www.s2is.org
International Journal on Smart Sensing and Intelligent Systems
ISSN 1178-5608

PLC BASED ADAPTIVE FUZZY PID SPEED CONTROL OF DC BELT CONVEYOR SYSTEM

Mohd Heidir Mohd Shah[1], Mohd Fua'ad Rahmat[2], Kumeresan A. Danapalasingam[2] and Norhaliza Abdul Wahab[2]

[1]Department of Industrial Electronic,
German Malaysian Institute, 43000 Kajang,
Selangor, Malaysia
Emails: mohdheidir@gmi.edu.my

[2]Department of Control and Mechatronics Engineering
Faculty of Electrical Engineering, Universiti Teknologi Malaysia,
UTM Skudai, 81310 Johor
Malaysia
Emails: fuaad@fke.utm.my
Emails: kumeresan@fke.utm.my
Emails: aliza@fke.utm.my

Submitted: Apr. 4, 2013 ***Accepted: May 3, 2013*** ***Published: June 5, 2013***

Abstract- Conveyor belt system is one of the most common transfer systems used in industry to transfer goods from one point to another in a limited distance. It is used in industries such as the electro-mechanical/mechanical assembly manufacturing to transfer work piece from one station to another or one process to another in food industries. The belt conveyor system discussed in this paper is driven by a DC motor and two speed controllers. The PID speed controller is designed to provide comparison to the main controller which is the Adaptive Fuzzy PID Speed controller. Both controllers are implemented in a real hardware where the algorithm will be written in PLC using SCL language. The experimental result shows that Adaptive Fuzzy PID controller performs better and adapted to the changes in load much faster than the conventional PID controller. This project has also proved that PLC is capable of performing high level control system tasks.

Index terms*:* DC belt conveyor system, speed controller, adaptive fuzzy PID, programmable logic controller, structured control language.

I. INTRODUCTION

Belt conveyor system is a belt driven transfer or transport system used to transfer or transport goods or material from one point to another within a limited distance. It is used widely in industries such as the electro-mechanical assembly, food manufacturing, coal mining, etc. Belt conveyor is a preferred transfer system in industries compared to the robotic arm and pneumatic or hydraulic pick and place system because of its simple design, light weight, cost effective, requires less maintenance and the potential to achieve high efficiency. However, this system is inherently difficult to control due to belt flexibility, vibration, friction, system uncertainties and nonlinearities [1]. Furthermore, the characteristic of the system might also vary when different loads are applied to the system.

One of the demands for the modern conveyor is the operation with variable speed. Most of the literatures found on speed control of belt conveyor system were focusing on energy efficiency in transfer system used in coal mining industries. Ristic and Jeftenic in [2] and Pang and Lodewijks in [3] implement Fuzzy Speed Control to adjust the belt speed at the occasion when the loading rate reaches certain point. Their simulation result shows an estimation of 4% energy efficiency improvement and this was verified by the practical measurement of the belt conveyors at a large bulk material terminal. The Fuzzy Speed Control has also been implemented by Yilmaz et al, in [4] to control two conveyors used in product packaging.

Controlling the speed of conveyor can be achieved by controlling the speed of its drives unit, in this case the DC motor. Therefore, literatures that implement speed control on DC motor with similar control objectives have also been reviewed. Arrofiq and Saad in [5] compare performance of the Self Tuning PI Fuzzy Logic Controller (STPIFLC) with PI Fuzzy Logic Controller (PIFLC). The results shows that although STPIFLC produced higher overshoot, the response is much faster compared to the PIFLC. Ji and Li in [6] have integrated the neural network and the traditional PID to constitute DC motor speed control system based on Back Propagation neural network self-tuning parameters PID control. Simulation results show that the self-tuning controller has better performance compared to the traditional PID controller in terms of lower overshoot, faster rise time and settling time under steady load. It has also shown that the self-tuning controller adapted to the change in load faster. Payakkawan et al, in [7] and El-Gammal and El-Samahy in [8] have shown that the PID controller for speed control of DC motor can be optimized by applying online tuning based on Particle Swarm Optimization (PSO) under the load torque disturbances and change of reference speed. Another control strategy that has gained popularity for speed control of DC motor is the Adaptive Fuzzy PID (AFPID) controller. Kandiban and Arulmozhiyal in [9], Yu et al, in [10] and Feng and Qian in [11] have all developed the said controller to control the speed of DC motor and their simulation results shown that the AFPID has a better performance compared to the traditional PID controller. This is proven as a lower overshoot and, faster settling and rise times are obtained.

In this paper, the control strategy is implemented in Programmable Logic Controller (PLC). PLC can be defined as a microprocessor-based control device with the original purpose of supplementing relay logic. In the early days, PLC can only perform logical operations [12]. PLC has been used extensively in industrial applications for control for decades due to their high reliability and robust architecture. The newest PLCs have moved past just a robust platform into a new realm of high computational power and processor speed. These, along with the PLC's highly expandable layout, make it an ideal platform for far beyond the classical applications. These new applications include implementing computational intelligence based modeling, optimization and control techniques that require fast processing power to be executed in real-time [13].

Several literatures have been found to have used PLC as a medium to implements their control strategies. Ghandakly et al, in [14] have implemented an adaptive controller for DC motor on Allen Bradley PLC5 system. Simple least square algorithm was used which only requires basic arithmetic operation. Therefore the programming language, the traditional ladder logic (LAD) has enough capability to perform that algorithm. Yilmaz et al, in [4] have used different type of PLC to implement fuzzy control to control the speed of belt conveyor system. The PLC used was OMRON PLC which requires additional Fuzzy logic unit, C200H-FZ001 to implement the fuzzy logic control. Special software was also required to program the knowledge base and a subprogram for ladder diagram must be prepared for transferring data between the PLC and the Fuzzy Logic unit. The programming language used for the fuzzy knowledge base was not discussed.

Structured Text Language (STL) is standard PLC programming language that has the capability to perform high level operation. Ferdinando in [15] has used that programming language in PLC TSX 37-21 medium to control the speed of DC motor using fuzzy logic. The program was made easy by additional plugin PL7 FUZZ installed. With this plugin, the shape of the membership function can be specified in graphical form.

Yulin in [16] and Velagic et al, in [17] have implement PI controller for the Permanent Magnet DC Motor using Siemens S7-200 PLC. In both literatures, a standard, pre-programmed PID subroutine was used. The use of this subroutine was easy as there was no programming needed for the control algorithm. They only need to assign PID parameter to the ladder block generated and specifying the address of set-point and output.

Junjie et al, in [18] shows how PID algorithm can be written in higher range of Siemens PLC which is S7-300 without using the standard pre-programmed block. This was achieved by using the Structured Control Language (SCL), a high level programming language for PLC similar to PASCAL. Although PID is considered as traditional controller, the usage of SCL has provided the base for more advanced control technique which is why the same language are used in this paper.

