Speed Control of a DC-Motor Using Artificial Neural Network

Ragavi Pobbathi Ashok
Department of Electrical and Electronics Engineering
The University of Auckland
Auckland, New Zealand.
rpob456@aucklanduni.ac.nz

Adedomola Wuraola and Nitish Patel
Department of Electrical and Computer Engineering
The University of Auckland
Auckland, New Zealand
awur978@aucklanduni.ac.nz
&
nd.patel@aucklandac.nz

Abstract- **This paper provides an insight into the speed control of a D.C motor using an artificial neural network controller by replacing the PID controller. Non-linear autoregressive exogenous (NARX) model neural network model type architecture to replace with a "Neural Network controller". Different types of inputs are used to test the network and with different architectures. Simulation results are presented to show effectiveness and advantage of speed control system of DC motor with ANNs in comparison with PID controller. The data from the ANN controller is sent to D.C motor in Simulink and the results are compared with the PID controller.**

Key words: D.C motor, Proportional-Integral-Derivative (PID) controller, Artificial Neural Network, Mat lab, Simulink.

1 INTRODUCTION

The development of high-speed DC motor is very important for the Industrial applications. Generally, a high-performance motor drive system must have a good dynamic speed command tracking and load regulating response. Applications with accurate speed response use DC motor as the main component to obtain the necessary results. DC motor is considered as the SISO system having torque/speed characteristics with compatible with mechanical loads. This advantage makes the DC motor controllable over a wide range of speeds by necessary adjustment of terminal voltage. But many controls issue problems arises such as variable and un-predictable inputs, noise propagation along the series, unknown parameters, change in load dynamics. Under these conditions the gain feedback controller fails to obtain the performance of the systems at acceptable level. The DC motors find their usage in various applications that include conveyors, turntables and other devices that require speed adjustments with constant and low levels of torque. DC motors also perform well under dynamic braking and reversing applications, which are most common in many industrial machines.

Despite several advanced techniques and algorithm, Proportional- Integral-Derivative (PID) has a solid space in the control systems used in majority of the industrial applications, because of its efficiency and performance. However, performance of the non-linear systems depends on the tuning of controllers. But the PID controller may produce overshoots as the performance of the system deteriorates due to the change in various system parameters. Self-tuning strategies well suits for the systems with time varying characteristics and inability to capture the unknown load characteristics over a widely ranged operation point, thus making the tuning difficult.

The ANN based controller are demonstrating high potential in the control of non-linear dynamic systems. ANN can be trained to estimate the evolution of the system which generated the data set. ANN controller have an inherent noise rejection capability, so they are expected to display superior performance under noisy operating environments. Also, the ANN's are proving to be very well suited for implementation in real time control systems because they do not have high computation demands and can be implemented with minimum complexity and cost.

This paper introduces a control system for permanent magnet DC motor drives based on PID and ANN controllers. The system provides automatic evaluation of the PID controller's parameters through a series of auto-tuning methods. The implemented ANN controller is trained offline using the data obtained from the experiments performed by the system. The main aim of the paper is to replace PID with ANN.

2 LITERATURE REVIEW

(**Aamir, 2013**), the author used supervised learning to train the neural network, where the data set is presented to train the network before simulation is run to get the output results. The ANN controller (Reference control Model) has two units Controller-Available to train the controller, according to has a Plant-Available for the plant to train the network. The implementation of the ANN controller is less complicated and less costly, but it consumes more time when the large data set is sent to train.

(Cheon, Kim, Hamadache, & Lee, 2015) paper focused on presenting the utilizing possibility of deep learning in the control sector. Main aim is to mimic the PID using the DBM (Deep belief network) algorithm. DBM has 2 main procedures: 1) Pre-training procedure – It is done using RBM (Restricted Boltzmann Machine) and to calculate the initial weights for the second step.2) Fine-tuning procedure- Learning process is performed by changing the weights so the input data follows the output data, this is used to develop deep learning controller. This learning algorithm is broadly based on the Deep Neural Network toolbox which was developed by (Tanaka & Okutomi, 2014). Deep learning consists of hidden layers and neurons that improve the learning performance. Using these features many large and complex problems that cannot be solved with conventional neural networks can be resolved by deep learning algorithms .DBM is considered as the complex process.

