Particle Swarm Optimization and Genetic Algorithm to Optimizing the Pole Placement Controller on Cuk Converter

M. R. Yousefi, M. Bayati Poudeh

Electrical Engineering Department, Islamic Azad University, Najafabad Branch, Iran

***Abstract*— In this paper a novel method to the design of pole placement controller for Cuk converters is presented This optimized method can control the voltage of DC-DC converter. In this method, average model of converter is employed and it is possible to approximate the system by a linear system and then linear control methods can be used. Pole Placement Control as one of these methods is designed by a systematic methodology based on Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). The proposed controllers were simultaneously designed and they have provided a coordinated control action and a satisfactory performance for the Cuk converter, as shown in the results.**

***Keywords*— Cuk Converter; Pole Placement; Genetic Algorithm; Particle Swarm Optimization.**

I. INTRODUCTION

With the sophistication of complicated electronic systems, the demand for dc-to-dc converters will become more and more severe. Buck, Boost, Buck- Boost and Cuk converters are among the most popular dc-to-dc converters. The Cuk converter avoids these problems. In fact, input /output current ripples can be made arbitrary small [1]. Most often DC-to-DC converters are used as interfaces between DC systems of different voltage levels. Examples for their applications are power supplies in computers and other electronic equipment as well as DC motor drives. The input to these converters is often an unregulated rectified line voltage, which will fluctuate due to changes of the line voltage magnitude [2], [3].

The DC–to–DC Cuk converter is one of the most studied power converters. Static and dynamic characteristics of Cuk converters have been widely discussed in the literature [4], [5], where powerful tools for analysis, modeling and design are available. Cuk converters feature excellent properties (capacitive energy transfer, magnetic components integrability, full transformer utilization) and good steady-state performances (wide conversion ratio, smooth input and output currents) [6]. Investigations of basic switching converters commenced in, the 1960. Over the next two decades, Middlebrook and Cuk made significant contributions on modeling, analysis, and general ization of the basic topologies [7], [8], developments on the Cuk converter [9]. With the maturity in modeling and analysis, the control of switching regulators began to motivate many studies. Contributions were made by Chetty and Thau on the design of analog controllers to improve the transient response of some particular converters [10].

DC–DC Switching converters are a traditional benchmark for testing nonlinear controllers, due to their inherent nonlinear characteristics [11]. After the pioneering studies of, a great deal of research has been directed at developing techniques for averaged modeling of different classes of switching converters, and for an automatic generation of the averaged models [12], [17]. Nevertheless, the problem of output voltage regulation for the DC–to–DC, Cuk converter has been addressed and solved using various techniques. In particular, classical linear design tools have been used in [13], whereas the applicability of advanced nonlinear methods, such as feedback linearization [14], sliding mode [15] and H_∞ design [16], has also been investigated.

In this paper, Section II describes the model of the Cuk converter. The Pole Placement control method is detailed in Section III, Genetic Algorithm is shown in Section IV, another optimization method, Particle Swarm Optimization, is described in Section V. The computer simulation results are presented and discussed in Section VI, and Finally, Section VII concludes this paper.

II. CÛK CONVERTER

A. Template Cûk Converter Circuit Model

The Cuk converter of Fig. 1 with switching period of T and duty cycle of d is considered. It includes: DC supply voltage V_{in}, input inductor L_1, power switch Q, transfer capacitor C_1, free-wheeling diode D, output inductor L_2 and filter capacitor C_2. R_1, and R_2 are parasitic resistances and R_L, is load resistance. In the following we will assume continuous conduction operation (total inductor current i_1+i_2 and voltage V_{C1} always positive), which improves components utilization and reduces harmonic and RF pollution versus load and supply. In the steady state, the dc voltage across capacitor C_1, equals the sum of input voltage V_{in} and output voltage V_{out}, the average inductor voltages being zero [12].

