ISSN No.: 2454- 2024 (online)

International Journal of Technical Research & Science

DESIGN AND IMPLEMENTATION OF PID AND ADAPTIVE TUNED PID CONTROLLER FOR INDUSTRIAL APPLICATION

Snehil Mishra[1], Dr. Shimi S. L.[2]
Email Id: anshul.snehil@gmail.com
[1]M.E. Student, Department of Electrical Engineering, NITTTR Chandigarh (India)
[2]Assistant Professor, Electrical Engineering, NITTTR Chandigarh (India)

Abstract- PID controllers have played a great role in industry for controlling different plants with accuracy. Also, various PID control algorithms have been framed to ease its implementation and improve the response of the system. However, conventional manual tuning method have faced the problems of lesser precision and higher response time. Various automatic PID tuning methods have been designed to solve this control problem. Adaptive tuned PID is one of the most widely applicable techniques in controlling a plant output automatically to the desired set point with more accuracy and precision. This research is focused on designing a laboratory scale model for PID and Adaptive tuned PID controller for balancing a shaft with rotor mounted on its one end through brushless motor that illustrates the case of a conventional paper mill industry. The hardware implementation of the design is performed using an arduino uno microcontroller programmed with different algorithms designed for manual tuned PID with constant value, manual tuned PID with variable inputs from potentiometer and adaptive tuned PID controllers respectively. The output of the hardware is depicted on various graphs by importing data to LabVIEW Virtual Instrument. Various transient response characteristics of the designed laboratory model are depicted in the results including the rise time, peak time, overshoot, delay time, settling time and maximum overshoot. Also, a software implementation of the designed model is also performed in MATLAB Simulink for both manual and adaptive tuned PID controller. Finally, a comparative analysis of both the controllers is framed.

Keywords: Adaptive PID; Rise time; Overshoot, Brushless motor; PID Tuning, Comparative Analysis

1. INTRODUCTION

One of the most widely used industrial controllers are PID controllers. Due to their ease of implementation and robustness, PID controllers are one of the most recognized and accepted controller in the industry. Although, their major shortcomings are improper tuning and precision when it comes to controlling the complex non-linear industrial processes [1,2]. With well tuning and more precision for PID controllers, one could achieve the targets of accomplishing these difficult problems of the Industry [3]. This is achieved with the help of fine tuning and adaptive techniques for PID controllers to tackle with the situations of nonlinear industrial processes. Most widely employed PID controllers are in the process control Industry [4,5]. PID controllers contributes to a total of approximately 95% of the closed loop operations of industrial automation sector [6]. These controllers receive inputs from sensors, meters, etc. and depending on PID control function they deliver output control signals to the controlled or manipulating devices such as relays, actuators, etc [7,8]. Adaptive tuning becomes indispensable for parameter variations, which set parameter controllers cannot adequately compensate. Recently, predictive control technique has been extensively appeared in the process control [9-12]. In the predictive control, a process model is requisite to predict the prospect effects of the control action at the present. First principles-based nonlinear models are complicated to build up for numerous industrial cases [13]. Adaptive control is always a vigorous area of explore. Efforts are towards robustness of adaptive systems. To design an adaptive controller means a controller with time varying parameters adjusting to lodge plant and disturbance uncertainties [14-15]. Model Reference Adaptive system (MRAS) has been explored much as a compelling method of adaptive control. Method uses the model either absolutely or overtly. Plant output or process output should follow the model output. Model Reference Adaptive Control (MRAC) approach should be followed when desirable performance stipulations are given in terms of model. Mostly, MRAS consists of two loops, ordinary feedback loop as an inner loop and outer loop with controller parameter adjustment mechanism to diminish the difference between model & process outputs [16-18]. It is identified as Reference Model-based Adaptive System (RMAS) also. For reference, consider the closed loop control system as shown in Fig. 1.1. It has the reference signal, r as input, y as output signal to be measured and controlled, and an error signal e, where,

$$e = r - y \quad (1.1)$$

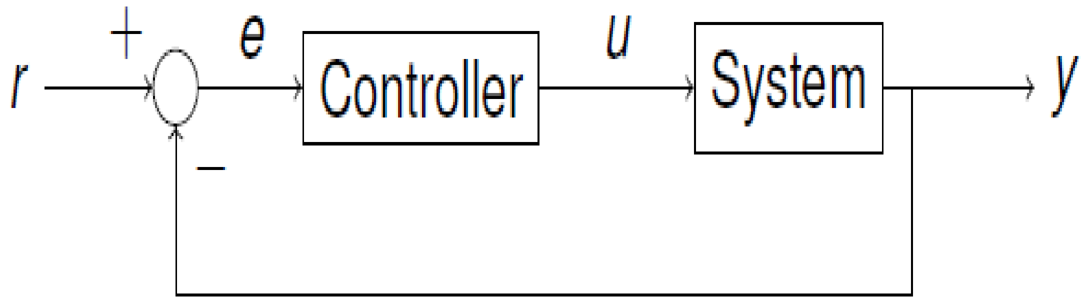

Fig. 1.1 Closed Loop Control System with PID Controller

The PID controller implemented in the above system is depicted in Fig. 1.2.

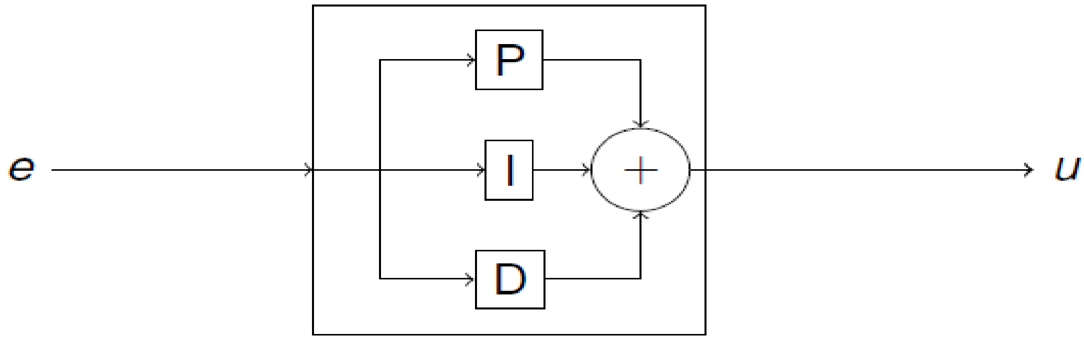

Fig. 1.2 PID Controller

As depicted in above figure, PID controller is a feedback based controller mostly implemented for closed loop control systems.

