Uludağ University Journal of The Faculty of Engineering, Vol. 24, No. 1, 2019 RESEARCH

DOI: 10.17482/uumfd.506766

BUCK, BOOST AND BUCK-BOOST CONVERTER DESIGNS WITH VARIOUS METAHEURISTIC METHODS

Fahri VATANSEVER [*]
Yiğit Çağatay KUYU [*]

Received: 02.01.2019 ; revised: 27.03.2019 ; accepted: 03.04.2019

Abstract: One of the basic circuit structures in the field of power electronics is DC-DC converters. As these design steps require many mathematical operations, these problems are hard to solve by hand. In addition, choosing the proper component values is always curial when adopting the computer-based designs to the real-world. In this study, the software is developed for the designs of buck, boost and buck-boost DC-DC converters via metaheuristic algorithms that calculate the parameters of the circuits. The components of the specified DC-DC converters are selected via the software with a user-friendly interface, under the desired criteria from the industrial series (E12, E24 and E96), by using eight different metaheuristic algorithms (artificial bee colony, differential evolution, genetic algorithm, particle swarm optimization, cuckoo search, harmony search, lightning search and gray wolf optimizer). The designs and analyses of DC-DC converters that are chosen according to the type and features (determining/selecting the components in accordance with the specified industrial series) can perform easily, fast and effectively through the software developed for this purpose.

Keywords: Power electronic, DC-DC converter, metaheuristic algorithms.

Çeşitli Metasezgisel Yöntemlerle Alçaltan, Yükselten ve Alçaltan-Yükselten Dönüştürücü Tasarımları

Öz: Güç elektroniği alanındaki temel devre yapılarından birisi DA-DA (DC-DC) dönüştürücüleridir. Farklı türleri olan bu devrelerin tasarım aşamalarında, elle çözümü zor olan birçok matematiksel işlemler gerekmektedir. Ayrıca bilgisayar tabanlı tasarımları; gerçek dünyaya uyarlarken, uygun bileşen değerlerinin seçilmesi her zaman çok önemlidir. Gerçekleştirilen çalışmada; alçaltan, yükselten ve alçaltan-yükselten DA-DA dönüştürücü devrelerin tasarımı için metasezgisel algoritmalarla hesaplamaları gerçekleştiren yazılım geliştirilmiştir. Kullanıcı dostu arayüze sahip yazılım ile seçilen türdeki DA-DA dönüştürücü devre elemanları, istenilen endüstriyel serilere (E12, E24 ve E96) uygun olarak sekiz farklı metasezgisel algoritma (yapay arı kolonisi, diferansiyel gelişim, genetik algoritma, parçacık sürü optimizasyonu, guguk kuşu arama, harmoni arama, yıldırım arama ve gri kurt optimizasyonu) ile kolay, hızlı ve etkin bir şekilde elde edilmektedir.

Anahtar Kelimeler: Güç elektroniği, DC-DC dönüştürücü, metasezgisel algoritmalar.

1. INTRODUCTION

DC-DC converter circuits, which carry a certain level of DC voltage to an another level (boosting or stepping down of voltage), are very important in the field of power electronics. The main design steps can be summarized: achieving the desired voltage level at the output,

[*] Bursa Uludag University, Faculty of Engineering, Electrical-Electronics Eng. Dept., 16059 Bursa/Turkey
Corresponding author: Fahri Vatansever (fahriv@uludag.edu.tr)

calculation of the components with different topologies of the circuits for minimum voltage fluctuation, determination of the critical components of the circuit for a continuous current and selection of the optimal component values in accordance with the industrial series.

