ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

Tuning of Fractional Order PI Controller for DC-DC Boost Converter Using Different Performance Indices

[1]R. Senthilkumar and [2]V. Manikandan

[1]*Research Scholar, Faculty of Electrical Engineering Anna University, Chennai, India*
[2]*Professor, Department of Electrical and Electronics Engineering, Coimbatore Institute of Technology, Coimbatore, India*

ARTICLE INFO

Article history:
Received 12 March 2015
Accepted 28 April 2015
Available online 5 May 2015

Keywords:
Boost converter, FOPI, GA, PSO, QBGA, Performance criteria

ABSTRACT

Background: In this paper, a Fractional Order Proportional-Integral controller has been proposed for analysis the dynamic response of the boost converter based on the various performance indices. Here, different metaheuristic control algorithm, like GA, PSO, Queen Bee assist GA algorithm (QBGA) is implemented to tune the controller. The search is to find out optimal controller parameters like K_p, *Ki* and *w*hile minimize the cost function of various integral error indices like ISE, IAE, ITAE, and ITSE.The performance parameters of boost converter and controller efforts in each integral error criteria are compared and effectiveness of the different controllers is evaluated. These can be obtained by using MATLAB/Simulink environment. Simulation results demonstrate that gains and orders optimization leads to a better transient response of the proposed fractional PI controller

© 2015 AENSI Publisher All rights reserved.

To Cite This Article: R.Senthilkumar and Dr.V.Manikandan, Tuning of Fractional Order PI Controller for DC-DC Boost Converter Using Different Performance Indices. ***Aust. J. Basic & Appl. Sci.,*** *9(11): 207-213, 2015*

INTRODUCTION

For many decades, PID controllers are commonly used in the most control industry due to their simplicity, low cost, and ability to solve most of the control problems was formulated by (Astrom *et al* 1995).It also has good dynamic performance, such as small settling time and low percentage overshoot. Due to this superior, PID controllers are still the most common controller.

The performance of PID controller can be improved by the use of a fractional order controller where integral and derivative actions have, in general, non-integer orders. The application of fractional controllers in different areas of a control system has been proposed by (D. P. Atherton *et al* 1999 and S. Majhi *et al* 1999) are increased in recent years. Compare to conventional PID controller, FOPID controller has two more adjustable parameters. By itself, the fractional $PI^{\lambda}D^{\delta}$ controllers provide a more flexible tuning strategy that can achieve optimal performance for a given system as per requirements. This paper attempts to study the behavior of fractional order PI controller for the proposed DC-DC boost converter system.

In this paper, a Fractional Order PI controller has been used to regulate the output DC voltage of a boost converter within a specified tolerance limit and improve the converter's dynamic performance. A very important aspect of designing FOPI controllers is to decide upon the values of K_P, K_i and λ.In order to get an optimum time domain specifications, the parameters of the fractional order PI^{λ} controller are tuned by minimization of different integral error criterion.

The present work proposes a design of the fractional order PI controller for a boost converter using optimization technique. The contribution of this work consists mainly in the design of feedback fractional order PI controller based on the minimization of different integral error criterion for various control algorithms and hence the controller parameters K_P, K_i, and λ are identified.

In order to solve the minimization of objective function, evolutionary computation techniques, such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Queen Bee assist GA algorithm (QBGA) have been successfully applied. The Simulink model of a boost converter is working to evaluate the objective function along with the evolutionary algorithm gives robust feedback fractional order PI controller. The development and implementation of the proposed system and controller are done using MATLAB/Simulink environment.