2. PROBLEM FORMULATION

The industry on which the proposed work may be carried out is a paper industry . The manufactured paper will be rolled up on reel. Initially the diameter of reel is small but as far as he manufacturing processed the diameter of the reel started varying i.e. started increasing. Variation in the diameter of reel is just because of continuous rolling of paper on it. The motor which connected to the reel will produce the variable speed to handle the variation of diameter of reel. For the synchronization of speed of motor with varying diameter there must a feedback control system. This feedback in the proposed design is to be taken from a potentiometer in order to balance a shaft on which the motor is mounted. A brushless motor with propeller is to be considered for base motor whose speed control will eventually balance the shaft. The angle of the shaft that is the feedback parameter is to be considered for controlling the speed. Presently the industries are using PID controllers.

- To design a conventional PID controller for paper industry using manual tuning.
- To design a model for manual tuned PID and adaptive PID in the MATLAB Simulink environment and obtain its transient response.
- Real Time monitoring of the hardware implemented on the shaft balancing model for PID with manual and adaptive tuning.
- Result analysis in the LabVIEW Virtual Instrument on computer for real time monitoring of the results of hardware.
- Comparison of Manual tuned PID controller and Adaptive PID controller from the above implementation of a laboratory prototype model in terms of its transient response characteristics.

3. VARIOUS TUNING ALGORITHMS FOR PID CONTROLLER

Comparison between the manual tuning based PID controller and neuro-fuzzy based adaptive PID tuning controller is to be framed by implementing both the control algorithms in real time. The algorithms are to be implemented in a controller to solve a problem of paper reel assembly. The manual tuning is achieved by various methods including manually controlling the speed of motor by analog value given to the arduino using external potentiometer and using external potentiometers to tune the PID values with the appropriate constants K_p, K_i and K_d respectively. Whereas the adaptive tuning of PID involves programming the controller with the PID adaptive tuning algorithm.

Two different algorithms have been implemented to design this model, including PID with manual tuning and PID with adaptive tuning. These are elaborated below:

3.1 PID Algorithm with Manual Tuning

The manual tuning of PID was the first step in testing the design in order to get the desired value of 'θ', i.e. the angle of the shaft for attaining the perfect balancing of the shaft by controlling the brushless motor speed. This was done in three different methods:

3.1.1 Controlling the Speed of Motor Using Potentiometer

This step involved manually controlling the speed of motor by analog value given to the arduino using external potentiometer and generating the PWM pulses to the corresponding analog input without any algorithm or tuning. This gave the rough idea for the point where balancing is attained and the angle and speed values for the same for maximum and minimum points around the stable position.

3.1.2 Tuning the PID Values with Constants Manually

In this step, the PID algorithm was implemented in the controller with feeding the constant values for the proportional (K_p), integral (K_i) and derivative (K_d) constants in the algorithm. It also required frequent changes in the K_p, K_i and K_d values. This required tuning multiple times in order to achieve the stability position. Also, there was a lot of fluctuations in the shaft during balance position.

3.1.3 Tuning the PID with three Potentiometers

This step gave more accurate results and stability in the design using external potentiometers to tune the PID values with the appropriate constants K_p, K_i and K_d respectively. This tuning method was more fast and precise as compared to manual tuning with potentiometers.

3.2 PID Algorithm with Adaptive Tuning

This algorithm involved programming the controller with the PID adaptive tuning algorithm for getting more stable and precise results for achieving the stability. It involved new commands and functions to be used for adaptive control of PID based on neuro-fuzzy control in the arduino library and programming.

4. DESIGN METHODOLOGY

The hardware implementation for the proposed model have been performed using the PID control of a sensor less brushless DC motor connected to a shaft (paper reel attachment) that is mounted at a particular height from the base. A propeller is attached on the top of brushless motor to provide it thrust to lift upwards in order to control the system to achieve set point (angle governed by potentiometer in our case). This brushless motor is allowed to lift in upward and downward directions based on the feedback taken from a potentiometer mounted on the middle of the shaft that generates an analog output voltage proportional to the angle of the shaft. The idea is to keep the angle of the shaft at a constant value so that it remains balanced at different loads. The speed of the brushless motor is controlled using an miniature dc to AC 3 phase converter that further receives the PWM signal from a controller output. The controller used here is an Atmega 32 microcontroller embedded in an Arduino uno board. The controller is programmed with an adaptive PID control algorithm tuned with the optimum parameters for the Kp, K_v and K_i constants. The mechanical hardware for the designed model is represented in the Fig. 4.1.

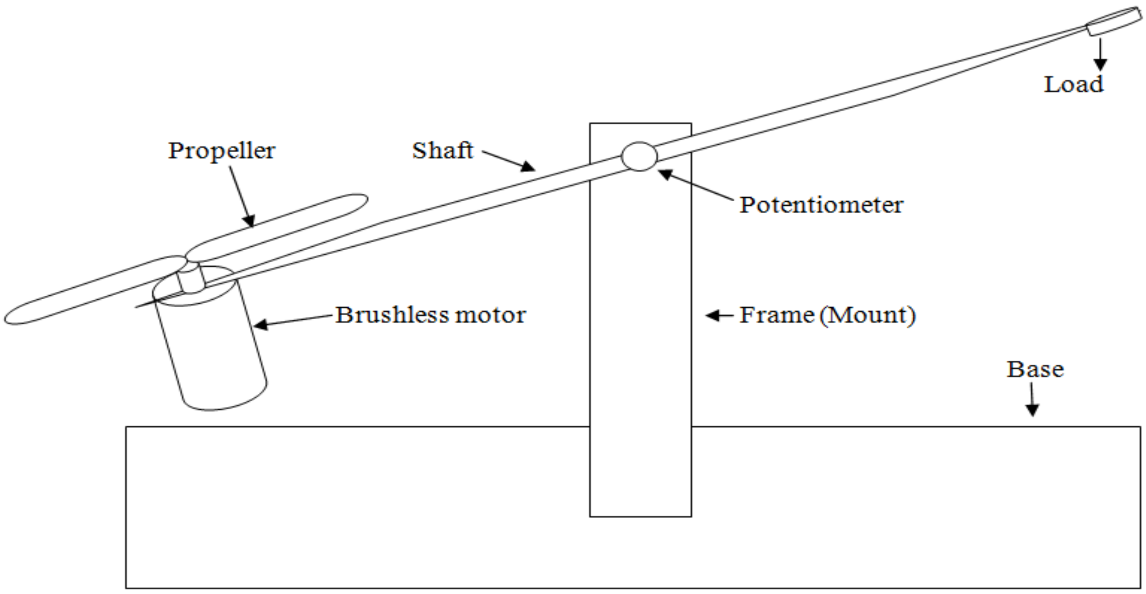

Fig. 4.1 Line Diagram of the Proposed Hardware Model

The laboratory model for the Adaptive PID implementation of the paper reel assembly is designed as per the above line diagram. As the reel of the paper roll always changes its diameter and thus the load on the conveyer belt, the speed of the motor changes.

