Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 20 (2018) 219–226

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

2nd International Conference on Materials Manufacturing and Design Engineering

Implementing Fuzzy Logic Controller and PID Controller to a DC Encoder Motor – "A case of an Automated Guided Vehicle"

Priyam Parikh[a,*], Saurin Sheth[b], Dr. Rupesh Vasani[c], Jigar Kumar Gohil[d]

[a] *SAL College of Engineering, Ahmedabad 380060, India*
[b] *G.H. Patel College of Engg. & Tech, Vallabh Vidhynagar, Anand 388120, India*
[c] *SAL Educational Campus, Ahmedabad 380060, India*
[d] *G.H, Patel College of Engg.& Tech, Vallabah Vidhyanagar, Anand 388120, India*

Abstract

It becomes essential to run AGV at a constant speed or RPM in the plant to reduce downtime and lead time. Fuzzy logic controller and PID controller are applied to achieve constant RPM in AGV. AGV contains DC brushed motor with encoder, motor driver, microcontroller and battery. Encoder gives feedback to microcontroller in the form of shaft position. Microcontroller reduces error in the system based on the parameters defined by the algorithms. The first phase of the paper gives brief information about the hardware, software and the algorithms. In the second phase of the paper, methodology for implementing the algorithm to the system is shown. In the final phase of the paper, results and discussions are mentioned based on the applied algorithms. Comparison between PID controller and fuzzy PID controller is also shown. Ziegler-Nichols Algorithms is used to find PID parameters. MATLAB simulink and fuzzy logic tool box are used for simulation. Arduino Microcontroller is used to accept the feedback given by the encoder and to control the speed of motor. In a nut shell, these control strategies help AGV to run at a constant RPM with reduced settling time, steady state error and overshooting.

Peer-review under responsibility of the scientific committee of the 2nd International Conference on Materials Manufacturing and Design Engineering.

Keywords: *Arduino controller; MATLAB Simulink; Fuzzy logic controlle; AGV; Encoder; PID controller; settling time; down time; overshooting*

* Corresponding author. Tel.:+919428355356;
E-mail address: priyam.parikh@sal.edu.in

Peer-review under responsibility of the scientific committee of the 2nd International Conference on Materials Manufacturing and Design Engineering.
10.1016/j.promfg.2018.02.032

1. Introduction

The present work demonstrates the methodology for implementing PID control algorithm and Fuzzy logic control algorithm to run AGV at a constant velocity. To achieve constant velocity of AGV, speed of a DC motor is controlled using above mentioned algorithms [1]. Fig. 1 shows the block diagram of a system, in which encoder is used as a feedback element. Motor is considered as a plant and Arduino (open source controller) is used as a controller. Set point is in the form of RPM and output is in the form of shaft position of the motor [2]. PID parameters are found using Ziegler Nichols algorithm. Fuzzy logic process is divided the three parts. Fuzzy logic Block diagram is shown in fig. 4. According to the reference point, first Fuzzification takes place. Interference engine comes in the picture between Fuzzification and Defuzzification process [3-5]. 1). Fuzzification which converts the measured data (e.g. RPM of the motor is 200) into rhetorical data (e.g. motor is too slow). Fuzzy rules can be defied after the Fuzzification process gets over [4]. 2). Interference Engine which provides appropriate coherence and analysis for an output simulation. 3). Defuzzification gives output on the basis of membership function and defined rules [5][13].

Nomenclature

s	Laplace operator	ω	Natural frequency of oscillations	K	gain of the system
K_m	Motor size constant	K_b	Back EMF constant	J	Inertia of the motor
ξ	damping ratio	ω_d	damped angular velocity	K_P	positional error constant,
K_v	velocity error constant	K_a	acceleration error constant	e_{ss}	steady state error
G(s)	Transfer function	Z-N	Ziegler Nichols method	FSL	Fussy logic system
C(t)	output response	T_d	delay time	T_r	Rising time
T_p	pick time	M_p	overshoot in %,	T_s	settling time

Fig.3 shows the front view of AGV, which is used in this project. AGV has to follow only black straight path, which is detected by IR sensors. The proposed work only gives an idea about the PID and Fuzzy logic controller implemented in AGV [3]. Rests of the things are not taken in account in this work. The complete setup for implementing fuzzy and PID controller is shown in fig. 2. Two major things are to be taken in account in this work. Self weight of AGV is considered as a load and friction is neglected in the system [6].

