ISSN (Print) : 2319-8613
ISSN (Online) : 0975-4024

Development of Firefly based I-PD Controller on Embedded Platform

S Meena[1], KChitra[2], R Vijayanand[3]
Research scholar, Faculty of Electronics engineering, SathyabamaUniversity, Chennai-600119, India[1].
Professor, SENSE,VIT University,Chennai,India[2].
Department of EIE, St. Joseph's College of Engineering, Chennai-600119, India[3].
meena27681@gmail.com[1], kchitra68@gmail.com[2], r.vijayanandrva@gmail.com[3]

***Abstract*—A PID controller is a versatile feedback mechanism in most of the industrial process due to its simplicity in structure and reliability, while the I-PD controller and PI-D are the variants of PID controllers which are particularly designed to reduce the influence of changes in the reference signal on controlled output. In this paper the one of the variants, I-PD is tuned with a modified approach of firefly algorithm. It has been implemented in Arduino (genuino UNO) board so that the controller can be used as a standalone controller with the processes where space as a constraint. Also the cost of the controller is very much reduced.The I-PD controller has been implemented in Arduino board for the first order, second order and higher order systems. In all the cases the response of Arduino is compared with the MATLAB response.**

Keyword -I-PD controller, non-linear PID controller, PID controller, PD-Kick, Microcontroller, Arduino.

I. Introduction

The dominancy of PID control still continues even during this modern time, because of its simplified structure and robustness. However the tuning of PID controller still faces many challenges not only due to the complexity in the design of the controlled systems but also due to changeability in external disturbances.

In PID controller, an impulse signal or a spike is generated in the controller output and in the output response for a sudden change in the set point. This spike or impulse signal is called the proportional or derivative kick which is shown in Fig.1(a).

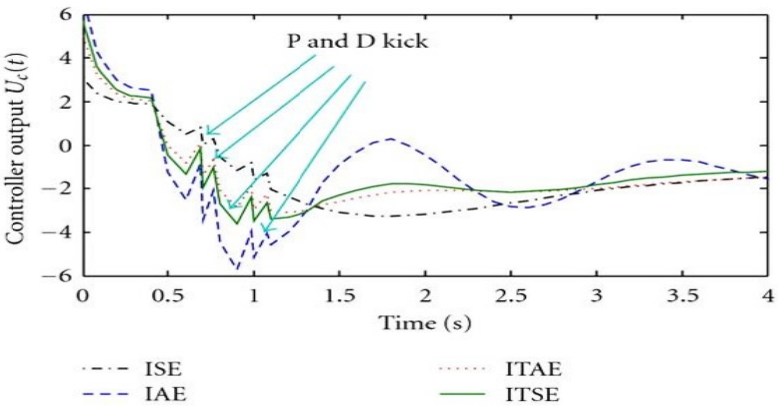

Fig. 1(a) proportional derivative kick plot

This kick leads to serious problems if it is given to final control elements such as control valve or motor.

I-PD controller is one among the variants of PID controller which is particularly designed to reduce the influence of sudden change in the set point over the controller output and the controlled output thereby.The PID controller performance can be improved by customizing its structure.

The controller output is given by

$$u(t)_{I-PD} = K_p e(t) + K_I \int e(t)dt + K_D \frac{de(t)}{dt}$$

For tuning the PID[17] and I-PD, researchers [1-8] proposed various tuning procedure through optimization techniques to control various stable and non-linear system by different methods to improve closed loop response. Still the tuning of controllers faces many challenges. In [14] various tunig rules for PID structure had been discussed. Among all the heuristic algorithms the Firefly algorithm takes less computation time. It surpasses all the 11 meta heuristic algorithms[15][18].

For the basic structure of I-PD is shown in Fig.1(b)

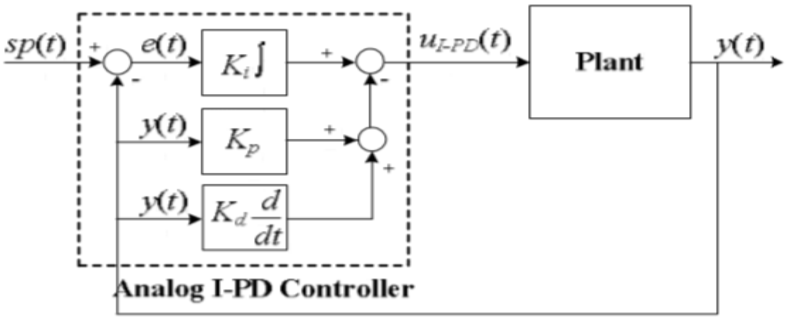

Fig.1(b) Basic I-PD controller

The modern technology not only focuses on the better control but also on the reduced physical size, better CPU performance and of course less price. These factors influences in almost entire human activities. Also with some processes the computers cannot be used due to the space constraint. In those processes embedded controllers can be worked as stand-alone controllers. This leads to implement I-PD controller function on microcontrollers. Researchers [9-13][19] proposed how to design and implementPID controller on a microcontroller.