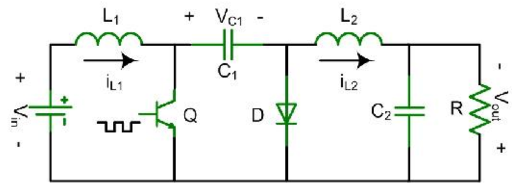

Figure 1. Circuit scheme of the Cûk Converter

B. Mathematical Model of the Cûk Converter

When the switch is closed (Fig. 2) current i, raises quite linearly, diode D is counter polarized and capacitor C_1, supplies energy to the output stage. Also current i_2, increases, while voltage V_{C1}, decreases. During the continuous conduction mode of operation, the state space equations are as follows:

$$
\begin{aligned}
\frac{di_{L1}}{dt} &= \frac{1}{L_1}(V_{in}) \\
\frac{dV_c}{dt} &= \frac{1}{C_2}(-i_{L2}) \\
\frac{di_{L2}}{dt} &= \frac{1}{L_2}(V_c - V_{out}) \quad For \quad 0 < t < dT \\
\frac{dV_{out}}{dt} &= \frac{1}{C_1}(i_{L2} - \frac{V_{out}}{R})
\end{aligned} \tag{1}
$$

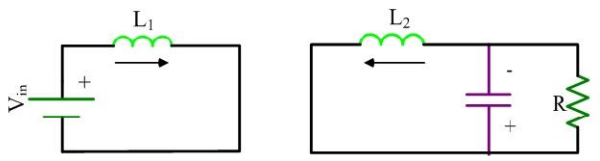

Figure 2. Sub-circuits for switch ON

When the switch is open (Fig. 3) both inductor currents flow through freewheeling diode D and decrease, while capacitor C_1 is recharged by current i_1 when the switch is OFF the state space equations are represented by:

$$
\begin{aligned}
\frac{di_{L1}}{dt} &= \frac{1}{L_1}(V_{in} - V_{out}) \\
\frac{dV_c}{dt} &= \frac{1}{C_2}(-i_{L1}) \\
\frac{di_{L2}}{dt} &= \frac{1}{L_2}(-V_{out}) \quad For \quad dT < t < T \\
\frac{dV_{out}}{dt} &= \frac{1}{C_1}(i_{L2} - \frac{V_{out}}{R})
\end{aligned} \tag{2}
$$

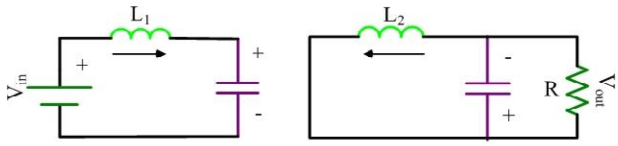

Figure 3. Sub-circuits for switch OFF

The model by state space averaging method is as following:

$$
\begin{aligned}
\frac{dx_1}{dt} &= -\frac{1-d}{L_1}x_2 + \frac{1}{L_1}v_{in} \\
\frac{dx_2}{dt} &= \frac{1-d}{C_1}x_1 - \frac{d}{C_1}x_3 \\
\frac{dx_3}{dt} &= \frac{d}{L_2}x_2 - \frac{1}{L_2}x_4 \\
\frac{dx_4}{dt} &= \frac{1}{C_2}x_3 - \frac{1}{RC_2}x_4
\end{aligned} \tag{3}
$$

Where x_1, x_2, x_3, and x_4, are the moving averages of i_{L1}, V_{C1}, i_{L2} and V_{out} respectively, and d represents the duty cycle.

III. POLE PLACEMENT DESIGN METHOD

Pole placement is a method that seeks to place the poles of the closed–loop system at some predetermined locations. Although this method has some drawbacks in handling complex systems, it is still fairly sufficient for most small control systems and it gives the best introduction to the design of complex systems [3],[12]. In designing the pole placement controller, the new poles must satisfy a few conditions stated below,

- The ripple in V_{out} must be reduced
- The overshoot of V_{out} must be improved,

Where it is needed one of the poles locations to be at the origin to fulfill the requirements stated above. The controller block diagram has been designed using SIMULINK-MATLAB application after the new poles locations were found based on Fig. 4.

