A Hybrid Differential Artificial Bee Algorithm Based Tuning of Fractional Order Controller for PMSM Drive

Anguluri Rajasekhar, Millie Pant, and Ajith Abraham

Abstract— **This article proposes a novel Hybrid Differential Artificial Bee Colony Algorithm (HDABCA), which combines Differential Evolution with Artificial Bee Colony Algorithm (ABCA), for designing the fractional order proportional-integral controller (FO-PI) in a surface-mounted Permanent Magnet Synchronous Motor (PMSM) drive. FOPI controllers' parameters involve proportionality constant, integral constant and integral order, and hence its design is more intricate than that of usual integral-order proportional-integral controller. Here the synthesis of controller is done according to the objective function considered. In order to digitally realize the fractional order PI controller, an Oustaloup approximation method is used. Simulations and comparisons of proposed HD-ABCA with conventional methods demonstrate the superiority of the proposed approach, especially for actuating fractional order plants.**

I. INTRODUCTION

According to recent studies, with the advancement of control theories and power electronics, microelectronics in connection with new motor design and magnetic materials since 1980's electrical (A.C) drives are making tremendous impact in the area of various high-performance variable speed control systems [1], [2]. Among A.C drives newly developed PMSM with high energy permanent magnetic materials like "Neodymium Iron Boron"(Nd-Fe-B) or "Samarium-cobalt alloy"(Sm2Co17) provide fast dynamics and computability with the applications controlled properly.

Electrically excited field windings are replaced by Permanent Magnets because of their advantages like elimination of brushes, slip-rings and rotor copper losses that yields high efficiency and also these magnets produce constant flux. The design criteria of synchronous servo motors, to be used in industrial applications differ from that of conventional synchronous motors because of 1) high air-gap flux density, 2) high power/weight ratio, 3) low noise and low inertia, 4) small torque ripple 5) large torque/inertia and compact design in wide range of speed [2]. These requirements can be met well by PMSM with sinusoidal flux distribution due to lower torque ripples.

Anguluri Rajasekhar is with Department of Electrical and Electronics Engineering, National Institute of Technology-Warangal, INDIA `rajasekhar.anguluri@ieee.org`

Millie Pant is with Department of Paper Technology, Indian Institute of Technology Roorkee, Roorkee 247667, India `millifpt@iitr.ernet.in`

Ajith Abraham is with Machine Intelligent Research Labs (MIR Labs), Scientific Network for Innovation and Research Excellence, USA and also with Faculty of Electrical Engineering and Computer Science, VSB - Technical University of Ostrava, Czech Republic `ajith.abraham@ieee.org`

Now-a-days vector control technique has made it possible to apply the PMSM's in high-performance industrial applications where only dc motor drives were previously available to achieve fast four-quadrant operation [3]. In order to make the most of a motor performance, a very effective control system is needed. Although many possible solutions are available, eg., non-linear, adaptive control [4], the market of electrical drives doesn't justify the expense needed to implement such sophisticated solution in industrial drives and Proportional-Integral (PI) based control system scheme still remains the more widely adopted solution. Such a propensity is supported by a fact that, although simple, a PI control achieves high performance when optimally designed [5]. PI controllers have been widely used for decades in industries for process control applications. The reason for their wide popularity lies in the simplicity of design and performances including low percentage overshoot and low maintenance cost.

An elegant way of enhancing the performance of PI controller is to use fractional-order controllers. Dynamic systems based on fractional order calculus have been a subject of extensive research in recent years, since the proposition of concept of the fractional-order controllers and the demonstration of their effectiveness in actuating desired fractional order system responses by Podlubny [6]. Fractional Order Proportional Integral (FO-PI) controller is a convenient fractional order structure that has been employed for control purposes. In an FO-PI controller (in general PI^{λ}) I-operations are usually of fractional order; therefore besides setting K_P, K_I we have another parameter i.e., order of fractional integration λ. If $\lambda = 1$ it is integral PI controller. Finding appropriate set of values for these three parameters to achieve optimum performance of PMSM drive in three dimensional hyper-space calls for real parameter optimization. Our tuning method focuses on minimizing the ITAE criterion.