(Alhanjouri) The author in this paper it is believed that neural control scheme consists of 2 parts, one is the neural identifier which determines the motor speed and the neutral controller is used to generate control signals for a converter. These two neural networks are trained by Levenberg-Marquardt back-propagation algorithm. It is operating at two stages: -the first, NARMA-L2 (Nonlinear Autoregressive-Moving Average) controller used to control the speed under different external loads conditions. The second, the controller is performance at different reference speed. The training of NARMA-L2 is simply a re-arrangement of the neural network plant model, which is trained offline, in batch form includes 2 stages:

1. System identification.
2. Control design System identification stage that developed a neural network model of the plant to be controlled. Control design stage use the neural network plant model to train the controller.

It provides satisfactory performance and good response (zero steady state) but it doesn't provide a smooth linear process.

(Jeen Ann Abraham, 2018) in this paper, Neural Network Predictive controller is used to improve the Performance and effectiveness than the deep learning controller. This algorithm is based on the cost Function over a finite period to track the reference trajectory by the plant (DC motor) with certain tolerance. NNPC has better control on DC motor with minor delay but the preciseness of Neural Network is less.

(Atri & Ilyas, 2012) this paper focuses on the ANN back Propagation Algorithm as stated by (George, 2008). The Backpropagation training technique adjusts the weights in all connecting links and thresholds in the nodes so that the difference between the actual output and target output are minimized for all given training patterns. It is an easy procedure, but the output is not fixed for the same parameters. ***(Atri & Ilyas, 2012)***

(Solanki, 2016) The author focused on NN training tool is used to train the Neural Network with the Input and output parameters obtained by the PID controller. And then the NN controller is replaced with the PID in Simulink to obtain the results. This process is only possible for the simple set of data but produces a larger error with complicated input.(**(Solanki, 2016).**

3 METHODOLOGY

3.1 Modelling using DC motor

The Control application was done by modelling the DC motor with voltage as input and speed as output followed by the stimulating same with different controllers in mat lab/Simulink. The dynamic equation are shown below

$$\frac{dw}{dt} = \frac{1}{J}\left(K_t i - bw\right) \quad (1)$$

$$\frac{di}{dt} = \frac{1}{L}\left(V - Ri - K_e w\right) \quad (2)$$

Where J is the moment of inertia of the rotor, K_t is motor torque constant, i is the armature current, b is the motor viscous friction constant, L is the electric inductance, R is the electric resistance, and K_e is electromotive force constant.

3.2 PID controller and tuning

The PID controller has a simple three term controller. The letter P, I and D stand for Proportional, Integral and Derivative. Most of PID controllers are tuned on-site. The transfer function of PID controller is given in equation where Kp is Proportional gain, Ki is Integral gain and Kd is Derivative gain. ID controller is used in a closed-loop unity feedback system. The variables indicate the tracker error, which is sent to the PID controller. The signal, from the controller to the plant is equal to proportional gain (Kp) multiply with magnitude of the error plus the integral gain (Ki) multiply with integral of the error plus the derivative gain (Kd) multiply with derivative of the error. The equation for signal is illustrated by equation.

C(s)=Kp+Kt/s+Kd*s=Kds^2+Kp*s+Ki / s

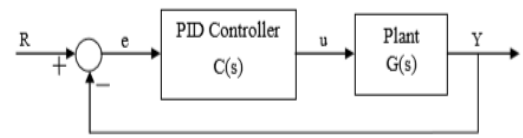


Figure A: Simulink representation of PID controller

As PID is dependent only on the three parameters, but it's difficult to find accurate values of the parameters to obtain the system stability. PID can be implemented both in the series and parallel configuration. Though parallel configuration produces the better performance. The equation of PID is given as

$$C = K_p c\left(1 + \frac{1}{T_i s} + \frac{T_d s}{\frac{T_d}{N}s+1}\right) \quad (3)$$

PID tuning is the method, where PID controller parameters can be tuned to achieve a robust design with the desired response time. This can be done through the Simulink software which involves the 2 steps.

