Research Journal of Applied Sciences, Engineering and Technology 10(7): 750-757, 2015
DOI:10.19026/rjaset.10.2427
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: July 18, 2014 Accepted: October 17, 2014 Published: July 10, 2015

Research Article

Evaluation of Arm Processor-based Bionic Intelligent Controller for a Buck-boost Converter

M.V. Mini and L. Padma Suresh
Electrical and Electronics Engineering, Noorul Islam Centre for Higher Education, Kumaracoil, Tamil Nadu, India

Abstract: This study focuses on performance-comparison of different tuning methods for a PI controller applied to a buck-boost converter. Comparison between the controllers is made by analysis of design methodology implementation issues and empirically measured performance. Design of PI controller is based on frequency response of the converter. The optimization of PI controller is based on ant colony algorithm. Experimental results show that, tuning the PI controller using ACO algorithm gave better performance than the conventional algorithm. This is mainly due to the fact ACO is capable of reducing the overshoot without oscillation.

Keywords: Ant colony algorithm, ARM processor, buck boost converter, PI controller

INTRODUCTION

DC-DC converters are been widely used in computer hardware and many industrial applications. They are also the dc-dc converters are of two types a buck converter, which decreases the voltage level from an input dc source and a boost converter which increases the voltage level from an input dc source. Erickson (1991) has given that a buck-boost converter is a combination of buck and boost converter which can decrease or increase the voltage level from an input dc source. The buck-boost converter has a Right Half-Plane (RHP) zero and hence is also called Non Minimum Phase (NMP) system. Deriving a control system for a non-minimum phase system is more difficult than that for a minimum phase system as discussed in Siotine and Li (1991) and Arulselvi *et al.* (2004). It is necessary, that designing a controller for a non-minimum phase system requires avoiding the cancellation of unstable pole/zero to guarantee the internal stability of the closed-loop system. Conventionally, PI (or PID) control logic is proposed for the buck-boost converter using root-locus or pole-placement methods as discussed by Alvarez-Ramirez *et al.* (2001), Guo *et al.* (2002) and Tsang *et al.* (2008). These controllers are popular due to their simple structure, and they can usually provide a good closed loop response characteristic. However, these methods fail to determine suitable PI (or PID) control gains when the system parameters are uncertain.

Proportional-Integral-Derivative (PID) controllers are frequently used in the control process to regulate the time domain behavior of different types of dynamic plants. Despite its simple structure, it seems so hard to find a proper PID controller as discussed by Oonsivilai and Pao-La-Or (2008). Various methods have been proposed to enable us to tune these parameters that will handle this issue.

Ziegler-Nichols tuning method is the most standard one used. However it is often difficult to find optimal PID parameters with these methods. Therefore, many optimization techniques like fuzzy logic (Tzafestas and Papanikolopoulos, 1990; Visioli, 2001), neural network as discussed by Cao *et al.* (2007), neural-fuzzy logic as discussed by Seng *et al.* (1999), immune algorithm as discussed by Kim (2001), simulated annealing as discussed by Zhou and Birdwell (1994), and pattern recognition are developed to tune the PID controllers. Several optimum tuning PID techniques based on many random search techniques such as Genetic Algorithm (GA) as discussed by Wang and Kwok (1994) and Mitsukura *et al.* (1999), Particle Swarm Optimization (PSO) as discussed by Selvan *et al.* (2003) and Ant Colony Optimization (ACO) are also used as given in Hsiao *et al.* (2004).

Researchers have acknowledged the capacity of ACO to search for and locate an optimum solution. This method is mainly inspired by the fact that ants can find the shortest route between their nest and a food source. Ant Colony Optimization as discussed by Dorigo *et al.* (1999) and Dorigo and Caro (1999) is yet another metaheuristic approach for solving combinatorial optimization problems. A few illustrations of such problems can be found with the Travelling Salesman Problem (TSP) as discussed by Reinelt (1994), quadratic assignment problem as discussed by Stützle

Corresponding Author: M.V. Mini, Electrical and Electronics Engineering, Noorul Islam Centre for Higher Education, Kumaracoil, Tamil Nadu, India

and Dorigo (1999), graph coloring problems as discussed by Costa and Hertz (1997), hydroelectric generation scheduling problems as discussed by Huang (2001), vehicle routing in Gambardella *et al*. (1999).

