Off-Line Tuning of Fractional Order PI$^{\lambda}$ Controller by Using Response Surface Method for Induction Motor Speed Control

Erdem Ilten, Metin Demirtas

Balikesir University, Balikesir, Turkey.
(e-mail: erdemilten, mdtas@balikesir.edu.tr)

Abstract: In this study, fractional order PI$^{\lambda}$ (FOPI$^{\lambda}$) speed control of induction motor is realized with dsPIC30F4011 microcontroller which is suitable for industrial applications. Voltage/frequency (V/f) control method is used for induction motor. Integer order PI (IOPI) controller and fractional order PI$^{\lambda}$ controller coefficients are optimized by using response surface method. The main advantage of FOPI controller is to provide more adjustable time and frequency responses of the control system allowing fulfillment of better as well as robust performance. FOPI controller is less sensitive than a classical PI controller when the parameter of the controlled system changes. FOPI has an extra variable to tune. This provides extra degrees of freedom to the dynamic properties of fractional order system. IOPI and FOPI$^{\lambda}$ controller results are compared. FOPI and IOPI settling times are obtained as 0.65 and 1.15 second, respectively. FOPI and IOPI overshoots are approximately obtained as the same value (Less than 1 %). Fractional order PI$^{\lambda}$ controller results are better than integer order PI controller.

Keywords: fractional order control, response surface methodology, induction motors, inverters, microcontrollers.

1. INTRODUCTION

In industry, variable-speed drive systems are used in many applications such as hybrid electric vehicles, heating, ventilation, air conditioning (HVAC) systems, driver controls and automotive controls (Arulmozhiyal et al., 2010; Waskar et al., 2012). In recent years, induction machines are preferred in variable-speed drive systems instead of direct current machines due to their low-cost, promising performance at bad environmental conditions, maintenance free, making less fault due to not containing brush and collector (Bowling, 2005; Toufouti et al., 2009). Although the induction machines have a lot of advantages, their control system is quite complex. In many studies, vector control method has been used for speed control of induction motor due to its dynamic response. However, the scalar control method has a simple structure due to easy applicable and low steady-state error (Suetake et al., 2011; Ustun and Demirtas, 2009). In the scalar control method, Voltage / frequency (V/f) ratio control is widely used in industrial applications. The torque value of induction motor is stable maintained at maximum torque value by changing frequency. Therefore, V/f ratio (or the flux) is kept constant approximately.

Computer or Digital Signal Processor (DSP) based hardware systems are widely employed in speed or position control of induction motor. However these systems are expensive for some applications. Microcontrollers are preferred to systems that are cheap and have a limited flexibility. Therefore, in this study dsPIC30F4011 microcontroller is chosen for speed control of induction motor. This controller is preferred for such as these applications because it is developed for three phase systems.

Integer order proportional integral (IOPI) control method is commonly employed in industrial closed-loop control system applications due to its simple algorithm. However in the recent years, fuzzy logic, sliding mode, fractional order proportional integral (FOPI$^{\lambda}$) etc. control methods are preferred in some industrial applications for some of their advantages (Efe, 2011).

Optimization is one of the most important problem in engineering applications. IOPI, FOPI$^{\lambda}$, sliding mode etc. controller parameters must be optimized for stability of the systems. Many methods are being used in optimization which include neural networks, genetic algorithm (Lazarevic, 2013; Farook and Raju, 2012; Das et al., 2012; Tabari and Kamyad, 2012; Padhee et al., 2011), Ziegler-Nichols method (Tajjudin et al., 2013; Poovarasan et al., 2012), particle swarm optimization (Atan et al., 2013; Dastranj et al., 2012; Rastogi et al., 2013; Bouarroudj et al., 2015; Rebai et al., 2015), simplex method, orthogonal test method and response surface method (Demirtas and Karaoglan, 2012; Arotaritei et al., 2014).

Fractional order calculus allows us to describe and model a real object more accurately than the classical integer methods. FOPI controller is symbolized as PI$^{\lambda}$. It permits us to adjust integral (λ) order in addition to the proportional and integral constants where the values of λ changes between 0 and 1. This also provides more flexibility and opportunity to better adjust the dynamical properties of the control system. The main advantage of Fractional-order controller is to provide more adjustable time and frequency responses of the control system. Therefore, the fractional-order controller will always provide better response than integer-order controller if it is properly tuned whatever may be the type of plant (integer or fractional). The fractional order is supposed to offer two advantages that are, $FOPI^{\lambda}$ is less sensitive than IOPI controller If the parameter of a controlled system changes. $FOPI^{\lambda}$ has an extra variable to tune.

