C_{theoritical} = F \times r$$

RESULT AND DISCUSSION

The experiments are carried in MOTORISED GYROSCOPE, VOLTAGE = 230V, CURRENT CAPACITY = 1.5A and MAXIMUM SPEED = 8000 RPM and results are generated which are given in Tabular form below .Table -1 the experimental value of angle of precision with angular velocity of the disc along with mass of load applied on the lever and gyroscopic couple experimental value

CALCULATIONS: Mass of disc = 3.5kg and radius of disc r =0.13m

$$I = \frac{mr^2}{2} = \frac{3.5 \times 0.13^2}{2} = 0.02986\ kg - m^2$$

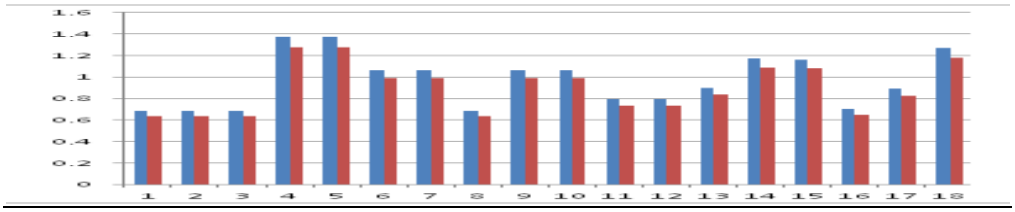
Exp. No.	mass(kg)	Ni(rpm)	Nf(rpm)	θi(deg)	θf(deg)	t(sec)	N_{avg}	W	Q_r	w_p	Np	$C_{experimental}$
1	0.5	2048	2050	25	95	13.52	2049	214.462	1.15133333	0.08490659	0.811209	0.543727807
2	0.5	2047	2049	33	94	13.84	2048	214.357333	1.06411111	0.07688664	0.734586	0.49212909
3	0.5	2053	2056	24	93	15.25	2054.5	215.037667	1.20366667	0.07892896	0.754098	0.506804815
4	1	2057	2058	23	178	20.03	2057.5	215.351667	2.70388889	0.13499196	1.289732	0.868052381
5	1	2070	2074	36	198	20.81	2072	216.869333	2.826	0.1358001	1.297453	0.879403167
6	0.775	2076	2096	33	143	17.6	2086	218.334667	1.91888889	0.10902778	1.041667	0.710803669
7	0.775	2096	2099	28	98	15.36	2097.5	219.538333	1.22111111	0.07949942	0.759549	0.52115167
8	0.5	2267	2274	24	95	15.69	2270.5	237.645667	1.23855556	0.07893917	0.754196	0.560160203
9	0.775	2240	2252	35	148	17.24	2246	235.081333	1.97122222	0.11434004	1.092421	0.802613155
10	0.775	2250	2238	37	150	17.98	2244	234.872	1.97122222	0.10963416	1.04746	0.768894844
11	0.577	2076	2093	40	114.5	14.51	2084.5	218.177667	1.29961111	0.08956658	0.855732	0.583507039
12	0.577	2213	2214	43	112	14.1	2213.5	231.679667	1.20366667	0.08536643	0.815603	0.59056111
13	0.654	2220	2217	29	121	16.85	2218.5	232.203	1.60488889	0.09524563	0.90999	0.660393355
14	0.852	2221	2223	24	165	19.12	2222	232.569333	2.45966667	0.12864365	1.229079	0.893368467
15	0.846	2220	2222	19.5	169	21.16	2221	232.464667	2.60794444	0.12324879	1.177536	0.855518544
16	0.571	2216	2215	24	114	17	2215.5	231.889	1.57	0.09235294	0.882353	0.639470747
17	0.648	2204	2207	24	126	17.94	2205.5	230.842333	1.77933333	0.09918246	0.947603	0.683659944
18	0.923	2203	2201	33	184	22.69	2202	230.476	2.63411111	0.11609128	1.109152	0.79894173

Table -2 showing the experimental value and theoretical gyroscopic couple value. It is found that error in three cases less than 33% and average error less than 22% .This variation of the error due to experimental setup and also device which calibration is not up to the experimental mark.

