



FPGA Implementation of Modified Reconfigurable Adaptive Gain Scheduling Controller

Ali R. Wheed^{a*}, Abbas H. Issa^b, Mohammed Y. Hassan^c

^{a*} Ministry of Youth and Sport, Baghdad, Iraq, eng.ali_elshemary@yahoo.com

^b University of Technology, Department of Electrical Eng., Baghdad, Iraq, 30050@uotechnology.edu.iq

^c University of Technology, Control and Systems Engineering Department, Baghdad, Iraq
60003@uotechnology.edu.iq

*Corresponding author.

Submitted: 18/07/2020

Accepted: 23/10/2020

Published: 25/07/2021

KEY WORDS

Adaptive controller, Dc-motor, FPGA, Gain scheduling, Inverse Neural, Reconfigurable.

ABSTRACT

This article aims to put forward a modified type of adaptive gain scheduling that will be able to deal with the immeasurable and unpredictable variations of system variables by adapting its value at each time instance to follow any change in the input and overcome any disturbance applied to the system without the need to predetermine gains values. In addition, the inverse neural controller will precede the gain scheduling to eliminate the need for complex system linear zing and parameter estimation. Therefore, the problems of needing complex mathematics for system linearization and gains calculations have been solved. The performance of the presented controller was tested by comparing the step response of a DC-motor controlled via the proposed technique and the response of that motor when controlled by the inverse neural controller and PID controller. MATLAB/Simulink has been used for making the simulations and obtaining the results. In addition, the FPGA implementation of the proposed controller has been presented. The results showed a remarkable improvement in the transient response of the system for all of the rising time, delay time, settling time, peak overshoot, and steady-state error.

How to cite this article: A. R. Waheed, A. H. Issa, M. Y. Hassan, "FPGA Implementation of Modified Reconfigurable Adaptive Gain Scheduling Controller," Engineering and Technology Journal, Vol. 39, No. 07, pp. 1105-1116, 2021.

DOI: <https://doi.org/10.30684/etj.v39i7.1772>

This is an open access article under the CC BY 4.0 license <http://creativecommons.org/licenses/by/4.0>

1. INTRODUCTION

In the early 1950s, when the design of the aircraft autopilot become true the thinking of designing new controllers has been animated. Moreover, the engineers competed to create a new controller that capable of dealing with many uncertainties and handling many variations in the parameters of the system. So a great deal of interest focused on the adaptive controllers for many reasons,

unpredictability, unexpected and sudden disturbances and faults, complex nonlinear systems, unknown system parameters [1].

One type of adaptive controller is gain scheduling, which is used for controlling the systems of time-varying. Originally, gain scheduling was firstly used for flight control, and then it witnessed increasing use in process control [2]. Therefore, a large number of gain scheduling techniques were developed, and the following are some of them:

Christos M., et al. (2010), proposed two advanced proportional controllers, the first is an integral-controller, and the second is a differential-controller used to control the speed and position of a switched-reluctance DC-motor. A filter (low-pass) has been added to the differential-controller to enhance reference-point tracking [3].

Fernando R., et al. (2011), a combination of the neural model for DC-servomotor and the neural-network controller, has been used to obtain the position control of the servomotor. A precisely defined mathematical model has needed to obtain the neural model that will be in the simulations [4].

Michal B., (2014), presented an LQR gain scheduling controller for a PMSM (Permanent Magnet Synchronous Motor) and replaced the system of PI controllers in the FOC by a feedback controller of multivariable state, and tuned the gain scheduling according to the changes in the operating circumstances. The system synthesized using the optimal control (LQ) method [5].

Kumar P., et al. (2015), investigated the high precision control in SRM (Switched Reluctance Motor) by adopting a gain scheduling PI-controllers for speed control and PD-Controller for position control. In addition, a filter (Low-Pass) is combined with the proportional part of the PD-Controller so that the impulse response created by the Pi-Controller will be suppressed [6].

Josaphat P., et al. (2017), designed a controller using Fuzzy-Gain scheduling to control the DC-Motor speed of a CNC-machine. The controller is used to overcome the load changes that may change the speed-response. The (PI), gain scheduling parameters are changed by the fuzzy controller to follow the load change [7].

Suppachai H., et al. (2018), presented a PSO-PID gain scheduling for control a water level. The authors used a Mitsubishi- PLC and monitoring pressure sensors with the SCADA-Omron package. Communication between the field and the front end is made by a built-in Ethernet module [8].

Lucky P., et al. (2019), submitted a strategy for control the speed of the IM induction motor through an FPGP PID controller (Fuzzy Prediction-Gain Proportional). The gain scheduling parameters (gains values) are modified by the fuzzy controller rule if there is any change in the IM speed to achieve optimal control parameters [9].