There are many studies about IOPI and $FOPI^{\lambda}$ controllers, such as dsPIC applications, tuning the parameters (Kesarkar and Selvaganesan, 2015) of proportional integral derivative (PID) controllers, robust PID controller designs (Parastvand and Khosrowjerdi, 2014) . (Petras et al., 2003) proposed the fractional order controller realized with PIC microcontroller. They used the Analog-Digital Converter (ADC) module of the microcontroller to obtain the voltage value and control the voltage output by using $FOPI^{\lambda}$ method. They said that due to the microcontroller's limited memory, the fractional order controller performance has been decreased. (Xue et al., 2006) presented a fractional order PID control of a DC-motor with elastic shaft. They compared the integer order PID (IOPID) and the fractional order $PI^{\lambda}D^{\mu}$ ($FOPI^{\lambda}D^{\mu}$) controller performances in simulation. They emphasized the $FOPI^{\lambda}D^{\mu}$ controller will outperform than the conventional IOPID controller if it is properly designed and implemented. (Ustun and Demirtas, 2009) presented modeling and control of a V/f controlled induction motor using genetic-ANFIS algorithm. A PI controller is used to control the induction motor. (Zong et al., 2007) presented a $FOPI^{\lambda}$ control algorithm for the permanent magnet synchronous motor speed adjusting system. They said that the simulation results indicate that the $FOPI^{\lambda}$ controller can improve the disturbance rejection performance of the PMSM speed-adjusting system. (Singhal et al., 2012) presented a design of $FOPI^{\lambda}D^{\mu}$ controller for speed control of DC motor. They compared the IOPID and $FOPI^{\lambda}D^{\mu}$ controllers. It is noteworthy that the $FOPI^{\lambda}$ control method has shorter settling time, less overshoot and more robustness under external disturbances than the IOPI control method in previous studies (Petras, 2009; Özdemir and İskender, 2010; Duarte-Mermoud et al., 2010; Zhao et al., 2005; Wang et al., 2009; Maiti et al., 2008; Wang and Pi, 2012; Zhang and Pi, 2011; Tavazoei, 2012; Li and Hori, 2007; Erenturk, 2013; Vaithiyanathan and Bhaba, 2013).

$FOPI^{\lambda}$ control method can be realized by using a digital controller (Petras et al., 2003; Xue et al., 2006; Ustun et al., 2009; Zong et al., 2007; Singhal et al., 2012; Petras, 2009; Duma et al., 2011). The $FOPI^{\lambda}$ controller uses all received data from the starting point of the system. Therefore, large memory is needed for $FOPI^{\lambda}$ controller. In this study, dsPIC10F4011 microcontroller is used that has limited memory. For this reason, a $FOPI^{\lambda}$ control algorithm is written with using only last 50 data sets.

The remaining part of this paper is organized as follows: in Section 2, induction motor V/f control method is explained; in Section 3, IOPI and $FOPI^{\lambda}$ controller equations are given; in Section 4, IOPI and the $FOPI^{\lambda}$ controller are compared in speed control of induction motor and the results are presented; finally, conclusion is given in Section 5. The nomenclature is listed in Table 1.

Table 1. Nomenclature.

K_p	proportional coefficient
K_i	integral coefficient
λ	order of integral
M_o	maximum overshoot (%)
T_s	settling time (s)
PID	proportional integral derivative
IOPI	integer order proportional integral
IOPID	integer order proportional integral derivative
$FOPI^{\lambda}$	fractional order proportional integral
$FOPI^{\lambda}D^{\mu}$	fractional order proportional integral derivative

2. INDUCTION MOTOR V/F CONTROL

In industrial induction motor drive systems, increasing or reducing the load of motor changes the rotational speed, revolutions per minute (RPM). This situation has revealed the need of controlling the motor torque. Increasing the torque when the load has increased and reducing torque when the load has reduced are maintained stable the motor RPM. This torque control can be done with keeping constant the induction motor voltage amplitude and frequency. The speed control of the motor can be done by changing the stator frequency without changing the V/f ratio. Also the motor current is limited with keeping constant the V/f ratio. Induction motor torque equation is as follows:

$$M = \frac{P_{mec}}{\omega_r} \tag{1}$$

In equation (1), M is the torque (Nm), P_{mec} is the motor output power (Watt), and ω_r is the angular velocity of the rotor (rad/s). Keeping constant the V/f ratio allows to keep constant current drawn from the source at the same time. Thus, it prevents induction motors from excessive current drawn from the source at start-up under load and provides their working more efficiently.

3. THE INTEGER ORDER PI AND THE FRACTIONAL ORDER PI CONTROLLERS

3.1 Integer Order PI Controller

IOPI controller equation is as follows:

$$u(t) = K_p e(t) + K_i \int e(t)dt \tag{2}$$

In equation (2), t is the time variable, $e(t)$ is the error function, K_p and K_i are the proportional and the integral gains of the controller. Fig. 1 shows the integer order PI controller block diagram. The error function $e(t)$ is the difference between the reference value $r(t)$ and the system output $y(t)$.

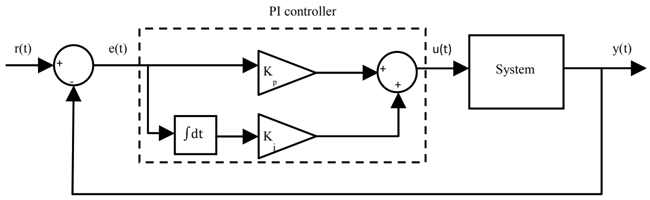

Fig. 1. PI controller block diagram.

3.2 Fractional Order PI$^{\lambda}$ Controller

FOPI$^{\lambda}$ controller equation is defined as follows:

$$u(t) = K_p e(t) + K_i I^{\lambda} e(t) \tag{3}$$

Here, t is the time variable, K_p and K_i are the proportional and the integral gains of the controller, respectively, I is the integral function and λ is the order of integral. The error function *e(t)* is the difference between the reference value *r(t)* and the system output *y(t)*. FOPI$^{\lambda}$ controller provides more flexibility than the IOPI controller as it allows to tune λ, in addition to tuning of K_p and K_i (Podlubny, 1994; Özdemir and İskender, 2010).