In the paper in section 2 the PID & I-PD tuning by the modified firefly algorithm and the corresponding results were discussed. The same were implemented in Arduino IDE environment and the results were discussed in section 3.

II. PID AND I-PD CONTROLLER TUNING USING MODIFIED FIREFLY ALGORITHM

Inspired by the behaviour of fireflies Xin-She Yang formulated the firefly algorithm. The brighter fireflies attract all the other flies and the brightness is always get associated with the objective functions.

The features of the algorithm is:

- All the flies are unisexual.
- Attraction and the brightness are associated with each other and the less bright one will be attracted towards the brighter one.
- All the fireflies move randomly when all are with the same brightness.

The Pseudocode for firefly algorithm[16] is

While(s<maxgeneration)

for i=1:n(n-number of fireflies)

for j=1:n

if(Ii>Ij)

firefly$_j$ move towards firefly$_i$;

The brightness of firefly$_j$ is changing with respect to the distance r using the expression

$$exp^{-\gamma r}$$

Get the new velocities and obtain the new solution;

End j;

End i;

End while;

End;

The velocity updation formula for any two fireflies is

$$x_i^{t+1} = x_i^t + \beta exp^{-\gamma r_{ij}^2}\left(x_j^t - x_i^t\right) + \alpha_t \varepsilon_t$$

Where

The initial population of fireflies is Xi ;β order is one;γ is absorption coefficient; $\alpha_t$is the parameter controlling the step size and$\varepsilon_t$is the vector.

Usually α_t is constant ranging from 0.1 to 1.In the modified algorithm the α_t is modified by the following equation.

$$\alpha_{new} = \alpha_{min} + \left(\frac{min(I)}{mean(I)} + \frac{mean(I)}{2*\max(I)}\right)*\alpha_{max}$$

Where α_{min} is 0.1 and α_{max} is 1;Here I is the intensity of light of all fireflies taken together.

The algorithm is implemented in MATLAB and Simulink environment for various transfer functions. The objective functions here are PO and ITAE.The modified approach is run for several times and the optimized values are noted down.

2.(a).Example 1:

In example 1 the process considered here is a stable second order system. From table I(a)& table I(b) it is inferred that the I-PD controller outperforms the PID controller in all the aspects. The performance indices such as ISE, ITAE and PO and t_s are less for I-PD controller.

Table: I(A) Performance Results for PID of Example 1

Transfer Function	PID Parameters	PO	t_s	ISE	ITAE
$\frac{2e^{(-1s)}}{(10s+1)(5s+1)}$	K_p=2.6696 K_i=0.2458 K_d=2.0396	1.48	40	9.7990	99.428

Table: I(B) Performance Results for I-PD of Example 1

Transfer Function	I-PD Parameters	PO	t_s	ISE	ITAE
$\frac{2e^{(-1s)}}{(10s+1)(5s+1)}$	K_p=-0.9817 K_i=0.0711 K_d=0.1025	1.13	50	9.4235	83.169

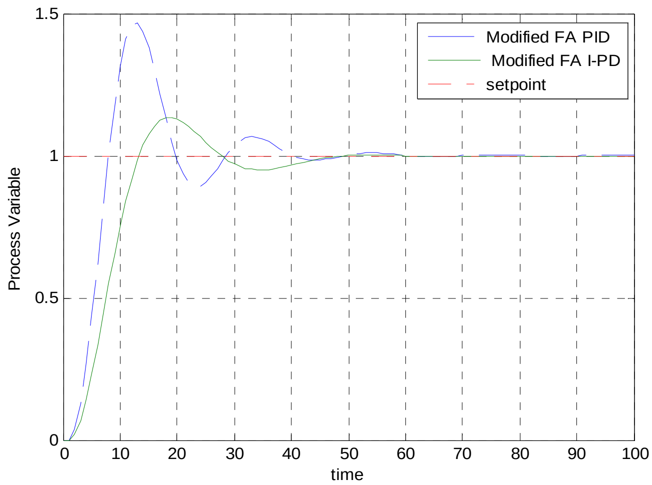

Fig.2(a) MATLAB Response for the Example 1

2(b).Example 2:

In example 2 the process considered here is a second order non minimum phase system. From table II(A)&tableII(B) it is inferred that the I-PD controller outperforms the PID controller in all the aspects. The performance indices such as ITAE and PO and t_s are less for I-PD controller.