In order to control the speed of the motor as per the reel diameter and thus load torque, the above model is implemented in real time depicting a brushless motor to control the angle of the shaft by controlling its speed proportional to the feedback input parameter, i.e. the tilt angle.

All the electrical specifications of the parts used in the hardware implementation is depicted in Table 4.1.

It includes the name of the components used along with its technical specifications.

Table- 4.1 Technical Specifications of Component

S.No.	Name of Component	Technical Specifications
1.	Brushless Motor	Speed Constant: 1300 K_v
2.	Propeller	Diameter: 5”
3.	ESC (Electronic speed controller, Single phase to three phase converter)	Current Rating: 30 A
4.	Potentiometer for feedback	10K Ω , 0 - 5 Volts
5.	Arduino Uno Board	Inbuilt ADCs (10 bit resolution), 32 Bit microcontroller, 6 Analog inputs, 15 Digital inputs, Power supply (5V), 2 Serial communication pins (Rx and Tx), Internal voltage regulation (7805)
6.	Programming cable for Arduino Uno	USB to serial cable
7.	Rechargeable Battery	12V, 7A (Lead Acid Battery)
8.	Mount for Frame	Aluminum
9.	Shaft	Wooden
10.	Connecting Leads And Connectors For BLDC Motor	High temperature sealed copper wires

The design and implementation of hardware is completed by taking various steps. These are explained below.

4.1 Development of the Framework

First of all, the framework is designed including the shaft on which the motor is mounting, its stand, base clamps and hubs for mounting potentiometer for angle measurement in order to take the feedback and other parts required to fix the motor and propeller on the shaft and circuit on the base. The motor represents the reel on which the paper is rolled on, as the diameter of reel changes continuously and so is the motor speed, the reel has to be loaded with the paper as per the requirement or load torque that in this case represents the shaft on which the motor is mounted and needs to be balanced by increasing or decreasing the speed of motor. Therefore, this motor and propelled based shaft balancing model requires a set point given externally through a potentiometer and the measured value of angle of the shaft using another potentiometer mounted on the center of the shaft.

4.2 Implementing PID Algorithms with Manual and Adaptive Tuning in Microcontroller

The above two parameters are implemented onto the arduino board as control system for controlling the speed of motor and thus position of shaft using an Electronic speed controller (ESC). This microcontroller is programmed with the algorithms for manual PID tuning and adaptive PID tuning successively to check the results and balance the shaft. It generates appropriate PWM pulses to perform this function. The PWM outputs of the Arduino are connected with the ESC to further control the speed of brushless motor.

4.3 Interfacing ESC with Brushless Motor

The brushless motor has three wires “A,” “B,” and “C,” at its output and their “free” ends, those that stick out of the motor, are connected to the ESC. The ESC uses electronics to connect any of these wires to positive or negative, to achieve one of six possible combinations that results in an electromagnetic field in a precise location in the motor. The timing and duration of these connections is critical—and unbelievably short.

Mechanical switches are simply incapable of the task. But high-power electronic switches—known as Metal Oxide Semiconductor Field Effect Transistors (MOSFETs, or FETs for short)—can turn on and off in a fraction of a second and are ideally suited for this application.
where the motor constant Kv and V is the power required in volts. Example, an outrunner motor with 12 poles that has a Kv (rpm per volt) of 1,500 and is powered with 24 volts (6S Li-Poly) will spin at 36,000 rpm (24 x 1,500 = 36,000). The six coil combinations needed for a full magnetic rotation must be repeated for every north pole in the motor. The example motor has 12 poles, so the controller must switch the FETs 36 times per revolution of the shaft (6 north poles x 6 steps per magnetic rotation). That means there are 1,296,000 electrical cycles per minute (36,000 rpm x 6 winding phases x 6 poles = 1,296,000), or 21,600 cycles per second. The controller must successfully switch between the phases every 1/21,600 second. The fig. 3.3 shows that the three motor wires—A, B, and C—can each be connected to positive or negative poles of the power source by the ESC.

4.4 FET Drive Circuit for Electronic Speed Controller

Each FET used in the ESC drive circuit has three connections: gate, source, and ground. To turn the FET on and create a circuit, the gate leg has to be driven to a point that is 5-10 volts higher than the voltage of the source leg on the FET, which is connected to the motor power source.

4.5 Motor Position Detection Circuit

The ESC has to know the precise location of the rotor magnet(s) to accurately sequence the connections that the FETs make. This is the trickiest thing that the ESC has to do. There are two main ways to go about this: sensored and sensorless. Sensored systems use electronic (Hall) sensors in the motor to track the rotor. This requires additional parts in the motor (sensors) and an additional wiring harness to connect the motor sensors to the controller. sensored motors and controllers are popular in RC car applications, because they provide a slightly smoother motor start than the sensorless controller. sensored systems were popular in the early days of RC brushless aircraft power systems; however, they are generally considered to be less reliable and less efficient than sensorless systems, so they are no longer popular for such applications. The schematic diagram of the hardware implementation and actual hardware.