II. CONTROLLER DESIGN

This paper presents two types of controller for speed control of DC Belt Conveyor System which are the PLC Based PID Controller and the PLC Based Adaptive Fuzzy PID Controller.

a. PLC Based PID Speed Controller

The structure of PLC Based PID Controller is as shown in Figure 1 with the output equation as follows,

$$u(t) = K_P e(t) + K_I \sum_{t=0}^{T} e(t) + K_D \left(e(t) - e(t-1)\right) \tag{1}$$

where,

$$K_I = K_P \frac{T_s}{T_I} \tag{2}$$

and,

$$K_D = K_P \frac{T_D}{T_s} \tag{3}$$

where K_P, K_I and K_D is the gain for proportional, sum of error and change of error respectively, T_I and T_D is the integral and derivative time and T_s is the sampling time. From Figure 1, u(t) is the input of belt conveyor system, e(t) is the error and T is the maximum sequence. Items in the dotted box are programmed in the PLC where the FIR filter is used to reduce noise and amplify tachogenerator's output signal.

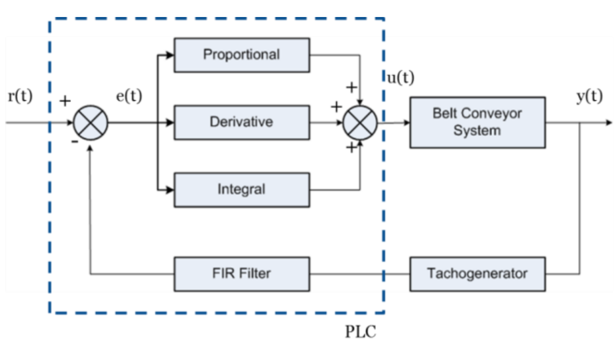


Figure 1. Structure of PLC Based PID Speed Controller

The control algorithm is written using a combination of SCL and LAD programming language. Programming structure of the controller is illustrated in Figure 2.

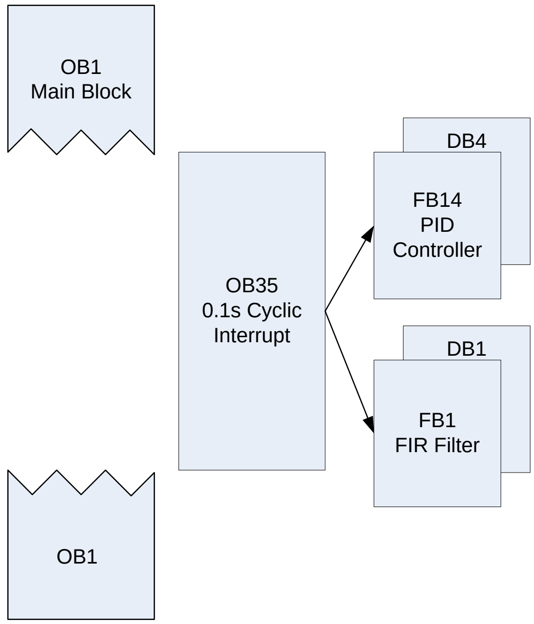
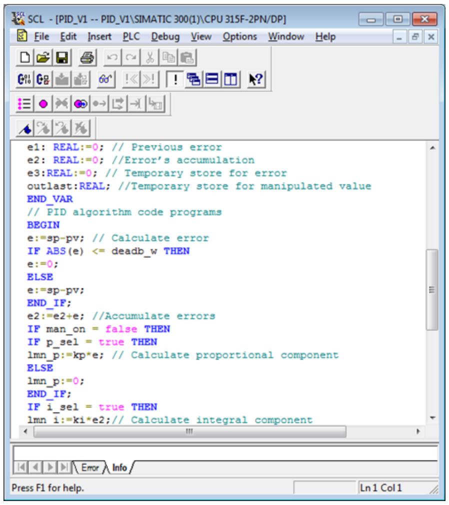


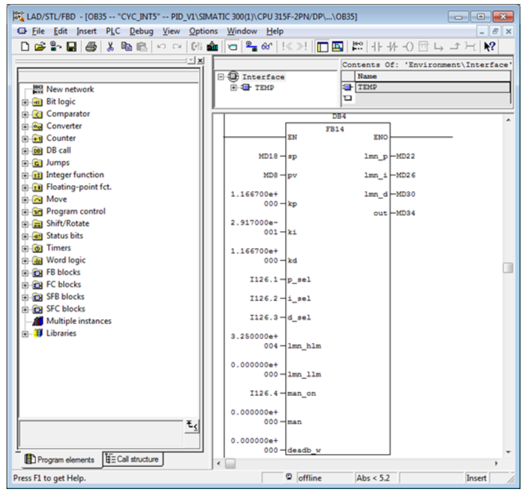
Figure 2. PLC's Structured Programming for PID Controller

The CPU in PLC works by first scanning the main programming block which is the organization block 1 (OB1). Therefore, the OB1 must be used and downloaded into PLC. However, the main algorithm cannot be written or called in OB1 because it is required to be processed in a fixed interval time while OB1 is scanned and processed by the CPU continuously. The OB1 is used in this project just to perform input handlings and data conversions and it was programmed in LAD. OB35 is one special programming block that has the capability to perform 0.1s cyclic interrupt. In other words, at every 0.1s, the program in OB35 will be scanned and processed by the CPU.

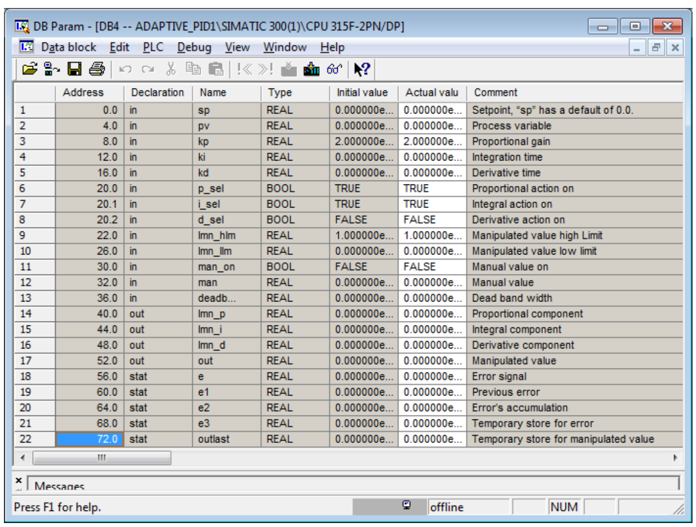
The PID and FIR filter algorithm were not written directly in OB35, they were programmed in 2 different Function Block (FB). The use of FB is attractive because it has features like variable parameterizing and an instance Data Block (DB) which will be auto generated as a temporary variable storage. In this case, the PID and FIR filter algorithms are written in FB14 and FB1 respectively using SCL programming language. In order to use SCL, an SCL source in the form of text editor needs to be created first. The complete program can then be compiled and converted into LAD block diagram. The DB generated for PID controller and FIR Filter was DB4 and DB1 respectively. Figure 3 shows a partial SCL program for PID Controller in SCL text editor, the parameterized FB in LAD environment of OB35 and the DB generated.