PI and PID controller have been widely used in DC-DC converters, mainly due to their simplicity. This paper focuses on optimizing a PI controller for Buck-Boost Converter using Ant Colony Algorithm (ACA).

MATERIALS AND METHODS

Buck boost converter: A Buck-Boost converter is a step-down and step-up DC-DC converter. The output of Buck-Boost converter's output is regulated based on the duty cycle of the Pulse Width Modulation (PWM) input at fixed frequency. Whenever the duty cycle (dc) is less than 0.5, the output voltage of the converter will be lower than the input voltage. However, when the duty cycle is above 0.5 the output voltage of the converter is higher than the input voltage. A Buck-Boost converter's basic power stage is shown in Fig. 1. Figure 1 VI is input voltage source, V_O is the output voltage, Sw is switching component, C is the capacitance, L is inductance, D is diode and R is the load resistance.

The converter contains two independence ac inputs, the control $\hat{d}$ (s) and line $\hat{v}_I$ (s) and one output, $\hat{v}_o$ (s).

The converter contains two independence ac inputs, the control $\hat{d}$ (s) and line $\hat{v}_I$ (s) and one output, $\hat{v}_o$ (s). The control-to-output transfer function (G_{vd}) is derived from small signal model of the converter as:

$$G_{vd}(s) = \left(-\frac{V_I - V_o}{D'^2}\right)\frac{\left(1 - s\dfrac{LI}{V_I - V_o}\right)}{1 + s\dfrac{L}{D'^2 R} + s^2\dfrac{LC}{D'^2}} \quad (1)$$

Plug in numerical values illustrated in Table 1 is substitute in (1):

$$G_{vd}(s) = 2.5\frac{(1 - 16.28s)}{.3X10^{-6}s^2 + 68.7X10^{-6}s + 1} \quad (2)$$

CONTROLLER DESIGN FOR BUCK-BOOST CONVERTER

PI controller: In a controller it is necessary to compare the output voltage to a reference value Vref (sometimes called a demand voltage) and then take appropriate remedial action to ensure that Vout = Vref. Usually, this is achieved by generating an error signal e = Vref-Vout which is minimized by the controller (sometimes referred to as a compensator) which then manipulates in such a manner so as to adjust Vout by varying the duty cycle (dc).

Referring to Fig. 2, the output voltage is compared to a reference producing an error signal, (e). The error signal is individually applied to each term of the compensator after which they are combined forming the duty cycle input command to the buck-boost converter. The proportional gain Kp acts as a feed-forward term allowing any changes in the error to be passed to the compensator output without delay. Kp must be carefully chosen because large values tend to induce instabilities in the system response. The integral term Ki is used to reduce the steady-state error at the expense of reducing the dynamic response.

The performance of each prototype controller is evaluated using the ISE (Integral Square Error), IAE (Integral Absolute Error) and ITAE (Integral Time Absolute Error) performance indexes. The ISE index evaluates a controller's performance by assigning it a score based on the error response of the system. Essentially, good performing controllers will have lower ISE scores than poor performing counterparts. To ensure that the phenomenon does not affect the controller's score, the absolute error is squared (which should be a small value in the vicinity of the transient event) which will also reduce its impact.

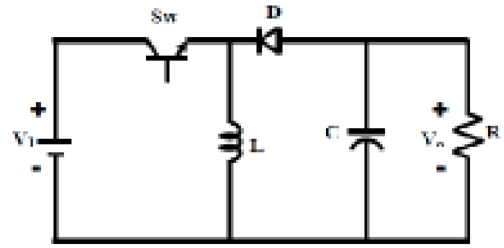

Fig. 1: Buck-boost converter

Table 1: Parameters of buck boost converter

Symbol	Parameter	Values
L	Inductance	220 μH
C	Capacitance	220 μF
R	Load resistance	20 Ω
V_1	Input voltage	12 V
V_o	Output voltage	-24 V