There are different types of fractional order derivative (and integral) definitions which can be chosen according to the problem structures (Petras, 2011). In this study, the Grünwald-Letnikov definition is used for the controller. It can be defined as

$${}_0D_t^{\alpha} x(t) = \lim_{h \to 0} \frac{1}{h^{\alpha}} \sum_{k=0}^{[t/h]} (-1)^k \binom{\alpha}{k} x(t-kh) \tag{4}$$

$$\binom{\alpha}{k} = \frac{\Gamma(\alpha+1)}{\Gamma(k+1)\Gamma(\alpha-k+1)} \tag{5}$$

where x is a time dependent function, α is the order of derivative $(n-1 \le \alpha < n,\ n \in N^{+})$, $\Gamma(.)$ is Euler's gamma function, h is the time increment, and [t/h] represents the integer parts of t/h. If the order of derivative α is changed with $-\lambda$ this definition corresponds to the fractional order integral I^{λ} in the sense of Grünwald-Letnikov. Advantage of Grünwald-Letnikov definition comes from its ability to discretize system (Petras, 2011). The Grünwald-Letnikov definition is preferred in this study, because the microcontroller's structure is a discrete-time operation system.

If the limit operation is removed from equation (4), the Grünwald-Letnikov fractional derivative becomes a numerical tool. This approximation is applied to calculate fractional integral I^{λ} by dividing the time interval $[0,T]$ to N equal parts therein the each parts has the size of $h=1/N$. The nodes are labeled as 0, 1, 2, …, N and the fractional order integral at node M is obtained as follows:

$${}_0I_t^{\lambda} x(t) = {}_0D_t^{-\lambda} x(hM) = \frac{1}{h^{-\lambda}} \sum_{j=0}^{M} w_j^{(-\lambda)} x(hM - jh) \tag{6}$$

$$w_0^{(\alpha)} = 1, \quad w_j^{(\alpha)} = \left(1 - \frac{\alpha+1}{j}\right) w_{j-1}^{(\alpha)}, \quad j = 1,2,\ldots N \tag{7}$$

Then the fractional order PI$^{\lambda}$ controller is discretized as follows:

$$u(Mh) = K_p e(Mh) + k_i \frac{1}{h^{-\lambda}} \sum_{j=0}^{M} w_j^{(-\lambda)} e(Mh - jh) \tag{8}$$

4. APPLICATION OF PI CONTROLLERS TO THE INDUCTION MOTOR

The experimental setup is designed for realization of speed control of induction motor. The system consists of single-phase rectifier, three-phase inverter, dsPIC30F4011 microcontroller, LabVIEW software and DAQ card, 1.1 kW induction motor, 1024 PPR encoder and 20 Nm electromagnetic Foucault Brake. Fig. 2 shows the block diagram of the experimental setup, Fig. 3 shows the control circuit of the experimental setup, and Fig. 4 shows the power circuit of the experimental setup.

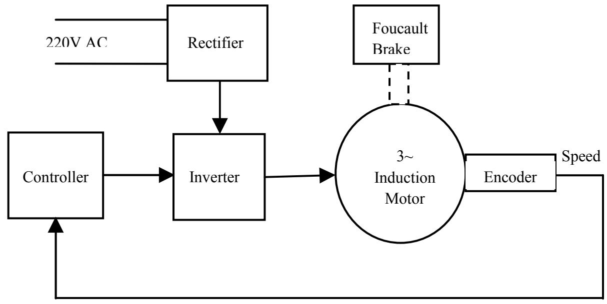

Fig. 2. Block diagram of experimental setup.

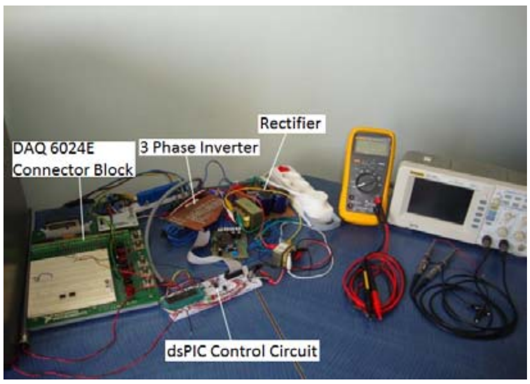

Fig. 3. Control circuit of experimental setup

Motor parameters are listed in Table 2.

Table 2. Motor parameters.

Power	1.1 kW
Rated voltage	380 V
Rated Current	2.7 A
Frequency	50 Hz
cosineφ	0.8
Stator resistance (Rs)	6.9 Ω
Stator inductance (Lls)	0.52 H
Rotor resistance (Rr)	6.5 Ω
Rotor inductance (Llr)	0.52 H
Mutual inductance (Lm)	0.5 H
Rotor inertia (J)	0.0088 kg.m^2
Fraction factor (B)	0.072 N.m.s
Pole pairs	2

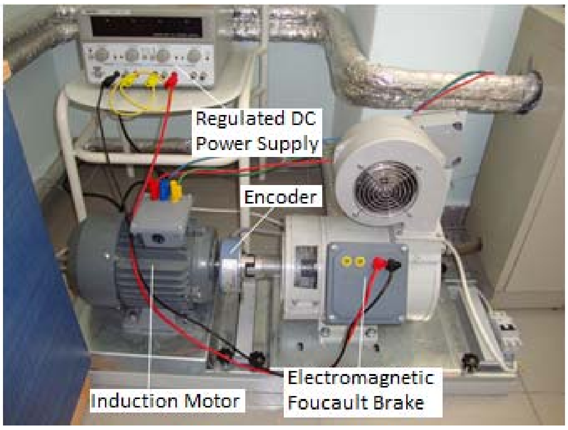

Fig. 4. Power circuit of experimental setup.