Corresponding Author: R.Senthilkumar, Research Scholar, Faculty of Electrical Engineering, Anna University, Chennai, Tamil Nadu, India.
E-mail: rsenthilkumarpe@gmail.com

2.0. Fractional Order Controllers:

Fractional order calculus is an area where the mathematicians deal with derivatives and integrals from non-integer orders have discussed by (S. E. Hamamci *et al* 2007 and I. Podlubny *et al* 1999). Podlubny has proposed a generalization of the PID controllers called fractional controller is used to improve the robustness and performance of a conventional controller. There are many different definitions of Fractional Order differentiations and integrations. Some of the definitions extend directly from integer-order calculus. The Grunewald-Letnikov, Riemann-Liouville, and Caputo statements are given by (I. Podlubny *et al* 1994) are the well-established definitions of fractional derivatives and integrals in fractional calculus. These definitions are given by

Grunewald-Letnikov definition: This definition is defined as

$$ {}_{a}^{\alpha}D_{t} f(t) = \lim_{h\to 0} \frac{1}{h^{\alpha}} \sum_{j=0}^{\left[\frac{t-a}{h}\right]} (-1)^{j} \binom{a}{j} f(t - jh) \tag{1}$$

Where $w_j^{\alpha} = (-1)^j \binom{a}{j}$ represents the coefficients of a polynomial $(1-z)^{\alpha}$

The coefficients can also be obtained from

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

Riemann-Liouville definition: The fractional order integration is defined by

$$ {}_{a}^{-\alpha}D_{t} f(t) = \frac{1}{\Gamma(\alpha)} \int_{a}^{t} (t-\tau)^{\alpha-1} f(\tau) d\tau \tag{2}$$

Where $0 < \alpha < 1$ and a is the initial time instance, assumed to be zero, i.e., a =0

Controlling industrial plants requires satisfaction of a wide range of a specification. Mostly for industrial applications, integer order controllers are used for controlling purpose. Now day's Fractional Order (FOPI) controller is used for industrial application to improve the system control performances. The most common form of a fractional order PI controller is the PI^{λ} controller has been discussed by (Lee KY *et al* 2002).FOPI controller provides an extra degree of freedom for tuning gives the better control performance. The orders of integral and derivative are not necessarily an integer, but any real numbers.

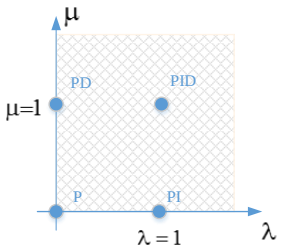

Fig. 1: Generalization of a fractional order PID controller.

As depicted in Fig.1, The FOPID controller generalizes the conventional integer order PID controller and expands it from the point to a plane. This expansion could provide much more flexibility in PI controller design. The continuous transfer function of a fractional controller with three adjustable parameters is given by the following function has been given by (Arijit Biswas *et al* 2009).

$$G_C(S) = K_p + \frac{K_i}{S^{\lambda}} \tag{3}$$

It is Clear, by selecting λ = 1 and μ = 1, a classical PID controller can be recovered. By selecting a suitable value of λ andμ, we get corresponds to the conventional PI & PD controllers. All these classical types of PID controllers are special cases of the $PI^{\lambda}D^{\mu}$ controller. The most common form of a fractional order PI controller is the PI^{λ} controller was discussed by (Y.Q. Chen *et al* 2002), involving an integrator of order λ where λ can be any real numbers. The continuous transfer function of fractional controller is obtained through Laplace transform and is given by,

$$G_C(S) = \frac{U(S)}{E(S)} = k_p + k_1 \frac{1}{S^{\lambda}}, (\lambda > 0) \tag{4}$$

Where $G_c(S)$ is the transfer function of the controller, E(S) is an error signal, and U(S) is controller's output. The differential equation of a fractional order controller is described in time a domain as.

$$u(t) = k_p e(t) + k_I D^{-\lambda} e(t) \tag{5}$$

It can be expected that the PI^{λ} controller may enhance the systems control performance. One of the most significant advantages of the PI^{λ} controllers is the better control of dynamical systems, which are described by fractional order mathematical models. Another advantage lies in the fact that the PI^{λ} controllers are less sensitive to changes of parameters of a controlled system has been described by (D. Xue *et al* 2006).

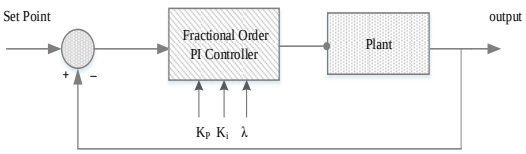

Fig. 2: Block diagram of FOPI controller.