In past few decades, many general-purpose optimization algorithms have been proposed for obtaining optimal solutions to intricate and complex optimization problems sprouting in the arena of engineering. Some of these algorithms are Evolution Strategies (ES), Evolutionary Programming (EP), Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE) and Artificial Bee Colony Algorithm (ABCA) etc. These algorithms are also known as Nature Inspired Algorithms because they are based on the intelligent simple rules of nature. The focus of the present study will be on DE, ABCA and their hybridized version named HDABCA.

While DE is similar to GA having the same operators, ABCA is a swarm intelligent based algorithm which em-

978-1-4577-1123-7/11/$26.00

ploys the foraging behavior of swarm of bees for solving numerical optimization problems. Both the algorithms have been successfully applied to a wide range of problems [7], [8], [9] etc. However, like other population-based optimization algorithms, these algorithms also suffer from certain drawbacks like long computational times because of their evolutionary/ stochastic nature. This crucial drawback sometimes limits their applications to offline problems with little or no real-time constraints. Also, these algorithms suffer from premature convergence. The objective of the present study is to judiciously combine the DE and ABCA so as to obtain an algorithm which is not only computationally fast but also maintains the quality of solution. Here we propose a new Hybrid Differential Artificial Bee Colony Algorithm (HDABCA) which combines ABCA with DE for finding the optimal parametric gain values of a Fractional order PI speed controller in a vector controlled PMSM drive.

II. Permanent Magnet Synchronous Motor

In general PMSM with approximately sinusoidal back electromotive force (i.e., back EMF) can be broadly categorized into two types 1) Interior (or buried) Permanent Magnet Synchronous Motors (IPMSM) 2) Surface mounted Permanent Magnet Synchronous Motors (SPMSM). In this paper we considered the SPMSM. In this type of motor the

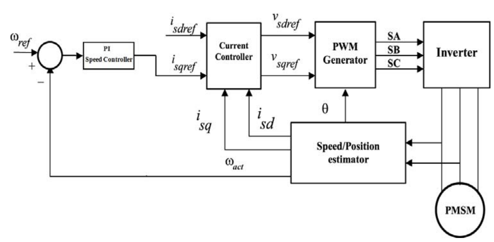

Fig. 1. Block diagram of PMSM Drive

magnets are mounted on the surface of the motor. Because the incremental permeability of these magnets is between 1.02-1.20 relative to external fields, the magnets have high reluctance and accordingly the SPMSM can be considered to have large and effective uniform air-gap. This property makes the saliency effect negligible. Thus quadrature axis synchronous inductance of SPMSM is equal to its direct axis inductance. As a result magnetic torque can only be produced by SPMSM, which arises from the interaction of magnet flux and quadrature axis current. The stator carries a three-phase winding which produces a near sinusoidal distribution of magneto motive force based on the value of stator current. They have the same role as the field winding in a synchronous machine except their magnetic field is constant and there is no control on it [10].

A. Mathematical Model of PMSM:

The PMSM drive model consists of a Pulse Width Modulation (PWM) inverter, a PWM generator, a current controller followed by speed controller and is also embedded with speed/position estimator. The schematic representations of these components are shown in Fig. 1. The PMSM drive receives power from three-phase AC supply and runs mechanical load at desired speed. The developed model of the drive system is used for design in current and speed controllers. The mathematical model of PMSM in *d-q* synchronously rotating frame of reference can be obtained from synchronous machine model. The PMSM can be represented by the set of following nonlinear [11] differential equations.

