



**Universidade de São Paulo**

**Biblioteca Digital da Produção Intelectual - BDPI**

---

Departamento de Engenharia Elétrica - EESC/SEL

Artigos e Materiais de Revistas Científicas - EESC/SEL

---

2013-01

# Genetic algorithms applied to phasor estimation and frequency tracking in PMU development.

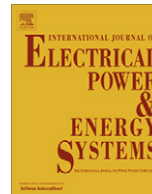
---

International Journal of Electrical Power and Energy Systems, London, v. 44, n. 1, p. 921-929, Jan.

2013

<http://www.producao.usp.br/handle/BDPI/46662>

*Downloaded from: Biblioteca Digital da Produção Intelectual - BDPI, Universidade de São Paulo*



# Genetic algorithms applied to phasor estimation and frequency tracking in PMU development

R.P.M. Silva<sup>a</sup>, A.C.B. Delbem<sup>b</sup>, D.V. Coury<sup>a,\*</sup>

<sup>a</sup> Department of Electrical and Computing Engineering, Engineering School of São Carlos, University of São Paulo, São Carlos (SP), Brazil

<sup>b</sup> Institute of Mathematical and Computer Sciences, University of São Paulo, São Carlos (SP), Brazil

## ARTICLE INFO

### Article history:

Received 27 November 2011  
Received in revised form 30 July 2012  
Accepted 31 July 2012  
Available online 29 September 2012

### Keywords:

Phasor estimations  
Genetic algorithms  
Phasor measurement unit  
Discrete fourier transform  
Phase locked-loop

## ABSTRACT

This paper presents an efficient intelligent tool applied to phasor measurements and frequency tracking of fundamental components for PMU application. The estimation task is modeled as an optimization problem in order to use genetic algorithms to search for optimal solutions. Very promising results are presented. This approach is compared to traditional methods considering the IEEE C37.118 standard and the results show that this intelligent tool offers better performance, especially during transient events, considering traditional methods. The proposed approach is implemented in hardware using FPGAs to take advantage of the intrinsic parallelism of genetic algorithms, making it applicable to real-time estimations.

© 2012 Elsevier Ltd. All rights reserved.

## 1. Introduction

Synchronized phasor measurement units were introduced in the mid-1980s as a solution for the need of more efficient and safer monitoring devices for Electric Power Systems. Since then, measuring Electric Power System (EPS) parameters of voltage and current in relatively distant buses has received great attention from researchers [1]. Such measurements are performed by phasor measurement units (PMUs), synchronized by Global Positioning System (GPS) satellites. The importance of PMUs can be emphasized by the following examples:

- **Monitoring:** State estimation is a process that determines the state of the power system to allow the system operator to make better decisions aimed at maintaining power system security in the face of various contingencies. Improvement in the accuracy of the state estimation of the power system network, becoming a state measurement, is one of the most immediate benefits of PMU application [2].
- **Advanced Network Protection:** From the protection theory, differential protection is recognized as one of the most reliable ways to protect EPS elements [3]. Synchronized measurements using

adequate communication protocols can use differential logic to protect transmission lines as measurements can be shared considering long distances in 1–2 cycles [2].

- **Advanced Control Schemes:** Control devices such as Smart Valve Controllers (SVCs) and power system stabilizers are designed to act in such a way that the defined control objective functions are optimized. Since the control is defined as a function of distant bus variables, synchronized measurements are a good way of sending remote measurements to the controller [3].
- **Computer Model Validation:** Last but not least, the validation of computational models of transmission and generation systems can be done by comparing results from these models with data from real systems obtained by PMUs [4].

In order to define the data formats, as well as measurement conventions for the PMUs, the IEEE published the IEEE-Std.1344 in 1991 [5]. A revised standard was issued in 2005 resulting in the IEEE C37.118 standard [6]. According to [2], the IEEE C37.118 standard does not specify the measurement technique to be utilized by a PMU, but it defines the performance requirements of a certain technique in a steady state condition [7]. Therefore, a wide range of methods for phasor estimation can be used in a PMU.

Some techniques used for this purpose must be highlighted: the Discrete Fourier Transform (DFT) and correction algorithms considering its limitations [8–13], algorithms based on the Least Error Squares method [14,15], the Phase Locked-Loop filters (PLLs) [16,17], as well as the Kalman filter [18,19].