1. Tune the controller in the PID Tuner by manually adjusting design criteria in two design modes. The tuner computes PID parameters that robustly stabilize the system.
2. Export the parameters of the designed controller back to the PID Controller block

and verify controller performance in Simulink.

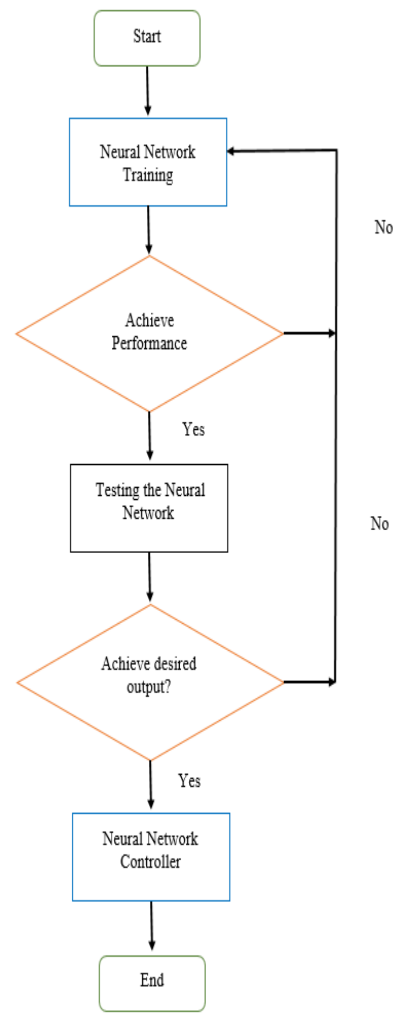
3.3 *Differential Drive*

The wheels of a car tend to spin at varied speeds and the speed matters the most when the vehicle is expected to turn. This reflects on the concept that every wheel of the vehicle travels at distances that have subtle changes during a turn, and the wheels inside tend to travel a shorter distance than the wheels outside. Since the speed is equal to the distance travelled divided by the time it takes to go that distance, the distance travelled by the wheel is estimated to be shorter at a lower speed. Also, the wheels in the front often travel at a different distance than those of the wheels at the rear end.

If the car does not have a differential drive, the wheels get locked together, forcing them to spin at the same speed. This becomes tedious for the car to make distinct turns and even if the car makes the turn, there wlould be a consequence where one tire would definitely slip. Although new range tyres with different properties and the concrete roads with varied gradients have made it less impossible for the force applied to make the tire slip. This existing amount of force would have to be transmitted through the car's axle from one wheel to another, putting a heavy strain on the components of the axle.

Thus, the differential is that device that splits the engine torque into two different ways, allowing each output obtained to spin at different speeds. This differential is found on all modern day cars, trucks and all-wheel-drive(full-time four-wheel-drive) vehicles as well. But, the all-wheel-drive vehicles seem to need a differential between each set of drive wheels. After analysis and experiments it is seen that they require one differential between the front and the back wheels as well, because the front wheels travel at different distances through a turn than that of the rear wheels.

In the following section of this paper, the concept of differential drive has been implemented using the PID and ANN controllers to bring out a comparison with the performance aspects of both PID and ANN.



MODEL Figure B-flow diagram for the training of neural network.

4 EXPERIMENTAL ANALYSIS

4.1 Replacement of PID with ANN

The complex input (v) is passed to the PID controller and results are noted. ANN controller is trained by considering the PID inputs and output. Training algorithm depends on the NARX recurrent network which is an alternative RNN (recurrent neural network) consisting of a single input, single output, and a delay on the inputs. The outputs are fed back to the input by the delay process. It is particularly effective for Time Series Prediction. Neural network is implemented in the Simulink to check the results. The training can be varied with the number of hidden neurons and delay. Here the delay is chosen to be 4 and the hidden neurons is set to 3.