As could be seen from the above literature, the gain scheduling values need to be pre-calculated during the design stage, which means complex mathematical steps will be used for system linearization, and then determining the gain values (using fuzzy, ...etc.). This means a well knowing of the operation environment and disturbance prediction must be available.

Our proposed method is an attempt to design a gain scheduling adaptive controller that has the following features:

- 1) 1. Avoid the need for system linearization and parameter estimation steps using the plant-inverse neural network controller (through training the neural network to produce the inverse of the plant).
- 2) 2. No pre-determined fixed gain values (the gain values will be adapted continually depending on the error and output values).
- 3) 3. The transient response parameters of the proposed controller should be better than those for the PID controller and Inverse neural controller (for the same controlled plant).

In addition to the introduction in section One, the architecture of the paper consists of; Motor modeling in section Two, Methodology in section Three, simulations and results in section Four, the controller implementation to the FPGA in section Five, and the conclusions in section six.

2. MOTOR MODELING

The DC motor is regarded as a common actuator in systems control. It produces direct rotational motion and could be coupled with drums, wheels, or and cables [10]. The equivalent electric circuit of the armature and the rotor free-body diagram is shown in Figure 1.

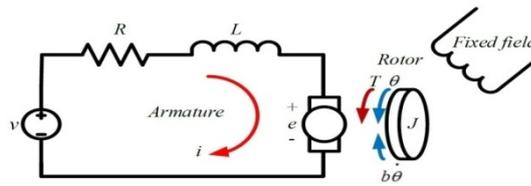


Figure 1: DC motor equivalent circuit of the [8]

The mathematical derivation of the DC motor is done according to the back induced-electromotive force equation and the torque equation, and depending on the fact that 'the produced torque by the motor proportion directly to the current of the armature by a factor K_t (regarding a constant magnetic field). As explained in Eq. (1) [10]:

$$T = K_{(t)}I \tag{1}$$

The TF of the speed $\omega(s)$ shown in Eq. (2):

$$\frac{\omega(s)}{V(s)} = \frac{K}{[(Ls + R)(Js + b) + K^2]} \tag{2}$$

Where:

- J is the Inertia in Kilogram. Meter.
- b is the Damping-ratio in Newton. Meter. Second.
- L is the rotor Inductance in milli-Henry.
- R is the Resistance in ohms.
- V is the Input voltage in volts.

The DC-Motor designed according to Eq. (1) and Eq. (2) shown in Figure 2, below:

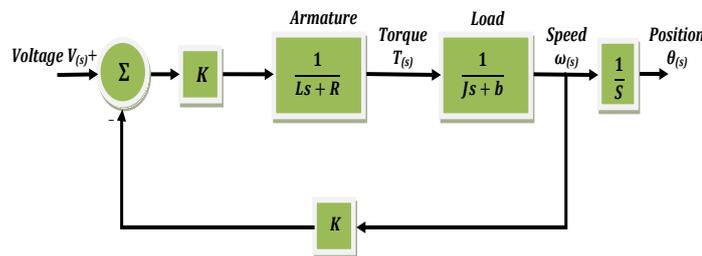


Figure 2: DC-Motor design using TF

The DC-motor simulation accomplished by the parameters listed in Table I below:

TABLE I: Simulation-Parameters of DC-motor [11]

Parameter	Amount,
Inductance(mH)	2.29
Resistance(Ω)	5.61
Torque Constant(Nm/A)	0.16
Damping factor(Nms)	5.0526 e – 2
Moments of inertia(kg. m2)	0.4366

3. NEURAL NETWORK

Feed-forward multilayer neural network training by the back-propagation method is regarded to have a very general nature. So, it could use to solve different problems in many areas. Such nets are found in several fields that have problems in mapping the given input-set to the target output-set. A multilayer artificial neural network that has one hidden layer (Z-layer) is shown in Figure (3). The bias terms in the hidden layers' output stand for the connection weights of units that constantly output one. Through the back-propagation learning phase, the signals will be sent in the reverse direction [12].

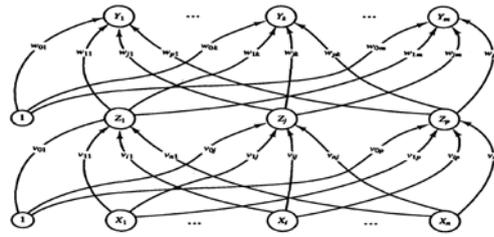


Figure 3: One hidden-layer back-propagation neural network [12]

The following equations show the operation of the back-propagation training method [12]:

x : is the training vector (input)

y : is the target vector (output)

For the hidden-layer (Z-layer), the input given by:

$$Z - in_j = v_{0j} + \sum X_i v_{ij} \tag{3}$$

the output is given by:

$$Z_j = f(Z - in_j) \tag{4}$$

For the output-layer (Y-layer), the input given by:

$$y - in_k = w_{0k} + \sum Z_j w_{jk} \tag{5}$$

the output is given by:

$$y_k = f(y - in_k) \tag{6}$$

In our work, the inverse of the plant was obtained by training the neural network using the back-propagation method by given sets of input-output pairs. Therefore, the generated neural network will give the inverse response of the original plant, as will be explained in section 4.