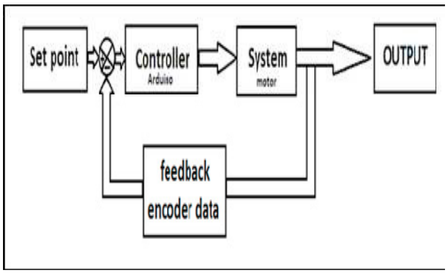

Fig. 1. Block Diagram of the PID Control system

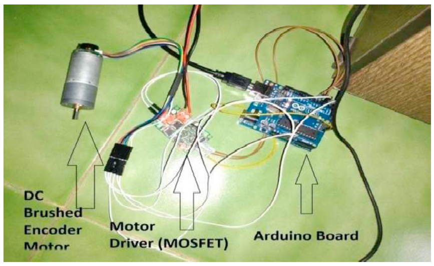

Fig. 2. Testing Setup

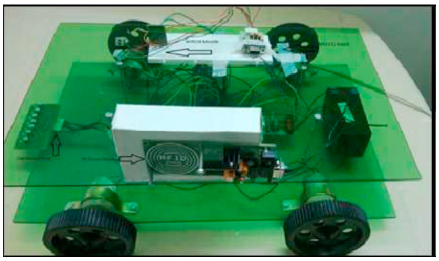

Fig. 3. Automated Guided vehicle used in this work

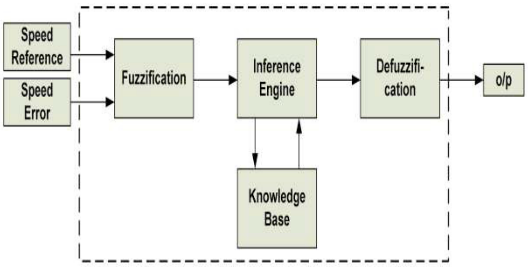

Fig. 4. Fuzzy Logic Controller Block Diagram [13]

Table 1. Specifications of AGV

Sr. No.	Parameters	Values
1	L x B x H	35cm x 25cm x 15cm
2	Mass of AGV	3.2 Kg
3	No of Motors	4
4	Types of Motors	DC brushed Encoder
5	Motor Specification	6 volts, 500 PRM, 3.6 Kg.cm at 1.2 Amp-
6	Wheel Diameter	10 cm
7	Path to be followed	Straight Dark line
8	Path Followed by	IR sensor Array
9	Motor Gear Ratio	36:1

Table 2. Specifications of a DC Encoder Motor used in AGV

Sr. No.	Motor Parameters	Symbols	Information
1	Motor Inductance	La	1 Henry
2	Armature Resistance	Ra	4Ω
3	Motor Inertia	J	$0.005 Kg.m^2$
4	Torque Constanta	Ka	0.01 N.m/Amp
5	Type of the system	See Eq 2	Type Zero
6	Order of the system	See Eq 1	Second Order
7	Type of feedback	-	Unity

Table 1 gives the full information about the AGV used in this project. Table 2 gives the full information about the Motor used in AGV.

2. Problem Description and Methodology

2.1. Problem Description

The main aim of the paper is to reduce settling time and overshoot in DC Encoder motor to Run AGV with a constant RPM. The second thing is to find transfer function of the system to apply PID and fuzzy on it. Third and final problem is to interface MATLAB simulink with an AGV via Microcontroller.

2.2. Methodology

In the proposed work, PID controller techniques are used using Ziegler Nichols Algorithm and Mamdani Technique is used for fuzzy logic controller [7]. At the end comparison of settling time, rise time, peak time, overshoots and delay time are shown between PID controller and fuzzy logic controller. Arduino 8 bit microcontroller is used for processing and applying the algorithms. MATLAB simulink is used for Simulations [15].