$$v_{sd} = r_s i_{sd} + p\lambda_{sd} - \omega_e \lambda_{sq} \quad (1)$$

$$v_{sq} = r_s i_{sq} + p\lambda_{sq} + \omega_e \lambda_{sd} \quad (2)$$

$$\lambda_{sd} = L_d i_{sd} + \lambda_m \quad (3)$$

$$\lambda_{sq} = L_q i_{sq} \quad (4)$$

$$T_e = \frac{3}{2}\frac{P}{2}[\lambda_m i_{sq} + (L_d - L_q) i_{sd} i_{sq}] \quad (5)$$

$$T_e = J\frac{2}{p}\frac{d\omega_e}{dt} + B\frac{2}{P}\omega_e + T_l \quad (6)$$

$$\omega_e = P\omega_r/2 \quad (7)$$

$$p\theta_r = \frac{2}{P}\omega_e \quad (8)$$

where v_{sq}, v_{sd}, i_{sq}, i_{sd} are *d-q* axis voltages and currents respectively. L_d, L_q are *d-q* axis inductances; λ_{sq}, λ_{sd} are *d-q* axis flux linkages; and ω_e, r_s are electrical speed of motor, and stator resistance respectively. λ_m is the constant flux linkage due to rotor permanent magnet; T_e is the electromagnetic torque; T_l is the load torque; P represents number of poles; p is the differential operator; B is the damping coefficient; θ_r is the rotor position; ω_r is the rotor speed; and J is the moment of inertia.

For constant flux operation when $i_{sd(ref)}$ equals zero [11], in vector-control technique or Field Oriented Control (FOC) the equations of PMSM are modified as

$$pi_{sd} = (v_{sd} - r_s i_{sd} + \omega_e L_q i_{sq})/L_d \quad (9)$$

$$pi_{sq} = (v_{sq} - r_s i_{sq} - \omega_e L_d i_{sd} - \omega_e \lambda_m)/L_q \quad (10)$$

$$\frac{d\omega_e}{dt} = \frac{1}{J}[\frac{P}{2}(T_e - T_l) - B\omega_e] \quad (11)$$

$$T_e = \frac{3}{2}\frac{P}{2}[\lambda_m i_{sq}] = K_t i_{sq} \quad (12)$$

III. Problem Formulation

A. Fractional Calculus: a brief-overview

Fractional Calculus (FC) is the branch of Mathematics, having 300 years of history. Recently this theory was applied to many fields of science and engineering [12], [13], [14], [15]. FC is a generalization of ordinary differential calculus which considers the possibility of taking real number power of differential and integration operator. There are many ways to describe fractional-order integrals and derivatives. The main concept of FC lies in developing a functioning operator D which is known as differ-integrator operator, according to Riemann-Liouville definition [16] it is mathematically defined as

$$_aD_t^\alpha f(t) = \frac{1}{\Gamma(m-\alpha)}\left(\frac{d}{dt}\right)^m \int_a^t \frac{f(\tau)}{(t-\tau)^{1-(m-\alpha)}} d\tau \quad (13)$$

Where $\Gamma(\cdot)$ is the Euler's `gamma` function in the range of $m-1 < \alpha < m$. Another alternative method of defining this D is by using the concept of fractional differentiation by Grünwald-Letnikov, which is given by

$$_{a}D_{t}^{\alpha} f(t) = \lim_{h\to 0} \frac{1}{\Gamma(\alpha)h^{\alpha}} \sum_{k=0}^{(t-a)/h} \frac{\Gamma(\alpha+k)}{\Gamma(k+1)} f(t-kh). \quad (14)$$

The generalized form of fractional order PI controller is the PI^{λ} controller, which involves an fractional integrator of order λ (can be any real number). The controller signal $u(t)$ can then expressed in time domain as

$$u(t) = K_P e(t) + K_I D^{-\lambda} e(t) \quad (15)$$

In the frequency domain the transfer function of this controller is given as

$$G_c(s) = K_P + \frac{K_I}{s^{\lambda}} \quad (16)$$

The interpretation of s^{λ} is that, on a semi-log plane, there is a line having slope of -20λdB/dec.