4. METHODOLOGY

I. Inverse neural controller

The capability of the inverse neural to obtain the linearization of nonlinear systems and the structure simplicity made it easy to implement in engineering. In addition, it is very beneficial to apply the Artificial Neural Networks (ANN) for system-control and identification for the account of the neural: massive parallelism, capable of learning, quick adaptation, and the ability of inherent-approximation [10].

So, by integrating the nonlinear-approximation ability of the neural network with the inverse system approach (generating plant-inverse ($\frac{1}{G(s)}$), by utilizing neural networks), the algorithmic problems of determining inverse systems could be avoided. Hence, the original nonlinear system control exchanged for two integrated linear subsystems feed-forward control [13].

Figure 4 explains the block diagram of a plant controlled by the inverse-plant neural controller:

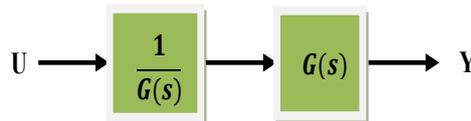


Figure 4: Inverse control

Where:

U and Y are the input and the output, respectively.

$G(s)$ is the plant TF.

$\frac{1}{G(s)}$ The inverse of the plant (obtained by training the neural network to give the inverse response of the plant)

The idea is to obtain a complete system (plant + controller) whose transfer function is as given by Equation (3):

$$\frac{Y}{U} = G(s) \times \frac{1}{G(s)} = 1 \quad (7)$$

This means the output will follow the change of input identically. However, because of technical and mathematical issues, it is not possible to create a system that exactly inverts the original one, so Eq. (7) will not be true and $\frac{Y}{U} \neq 1$. That means errors will appear in the system transient response like overshoot, slow rising time, and large steady-state error.

To overcome this problem, an adaptive-gain factor acts as adaptive-gain scheduling used in this work.

II. Adaptive Gain Scheduling

The gain scheduling has been chosen because it is a special type of nonlinear feedback. It has a linear controller with changeable parameters that will update according to the conditions of operating. Another advantage of gain scheduling is the fast parameters change responding to the change in the process [14].

The presented idea is to use the inverse-plant neural controller in cascade with a modified adaptive gain scheduling, as explained in Figure 5.

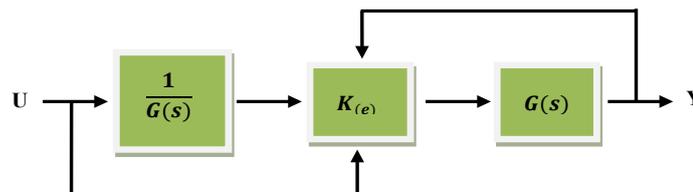


Figure 5: Proposed control method block diagram

From Figure 4, the adaptive gain $K(e)$ connected in cascade with the inverse controller ($\frac{1}{G(s)}$), is a function of error (U-Y). Therefore it will adapt its value continually and let the output tracking the input by clearing the error in the steady-state region. Therefore, no offset between the system response and the input. Equation (8) explains the way of how the gain scheduling value will be adapted:

$$K(e) = Y + [U \times (\frac{1}{G(s)}) \times e] \quad (8)$$

As could be noted, the proposed controller does not include the insertion of an additional system (no poles and no zeros added). Therefore, the characteristic equation of the original system will not change. The only change is the effect of the additional gain and this will be judged using simulation because it is known that for gain scheduling: "The performance and the stability of the system could be evaluated typically by simulation"[14].

5. SIMULATION AND RESULTS

Simulation designs and results obtaining have done using MATLAB/Simulink. The simulation comprises modeling of the PID controller, modeling the neural controller of the inverse-plant, also the model of the proposed control method, and comparing their results in controlling the same plant (the same DC-motor).

I. Step response comparison between PID and proposed controller

The following is a comparison of DC-motor step response first when controlled by a PID controller and the second when controlled by the proposed control technique. The PID controller parameters were tuned using MATLAB/Simulink auto-tune, and the step response of both controllers

obtained for the same plant and the same input (step signal). Figures 6-7, show the PID-controlled system and the proposed method, respectively.