(a)



(b)



DB Param - [DB4 -- ADAPTIVE_PID1\SIMATIC 300(1)\CPU 315F-2PN/DP]

	Address	Declaration	Name	Type	Initial value	Actual valu	Comment
1	0.0	in	sp	REAL	0.000000e...	0.000000e...	Setpoint, "sp" has a default of 0.0.
2	4.0	in	pv	REAL	0.000000e...	0.000000e...	Process variable
3	8.0	in	kp	REAL	2.000000e...	2.000000e...	Proportional gain
4	12.0	in	ki	REAL	0.000000e...	0.000000e...	Integration time
5	16.0	in	kd	REAL	0.000000e...	0.000000e...	Derivative time
6	20.0	in	p_sel	BOOL	TRUE	TRUE	Proportional action on
7	20.1	in	i_sel	BOOL	TRUE	TRUE	Integral action on
8	20.2	in	d_sel	BOOL	FALSE	FALSE	Derivative action on
9	22.0	in	lmn_hlm	REAL	1.000000e...	1.000000e...	Manipulated value high Limit
10	26.0	in	lmn_llm	REAL	0.000000e...	0.000000e...	Manipulated value low limit
11	30.0	in	man_on	BOOL	FALSE	FALSE	Manual value on
12	32.0	in	man	REAL	0.000000e...	0.000000e...	Manual value
13	36.0	in	deadb...	REAL	0.000000e...	0.000000e...	Dead band width
14	40.0	out	lmn_p	REAL	0.000000e...	0.000000e...	Proportional component
15	44.0	out	lmn_i	REAL	0.000000e...	0.000000e...	Integral component
16	48.0	out	lmn_d	REAL	0.000000e...	0.000000e...	Derivative component
17	52.0	out	out	REAL	0.000000e...	0.000000e...	Manipulated value
18	56.0	stat	e	REAL	0.000000e...	0.000000e...	Error signal
19	60.0	stat	e1	REAL	0.000000e...	0.000000e...	Previous error
20	64.0	stat	e2	REAL	0.000000e...	0.000000e...	Error's accumulation
21	68.0	stat	e3	REAL	0.000000e...	0.000000e...	Temporary store for error
22	72.0	stat	outlast	REAL	0.000000e...	0.000000e...	Temporary store for manipulated value

(c)

Figure 3 PID programming in (a) SCL text editor, (b) parameterized FB in OB35 and (c) DB for temporary variable storage

The PID parameters were obtained via Ciancone correlation technique with fine tuning, and these parameters will be the initial values for the next controller.

b. PLC Based Adaptive Fuzzy PID Speed Controller

One of the main objectives in this project is to design and implement a controller that has improved performance compared to the traditional PID controller, therefore, an adaptive tuning of PID controller via Fuzzy Logic was proposed. The proposed Adaptive Fuzzy PID (AFPID) controller structure is as shown in Figure 4.

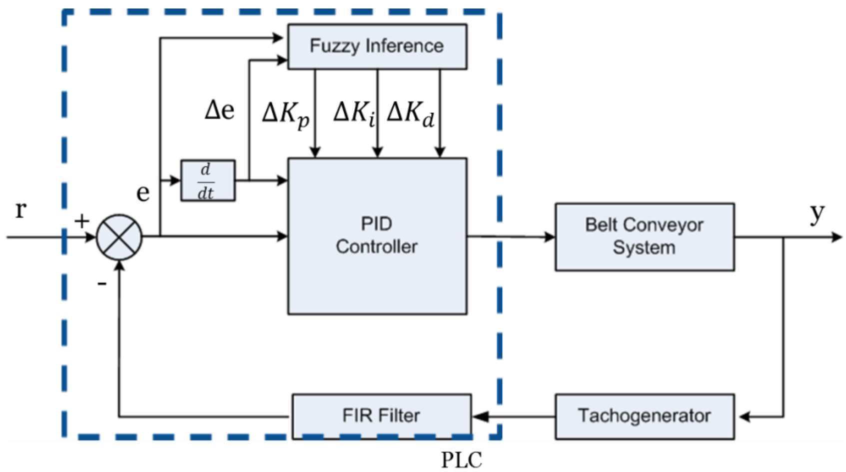
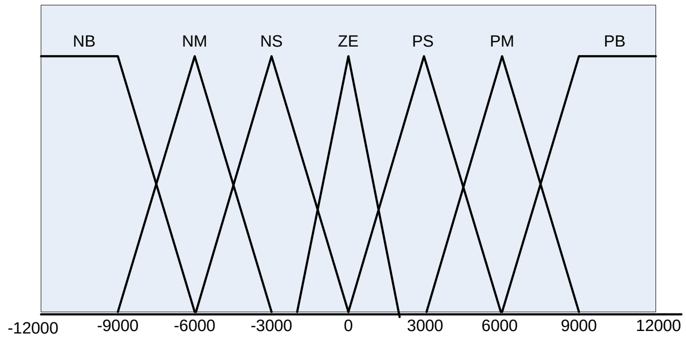


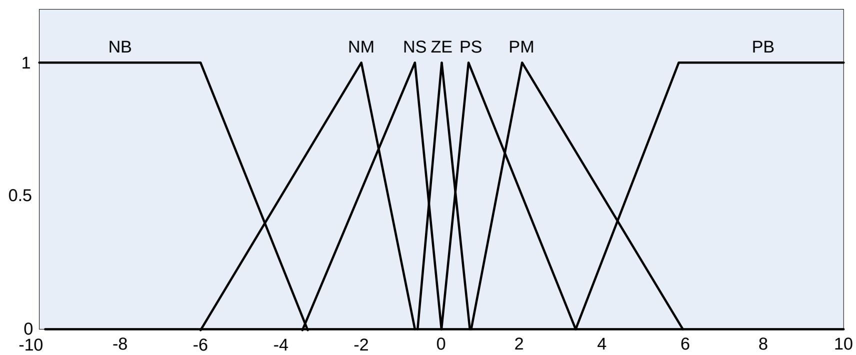
Figure 4. Structure of PLC Based AFPID Controller

In this controller, the PID parameters previously obtained via Ciancone Correlation were used as the initial value. The Fuzzy Inference System (FIS) is developed to adjust the PID parameters around their initial values based on the output error and the changes rate. That makes the FIS to have 2 inputs which are the error (e) and change of error (Δe) and 3 outputs which are the changes of Proportional gain (ΔK_P), changes of Integral gain (ΔK_I) and changes of Derivative gain (ΔK_D).

In the PLC, the range of analog I/O that can be read or generated is -10V – 10V. This signal is read as a 16bit integer value representation for example 0V, 5V and 10V are represented as 0, 16384 and 32768 in integer respectively. Therefore, the inputs membership function must be assigned according to the integer values as shown in Figure 5(a). From the experiment of the PLC based PID Controller, the ranges for the output functions should be $\Delta K_P \in [-1,1]$, $\Delta K_I \in [-0.02,0.02]$ and $\Delta K_D \in [-2.5,2.5]$ with the interval set at $[-10,10]$. The FIS outputs with gains are $\Delta K_P = 0.1\Delta K'_P$, $\Delta K_I = 0.002K'_I$ and $\Delta K_D = 0.25\Delta K'_D$. Membership function for the output is as shown in Figure 5(b). Max-Min inference system and Centroid Defuzzification technique were employed in the controller.



