

AI based Vector Control Method for BLDC Motor with Multi Switch Three-Phase Topology

Ch. Vinay Kumar¹, G Madhusudhana Rao^{2*}, A Raghu Ram³

¹ Asst.Professor-MGIT, ²Professor-Bule Hora University, Ethiopia, ³Professor-JNTUCEH Hyderabad

ABSTRACT

Brushless direct current (BLDC) Motors are extensively used because of their characteristics. Such characteristics are high dynamic response and high-power density. BLDCM control system is a nonlinear, multi-variable, strong-coupling system. In this paper it is proposed that a neural network controller is used for the five level switch of the BLDC motor to enhance the power factor and reduce the current distortions with respect to its rise time, startup torque. This method is also done in comparison with the PID controllers. The working principle of the BLDC is with the help of five-switch control scheme can be implemented here. The different values of load were used to consider the total operation of the BLDC motor is to be controlled. After the completion of the training and testing of the neural network, it might be maintain the constant load values and its variables. To calculate the duty ratio of the DC-DC converter, it will be adjusted to regulate the speed of the BLDC motor. However the DC link of the five switching inverter is used here for the boosting of the voltage. The effectiveness of the proposed control technique can be realized with help of the speed sensor. Various tests have been conducted in the simulation the proposed technology is the robust technology and it is proven that very effective and suitable control technique.

Keywords

Neural Networks, BLDC Motor, Five-Switch converter, Current Distortions, Three Phase Power, Vector Control and

I. Introduction

Brushless DC motor (BLDCM) is a very broadly used one in the applications of the Electrical Engineering, Automobile and mining, Industrial applications due its feasibility and advantages of high performance, rugged maintainance, smooth operation and etc., Brushless DC motor (BLDCM) has been widely used in electrical equipment, mining and other industrial application fields due to its advantages of high power density, high efficiency, easy maintenance, silent operation and so on [1]-[3]. Brushless DC motor control system is a multi-variable, nonlinear, strong-coupling system. The high performance control strategy should be robust and adaptive. The conventional proportional-integral-derivative (PID) controller has shortcomings when controlling nonlinear and uncertain target. The interest in emerging intelligent controller for Brushless DC motor has been increased significantly. Some intelligent control methods have been applied in the Brushless DC motor controller [4]-[11]. To finding the problems like speed response, steady state error, power factor improvement in Brushless DC motor, and the conventional controllers are not sufficient controllers. Instead of using these conventional controllers it is suggested that the non

conventional controllers like fuzzy logic control technique, neural network technique will be the effective controllers for the Brushless DC motor. Here it is observed the fuzzy logic designed five level switching converter was developed to minimize the over currents and over voltages in the converter for speed control of brushless DC motor [4]. There are variety of methods are been considerable efforts in solving the soft computing approach to control of brushless motors. In this paper it is proposed that a novel method is identified to reduce the problems of the current and voltages of Brushless DC motor with the artificial neural Networks comparison with the PID controllers. The neural networks which show the effectiveness of these controllers such as speed response, steady-state error, among others. In this proposed method the tuning of the PID constants using the development of the neural networks, which it has been shows a better efficiency and performance than the conventional controllers [4]. BLDC motors are generally controlled employing three-phase inverters, so it requires a rotor position sensor to control the inverter in order to determine the switching instants and provide an appropriate sequence for switching. The position of the motor in these engines is generally determined by position sensors such as Hall Effect sensors, resolvers, and

absolute position sensors; otherwise, it must be prepared by estimating the parameters that specify the rotor position using Sensorless techniques [12], [13]. The driving range for each phase is usually 120 electrical degrees, which is divided into two 60-degree steps. For example, S5-S6, S1-S6, S1-S2, S3-S2, S3-S4, and S5-S4 are six steps of a 360-degree electric cycle. Therefore, only two phases are carried out at a time. The above can be seen graphically in Figure 1. From the operation of the engine and their elements, the BLDC engine is modeled to simulate its behavior. The brushless motor model implemented in the PSIM simulation environment is shown in Figure 3. This figure shows the three-phase inverter, which has a sensor to monitor the current supplied to the BLDC. It is also possible to see that the BLDC motor is coupled with a sensor for speed reading. This sensor generates the control signal that determines the pulse width of the PWM (Pulse Width Modulation) that activates each of the inverter's drivers, in addition to the necessary switching stage that depends on the state of the Hall Effect sensors. This switching allows the three-phase inverter to function normally in addition to preventing a short circuit in any of the three branches.