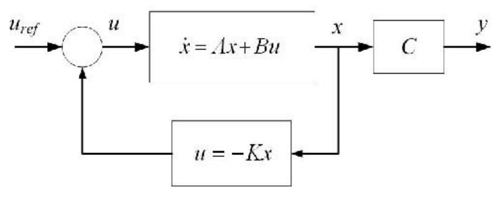

Figure 4. Pole Placement Controller

The basic concept behind the method is to get K, which will satisfy the closed–loop transfer function at desired pole locations s_i , $i = 1, 2, \dots n$. Implementation of the method will be described here, through the following illustrative example in which a regulator is assumed, i.e., no reference input. (The reference input will be added after some discussion on the state estimators) [18]. Suppose the system as (4).

$$\underline{\dot{x}}(t) = Ax(t) + Bu(t) \tag{4}$$

System is to be controlled by full state feedback such that (5).

$$u(t) = -Kx(t) \tag{5}$$

Where the closed–loop poles are placed at locations $p_1, p_2, \dots p_n$. This means that the required closed–loop transfer function of the controlled system is given by (6).

$$\psi(s) = (s - p_1)(s - p_2)(s - p_3)\dots(s - p_n) = 0 \tag{6}$$

Which can be expanded as (7).

$$\psi(s) = S^n + q_n S^{n-1} + q_{n-1} S^{n-2} + \dots + q_3 S^2 + q_2 S + q_1 = 0 \tag{7}$$

Let the matrix A and input matrix B respectively, as (8).

$$B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \qquad A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \tag{8}$$

Therefore, if the feedback matrix K is as (9).

$$K = \begin{bmatrix} k_1 & k_2 & \cdots & k_n \end{bmatrix} \tag{9}$$

Then the closed-loop system has (10), system matrix.

$$A - BK = \begin{bmatrix} a_{11} - b_1k_1 & a_{12} - b_1k_2 & \cdots & a_{1n} - b_1k_n \\ a_{21} - b_2k_1 & a_{22} - b_2k_2 & \cdots & a_{2n} - b_2k_n \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} - b_nk_1 & a_{n2} - b_nk_2 & \cdots & a_{nn} - b_nk_n \end{bmatrix} \tag{10}$$

Whose characteristic function is shown in (11).

$$\psi(s) = |sI - A + BK| = \begin{bmatrix} s - a_{11} + b_1k_1 & -a_{12} + b_1k_2 & \cdots & -a_{1n} + b_1k_n \\ -a_{21} + b_2k_1 & s - a_{22} + b_2k_2 & \cdots & -a_{2n} + b_2k_n \\ \vdots & \vdots & \vdots & \vdots \\ -a_{n1} + b_nk_1 & -a_{n2} + b_nk_2 & \cdots & s - a_{nn} + b_nk_n \end{bmatrix} = 0 \tag{11}$$

Comparison of this characteristic equation and demanded in (7) can lead to the determination of the values of k_i and hence matrix K. However, as it can be seen, the algebra behind such a problem is very cumbersome and might in some cases be insoluble. On the other hand, however, if system (A,B) is controllable, the closed-loop system can be expressed in its controllable canonical form as (12).

$$A^* - B^*K^* = \begin{bmatrix} a^*_{11} - b^*_1k^*_1 & a^*_{12} - b^*_1k^*_2 & \cdots & a^*_{1n} - b^*_1k^*_n \\ a^*_{21} - b^*_2k^*_1 & a^*_{22} - b^*_2k^*_2 & \cdots & a^*_{2n} - b^*_2k^*_n \\ \vdots & \vdots & \vdots & \vdots \\ a^*_{n1} - b^*_nk^*_1 & a^*_{n2} - b^*_nk^*_2 & \cdots & a^*_{nn} - b^*_nk^*_n \end{bmatrix} \tag{12}$$

In this case, the closed-loop transfer function becomes (13).

$$|sI - A^* + B^*K^*| = \begin{bmatrix} s & -1 & \cdots & 0 \\ 0 & s & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ -a_1 - k^*_1 & -a_2 - k^*_2 & \cdots & s - a_n - k^*_n \end{bmatrix} \tag{13}$$