\* Corresponding author at: Department of Electrical and Computing Engineering, Engineering School of São Carlos, University of São Paulo, Av. Trabalhador Sancarlene, 400 São Carlos, SP, Brazil. Tel.: +55 16 33738133; fax: +55 16 3373 9372.

E-mail addresses: [rapphil@usp.br](mailto:rapphil@usp.br) (R.P.M. Silva), [acbd@icmc.usp.br](mailto:acbd@icmc.usp.br) (A.C.B. Delbem), [coury@sc.usp.br](mailto:coury@sc.usp.br) (D.V. Coury).

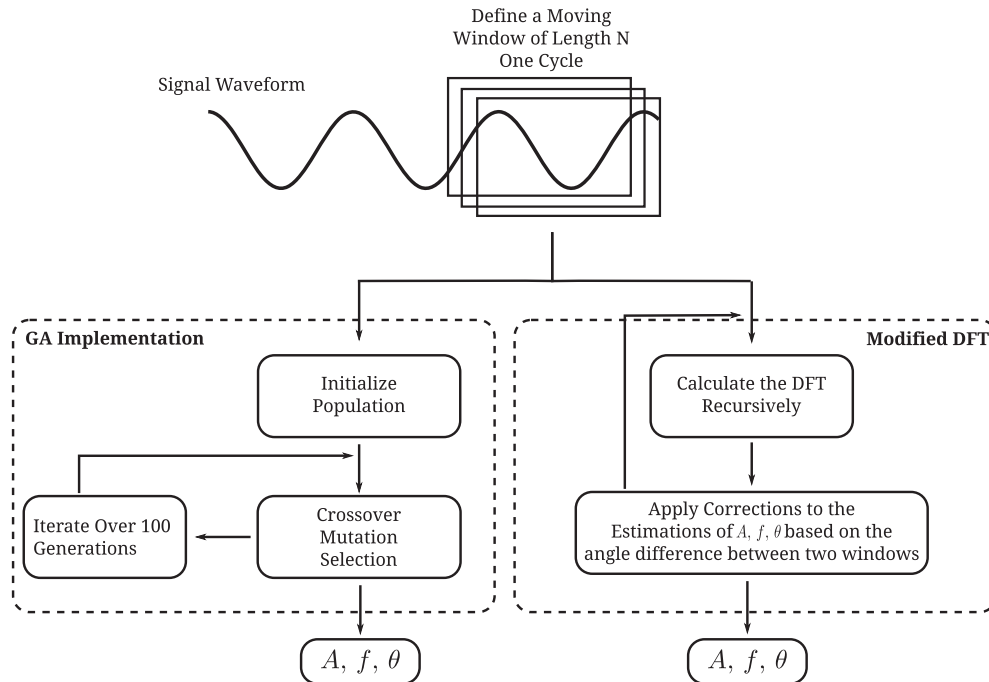


Fig. 1. Flow charts for the GA and DFT implementations.

According to [7], most algorithms used in commercial PMUs are based on the DFT. One of the first experiments involving synchronized measurements used the DFT filter to perform phasor calculations [20]. Concerning the DFT application, the weakness of the filters lies in the error caused when a frequency deviation exists. Phadke et al. [10] present a correction algorithm on the DFT filter results and frequency tracking based on the phase error caused by frequency deviations. Benmouyal [12] proposes an approach based on the DFT where the sampling rate is adjusted according to the frequency deviation, enabling synchronous sampling. Begovic et al. [13] present a method based on polynomial fitting of the DFT. Hart et al. [11] proposes the implementation of frequency tracking and phasor estimation for a generator relay using the adjustment of the size of the data window according to the frequency estimation.

Refs. [9,8] also modify the DFT results. Instead of considering the partial error of the phase resultant from the DFT, this estimation process is based on the total error and consequently improves the performance of the algorithm. Another method that should be highlighted due to its simplicity is based on PLL systems. A set of non-linear differential equations governs the dynamics of the algorithm. The algorithm estimates the phasor and frequency parameters by a set of recursive equations obtained by applying the gradient descent method to the set of differential equations and they are discretized by first-order approximations for derivatives [16]. Ziarani and Konrad [17] improved this method, gaining speed of convergence.

As an alternative for traditional methods, techniques such as Genetic Algorithms (GAs) have been used over the last years. The continuous development of hardware have enabled real-time implementation of complex methods for a variety of applications. El-Naggar and Youssef [21] presents a pioneer publication using GAs in estimation problems for frequency relays. Pursuing real-time applications, an optimized GA was presented in [22,23]. Ref. [24] presents experimental results of a GA estimating frequency effectively executed in FPGAs in real-time.