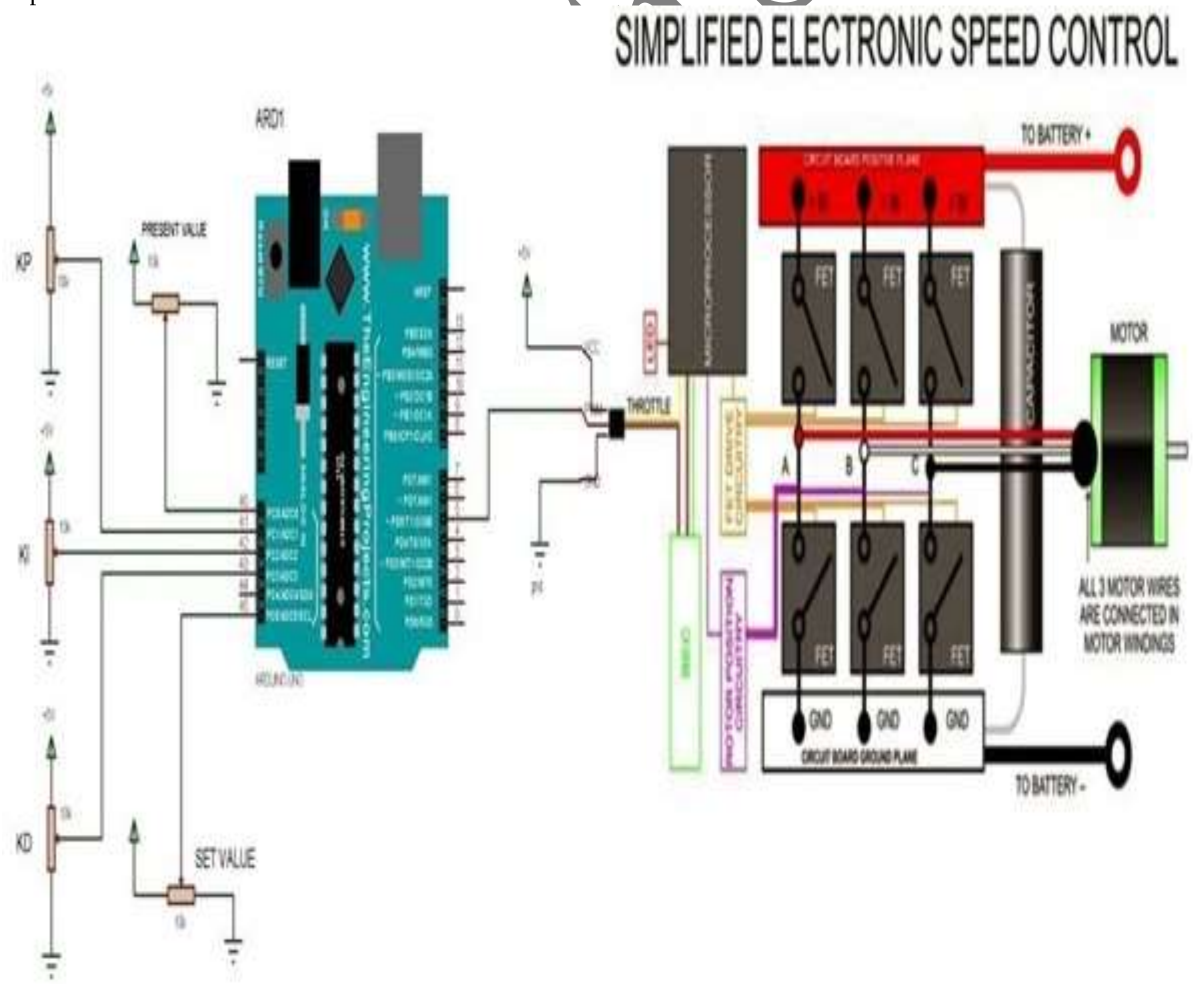

Fig. 4.2 Schematic Diagram of the Hardware Implementation

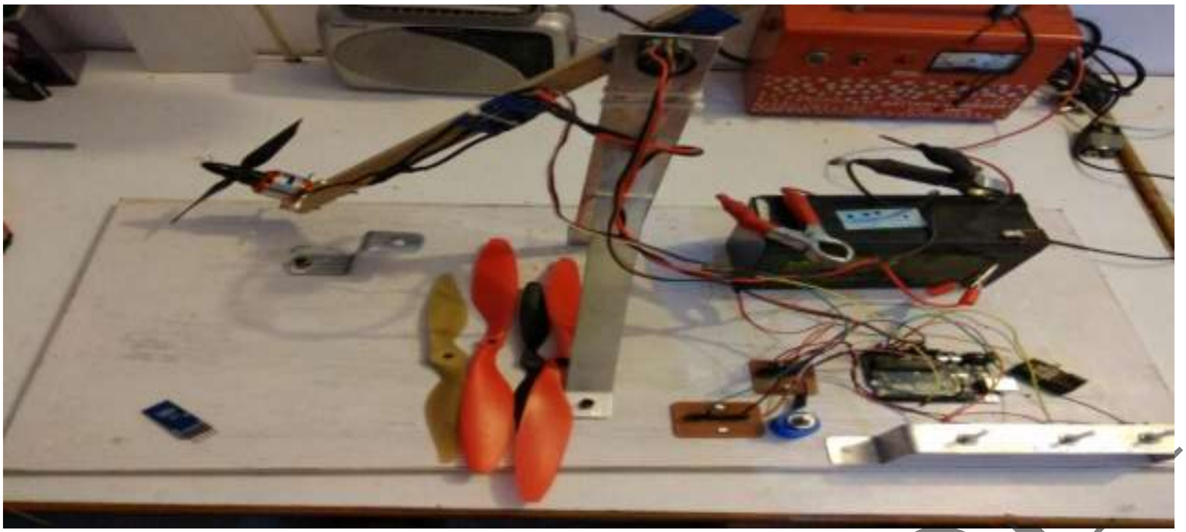

Fig. 4.3 Fabricated Design of Real Time Hardware Implementation

5. RESULT AND DISCUSSIONS

Various PID algorithms with different tuning methods were simulated in MATLAB SIMULINK to obtain the results.

5.1 Manual Tuned PID Controller in Simulink

Here the parameters for PID control were tunable manually by feeding the values of PID constants: Kp, Ki and Kd. Fig. 5.1 and Fig. 5.2 represents a step input to the PID tuner and its output that closely follows the input but has many oscillations in the response.