Whose expansion can easily be determined to be (14).

$$\psi(s) = s^n + (a_n + k^*_n)s^{n-1} + (a_{n-1} + k^*_{n-1})s^{n-2} + \dots + (a_2 + k^*_2)s + (a_1 + k^*_1) \tag{14}$$

Comparison of this equation with the demanded one in (7) shows the (15).

$$a_i + k^*_i = q_i \quad , \quad i = 1, 2, \dots, n \tag{15}$$

From which the elements of the feedback matrix can be computed as (16).

$$, \quad i = 1, 2, \dots, n \qquad k^*_i = q_i - a_i \tag{16}$$

Or in vector form as (17).

$$K^* = q - a \tag{17}$$

Where

$$q = \begin{bmatrix} q_1 & q_2 & \cdots & q_n \end{bmatrix} \tag{18}$$

$$a = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix} \tag{19}$$

In the emphasized again that this procedure applies only for SISO systems in controllable canonical form, and that order of the elements in vectors a, q and K^* are as shown above. Improper order of the elements will give wrong results. This matrix K^* is the feedback gain for the system in controllable canonical form, the effort is such that is express in (20).

$$u(t) = -K^* x^*(t) \tag{20}$$

Where

$$x^*(t) = P^{-1}x(t) \tag{21}$$

Therefore, for the original system (not in control canonical form) this control effort becomes as (22).

$$u(t) = -K^* P^{-1} x(t) \tag{22}$$

So that the corresponding feedback gain matrix K is;

$$K = K^* P^{-1} \tag{23}$$

IV. Genetic Algorithm

Genetic algorithm, a core member of soft computing or computational intelligence, is a method for discrete optimization, motivated by the manner in which nature optimizes biological systems [22]. A genetic algorithm (GA) (Goldberg, 1989) is an iterative search procedure which operates on a set of strings ('chromosomes'). The strings encode possible solutions to the problem which the GA is being used to solve, and the set is manipulated and amended repeatedly in the following way: each string is evaluated with regard to its 'fitness' (its suitability as a solution). The strings are then 'mated', with new strings being created from elements of the strings which mate, and with the fitter strings having a greater likelihood of taking part in the reproductive process. Thus, the fitter a string is, the more likely it is to pass on its characteristics via partial inclusion in new strings to the next 'generation' of strings. With the new generation, the process is repeated. If all goes well, natural selection of the fittest should ensure that overall, the fitness of the 'population' of strings will rise as will the fitness of the fittest individual solution found. In this way, genetic algorithms can be used in solving optimization problems [18].

1. **Selection:** Selects the fittest individuals in the current population to he used in generating the next population.
2. **Crossover:** Causes pairs, or larger groups of individuals to exchange genetic information with one another.
3. **Mutation:** Causes individual genetic representations to be changed according to some probabilistic rule.

Genetic algorithms are more likely to converge to global optima than conventional optimization techniques, since they search from a population of points, and are based on probabilistic transition rules. Conventional optimization techniques are ordinarily based on deterministic hill-climbing methods, which may find local optima. Genetic algorithms can also tolerate discontinuities and noisy function evaluations.

V. PARTICLE SWARM OPTIMIZATION (PSO)

Kennedy and Eberhart first introduced particle swarm optimization (PSO) in 1995 as a new heuristic method [19]. The original objective of their research was to mathematically simulate the social behavior of bird flocks and fish schools. As their research progressed, they discovered that with some modifications, their social behavior model can also serve as a powerful optimizer. The first version of PSO was intended to handle only nonlinear continuous optimization problems. However, many advances in PSO development elevated its capabilities to handle a wide class of complex engineering and science optimization problems. Summaries of recent advances in these areas are presented in [20] and will not be addressed in this paper due to space limitations.