This work presents an application of GAs as an alternative method to phasor estimation and frequency tracking in an electric

power system. Some inherent characteristics of the GAs are explored in order to obtain a robust method when compared to traditional ones. The proposed approach leads to a more robust estimation, especially when the system is facing transient events. A good performance of the proposed algorithm is assured with the aid of computer simulations and a laboratory set-up in order to demonstrate its functionality is presented. The results are compared to two traditional techniques: Discrete Fourier Transform and Phase Locked-Loop with very promising results.

Fig. 1 shows a flowchart concerning the new methodology, as well as the flowchart for the implementation of the modified DFT algorithm [8,9] considered as the traditional method normally used in PMU implementations. The Phase Locked-Loop algorithm was also implemented using Ref. [17]. This work is organized as follows: the GA fundamentals are described in Section 2, where the application of GA for phasor measurement is also discussed. Section 3 presents the accuracy metric and experimental results concerning the implemented methods, as well as the implementation of GA in FPGA. Finally, Section 4 presents the conclusions concerning the new technique proposed.

## 2. Genetic algorithms

GAs can be seen as optimization algorithms from a broad set of methods called evolutionary algorithms, which are inspired in aspects of natural systems [25,26]. GAs are founded on principles of natural selection and population genetics proposed by Holland and Goldberg [27,28]. These algorithms operate with a set of possible solutions (called a population of individuals) to solve a problem. The best individuals according to an objective function are modified, using genetic operators, in order to supposedly improve those individuals towards a global optimum. In general, an individual is encoded as a string (chromosome) containing the potential values of the variables of an optimization problem. Thus, the genetic operators change these strings generating new strings. Then, the most relevant individuals are selected to survive, completing a cycle of evolution, called generation. After several generations, some