Where $\lambda = +1$ implies normal PI controller and $\lambda = 0$ implies proportional gain. All these different forms of PI controller are the special cases of fractional PI^{λ} controller. It is very interesting to note that FOPI controller generalizes the integer-order PI controller and expands it from point to plane. This will add more flexibility to controller design and the controlling of real word process is more precise and accurate.

IV. Digital realization of FOPI controller

The usual way of using transfer functions including fractional orders of s is to approximate it to integer order transfer functions. To actually approximate a fractional transfer function, an integer transfer function would have to involve an infinite number of poles and zeros (which is very difficult). But it is possible to obtain logical approximations with a finite number of zeros and poles.

Out of many real approximation methods, one of the well-known approximations is by Oustaloup method [17] which uses recursive distribution of poles and zeros. According to Oustaloup the transfer function is given by

$$s^{\alpha} \approx k \prod_{n=1}^{N} \frac{1+(s/\omega_{z,n})}{1+(s/\omega_{p,n})} \quad (17)$$

The approximation is valid in the frequency range of $[\omega_l, \omega_h]$. The value of poles and zeros (N) is fixed, The required performance of the approximation mainly depends on: truncated values causing simpler approximations, but may lead to ripples in both phase and gain regions. These ripples may be functionally removed by increasing poles and zeros count, ultimately leading to heavier computation. Required frequencies of poles and zeros in Eqn (17) is given according to

$$\omega_{z,1} = \omega_l \sqrt{\eta} \quad (18)$$

$$\omega_{p,n} = \omega_{z,n}\varepsilon,\ n = 1,\ldots,N \quad (19)$$

$$\omega_{z,n+1} = \omega_{p,n}\eta,\ n = 1,\ldots,N-1 \quad (20)$$

$$\varepsilon = (\omega_h/\omega_l)^{v/N} \quad (21)$$

$$\eta = (\omega_h/\omega_l)^{(1-v)/N} \quad (22)$$

In the case of $\alpha < 0$, this can be handled by reversing Eqn (17).

A. Formulation of the objective function

There are various performance criteria for design of controllers, of which some are integral absolute error (*IAE*), integral squared error (*ISE*). The main drawback of using these *ISE* and *IAE* criteria is a dynamic response with a relatively less overshoot but a heavy settling time because they will calculate the errors uniformly over time [18]. If integral-time-weighted-squared-error (*ITSE*) is used the drawback of prior mentioned methods can be eliminated, but it cannot ensure to have a desirable stability (and is also computationally complex). So, we employed Integral-time-weighted-absolute-error (*ITAE*) which has an advantage of producing lesser oscillations and overshoot along with less settling time, which is Mathematically defined as

$$ITAE = \int_0^{\infty} t(|\omega_{ref} - \omega_{act}|) = \int_0^{\infty} t(|e(t)|) \quad (23)$$

The performance of the drive depends on the fractional controller parameters values which indeed depend on the objective function to be minimized. So, to get a optimal set of parametric values for K_P, K_I, and λ to meet the user defined specifications for a given process call for real parameter optimization in three-dimensional hyperspace.

V. Hybrid Differential Artificial Bee Colony Algortihm

A. Artificial Bee Colony

In ABCA, each solution to the problem is considered as *food source* and represented by a D-dimensional real-valued vector, whereas the fitness of the solution corresponds to the *nectar amount* of associated food resource. The algorithm starts by initializing all employed bees with randomly generated food sources (solutions). In each iteration every employed bee finds a food source in the neighborhood of its current food source and evaluates its *nectar* amount i.e., (*fitness*). In general the position of i_{th} food source is represented as $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$. After the information is shared by the employed bees, *onlooker* bees go to the region of food source at X_i based on the probability P_i defined as

$$P_i = \frac{fit_i}{\sum_{k=1}^{FS} fit_k} \quad (24)$$

FS is total number of food sources. Fitness value fit_i is calculated by using following equation.

$$fit_i = \frac{1}{1+f(X_i)} \quad (25)$$

here $f(X_i)$ is the objective function to be minimized, in our problem ITAE. The *onlooker* finds its *food source* in the region X_i, by making use of following equation

$$x_{new} = x_{ij} + r * (x_{ij} - x_{kj}) \tag{26}$$

where $k \in (1,2,\ldots,FS)$ such that $k \notin i$ and $j \in (1,2,\ldots,D)$ are randomly chosen indexes. r is a uniformly distributed random number in the range $[-1,1]$.