(a)



(b)

Figure 5. Membership functions for (a) e and Δe, (b) ΔK_P, ΔK_I and ΔK_D

The control algorithms were written in similar manner to the one in section (a) of this chapter. The PID algorithm developed earlier is retained, while fuzzy logic algorithm is added. Figure 6 shows the structure of AFPID program. Note that, 8 more programming blocks were added to the original PID algorithm developed earlier. In this program, fuzzy control algorithm was organized in FB12 using LAD and DB3 is generated for its temporary variable's storage. This was done so that the fuzzy logic control system can be divided into several functions. All the functions called in FB12 were written in SCL, in this case, the input membership functions were written in FC2 for trapezoid function and FC3 for triangular function. FC is another programming block in S7-300 that can be parameterized but DB will not be generated for this block. The Max-Min inference system for K_P , K_I and K_D were written in FC10, FC12 and FC14 respectively and the centroid defuzzification is written in FC15. Saturation is added and programmed in FC16 done to limit the values according to the membership ranges specified earlier especially during the first step response, the value will be very big and out of range.

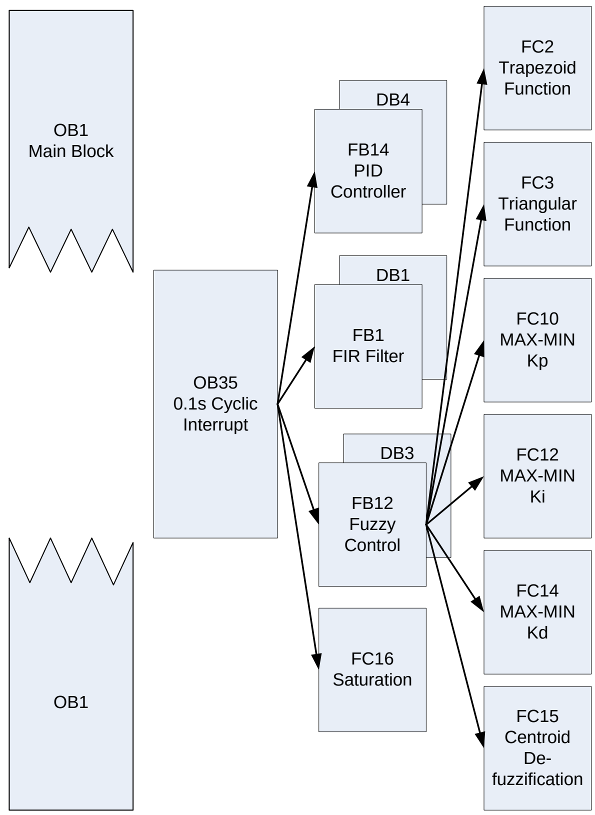


Figure 6. PLC's Program Structure for AFPID Speed Controller

III. EXPERIMENTAL SETUP

The experimental setup is shown in Figure 7. The highly modular Siemens S7-300 PLC is used. It is equipped with 24V, 5A power supply, 315F-2PN/DP CPU module, 4 analog channel input and 2 channel analog output module, 32 digital input channel, 16 digital output channel, an AS-I communication processor and an Ethernet switch which used to load the program in programming station into PLC. 0-10V potentiometer is used as set-point variable which is supplied to channel 1 of PLC's analog input module. It is also connected to channel 1 of oscilloscope for measurement. The output from the controller is supplied to the driver of the belt conveyor system from channel 1 of PLC's analog output module. The speed response measured by tachogenerator is fed back to the controller via channel 2 of PLC's analog input module.

Tachogenerator's signal is then filtered and amplified in PLC and supplied back to the channel 2 of oscilloscope.

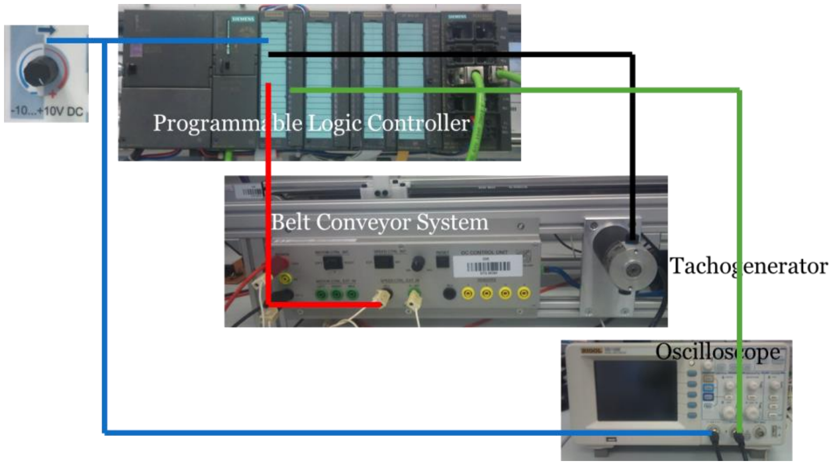


Figure 7. Experimental Setup for both PID and AFPID Speed Controller

IV. RESULTS AND DISCUSSIONS

This chapter presents the PLC based experimental results for both PID and AFPID controller. The experiments were done on a real DC belt conveyor system at three modes of speed. The modes are low speed set-point at 2.12 V or equivalent to 395 rpm, the medium speed at 4.4 V or equivalent to 819 rpm and the high speed at 6.64 V or equivalent to 1,236 rpm. The experiments were also done in four different initial load condition ranges from 0 kg to 3 kg to observe how the load can affect performance of both controllers. The results are as shown in Figure 8, 9 and 10 and the data analysis is as shown in Table 1 and 2.