The two kind of Brushless [14] permanent magnet machine are used for traction application

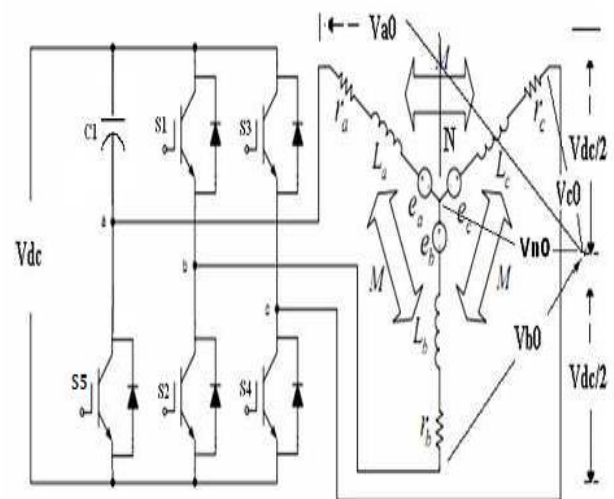
- BLDC motor in which the air gap flux distribution and counter EMF, or back EMF waveform are approximately trapezoidal as like in conventional DC machine.
- Standard permanent magnet [15], [16] synchronous motor (PMSM) in which the air gap flux distribution and back EMF waveform are both sinusoidal is sometimes called as a Sinusoidal Brushless DC motor when operated in a self-controlled mode.

Indeed the drive characteristic and control methods are very similar for the trapezoidal and sinusoidal machines. In both cases, the motor must be energized with controlled currents that are synchronized with rotor position. However a distinction is now being drawn between the two drives, because of the differences in machine construction and the standard synchronous motor requires sinusoidal current excitation, whereas the trapezoidal machine is energized with square-wave and quasi-wave currents.

The rotor position sensors of the trapezoidal machine usually consist of a number of simple

position detectors, such as Hall Effect devices that can sense rotor magnetic field and determine the phase switching points. The sinusoidal machine requires more precise position information to allow accurate synthesis of the sinusoidal current waveforms. Following are the significant of using brush less motor design;

- The position sensor system for the shaft is simple and needs only to deliver six digital signals for commanding the transistor of the inverter.
- The quasi square-wave armatures current are mainly characterized by their maximum amplitude value, which directly controls the machine torque.
- The inverter performance is very much reliable because there is natural dead time for each transistor.
- The first and third characteristics allow reducing the complex circuitry required by other machine and allowing the self-synchronization process for the operation of the machine. The second characteristic allows designing a circuit [17] for controlling only a DC component which represents the maximum amplitude value of the trapezoidal current, I_{MAX} .



- Fig 1: Inverter Circuit with BLDC Machine
- The Fig. 2 shows the way the stator currents are controlled, through the feedback signal $I_{MAX}(t)$. The slopes m_1 (up) and m_2 (down) are a function of the dc link voltage, and the motor model. Therefore motor model comprises the phase inductance and the back EMF. Then, the maximum error based on a

carrier e_{MAX} , will depend on the slopes m_1 and m_2 , on the operating point, and the carrier frequency. This situation can be expressed mathematically, with the help of Fig. 4, as seen in

$$\begin{aligned} \Rightarrow |m_1| &= \frac{E_{pp}}{xT} = \frac{E_{pp} \cdot f}{x} \\ \frac{|m_2|}{|m_1| + |m_2|} &= \frac{E_{pp}}{(1-x)T} = \frac{E_{pp} \cdot f}{1-x} \\ E_{pp} &= \frac{1}{f} \left(\frac{|m_1| \cdot |m_2|}{|m_1| + |m_2|} \right) \quad (1) \end{aligned}$$

- The term 'x' in equation 1 represents the fraction of the period of the carrier 'T', when the current is increasing. According to this equation, the term 'x' defines the ideal output of the PWM pattern. Then, the control parameters have to be adjusted according to the index 'x'. E_{pp} represents the peak-to-peak error of the signal.

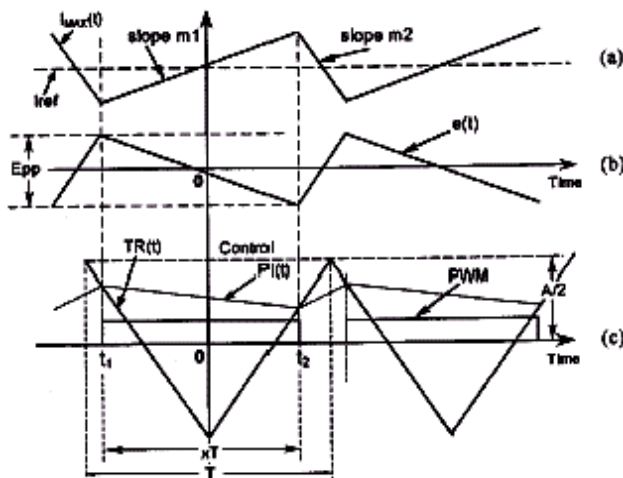


Fig 2: Ideal waveforms of (a) feedback signal I_{MAX} , (b) error signal $e(t)$, and (c) control signals.[D1]

According to the time reference of Fig. 2, the following relations can be written:

$$\left. \begin{aligned} (t_2 + t_1) &< x \times T \\ (t_2 - t_1) &= x \times T \\ |t_2| &\leq |t_1| \\ e(t_2) + e(t_1) &= E_{pp} \\ e(t_2) - e(t_1) &= E_{pp} \end{aligned} \right\} \quad (2)$$

$$e(t_1) = \frac{E_{pp}}{2}$$

same value at t_1 and t_2 , because these are intersection points that define the PWM generation

$$\begin{aligned} PI(t_1) &= TR(t_1) \rightarrow K_p \times e_c(t_1) + M \\ &= -\frac{2 \cdot A}{T} \times t_1 - \frac{A}{2} \quad (3) \end{aligned}$$

$$\begin{aligned} PI(t_2) &= TR(t_2) \rightarrow K_p \times e_c(t_2) + M \\ &= \frac{2 \cdot A}{T} \times t_2 - \frac{A}{2} \quad (4) \end{aligned}$$