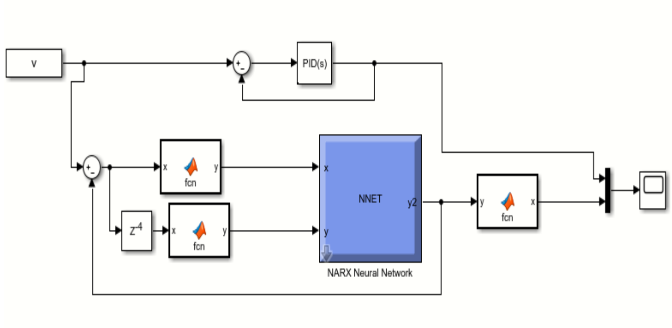


Figure 1 - SIMULINK model representing the complex input with PID and ANN controllers

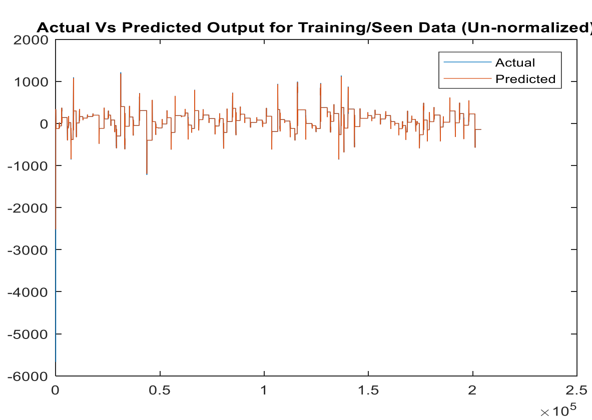


Figure 1a- Results showing the training of ANN (predicted) controller and PID (actual) controller.

4.1.1 Simulation effects

The above results were tested for different fixed size step, delay, activation function, to find the best among each of them.

4.1.1.1 Effect of fixed step size on Performance Accuracy (10 hidden neurons)

The training is performed with the obtained inputs and output from the PID controller. With the following Training Parameters: Tansig hidden layer, maximum epoch 150, stopping criteria epoch, minimum gradient, only the maximum epoch is changed everything else stays default. Such as input delays=1:2; feedback delays=1:2; hidden layer Size 10; with smaller step size with more data and less error. Worst performance is shown in bold.

Fixed Step Size	Training Error	Test Error
0.001	2.4116e-06	1.9052e-06
0.01	2.3287e-06	1.8463e-06
0.1	7.4570e-05	6.1457e-05
0.2	**2.8739e-04**	**3.4440e-04**

Table 1 - Performance result table with 10 hidden neurons

4.1.1.2 Effect of fixed step size on Performance accuracy (3 hidden neurons)

Same training parameters as the above was considered and the results were obtained.

Fixed Step Size	Training Error	Test Error
0.001	2.5779e-06	2.0565e-06
0.01	2.5016e-06	1.9960e-06
0.1	9.5324e-05	7.7856e-05
0.2	**3.6994e-04**	**4.0580e-04**

Table 2 - Performance result table with 3 hidden neurons

From the results obtained, we can conclude that the fixed step size of 0.01 will give the better performance and the minimum allowed number of neurons in the hidden layer is 3.

4.1.1.3 Delay effects

Effect of delay on the accuracy of neural network replacement of PID in DC motor speed control. Using the Fixed step size obtained in 4.1.1.2. and the minimum hidden layer of 3 neurons, the following results are obtained by varying the delay, all other parameters are kept constant.

Delay	Training Error	Test Error
2	2.5016e-06	1.9960e-06
3	2.5152e-06	2.0026e-06
6	2.4874e-06	1.9868e-06
9	2.4872e-06	1.9871e-06
12	2.4856e-06	1.9859e-06

Table 3 - Results of Delay effects

From the observations obtained, as the delay increases, the accuracy gets better even though the difference is small. But, the higher the value of network delay, the more "complex" the architecture.

4.1.1.4 Effect of activation function on the performance and convergence Speed

The experiment is formed to find the best Activation function. Using the fixed size obtained in 4.1.1.2, delay from 4..1.1.3, tested for three different hidden layers(3,5,10) ,The following results are obtained by using seven different architectures.