PSO method is a member of wide category of swarm intelligence methods for solving the optimization problems. It is a population-based search algorithm where each individual is referred to as particle and represents a candidate solution. Each particle in PSO flies through the search space with an adaptable velocity that is dynamically modified according to its own flying experience and also to the flying experience of the other particles. In PSO, each particle strives to improve itself by imitating traits from their successful peers. Further, each particle has a memory and hence it is capable of remembering the best position in the search space ever visited by it. The position corresponding to the best fitness is known as *pbest* and the overall best out of all the particles in the population is called *gbest*. The modified velocity and position of each particle can be calculated using the current velocity and the distance from the $pbest_{L,K}$ to $gbest_K$ as shown in the following formulas [21]:

$$v^{t+1}_{L,K} = Ev^t_{L,K} + W_1 P_1(\ldots)(pbest_{L,K} - x^t_{l,k}) + W_2 P_2(\ldots.)(gbest_K - x^t_{L,K}) \quad (24)$$

$$x^{t+1}_{L,K} = x^t_{L,K} + v^{t+1}_{L,K} \quad (25)$$

With $K=1; 2; \ldots ; n$ and $L=1; 2; \ldots ;m$, where n is the number of particles in a group; m the number of members in a particle; t the number of iterations (generations); $v^t_{L,K}$ the velocity of particle j at iteration $V^{\min}_K \langle v^t_{L,K} \langle v^{\max}_K$, E the inertia weight factor; W_1 and W_2 are the cognitive and social acceleration factors, respectively; P_1 and P_2 are the random numbers uniformly distributed in the range (0,1); $x^t_{L,K}$ is the current position of particle K at iteration t; $pbest_K$ the *pbest* of particle *K*; *gbest* the *gbest* of the group. The *K*-th particle in the swarm is represented by a g-dimensional vector $x_L=(x_{L,1},x_{L,2},x_{L,3,\ldots\ldots}, x_{L,K})$ and its rate of position change (velocity) is denoted by another g-dimensional vector $v_L=(v_{L,1},v_{L,2},v_{L,3,\ldots\ldots}, v_{L,K})$.The best previous position of the *K*-th particle vector $pbest_L=(pbest_{L,1}, pbest_{L,2,} pbest_{L,3,\ldots\ldots}, pbest_{L,K})$. The index of best particle among all the particles in the group is represented by $gbest_K$. In PSO, each particle moves in the search space with a velocity according to its own previous best solution and its group's previous best solution. The velocity update in a PSO consists of three parts: momentum, cognitive and social parts. The balance among these parts determines the performance of a PSO algorithm. The parameters W_1 and W_2 determine the relative pull of *pbest* and *gbest* and the parameters P_1 and P_2 help in stochastically varying these pulls. In the above equations, superscripts denote the iteration number.

In this work, following parameters are chosen.
Number of particles=100, C_1 and C_2=0.9, w_{max}=0.8, w_{min}=0.15, v_{max}=15% of range of parameter Max., Number of functional evaluation=700.

VI. RESULT OF SIMULATION

With regard to the state equations for the converter and taking into consideration in V_{in}=24v, L_1=3mH, L_2=1.9mH, C_1=47μF, C_2=100μF, R=15Ω, F=20KHz, open loop response will be in the form of Fig. 5. To optimize the V_{out} the pole placement control method with GA and PSO is used.

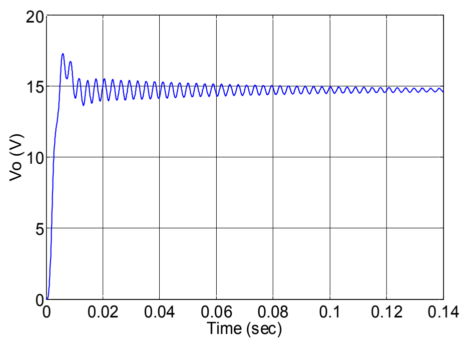

Figure 5. Open Loop response of Cuk converter

As it is evident from the study of outputs, the open loop system cannot hold the output in constant and ideal conditions. Of course, the outputs will eventually approximate the ideal rate with regard to the stability of the system.