Circuit components calculated by solving the design steps consisted of the systems of equations with multiple variables may not be existed in the industrial series. This requires all calculations to be made again by selecting the appropriate industrial series. It is challenging and time-consuming to perform all the design steps by hand as the complexities of the problem rise. Metaheuristic algorithms are able to solve these kinds of the problems quickly (León-Aldaco et al., 2015). In addition, as conventional methods require derivative information and are highly sensitive to initial points in some cases, metaheuristic algorithms are one step ahead of them in solving complex problems. In these optimization processes, metaheuristic algorithms can be utilized effectively and efficiently (Simon, 2013; Price et al., 2005; Dasgupta, 1997; Yang, 2014; Yang, 2010; Kuyu and Vatansever, 2016; Kuyu and Vatansever, 2016; Vatansever et al., 2015). The basic properties and advantages of metaheuristics can be found in the related studies (Vasant, 2012; Du and Swamy, 2016).

In this work, artificial bee colony (ABC) (Karaboga, 2005), differential evolution (DE) (Storn and Price, 1995), genetic (GA) (Goldberg, 1989), particle swarm optimization (PSO) (Kennedy and Eberhart, 1995), cuckoo search (CS) (Yang and Deb, 2009), harmony search (HS) (Geem et al., 2001), lightning search (LSA) (Shareef et al., 2015) and grey wolf optimizer (GWO) (Mirjalili et al., 2014) algorithms are used for the selection of the optimal component values of the DC-DC buck, boost and buck-boost converters. The component values of the converter circuits in the desired property can be calculated/chosen according to the industrial series via the metaheuristic algorithms easily, fast and effectively. The main objectives of this study are to reduce the time of design processes and to find proper component values automatically by combining a lot of metaheuristics with a software. Thus the most economical and convenient circuits can be designed according to available components.

This paper is organized as follows: DC-DC converters and the used metaheuristic algorithms are briefly described in Section 2-3. The designed simulator and its applications (results) are given in Section 4 and finally, the study conclusions are detailed in Section 5.

2. DC-DC CONVERTERS

The DC-DC converters changing the DC voltage levels at the input and output can be divided into three groups:

- Buck (step-down): DC/DC switching voltage regulator that reduces the input voltage.
- Boost (step-up): DC/DC switching voltage regulator that increases the input voltage.
- Buck-boost: DC/DC switching voltage regulator that reduces or increases the input voltage.

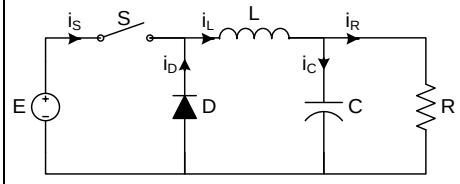
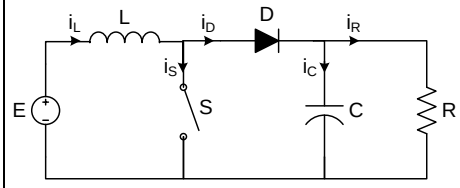
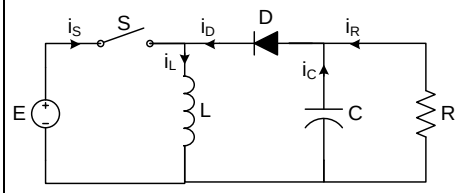
The main topologies and some equations of the converters are summarized in Table 1 (Rashid, 2013; Gürdal, 2008).

3. METAHEURISTIC ALGORITHMS

Metaheuristic algorithms are inspired from various methodologies, such as biologic and social behaviors in the nature, as in the artificial bee colony, genetic, differential evolution, particle swarm optimization algorithms, or physic-based like the lightning search algorithm. These algorithms are frequently used in the field of optimization related to social and engineering sciences. The pseudo codes of the algorithms used in this study are also summarized in Table 2 (Karaboga, 2005; Storn and Price, 1995; Goldberg, 1989; Kennedy and Eberhart, 1995; Yang and Deb, 2009; Geem et al., 2001; Shareef et al., 2015; Mirjalili et al., 2014).