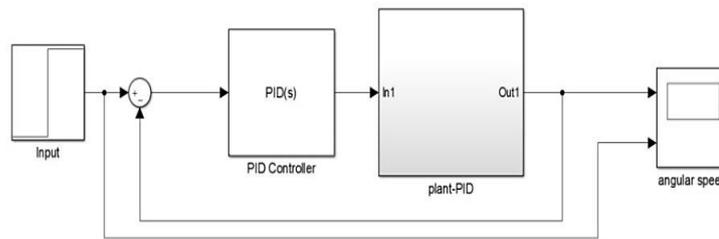


Figure 6: System using PID-controller

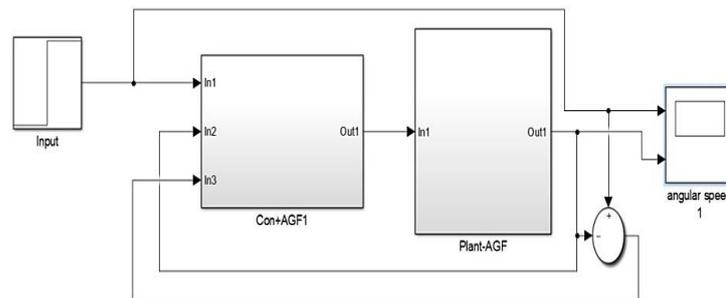


Figure 7: System using Proposed-method

Figure 8, shows the responses for both systems.

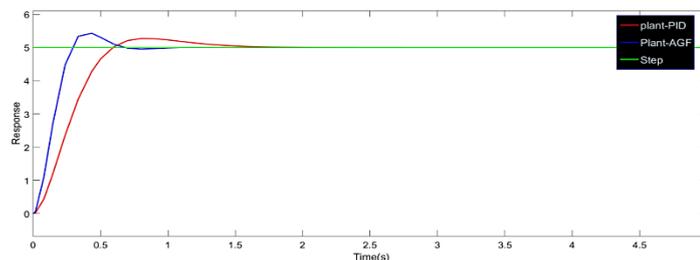


Figure 8: System step response of PID and the proposed controller

From Figure 8, it is evident that the response using the proposed technique is much better than the PID response, for having less rising time, less peaking time, less settling time, but it has a little higher overshoot. Even though for the proposed system response, the peak overshoot is greater than that for the PID response but it is still regarded as a better response because the difference in overshoots is only 0.156, and less: rising, delay, peak, and settling times. This leads to the fact the step response (of the plant) when controlled via conventional PID is much slower than if it is controlled by the proposed controller. Table II Summarize the results.

TABLE II: PID-AGF transient response results summary

Response Parameter	PID	AGF
Rising Time (sec)	0.4	0.2
Peak Time (sec)	0.855	0.405
Overshoot	0.274	0.432
Settling Time (sec)	1.975	0.725

II. . Step response of inverse controller and proposed controller

This comparison is to certify the enhancement achieved by the proposed control method over the conventional inverse-plant neural controller. Figure 9 is the block diagram of the conventional inverse neural controller.

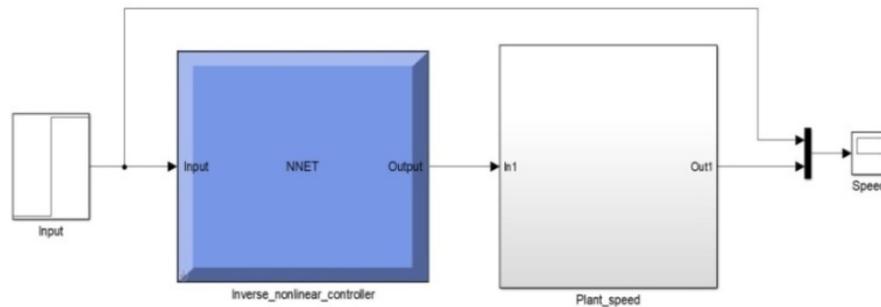


Figure 9: The conventional inverse neural controller

The inverse neural controller is composed of one neural network it is a single input-single output with three hidden layers. All of the layers have a purlin activation function. The network training made using the back-propagation method to obtain the plant inverse response. The training process made with MATLAB/M-file, and the instruction genism(net) used to generate the Simulink block diagram of the network. Figure 10 shows the architecture of the neural network.

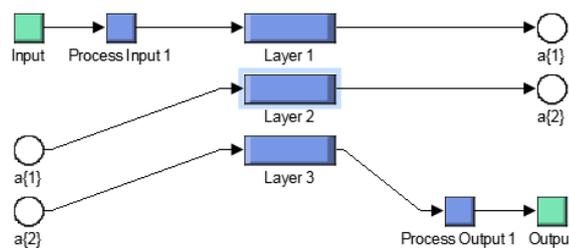


Figure 10: Neural network architecture

The step response using the two controllers in the control of the same plant is shown in Figure 11, strengthens the results in Figure 8, and proves the superiority of the proposed method because it uses the same inverse neural controller with the addition of only the adaptive gain scheduling.