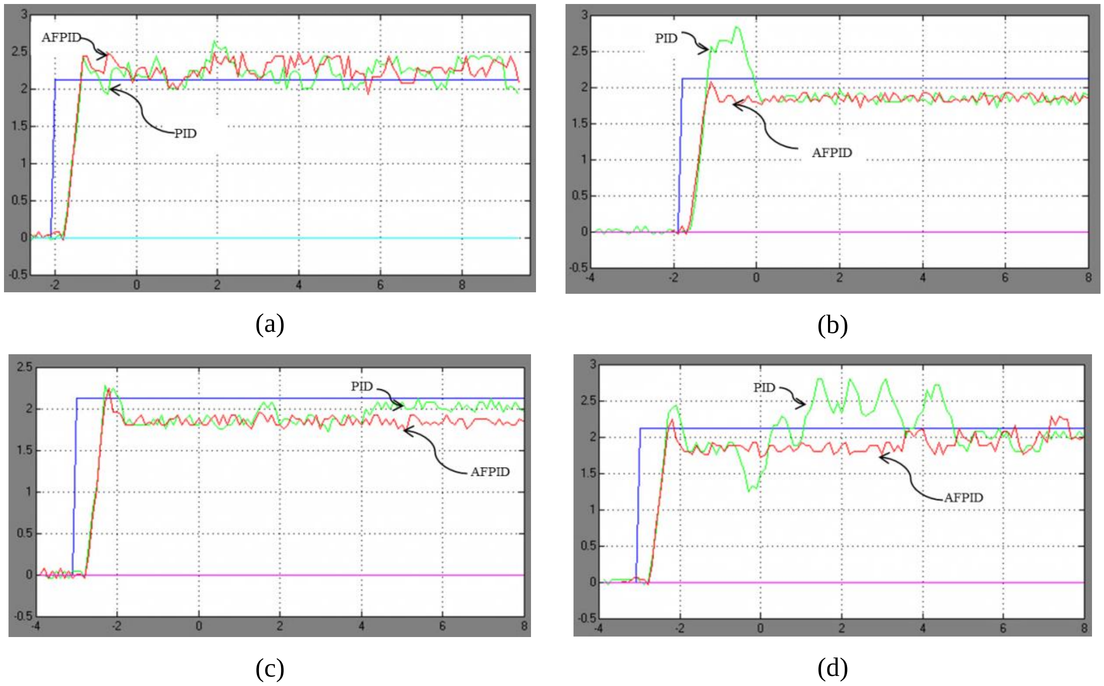


Figure 8. PLC based controller response to low speed set-point with initial load conditions of (a) 0 kg, (b) 2 kg, (c) 2.5 kg and (d) 3 kg.

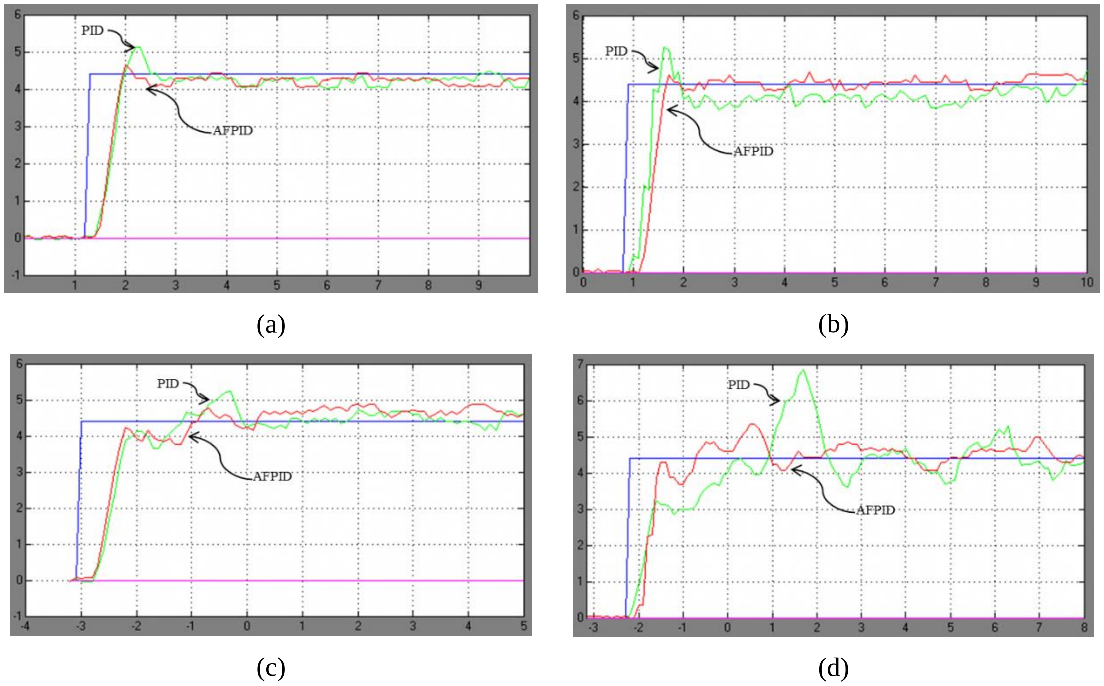
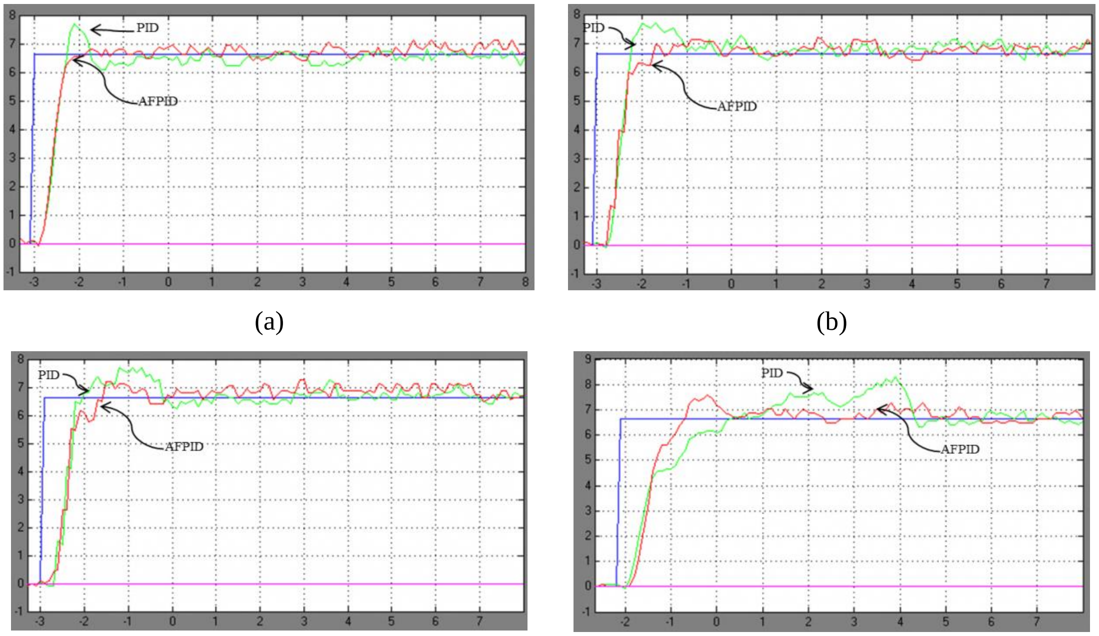


Figure 9. PLC based speed controller response to medium speed set-point with initial load conditions of (a) 0 kg, (b) 2 kg, (c) 2.5 kg and (d) 3 kg.



(a) (b)

© (d)

Figure 10. PLC based speed controller response to high speed set-point with initial load conditions of (a) 0 kg, (b) 2 kg, (c) 2.5 kg and (d) 3 kg.