Table 1. DC-DC converters



Type	Circuit	Parameter	Equation
Buck		*Duty ratio*	$D = \frac{V_C}{E}$
		Minimum current	$I_{min} = ED\left\{\frac{1}{R} - \frac{(1-D)T}{2L}\right\}$
		Maximum current	$I_{max} = ED\left\{\frac{1}{R} + \frac{(1-D)T}{2L}\right\}$
		Critical inductance	$L_{min} = \frac{TR}{2}(1-D)$
		Voltage fluctuation	$\Delta V_C = \frac{(I_{max} - I_{min})}{8C}T$
Boost		*Duty ratio*	$D = 1 - \frac{E}{V_C}$
		Minimum current	$I_{min} = \frac{E}{R(1-D)^2} - \frac{EDT}{2L}$
		Maximum current	$I_{max} = \frac{E}{R(1-D)^2} + \frac{EDT}{2L}$
		Critical inductance	$L_{min} = \frac{TRD}{2}(1-D)^2$
		Voltage fluctuation	$\Delta V_C = \frac{V_C DT}{RC}$
Buck-Boost		*Duty ratio*	$D = \frac{V_C}{E + V_C}$
		Minimum current	$I_{min} = \frac{ED}{R(1-D)^2} - \frac{EDT}{2L}$
		Maximum current	$I_{max} = \frac{ED}{R(1-D)^2} + \frac{EDT}{2L}$
		Critical inductance	$L_{min} = \frac{TR}{2}(1-D)^2$
		Voltage fluctuation	$\Delta V_C = \frac{V_C DT}{RC}$

4. APPLICATIONS

In this work, a graphical user interface was designed using MATLAB (MathWorks, 2013) to run the applications. The component values of the DC-DC converters for the critical/minimum continuous current condition are found by using ABC, DE, GA, PSO, CS, HS, LSA and GWO in the program designed. The flowchart and the main screenshot of the program can be shown in Figures 1-2, respectively.

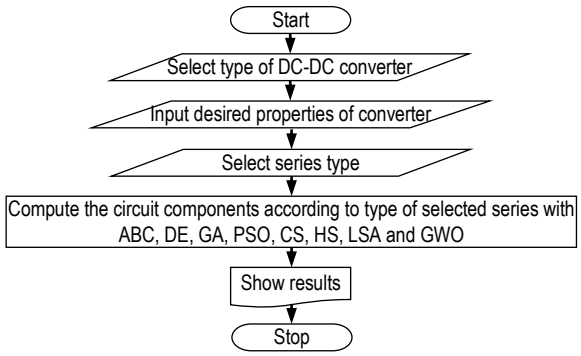


Figure 1:
Basic flow-chart of the designed program

Table 2. The algorithms used in this work

Algorithm	Years	Developers	Sample pseudo code
Artificial bee colony (ABC)	2005	Dervis Karaboga	▪ Generate initial population ▪ While (stopping criterion) ○ Produce new solutions (food source positions) ○ Apply the greedy selection process ○ Calculate the probability values for the solutions by means of their fitness values ○ Produce the new solutions (new positions) ○ Apply the greedy selection process ○ Determine the abandoned solution (source) ○ Memorize the best solution (food source position) ▪ end while
Differential evolution (DE)	1996-1997	Rainer Storn and Kenneth Price	▪ Generate initial population ▪ While (stopping criterion) ○ For each individual ➢ Compute a mutant vector ➢ For each dimension • Generate a trial vector by using recombination operator ➢ end for ➢ Select and update the solution ○ end for ▪ end while
Genetic algorithm (GA)	1960s and 1970s	John Holland and his collaborators	▪ Generate initial population ▪ While (stopping criterion) ○ Generate a new generation of population by using selection, crossover, mutation, elitism mechanisms ○ Calculate fitness values of individuals ▪ end while
Particle swarm optimization (PSO)	1995	James Kennedy and Russell C. Eberhart	▪ Initialize locations and velocity of particles ▪ While (stopping criterion) ○ Calculate the fitness value of each particle ○ Determine the location of the particle with the highest fitness ○ For each particle calculate its velocity ○ Update the location of each particle ▪ end while
Cuckoo search (CS)	2009	Xin-She Yang,and Suash Deb	▪ Generate initial population ▪ While (stopping criterion) ○ Calculate the fitness values of host nests ○ Apply the replacing process ○ Determine the cuckoo societies ○ Memorize the best cuckoo solution ▪ end while
Harmony search (HS)	2001	Zong Woo Geem and collaborators	▪ Initialize harmony memory ▪ While (stopping criterion) ○ Calculate harmony memory solutions ○ Improve new harmonies ○ Update the harmonies ○ Memorize the best harmony ▪ end while
Lightning search algorithm (LSA)	2015	Hussain Shareef, Ahmad Asrul Ibrahim, Ammar Hussein Mutlag	▪ Generate population of step leaders ▪ Calculate projectiles energies ▪ While (stopping criterion) ○ Update leader tips energies ○ Update best and worst step leaders ○ Calculate projectiles energies ○ Update step leaders ▪ end while
Grey Wolf Optimizer (GWO)	2014	Seyedali Mirjalili, Seyed Mohammad Mirjalili, Andrew Lewis	▪ Generate initial search agent positions ▪ Calculate the fitness of each search agent ▪ Rank search agents as alpha, beta, delta and omega based on fitness values ▪ While (stopping criterion) ○ Update the locations of the search agents ○ Update the coefficient vectors ○ Calculate the fitness of all search agents ▪ end while