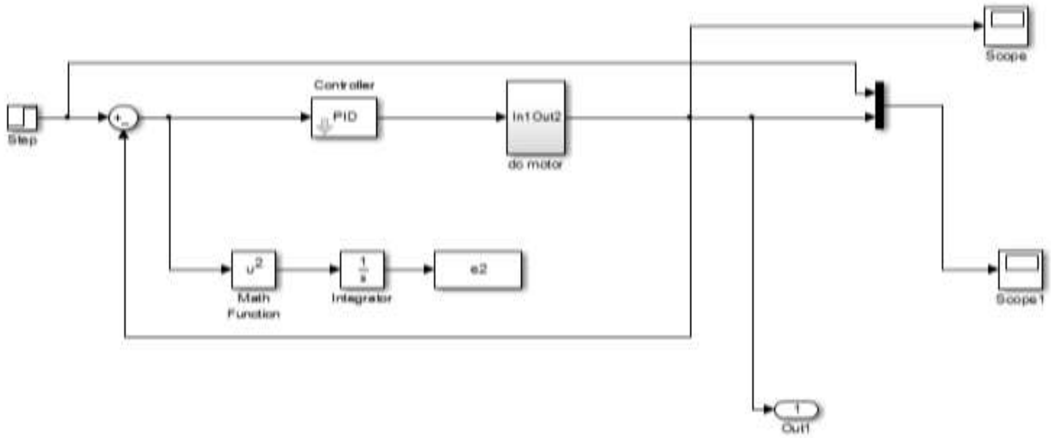

Fig. 5.1 PID For Motor Control Model with Manual Tuning Parameters

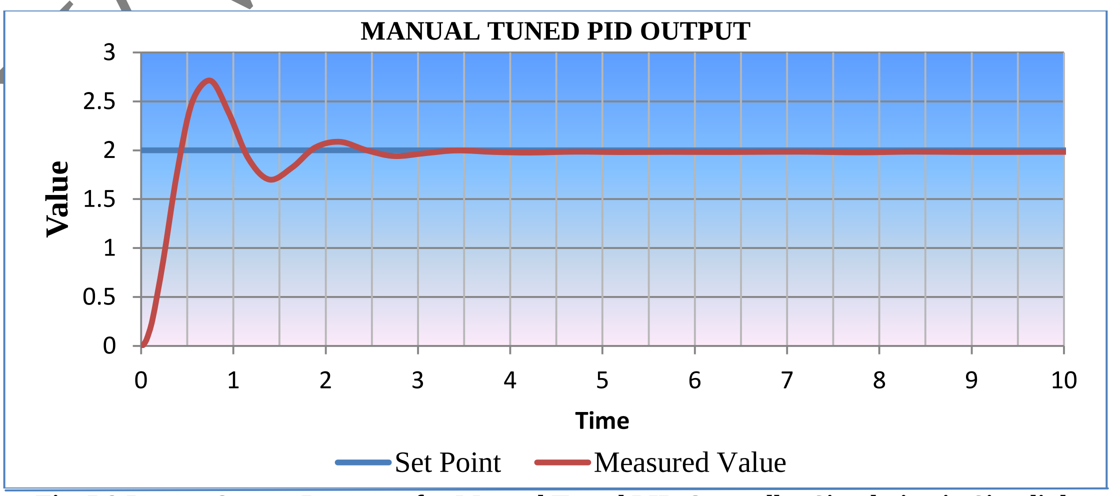

Fig. 5.2 Input - Output Response for Manual Tuned PID Controller Simulation in Simulink

Table-5.1 depicts various parameters of transient response characteristics for Manual tuned PID model in Simulink including rise time, peak time, overshoot, settling time, delay time and maximum overshoot at set point of 2.

Table-5.1 Transient Response Characteristics of Manual Tuned PID Simulation

S. No.	Transient Response Characteristic	Value (in seconds)
1	Rise Time	0.42
2	Overshoot	0.5
3	Settling time	4.1
4	Peak Time	0.75
5	Delay Time	0.4
6	Maximum Overshoot	2.71 (Max)

5.2 Adaptive Tuned PID Controller In Simulink

Next step was to design a system that could tune itself automatically by minimizing the error input based on the given input response and measured response. This model was the adaptive tuned PID controller that could adjust according to the desired set point automatically without the requirement of manual adjustment.
The functional block parameters for motor were set to be:
R = 2 ohms
L = 0.5 H
K_m = 0.1
K_f = 0.2
J = 0.02
The model for adaptive PID control of DC motor and its input - output response is depicted in Fig. 5.3 and Fig. 5.4 respectively.