Each bee will search for a better food source for a certain number of cycles (*limit*), and if the fitness value doesn't improve then that particular bee becomes scout bee. The food source is initialized to that *scout bee* randomly.

B. Differential Evolution

The working of DE is as follows, In a $D-$dimensional search space, for each target vector $x_{i,g}$ a mutant vector is generated by

$$v_{i,g+1} = x_{r_1,g} + F * \left(x_{r_2,g} - x_{r_3,g}\right) \tag{27}$$

where $r1$, $r2$, $r3 \in \{1,2,\ldots,NP\}$ are randomly chosen integers, different from each other and also different from the current running index i. The parameter $F(>0)$ is known as scaling factor which controls the amplification of the differential evolution $\left(x_{r_2,g} - x_{r_3,g}\right)$. In order to increase the diversity of the perturbed parameter vectors, crossover is introduced. The parent vector is mixed with the mutant vector to produce a trail vector $u_{ji,g+1}$, given by

$$u_{ji,g+1} = \begin{cases} v_{ji,g+1}\, if\,(rand_j \leq CR)\, or\,(j = j_{rand}) \\ x_{ji,g}\, if\,(rand_j > CR)\, and\,(j \neq j_{rand}) \end{cases} \tag{28}$$

Here $j = 1,2,\ldots,D$; $rand_j \in [0,1]$; CR is the crossover constant which take values in the range of $0 \leq CR \leq 1$ and $j_{rand} \in (1,2,\ldots,D)$ is the randomly chosen index. Selection is the key step to choose the vector between the target vector and the trail vector with the aim of creating an individual for the next generation.

C. Hybrid Differential Artificial Bee Colony Algorithm

In general hybridization schemes are broadly categorized into two types: the staged pipelining type hybrid and the other is additional-operator type hybrid. In the first type of hybridization, an optimization process is applied to each and every individual of the population, and the search space is further improved by using the second optimization method. In the second type of hybridization, the optimization algorithm is applied as a standard genetic operator for a given corresponding probability. In the present study, we have employed pipelining type hybrid method because of its advantages. In HDABCA, we apply ABCA to all individuals in the population from which we select the n best vectors based on the fitness values to generate the initial population for local search via DE. This search process continues till it stopping criteria is satisfied.

D. Design of FOPI Controller using HDABCA

The DE, ABCA and HDABCA are used to design the controller parameters K_P, K_I and λ such that the drive exhibits desired response and robust stability as evaluated by the design criteria. The search space of the controller parameters is three-dimensional and the three dimensions being K_P, K_I, λ. So, in HDABCA method for a given dimension the i_{th} food source is represented as

$$X_i = [K_P, K_I, \lambda]$$

- The parametric constants K_P and λ are set between $[0,1]$ and K_I is bounded in range 0 and 10
- ω_l and ω_h in (18) $-$ (22) are set to 10^{-5} and 10^{5} rad/s, respectively
- The approximation in (17) is set to $N = 3$

VI. Numerical Results

A. Design Specifications of the Drive

Like other motors PMSM drive also consists of certain design criterion to be satisfied, defined or set by the user. Since the basic requirement of a motor is that it should rotate at the desired speed before or on application on load, the steady-state error of the motor should be less than 0.01. The other performance requirement is that motor must accelerate to its steady-state speed as soon as it gets a power supply. In this case, we want it to have a settling time of $0.6sec$. Since a speed faster than the reference may damage the equipment, we want to have an overshoot of less than 2%. Simualtion is done for time $T = 1sec$ under a load torque of $5-Nm$ with a reference speed of $500\,rpm$.