Single-phase rectifier has been used to obtain the DC voltage from AC line voltage to supply the inverter circuit. The inverter circuit has been designed to control the speed and frequency of the induction motor. dsPIC30F4011 microcontroller has been used as a control unit and to compare the reference value and the system output for obtaining the error information. PI algorithm runs by using this error information. LabVIEW software and NI 6024E DAQ card have been used to obtain the system outputs to storage in EXCEL file and displaying them graphically. Interpretation of experimental results can be made easier in this way. A 1.1 kW induction motor has been preferred in this study because it is cheap and easy applicable in experimental study. 1024 PPR encoder has been used to minimize the feedback error. The IOPI controller and the FOPI$^{\lambda}$ controller have been applied in the same experimental setup to see the difference between the two algorithms. First of all, IOPI controller algorithm has been designed and applied. The equation of discrete-timed IOPI controller is as follows:

$$u(k) = K_p e(k) + Ki\frac{e(k)+e(k-1)}{2}h \tag{9}$$

In equation (9), *e(k)* is the error function, *h* is the time interval, K_p is the gain of the proportional controller, K_i is the gain of the integral controller. K_p and K_i gains are written in dsPIC30F4011 and the controller signals are generated to driving the inverter circuit. The controller changes the voltage amplitude and frequency of the induction motor by adjusting the pulse width and frequency of the sinusoidal PWM output. The results of these algorithm applications have been stored in EXCEL file by using the LabVIEW environment. Transferring data to EXCEL file effects the LabVIEW system slightly negative. However, it is useful to analyzing the data and transforming into the graph.

Response surface method has been used for optimizing the coefficients of the controllers. In response surface method, mathematical model of the system is not need to optimizing the coefficients. The method establishes a high accuracy mathematical model with using relationships between the factors and the responses of the system. This is the advantage of the response surface method in comparison with other existing methods. There are five forms for response surface method as shown in Table 3.

Table 3. Available response surface designs (with number of runs).

Design		Factors								
		2	3	4	5	6	7	8	9	10
Central Composite full	unblocked	13	20	31	52	90	152			
	blocked	14	20	30	54	90	160			
Central Composite half	unblocked				32	53	88	154		
	blocked				33	54	90	160		
Central composite quarter	unblocked							90	156	
	blocked							90	160	
Central Composite eighth	unblocked									158
	blocked									160
Box-Behnken	unblocked		15	27	46	54	62		130	170
	blocked			27	46	54	62		130	170

Factors represent the number of input parameters. IOPI has two factors (K_p and K_i), FOPI$^{\lambda}$ has three factors (K_p, K_i and λ). The other forms need more input parameters. Therefore, the only suitable form is central composite full design for optimization of both controllers (IOPI and FOPI$^{\lambda}$). General second-order polynomial response surface mathematical model (full quadratic model) is given in Eq. (10) (Demirtas and Karaoglan, 2012).

$$Y_u = \beta_0 + \sum_{i=1}^{n}\beta_i X_{iu} + \sum_{i=1}^{n}\beta_{ii} X_{iu}^2 + \sum_{i<j}^{n}\beta_{ij} X_{iu} X_{ju} + e_u \tag{10}$$

In Eq. (10) Y_u is the corresponding response, X_{iu} are coded values of i^{th} input parameters, β_0, β_i, β_{ij} and β_{ij} are the regression coefficients, *i* and *j* are the linear and quadratic coefficients and e_u is the residual experimental error of u^{th} observation (random error) (Yalcinkaya, O., Bayhan, 2009; Rashid et al., 2011).

Table 4. Limits of the IOPI control coefficients.

Coefficient	Lower limit	Upper limit
K_p	0.01	0.1
K_i	0.0001	0.01

Table 5. Design of experiments matrix for IOPI control.

Experiment number	Kp	Ki	Mo	Ts
1	0.010	0.00010	0	10.9
2	0.100	0.00010	33.12	1
3	0.010	0.01000	0	10.8
4	0.100	0.01000	36.06	1.1
5	0.010	0.00505	0	10.4
6	0.100	0.00505	31.77	1
7	0.055	0.00010	9.79	1.4
8	0.055	0.01000	10.48	1.4
9	0.055	0.00505	8.93	1.4
10	0.055	0.00505	9.99	1.4
11	0.055	0.00505	10.28	1.4
12	0.055	0.00505	10.61	1.4
13	0.055	0.00505	9.46	1.4

In response surface method, the factors must be defined between upper and lower limits. The upper and lower limit values are defined based on the experience of the designer. In this study, K_p and K_i coefficients are the factors of the response surface method. If K_p and K_i exceed the upper limit, the system runs unstable. If K_p and K_i fall below the lower limit, the settling time of the system is too long.

For IOPI control, the factor limits of K_p and K_i are listed in Table 4. Table 5 shows the experimental design. The design requires standard eight experiments for cube and axial points and five experiments for center points (0.0) with totally 13 experiments.

Response surface plots of M_o and T_s of IOPI controller are given in Fig. 5 and Fig. 6, respectively.

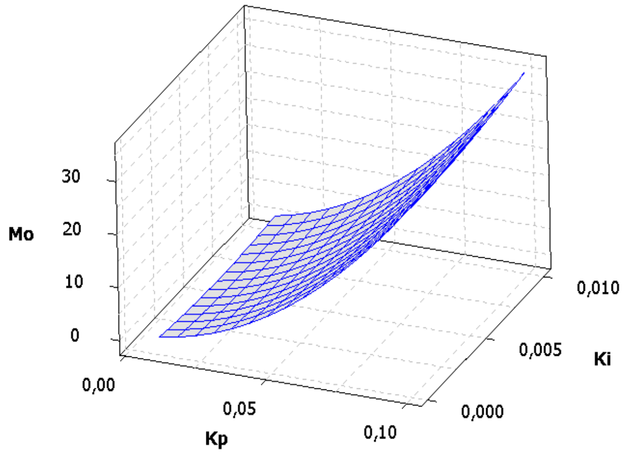

Fig. 5. Response surface plot of M_O for IOPI controller.