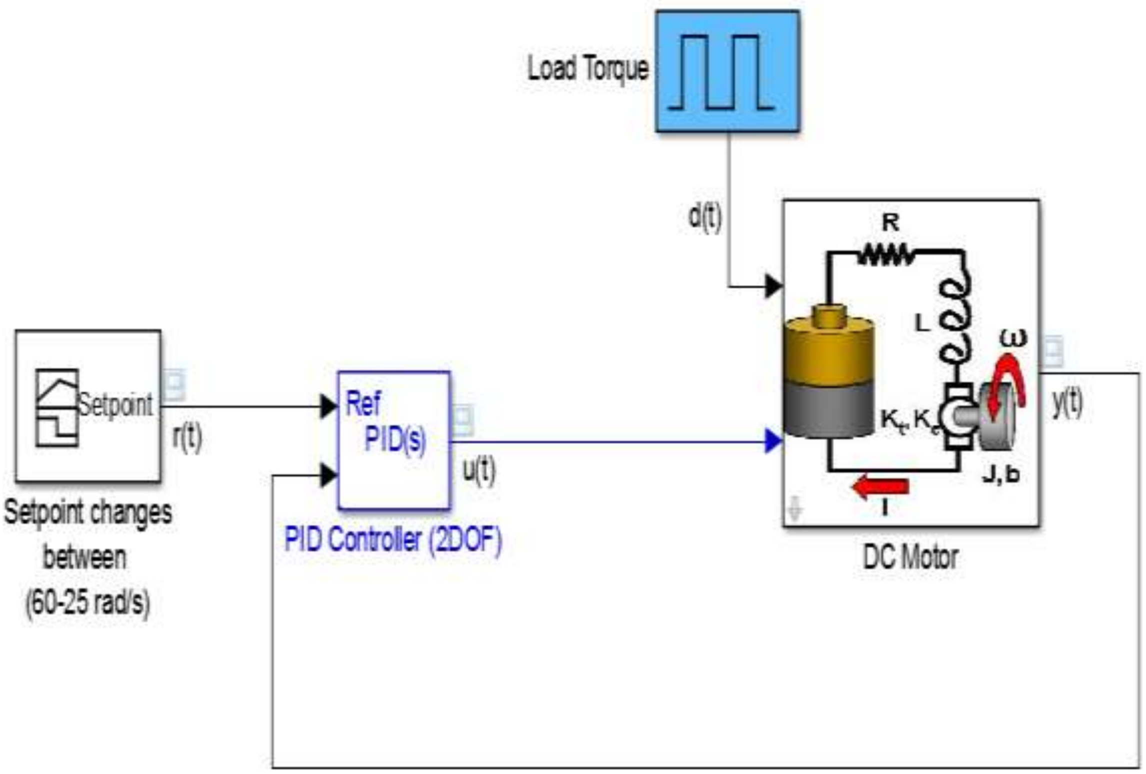

Fig. 5.3 Adaptive Tuned PID Controller Model in SIMULINK

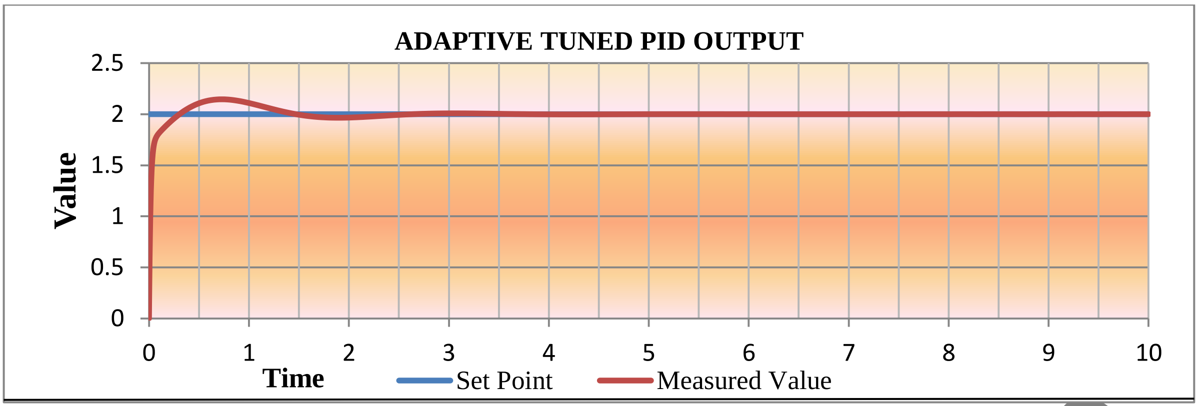

Fig. 5.4 Input - Output Responses for Adaptive Tune PID Controller with Variable Step Input

Table 5.2 depicts various parameters of transient response characteristics for Adaptive tuned PID model in Simulink including rise time, peak time, overshoot, settling time, delay time and maximum overshoot at set point of 2.

Table-5.2 Transient Response Characteristics of Adaptive Tuned PID Simulation

S. No.	Transient Response Characteristic	Value (in seconds)
1.	Rise Time	0.18
2.	Overshoot	0.32
3.	Settling time	1.5
4.	Peak Time	0.67
5.	Delay Time	0.015
6.	Maximum Overshoot	2.2 (Max)

After performing the simulations in MATLAB. The hardware was tuned as per the parameters attained from the simulation model and tested with the motor propeller and shaft balancing system for paper reel assembly problem. The results were acquired by self designed DAQ based on arduino uno board and serially communicated to the PC in the LabVIEW environment. Different results were obtained at various loads and fluctuations in the shaft angle. The output response generally shows the measured PID variable, that is the desired angle and the set point that is set for the shaft using external potentiometer. It involves the designing of a Virtual Instrument in LabVIEW to view the real time response of the designed laboratory prototype for manual and adaptive tuning. The output of laboratory model was fed to the computer using a serial cable this data was interpreted in the LabVIEW designed VI showing the response values in the tabular as well as graphical form. Various PID algorithms with different tuning methods were simulated in MATLAB SIMULINK and in real Time on Hardware platform to obtain the results. The VI designed in LabVIEW for monitoring the results of hardware in real time is depicted in Fig. 5.5.

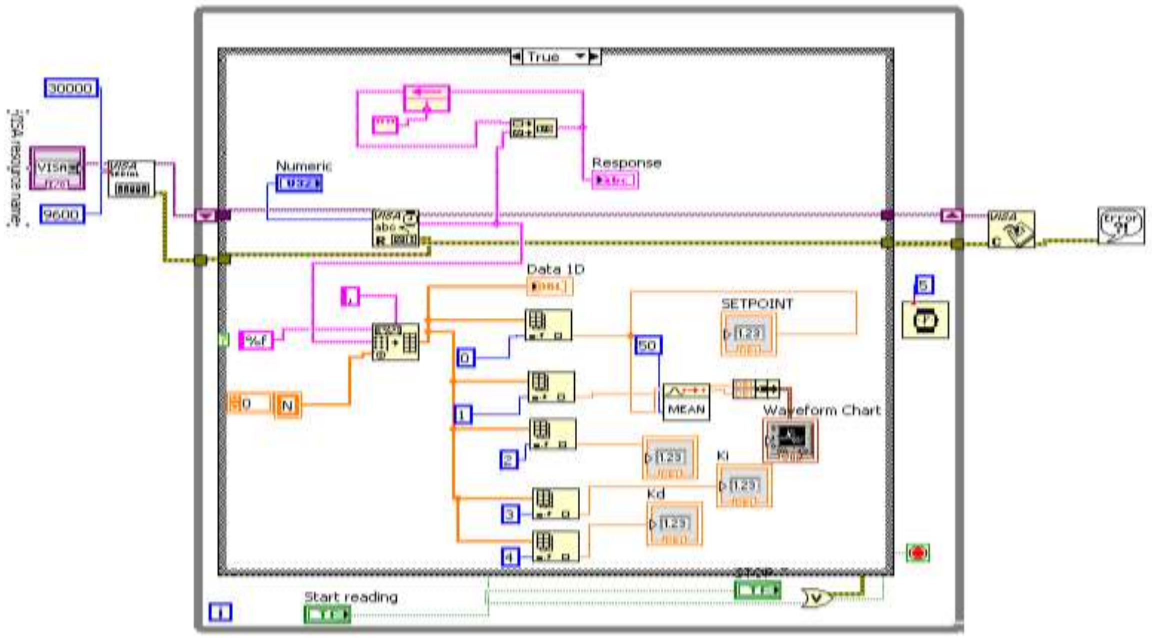

Fig 5.5 LabVIEW Block Diagram for Manual PID Data Recording in Real Time from Hardware

5.3 Transient Response Of Manual Tuned Pid Controller In Real Time

A transient response is the response of a system to a change from an equilibrium or a steady state. Damping oscillation is a typical transient response, where the output value oscillates until finally reaching a steady-state value (The underdamped response case). The transient response of manual tuned PID controller is depicted in figure 5.6. From the above graph of transients response of Manual PID tuned at particular value of Kp=0.32, Ki=64.51, Kd=2.53 is having a decaying oscillation with underdamped Response.