B. Experimental Settings of PMSM

The following experimental settings are taken into account for the PMSM drive to perform the experimental simulations.

TABLE I
PARAMETER SETTING OF PMSM DRIVE

Variable	Actual implication	Value	Units
r_s	Stator resistance	2.0	Ω
L_{sd}	*d-axis* inductance	2.419	mH
L_{sq}	*q-axis* inductance	2.419	mH
J	Moment of inertia	0.00344638	kg-m^2
λ_m	Magnet mutual flux	0.27645	V/rad/sec
B	Damping coefficient	0.0027715	Nm/rad/sec
P	Number of poles	8	–

C. Comparative Simulations

1) Integer order PI controller: In this section the performance of PMSM controlled by Integer Order Proportional Controller (IO-PI) is investigated for different tuning algorithms. Fig.2 shows the speed response of PMSM before the application of load and Fig.3 shows the response after the load is applied (i.e., after T=0.5 sec). It is very clear from the plots and the comparisons Table that performance of PI controller is increased when designed compatibly by using hybrid Differential Artificial Bee colony algorithm. Also it

TABLE II

ALGORITHMIC PARAMETERS

Parameters	DE	ABCA
popsize	20	20
F	0.5	Not required
CR	0.8	Not required
Colony size	–	20
n_e	–	50% of colony
n_o	–	50% of colony
n_e	–	1
FS	Not required	10
limit	Not required	$n_e * D$

popsize, population size; CR, crossover factor in DE; F, scaling factor; n_e, employed bee number; n_o, onlooker bee number; n_s, scout number; FS, no of food sources (*position*); D, dimension of the problem

can be noticed that the Peak over shoot, settling time and steady state error is considerably decreased.

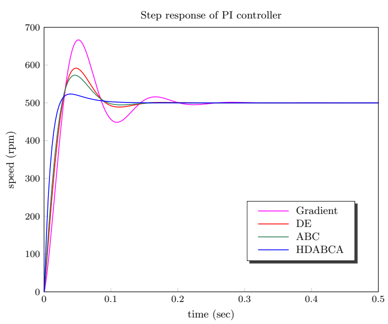

Fig. 2. Step response of PMSM controlled by IO-PI before load

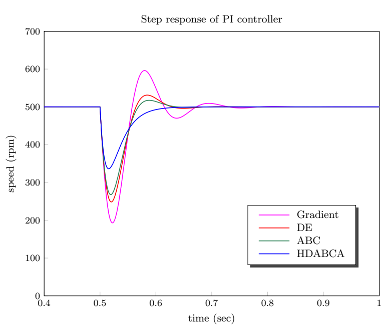

Fig. 3. Step response of PMSM controlled by IO-PI after load

2) Fractional order PI controller: After the extensive study of PI controller, we analyzed if the performance can be further improved. For this purpose we employed Fractional Order Controllers. This section provides the study of PMSM controlled by Fractional order Proportional Controller (FO-PI), for different proposed tuning algorithms. Fig.4 shows the speed response of PMSM before the application of load and Fig.5 shows the response after the load is applied (i.e., after T=0.5 sec). From the obtained responses and observations it is eminent that FO-PI controller designed with hybrid method performs better than conventional steepest descent method and the basic ABCA and DE algorithms. Almost every design constraint is satisfied with the proposed method.