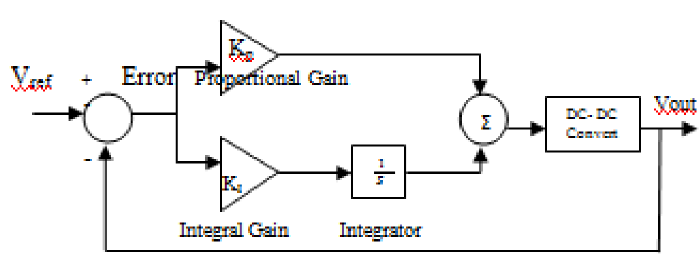

Fig. 2: Block diagram of PI control system

Design of PI controller: The PI controller designed for a buck-boost converter is given in equation. The controllers can be introduced in either feed-back or feed-forward path which will control the steady state error and transient performance. In most of the practical control systems, the input to the controlling device is error.

In case of PI controller, the input to the controlling device is proportional as well as integral of the error function. The order of the system is increased when the system is combined with the controller. The effect of compensation on the system dynamics cannot be visualized easily. The higher the order of the system the more it becomes unstable.

Integral action remains active as long as the error is present. Thus, it makes a steady state error zero so the PI controller is designed based on the frequency domain specification.

The controller transfer function is given in Eq. (3):

$$G_c(s) = K_p + \frac{K_I}{s} \tag{3}$$

The phase margin ϕ_m at ω is determined from the settling time. Phase and Magnitude response equation is given in (4) and (5):

$$\phi_m + \angle G(j\omega)H(j\omega)G_c(j\omega) = \angle 180 \tag{4}$$

ϕ_m is desired phase margin at ω:

$$\left|G(j\omega)H(j\omega)G_c(j\omega)\right| = 1 \tag{5}$$

Solving these two Eq. (4) and (5) we get the K_I and K_P value:

$$G_c(s) = 0.025 + \frac{0.157}{s} \tag{6}$$

Ant Colony Optimization (ACO): Ant colony optimization algorithms are especially suited for finding solutions to difficult optimization problems. A colony of artificial ants helps to find good solutions, by using the emergent property of the ants' cooperative interaction. Ant colony algorithms are adaptive and robust in nature due to their similarities with ant. This property can be applied to different optimization problems as well as different versions of the same problem.

The main traits of artificial ants are derived from their natural model. Such borrowed traits include:

- Cooperative existence in colonies with other ants
- Indirect information transmission by depositing pheromone (stigmergic communication)
- Repetitive local moves in a sequence to find the shortest path to a destination point
- Applying a stochastic decision policy using local information alone to find the best solution. In order to a particular optimization problem, artificial ants are enriched with additional capabilities which are not present in real ants

For a given optimization problem, the best solution is searched by finite sized ant colony. Each ant can find a solution or at least part of the solution to the optimization problem on its own, but the optimal solution can be achieved only when many ants work together. Since the optimal solution can only be achieved through global cooperation of all the ants in a colony, it is a promising result of such cooperation. The ants do not communicate directly while searching for a solution, but they communicate indirectly by adding pheromone to the environment. The ant finds the shortest path for a particular problem from a given starting state by moving through a sequence of neighboring states. It moves based on a non-deterministic local search policy influenced by its own internal state (private information), the pheromone trails and local information encoded in the environment (together public information). Ants use this private and public information to decide when and where to deposit pheromones. The amount of pheromone deposited by an ant is proportional to the quality of the movement made by an ant. It concludes more the pheromone, the better the solution, obtained. Once an ant has found a solution; it dies, that is, it is deleted from the system.

The series controllers are very frequent because of higher order systems. For a continuous system, the transfer function of a PI controller defined in Eq. (3). The design implies the determination of the values of the constants K_p and K_I, meeting the required performance specifications.

The textbook version of the PI controller in continuous time is:

$$u(t) = K_p + K_I \int_0^t e(\tau)d\tau = u_p(t) + u_i(t) \tag{7}$$

where, e (t) = r (t) - y (t) is the difference between the reference signal r (t) and the output, y (t) of the controlled process.