With equation 3 and equation 4, K_p and M are obtained

$$K_p = \frac{\frac{2 \cdot A}{T} \cdot (t_2 + t_1)}{\alpha \cdot (e(t_2) - e(t_1))} < \frac{\frac{2 \cdot A}{T} \cdot x \cdot T}{\alpha \cdot (e(t_2) - e(t_1))} \quad (5)$$

Now, when the current I_{MAX} is increasing, the error $e(t)$ can be modeled as

$$e(t) = -m_1 \times t + n_1, \text{ para } t \in [t_1, t_2] \quad (6)$$

If the rate of conversion of the magnitudes is $1:\alpha$, then the expressions of the error $e_{ci}(t)$ becomes as shown in

$$e_c(t) = \alpha \times e(t). \quad (7)$$

On the other hand, the output of the PI control can be expressed as:

$$PI(t) = K_p \times e_c(t) + K_I \times \int_0^t e_c(t) \times dt \quad (8)$$

Now, if it is assumed that the operation of the machine is stable, and operates under steady state, the integrator output is close to a constant. That means

$$M = K_I \times \int_0^t e_c(t) \times dt = \text{constant} \quad (9)$$

The triangular carrier and the PI output have the equation (9)

$$M = \frac{1}{2} \left[\frac{2 \cdot A}{T} (t_2 - t_1) - A - K_p \cdot \alpha \cdot (e(t_2) + e(t_1)) \right] \quad (10)$$

Now, with the help of equation (8), (9) and (10), can be resumed as

$$K_p < \frac{1}{\alpha} \cdot \left(\frac{2 \cdot A}{E_{pp}} \right) \cdot x \quad (11)$$

$$M = A \cdot \left(x - \frac{1}{2} \right) \quad (12)$$

Replacing equation (9) in (11), the value K_I of is obtained

$$K_I = \frac{A \cdot \left(x - \frac{1}{2} \right)}{\int_0^{t_1} e_c(t) \times dt} = \frac{A \cdot \left(x - \frac{1}{2} \right)}{\alpha \cdot \int_0^{t_1} e(t) \times dt} \quad (13)$$

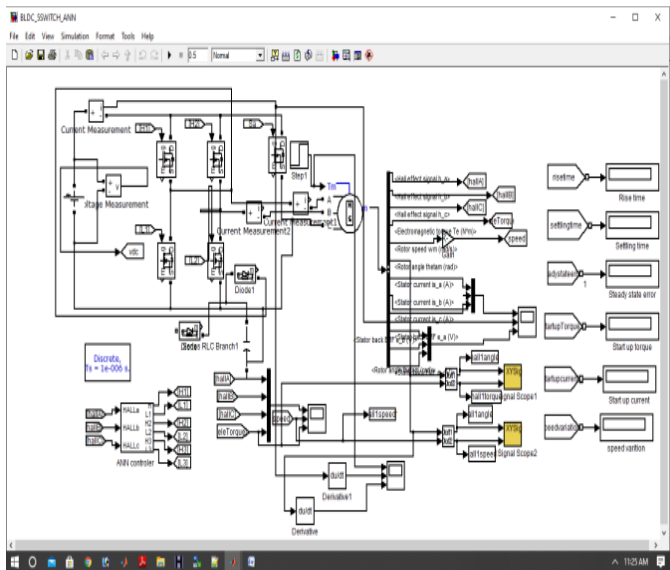


Fig 3: Simulation Diagram of ANN based Five Switch BLDC Motor

Boost Converter: Boost converter, step-up converter is a power converter with an output DC voltage greater than its input DC voltage. It is a class of switching mode power supply, called SMPS, containing at least two Semiconductors switches, a diode and a transistor, and at least one energy storage element. Switch turns on and incrementally stores energy from V_1 in L . Switch turns off and this energy and additional energy from input is transferred to output. Therefore, L used as a temporary storage element filters made of inductor and capacitor combinations are often added to a converter's output to improve performance. The output voltage function for this converter is

$$V_o = \frac{V_{IN}}{1-D} \quad (14)$$

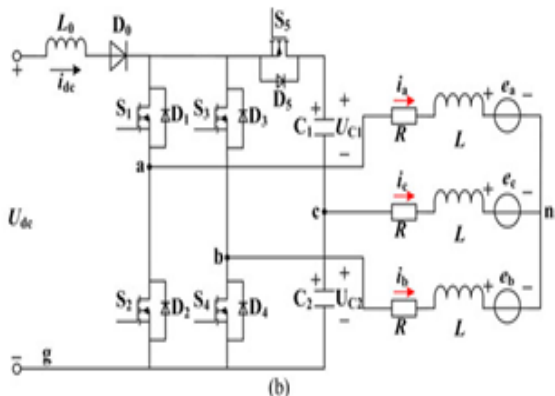


Fig 4: BLDCM drives, three-phase currents five commutating points.