3.0. DC-DC Boost Converter:

The DC-DC boost converters are widely used in many applications e.g., dc motor drives, power systems etc. The output voltage of a converter is easily influenced by external perturbations, when it is employed as a front end converter. The objective of the control of a DC to DC converter is to achieve a constant output voltage despite input voltage disturbances and load variations. Thus, it is necessary to provide regulated output voltage. The boost converter is the basic dc-dc converter with an output voltage greater than its input voltage.

Many control strategies, both linear and non-linear, have been used for the control of boost converter such as PI, Sliding mode control etc T. Habetler *et al* 2001. The control strategy lies in the manipulation of the duty cycle of the switch which causes the voltage change have been discussed by (Martins, D *et al* 2002, and Rajib Baran Roy 2012).

The behavior of boost converter is described by state averaging equation during ON state is

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix}=\begin{bmatrix}\frac{-r_L}{L} & 0\\ 0 & \frac{-1}{C(R_L+r_c)}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}+\begin{bmatrix}\frac{1}{L}\\ 0\end{bmatrix}V_{in} \quad (6)$$

The output voltage of DC-DC boost converter during ON state is,

$$v_o=\begin{bmatrix}0 & \frac{R_L}{(R_L+r_c)}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} \quad (7)$$

The OFF state of the boost converter is

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix}=\begin{bmatrix}-\left(\frac{r_1+R_L/r_c}{L}\right) & -\frac{R_L}{L(R+r_c)}\\ \frac{R_L}{C(R+r_c)} & -\frac{x_2}{C(R_L+r_c)}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}+\begin{bmatrix}\frac{1}{L}\\ 0\end{bmatrix}V_{in} \quad (8)$$

The output voltage of DC-DC boost converter during OFF state is,

$$v_o=\begin{bmatrix}r_c/R_L & \frac{R_L}{(R_L+r_c)}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} \quad (9)$$

In the above, the state variables x_1 and x_2 are inductor current i_L and capacitor voltage V_c, respectively. The above state equation is modeled in MATLAB tool and gets the dynamic characteristic of a boost converter.

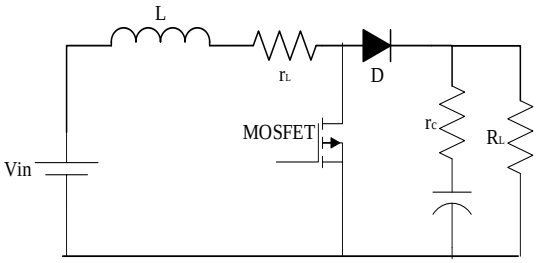

Fig. 3: DC-DC boost converter topology.

A closed-loop boost converter using an MOSFET as a switching element is shown in the Fig.3. The design specifications of the proposed DC-DC boost converter is taken from (Kinattingal Sundareswaran *et al* 2009), is input voltage, V_{in} = 36V; switching frequency, f_s = 2 kHz; inductance L = 33mH; equivalent resistance of the inductor, r_L = 3Ω; capacitance, C = 1000μF; equivalent resistance of the capacitor, r_c =0.5Ω, and load resistance, R_L = 100 Ω. In the feedback control system, the actual output voltage V_0, and its reference value V_{ref} are fed into the controller, e(t) so obtained is processed by the proposed FOPI controller.

4.0. Formulation of the Objective Function:

The integral error is normally accepted as a good measure of the system performance. The following are some commonly used control quality criterion based on the time domain integral error. The error criterion is measured as a cost function, and the values of fractional controllers are continuously tuned until the objective function of the closed loop converter is attained minimum. Here various time domain integral indices (10) considered as a cost function to evaluate the performance of a controller.