Number of Neurons	**3**	
Function	**Epoch**	**Test Error**
ElliotSig	800	2.0010e-04
SQNL	761.69	1.9967e-04
Tansig	799.8	1.9968e-04
Relu	124.35	4.5861e-04
SQLU	674.57	2.0031e-04
ELU	681.81	2.0030e-04
Leaky Relu	132.06	3.2981e-04

Table 4 - Results of effects of Activation functions with 3 hidden neurons.

Number of Neurons	**4**	
Function	**Epoch**	**Test Error**
ElliotSig	800	2.0004e-04
SQNL	776.47	1.9951e-04
Tansig	800	1.9921e-04
Relu	132.23	3.7159e-04
SQLU	771.14	2.0023e-04
ELU	776.21	2.0021e-04
Leaky Relu	128.06	3.7556e-04

Table 5 - Results of effects of Activation functions with 4 hidden neurons.

Number of Neurons	**10**	
Function	**Epoch**	**Test Error**
ElliotSig	800	1.3097e-04
SQNL	799.56	1.1006e-04
Tansig	800	1.0005e-04
Relu	245.62	2.5861e-04
SQLU	800	1.2006e-04
ELU	798.65	1.4004e-04
Leaky Relu	376.4	2.2994e-04

Table 6 - Results of effects of Activation functions with 10 hidden neurons.

From the observation it can be noted that Tansig function has less error when compared to the other activation functions. But higher the value of hidden layer, lower the accuracy.

4.2 *Using simple input (Step input)*

In Simulink, the DC motor is implemented with step input and the output is recorded. With the same step input the DC motor is implemented with PID controller, and the PID is tuned to get the expected results. The input and the output of PID is noted, as the main aim of this study is to replace the PID with the ANN controller. NARX (nonlinear autoregressive exogenous model) recurrent network is used to train the neural network. The training is started by providing the inputs and output of the

PID controller. The number of the hidden layers and the delay can be changed depending upon the result obtained. The training procedure is repeated by the changing the number of the hidden neuron and the delay. The output can be measured in the Regression graph. This can be showed by the following but for this Step input the number of the hidden layer is set to 3 neuron and delay is set to 3. After training the neural network controller is simulated in Simulink.

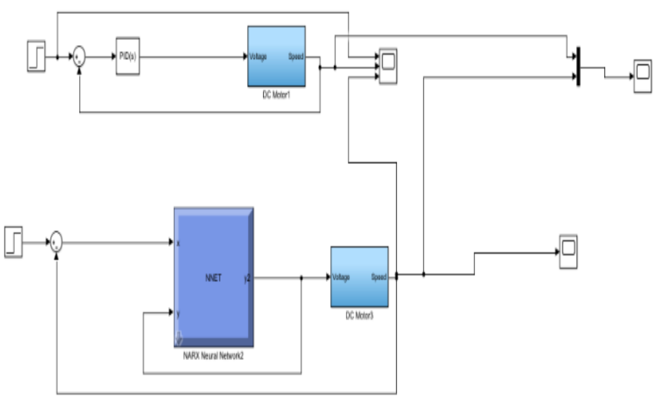


Figure 2 - SIMULINK model representing the step input with PID and ANN controllers with DC motor

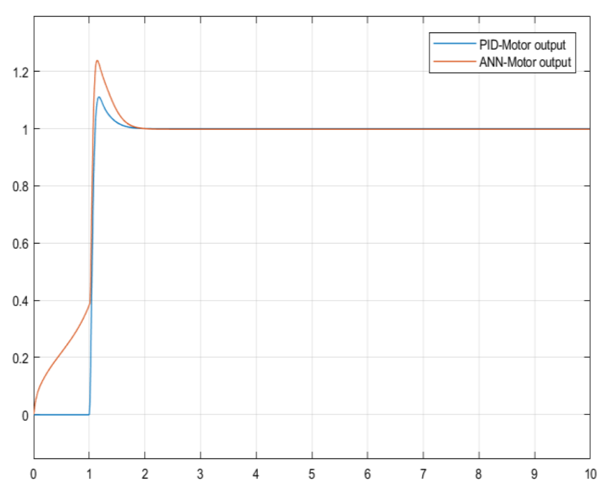


Figure 2a –Result showing the training of ANN controller and PID controller with DC motor for step input

4.3 Using complex input

A complex input is generated with discontinuous amplitude and phase, which is sent as an input to the DC motor. The output is tuned using PID until the expected results are obtained. The ANN is trained with the PID inputs and output which produces the same result as the PID.