II. NEURAL NETWORKS

Traditionally rheostat armature control method is used for low power [18] DC motors, Use of conventional PID controllers, neural network controllers (NNC). Constant power field weakening [19] controller is based on load-adaptive multi-input, multi output linearization technique (in high speed regimes). A single phase uniform PWM ac-dc buck-boost converter [20] with only one switching device is used for armature voltage control. Through experience gained in designing trajectory [21] controllers based on self-tuning and PID control, it is seen that the neural network controller gives comparable performance in speed tracking [22]. In addition to those mentioned above, a unique advantage of the neural network controller is, its ability to cope with bad measurement data that occur during training and testing. However, a key drawback is the inadequate integral gain in the feedback loop, resulting in steady-state errors of the shaft position. Direct position tracking can alleviate this problem. The traditional means adapted by the motion control industry with motor drives have been the approach of linearizing the dynamic system and designing a linear feedback controller. However, in high-performance motor drives such an approach is seldom satisfactory, as it results in poor speed and position tracking when sudden changes in load result in continuous acceleration or deceleration of the motor/load system. Adaptation [23] is necessary to ensure optimal performance.

The neural network consists of junctions, which are connected with LINKS are also called processing units. For each junction a number is ordered, which is called weight. The weights are the tools to store the long distance estimation in the neural networks. The learning process [24] occurs with the appropriate modification of weights. These weights are modified so that the network input and output behavior is in consonance with the environment, which provides the input data.

The calculation of algorithm consists of the following basic steps:

- Calculation of the output of the network, with inputs and weights.

- Modification of weights with learning algorithm.

A single input neuron consists of a scalar input 'p', multiplied by the scalar weight 'w', to form 'wp' which is fed to the summer along with bias 'b' multiplied by 1 is as following:

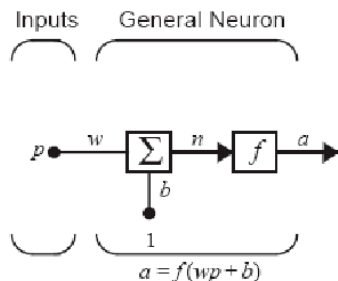


Fig 5: Basic Neural Network

The net input is 'wp+b' and the output 'a' is; $a=f(wp+b)$; f- Transfer function W & b can be adjusted by learning rule.

BACK PROPAGATION ALGORITHM

The Back-propagation algorithm is also known as the generalized delta rule. The algorithm provides a method of calculating the gradient of the error function through the use of differentiation (chain rule). As the name implies, the error is propagated backward from the output, one layer at a time.

The first step towards the implementation of the back propagation procedure involves the forward propagation of training examples through the Network similar to the Perceptron. These training data are fed to the Network layer by layer. At each layer, activation potential is determined. Forward propagation is done to calculate cell activation. The error after initial computation in the forward pass is propagated from the output layer, layer by layer to the front. Repetition of the process will be done till a relatively small rate of change of error is achieved. The recommended rate of change of error of 0.1 to 1 percent is sufficient to terminate training. However, the rate of change of error varies from different applications; hence 1 percent should only be used as a guideline.

There are two modes of training for the algorithm: *Pattern and Batch Training*:

In the Pattern mode, the update of weights is performed after each training example. In the batch mode, the weight is updated after a whole set of training example (an epoch). The fact that batch mode follows the direction of steepest

descent in the error function guarantees that it will find a set of weights that correspond to a minimum of the error function. Instead of utilizing a hard limiting Linear Threshold Unit, a sigmoid is employed. The main difference between a LTU and sigmoid is that the sigmoid is differentiable. Computations in the algorithm are carried out in parallel, one layer at a time.

The basic approach to application of Back propagation Algorithm is as followed:

- A preferred activation will be [1, -1] as compared to traditional [0, 1]. Theoretical and experimental evidence has been established.
- A small positive value for the step size, p, is chosen. Even though a larger value of p results in faster convergence, such convergence is up to a certain limit. To large a value of p results in the algorithm to become unstable and failure to converge. Hence it is recommended to keep the value of p to a maximum of 0.1.
- The initial weights will have to be small and randomly generated and assigned. A practical range of weights is to keep them $[-2/z, 2/z]$ for a cell with z inputs.
- Forward propagation is done to compute the weighted sum, S_i and activation, $u_i = f(S_i)$ for every cell.
- Back propagation [25] is done from the output, layer by layer to the front. In this process, the synaptic weights are adjusted in accordance to error-correction rules listed in equation

$$f^1(s_i) = u_i(1-u_i) \quad (15)$$

$$\delta_i = \begin{cases} (c_i - u_i) \\ (\sum_{m:m>1} w_{m,i} \delta_m) f^1(s_i) \end{cases} \quad (16)$$

If u_1 is a output unit for other

Lastly the weights are updated with the formula given by

$$W_{ij}^* = W_{ij} + p\delta u_j \quad (17)$$

STEEPEST DESCENT ALGORITHM FOR THE APPROX MEAN SQUARE ERROR

$$w_{ij}^m(k+1) = w_{ij}^m(k) - \alpha \frac{\partial \hat{f}}{\partial w_{i,j}^m} \quad (18)$$

$$b_j^m(k+1) = b_j^m(k) - \alpha \frac{\partial \hat{f}}{\partial b_j^m} \quad (19)$$

Where α is the learning rate, i, j, m - Layers of neurons F - Performance Index w, b - respective weights and bias as per their sub-script and superscript.