A. The Result of Simulation with Pole Placement Control Method

By applying pole placement controlling on the system and with regard to Table I, the following results are obtained (Fig. 6).

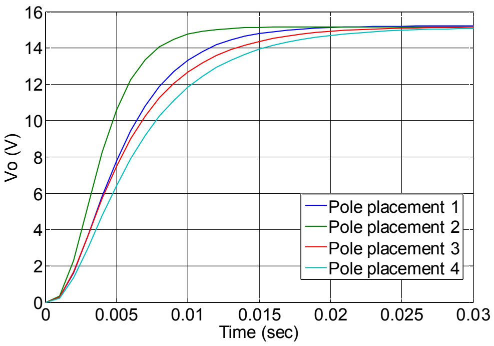

Figure 6. Response Cuk converter response with Pole Placement controller

B. The Application of Ggenetic Algorithm for Adjusting Pole Placement Controlling Coefficients

For this purpose the genetic algorithm on the system, within several successive seasons, has improved the convergence of the system and has brought the best chromosome of that generation into optimum amount from generation to generation which its result is shown in Fig. 7.

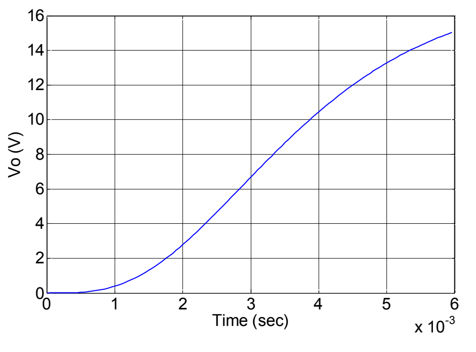

Figure 7. Cuk converter response with pole placement -GA controller

C. The Application of Particle Swarm Ooptimization for Adjusting Pole Placement Controlling Ccoefficients

By applying Particle swarm optimization for the optimization of pole placement controller response. The result in Fig. 8, is obtained.

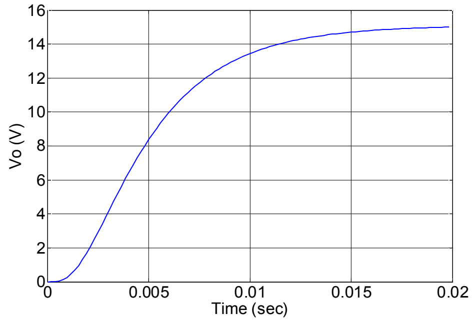

Figure 8. Cuk converter response with Pole Placement-PSO controller

VII. Conclusion

In usage of power converters, it is important that the voltage ripple of converter is reduced and a completely DC voltage output is produced. Pole placement controller as a linear controller is able to control the dynamic behavior of Cuk converter. The use of genetic algorithm and Particle Swarm Optimization for the calculation of optimum coefficients of the gains in the design of this controller can bring about optimum dynamic response. Also, the results of Pole Placement controller method are more satisfactory with Particle Swarm Optimization than the results of genetic algorithm method.

TABLE I

Pole Placement parameters

Pole Placement 1	p=[-991 -900 -1093 -1248]
Pole Placement 2	p=[-2042 -523 -852 -1529]
Pole Placement 3	p=[-2678 -922 -2482 -227]
Pole Placement4	p=[-2083 -4875 -202 -680]
Pole Placement -GA	p=[-1018 -1114 -1103 -976]
Pole Placement -PSO	p=[-3132 -554 -306 -2612]

References

[1] P. Medagam, F. Pourboghrat, “Nonlinear Robust Control of Cuk Converter Using Radial Basis Functions”, *38th North American Power Symposium, NAPS 2006*, pp.229 – 234, 2006.

[2] A. H. Zaeri, M. B. Poodeh, S. Eshtehardiha, “Improvement of Cûk Converter Performance with Optimum LQR Controller Based on Genetic Algorithm”, *IEEE. 2007., International Conference on Intelligent & Advanced Systems*, 2007.