The PI controller is implemented to improve the dynamic response in addition to reducing or eliminating the steady state error. To characterize the performance of the PI controller systems, performance of the transient response such as rise time (t_r), the Integral Square Error (ISE), Overshoot (O_s), settling time (t_s), Integral Absolute Error (IAE), Integral Time Absolute Error (ITAE), Integral Time Square Error (ITSE) are computed. Tuning the parameters of the PI controllers using the multi objective ant colony optimization is indicated in Fig. 3.

As shown in Fig. 4, the gains K_p and K_I of the PI controller are generated by the multi objective ACO algorithm for the buck boost converter. The present

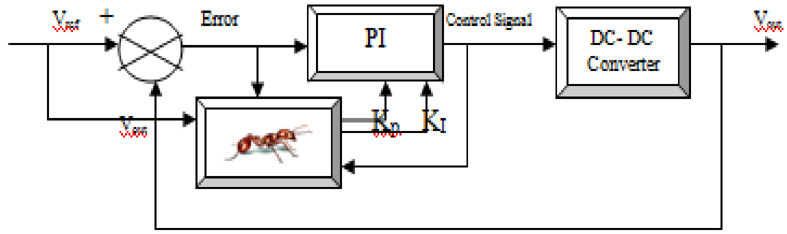

Fig. 3: PI control system

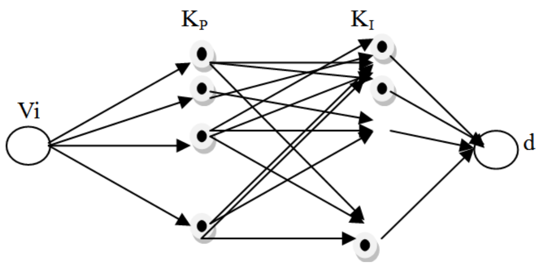

Fig. 4: Ant colony optimization graph

optimization problem is represented directly in the form of a construction graph, for better exploitation of the ACO algorithm.

The population is represented by a 100*2 matrix, where the ant selects the optimum parameters K_p and K_I of the PI control system by minimizing an objective function L^A. The graph (Fig. 4) illustrates the design of PI problem using ant colony algorithm. In this study, each parameter of K_p and K_I is coded by 100 numbers (nodes), respectively. Therefore, only one node represents the optimum solution values of the parameters K_p and K_I.

The basic step in applying optimization method is to choose the optimization criteria that are used to evaluate fitness. Since the PI controller has many performance indexes of the transient response, then they can be combined into one objective function composed of the weighted sum of objectives.
The objective function must be set:

$$L^A = \min(\phi F) \tag{8}$$

where, $F = [f_1\ f_2\ f_3\ f_4\ f_5\ f_6\ f_7]^T$: vector of objective functions, f_1: setting Time (T_s), f_2: Overshoot (O_S), f_3: rise Time (T_r), f_4: Integral Absolute Error (IAE), f_5: Integral Square Error (ISE), f_6: Integral Time Absolute Error (ITAE), $\Phi = [\lambda_1\ \lambda_2\ \lambda_3\ \lambda_4\ \lambda_5\ \lambda_6\ \lambda_7]$: vector of nonnegative weights and f_7: Integral Time Square Error (ITSE).

The purpose of multi-objective optimization problem is to strike a balance between numerous conflicting objectives. Considering all objectives in these problems, we may find more than one solution that optimizes all the objectives and there is no apparent superiority of any of these solutions over others. We can never have a single-best solution which would be better than the remainder. Therefore, a set of solutions which are better than remainder solutions called the Pareto front is faced. Among the feasible solutions, solutions belonging to the Pareto front are called as non-dominated solutions, while the remaining solutions are called as dominated. As none of the Pareto set solutions were found to be better than the any of the non-dominated solutions, all of them are equally acceptable as long as the objectives are met.

ACO uses a pheromone matrix $\tau = \{\tau_{ij}\}$ for the construction of potential good solutions. The initial values of τ are set $\tau_{ij} = \tau_0$ for all (i, j), where $\tau_0 > 0$.