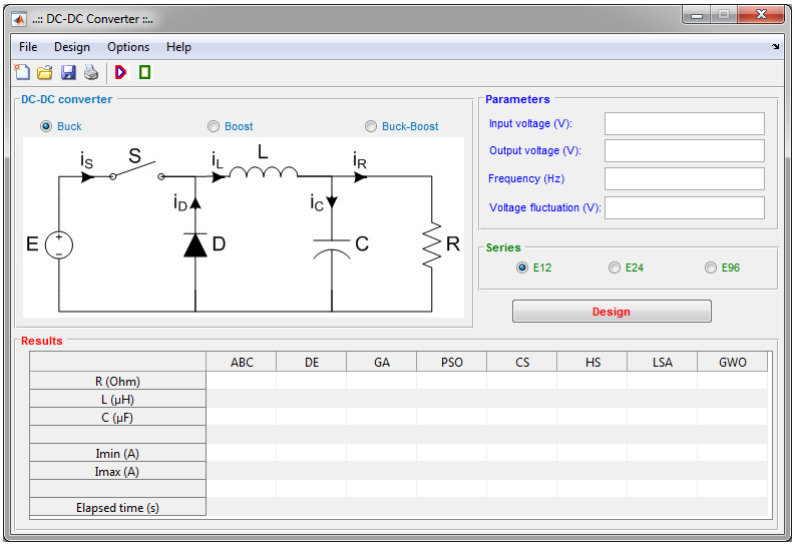


Figure 2:
Screenshot of the designed program

In the first application, the buck converter that reduces $20\,V$ to $12\,V$ (frequency of the circuit is $10\,kHz$, desired voltage fluctuation in the output is $0.1\,V$) is designed by using E12 series. The results of the first application are given in Table 3. As can be shown in this table, considering component values found by the algorithms, ABC, GA and GWO reach the similar results and GWO appears to be slightly faster in terms of run-time. Besides, the resistor values of the algorithms are the same, except DE and LSA; however, their resistor values are still compatible with E12 series. Comparing the values of I_{max} obtained by the algorithms, ABC, GA, HS and GWO algorithms achieve the similar results that are far from the results obtained by DE and LSA algorithms. When regarding the outcomes of I_{min}, the highest value belongs to DE with 1.667 A. while the lowest one is obtained by PSO algorithm, having 0.942 A. In terms of run time, GWO is able to design the circuit faster than the other competitors, getting a run time of 0.091 sec., and DE is behind GWO with a small difference which is only 0.007 sec. It is worth mentioning that all the algorithms have a capable of designing the circuit in a reasonable time.

