The Use of Genetic Programming to Evolve Passive Filter Circuits

Ogri J. Ushie[1,*], Maysam F. Abbod[2], and Julie C. Ogbulezie[3]

[1,2]Department of Electronic and Computer Engineering, College of Engineering, Design and Physical Sciences, Brunel University London, Uxbridge, UK.

[1,3]Department of Physics, University of Calabar, Calabar, Nigeria.

Received 15 January 2016; received in revised form 20 March 2017; accepted 27 July 2017

Abstract

This paper introduces the use of Genetic Programming (GP), Genetic Folding and symbolic circuit analysis in Matlab for the evolution of passive filter circuits. Instead of combining MATLAB and PSPICE in electronic circuit simulation, in this work, only MATLAB is used. It helps to reduce elapsed time for transferring the simulation between the two software packages. The circuit evolved from GP using the Matlab program and is automatically converted into a symbolic netlist also by using a Matlab code. The netlist is fed into symbolic circuit analysis in Matlab (SCAM); the SCAM is used to generate matrices that are used for simulation. In this case, it is used to analyse frequency response of passive low-pass, high-pass and band-pass filter circuits. The algorithm is tested with four different examples and the results presented have proved that the algorithm is efficient concerning the design wise. The work has provided an alternative way of using GP for the evolution of passive filter circuits.

Keywords: genetic folding, genetic programming, netlist, passive filter circuits, symbolic circuit analysis in Matlab

References

[1] K. K. Anumandla, R. Peesapati, S. L. Sabat, S. K. Udgata, and A. Abraham, "Field programmable gate arrays-based differential evolution coprocessor: a case study of spectrum allocation in cognitive radio network," IET Computers & Digital Techniques, vol. 7, no. 5, pp. 221-234, August 2013.

[2] E. A. Coyle, L. P. Maguire, and T. M. McGinnity, "Design philosophy for self-repair of electronic systems using the UML," IEE Proceedings-Software, vol. 149, no. 6, pp. 179-186, December 2002.

[3] S. Maheshwari, "Analogue signal processing applications using a new circuit topology," IET Circuits, Devices & Systems, vol. 3, no. 3, pp. 106-115, June 2009.

[4] A. Tyrrell, R. Krohling, and Y. Zhou, "Evolutionary algorithm for the promotion of evolvable hardware," IEE Proceedings-Computers and Digital Techniques, vol. 151, no. 4, pp. 267-275, July 2004.

[5] S. Vakili, S. M. Fakhraie, and S. Mohammadi, "Evolvable multi-processor: a novel MPSoC architecture with evolvable task decomposition and scheduling," IET Computers & Digital Techniques, vol. 4, no. 2, pp. 143-156, March 2010.

[6] J. Wang, Q. S. Chen, and C. H. Lee, "Design and implementation of a virtual reconfigurable architecture for different applications of intrinsic evolvable hardware," IET Computers & Digital Techniques, vol. 2, no. 5, pp. 386-400, September 2008.

[7] B. Singh, V. Verma, A. Chandra, and K. Al-Haddad, "Hybrid filters for power quality improvement," IEE Proceedings-Generation, Transmission and Distribution, vol. 152, no. 3, pp. 365-378, May 2005.

[8] S. Maheshwari and B. Chaturvedi, "High-input low-output impedance all-pass filters using one active element," IET Circuits, Devices & Systems, vol. 6, no. 2, pp. 103-110, March 2012.

[9] G. Alpaydin, S. Balkir, and G. Dündar, "An evolutionary approach to automatic synthesis of high-performance analog integrated circuits," IEEE Transactions on Evolutionary Computation, vol. 7, no. 3, pp. 240-252, June 2003.

[10] E. Martens and G. Gielen, "Classification of analog synthesis tools based on their architecture selection mechanisms," Integration, the VLSI Journal, vol. 41, no. 2, pp. 238-252, February 2008.