The probability P_{ij}^A (t) of choosing a node j at node i is defined in (9). At each evolution of the algorithm, the ant constructs a complete solution using (9), starting at source node:

$$P_{ij}^A(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\Sigma_{i,j \in T^A} [\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}, if\ i, j \in T^A \tag{9}$$

where, η_{ij} representing heuristic functions, constant α-determine the relative influence of pheromone values where constant-β determine the relative influence of the heuristic values and at a given time, T^A: is the path effectuated by the ant A.

The pheromone evaporation is a way to elude unlimited increase of pheromone trails and it allows the forgetfulness of the poor decisions:

$$\tau_{ij}(t) = \rho\tau_{ij}(t-1) + \sum_{A=1}^{NA} \Delta\tau_{ij}^A(t) \tag{10}$$

Where the quantity of pheromone on each path, NA represents number of ants, ρ indicates the evaporation rate. Evaporation rate lies between zero and one ($0<\rho\leq 1$).

The following general algorithm can describe the proposed algorithm.

Begin:

Step 1: Initialize randomly potential solutions of the parameters (K_i, K_p) by using uniform distribution.
Initialize the heuristic value and the pheromone trail.
Initialize the Pareto set to an empty set.

Step 2: Place the A^{th} ant on the node.

Compute the heuristic value associated in the multi objective L^A.
Choose the successive node with probability:

$$P_{ij}^{A}(t) = \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]\beta}{\Sigma_{i,j\in T^{A}}[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}, \qquad ifi, j \in T^{A} \tag{11}$$

where, η_{ij} = 1/K_j, j = [P, I]: representing heuristic functions, at a given time T^A: represents the path effectuated by the ant A. The quantity of pheromone on each path may be defined as:

$$\Delta\tau_{ij}^{A} = \begin{cases} \dfrac{L^{\min}}{L^{A}}, & ifi, j \in T^{A} \\ 0, & else \end{cases} \tag{12}$$

where, L^A is the value of the objective function found by the ant A. Till the current iteration, L min is the best optimal solution brought out by the set of the ants.

Step 3: Use pheromone evaporation given by (10) to avoid an infinite progression of pheromone trails and allow the forgetfulness of bad choices:

$$\tau_{ij}(t) = \rho\tau_{ij}(t-1) + \Sigma\sum_{A=1}^{NA}\Delta\tau_{ij}^{A}(t) \tag{13}$$

where,
NA : Number of ants
ρ : The evaporation rate 0<ρ≤1

Step 4: Evaluate the obtained solutions according to the different objectives.
Update the Pareto archive with the non-dominated ones.
Reduce the size of the archive if necessary.

Step 5: Display the optimum values of the optimization parameters.

Step 6: Globally update the pheromone, according to the optimum solutions calculated at Step 5. Iterate from Step 2 until we reach the maximum number of iterations.

End

RESULTS AND DISCUSSION

Simulation and experimental results: In this section, the numerical results obtained using the proposed algorithm is presented and discussed. The various of the parameters in ACO are, m = 100 (numbers of ants), α = 0.5, β = 0.5, ρ = 0.5 and maximum generation = 50. The objective function is given in Eq. (8):

$$L^{A} = \min(\lambda_2 f_2 + \lambda_3 f_3 + \lambda_4 f_4 + \lambda_5 f_5 + \lambda_6 f_6) \tag{14}$$

The objective function here is f_2: the overshoot to measure the performance of the closed-loop system, f_4: Integral Time Absolute Error and f_5: the integral square error f_6: Integral Absolute error that should be minimized. Therefore the vector of weights is Φ = (0 1 0 1 1 1 0). The closed loop PI controller cascaded with the converter was tuned for the values K_P and K_i first by using multi objective ant colony algorithm. Hence, the percent maximum overshoots, the settling time, the rise time and the integral of the squared error were computed.

The graphs of the obtained three-dimensional Pareto optimal fronts for the generated problem corresponding to the buck-boost converter shown in Fig. 5.

Figure 6 Report the evaluation of the objective function of the converter. It is observed that the objective function value decreases substantially.