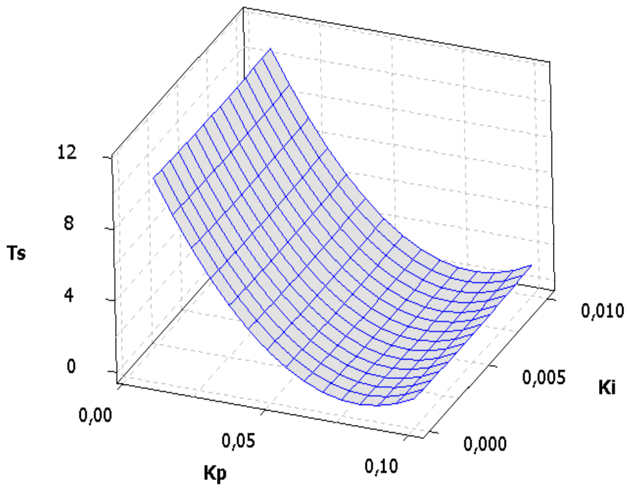

Fig. 6. Response surface plot of T_S for IOPI controller.

For FOPI$^\lambda$ control, the factor limits of K_p, K_i and λ are listed in Table 6. Table 7 shows the experimental design. The design requires standard fourteen experiments for cube and axial points and three experiments for center points (0.0) with totally 17 experiments.

Table 6. Limits of the FOPI$^\lambda$ control coefficients.

Coefficient	Lower limit	Upper limit
K_p	0.01	0.1
K_i	0.0001	0.01
λ	0.1	0.9

Table 7. Design of experiments matrix for FOPI$^\lambda$ control.

Experiment number	Kp	Ki	λ	Mo	Ts
1	0.010	0.00010	0.1	0.00	8.5
2	0.100	0.00010	0.1	37.28	1.1
3	0.010	0.01000	0.1	0.00	8.1
4	0.100	0.01000	0.1	34.28	1.1
5	0.010	0.00010	0.9	0.00	8.6
6	0.100	0.00010	0.9	32.85	1.0
7	0.010	0.01000	0.9	0.00	6.2
8	0.100	0.01000	0.9	53.42	1.2
9	0.010	0.00505	0.5	0.00	8.0
10	0.100	0.00505	0.5	35.85	1.1
11	0.055	0.00010	0.5	9.14	1.4
12	0.055	0.01000	0.5	9.28	1.3
13	0.055	0.00505	0.1	8.14	1.5
14	0.055	0.00505	0.9	13.14	1.4
15	0.055	0.00505	0.5	8.42	1.4
16	0.055	0.00505	0.5	9.00	1.4
17	0.055	0.00505	0.5	8.57	1.4

Response surface plots of FOPI$^\lambda$ controller's M_o and T_s are given in Fig. 7 to Fig. 12.

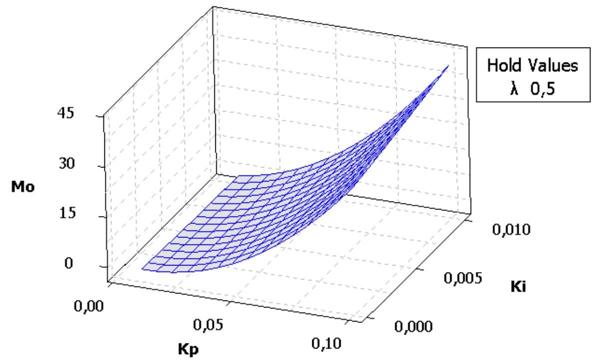

Fig. 7. Response surface plot of M_O vs Ki; Kp for FOPI$^\lambda$ controller.

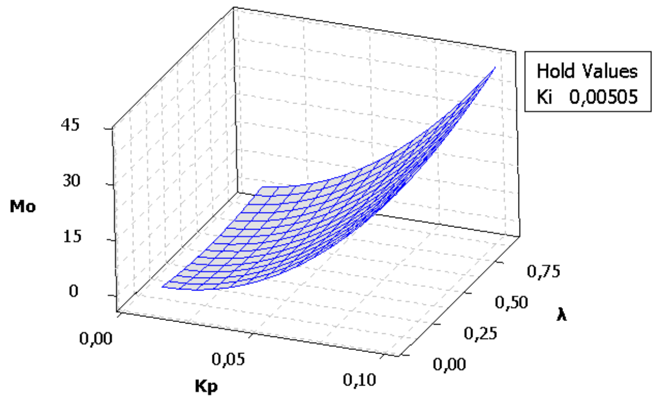

Fig. 8. Response surface plot of M_O vs λ; Kp for FOPI$^\lambda$ controller.

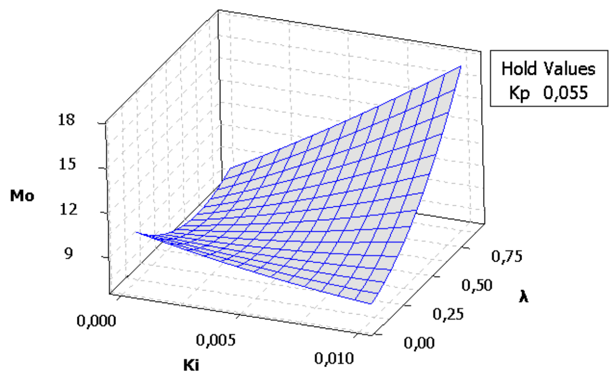

Fig. 9. Response surface plot of M_O vs λ; Ki for FOPI^λ controller.