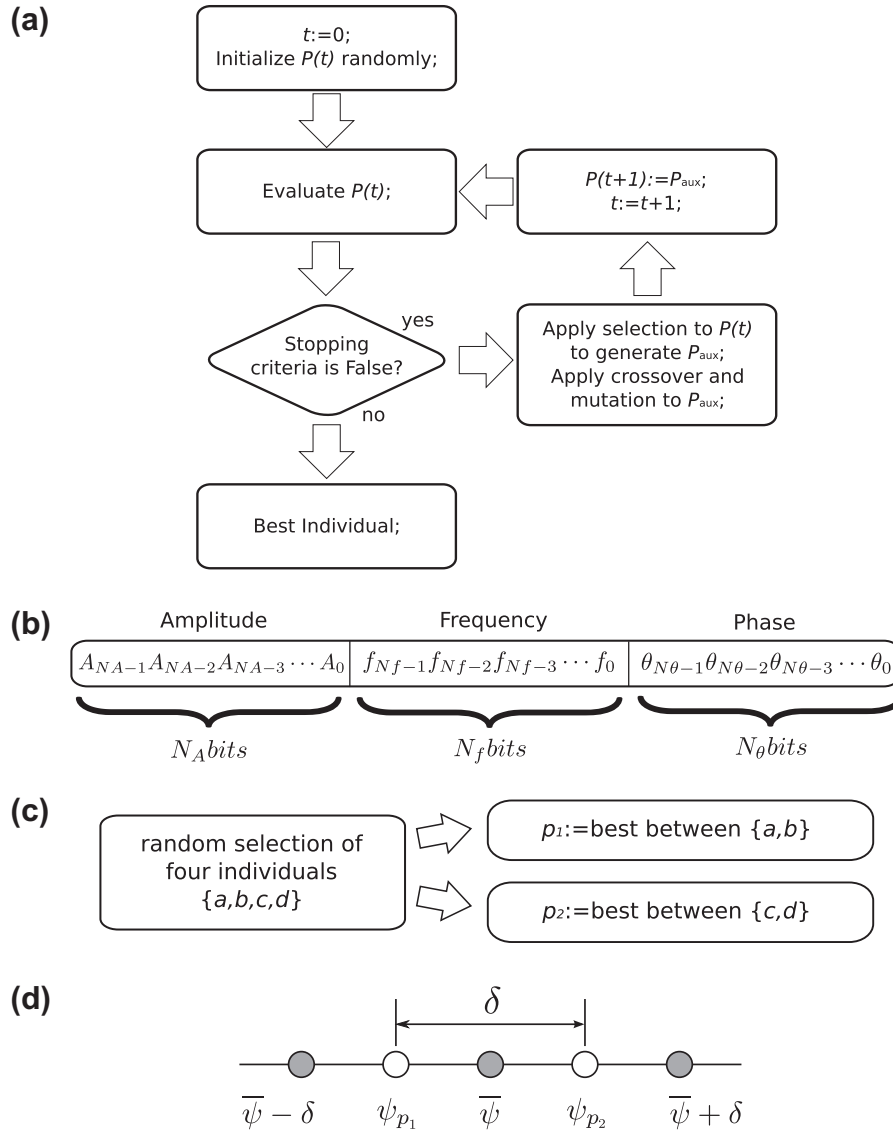


Fig. 2. Architecture of a GA applied to phasor measurement. (a) Execution flow of a GA. (b) Individual encoding. (c) Selection operator. (d) Crossover operator.

intrinsic properties of the GA [25,26] guide individuals towards the global optimum of the problem. Section 2.1 annotates the essential components of a GA. Section 2.2 presents an architecture implemented in FPGA in order to parallelize as much as possible the processing involved in the GA and to apply it in real-time to work as a PMU performing amplitude, phase and frequency estimations.

### 2.1. Overview of a GA

The fundamental principle of a GA is that the fittest individual of a population has the highest probability to survive. Thus, it is essential to evaluate as many solutions as possible. This evaluation is performed by associating a fitness value for each individual. The fitness is computed from the objective function of the problem.

Crossover and mutation are the two main genetic operators applied to modify survivors to generate new individuals (supposedly better ones) for the next generation. These operators are responsible for establishing how individuals exchange or change their genetic characteristics (values of the problem variables) in order to produce new individuals. A typical execution flow of a GA is shown in Fig. 2a, where  $t$  is the generation index and  $P(t)$  is the population at generation  $t$ .

Three main characteristics of GAs are highlighted here. Firstly, these algorithms can determine the optimum of complex optimization problems, even if they are discrete or their derivatives are not defined. Secondly, a random initialization creates a population with individuals in the overall search space, defined for the maximum and minimum values of each parameter of the problem. This process generally improves the search for a global optimum instead of local optima. Finally, the implicit parallelism of GAs can generate the newest population with parallel processing, since the generation of new individual depends on individuals from previous generations only.

### 2.2. Genetic algorithms applied to phasor estimation

The optimization problem using GAs to estimate phasors of an Electric Power System is defined considering a sinusoidal model for the voltage input signal. The electrical signal can be analyzed considering a sliding window with length  $N$ . Mathematically, the cost function value at time instant  $n$  is defined as:

$$e[n] = C(n, A, f, \theta) = \sum_{k=0}^{N-1} |u[n-k] - A \sin(2\pi f k T + \theta)| \quad (1)$$

**Table 1**  
Parameters used in the GA implementation.

Parameter	Value
Crossover rate	80%
Mutation rate	1%
Frequency initialization	[59, 5:60, 5] Hz
Amplitude initialization	[0.9:1.1]
Phase initialization	[0:2 $\pi$ ]
Population size	60
Maximum number of generation-stop criteria	100

where  $N$  is the number of samples of the window,  $u$  is the input signal and  $\{A, f, \theta\}$  is the set of parameters to be found in order to minimize the summation, that is, to estimate the input signal.

### 2.2.1. Encoding

In this work, a binary code scheme is used to represent the parameters of the set  $\{A, f, \theta\}$ . Each chromosome  $\psi$  has three genes:  $\psi_A, \psi_f$  e  $\psi_\theta$ . Each gene indexes an array ( $a_A, a_f$  or  $a_\theta$ ) that stores quantized values of one of the three parameters. The indices use different numbers of bits of a chromosome ( $N_A = 11, N_f = 24$  and

$N_\theta = 11$ , as shown in Fig. 2b). Thus, a chromosome is an array  $\psi = [\psi_A \psi_f \psi_\theta]$  of 46 bits [23].