The minimization of performance indices is defined as,

$$\left.\begin{aligned} IAE &= \int_0^T |e(t)|dt, \qquad ISE = \int_0^T \left(e(t)\right)^2 dt \\ ITAE &= \int_0^T t\,|e(t)|dt, \qquad ITSE = \int_0^T t\left(e(t)\right)^2 dt \end{aligned}\right\} \tag{10}$$

Where $e(t) = \left[V_{ref} - V_0\right]$ is the error signal in a time domain

The aim of optimization is to find the optimal values of PI^{λ} controller in order to minimize the all four performance indices (10) and compare them to find the most suitable one. Then fitness function is given by

$$fitness\ value = \frac{1}{performance\ index} \tag{11}$$

5. 0.Tuning of FOPI Controller:

Optimization is carried out to obtain the fractional controller parameters like proportional gain k_p, a constant of integral term k_i and the order of fractional integrator λ. The minimization of control objectives (10) gives optimal parameters of controllers. This can be achieved by tuning the controller parameter with a different optimization technique like GA, PSO, and QBGA.

5.1. GA Tuning:

In 1975, Holland proposed GA as a heuristic method based on "survival of fittest" The genetic algorithms are part of an evolutionary algorithm, which are computational models inspired by natural evolution. Genetic algorithms (GA) are a class of stochastic search algorithms that start with a population of randomly generated candidates and 'evolve' towards better solutions.GA is a method for moving from one population of chromosomes to a new population using natural selection.

Implementation of GA for a problem starts with parameter encoding (i.e., the representation of the problem). Generally, binary strings are used to represent the decision variables of the optimization problem. Each individual in the genetic population represents a candidate solution. Here, the genetic algorithm is used to calculate the optimal control parameters of a fractional controller while minimizing the integral error.

Steps for the Genetic Algorithm:

1. Initialization: Generate random population of n chromosomes. These chromosomes represent the value of k_p, k_i and λ.
2. Fitness function: Evaluate the fitness f(x) of each chromosome x in the population using the evaluation function given by (10).
3. Create the new population by repeating the following steps until the new population is complete. Update each individual's fitness value $p_{best.}$
1. Selection: Select two parent chromosomes from a population according to fitness a function.
2. Crossover: Crossover the two parents to form new offsprings.
3. Mutation: Mutate new offspring at each position in the chromosome.
4. The comparison is made between parents and offsprings. Select the better chromosome as parents of next generations.
5. If the end condition (number of iterations) is satisfied, stop and return the best solution in the current population. Otherwise, the above steps are repeated.

Fractional order controller parameters will be optimized by applying GA. The algorithms are implemented in MATLAB Genetic Algorithm Toolbox.

5.2. PSO Tuning:

Particle swarm optimization is a population-based stochastic search algorithm which is the most recent development of metaheuristic optimization technique. It was first introduced by Kennedy and Eberhart in 1995.The merits of the PSO are its simplicity of the algorithm as it involves two equations (12), (13) and stable convergence characteristic of good computational efficiency.

The PSO consists of a swarm of particles moving in an *N*-dimensional solution space where a certain quality measure and fitness are being optimized. Each particle has a position represented by a position vector $S_i = (S_{i1}, S_{i2\ldots}\ S_{iN})$ and a velocity represented by a velocity vector $V_I = (V_{i1}, V_{i2}\ldots V_{IN})$.The maximum velocity $V_{max} = (V_{max1}, V_{max2}\ldots V_{maxN})$. Each particle remembers its own best position so far in a vector $P_i = (P_{i1}, P_{i2}\ldots P_{iN})$.The best position vector among all the neighbors of a particle is then stored in the particle as a vector $G_i = (G_{i1}, G_{i2}\ldots G_{iN})$. The modified velocity and position of each particle can be manipulated according to the following equations:

$$V_i^{t+1} = W \cdot V_i^t + C1 \cdot R1 \cdot \left(P_i^t - S_i^t\right) + C2 \cdot R2 \cdot \left(G_i^t - S_i^t\right) \tag{12}$$

$$S_i^{t+1} = S_i^t + V_i^{t+1} \tag{13}$$

Where *W* can be expressed by the inertia weights approached by (Y. Shi *et al* 1998) and often decreases linearly from *Wmax* to *Wmin.* In general, the inertia weight *w* is set according to the following equation

$$W^t = W_{max} - \left[\frac{\left(W_{max} - W_{min}\right)}{Iter_{max}}\right].iter \tag{14}$$

Where $iter_{\max}$ represents the maximum number of iterations, and *iter* is the number of current iteration or generation. Also, *c*1 and *c*2 are the acceleration constants which influence the convergence speed of each particle, *r*1 and *r*2are random numbers in the range of (0, 1), respectively.