Table 1. PLC based PID controller performance analysis

Set-point V(rpm)	**Load (kg)**	t_r **(s)**	t_s **(s)**	**%OS**	**SSE**
2.12V (395rpm)	0.0	0.50	0.80	15.09	94.48
	2.0	0.56	1.88	33.96	138.46
	2.5	0.65	1.22	7.55	111.68
	3.0	0.62	8.00	32.08	155.78
4.40V (819rpm)	0.0	0.60	1.23	16.36	409.89
	2.0	0.54	0.95	19.09	414.92
	2.5	0.75	2.91	19.09	485.11
	3.0	0.58	8.00	55.45	793.89
6.64V (1236)rpm	0.0	0.60	1.32	17.47	956.03
	2.0	0.67	1.89	15.66	1079.56
	2.5	0.70	2.56	15.66	1096.49
	3.0	0.77	6.40	25.30	1448.70

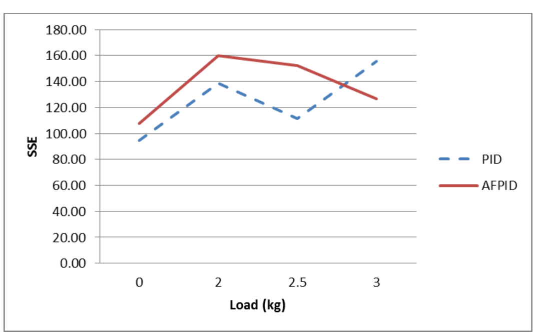
Table 2. PLC based AFPID controller performance analysis

Set-point V(rpm)	Load (kg)	t_r (s)	t_s (s)	%OS	SSE
2.12V (395rpm)	0.0	0.47	0.80	15.09	107.57
	2.0	0.59	0.84	0.00	159.98
	2.5	0.65	0.97	5.66	152.26
	3.0	0.63	0.97	5.67	126.83
4.40V (819rpm)	0.0	0.55	0.76	5.45	444.20
	2.0	0.66	0.78	4.55	430.82
	2.5	0.67	1.97	8.18	487.05
	3.0	0.63	2.53	21.82	455.36
6.64V (1236)rpm	0.0	0.64	0.76	4.55	979.44
	2.0	0.71	0.91	7.23	1008.93
	2.5	0.65	1.22	8.43	1149.75
	3.0	0.77	2.16	14.46	1122.67

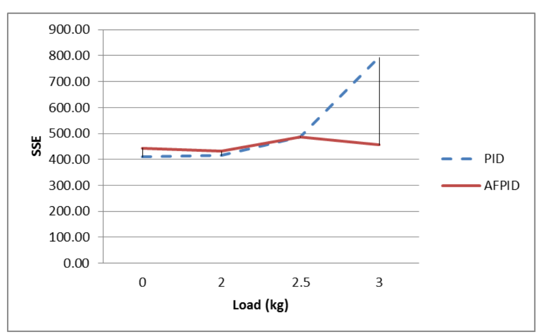
The controller response shows that at low speed performance of both controllers were not much different when no load is applied initially. When 2 kg of load is applied, PID controller produced high overshoot while better performance is observed for AFPID. Both controllers perform in similar manner when 2.5 kg load is applied. PID controller started to become unstable when 3 kg load is applied. Similar trend can be observed in medium and high speed set-point.

In this experiment, set-point tracking performance is measured using sum of square error (SSE). From the data, it can be observed that PID controller performs slightly better in set-point tracking except when 3 kg load is applied to the system. This can be observed in Figures 11 and 12.

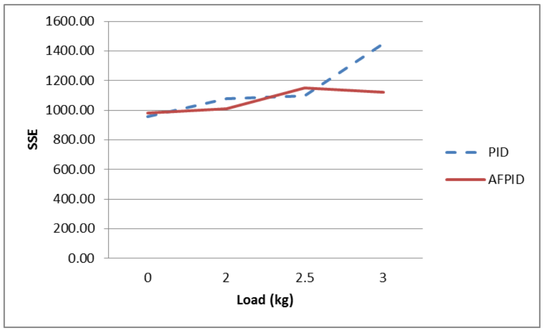
.



(a)

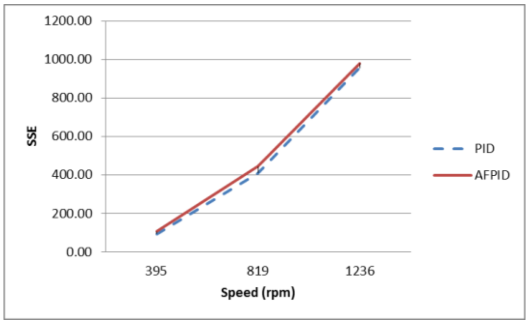


(b)

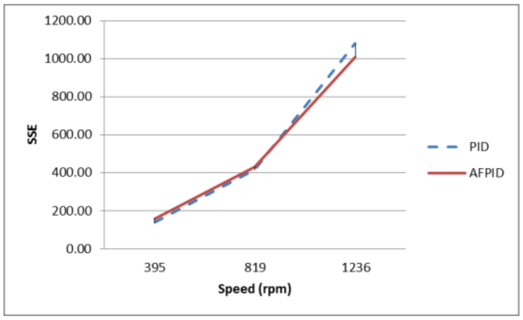


(c)

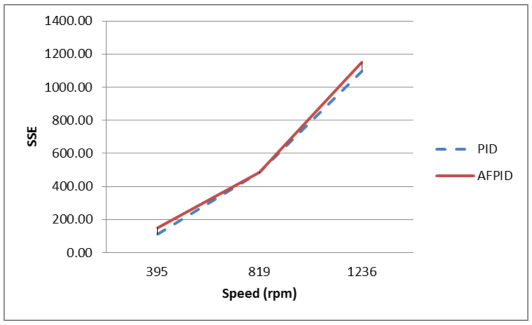
Figure 11. Tracking Performance of PLC based PID versus PLC based AFPID against load at (a) Low speed, (b) Medium speed and (c) High speed



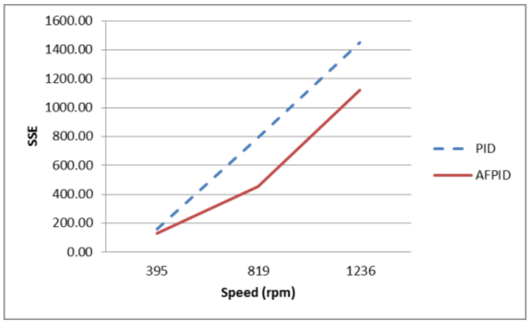
(a)



(b)