3. Mathematical Modeling and Simulations

3.1 Transfer Function of the system

Fig. 5 shows a general purpose block diagram for the simulation of the system. The system has two major parts, one is mechanical part and the other is an electrical part. It can be observed from the block diagram that, the proposed system is a second order system. The known parameters of motor are given in table 2 [8].

The type of the system and order of the system can be found using equation 1 and equation 2 [1][7]. The type of the system is zero and order of the system is two. The open loop transfer function of the system is given in equation 3, in which K_a is unknown parameter (gain of the system) and can be found using Routh's criterion. Equation 4 is a characteristic equation for stability). Equation 3 leads to equation 3a, where 200Ka=K (Overall Gain of the system). According to table 3, using Routh's Criterion, Value of Ka>-0.04, so here value Ka is taken as 0.04. The given system is under damped and the value of ξ=0.60, which is less than one. Equation 3b is the close loop transfer function of the system [9][10].

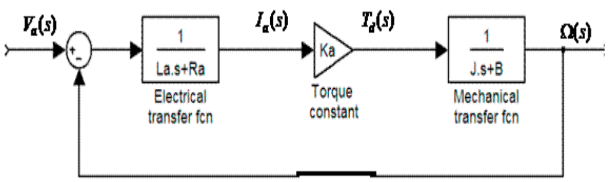

Fig. 5 System Block Diagram for Simulation

$$G(s)H(s) = K\frac{(1+T_1(s))(1+T_2(s))......(1+T_n(s))}{(1+T_a(s))(1+T_b(s))......(1+T_m(s))} \quad (1)$$

$$G(s)H(s) = K\frac{(1+T_1(s))(1+T_2(s))\ldots\ldots(1+T_n(s))}{s^j(1+T_a(s))(1+T_b(s))\ldots\ldots(1+T_m(s))} \quad (2)$$

$$G(s)H(s) = \frac{K_a}{(s+4)(0.005s+0.01)} \quad (3) \qquad 1 + K_a G(s)H(s) = 0 \quad (4)$$

$$G(s)H(s) = \frac{200K_a}{(s+4)(s+2)}$$ (3a), where Ka=0.04 and viscous Friction B is 0.01

$$\frac{G(s)}{1+G(s)H(s)} = \frac{16}{(s+4)(s+2)}$$ (3b) (Close loop Transfer Function of the system)

3.2 Steady state analysis of the system

Equation 5 is suitable for the under damped system is ($\xi<1$). Steady state parameters (K_P, K_v, K_a and $e_{ss)}$ can be found using equation 6 to 9 [4].

$$C(t) = (\left(\frac{\omega_n}{\sqrt{(1-\xi)}}\right) e^{-\xi\omega_n t} \sin \omega_d t) \quad (5) \qquad K_p = \lim_{s\to 0}(G(s)H(s)) \quad (6)$$

$$K_v = \lim_{s\to 0} s(G(s)H(s)) \quad (7) \qquad K_a = \lim_{s\to 0} s^2(G(s)H(s)) \quad (8)$$

$$E_{ss} = \frac{1}{1+K_p} \quad (9)$$

3.3 Transient analysis of the system

Transient parameters are shown in Equation 10 to 14. Every system opposes an oscillatory behaviour (damping). This tendency controls the closed loop poles of the system. The response of the system is decided by the close loop poles of the system [1].

$$T_d = \frac{(1+0.7\xi)}{\omega_n} \quad (10) \qquad T_r = \frac{(\pi-\beta)}{\omega_d} \quad (11)$$

$$T_p = \frac{(\pi)}{\omega_d} \quad (12) \qquad M_p \text{ in } \% = 100\left(e^{\frac{-\xi\pi}{\sqrt{1-\xi^2}}}\right) \quad (13) \qquad T_d = \frac{(4\xi)}{\omega_n} \quad (14)$$

3.4 Simulation of the PID controller using Ziegler-Nichols Algorithm

After finding the transfer function of the system, it becomes vital to find Kp, Ki and Kd (PID parameters). Z-N algorithm is one of the powerful algorithms to tune and to find Kp, Ki and Kd parameters. It also works well for the second order system [1-3]. ZN algorithm deals with Ultimate time and ultimate gain, so it is necessary to find the response of the system without any controller. Response of the system helps to find Kp, Ki and Kd using table 5 and fig. 7. A tangent line is shown in fig. 7 is called ultimate time (T, where T=0.62 from the graph). Length L is almost equal to the delay time (L=0.06 from the graph). Figure 8 is the simulation block diagram of the system to compare the simulated response [2].