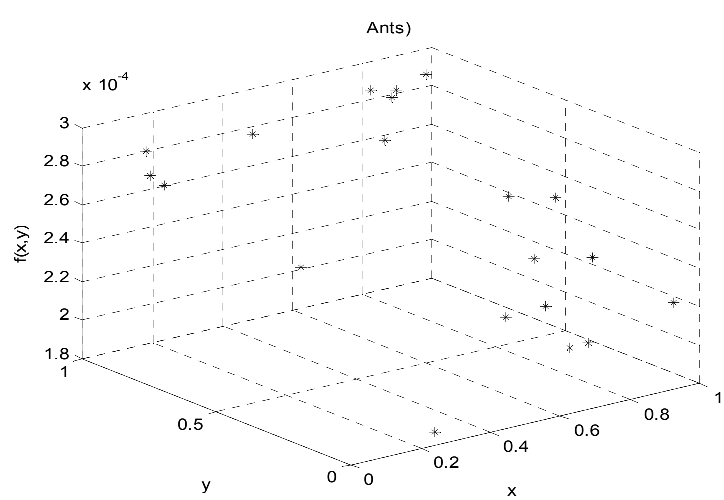

Fig. 5: Multi-objective optimization of Pareto set of the converter

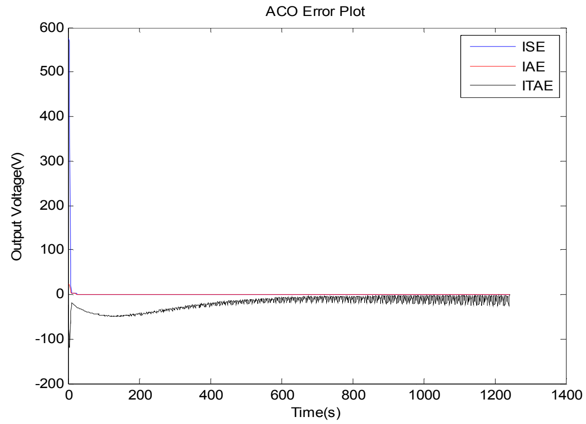

Fig. 6: Evaluation of the objective function

Initially, each parameter (K_p, K_i) is distributed randomly and uniformly with an average value. After several iterations, the multi objective ant colony algorithm generated the best solutions of the PI parameters (K^{best}_p, K^{best}_i).

After that, each parameter (K^{best}_p, K^{best}_i) is distributed randomly and uniformly with an average value which is equal to the value founded in the last generation. Finally, the multi objective ant colony algorithm generated the optimal solutions (K^{opt}_p, K^{opt}_i).

The prototype of Buck-Boost converter tested in the laboratory is shown in Fig. 7.

Figure 8 shows the experimental result of duty cycle varying the line voltage of the converter. Varying line voltage and load it will adjust the duty cycle and make the voltage as stable.

Figure 9 shows the transient response of the buck-boost converter during the staring up. During the starting up converter settling time is 2.8 µsec and it have steady state error.