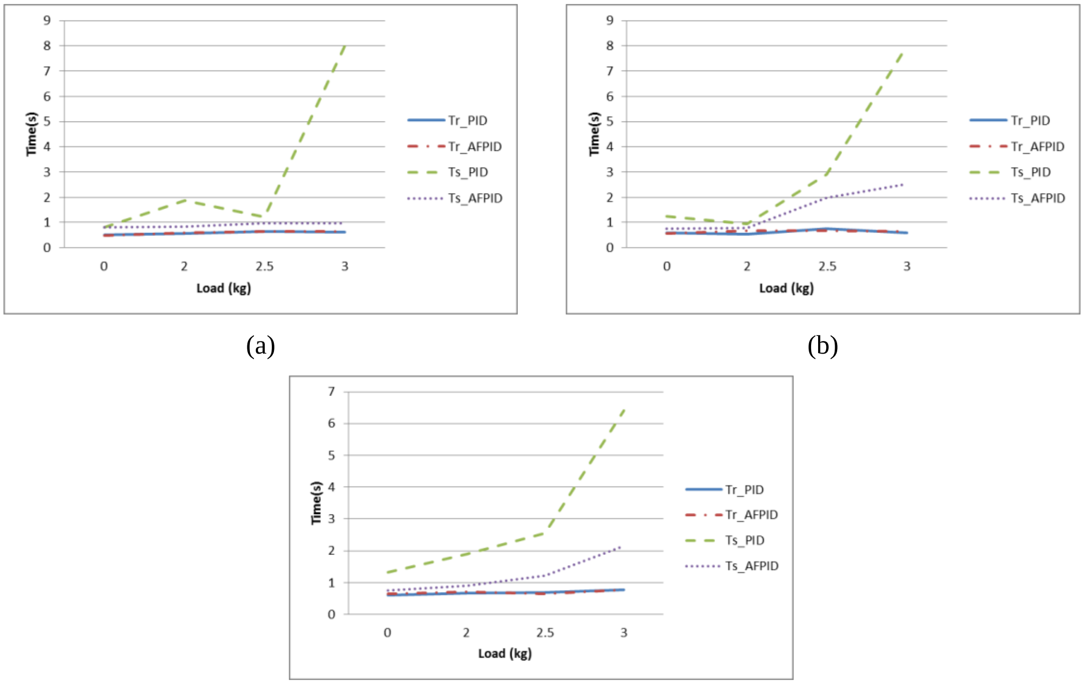
(c)



(d)

Figure 12. Tracking Performance of PLC Based PID versus PLC based AFPID against speed. with initial load conditions of (a) 0 kg, (b) 2 kg, (c) 2.5 kg and (d) 3 kg.

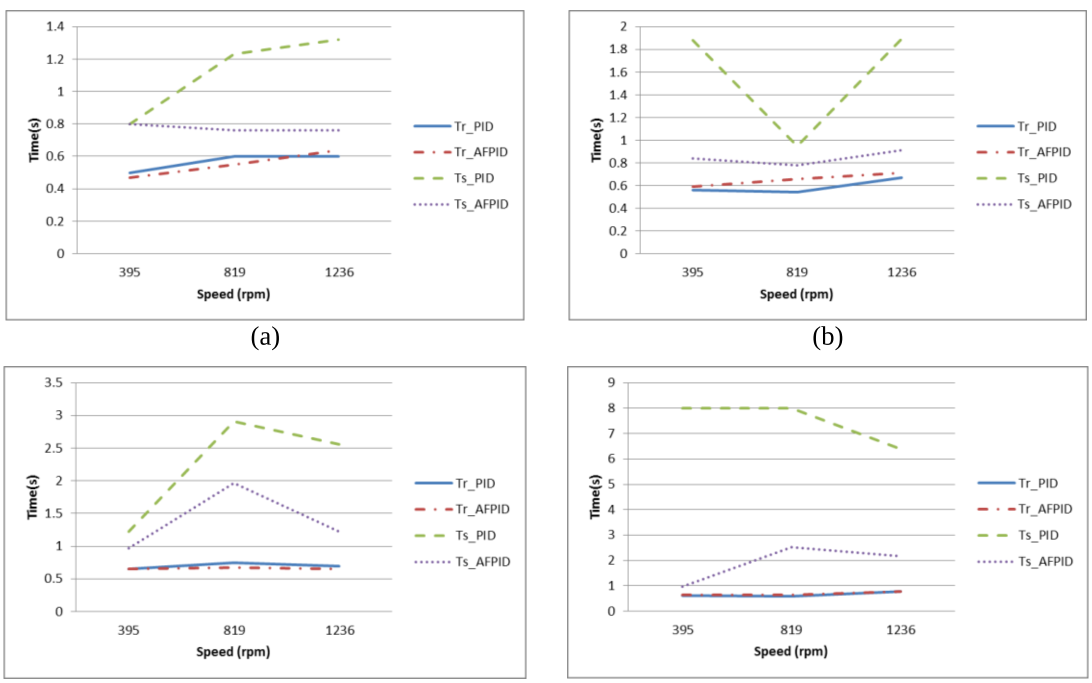
Data in the table also shows that both controllers have almost the same rise time at every experiment but huge different can be observed for the settling time. AFPID takes faster time to settle compared to PID and the margin is bigger as the speed and the load are increased. These trends are illustrated in Figure 13 and 14.



(a) (b)

(c)

Figure 13. PLC Based PID and AFPID response, time against load at
(a) Low Speed, (b) Medium Speed and (c) High Speed.

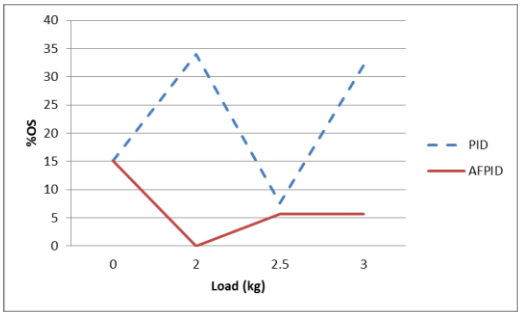


(a) (b)

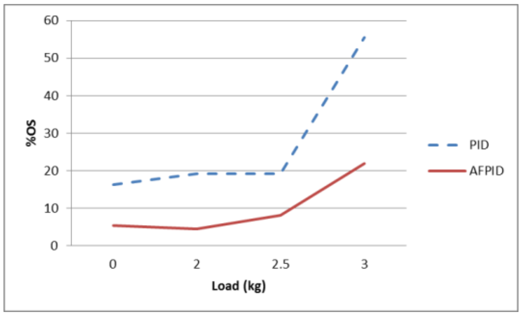
(c) (d)

Figure 14. PLC Based PID and AFPID response, time against speed with initial load condition of
(a) 0 kg, (b) 2 kg, (c) 2.5 kg and (d) 3 kg.

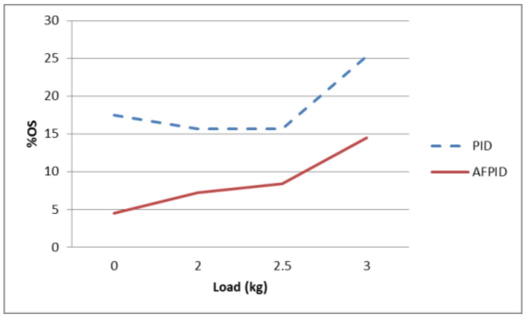
Another performance criterion that has been measured is the overshoot. It can be seen from data in Table 1 and 2 that AFPID has lower overshoot compared to PID at all speeds and loads. Data in the table is plotted in Figure 15 and 16.