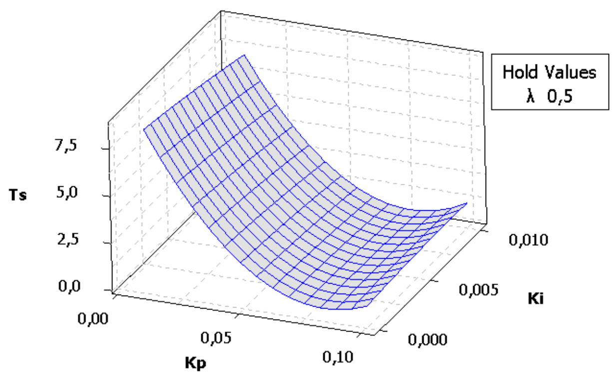

Fig. 10. Response surface plot of T_S vs Ki; Kp for FOPI^λ controller.

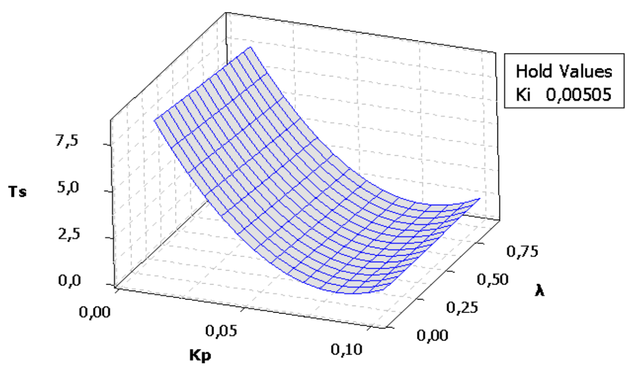

Fig. 11. Response surface plot of T_S vs λ; Kp for FOPI^λ controller.

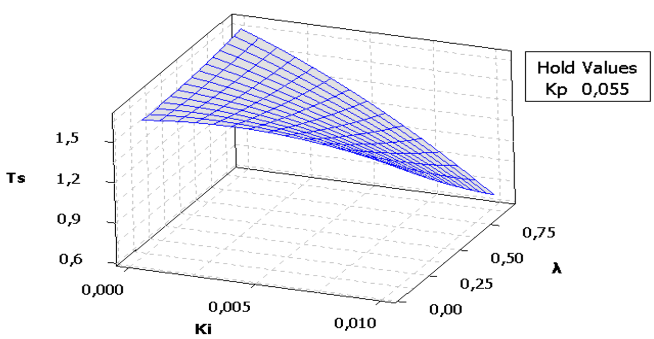

Fig. 12. Response surface plot of T_S vs λ; Ki for FOPI^λ controller.

Response surface method has been used for tuning the IOPI and the FOPI^λ controllers' gains. Optimum IOPI coefficients K_p is 0.0364 and K_i is 0.0044. Optimum FOPI^λ coefficients K_p is 0.0452, K_i is 0.010 and λ is 0.1566. Then the controller performances have been compared. Fig. 13 shows these results.

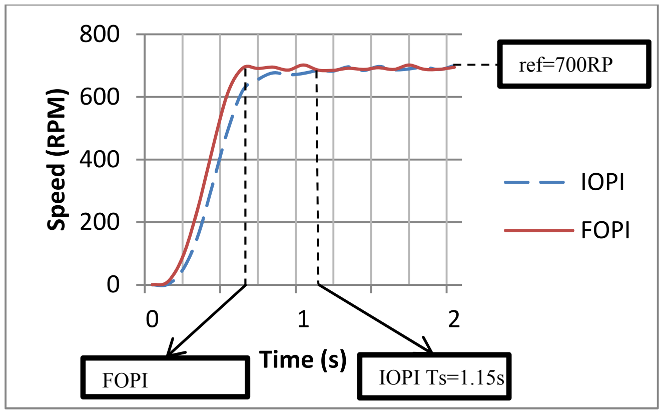

Fig. 13. Comparison of the IOPI and FOPI^λ controllers.

If the results are examined in Fig. 13, it is seen that the FOPI^λ controller has a shorter settling time than the IOPI controller (FOPI^λ T_s=0.65, IOPI T_s=1.15).

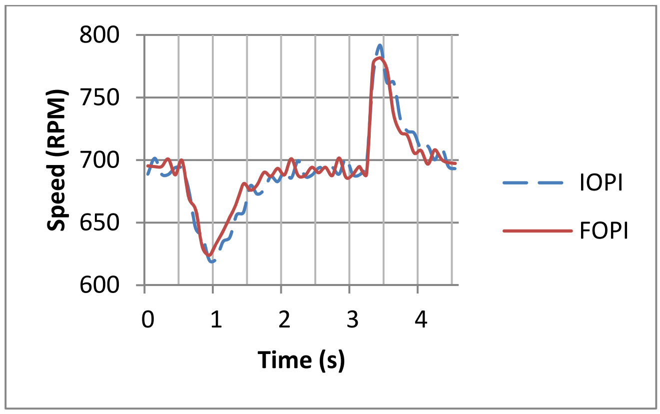

Fig. 14. The performances of IOPI and FOPI^λ controllers under external disturbance.

Fig. 14 shows the performances of IOPI and FOPI^λ controllers under external disturbance.

FOPI^λ controller's M_o and T_s is better than the IOPI controller's under external disturbance effect as shown in Fig. 14. There is small difference between the responses with IOPI and FOPI^λ controller because the FOPI^λ controller performance has been decreased due to low cost industrial microcontroller's limited memory (Petras et al., 2003). In this study, the FOPI^λ control algorithm runs by using last 50 data sets. The FOPI^λ controller performance approaches the IOPI controller performance when the numbers of using data sets are decreased. The FOPI^λ controller performance can be seen clearly in previous simulation studies (Petras, 2009; Xue et al., 2006; İ Özdemir and İskender, 2010).