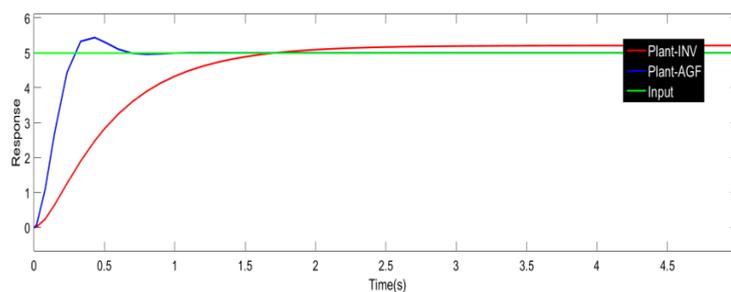


Figure 11: System response using both Conventional Inverse and Proposed method

The proposed method again has less rising, peak, settling times, and a little higher overshoot. Again, it will reach the steady-state condition faster than the conventional inverse neural controller and approximately with no steady-state error, as illustrated in Table III.

TABLE III: INV. Neural-AGF transient response results summary.

Response Parameter	INV. Neural	AGF
Rising Time (s)	1.7	0.2
Delay Time (s)	0.43	0.14
Max-Overshoot	0.2	0.432

Settling Time (s)	2.5	0.725
-------------------	-----	-------

6. FPGA IMPLEMENTATION

To ensure that; the simulated proposed control method is implementable in real processing cards, it is implemented to the FPGA-SPARTAN 3 by using ISE Design suit 14.7 and system generator. The FPGA is meant to work in a loop with MATLAB/Simulink and the results obtained online.

The implementation has been accomplished with 12% of LUT's and 13% of occupied slices available by the FPGA device. In addition, the total supplied power to the FPGA is just 0.329 watt. Figures 12-13, respectively, showing the device utilization summary and the power consideration of the FPGA.

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of 4 input LUTs	6,052	47,744	12%	
Number of occupied Slices	3,297	23,872	13%	
Number of Slices containing only related logic	3,297	3,297	100%	
Number of Slices containing unrelated logic	0	3,297	0%	
Total Number of 4 input LUTs	6,535	47,744	13%	
Number used as logic	6,052			
Number used as a route-thru	483			
Number of bonded IOBs	160	469	34%	
Number of DSP48As	122	126	96%	
Average Fanout of Non-Clock Nets	1.93			

Figure 12: FPGA device utilization summary

Device		On-Chip	Power (W)	Used	Available	Utilization (%)	Supply Summary		
Logic		Logic	0.003	6428	47744	13	Source	Voltage	Total
Family	Spartan3adep	Signals	0.007	14578	—	—	Vccint	1.200	0.078
Part	xc3ad3400a	DSPs	0.001	122	126	97	Vccaux	2.500	0.042
Package	fg676	IOs	0.140	160	469	34	Vcco25	2.500	0.053
Temp Grade	Commercial	Leakage	0.181						
Process	Typical	Total	0.329						
Speed Grade	-4								
Environment		Thermal Properties		Effective TjA	Max Ambient	Junction Temp	Supply Power (W)		
Ambient Temp (C)	25.0	(C/W)		14.7	80.2	29.8	Total	Dynamic	Quiescent
Use custom TjA?	No	(C)					0.329	0.148	0.181
Custom TjA (C/W)	NA								
Airflow (LFM)	0								
Characterization									
PRODUCTION	v1.1.05-25-09								

Figure 13: Power considerations of the FPGA device

The block diagram of the system is shown in Figure 10. Also, the FPGA in the operation state (LED on) is illustrated in Figure 14.

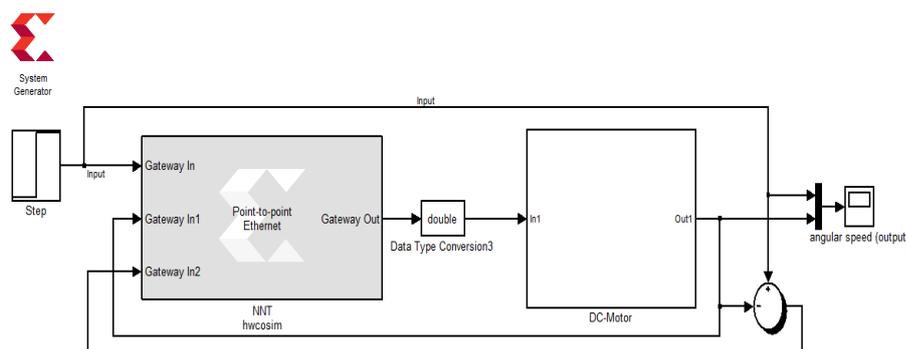


Figure 14: System with FPGA in the loop



Figure 15: FPGA-Card (LED is ON)

I. VARIABLE INPUT TESTS

The system step-response shown in Figure 16 declares that the controller was implemented successfully, and the results match the simulated controller results that prove the ability to use the proposed control method in controlling real and online systems (It is reliable).