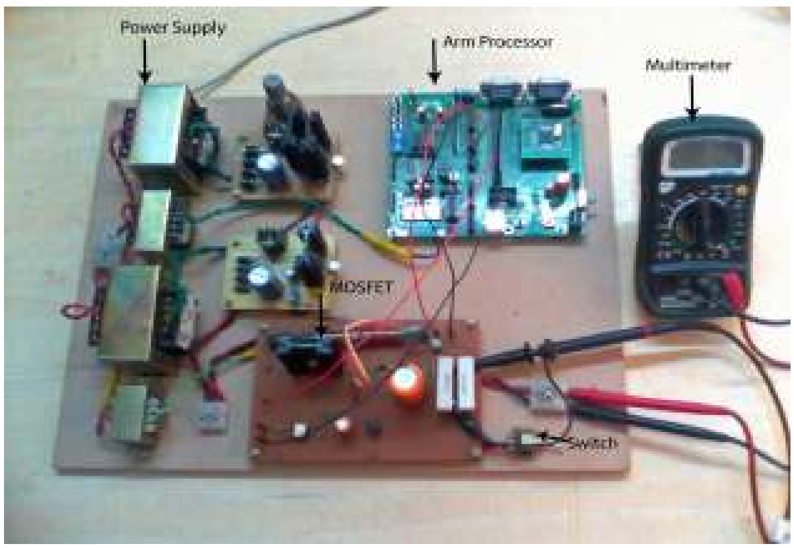

Fig. 7: Prototype of buck-boost converter

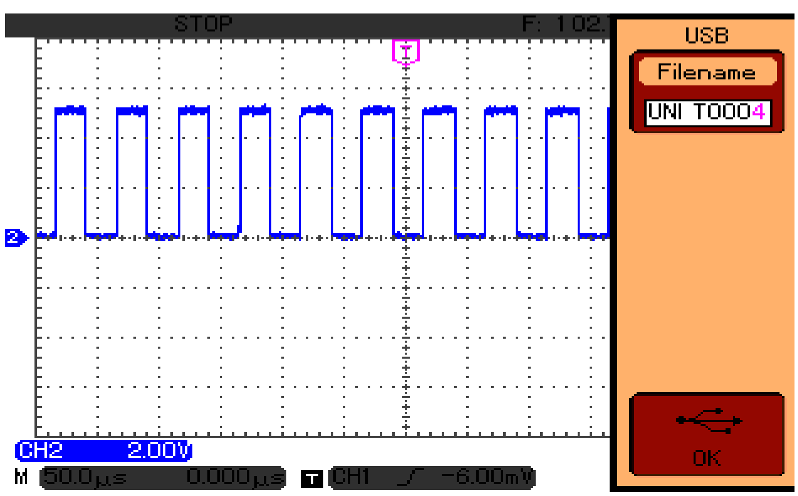

Fig. 8: Duty cycle of PWM

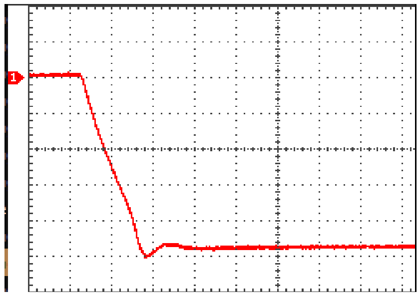

Fig. 9: The output voltage transient response of the converter during starting-up (5 V/div and 1 μsec/div)

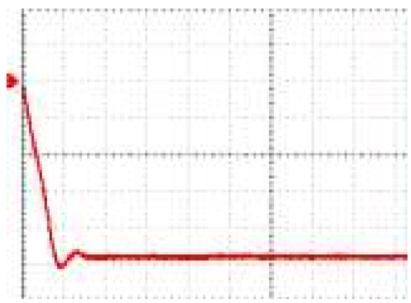

Fig. 10: The output voltage transient response of the converter during starting-up (5 V/div and 1 μsec/div)

Figure 10 shows the transient response of the buck-boost converter during the staring up. During the starting up converter settling time is 1.6 μsec and it have very less amount of steady state error.

CONCLUSION

Conventional PI controller and optimized PI controllers were designed and implemented for a Buck-Boost converter. The linear PI controller was designed for the converter using frequency response techniques. The conventional PI controller was applied during steady-state to achieve stable steady-state response. Optimized PI controller was designed based on ACO. The performance of the controllers was compared based on the experimental results. Experimental results show that fast transient response and stable steady-state response could be achieved for the buck-boost converters using optimized PI converter.

REFERENCES

Alvarez-Ramirez, J., I. Cervantes, G. Espinosa-Perez, P. Maya and A. Morales, 2001. A stable design of PI control for DC-DC converters with an RHS zero. IEEE T. Circuits-I, 48(1): 103-106.

Arulselvi, S., G. Uma and M. Chidambaram, 2004. Design of PID controller for boost converter with RHS zero. Proceeding of the 4th International Power Electronics and Motion Control Conference (IPEMC, 2004), 2: 532-537.

Cao, C., X. Guo and Y. Liu, 2007. Research on ant colony neural network PID controller and application. Proceeding of the 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD ’07), pp: 253-258.