The science of artificial neural networks is based on the biological neuron. In order to understand the structure of artificial networks, the basic elements of the neuron should be understood. Neurons are the fundamental elements in the central nervous system. A neuron is made up of 3 main parts -dendrites, cell body and axon. The dendrites receive signals coming from the neighboring neurons. The dendrites send their signals to the body of the cell. The cell body contains the nucleus of the neuron. If the sums of the received signals are greater than a threshold value, the neuron fires by sending an electrical pulse along the axon to the next neuron. The model of artificial neuron is based on the components of the biological neuron as shown in Fig 6 & 7. The inputs X_0 - X_3 represent the dendrites. Each input is multiplied by weights W_0 - W_3 . The output of the neuron model, Y is a function, 'F' of the summation of the input signals.

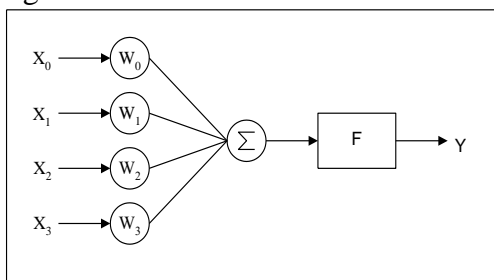


Fig 6: Artificial Neuron Model

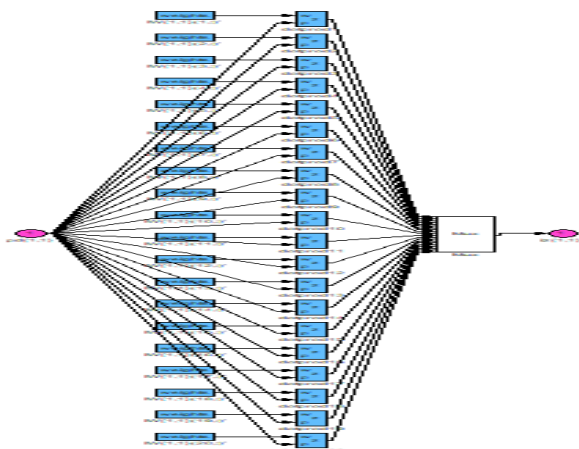


Fig 7: Multilayered Artificial Neural Networks

SIGNIFICANCE OF USING ANN

The main advantage of neural networks is to train a neural network to perform a particular function by adjusting the values of connections (weights)

between elements. For example, if we want to train a neuron model to approximate a specific function, the weights that multiply each input signal will be updated until the output from the neuron is similar to the function. Neural networks are composed of elements operating in parallel. Parallel processing allows increased speed of calculation compared to slower sequential processing. Artificial neural networks [26], [27] (ANN) have memory. The memory in neural networks corresponds to the weights in the neurons. Neural networks can be trained offline and then transferred into a process, where adaptive learning takes place. In this case, a neural network controller could be trained to generate control signal for driving logic offline in the MATLAB (SIMULINK) environment. After training, the network weights are set. The ANN is placed in a feedback loop with the natural control generator. The network will adapt the weights to improve performance, as it generates control signal. In spite of these advantages ANN has some disadvantages. The main disadvantage of ANN is, they operate as black boxes. The rules of operation in neural networks are completely unknown. Another disadvantage is the amount of time taken to train networks. It can take considerable time to train an ANN for certain functions.

ANN MODELING FOR DC MACHINE

Neural networks have 3 main modes of operation – supervised, reinforced and unsupervised learning. In supervised learning the output from the neural network is compared with a set of targets, the error signal is used to update the weights in the neural network. Reinforced learning is similar to supervised learning. However there are no targets given, the algorithm is given a grade of the ANN performance. Unsupervised learning updates the weights based on the input data only. In case of generating control signals, supervised learning is used.

Neural Network Structures

There are 3 main types of ANN structures - single layer feed-forward network, multi-layer feed-forward network and recurrent networks.

Type of activation function (F)

There are 3 main types of activation function - tan-sigmoid, log-sigmoid and linear as shown in Fig 2.9. Different activation functions affect the performance of an ANN.

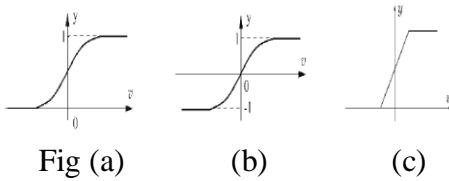


Fig 8: Activation function (a) Tan-sigmoid (b) Log-sigmoid (c) Linear

In proposed method neural network is used to determine the exact instant of crossing of reference sine wave with carrier i.e. $(t_k + \Delta t_k)$, when input given to neural network is time instant ' t_k ' as shown in Fig 8. Supervised learning method is used to train the neural network. In supervised learning we have to specify the input-output target pair for train neural network. Here input is given as time instants ' t_k ' at which carrier waveform achieves their peaks (both positive and negative) and corresponding target is specified by exact instant of crossing of carrier and reference i.e. $(t_k + \Delta t_k)$ for corresponding value of ' t_k '. Multilayer neural network having 3 layers is built. First two layers have 35 neurons with transfer function as tan-sigmoid and last layer has 1 neuron with transfer function as linear. MATLAB package provides many powerful tools for building and training neural network.