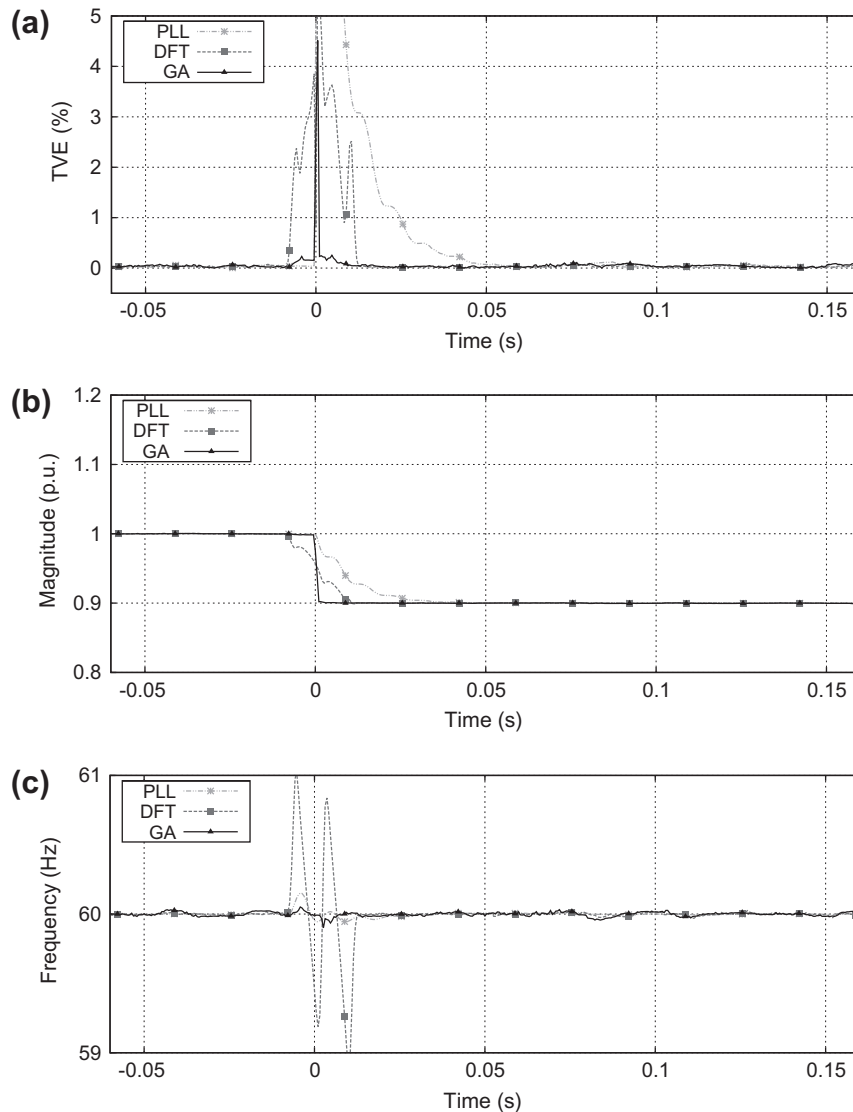
### 2.2.2. Selection

The selection process consists of randomly choosing individuals for reproduction. These individuals are called parents. Two parents are selected according to the algorithm described in Fig. 2c. Four individuals  $\{a, b, c, d\}$  are randomly chosen from the current population. Parents  $p_1$  and  $p_2$  are results of competition (according to their fitnesses) between pairs  $\{a, b\}$  and  $\{c, d\}$ , respectively. This type of selection has been reported to give adequate results for various application domains.

### 2.2.3. Crossover

The crossover process guides the evolutionary process towards potentially better solutions. This operator interchanges genetic material from chromosomes  $p_1$  and  $p_2$  resulting from the selection stage to create offspring that can benefit from the parent's fitness.

The crossover between parents  $\psi_{p_1}$  and  $\psi_{p_2}$  is performed as represented generically in Fig. 2d. First, the mean of the indices in  $\psi_{p_1}$  and  $\psi_{p_2}$  for a parameter and distance  $\delta$  between them are



**Fig. 3.** Results for the magnitude step test: (a) TVE for the magnitude step test. (b) Magnitude estimation for the magnitude step test. (c) Frequency estimation for the magnitude step test.