[3] M. B. Poudeh, S. Eshtehardiha, M. H. Ershadi, “Optimizing the Classic Controllers to Improve the Cuk Converter Performance Based on Genetic Algorithm”, *International Conference on Smart Manufacturing Application, ICSMA 2008,* pp. 329-334, 2008.

[4] R. D. Middlebrook, S. Cuk: Advances in Switched Mode Power Conversion. Teslaco Publishing, Pasadena, 1981.

[5] A. Pietkiewicz, D. Tollik: "Modelling and analysis of the current-mode controlled Cuk converter with coupled inductors", *PESC 1988*, pp. 398-408, 1988.

[6] L. Malesani, R. G. Spiazzi, P. Tenti, "Performance Optimization of Cuk Converters by Sliding-Mode Control", *IEEE Trans. on Power Electronics*, Vol. 10, No. 3, pp. 302-309, 1995.

[7] R. D. Middlebrook and S. Cuk, "A General Unified Approach to Modelling Switching-Converter Power Stages," *PESC'76*, pp. 18-34, 1976.

[8] S. Cuk, R. D. Middlebrook, "A General Unified Approach to Modelling Switching-Converter in Discontinuous Conduction Mode", *PESC '77 Rec.*, 1977, pp. 36-57.

[9] "Advances in Switched-Mode Power Conversion," *TESLA Co,* vol. XI, pp. 311-330, 1984.

[10] F. H. F. Leung, P. K. S. Tam, C. K. Li, "The Control of Switching DC-DC Converters a General LQR Problem", *IEEE Trans. on Industrial Electronic*s, Vol. 38, No. 1, pp. 65-71. 1991.

[11] J. Sun, H. Grotstollen, "Symbolic Analysis Methods for Averaged Modeling of Switching Power Converters," *IEEE Trans. Power Electron.*, Vol. 12, pp. 537–546, 1997.

[12] S. Eshtehardiha, M. B. Poudeh, "Application of the Genetic Algorithm to the Reduction of Ripple Voltage Output of Cûk Converter", *International Conference on Electrical Engineering Design & Technologies, 2007.*

[13] G. J. Kassakian, M. Schelecht, and C. G. Verghese, Principles of Power Electronics, *Addison–Wesley*, *Reading, MA*, 1991.

[14] G. C. Sanders, G. Verghese, D. F. Cameron, "Nonlinear Control Laws for Switching Power Converters", *25th IEEE Conf. on Decision and Control*, 1986.

[15] H. Sira-Ramirez, M. Ilic–Spong, "Sliding Motions on Bilinear Switched Networks", *IEEE Trans. on Circuits and Systems*, CAS–34(8): 919:933, 1987.

[16] H. Rodriguez, R. Ortega, A. Astolfi, "Adaptive Partial State Feedback Control of the DC-to-DC Cuk Converter", *Proceedings of the 2005 American Control Conference*, Vol. 7, pp. 5121-5126, 10 June 2005.

[17] J. Mahdavi, A. Emadi, H. A. Toliyat, "Application of State Space Averaging Method to Sliding Mode Control of PWM DC/DC Converters", *IEEE Industry Applications Conference*, Vol. 2, pp. 820-827, 1997.

[18] J. Kennedy, R. Eberhart, "Particle Swarm Optimization," *Proc. IEEE Int. Conf. Neural Netw.*, 1995, vol. 4, pp. 1942–1948.

[19] R. C. Eberhart, Y. Shi, "Guest editorial," *IEEE Trans. Evol. Comput. (Special Issue on Particle Swarm Optimization)*, vol. 8, no. 3, pp. 201–203, Jun. 2004.

[20] S. Panda, N. P. Padhy, "Optimal Location and Controller Design of STATCOM for Power System Stability Improvement Using PSO", *Journal of the Franklin Institute*, Vol. 345, pp. 166-181, 2008.

[21] D. B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. Piscataway, NJ: IEEE Press, 2000.

[22] T. Routen, "Genetic Algorithm and Neural Network Approaches to Local Access Network Design", *Proceedings of the Second International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, *MASCOTS '94*, pp.239-243, 1994.