Usually, the neural networks have several layers composed of a different number of neurons each of them. In general, neural networks are composed of three different layers: input, hidden, and output. The information goes from one layer to another through the transfer functions. The transfer function determines how the information is transmitted between layers. Weights are directly associated with the connection between layers. That is, if a connection has weight with a high value, then it will make more contributions to the network output. One of the most used neural networks is called multi-layer perceptron, and the best-known training method for this type of network is back propagation. This algorithm is used more for learning this kind of multi-layer perceptron network. The back propagation algorithm trains the neural network from sample vectors of the system that are of interest for their modeling, such as the texture of an object or, in our case, the speed of a motor.

Summarizing, the back propagation training system consists of the following steps:

1. Initialize the weights of the network randomly.

2. Enter input data from among those to be used for training.
3. Let the network generate an output data vector (forward propagation).
4. Compare the network output with the desired output.
5. The difference between the generated and the desired output (called error) is used to adjust the weights in the output layer.
6. The error spreads backwards (hence the name of back propagation), towards the previous neuron layer, and is used to adjust the weights of that layer.
7. Continue propagating the error backwards and adjusting the weights until the input layer is reached.

The proposed controller can meet the requirements of stability, rapidity, accuracy and robustness in BLDCM drivers. One of the advantages that back propagation presents is a speed of convergence and robustness compared to other types of training. For this reason, in this work, a multi-layer perceptron neuronal network has been implemented using back propagation as a training algorithm. This network was used for speed control of a brushless motor. The architecture of the neural network used for the development of this work has a topology 2:8:1 and uses only two layers. The relation 2:8:1 means that the neural network has an input vector of a size of two elements. This means that every example presented to the neural network only has two elements. The second number 8 belongs to the number of neurons that have the first hidden layer. Finally, number 1 tells us that there is only one neuron in the output layer. As it is possible to see, the neural network needs only two layers of neurons in order to obtain an excellent performance of the control system proposed. However, just with this configuration, the neural network is capable of learning the vectors related to brushless motor control. As previously noted, it is necessary to have example vectors related to the problem to be solved by the neural network. These sample vectors must come from real examples of the objects or the system to be modeled. In this case, examples of the motor's response to different applied torques are required. For this reason, a classical Proportional, Integral, and Derivative (PID) controller has been compared in order to obtain values to train the proposed network.

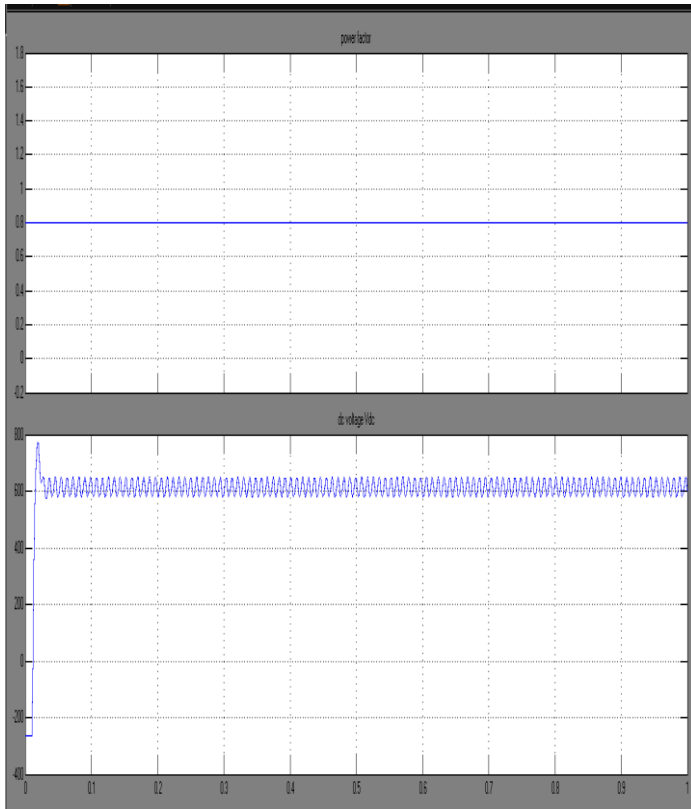


Fig : ANN based power factor (PF 0.8) & dc voltage (Vdc 600)