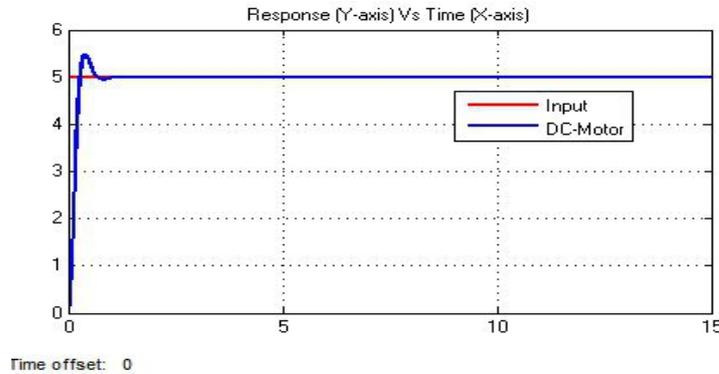


Figure 16: Step - response of the system with FPGA in a loop

For more emphasis on the reliability and robustness of the proposed control method, another input of repeated sequence was used to test the system response. Figure 17 shows that the proposed control method is capable of tracking the repeated sequence input in a fast, reliable, and robust manner with minimum overshoot and minimum response time.

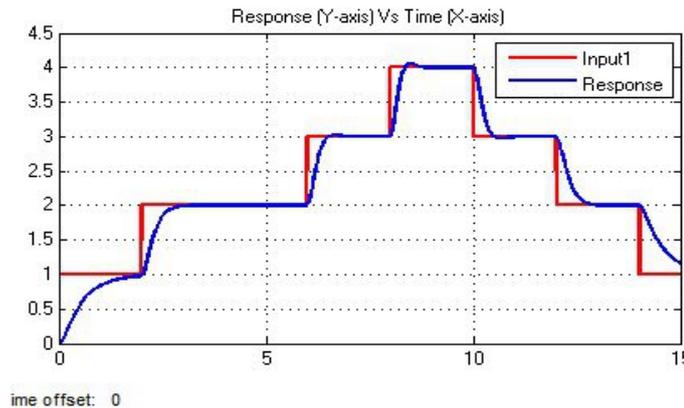


Figure 17: System-FPGA response for repeated sequence input

Another input is tested that is not stepped and of variable time intervals used to examine the proposed system's response to the continuous change in referenced input, as shown in Figure 18.

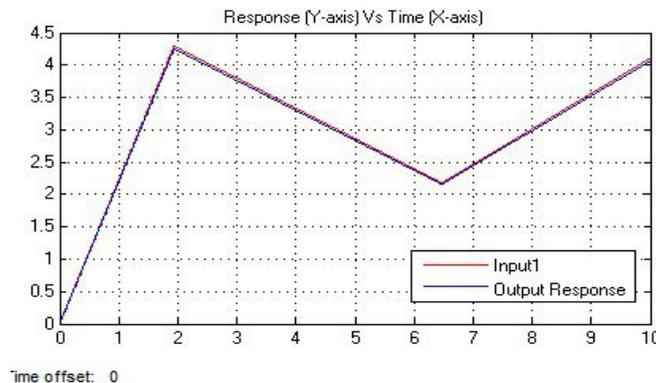


Figure 18: System-FPGA responses to continuously changing input.

From Figure 18, it is clear that the output tracked the input change at each time instant. The tracking error of which the output follows the referenced input is approximately zero. There is no

overshoot and no rising time because the input is not rapidly changed from 0 to 5 (as in the step input test). In other words, the value of error needed to be handled by the controller is less.

II. . OUTPUT DISTURBANCE TEST

The following Figures 19-20 show the system response to a unity impulse of 2 seconds duration added at the output of the motor to simulate an opposite sudden output disturbance.