THEORITICAL CALCULATIONS: (Table No-2)

Exp. No.	**F**	$\mathbf{C_{theoritical}}$	$C_{experimental}$	% of error
1.	4.90500	0.6376500	0.543727807	14.72942725
2.	4.90500	0.6376500	0.492129090	22.82143971
3.	4.90500	0.6376500	0.506804815	20.51990668
4.	9.81000	1.2753000	0.868052381	31.93347600
5.	9.81000	1.2753000	0.879403167	31.04342770
6.	7.60275	0.9883575	0.714722290	27.68585356
7.	7.60275	0.9883575	0.560160203	12.15240283
8.	4.90500	0.6376500	0.802613155	18.79323474
9.	7.60275	0.9883575	0.768894844	22.20478486
10.	7.60275	0.9883575	0.583507039	20.70278647
11.	5.66037	0.7358481	0.590561110	19.74415508
12.	5.66037	0.7358481	0.660393355	20.82053064
13.	6.41574	0.8340462	0.893368467	17.77977431
14.	8.35812	1.0865556	0.855518544	20.70483542
15.	8.29926	1.0789038	0.639470747	1.680842965
16.	5.00310	0.6504030	0.683659944	17.27195348
17.	6.35688	0.8263944	0.798941730	32.12637494
18.	9.054630	1.1771019	0.897598279	24.23756269

Figure-23. showing the variation of the experimental and theoretical value of the gyroscopic couple value.



(Figure No -9.)

The error estimations further for performing by using back propagation (neural networking) methods. In the network the input data is divided into two groups termed as training data and testing data set. For this work the training data consisting of 20 inputs and testing data consist of 8 inputs. The training and testing data are set as input ω and ω_p. These data is normalized between 0 to 1 by using the input data itself divided by the maximum among them. Output data for training and testing are gyroscopic couple which is normalized by taking the logarithmic of output data divided by maximum output data. The net generated is newff and trained by logsig, purelin and trainlm transfer functions. The number of neurons in hidden layer -1 and hidden layer-2 are taken 1 and 3 . The transfer function is taken as logsig and trainlim to train the network .Once net is generated this net is generated on basis of training data and training function which is used to simulate to testing data then to find the root mean squared error of training and testing data separately and to predict the output . The Training and testing error are errtrain=0.065 and errtest=0.0578. Training Data are given in table -3 where Cexp and Cnn are experimental and predicted by neuaral network of the gyroscopic couple value and errors are estimated and found error are negligible,while Testing Data are given in table -4 where Cexp and Cnn are experimental and predicted by neuaral network of the gyroscopic couple value and errors are estimated and found error are less than 10%. Are given below

Cexp	Cnn	error
0.5437	0.5407	0.551775

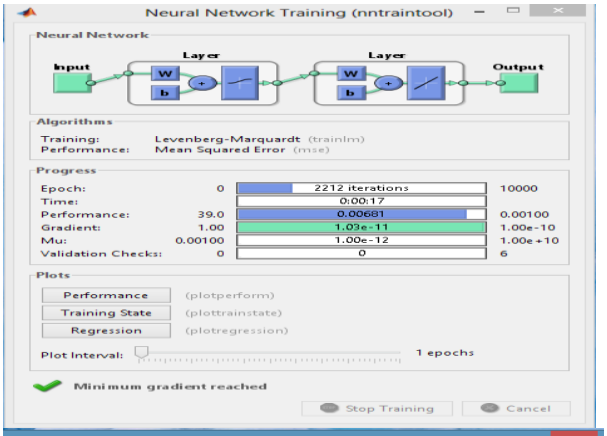
0.4921	0.4914	0.142248
0.5068	0.5054	0.276243
0.8681	0.8802	-1.39385
0.8794	0.8907	-1.28497
0.7108	0.7097	0.154755
0.5212	0.5216	-0.07675
0.5602	0.5692	-1.60657
0.8026	0.7983	0.535759
0.7689	0.7651	0.494213
0.5835	0.5811	0.411311
0.5906	0.5935	-0.49103
0.6604	0.6593	0.166566
0.8934	0.8902	0.358182
0.8555	0.852	0.409117
0.6395	0.6398	-0.04691
0.6837	0.6819	0.263273
0.7989	0.7963	0.325447
0.8976	0.8934	0.467914
1.0811	1.0781	0.277495

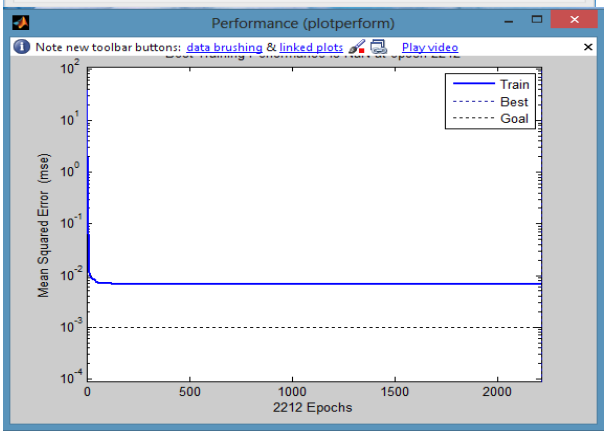
(Table-3) (Training Data)