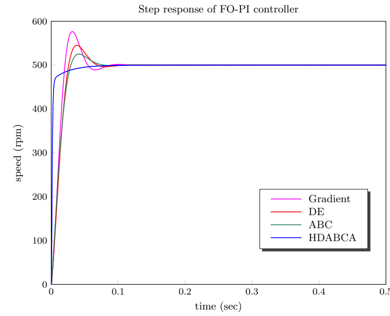

Fig. 4. Step response of PMSM controlled by FO-PI before load

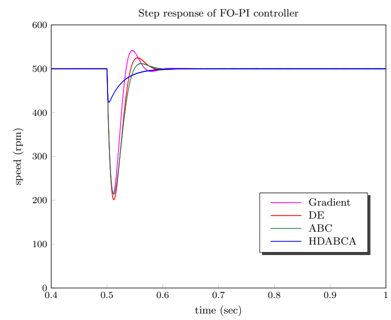

Fig. 5. Step response of PMSM controlled by FO-PI after load

3) Best IO-PI vs FO-PI: In this section we zoomed and compared the performance of best IO-PI controller and FO-PI controller, to resolve ambiguity, and to choose which of the controller is suited for an optimum performance. From the extensive simulations and comparisons it is quite interesting to note that PMSM controlled PI controller when designed with proposed method performs better than the remaining methods, but still there is dissatisfactory result in peak overshoot and settling time (with in a given bound). This is easily resolved by using the fractional order control due to which an additional constraint, rises time also got improved. Fig.6 and Fig.7 shows the step response of IO-PI and FO-PI controlled PMSM.

VII. CONCLUSIONS

In this paper, an intelligent optimization method, HD-ABCA (a hybrid version of DE and ABCA), for designing FO-PI controller for a PMSM drive is presented. Simulations and comparisons shown in the paper clearly indicate that a properly designed and implemented FO-PI controller provides better results than that of traditional integer-order PI controller. It can be seen that the performance of PMSM controlled by best tuned FO-PI controller is quite satisfactory in comparison to IO-PI controller. In this application, for FO-PI controller, Oustaloup approximation method was used for digital realization purposes. The proposed method has been shown to outperform a state-of-the-art version of the DE algorithm and ABCA method especially for the fractional-order controller. Our further research would include the performance and analysis fractional order controller in a sensor less motor.

TABLE III

COMPARISONS STEP RESPONSE OF PI CONTROLLER USING DIFFERENT METHODS

Method	K_P	K_I	$po(\%)$	t_r (sec)	t_s (sec)	e_{ss}
Gradient	0.0672	7.8764	33.3198	0.0216	0.6627	0.0301
DE	0.1027	8.5368	18.3619	0.0207	0.6134	0.0266
ABCA	0.1783	8.9728	14.6448	0.0199	0.6086	0.0273
HDABCA	0.2896	9.5124	3.6757	0.0152	0.5927	0.0059

TABLE IV

COMPARISONS OF STEP RESPONSE OF FO-PI CONTROLLER USING DIFFERENT METHODS

Method	K_P	K_I	λ	$po(\%)$	t_r (sec)	t_s (sec)	e_{ss}
Gradient	0.1142	6.731	0.2	15.3099	0.0146	0.5623	0.00792
DE	0.1815	7.9183	0.5	9.0316	0.0185	0.5718	0.00909
ABCA	0.2406	8.239	0.5	5.0251	0.0202	0.5683	0.00734
HDABCA	0.3592	8.8691	0.7	0.0210	0.0033	0.5521	0.00532

TABLE V

BEST IO-PI VS FO-PI

Controller	K_P	K_I	λ	$po(\%)$	t_r (sec)	t_s (sec)	e_{ss}
IO-PI	0.2896	9.5124	1	3.6757	0.0152	0.5927	0.0059
FO-PI	0.3592	8.8691	0.7	0.0210	0.0033	0.5521	0.00532