* Corresponding author. E-mail address: ogri.ushie@unical.edu.ng

[11] O. Mitea, M. Meissner, L. Hedrich, and P. Jores, "Automated constraint-driven topology synthesis for analog circuits," Design, Automation & Test in Europe Conference & Exhibition, IEEE press, May 2011, pp. 1-4.
[12] M. Harman, W. B. Langdon, Y. Jia, D. R. White, A. Arcuri, and J. A. Clark, "The GISMOE challenge: Constructing the pareto program surface using genetic programming to find better programs (keynote paper)," Proc. IEEE/ACM International Conf. Automated Software Engineering, IEEE press, April 2012, pp. 1-14.
[13] S. Forrest, T. Nguyen, W. Weimer, and C. Le Goues, "A genetic programming approach to automated software repair," Proc. Annual Conf. Genetic and Evolutionary Computation, ACM press, July 2009, pp. 947-954.
[14] W. Weimer, S. Forrest, C. Le Goues, and T. Nguyen, "Automatic program repair with evolutionary computation," Communications of the ACM, vol. 53, no. 5, pp. 109-116, May 2010.
[15] W. Weimer, T. Nguyen, C. Le Goues, and S. Forrest, "Automatically finding patches using genetic programming," Proc. International Conf. Software Engineering, ACM press, May 2009, pp. 364-374.
[16] J. D. Lohn and S. P. Colombano, "A circuit representation technique for automated circuit design," IEEE Transactions on Evolutionary Computation, vol. 3, no. 3, pp. 205-219, September 1999.
[17] J. R. Koza, "Human-competitive results produced by genetic programming," Genetic Programming and Evolvable Machines, vol. 11, no. 3-4, pp. 251-284, May 2010.
[18] J. R. Koza, S. H. Al-Sakran, and L. W. Jones, "Cross-domain features of runs of genetic programming used to evolve designs for analog circuits, optical lens systems, controllers, antennas, mechanical systems, and quantum computing circuits," Proc. 2005 NASA/DoD Conf. Evolvable Hardware, IEEE press, September 2005, pp. 205-212.
[19] J. R. Koza, F. H. Bennett III, D. Andre, M. A. Keane, and F. Dunlap, "Automated synthesis of analog electrical circuits by means of genetic programming," IEEE Transactions on Evolutionary Computation, vol. 1, no. 2, pp. 109-128, July 1997.
[20] J. R. Koza, "Genetic programming as a means for programming computers by natural selection," Statistics and Computing, vol. 4, no. 2, pp. 87-112, June 1994.
[21] M. Walker, "Introduction to genetic programming," Tech.Np: University of Montana, 2001.
[22] T. Sripramong and C. Toumazou, "The invention of CMOS amplifiers using genetic programming and current-flow analysis," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 21, no. 11, pp. 1237-1252, November 2002.
[23] S. Silva and J. Almeida, "GPLAB-a genetic programming toolbox for MATLAB," Proc. Nordic MATLAB Conference, 2005, pp. 273-278.
[24] K. Rodríguez and R. Mendoza, "A Matlab genetic programming approach to topographic mesh surface generation," Engineering Education and Research Using MATLAB, pp.427-442, 2011.
[25] H. Hou, S. Chang, and Y. Su, "Economical passive filter synthesis using genetic programming based on tree representation," IEEE International Symp. Circuits and Systems, IEEE press, July 2005, p. 3003.
[26] S. Chang and Y. Su, "Automated passive filter synthesis using a novel tree representation and genetic programming," IEEE Transactions on Evolutionary Computation, vol. 10, no. 1, pp. 93-100, February 2006.
[27] A. Senn, A. Peter, and J. G. Korvink, "Analog circuit synthesis using two-port theory and genetic programming," AFRICON, IEEE press, November 2011, pp. 1-8.
[28] M. Mezher and M. F. Abbod, "A new genetic folding algorithm for regression problems," UKSim 14th International Conf. Computer Modelling and Simulation (UKSim), IEEE press, May 2012, pp. 46-51.
[29] S. Boyd, S. Kim, L. Vandenberghe, and A. Hassibi, "A tutorial on geometric programming," Optimization and Engineering, vol. 8, no. 1, pp. 67-127, March 2007.
[30] G. Gielen and W. Sansen, Symbolic analysis for automated design of analog integrated circuits, Springer Science & Business Media, New York, 2012.
[31] O. J. Ushie, M. Abbod, and E. Ashigwuike, "Matlab symbolic circuit analysis and simulation tool using PSpice netlist for circuits optimization," International Journal of Engineering and Technology Innovation, vol. 5, no. 2, pp. 75-86, April 2015.
[32] E. Cheever, "Symbolic Circuit Analysis in MATLAB (SCAM)," Swarthmore College, http://www.swarthmore.edu/, November, 2005.
[33] M. Jamil and X. Yang, "A literature survey of benchmark functions for global optimisation problems," International Journal of Mathematical Modelling and Numerical Optimisation, vol. 4, no. 2, pp. 150-194, 2013.
[34] R. Feldt, M. O'Neill, C. Rayn, P. Nordin, and W. B. Langdon, "GP-beagle: A benchmarking problem repository for the genetic programming community," Late Breaking Papers at GECCO, UCL Discovery press, June 2000, pp. 1-8.