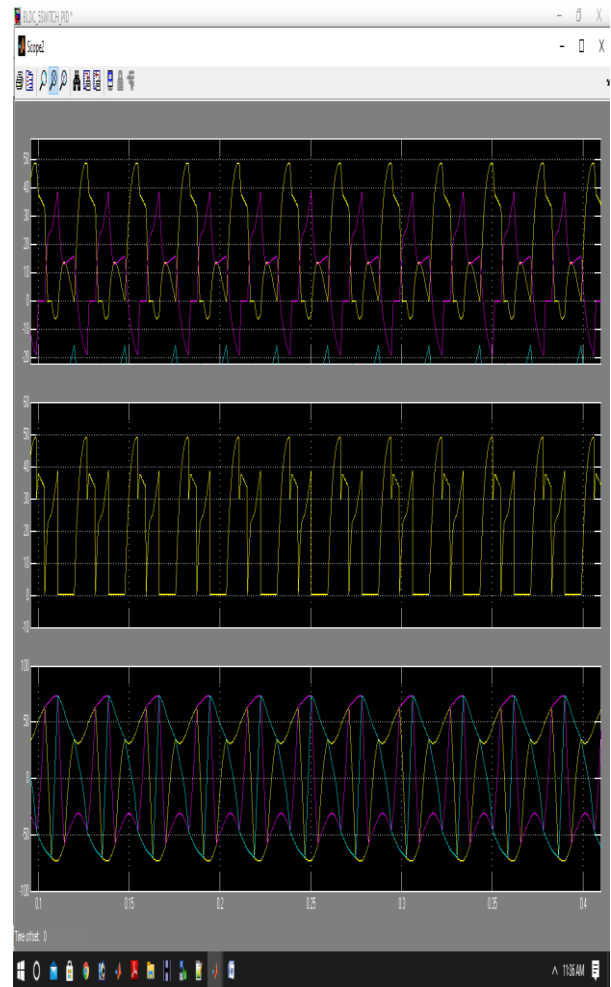


Fig 8: BLDCM five switch wave forms using Neural Network method

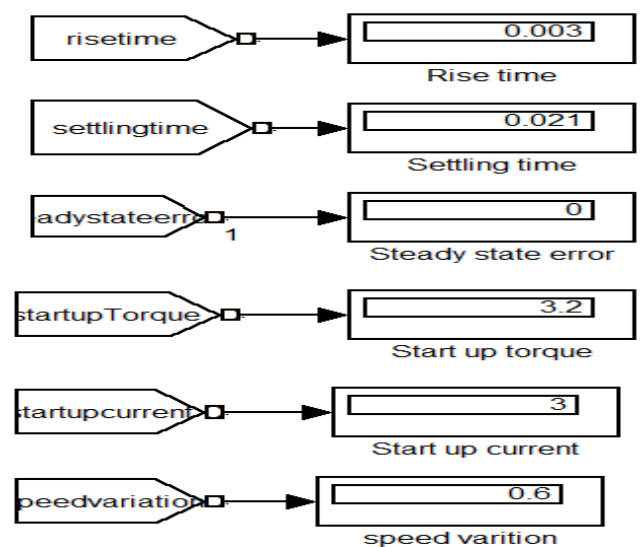


Fig : Parameter Results of Simulation

CONCLUSION

In this work, the development of the current control of a BLDC motor using a multi-layer perceptron type neural network based five switch controls was presented. From the simulation, the test results are carried out in this paper, it was seen that the control implemented from the proposed neural network scheme is well-organized, and reasonable results than the conventional controllers in the presence of constant load are obtained. As per the results obtained in the various simulations carried out in this implementation, it was obvious that the neural network is superior to conventional controllers. Due to this, in most of the applications, it is difficult to find a constant load torque;

Therefore, the use of conventional controllers control may not be suitable for a system. The working principle of the BLDC is with the help of five-switch control scheme can be implemented here. The different values of load were used to consider the total operation of the BLDC motor is to be controlled. After the completion of the training and testing of the neural network, it might be maintain the constant load values and its variables. To calculate the duty ratio of the DC-DC converter, it will be adjusted to regulate the speed of the BLDC motor. However the DC link of the five switching inverter is used here for the boosting of the voltage. If the loads are variable loads then the conventional controllers are not capable to sustain it parameters. At this point, the neural network was superior since it is sufficient of supporting torque changes without the need for a speed sensor. In conclusion, it is noted that the development of the neural network is computationally less expensive.

REFERENCES

- [1] X. Huang, A. Goodman, C. Gerada, Y. Fang and Q. Lu, "Design of a five-phase brushless DC motor for a safety critical aerospace application", IEEE Transactions on Industrial Electronics, vol. 59, no. 9, (2012), pp. 3532-3541.
- [2] K. T. Chau, C. C. Chan and C. Liu, "Overview of permanent-magnet **brushless** drives for electric and hybrid electric vehicles", IEEE Transactions

Parameters		ANN
Rise time	0-700 RPM	0.003
	700-900RPM	0.003
Settling time	0-700 RPM	0.020
	700-900RPM	0.003
Steady state error	0-700 RPM	0.9%
	700-900RPM	0.9%
Start up torque	0-700 RPM	3.2 N.N
	700-900RPM	1.3 N.M
Start up current	0-700 RPM	3A
	700-900RPM	1A
Speed variation	0.7%	0.6%
Power factor	0.7	0.8
DC voltage	500 Vdc	600Vdc

on Industrial Electronics, vol. 55, no. 6, (2008), pp. 2246-2257.