Table:II(A) Performance Results for PID of Example 2

Transfer Function	PID Parameters	PO	t_s	ISE	ITAE
$\frac{(-0.2S+1)e^{(-0.1s)}}{(S+1)(S+1)}$	K_p=1.2624 K_i=0.5324 K_d=0.0999	0	10	3.444	372.12

Table:II(B) Performance Results for I-PD of Example 2

Transfer Function	PID Parameters	PO	t_s	ISE	ITAE
$\frac{(-0.2S+1)e^{(-0.1s)}}{(S+1)(S+1)}$	K_p=-0.8347 K_i=0.4480 K_d=-0.2276	0	8	3.7145	333.88

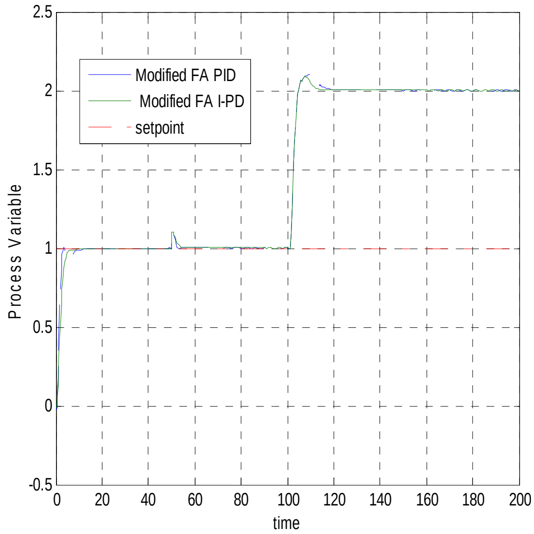

Fig.2(b) MATLAB Response for the Example 2

III. PID & I-PD CONTROLLER ON ARDUINO

The PID and I-PD controller is implemented on Arduino board using Arduino IDE.The controlled output is obtained on the board at the specified pin.It can be viewed through the serial port and also can be logged in a excel file.

From the file the obtained controlled output is exported to MATLAB so that it can be compared with the existing output in the MATLAB environment.

(a).ALGORITHM OF PID CONTROLLER:

1.Initialize the PID parameters

2.Set the initial conditions as zero;

3.Set the setpoint as 1;

4.Compute m(n),e(n) and c(n) using the following

Equations

$$m(n) = cosp + Kc\left\{e(n) + \left(\frac{T}{\tau_I}\right)err\_accu + (\tau_D/T)\big(e(n) - e(n-1)\big)\right\}$$

$e(n) = y(n) - setpoint$

Where m(n) is the manipulated variable and c(n) is the controlled variable

(b).ALGORITHM OF I-PD CONTROLLER:

1.Initialize the PID parameters

2.Set the initial conditions as zero;

3.Set the setpoint as 1;

4.Compute m(n),e(n) and c(n) using the followingEquations

$$m(n) = cosp + Kc\left\{-e(n) + \left(\frac{T}{\tau_I}\right)err_{accu} - (\tau_D/T)\big(e(n) - e(n-1)\big)\right\}$$

$$e(n) = y(n) - setpoint$$

Where m(n) is the manipulated variable and c(n) is the controlled variable.

3.(a).Arduino output:

The program is written in arduino IDE environment and it is downloaded through the serial port to the arduino ATMEGA 328 board.The output can be obtained at the COM port and also at the specified pin.The obtained output is then loaded in MATLAB workspace and thus the results can be shown in the form of graph.