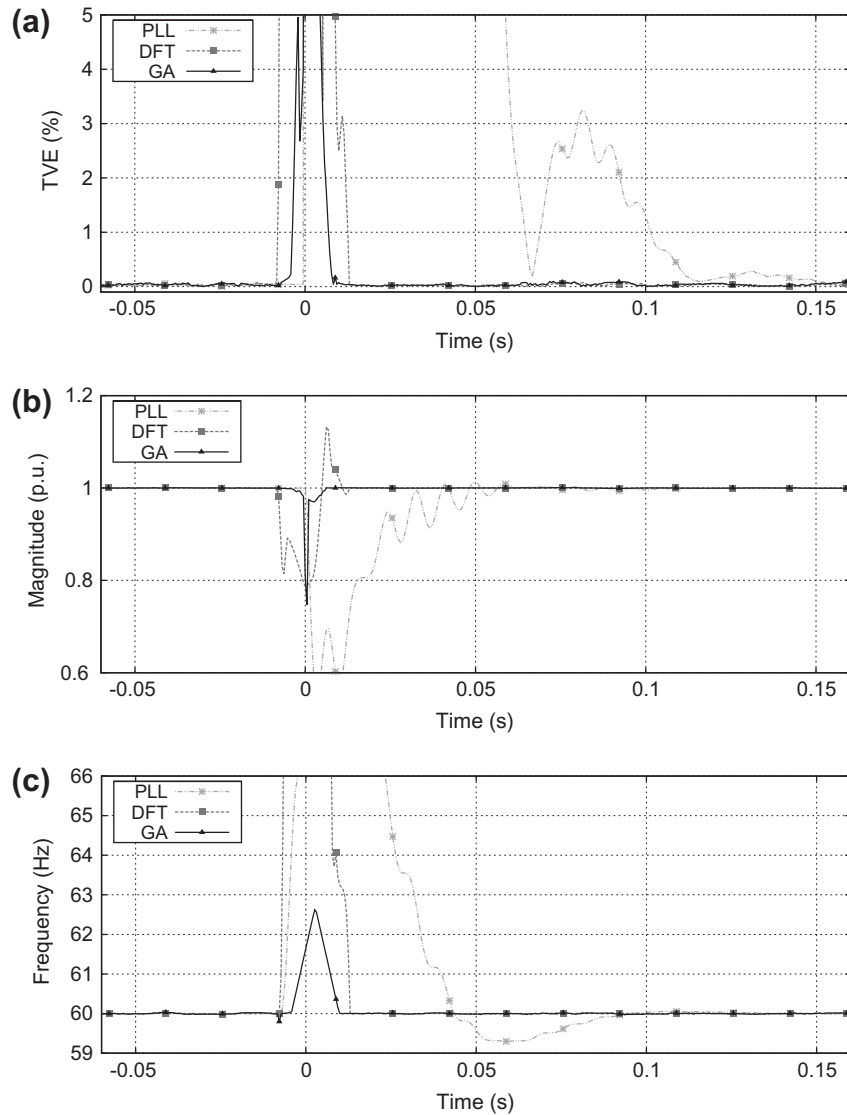


Fig. 4. Results for the phase step test: (a) TVE for the phase step test. (b) Magnitude estimation for the phase step test. (c) Frequency estimation for the magnitude step test.

computed. Then, an index for the parameter of the offspring is randomly chosen from the five possible values created by adding (or subtracting)  $\delta$  to (from) the indices of this parameter in  $\psi_{p_1}$  and  $\psi_{p_2}$  or by preserving their original values.

This procedure leads to a 20% probability in choosing each value and it is repeated three times, one time for each parameter of  $\psi$ . In this work, the crossover actually modifies only 80% (the crossover rate) of selected parents  $p_1$  and  $p_2$ .