5. CONCLUSION

In this study, a dsPIC based induction motor speed control system has been realized by using the IOPI and the FOPI^λ

controllers. Response surface method has been used for tuning the IOPI controller and the FOPI$^{\lambda}$ controller coefficients. No load starting experiment has been done and external disturbance effects have been examined for the IOPI controller and the FOPI$^{\lambda}$ controller. Experimental results have been presented and compared with each controller method. The FOPI$^{\lambda}$ controller has a shorter settling time and smaller overshoot than the IOPI controller. Moreover the fractional order controller has less overshoot and shorter settling time under external disturbances than the integer order controller. Even if the FOPI$^{\lambda}$ controller performance has been decreased by the microcontroller's limited memory, the FOPI$^{\lambda}$ controller has still better performance than the IOPI controller. In future works, the performance of the FOPI$^{\lambda}$ controller can be seen clearly with the development of high memory capacity industrial controller units.

ACKNOWLEDGMENT

This work was supported by Balikesir University under grant BAP 2013/63.

REFERENCES

Arotaritei, D., Constantin, G., Constantin, C., Cretu, A.L. (2014). Modelling the Cutting Process using Response Surface Methodology and Artificial Intelligence Approach: a Comparative Study. *Journal of Control Engineering and Applied Informatics*, Vol.16, No.2, pp. 14-27.

Arulmozhiyal, R., Baskaran, K., Deverajan, N., Kanagaraj, J. (2010). Real Time MATLAB interface for Speed Control of Induction Motor Drive Using dsPIC 30F4011. *International Journal of Computer Applications*. vol. 1 no. 5, pp. 85-90.

Atan, O., Turk, M., Tuntas, R. (2013). Fractional Order Controller Design for Fractional Order Chaotic Synchronization. *International Journal of Natural and Engineering Sciences*, vol. 7(3).

Bouarroudj, N., Boukhetala, D., Boudjema, F. (2015). A Hybrid Fuzzy Fractional Order PID Sliding-Mode Controller design using PSO algorithm for interconnected Nonlinear Systems. *Journal of Control Engineering and Applied Informatics*, Vol. 17, No. 1, pp. 41-51.

Bowling, S. (2005). An Introduction of to AC Induction Motor Control Using the dsPIC30F MCU. Microchip.

Das, S., Pan, I., Das, S., Gupta, A. (2012). Improved Model Reduction and Tuning of Fractional Order PI$^{\lambda}$D$^{\mu}$ Controllers for Analytical Rule Extraction with Genetic Programming. *ISA transactions*, vol. 51, no 2, pp. 237-261.

Dastranj, M.R. Rouhani, M., Hajipoor, A. (2012). Design of Fractional Order PID Controller Using PSO Algorithm. *International Journal of Computer Theory and Engineering*, vol. 4, no 3.

Demirtas, M., Karaoglan, A.D. (2012, April). Optimization of PI parameters for DSP-based permanent magnet brushless motor drive using response surface methodology. *Energy Conversion and Management*. vol. 56, pp. 104-111.

Duarte-Mermoud, M.A., Mira, F.J., Pelissier, I.S., Travieso-Torres, J.C. (2010). Evaluation of a Fractional Order PI Controller Applied to Induction Motor Speed Control. in *2010 8th IEEE International Conference on Control and Automation Xiamen*, China, June 9-11.

Duma, R., Trusca, M., Dobra, P. (2011). Tuning and Implementation of PID Controllers using Rapid Control Prototyping. *Journal of Control Engineering and Applied Informatics*, Vol. 13, No. 4, pp. 64-73.

Efe, M.O. (2011). Fractional Order Systems in Industrial Automation - A Survey. *IEEE Transactions on Industrial Informatics*. vol. 7(4), pp. 582 – 591.

Erenturk, K. (2013, Sept.). Fractional Order PI$^{\lambda}$D$^{\mu}$ and Active Disturbance Rejection Control of Nonlinear Two-Mass Drive System. *IEEE Transactions on Industrial Electronics*. vol. 60(9), pp. 3806-3813.

Farook, S., Raju, P.S. (2012). Robust Fractional Order PID Controller to Optimise Bilateral Contracts in a Deregulated Power System. *International Electrical Engineering Journal*, vol. 3, no 3, pp. 817-826.

Kesarkar, A.A., Selvaganesan, N. (2015). Novel Tuning Expressions for Fractional Order ($[PD]^{\beta}$ and $[PI]^{\alpha}$) Controllers Using a Generalized Plant Structure. *Journal of Control Engineering and Applied Informatics*, vol. 17, No. 1, pp. 70-80.

Lazarevic, M. (2013). Further Results on Fractional Order Control of a Mechatronic Systems. *Scientific Technical Review*, vol. 63, no 3, pp. 22-32.

Li, W. Hori, Y. (2007, Feb.). Vibration Suppression Using Single-Neuron Based PI Fuzzy Controller and Fractional-Order Disturbance Observer. *IEEE Transactions on Industrial Electronics*. vol. 54(1), pp. 117-126.

Maiti, D., Biswas, S., Konar, A. (2008). Design of a Fractional Order PID Controller Using Particle Swarm Optimization Technique, in *2nd National Conference on Recent Trends in Information Systems (ReTIS-08)*.

Özdemir, N., İskender, B.B. (2010, April). Fractional Order Control of Fractional Diffusion Systems Subject to Input Hysteresis. *ASME Journal of Computational and Nonlinear Dynamics*. vol. 5.

Padhee, S., Gautam, A., Singh, Y., Kaur, G. (2011). A Novel Evolutionary Tuning Method for Fractional Order PID Controller. *International Journal of Soft Computing and Engineering*, vol. 1(3).