Table 3. The results of the first application

	ABC	DE	GA	PSO	CS	HS	LSA	GWO
$R(\Omega)$	12	6.8	12	12	12	12	8.2	12
$L(\mu H)$	4999.927	2461.067	5000.000	4191.247	4625.968	4985.408	3001.274	5000.000
$C\ (\mu F)$	12.000	24.379	12.000	14.315	12.970	12.035	19.991	12.000
$I_{min}(A)$	0.952	1.667	0.952	0.942	0.948	0.951	1.543	0.952
$I_{max}(A)$	1.048	1.862	1.048	1.057	1.051	1.048	1.383	1.048
Time (sec)	0.142	0.098	0.184	0.5902	0.3081	0.3375	0.282	0.091

In the second application, the boost converter that raises the voltage from 40 V to 150 V (frequency of the circuit is 5 kHz, desired voltage fluctuation in the output is 0.1 V) is designed in accordance with E24 series and the comparative results can be shown in Table 4. The resistor values of the boost converters are found to be 36 Ω by ABC, DE, HS and LSA algorithms, 27 Ω by GA, 51 Ω by PSO algorithm and 47 Ω by GWO algorithm, respectively. In terms of I_{min} and I_{max} values, it can be seen that ABC, DE and HS algorithms achieve very similar results in the range of 14 to 17 A. As different to the first application, DE is the fastest algorithm which

achieves the results in 0.058 seconds whereas GWO is the second fastest with a run time of 0.095 sec, and ABC comes after GWO, which is almost two times slower than DE.

Table 4. The results of the second application

	ABC	DE	GA	PSO	CS	HS	LSA	GWO
$R(\Omega)$	36	36	27	51	33	36	36	47
$L(\mu H)$	3487.883	3503.567	2782.241	4954.074	3223.961	3549.415	3569.358	4657.413
$C\ (\mu F)$	6111.111	6111.111	8148.148	4313.725	6666.666	6111.111	6111.111	4680.851
$I_{min}(A)$	14.783	14.787	19.779	10.437	16.135	14.798	16.446	11.338
$I_{max}(A)$	16.466	16.462	21.887	11.621	17.955	16.451	14.803	12.597
Time (sec)	0.124	0.058	0.243	0.432	0.3382	0.3158	0.298	0.095

In the third application, the buck converter is designed with E96 series by using the same input values of the first application. The results given in Table 5 show that DE is the best algorithm with respect to run time as similar to second applications whereas PSO is the slowest one with a run time of 0.527 sec. which is close to that of GA. When comparing Table 5 with Table 3 in terms of the resistor values found by the algorithms, it can be seen that the probability of the finding different resistor rises as the number of E-series (from E12 to E96) used increases. This is because E96 series have more possible combinations of component values than E12 series. It can be observed from the both tables that the run time of most of the algorithms slightly increases. Generally speaking, the results show that the resistor values found by the algorithms are totally compatible with the industrial series and the run time of all the algorithms is remarkably low for all the applications.

Table 5. The results of the third application

	ABC	DE	GA	PSO	CS	HS	LSA	GWO
$R(\Omega)$	13.3	7.68	13	11.8	11.8	9.31	12.1	10.5
$L(\mu H)$	4999.342	2885.139	4882.611	4518.699	4517.772	3519.183	4558.361	3980.694
$C\ (\mu F)$	12.001	20.796	12.288	13.278	13.477	17.049	13.162	15.072
$I_{min}(A)$	0.854	1.479	0.873	0.964	0.963	1.221	0.939	1.0826
$I_{max}(A)$	0.950	1.645	0.972	1.070	1.071	1.357	1.044	1.2031
Time (sec)	0.162	0.074	0.355	0.527	0.341	0.340	0.269	0.118

5. CONCLUSION

Determining proper component values is a crucial part on the design of the circuits and components chosen from the industrial series make the designs easier, more reliable and flexible to realize. Besides, obtaining the component values with parallel or series connections can be time-consuming while designing the circuits. In this work, the resistor values of DC-DC converters, which are compatible with the industrial series (E12, E24, E96), have been found and some current values have been calculated via the eight different algorithms called ABC, DE, GA, PSO, CS, HS, LSA and GWO. The software developed for this purpose can design the related circuits comparatively with the selected metaheuristic algorithms and is able to find proper component values automatically, easily, fast and effectively. As a future work, it is planned to develop a web platform that will automatically identify all components of DC-DC converters with up-to-date metaheuristic algorithms.