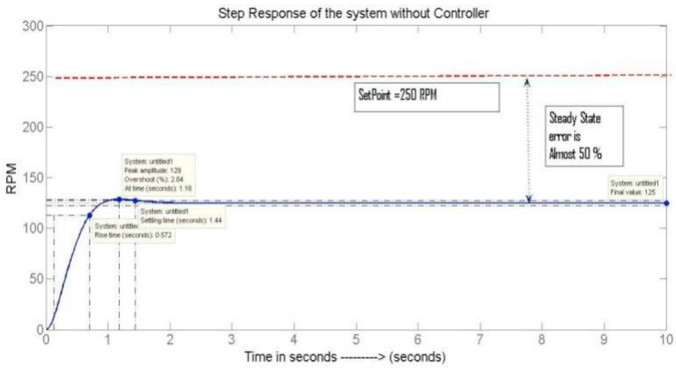

Fig. 6. Response of the System without any controller

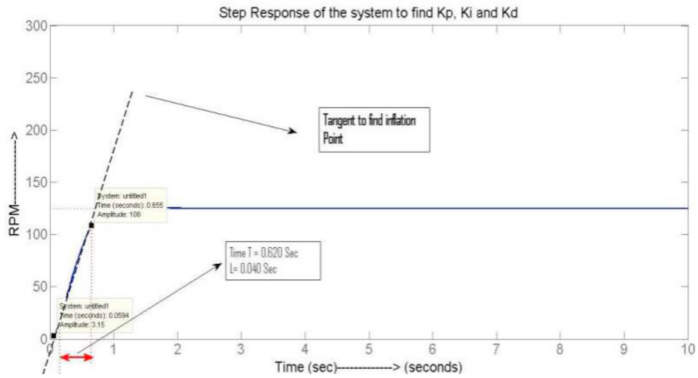

Fig. 7. Response of the System to find Kp, Ki and Kd

Four responses are taken in the simulation. Step response, response without any controller, PID controller's response and fuzzy logic controller's response. That is why MUX is connected before scope to integrate all the response [9-13]. It can be observed from the fig. 6. that, steady error is almost 50 % in the system (when there is no controller). Equation 15 and 16 are the transfer function for the PID controller. Finding T and L to tune Kp, Ki and Kd is shown in Fig. 7. A tangent is drawn at inflation point to calculate T and L, where T is a time from delay time to rise time and L is a time from origin to delay time [14]. Delay time is from when system starts responding. Rise time is from when system starts taking a curve. Here T=0.62sec and L=0.06 sec from the graph. It should be noted that simulation is done for unit step input, which can be scaled at any ratio [1][3].

$$\mathrm{PID(s)} = \mathrm{K_p} + \left(\frac{1}{\mathrm{s}}\right)(\mathrm{K_i}) + \mathrm{K_d s} \quad (15) \qquad \mathrm{PID(s)} = 12 + \left(\frac{1}{\mathrm{s}}\right)(0.12) + 0.03\mathrm{s} \quad (16)$$

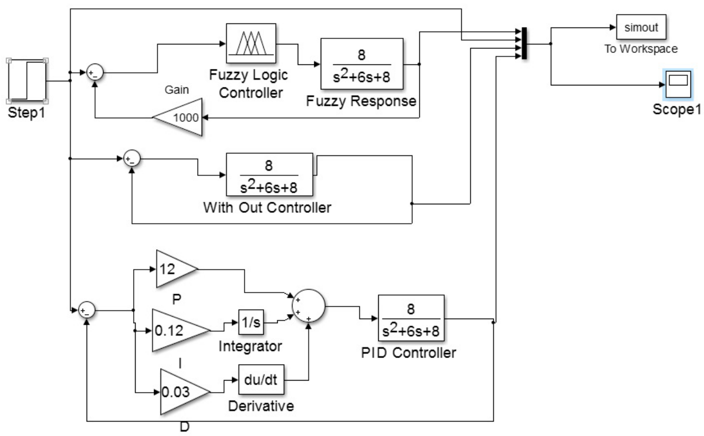

Fig. 8 Simulation Block diagram of the system

Table 3. Routh's Criterion

Laplace Operator	First Coefficient	Second Coefficient
S^2	1	8+200Ka
S^1	6	0
S^0	(8+200Ka)	-