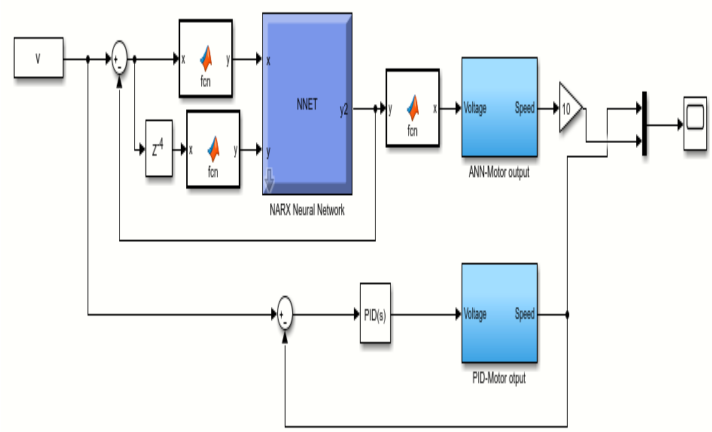


Figure 3 - SIMULINK model representing the complex input with PID and ANN controllers with DC motor

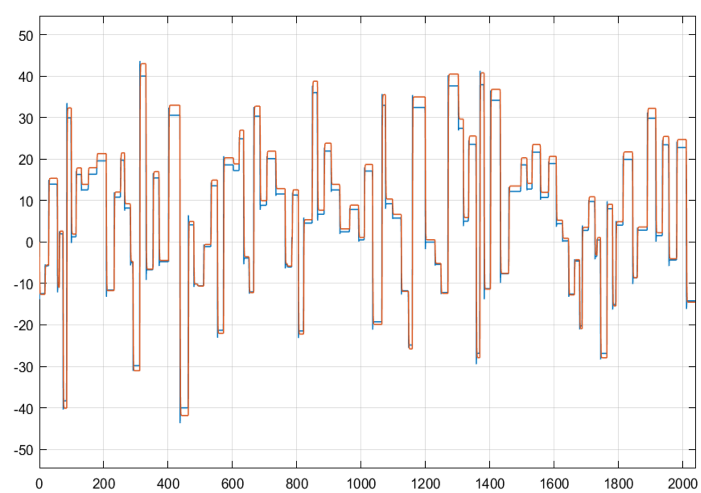


Figure 3a –Result showing the training of ANN controller (blue) and PID controller with DC motor for step input (red).

4.4 Differential Drive

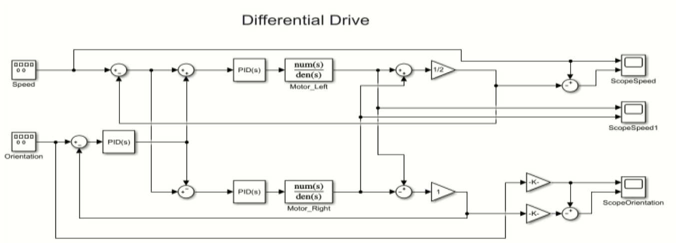


Figure 4a - SIMULINK model representing the differential drive with PID controllers.

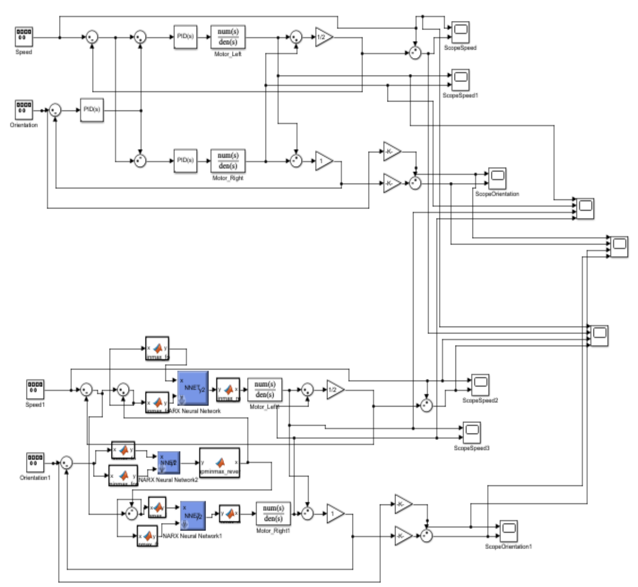


Figure 4b - SIMULINK model representing the differential drive with PID and ANN controllers.