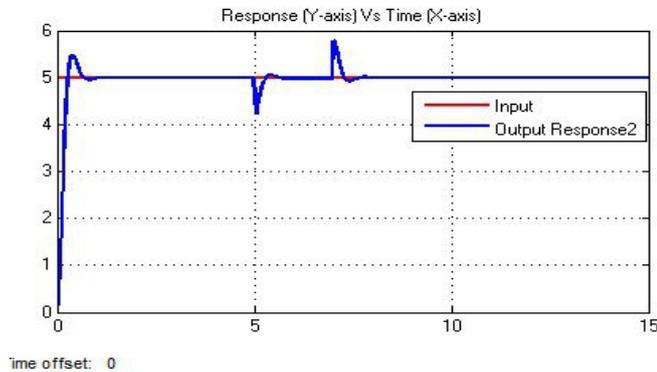


Figure 19: System responses to a negative output disturbance

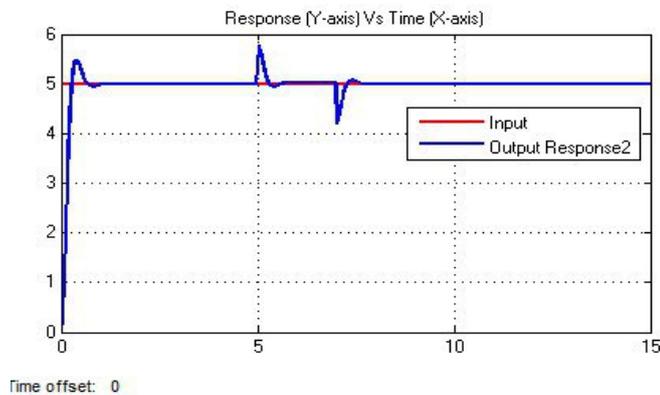


Figure 20: System responses to positive disturbance

From Figures 19 - 20, it is obvious that the system response does not fall to the entire value of the disturbance, and there are no oscillations during the duration of the disturbances (2 seconds from 5th-7th sec.). Moreover, the system returned to its steady-state value in 0.4 seconds after the disturbance vanishing. Therefore, it will be clear that the system is robust in handling disturbances that may happen suddenly as a load change or as an operating environment variation.

7. COMPARISON TO PREVIOUS RESEARCHES

The comparison of our work to some of the previous researches listed in Table IV. Because the specifications of the DC-motors are different in each research and this may affect the accuracy of comparison, the following steps used:

- 1) Compare the transient response parameters of our proposed technique to the parameters of our PID and obtaining the percentage enhancement to each transient parameter.
- 2) Repeat step (1) for each of the other researches and their PIDs (obtaining their percentage enhancements).
- 3) Summarize the results in Table IV.

TABLE IV: Comparison to some previous works

Research	Control Technique	Tr	Overshoot	Ts
Our Research	Adaptive Gain Scheduling	50%	---	63.29%
Ref [15]	RTD-A-CWO	56.45%	---	---
Ref [16]	SMC	---	100%	43.75%

Ref [17]	Fuzzy-PID	---	91.7%	28.57%
Ref [18]	AFEP	---	63.63%	42.57%
Ref [19]	FLC	---	90%	33.33%

Note: the --- sign denotes no enhancement

From Table IV, the following observations recorded:

- 1) Concerning the rising time (T_r), our proposed technique has the advantage above all of the other researches except ref [15], which has more enhancement in (T_r) but it has neither overshoot enhancement nor settling time enhancement.
- 2) Concerning the overshoot, our proposed technique has no overshoot enhancement among other researchers, but it overtaking other researches in the rising (T_r) and the settling time (T_s) enhancements, which means our controller provides a faster response.
- 3) Concerning the settling time (T_s), it is clear that our proposed technique has the lowest settling time (T_s) so, it is better than other researches.

From the above records, it is obvious that our proposed control technique provides the operation of controlling the speed of DC-motor faster than other techniques in the other researches with a very small and acceptable overshoot. In addition to that, our proposed technique requires fewer design steps, less design time, and does not require complex math-operations. All of these advantages give the lead to our proposed technique over other techniques listed in the table of comparison.

8. CONCLUSION

A prove could be concluded from the results is the proposed technique (Inverse-plant neural controller followed by a modified adaptive reconfigurable gain scheduling) has the advantages of the minimum rising, delay, peak, and settling times over the conventional inverse neural controller and the PID controller. In addition, the FPGA implementation results proved that the proposed controller is reliable to implement in real systems, robust in tracking the input changes and handling disturbances that may happen in the output.

Another conclusion is the gain scheduling successfully achieved without the need to pre-calculate the gain values because the gains continually obtained according to Eq. (8) depending on the change in the error and the output signal. This means obtaining high performance from the controlled system and demonstrating that the proposed method is capable of providing the optimal control-parameters successfully. This work performs the control operation in a simple and time-effective way and very efficient response.