Table 4. Fuzzy Rules for the System

RPM	Range RPM	Voltage	Range (Volt)
S1	0,0,90	PWM3	1.5, 2.0, 2.6
S2	50,95, 140	PWM3	1.5, 2.0, 2.6
S3	110,150,190	PWM2	0.7,1.3,2.0
S4	160,210,260	PWM1	0, 0 , 1.3

Table 5 P, PI and PID values for using Z-N algorithm

Controller	K_p	K_i	k_d	K_p	K_i	k_d
P	T/L	-	-	10	-	-
PI	0.9(T/L)	L/0.3	-	9.3	0.2	-
PID	1.2(T/L)	2L	0.5 L	12	0.12	0.03

4. Implementing Fuzzy logic Controller

Fig. 9a and 9b show the Input and Output Membership function for the fuzzy logic controller respectively. There are four membership functions in input and three membership functions in output. Input functions are related to RPM and Output Membership functions are in the form of voltage. The range of Input RPM is 0 to 270 RPM and output voltage range is 0 to 2.6 voltages [4][7]. Inputs are denoted as S1, S2, S3 and S4, whereas outputs are denoted as PWM1, PWM2 and PWM3. PWM means pulse width modulation. Set point in the system is 250 RPM and PWM works with 5 volts, so 2.5 to 2.6 volts are acceptable.

a b

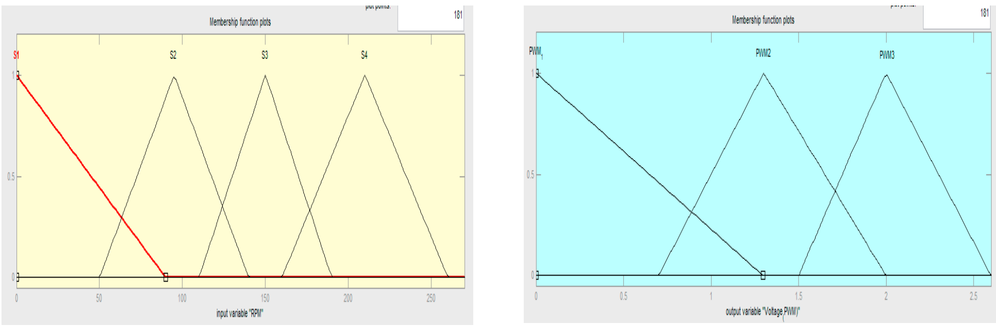

Fig. 9 (a) Input Membership Function; (b) Output Membership Function

Fig. 10a and 10b show the result of surface view and rule view of the system respectively. It can be observed from the surface viewer that, how voltage gets dropped with the increment of RPM with the use of Fuzzy logic controller. According to rule viewer at 250 RPM, voltage should be 0.58. Fuzzy rules are shown in table 5.