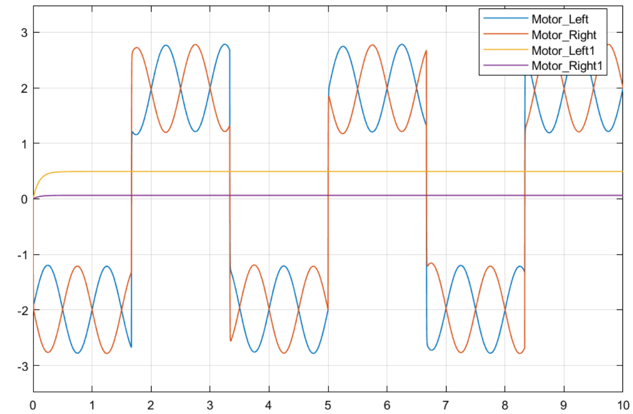


Figure 4c – Results showing speed the wheel with ANN controller and PID controller.

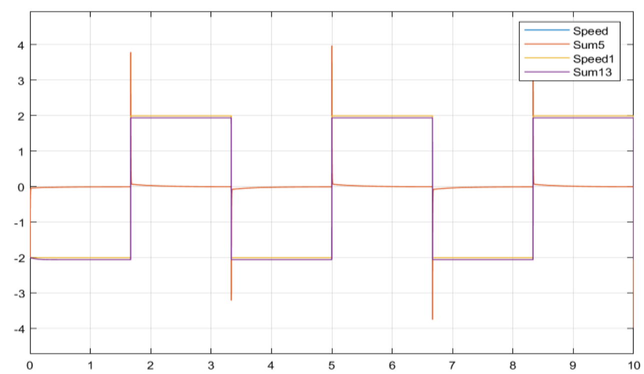


Figure 4d – Results showing speed the wheel with ANN controller and PID controller

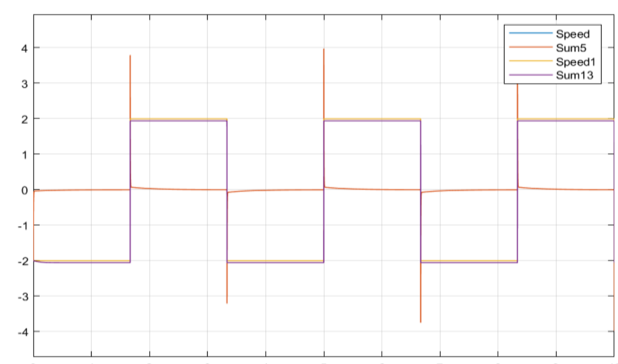


Figure 4e – Results showing orientation of the wheel with ANN controller and PID controller.

From the above graphs, it can be identified that ANN controller failed to replace the PID controller as the differential drive and it is a complex process. A further study is required to compute the differential drive with the ANN controller.

4.5 Comparative analysis of PID and ANN

Effect of PID and ANN with different types of Input.

4.5.1 Test results 1

STEP and SINE input (continuous input with the same time-period and phase)

- Case 1 –amplitude is set to 1
- Case 2-amplitude is set to 20
- Case 3-amplitude is set to 100

Considering the cases, PID is tuned for the case 1, which gives the result as the input. The same tuned PID is used for rest all cases and it gives the necessary result. Same results are obtained when the input is passed to ANN controller. From this we can conclude that PID is not dependent with the change in amplitude with the same time period and phase.