The PO,t_s and the performance indices for the example 1 is shown in Table III(A),which are evaluated in Arduino environment.On observing the results it is infered that the PO of I-PD is considerably less than that of PID. From table III(B) it can also be said that the PO is less in I-PD than PID

Table:III(A) Performance Results for PID in Arduino for Example 1

Transfer Function	PID Parameters	PO	t_s	ISE	ITAE
$\frac{2e^{(-1s)}}{(10s+1)(5s+1)}$	K_p=2.6696 K_i=0.2458 K_d=2.0396	1.95	55	9.19	232.47

Table:III(B) Performance Results for I-PD in Arduino for Example 1

Transfer Function	I-PD Parameters	PO	t_s	ISE	ITAE
$\frac{2e^{(-1s)}}{(10s+1)(5s+1)}$	K_p=-0.9817 K_i=0.0711 K_d=0.1025	1.84	67	12.15	294.18

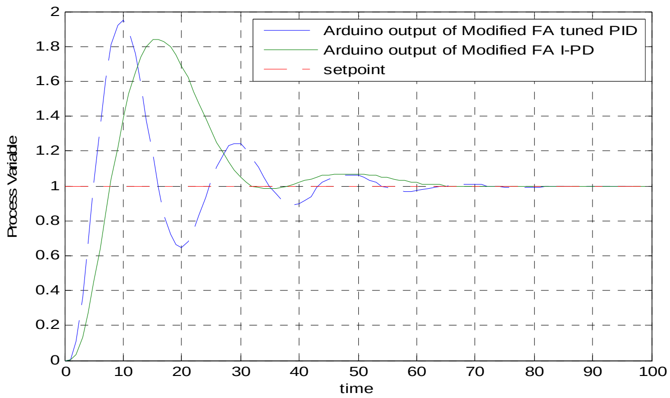

Fig.3(a). Arduino output for the example (1)

The table IV(A) and table IV(B) shows the results of PID & I-PD controller for example:2 respectively. From table IV(B) it is well known that not only PO but also all the performance indices are less comparatively.

Table:IV(A) Performance Results for PID in Arduino of Example 2

Transfer Function	PID Parameters	PO	t_s	ISE	ITAE
$\frac{(-0.2S+1)e^{(-0.1s)}}{(S+1)(S+1)}$	K_p=1.2624 K_i=0.5324 K_d=0.0999	1.73	10	2.24	11.12

Table:IV(B) Performance Results for I-PD in Arduino of Example 2

Transfer Function	PID Parameters	PO	t_s	ISE	ITAE
$\frac{(-0.2S+1)e^{(-0.1s)}}{(S+1)(S+1)}$	K_p=-0.8347 K_i=0.4480 K_d=-0.2276	1.56	10	1.96	7.45

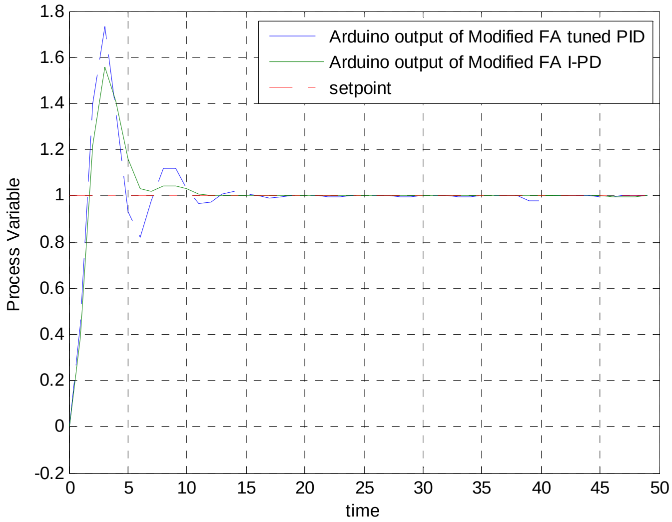

Fig.3(b).Arduino output for the example 2

IV. CONCLUSION:

Thus the parameters of PID and I-PD (a PID variant)are found out with the firefly algorithm and are implemented in Arduino platform. From the obtained results it can be said that the I-PD controller is better in both the set point and disturbance rejection. The implementation in embedded platform leads to a formation of standalone controller in the places where space and cost is a main constraint.In both the cases the I-PD outperforms the PID.