a b

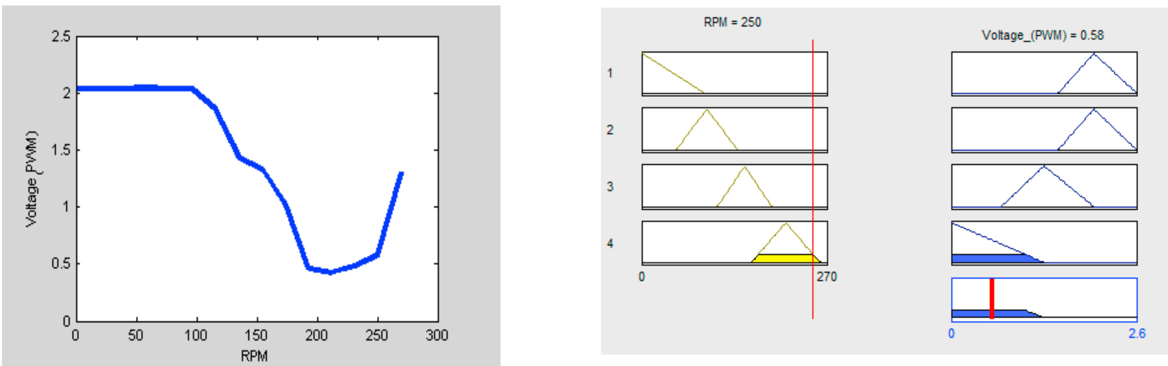

Fig. 10 (a) Surface viewer (b) rule viewer of Voltage and RPM for the given system

5. Results and Discussion

Fig. 11 and 12 show the simulated and actual response of the system respectively. The simulation is done with unit step input (which can be considered our set point). The purple line is the input line, which is a unit step response. The blue one is the response of the system without applying any control algorithm. Red and yellow lines are the PID and fuzzy response of the system respectively. All steady state and transient parameter values are shown in table 6.

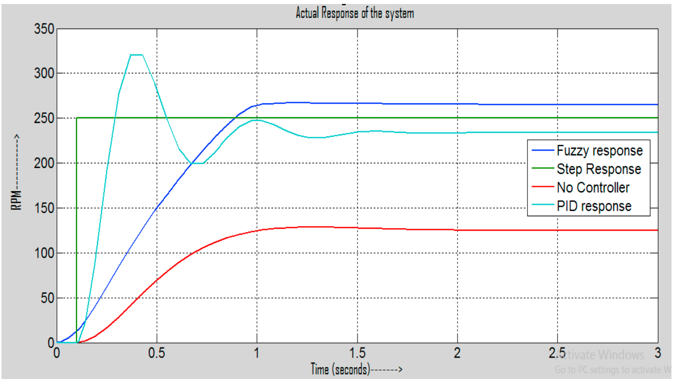

Fig. 11 Simulated Response of the System

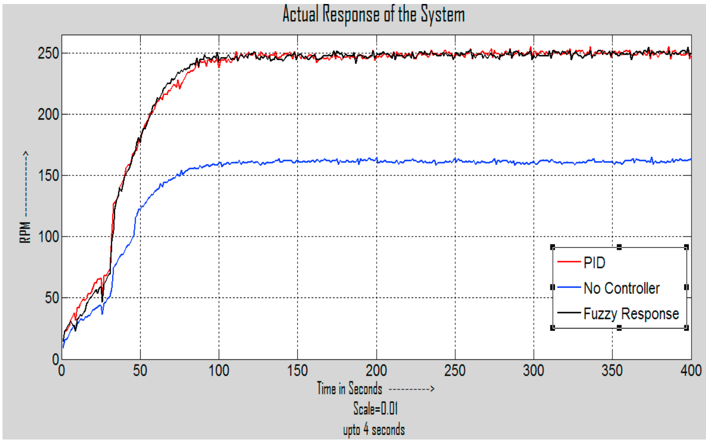

Fig. 12 Actual Response of the System

Table 6 Final results of the system (Simulated and Actual results)

Sr No	Parameters	Without controller Simulated	Without controller Practical	Simulated Value PID	Practical Value PID	Fuzzy Logic Simulated	Fuzzy Logic Practical
1	Ess	50%	37%	6%	2%	3%	0.5%
2	Mp	Undershoot	Undershoot	16%	2.5 %	3%	1 %
3	Ts	2.1 sec	1.5 sec	2.5 Sec	1.5 Sec	0.9 sec	0.8 Sec
4	Tp	1.3 Sec	1.3 Sec	0.4 sec	1.3 sec	0.8 sec	0.7 sec
5	Tr	0.75 sec	0.75 sec	0.3 sec	0.7 sec	0.5 sec	0.6 sec
6	Td	0.1sec	0.1sec	0.1 sec	0.01sec	0.05 sec	0.01 sec

6. Conclusion based on Results

- Steady state error, overshot and rise time of the system in PID controller is comparatively lesser for fuzzy logic controller.
- AGV achieves constant speed in just 0.8 seconds with only 1% overshoot.
- Only 1% overshoot is captured Fuzzy logic controller
- Steady state error is less compared to PID Controller
- Rising Time is also lesser in Fuzzy logic controller.
- Fuzzy logic controller gives more stability and accuracy to the system.
- Small oscillations in the response could be reduced using ANFIS controller in future.