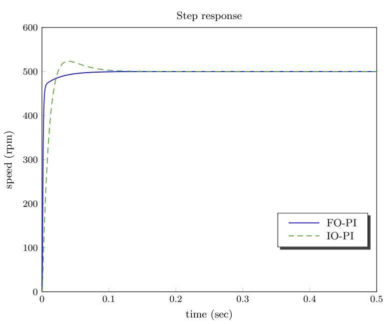

Fig. 6. Step response of PMSM controlled by PI and FO-PI before load

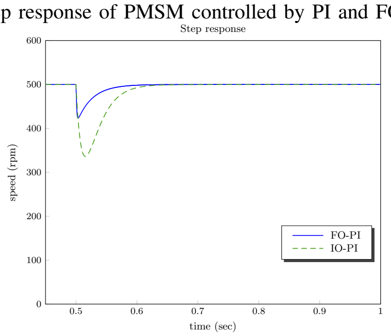

Fig. 7. Step response of PMSM controlled by PI and FO-PI after load

REFERENCES

[1] B.K. Bose, "Power electronics and motion control-Technology status and recent trends ", IEEE Trans. Ind. Applications, vol.29, 1993, pp.902-909.

[2] Peter Vas, *Sensorless and Direct Torque Control*, 1st Edition, Oxford University Press; 1998.

[3] W. Leonhard, "Microcomputer control of high dynamic performance ac drives - A survey ", Automatica, vol.22, 1986, pp. 1-19.

[4] J. J. Slotine, Li, W, *Applied Nonlinear Control*. Englewood Cliffs., NJ: Pearson Education, 1991.

[5] K. J. Astrom, T. Hugglund, *The Future of PID Control*. Control Eng. Pract., 9, 1163-1175, 2001.

[6] I. Podlubny, Fractional-order Systems and Fractional-order Controllers, The Academy of Sciences Institute of Experimental Physics, UEF-03-94, Kosice, Slovak Republic, 1994.

[7] D. Karaboga, "An Idea based on Bee Swarm for Numerical Optimization ". Technical Report, TR-06, Erciyes University Engineering Faculty, Computer Engineering Department, 2005.

[8] S. Das, A. Biswas, Ajith Abraham, and S. Dasgupta, "Design of Fractional Order PI$^\lambda$D$^\mu$ Controllers with an Improved Differential Evolution". Engineering Applications of Artificial Intelligence, Elsevier Science, 22 (2), 343-350, 2009.

[9] D. Karaboga, B. Basturk, "A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) algorithm". Journal of Global Optimization, Springer Netherlands, 39, 459-471, 2007.

[10] T. Sebastian, G. R. Slemon, "Transient Modeling and Performance of Variable- Speed Permanent-Magnet Motors ", IEEE Trans. on Ind Applications, 25(1), 1986, pp.101-106.

[11] P. Pillay, R. Krishnan, "Modeling, Simulation, and Analysis of Permanent-Magnet Motor Drives, Part I: The Permanent-Magnet Synchronous Motor Drive". IEEE Trans. on Ind Applications, 25(2), 1989, 265-273.

[12] M. Zamani, M. K. Ghartemani, N. Sadati, and M. Parniani, "Design of a fractional order PID controller for an AVR using particle swarm optimization", Control Eng. Pract., (17), pp.1380-1387, 2009.

[13] K. B. Oldham, J. Spanier, The Fractional Calculus. AcademicPress, NewYork; 1974.

[14] I. Petras, "The fractional order controllers: methods for their synthesis and application". Journal of Electrical Engineering 50(910), pp.284288,1999.

[15] I. Podlubny, "Fractional-order systems and $PI^\lambda D^\mu$ -controllers, *IEEE Trans. Automatic Control*, vol.44, 1999,pp.208-214.

[16] YangQuan Chen, Dingyu Xue and Huifang Dou. "Fractional calculus and biomimetic control". *In Proc. of the First IEEE Int. Conf. on Robotics and Biomimetics (RoBio04)*, pages robio2004 347, Shengyang, China, August 2004. IEEE.

[17] A. Oustaloup, F. Levron, B. Mathieu, and F.M. Nanot, "Frequency band complex noninteger differentiator: characterization and synthesis". *IEEE Trans. Circuits Syst. I*, vol. 47, 2000, pp.25-39.

[18] R. A. Krohling, J. P. Rey, "Design of Optimal Disturbance Rejection PID Controller Using Genetic Algorithms", IEEE Trans. on Evolutionary Computation, 5, 2001, 78-82.