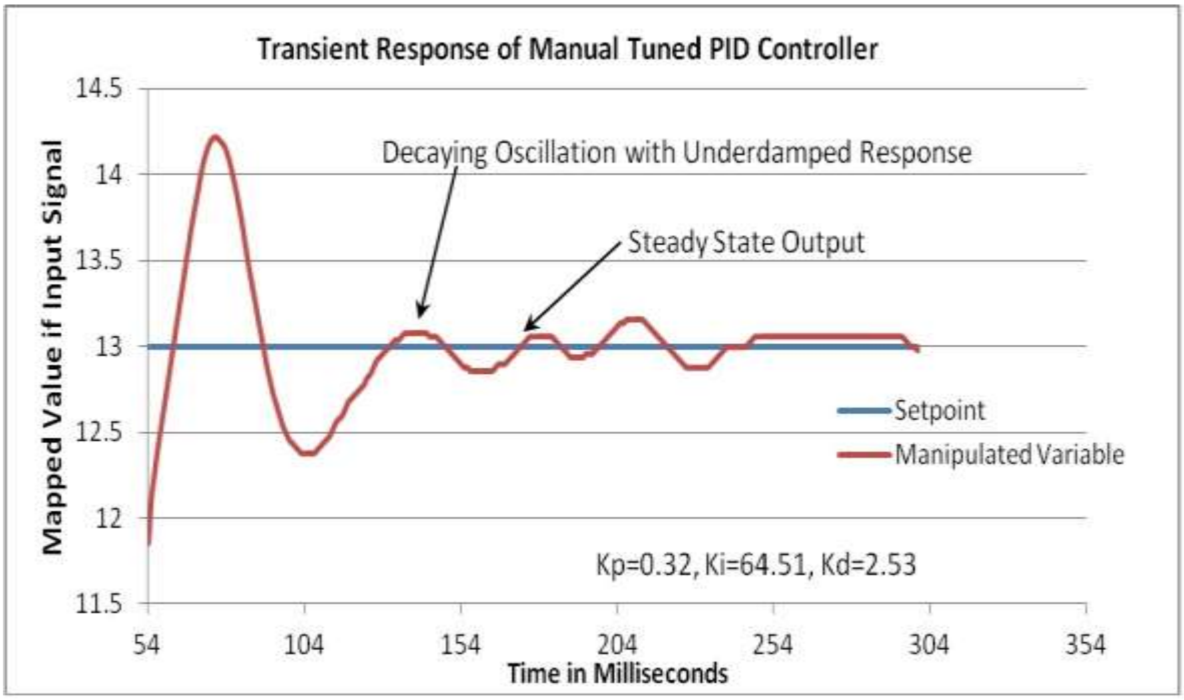

Fig. 5.6 Transient Response of Manual Tuned PID (Real Time Hardware)

Table-5.3 Properties of Transient response of Manual PID

S. No.	Parameters	Values	Scaling Factor	Net Value (in ms)
1	Rise Time	8	5	40
2	Settling Time	126	5	630
3	Maximum Overshoot	1.22	None	1.22
4	Delay Time	5	5	25
5	Peak Time	23	5	115
6	Steady State Error	0.16 (Max)	None	(0.16) Max

5.4 Transient Response of Adaptive PID Controller in Real Time

In adaptive PID controller we two type of tuning parameter one is call Aggressive parameter and other is call conservative parameter. We set the adaptive controller to use the conservative parameter when controller is near the set point and more aggressive Tuning parameter when we are farther away from the set point.The transient response of manual tuned PID controller is depicted in figure 5.7. From the above graph of transients response of Manual PID tuned at particular value of aggressive K_p=4, aggressive K_i=0.2, aggressive K_d= 1 and value of conservative K_p=0.09, conservative K_i=115.34, conservative k_d=2.45.

Table-5. 4 Properties of Transient Response of Adaptive PID

S. No.	Parameters	Values	Scaling Factor	Net Value (in ms)
1	Rise Time	2251-2216=35	5	175
2	Settling Time	2358-2216=142	5	710
3	Maximum Overshoot	1.00	None	1.00
4	Delay Time	2251-2216=35	5	120
5	Peak Time	2276-2216=60	5	300
6	Steady State Error	0.24 (Max)	None	(0.24) Max