REFERENCES

1. Dasgupta, D., Michalewicz, Z. (Eds.) (1997) *Evolutionary Algorithms in Engineering Applications*, Springer, Berlin.

2. Du, K. and Swamy, M. N. S. (2016) *Search and Optimization by Metaheuristics*, Birkhauser, Switzerland.

3. Geem, Z.W., Kim, J.H., Loganathan, G.V. (2001) A new heuristic optimization algorithm: harmony search, *Simulation*, 76 (2), 60-68. doi: 10.1177/003754970107600201

4. Goldberg, D. E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, USA.

5. Gürdal, O. (2008) *Güç Elektroniği*, Seçkin Yayıncılık, Ankara.

6. Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization, *Technical report-TR06*, Erciyes University, Engineering Faculty, Computer Engineering Department, Kayseri.

7. Kennedy J. and Eberhart R. (1995) Particle swarm optimization, *IEEE International Conference on* Neural *Networks*, 1942-1948. doi:10.1109/ICNN.1995.488968

8. Kuyu, Y.C. and Vatansever, F. (2016) A new intelligent decision making system combining classical methods, evolutionary algorithms and statistical techniques for optimal digital FIR filter design and their performance evaluation, *International Journal of Electronics and Communications*, 70, 1651-1666. doi: 10.1016/j.aeue.2016.10.004

9. Kuyu, Y.C. and Vatansever, F. (2016) Optimization of the Rectifier Circuit Components with Evolutionary Algorithms, *International Scientific Symposium "Electrical Power Engineering 2016"*, Varna/Bulgaria, 6-8. 100-103.

10. León-Aldaco, D., Estefany, S., Calleja, H., Jesús, A. A. (2015) Metaheuristic optimization methods applied to power converters: A review, *IEEE Transactions on Power Electronics*, 30 (12), 6791-6803. doi: 10.1109/TPEL.2015.2397311.

11. MATLAB, The MathWorks, Inc. , https://www.mathworks.com/

12. Mirjalili, S., Mirjalili, S.M., Lewis, A. (2014) Grey wolf optimizer, *Advances in Engineering Software*, 69, 46-61. doi: 10.1016/j.advengsoft.2013.12.007

13. Price, K.V., Storn, R.M., Lampinen, J.A. (2005) *Differential Evolution: A Practical Approach to Global Optimization*, Springer, Berlin.

14. Rashid, M.H. (2013) *Power Electronics: Circuits, Devices & Applications*, 4th ed., Pearson Education.

15. Shareef, H., Ibrahim, A.A., Mutlag, A.H. (2015) Lightning search algorithm, *Applied Soft Computing*, 36, 315-333. doi: 10.1016/j.asoc.2015.07.028

16. Simon, D. (2013) *Evolutionary Optimization Algorithms*, Wiley, New Jersey.

17. Storn, R. and Price, K. (1995). Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces. *Technical report-TR95*, International Computer Science Institute.

18. Vasant, P. M. (2012) *Meta-heuristics Optimization Algorithms in Engineering, Business, Economics, and Finance*, IGI Global, USA.

19. Vatansever, F., Yalcin, N.A., Kuyu, Y.C. (2015) Evrimsel Algoritmalarla Tristörlü Doğrultucu Devrelerindeki Tetikleme Açılarının Hesaplanması, *Uludağ University Journal of The Faculty of Engineering*, 20(2), 67-77. doi: 10.17482/uujfe.13975

20. Yang, X.-S. (2010) *Engineering Optimization: An Introduction with Metaheuristic Applications*, Wiley.

21. Yang, X.-S. (2014) *Nature-Inspired Metaheuristic Algorithms*, Elsevier.

22. Yang, X.S., Deb, S. (2009) Cuckoo search via lévy flights, *World congress on nature & biologically inspired computing (NABIC)*, 210-214.