Parastvand, H. and Khosrowjerdi, M.J. (2014). A New Data-Driven Approach to Robust PID Controller Synthesis. *Journal of Control Engineering and Applied Informatics*, vol. 16, No. 3, pp. 84-93.

Petráš, I., Grega, Š., Dorčák, Ľ. (2003). Digital Fractional Order Controllers Realized by PIC Microprocessor - Experimental Results. *Proc. of the ICCC'2003 conference*, High Tatras, Slovak Republic.

Petráš, I. (2009). Fractional – Order Feedback Control of A DC Motor. *Journal of Electrical Engineering*. vol. 60(3).

Petráš, I. (2011). Fractional Order Nonlinear Systems: Modeling, Analysis and Simulation. *Nonlinear Physical Science*.

Poovarasan, J., Ramireddy, B., Kayalvizhi, R. (2012). Design and Implementaiton of Fractional Order Controllers for

DC Motor Position Servo System. *American Jr. of Mathematics and Sciences*, vol. 1, no 1.

Podlubny, I. (1994). Fractional-Order Systems and Fractional Order Controllers. *Inst. Exp. Phys., Slovak Acad. Sci*.

Rashid, U., Anwar, F., Ashraf, M., Saleem, M., Yusup, S. (2011) Application of response surface methodology for optimizing transesterification of moringa oleifera oil: Biodiesel production. *Energy Conversion and Management*, 52: 3034-42.

Rastogi, A., Tiwari, P. (2013). Optimal Tuning of Fractional Order PID Controller for DC Motor Speed Control Using Particle Swarm Optimization. *International Journal of Soft Computing and Engineering*, vol. 3(2).

Rebai, A., Guesmi, K., Hemici, B. (2015). Design of an optimized fractional order fuzzy PID controller for a piezoelectric actuator. *Journal of Control Engineering and Applied Informatics*, Vol. 17, No. 3, pp. 41-49.

Singhal, R., Padhee, S., Kaur, G. (2012). Design of Fractional Order PID Controller for Speed Control of DC Motor. *International Journal of Scientific and Research Publications*, 2 (6).

Suetake, M., da Silva, I.N. Goedhel, A. (2011, March). Embedded DSP-Based Compact Fuzzy System and Its Application for Induction-Motor V/f Speed Control. *IEEE Transactions on Industrial Electronics*. vol. 58, no. 3.

Tabari, M.Y., Kamyad, A.V. (2012). Design Optimal Fractional PID Controller for DC Motor with Genetic Algorithm. *International Journal of Scientific & Engineering Research*, vol. 3(12).

Tajjudin, M., Rahiman, M.H.F., Arshad, N.M., Adnan, R. (2013). Robust Fractional-Order PI Controller with Ziegler-Nichols Rules. *World Academy of Science, Engineering and Technology*, vol. 7.

Tavazoei, M.S. (2012). From Traditional to Fractional PI Control: A Key for Generalization. *IEEE Industrial Electronics Magazine*. vol. 6(3), pp. 41 – 51.

Toufouti, R., Meziane, S., Benalla, H. (2009). New Direct Torque Neuro-Fuzzy Control Based SVM forDual Two Level Inverter-Fed Induction Motor. *Journal of Control Engineering and Applied Informatics*, Vol.11, No.2, pp. 3-13.

Ustun, S.V., Demirtas M. (2009). Modeling and control of V/f controlled induction motor using genetic-ANFIS algorithm. *Energy Conversion and Management*, 50:786–91.

Vaithiyanathan, K., Bhaba, P.K. (2013) “Real Time Implementation of Fractional Order Controller (PI^{λ}) in an Air Flow Temperature System”, *International Journal of Recent Technology and Engineering*, vol. 1(6).

Wang, C., Luo, Y., Chen, Y. (2009, June). Fractional order proportional integral (FOPI) and [proportional integral] (FO[PI]) controller designs for first order plus time delay (FOPTD) systems, in *Control and Decision Conference CCDC '09*, China, 17-19 June.

Wang, R., Pi, Y. (2012). Fractional-order PI Speed Control for Permanent Magnet Synchronous Motor, in *International Conference on Mechatronics and Automation (ICMA)*.

Waskar, S.A., Bombale, U.L., Sonawane, T.B. (2012, June). dsPIC Based SPWM Controlled Three Phase Inverter Fed Induction Motor Drive. *International Journal of Computer Applications*. vol. 47, no. 16.

Xue, D., Zhao, C. and Chen, Y. (2006). Fractional Order PID Control of A DC-Motor with Elastic Shaft: A Case Study. *Proceedings of the 2006 American Control Conference*, Minnesota, USA.

Yalcinkaya, O., Bayhan, G.M. (2009) Modelling and optimization of average travel time for a metro line by simulation and response surface methodology. *Europen Journal of Operational Research*, 196(1): 255-33.

Zambada, J. (2005). Module, Driving an ACIM with the dsPIC® DSC MCPWM. Microchip.

Zhang, B.T., Pi, Y. (2011). Robust fractional order proportion-plus-differential controller based on fuzzy inference for permanent magnet synchronous motor. *IET Control Theory and Applications*. vol. 6(6).

Zhao, C., Xue, D., Chen, Y. (2005, July). A fractional order PID tuning algorithm for a class of fractional order plants. in *Proceedings of the IEEE International Conference on Mechatronics & Automation,* Niagara Falls, Canada.

Zong, K., Li, S., Lin, X. (2007). The Application of Fractional-Order PI Control Algorithm to The PMSM Speed-Adjusting System. *Springer-Verlag*, 660-669.