(a)

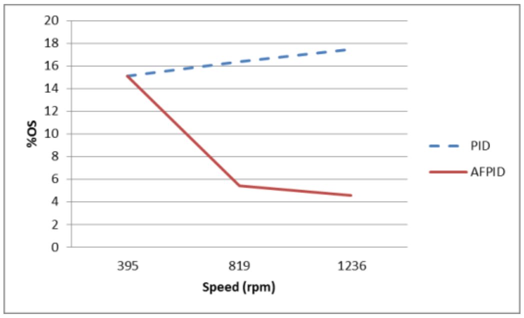


(b)

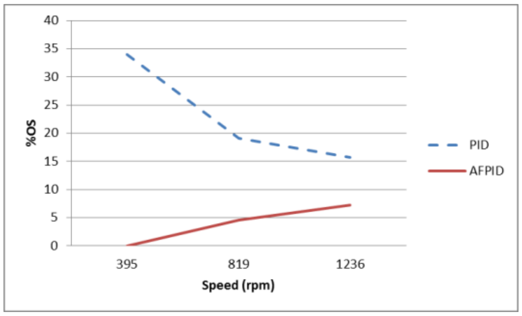


(c)

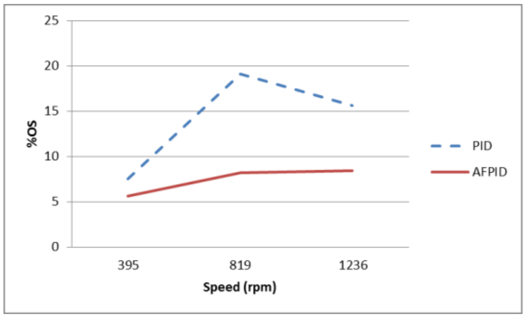
Figure 15. PLC Based PID and AFPID overshoot against Load at
(a) Low Speed, (b) Medium Speed and (c) High Speed.



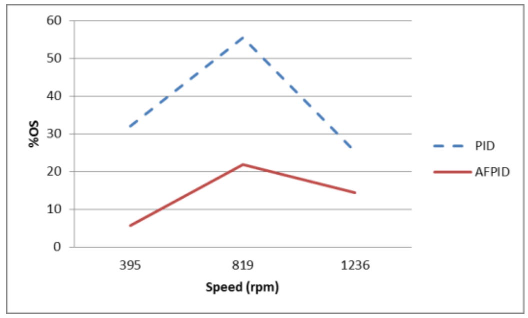
(a)



(b)



(c)



(d)

Figure 16. PLC Based PID and AFPID overshoot against speed with initial load condition of (a) 0 kg, (b) 2 kg, (c) 2.5 kg and (d) 3 kg.

V. CONCLUSION

An Adaptive Fuzzy PID Controller for speed control of DC Belt Conveyor System has been developed. The experimental results have shown that overall this controller performs better than a conventional PID controller. However, the PID controller has also shown better performance than AFPID in set-point tracking. The fuzzy logic online tuning procedure has proven that it can adapt to changes in load faster than the PID controller alone. Both PID and AFPID controllers have also been implemented in PLC. The usage of PLC is very attractive due to its reputation of having most reliable control device especially in automation industries. The experimental results have shown that PLC has the capability to perform high level control architecture. This project has opened up a window for more research to be done in the PLC environment for implementations of advanced control techniques.

REFERENCES

[1] C. Mitsantisuk, S. Katsura and O. Kiyoshi, "Sensorless Interaction Force Control Based on Modal Space Design for Twin Belt-driven System," in *SICE Annual Conference*, Japan, 2007.

[2] B. Jeftenic, L. Ristic, M. Bebic and S. Statkic, "Controlled Induction Motor Drives Supplied by Frequency Converters on Belt Conveyors - Modeling and Commissioning," *35th Annual Conference of IEEE in Industrial Electronics*, pp. 1063 - 1068, 2009.

[3] Y. Pang and G. Lodewijks, "Improving Energy Efficiency in Material Transport Systems by Fuzzy Speed Control," in *3rd IEEE International Symposium for Logistics and Industrial Informatics*, Budapest, Hungary, 2011.

[4] S. Yilmaz, B. Cakir, A. Gedik and H. Dincer, "Speed Control of a Conveyor System by Means of Fuzzy Control Aided PLC," in IEEE International Symposium on Industrial Electronics, 1999.

[5] M. Arrofiq and N. Saad, "A Simulation of PLC-based Self-tuning Fuzzy Logic Controller for DC Motor," in International Symposium on Information Technology, 2008.

[6] H. Ji and Z. Li, "Design of Neural Network PID Controller Based on Brushless DC Motor," in Second International Conference on Intelligent Computation Technology and Automation, 2009.

[7] P. Payakkawan, K. Klomkarn and P. Sooraksa, "Dual-line PID Controller Based on PSO for Speed Control of DC Motors," in 9th International Symposium on Communications and Information Technology, 2009.

[8] A. A. A. El-Gammal and A. A. El-Samahy, "Adaptive Tuning of a PID Speed Controller for DC Motor Drives Using Multi-Objective Particle Swarm Optimization," in 11th International Conference on Computer Modelling and Simulation, 2009.

[9] R. Kandiban and R. Arulmozhiyal, "Design of Adaptive Fuzzy PID Controller for Speed Control of BLDC Motor," International Journal of Soft Computing and Engineering , vol. 2, no. 1, pp. 386 - 391, 2012.

[10] B. Yu, H. Zhu and C. Xue, "Research on Adaptive Fuzzy PID Synchronous Control Strategy of Double-Motor," International Journal of Intelligent Syatems and Application, vol. 5, pp. 28 - 33, 2011.

[11] Y. Feng and H. Qian, "Research on Speed-Control of BLDCM Based on Intelligent Control Algorithm," in International Conference on Computer Application and System Modeling, 2010.

[12] O. Karasakal, E. Yesil, M. Guzelkaya and I. Eksin, "Implementation of a New Self-Tuning Fuzzy PID Controller on PLC," Turk J Elec Engin, vol. 13, no. 2, pp. 277 - 286, 2005.

[13] C. Parrott and G. K. Venayagamoorthy, "Real-Time Implementation of Intelligent Modeling and Control Techniques on a PLC Platform," Industry Applications Society Annual Meeting, pp. 1 - 7, 2008.

[14] A. A. Ghandakly, M. E. Shields and M. E. Brihoum, "Design of an Adaptive Controller for a DC Motor Within an Existing PLC Framework," Conference Record of the 1996 IEEE, vol. 3, pp. 1567 - 1574, 1996.

[15] H. Ferdinando, "The Implementation of Low Cost Fuzzy Logic Controller for PLC TSX 37-21," in International Conference on Intelligent and Advanced Systems, 2007.

[16] D. Yulin, "The Analysis and Implement of PLC-based PI Control for the Permanent Magnet DC Motor," in Second International Conference on Communication Systems, Networks and Applications, 2010.

[17] J. Velagic, K. Obarcanin, E. Kapetanovic, S. Huseinbegovic and N. Osmic, "Design of PLC-based PI Controller for the Permanent Magnet DC Motor under Real Constraints

and Disturbances," in IEEE Control Applications & Intelligent Control, 2009.

[18] R. Junjie, Z. Quan and L. Hongxing, "Structured Control Language and its Application in Control Algorithm Programming," in International Conference on Measuring Technology and Mechatronics Automation, 2010.