REFERENCES

[1] V.Rajinikanth and K.LathaI-PD Controller Tuning For Unstable System Using Bacterial Foraging Algorithm:A Study Based On Various Error Criterion, Hindawi Publishing Corporation Applied Computational Intelligence And Soft Computing Volume 2012, Article Id 329389, 10 Pages Doi:10.1155/2012/329389.
[2] P.A. BalakrishnanOptimization Of I-PD Controller For A FOLIPID Model Using Particle Swarm Intelligence, International Journal Of Computer Applications (0975 – 8887) Volume 43– No.9, April 2012.J.
[3] Morimasa Ogawa And Tohru Katayama A Robust Tuning Method For I-PD Controller Incorporating A Constraint On Manipulated Variable, Trans. Of The Society Of Instrument And Control Engineers Vol.E-1, No.1, 265/273 (2001).
[4] S J Suji Prasad Optimization Of I-PD Controller Parameters With Multi Objective Particle Swarm Optimization, Journal Of Theoretical And Applied Information Technology 20th August 2014. Vol. 66 No.2.
[5] Gao, Ruiyao And O'dwyer, Aidan And Coyle, Eugene: A Non-Linear PID Controller For CSTR Using Local Model Networks. Proceedings Of The IEEE 4th World Congress On Intelligent Control And Automation (Wcica 2002), Shanghai, China, 10-14 June.
[6] WicharnChatrattanawuth, NapatpongSuksariwattanagul, TawornBenjanarasuth And JongkolNgamwiwit Fuzzy I-PD Controller For Level Control ,Sice-Icase International Joint Conference 2006 Oct. 1821, 2006 In Bexco, Busan, Korea.
[7] A Nassirharand, N Hoq,H S TzouDesign Of Nonlinear PID Controllers Using System Step Response.
[8] K. K. Tan, Member, IEEE, Tong Heng Lee, Member, IEEE, And Huixing X. Zhou Micro-Positioning Of Linear-Piezoelectric Motors Based On A Learning Nonlinear PID Controller, IEEE/ASME Transactions On Mechatronics, Vol. 6, No. 4, December 2001.
[9] James W. Gray PID Routines For Mc68hc11k4 And Mc68hc11n4 Microcontrollers.
[10] Omar Jones, Design And Development Of An Embedded Dc Motor Controller Using A PID Algorithm, TekniskaHögskolanLinköpingsUniversitet.
[11] Poonam M BaikarDesign Of PID Controller Based Information Collecting Robot In Agricultural Field ,International Journal Of Advanced Research In Electrical, Electronics And Instrumentation Engineering, Vol. 3, Issue 8, August 2014.
[12] Sushant R. BurjeDesign And Development Of Microcontroller Based Electronic Speed Governor For Genset/Automotive Engine, Research Inventy: International Journal Of Engineering And Science Issn: 2278-4721, Vol. 1, Issue 5 (October 2012), Pp 26-33.
[13] PetrDostálek, Libor Pekař, VladimírVašek, Jan DolinayMicrocontroller Based Self-Tuning Digital PID Controller, Latest Trends On Systems (Volume I)
[14] O'Dwyer A. Handbook of PI and PID Controller Tuning Rules. 2nd edition, World Scientiic: Singapore; 2006.
[15] M. GnanaSundari, et al., Application of improved firefly algorithm for programmed PWM in multilevel inverter with adjustable DC sources, Appl. Soft Comput. J. (2015)
[16] https://en.wikipedia.org/wiki/Firefly_algorithm
[17] Meena S, Chitra K. A New approach of PID tuning for Nonlinear SISO system based on Particle Swarm Optimization Techniques.International Journal of Applied engineering Research 2014;9(23):-21701-11.
[18] Meena S, Chitra K. Modified approach of firefly algorithm for non-minimum phase systems. Indian Journal of Science & Technology. Vol 8(23).72264.September 2015.
[19] S.Meena ,K.Chitra, R.VijayAnand. Development of I-PD controller on Embedded platform. International Conference on Signal Processing , Communication, Power and Embedded Sysytems. (Scopes- 2016).

AUTHOR PROFILE

S Meena has completed her B.E in National Engineering College, Kovilpatti and her M.E from Sathyabama University with specialistion in Electronics & Control. She has 11 years of experience in teaching. Her areas of research are Process control and embedded systems.

Dr. K. Chitra is Currently working as a Professor, in the SENSE at VIT , Chennai. Dr. K. Chitra received her B.E. Degree in Electronics and communication Engineering from Bharathiar University, Coimbatore in 1990, M.E. Degree in Applied Electronics from Bharathiar University in 1992 and Ph.D. degree in Optical Communication from Anna University Chennai in 2008. Dr.K.Chitra has spent her 26 years of experience in teaching and guiding projects for undergraduate and postgraduate students. Dr. K. Chitra has added 15 international publications to her credit. She has few funded projects from Government of India. Dr. K. Chitra's areas of interests include optical communication, optical networks, wireless sensor and computer networks, Biomedical Engineering and microwave Engineering.

R Vijayanand has completed his B.E from St.Joseph's College of Engineering. His areas of interest include embedded sytems, C programming and process control.