- [3] N. Milivojevic, M. Krishnamurthy, A. Emadi and Stamenkovic, "Theory and implementation of a simple digital control strategy for brushless DC generators", IEEE Transactions on Power Electronics, vol. 26, no. 11, (2011), pp. 3345-3356.
- [4] S. Sivakotiah and J. Rekha, "Speed Control of Brushless Dc Motor On Resonant Pole Inverter Using Fuzzy Logic Controller", International Journal of Engineering Science and Technology, vol. 3, no.10, (2011), pp. 7360-7367.
- [5] U. Ansari, S. Alam, S. Minhaj and N. Jafri, "Modeling and control of three phase BLDC motor using PID with genetic algorithm", Computer Modelling and Simulation, UkSim 13th International Conference on IEEE, (2011).
- [6] C. Reddy, S. RAMI and M. S. Kalavathi, "Performance evaluation of hybrid fuzzy logic controller for brushless dc motor drive",

- International Journal of Engineering Science and Technology, vol. 3, no. 6, (2011), pp. 4749-4758.
- [7] A. Al-Mashakbeh and S. Othman, "Proportional Integral and Derivative Control of Brushless DC Motor", European Journal of Scientific Research, vol. 35, no. 2, (2009), pp. 198-203.
- [8] H. N. Nounou and H. Rehman, "Application of adaptive fuzzy control to ac machines", Applied Soft Computing, vol. 7, no. 3, (2007), pp. 899-907.
- [9] R. Kandiban and R. Arulmozhiyal, "Design of Adaptive Fuzzy PID Controller for Speed control of BLDC Motor", International Journal of Soft Computing and Engineering, vol. 2, no. 1, (2012), pp. 386-391.
- [10] C. H. Wang, H. L. Liu and T. C. Lin, "Direct adaptive fuzzy-neural control with state observer and supervisory controller for unknown nonlinear dynamical systems", Fuzzy Systems, IEEE Transactions, vol. 10, no. 1, (2002), pp. 39-49.
- [11] W. D. Chang and J. J. Yan, "Adaptive robust PID controller design based on a sliding mode for uncertain chaotic systems", Chaos, Solitons & Fractals, vol. 26, no. 1, (2005), pp. 167-175.
- [12] Mozaffari Niapour, S.; Shokri Garjan, G.; Shafiei, M.; Feyzi, M.; Danyali, S.; Bahrami Kouhshahi, M. Review of permanent-magnet brushless DC motor basic drives based on analysis and simulation study. Int. Rev. Electr. Eng. 2014, 9, 930–957. [CrossRef]
- [13] 45. Frolov, V.Y.; Zhiligitov, R.I. Development of sensorless vector control system for permanent magnet synchronous motor in Matlab Simulink. J. Min. Inst. 2018, 229, doi:10.25515/PMI.2018.1.92. [CrossRef]
- [14] Devendra Rai "Simulink Simulator usage manual for Brushless DC motor" Dept of E&CE NIT (Surathkal) Karnataka
- [15] P.Pillay and R.Krishnan "Application Characteristics of permanent magnet synchronous and brushless DC motors for servo drives" IEEE Tran. Ind. Applicant, Vol. 27, PP. 986 – 996, Sept / Oct 1991.
- [16] T.Low and M.A Jabbar, "Permanent-magnet motors for brushless operation", IEEE Trans. Ind Application, Vol. 26, PP. 124-129, Jan / Feb 1990
- [17] M.H.Rashid "Power Electronics Circuit, Device and Applications" PHI Publication Ltd., 3rd Edition, 2005
- [18] Samir Salama "Simulation of power electronics system using "SimuPec: The new power electronics tool box for "Simulink / MATLAB" 2004, Annual, 35th IEEE, Tran. Power Electronics specialists' conference.
- [19] Abido M. A., "Optimal design of power-system stabilizers using particle swarm optimization", IEEE Trans. Energy Conversion, Sep. 2002, vol. 17, p. 406 – 413.
- [20] S.G German-Gankin – "The computing modeling for power electronic systems in Matlab", 2001
- [21] Rasid, "High performance nonlinear MIMO field weakening controller of a separately excited dc motor," Electric Power Systems Research, vol. 55, issue 3, Sep. 2000, pp. 157-164.
- [22] El-Sharkawi, M. A., & Weerasooriya, S. , "Adaptive tracking control for high performance dc drives". IEEE Transactions on Energy Conversion, volume-5, September 1989.,122-128.
- [23] Kotaru, R., & Rubai, A. , "Online Identification and control of a dc motor using learning adaptation of neural networks". IEEE Transactions on industry applications, volume-36 ,June 2000, 935-942.
- [24] Kotaru, R., & Rubai, A. , "Online Identification and control of a dc motor using learning adaptation of neural networks". IEEE Transactions on industry applications, volume-36 ,June 2000, 935-942.
- [25] S.Weerasoorya and M.A Al-Sharkawi "Identification and control of a DC Motor using back-propagation neural networks" IEEE transactions on Energy Conversion, Vol.6, No.4 pp663-669,December 1991
- [26] Antsaklis, P. J. "Neural Networks in control systems", IEEE control systems magazine, volume-10 , April 1990. Pp.3-5.
- [27] Hagan, M. T., Demuth, H. B., & Jesus, O. , "An Introduction To The Use of Neural Networks". Electrical and Computer Engg. , 2000, pp. 1-23