Costa, D. and A. Hertz, 1997. Ants can colour graphs. J. Oper. Res. Soc., 48(3): 295-305.

Dorigo, M. and G.D. Caro, 1999. The Ant Colony Optimization Meta-heuristic. New Ideas in Optimization, McGraw Hill, London, UK.

Dorigo, M., G.D. Caro and L.M. Gambardella, 1999. Ant algorithms for discrete optimization. Artif. Life, 5(2): 137-172.

Erickson, R.W., 1991. Fundamentals of Power Electronics. 5th Edn., Kluwer Academic Pub., USA.

Gambardella, L.M., E.D. Taillard and G. Agazzi, 1999. MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows. In: Corne, G.D., M. Dorigo and F. Glover (Eds.), New Ideas in Optimization. McGraw Hill, London, UK, pp: 63-76.

Guo, L., J.Y. Hung and R.M. Nelms, 2002. PID controller modifications to improve steady-state performance of digital controllers for buck and boost converters. Proceeding of 17th Annual IEEE Applied Power Electronics Conference and Exposition (APEC, 2002), 1: 381-388.

Hsiao, Y.T., C.L. Chuang and C.C. Chien, 2004. Ant colony optimization for designing of PID controllers. Proceeding of the IEEE International Symposium on Computer Aided Control Systems Design. Taipei, Taiwan, pp: 321-326.

Huang, S.J., 2001. Enhancement of hydroelectric generation scheduling using ant colony system based optimization approaches. IEEE T. Energy Conver., 16(3): 296-301.

Kim, D.H., 2001. Tuning of a PID controller using an artificial immune network model and local fuzzy set. Proceeding of the Joint 9th IFSA World Congress and 20th NAFIPS International Conference (NAFIPS ’01), 5: 2698-2703.

Mitsukura, Y., T. Yamamoto and M. Kaneda, 1999. A design of self-tuning PID controllers using a genetic algorithm. Proceedings of the American Control Conference, pp: 1361-1365.

Oonsivilai, A. and P. Pao-La-Or, 2008. Application of adaptive tabu search for optimum PID controller tuning AVR system. WSEAS T. Power Syst., 3(6): 495-506.

Reinelt, G., 1994. The Traveling Salesman: Computational Solutions for TSP Applications. Lecture Notes in Computer Science, Springer, Berlin, Germany, Vol. 840.

Selvan, S.E., S. Subramanian and S.T. Solomon, 2003. Novel technique for PID tuning by particle swarm optimization. Proceeding of the 7th Annual Swarm Users/Researchers Conference (SwarmFest '03).

Seng, T.L., M.B. Khalid and R. Yusof, 1999. Tuning of a neurofuzzy controller by genetic algorithm. IEEE T. Syst. Man Cy. B, 29(2): 226-236.

Siotine, E. and W. Li, 1991. Applied Nonlinear Control. Prentice-Hall, Inc., Englewood Cliff, NJ.

Stützle, T. and M. Dorigo, 1999. ACO Algorithms for the Quadratic Assignment Problem. In: Corne, D., M. Dorigo and F. Glover (Eds.), New Ideas in Optimization. McGraw Hill, London, UK, pp: 33-50.

Tsang, K.M., W.L. Chan and X.L. Wei, 2008. Robust DC/DC buck converter using conditional integrator compensator. Electron. Lett., 44(2): 152-153.

Tzafestas, S. and N.P. Papanikolopoulos, 1990. Incremental fuzzy expert PID control. IEEE T. Ind. Electron., 37(5): 365-371.

Visioli, A., 2001. Tuning of PID controllers with fuzzy logic. IEE P-Contr. Theor. Ap., 148(1): 1-8.

Wang, P. and D.P. Kwok, 1994. Optimal design of PID process controllers based on genetic algorithms. Control Eng. Pract., 2(4): 641-648.

Zhou, G. and J.D. Birdwell, 1994. Fuzzy logic-based PID autotuner design using simulated annealing. Proceeding of the IEEE/IFAC Joint Symposium on Computer-Aided Control System Design, pp: 67-72.