References

- [1] R. Alphinias et al., Comparison of Conventional Closed Loop Controller with an Adaptive Controller for a Disturbed Thermodynamic System, IEEE, Evolving and Adaptive Intelligent Systems (EAIS), Slovenia, May 31 - June 2, 2017. DOI: [10.1109/EAIS.2017.7954841](https://doi.org/10.1109/EAIS.2017.7954841)
- [2] K. Astrom et al., Advanced PID control, Lund Institute of TechnologyLund University, Sweden, 2006. [Publication in Lund University research portal](#)
- [3] Ch. Mademlis et al., Gain-Scheduling Regulator for High-Performance Position Control of Switched Reluctance Motor Drives, IEEE Trans. Ind. Electron., 57(2010) 2922-2931. DOI: [10.1109/TIE.2009.2038400](https://doi.org/10.1109/TIE.2009.2038400)
- [4] F. Rios-Gutierrez et al., Efficient Position Control of DC Servomotor Using Back Propagation Neural Network, IEEE, Seventh International Conference on Natural Computation, 2011, 653-657. DOI: [10.1109/ICNC.2011.6022230](https://doi.org/10.1109/ICNC.2011.6022230)
- [5] M. Brasel, A gain-scheduled multivariable LQR controller for Permanent Magnet Synchronous Motor, IEEE, Conf., 2014, 722- 725. DOI: [10.1109/MMAR.2014.6957443](https://doi.org/10.1109/MMAR.2014.6957443)
- [6] K. Kumar et al., Performance Analysis of Position Control of Switched Reluctance Motor Drive, IEEE, IAS Joint Industrial and Commercial Power Systems / Petroleum and Chemical Industry Conference, 2015, 105-114.
- [7] J. Pramudijanto et al., Design of Fuzzy Gain Scheduling Controller for DC Motor Speed Adjustment on Spindle CNC Milling, IEEE, International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation, 2017, 127-130. DOI: [10.1109/ICAMIMIA.2017.8387571](https://doi.org/10.1109/ICAMIMIA.2017.8387571)
- [8] S. Howimanporn et al., Design of PLC for Water Level Control Employing Swarm Optimization-Based PID Gain Scheduling, IEEE, 2018, 63-67. DOI: [10.1109/ICCR.2018.8534490](https://doi.org/10.1109/ICCR.2018.8534490)

- [9] I. Ferdiansyah et al., Implementation of FPGP-PID for Dynamic Speed Control of Three Phase Induction Motor Based on IFOC, IEEE, 6th International Conference on Instrumentation, Control, and Automation (ICA) Bandung, Indonesia. 31 July – 2 August 2019, 24-29.
- [10] A. Telba et al., Motor Speed Control Using FPGA, Proceedings of the World Congress on Engineering 2014 Vol I, 2014, London, U.K.. [DOI:10.1142/9789814667364_0034](https://doi.org/10.1142/9789814667364_0034)
- [11] Data Sheet, DC Motor type DG01D-A130GEARMOTOR. Web link <https://cdn.sparkfun.com/datasheets/Robotics/DG01D.pdf>
- [12] L. Fausett et al., Fundamentals of Neural Networks, Prentice Hall; Upper Saddle River, NJ 07458.
- [13] X. Dai et al., Neural Network Inverse Synchronous Control of Two-motor Variable Frequency Speed-Regulating System, IEE, Conf. 2006, 1070-1075. [DOI: 10.1109/ICNSC.2006.1673300](https://doi.org/10.1109/ICNSC.2006.1673300)
- [14] K. Astrom et al., Adaptive Control, Prentice Hall; 2nd edition (December 31, 1994),
- [15] V. Bagyaveereswaran et al., Gain Scheduling of a Robust Setpoint Tracking Disturbance Rejection and Aggressiveness Controller for a Nonlinear Process, processes, 2019.
- [16] E. Dursun et al., Speed Control of a DC Motor with Variable Load Using Sliding Mode Control, Int. J. Comp. Electr. Eng., 8(2016). [DOI:10.17706/IJCEE.2016.8.3.219-226](https://doi.org/10.17706/IJCEE.2016.8.3.219-226)
- [17] R. Arulmozhiyal et al., Design of Fuzzy PID Controller for Brushless DC Motor, IEEE, International Conference on Computer Communication and Informatics (ICCCI -2012), 2012. [DOI: 10.1109/ICCCI.2012.6158919](https://doi.org/10.1109/ICCCI.2012.6158919)
- [18] G. Chang et al., Intelligent Controller Design for PM DC Motor Position Control Using Evolutionary Programming, IEEE, 2011. [DOI: 10.1109/CCIENG.2011.6007951](https://doi.org/10.1109/CCIENG.2011.6007951)
- [19] J. abdaltayef et al., Real –Time DC Motor Position Control by (FPID) Controllers and design (FLC)Using Labview software Simulation, IEEE, 2019. [DOI:10.1109/ICCAE.2010.5451580](https://doi.org/10.1109/ICCAE.2010.5451580)