Cexp	Cnn	error
0.5352	0.5481	2.41031
0.5632	0.5987	6.30327
0.5596	0.5953	6.37956
0.4919	0.5296	7.66416
0.9076	0.9612	5.90569
0.9704	1.0218	5.29678
1.0334	1.088	5.28353
0.7017	0.7413	5.64344

(Table-4) (Testing Data)

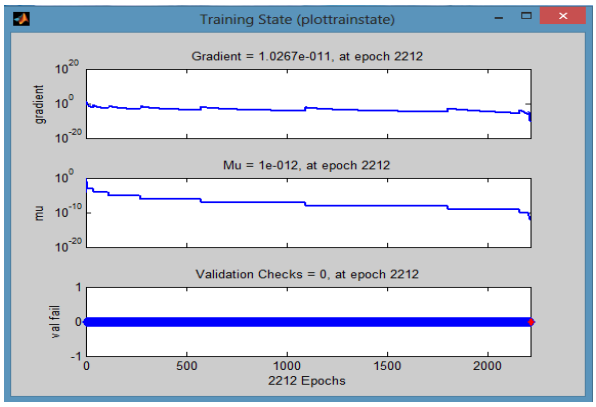
The Neural Network results and plots are given as below. The Plot -1 is Neural Network Training which is reached to goal by 10000 epoch within 39 sec and plot-2 mean square error almost constant Plot-3 and Plot-4 showing the training state and regression and is found the results in Plot -4 shows the error between experimental and predicted by neural network of the gyroscopic couple value are fitted as line of 45^0. The error is very small.

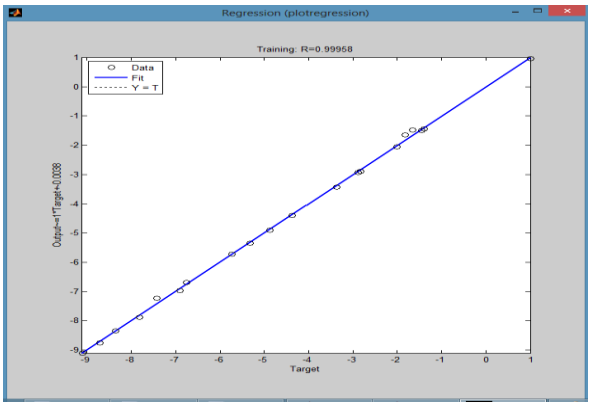




Plot-1. Neural Network Training Plot

Plot-2. Performance Plot





Plot-3. Training state Plot

Plot-4. Regression plot

CONCLUSION

It is observed that the prediction by the Neural Network is very close to the experimental value of Gyroscopic couple value by taking the number of neurons in hidden layer -1 and hidden layer-2 are taken 1 and 3 and the transfer function taken as logsig and trainlim , this is because only the training data set containing two set of input. The number of input parameters is increased in training data, then the limit of prediction by Neural Network may be estimated as upper and lower range of NN value .

REFERENCE

[1] "Minimization of Error in Training a Neural Network Using Gradient Descent Method "is a journal that was published by International Journal of Technical Research,This journal was presented by McCulloch and Pitts in McCulloch Pitts

[2] "Prediction using Artificial Neural Networks: Generalization Beyond the Calibration Range" –by Minsky and Papert

[3]. H. Abdi, D. Valentin, B. Edelman, *Neural Networks*, Thousand Oaks, CA: SAGE Publication Inc., 1999.
[4] S. Haykin, *Neural Networks*, New York, NY: Nacmillan College Publishing Company, Inc., 1994.
[5] T. Masters, *Practial Neural Network Recipes in C++*, Toronto, ON: Academic Press, Inc., 1993.
[6] R. Schalkoff, *Artificial Neural Networks*, Toronto, ON: the McGraw-Hill Companies, Inc., 1997.
[7] S. M. Weiss and C. A. Kulikowski, *Computer Systems That Learn*, San Mateo, CA: Morgan Kaufmann Publishers, Inc., 1991.
[8] P. D. Wasserman, Neural Computing: Theory and Practice, New York, NY: Van Nostrand Reinhold, 1989.