References

[1] P. Parikh, R. Vasani, and S. Sheth, "Velocity Analysis of a DC Brushed Encoder Motor using Ziegler-Nichols Algorithm: A Case of an Automated Guided Vehicle," *Indian Journal of Science and Technology* 9.38 (2016).

[2] P. Parikh, S. Sheth, and T. Patel, "Positional Analysis of a DC Brushed Encoder Motor Using Ziegler-Nichols Algorithm," *CAD/CAM, Robotics and Factories of the Future*. Springer India, 2016. 637-650.

[3] S. Maheriya, and P.Parikh, "A Review: Modelling of Brushed DC Motor and Various type of Control Methods," *Journal for Research| Volume* 1.12 (2016).

[4] P. Parikh, N. Modi, and R. Prajapati, "Control of Industrial Pneumatic & Hydraulic Systems using Serial Communication Technology & Matlab."

[5] P. Parikh, K. Joshi, and S. Sheth, "Color Guided Vehicle–An Intelligent Material Handling Mechatronic System," *Proceedings of the 1st International and 16th National Conference on Machines and Mechanisms (iNaCoMM 2013), IIT Roorkee, India*. 2013.

[6] Parikh P., Shah H. and Sheth S," A Mechatronics design of a line tracker robot using Ziegler Nichols control technique for P, PI and PID controllers," International Mechanical Engineering Congress (IMEC- 2014), June 13-15, 2014. DOI: 10.13140/RG.2.1.4107.4722

[7] P. Parikh, H. Shah and S. Sheth., "Development of a multi-channel wireless data acquisition System for swarm robots - A Mechatronic Approach using Arduino UNO and MATLAB," International Journal of Engineering Development and Research (IJEDR), ISSN:2321-9939,2 (1), pp. 717-725.

[8] K. Tamboli, S. Sheth, V. Shah, V. Modi, V. Gandhi, and N. Amin, "Design and Development of a Mechatronic System for the Measurement of Railway Tracks," Proceeding of the International conference CCEED under IEEE, pp. 264-269.

[9] T. Patel, S.Sheth, and P. Patel," Design of Semiautomatic Hydraulic Blanking Machine Using PLC," National Conference on Innovative & Emerging Technologies (NCIET- 2015), pp. 410-412. DOI: 10.13140/RG.2.1.4529.6803

[10] Parikh, P., R. Vasani, S. Sheth, and J. Gohil. "Actuation of Electro-Pneumatic System using MATLAB Simulink and Arduino Controller-A case of a Mechatronics systems Lab." (2017).

[11] Madhusudan, Kanoria Shubham Anil1 Pandya Jeet, and Priyam Anilkumar Parikh. "Design and Development of Universal Motor Control Unit using MATLAB and Arduino." (2017).

[12] Biswajit, Priyam Parikh1 Reenav Shukla2 Mansi, and Ghosh3 Nisarg Vivek Kulkarni4 Bryan Clifford. "Actuation of AC and DC load using MATLAB GUI and Serial Communication." (2017).

[13] Nordin Saad, Muawia A. Magzoub, Rosdiazli Ibrahim and Muhammad Irfan, "An optimized fuzzy-fuzzy controller for PWM driven variable speed drives", DOI: 10.5772/61086.

[14] Parikh, P., Vasani, R. and Pujara, A., A Mechatronics Design of a Wireless Electropneumatic system using Android and MATLAB GUI.

[15] Parikh, P., Singh, H., & Nadar, S. K. Design and Development of a Programmable Logic Controller Using Atmel Controller and MATLAB Simulink.