Steps in PSO Based FOPI Controller Optimization:

Step 1: Initialize the swarm from the solution space. Initialize a swarm of particles with random velocity and position in the solution space.

Step 2: Evaluate the objective function of every particle. Evaluate the fitness value of each particle in the population. The advantage of each particle in the swarm is found by using an objective function called evaluation function. This function is defined to minimize the cost function given by (10).

Step 3: Compare the fitness of particle with *pbest* and if the current value is better than *pbest* replace the local best value.

Step 4: updating: Modify the current velocity and position of the particle as per (12) and (13).

Step 6: The above steps are repeated until the number of iterations reached. Record the optimized k_p , *ki and* λ.

Step 7: perform closed loop test with optimized values of controller parameters and calculate time domain specification for the converter.

The solution space is a three-dimensional, being k_p, k_i *and* λ. So each particle has three-dimensional positions and velocity vector. The optimization performed with this initial parameter, number of particles 20, a maximum iteration 50, the number of dimensions 3, with the objective function (10).

5.3. Queen Bee Based GA Tuning:

The queen bee based GA algorithm is one among the different type of GA implementations have been proposed by (Sung Hoon Jung 2009).The QBGA is an algorithm based on the interactions of bees in a hive. Basically, there exists a single queen bee with which all other bees, known as drones, mate. Occasionally a female bee is produced that ousts the current queen becomes the new queen. The software implementation of this algorithm is done by creating drones with randomized genes. The sequential steps of the Bees GA are given as follows.

1. Generation of bees In the initialization, bees are randomly generated in the feasible search space.
2. Identification of a queen bee Among the randomly generated bees, a queen bee is selected as the one which has the best binary structure in minimizing the function (10).
3. Drones mate with the queen, produce two offsprings. Among the two, only the fittest alone survives and the other one is discarded and is equivalent to killing by a virgin queen bee.
4. Every virgin queen competes and best competes with a current queen; whichever survives becomes new nest queen.
5. Terminate the program and select the new queen bee as the optimum solution, if termination criterion is reached; else go to step 3. The termination criterion is taken as 50 iterations.
6. This iteration is repeated several times by regenerating the drones until the best optimal solution is found.

The parameter for a genetic algorithm with queen bee evolution algorithm is shown in Table.1.

Table 1: parameters for Experiments.

Parameters	Values
Crossover probability (p_c)	0.6
Mutation probability (p_m)	0.09
Population size	10
Recombination probability (p_r)	0.8
No of iteration	50

6.0. Simulation Results and Discussion:

The performance of the boost converter with proposed controller under transient conditions is verified by using optimization technique such as genetic algorithm, particle swarm optimization and queen bee based GA algorithm.Table.2 summarizes the parameters of different cost functions, like ISE, IAE; ITAE and ITSE for boost converter under various controller optimization techniques. Also, the time domain parameters of boost converter for all tuning methods are reported in Table.2.

The simulation results for a different controller algorithm based on various performance indices are shown in figure (4-6).

6.1. Comparison of Transient Characteristics:

Table 2: Optimal parameters and Performance indices of various tuning methods.

CONTROLLER	PER.IND	K_p	K_i	λ	t_r	t_s	t_p	E_{ss}	Minima of performance indices
GA-FOPI	ISE	0.2824	10.2315	0.9172	0.218	0.3755	0.3755	0.43	112.2
	IAE	0.4984	17.558	0.6991	0.0361	2.9793	0.0769	1.07	5.65
	ITAE	0.6324	28.7252	0.9572	0.0415	0.0365	0.1222	0.5526	2.577
	ITSE	0.4984	17.558	0.6991	0.0361	2.9793	0.0769	1.07	6.429
PSO-FOPI	ISE	0.1336	32.5579	0.9839	0.073	0.1587	0.1285	0.22	87.49
	IAE	4.761	7.6836	1.1789	0.0361	1.4402	2.4269	0.5649	4.046
	ITAE	4.1463	8.5365	0.9121	0.0356	1.2737	2.9789	0.7188	5.527
	ITSE	2.3609	8.6571	1.0172	0.0332	0.8822	2.3458	0.3293	12.77
QBGA-FOPI	ISE	0.3685	34.0936	1.0215	0.0657	2.9883	0.1098	0.1567	83.28
	IAE	0.0086	26.7149	0.8731	0.0341	2.9192	1.3123	0.0493	4.02
	ITAE	0.6505	30.0409	1.0906	0.09	0.3628	0.1998	0.2121	2.41
	ITSE	0.7667	33.5822	1.0646	0.0325	0.2753	0.1823	0.2967	2.788