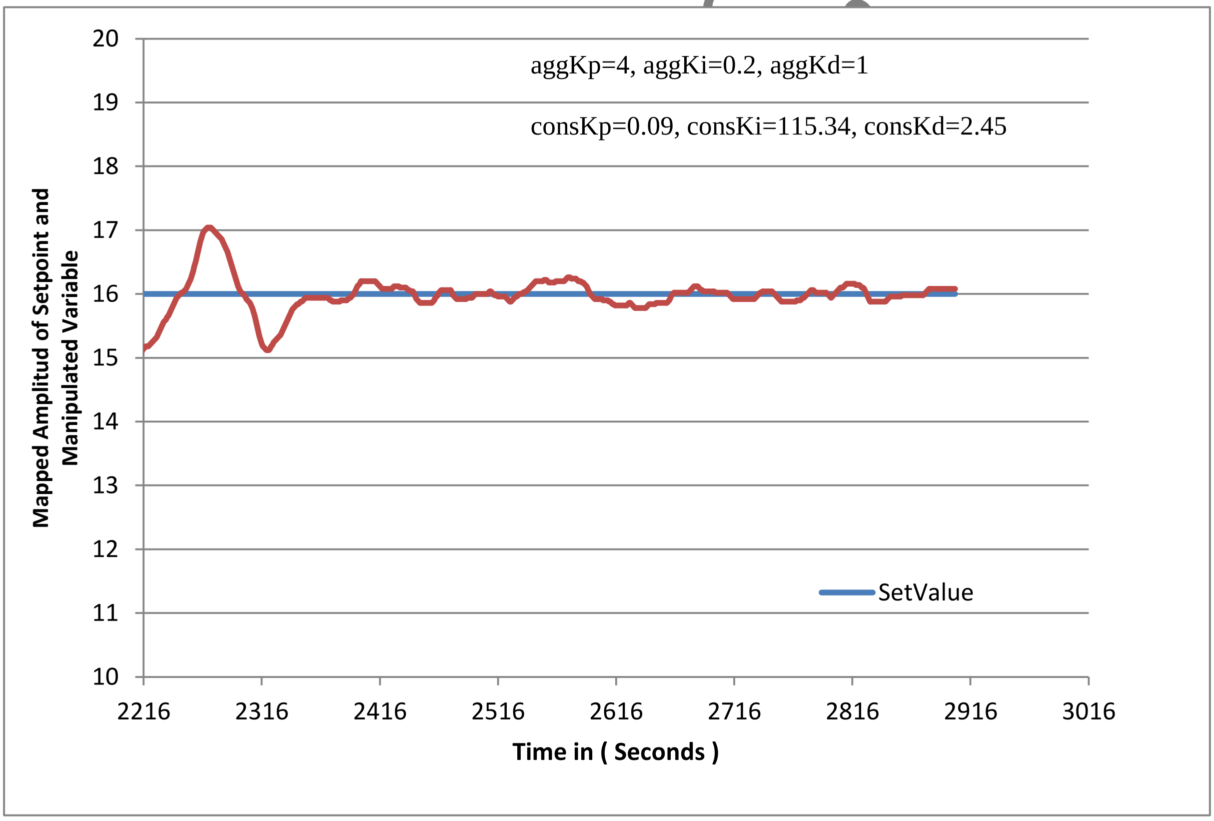

Fig 5.7 Transient Response of Adaptive PID Controller in Real Time

From above transient response it is clear that system having under damped response having damping ratio <1.

CONCLUSION

From the results obtained by software simulation and hardware implementation of Adaptive and manual tuned PID controller, it can be concluded that the response time parameters steady state error for adaptive tuned PID such as is slightly less that manual tuned PID controller, settling time of Adaptive PID controller is far better or Less than Manual PID Controller however, the response parameters like steady state error and maximum overshoot have been reduced significantly along with improved system performance. It has also been observed that by increasing the proportional and integral control constants, the oscillations in the graph also increases however the steady state error and offset is reduced. While on the other hand, by reducing these variables, the steady state error increases with decreased oscillations in the response. For optimal results, the value of the constants should be kept in a range that gives best response output with lesser oscillations at reduced offset.

REFERENCES

[1] I.D.Landau, "A Survey of Model Reference Adaptive Techniques-Theory and Applications ", Automatica, Vol. 10, pp 353-379, pergamon press, 1974, Printed in Great Britain.

[2] Kalpesh B. Pathak, Dipak M. Adhyaru (Member,IEEE), "Survey of Model Reference Adaptive Control", NUiCone- 012, Nirma University International Conference of Engineering ,pp 1-6 06-08 December, 2012

[3] Michel Van Dessel, "Control of an Industrial Process Using PID Control blocks in AutomationController",pp 1-9,2012.

[4] Kiam Heong Ang, Gregory Chong, Student and Yun Li, Member, IEEE, "PID Control System Analysis, Design, and Technology", IEEE Transactions on Control Systems Technology, Vol. 13, No. 4, pp 559-575, July 2005

[5] Aydogan Savran , Gokalp Kahraman, "A Fuzzy Model Based Adaptive PID Controller Design For Nonlinear And Uncertain Processes",Elsevier ISA Transactions 53, Issues 2, pp 280-288, March 2014

[6] R. Prakash, R. Anita, "Neuro- PI Controller Based Model Reference Adaptive Control for Nonlinear Systems", International Journal of Engineering, Science and Technology ,Vol. 3, No. 6, pp. 44-60, 2011

[7] Leonid Reznik, Omar Ghanayem, Anna Bourmistrov, "PID plus fuzzy controller structures as a design base for industrial applications", Elsevier Engineering Applications of Artificial Intellegence, volume 13, Issues 4, pp 419-430, August 2000

[8] Alireza Fereidouni , Mohammad A.S. Masoum and Moayed Moghbel "A New Adaptive Configuration of PID Type Fuzzy Logic Controller", Elsevier ISA Transactions, In Press Corrected Proof-Note to Users,pp 1-19, 18 December 2014

[9] Fauad Mrad, Sandra Hala Dandach, Sima Azar Ghassan Deeb, "Operator-Friendly Common Sense Controller with Experimental Verification using LabVIEW™ ", Proceedings of the IEEE International Symposium on Intelligent Control Limassol, Cyprus,pp 1-5, June 27-29,2005.

[10] Petr Husek and Otto Cerman , "Fuzzy Model Reference Control With Adaptation of Input Fuzzy Sets", Elsevier Knowledge Based Systems, volume 49,pp 116-122 September 2013.

[11] Kumpati S. Narendra and Jeyendran Balakrishnan, "Adaptive Control Using Multiple Models", IEEE Transactions on Automatic Control , Vol. 42, No. 2, pp 171-187, February 1997.

[12] George A. Rovithakis and Manolis A. Christodoulou, "Adaptive Control of Unknown Plants Using Dynamical Neural Networks", IEEE Transactions On System, Man And Cybernetics, Vol. 24, No. 3, pp 400- 412, March 1994.

[13] Kumpati S. Narendra and Jeyendran Balakrishnan, "Improving Transient Response of Adaptive Control Systems using Multiple Models and Switching", IEEE Transactions on Automatic Control, Vol. 39, No. 9, pp 1861-1866, September, 1994.

[14] Gang Tao and Petros A. Ioannou, "Model Reference Adaptive Control for Plants with Unknown Relative Degree", IEEE Transactions on Automatic Control, Vol. 38, No. 6, pp 96-982, June 1993.

[15] Wen-long Song and Jun Cao, "Model Reference Adaptive Sliding Mode Control of Uncertain Systems Subject to Input Nonlinearity", Advanced Motion Control, 9th IEEE International Workshop , Istanbul, pp 715-720, 2006.

[16] Jae Kawn Lee, Bo - Hyeok-Suh, Ken - Ichi Abe, "Model Reference Adaptive Control of Nonlinear System using Feedback Linearization", SICE '95. Proceedings of the 34th SICE Annual Conference. International Session Papers, Hokkaido, pp 1571-1576, 26-28 July 1995.

[17] Cervin, Henriksason, Lincoln, Eker and Arzen, "Analysis And Simulation of Timing Using Jitterbug and TrueTime", IEEE Control System Magazine, pp 16-30, June 2003.

[18] Gerhard Kreisselmeier and Brian D.O. Anderson, Fellow "Robust Model Reference Adaptive Control", IEEE Transactions on Automatic Control, Vol.-31, No. 2., pp 127-133, February 1986.

[19] Hans Petter Halvorsen, "Data Acquisition in LabVIEW" Tutorial, Telemark University College, Department of Electrical Engineering, Information Technology and Cybernatics, Norvey, August 2013.