4.5.2 Test results 2

Random signal (discrete input with same time-period and phase)

- Case 1-amplitude is set between -1 and 1
- Case 2-amplitude is set between -2 and 1
- Case 3-amplitude is set between -3 and 1

Considering the cases, the tuned PID of the 1st test is used to check the result. The PID produces a bad result. Whereas the ANN produces a good result than PID. We can conclude that PID fails to produce good results, when the input has different amplitude variation with the standard tuned PID.
As most plants do not have a standard method, practices, training, or tools for loop tuning. Many plants still “tune by feel”. This leads to inconsistent and often terrible tuning results.
The effects of poor tuning are:
• Sluggish loops do not respond to upsets, causing disturbances to propagate
• Overly-aggressive loops oscillate, creating new disturbances.
Operators put the loops in manual. The loops are unable to respond, increasing the risk of safety, environmental, and quality incidents.

4.5.3 Speed analysis with different DC motor parameters

The replacement of PID controller with ANN controller is tested for the different motor parameters, and tested for the various types of inputs, test were conducted for various other scenarios and the best accurate results were chosen. It was also inferred that the results obtained were aligned to the expected values.

5 DISCUSSION

The use of PID as a controller in control system applications is ubiquitous. Although the theory behind PID is really simple but the design and implementation of PID controllers are known to be time consuming and difficult. Moreover, PID are linear systems and can only learn a particular plant based on their predefined settings. NN on the other hand are characterised with learning opportunities and adaptability in nature. This means that a NN based controller is robust to several input types and even maybe different plant architectures as will be proved in this work.

6 CONCLUSION

This paper is a result of proof that the PID Controller can be replaced with an ANN controller to increase the performance and the speed associated with the DC motor. Firstly, the input was directly passed through the DC motor and the corresponding results were obtained. Furthermore, to obtain better results, PID controllers were used to boost the speed of the DC motor and the results that were obtained were seen to be the expected ones. But, there were coinciding issues with the usage of a PID Controller. The parameters associated with the PID, i.e., Proportion, Integration and the Derivatives were fixed and there were issues related to the tuning of the PID controller as well. Also, the robustness of the controller was seen to be lesser than other controllers. To combat the disadvantages of the PID Controller, ANN controllers were used. Experiments were conducted to find the acceptable parameters associated with the ANN such as the hidden layers, delays, step sizes and activation functions and the best of the properties were found and chosen through the changes in the parameters that were related with each function. From the experiments conducted, the results were proven to be effective and robust. It was also seen that the results generated using the ANN controllers seemed to align to an extent with some errors that were negligible and acceptable. Thus, it was inferred from this project that replacing PID controllers with ANN controllers were an effective step to improve the performance and speed of the DC motor.

7 REFERENCES

- **Aamir, M. (2013). On replacing PID controller with ANN controller for DC motor position control. arXiv Preprint arXiv:1312.0148,**
- **Alhanjouri, M.Speed control of DC motor using artificial neural network.**
- **Atri, A., & Ilyas, M. (2012). Speed control of DC motor using neural network configuration. International Journal of Advanced Research in Computer Science and Software Engineering, 2(5)**
- **Cheon, K., Kim, J., Hamadache, M., & Lee, D. (2015). On replacing PID controller with deep learning controller for DC motor system. Journal of Automation and Control Engineering Vol, 3(6)**
- **George, M. (2008). Speed control of separately excited DC motor. American Journal of Applied Sciences, 5(3), 227-233.**
- **Solanki, S. (2016). Brushless DC motor drive during speed regulation with artificial neural network controller. International Journal of Engineering Research and Applications, 6(6), 1.**
- **Tanaka, M., & Okutomi, M. (2014). (2014). A novel inference of a restricted boltzmann machine. Paper presented at the Pattern Recognition (ICPR), 2014 22nd International Conference On, 1526-1531.**
- **Dorf, R. C. and R. H. Bishop (2011). Modern control systems, Pearson.**
- **Huang, G. and S. Lee (2008). PC-based PID speed control in DC motor. Audio, Language and Image Processing, 2008. ICALIP 2008. International Conference on, IEEE.**
- **Xue, D., et al. (2006). Fractional order PID control of a DC-motor with elastic shaft: a case study. American Control Conference, 2006, IEEE.**