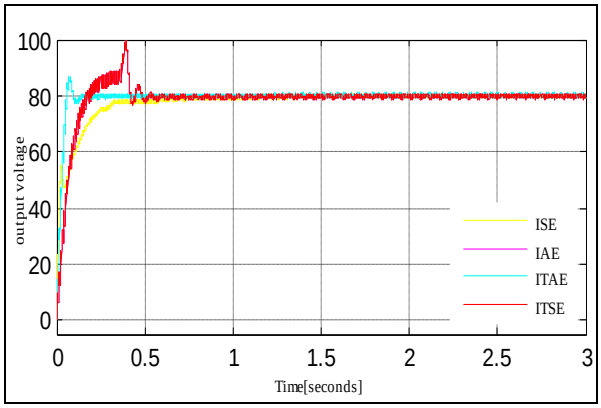

Fig. 4: Genetic algorithm FOPI controller.

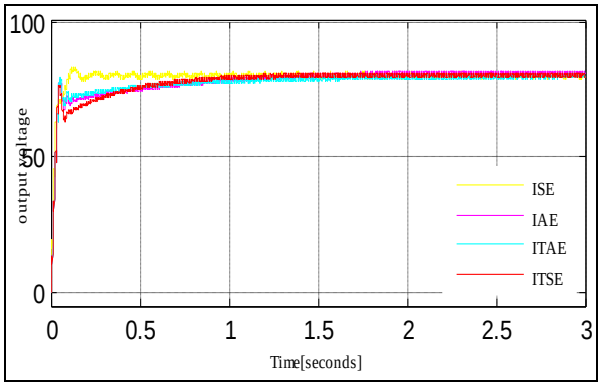

Fig. 5: PSO fractional order PI controller.

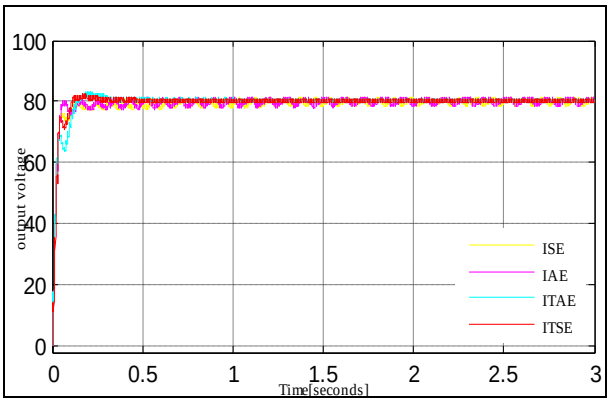

Fig. 6: Queen Bee based GA fractional PI controller.

All of the parameters are obtained from different algorithms for various performance indices. Each cost function produces different optimum controller parameters based on optimization technique. The bold values in the Table 2 show the minimum value of a concern transient characteristic for all type of integral error criteria for different controller's technique.

.

From the Table.2 the following results are obtained:

It can be seen from the table that almost all optimization methods, the rise time of all performance indices have more or less same value.

GA Tuning Method:

1. Both IAE and ITSE performance indices have same transient response characteristics.
2. The rise time of both ISE and ITAE is the same value.
3. Steady state error of ITAE function is less as compare other three performance indices.

PSO Tuning Method:

1. The overall transient response of ITSE function is less except peak time.
2. The PSO search can achieve faster search and better solutions as compared with GA.

QBGA Tuning Method:

1. Rise time, settling time of ITSE is less compared to other performance indices
2. Minimum peak time and steady state error in ISE and IAE function respectively as compared to other indices.
3. Queen Bee based GA algorithm is quickly approached the global optimum value as compared to conventional GA and PSO.

7.0. Conclusions:

The design of a controller for the boost converter is perceived as an optimization task and the controller constants are estimated through evolutionary algorithms. The dynamic performance of the boost converter is compared with a different metaheuristic control algorithm based on minimization of a variety of performance indices. To compare the performance, the design problem of a controller is considered as an optimization problem. GA, PSO, and QBGA optimization techniques are employed for compute the tuning parameters of fractional order PI controller.

With the help of fractional order PI controller, the response of the boost converter is designed much more flexibility. It is observed from the characteristic table that the controller performance depends on the type of optimization technique to be used and also on the choice of integral performance indices.

REFERENCES

Arijit Biswas, Swagatam Das, Ajith Abraham and Sambarta Dasgupta, 2009.”Design of fractional-order $PI^{\lambda}D^{\mu}$ controllers with an improved differential evolution”. Engineering Applications of Artificial Intelligence, 22(2): 343-350.

Atherton, D.P. and S. Majhi, 1999.”Limitations of PID Controllers”. American Control Conference, 6: 3913-3917.

Xue, D., C. Zhao and Y.Q. Chen, 2006. “Fractional Order PID Control of a DC Motor with Elastic Shaft: A Case Study”. Proc. of the American Control Conference, Minneapolis, Minnesota, USA, 14-16: 3182–3187.

Podlubny, I., 1999. “Fractional-order systems and $PI^{\lambda}D^{\mu}$-controllers”. IEEE Trans. Autom. Control, 44(1): 208-214.

Podlubny, I., 1999.”Fractional Deferential Equations”. Academic Press, SanDiego.

Podlubny, I., 1994.”Fractional-order systems and fractional-order controllers”. Preprint No UEF 0394, Sloval Acad. Sci. - Institute of Experimental Physics, 18.

Astrom, K. and T. Hagglund, 1995. “PID Controllers: Theory, Design, and Tuning”. Research Triangle Park, NC: Instrument Society of America, North Carolina.

Kennedy, J., R. Eberhart, 1995.”Particle swarm optimization”. In: Proceedings of the IEEE international conference of neural network (ICNN’95), IV: 1942–8.

Kinattingal Sundareswaran and V.T. Sreedevi, 2009. “Boost converter controller design using Queen-Bee-Assisted GA”. IEEE Transaction on industrial Electronics, 56-3.

Lee, K.Y., M.A. EI-Sharkawi, 2002.”Modern heuristic optimization techniques with applications to power systems”. IEEE Power Engineering Society.

Martins, D., C. Weber and R. Demonti, 2002. “Photovoltaic power processing with high Efficiency using maximum power ratio technique”. Proc. 28th IEEE IECON, 2: 1079-1082.

Rajib Baran Roy, 2012. “Design and performance analysis of the solar PV DC water pumping System”. Canadian Journal on Electrical and Electronics Engineering, 3-7.

Hamamci, S.E., 2007. “An algorithm for stabilization of fractional-order time delay systems using fractional-order PID controllers”. IEEE Trans. Autom. Control, 52(10): 1964-1969.

Majhi, S., 1999.”Relay feedback Process Identification and Controller Design”. Ph.D Dissertation, Univ. of Sussex, Brighton, UK.

Sung Hoon Jung, 2003. “Queen-bee Evolution for Genetic Algorithms”, In Electron Letters, 39(6): 575-576.

Habetler, T., R. Harley, 2001.”Power electronic converter and system control”. Proceeding of the IEEE, 89(6): 913–924.

Shi, Y., R. Eberhart, 1998.”A modified particle swarm optimizer”. In: Proc.IEEE Int. Conf. on Evolutionary Computation, 69-73.

Chen, Y.Q., K.L. Moore, 2002. “Discretization schemes for fractional differentiators and Integrators”. IEEE Transactions on Circuits System 1: Fundamental Theory Applications, 49(3): 363-367.