#### 2.2.4. Mutation

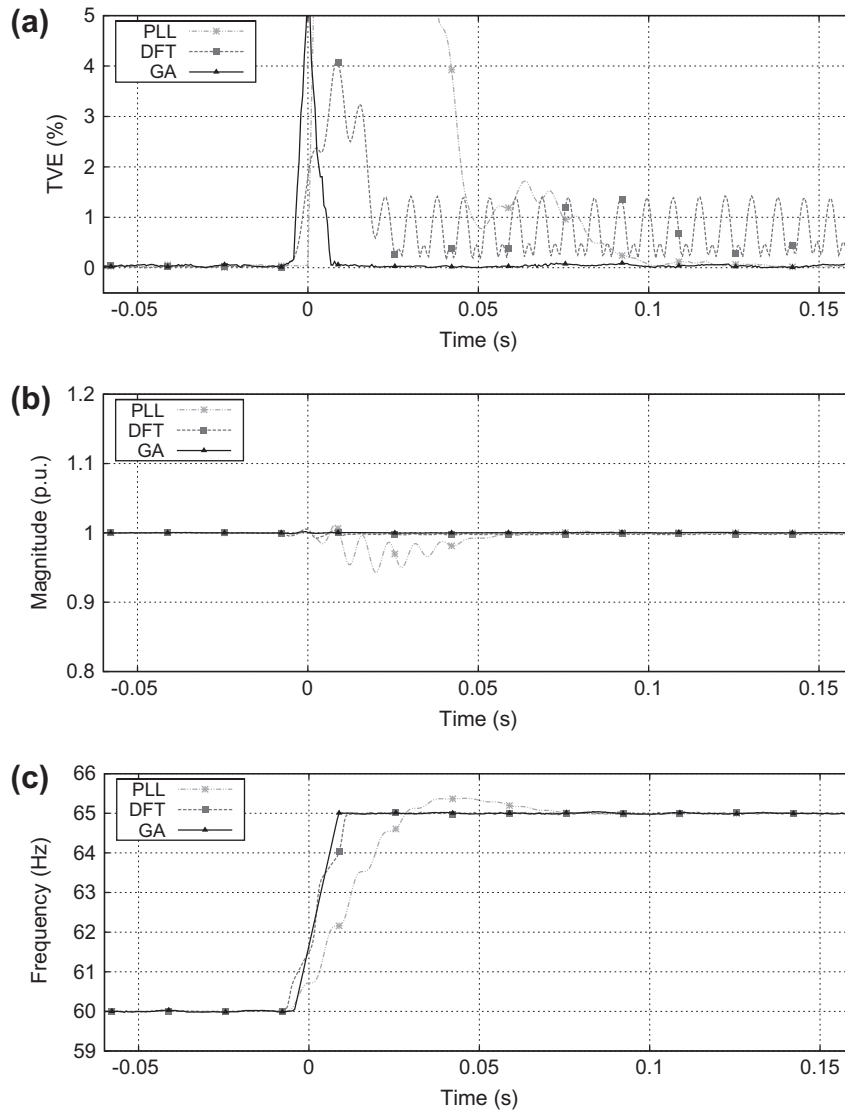
Mutation is the reproduction operator responsible for generating diversity of genetic material in the population. Basically, it is applied separately to each parameter of the offspring resulting from the crossover. The mutation occurs according to a probability (the mutation rate) and the strategy used is to add (or subtract) 1 to (from) a parameter index of  $\psi$ . The mutation rate used was 1%.

Table 1 summarizes the parameters used concerning the GA implementation, such as crossover and mutation rates, frequency/phase/amplitude initialization, population size and maximum number of generations. A maximum number of generations (100) was used as a stop criteria for the GA process.

### 3. Experimental results concerning the studied method

The results presented in this work are divided into two categories. The first one is related to results obtained using synthetic data for transient situations, generated according to the IEEE C37.118 standard, and steady-state situations with different noise levels. The second category is related to results obtained with data from simulations using the Alternative Transient Program (ATP) software [29] and a GA implemented in FPGA. The quality of the results using synthetic data is evaluated using a metric called Total Vector Error (TVE), defined in the standard. The results of the tests performed using data from the ATP software are validated using a reference frequency provided by the software itself.

The proposed algorithm was based on GAs and for comparison purposes two other methods were implemented, one based on PLL [17] and another one based on the DFT [8,9]. The three techniques were implemented using the C++ programming language. The calculated phasors, using the implemented algorithms, were obtained with reference to the center of the analyzed data window. The GA and DFT algorithms used a data window of one cycle of the fundamental component. The GA was implemented in FPGA using the VHDL [30] hardware description language with



**Fig. 5.** Results for the frequency step test: (a) TVE for the frequency step test. (b) Magnitude estimation for the frequency step test. (c) Frequency estimation for the frequency step test.

the same design defined for the software implementation. The output of the implemented algorithms using synthetic and simulated data are described in figures and tables presented in the next sections.

### 3.1. Measurement accuracy

The IEEE C37.118 standard [6] uses the TVE to measure the accuracy of the phasors provided by a PMU. It is the vectorial difference between the measured (MEAS) and expected (IDEAL) values of a phasor for a measurement at a given instant of time ( $k$ ). The TVE is defined in the following equation:

$$TVE_{(k)} = 100\% \frac{|\vec{X}_{MEAS(k)} - \vec{X}_{IDEAL}|}{|\vec{X}_{IDEAL}|} \quad (2)$$

This metric mixes together three possible source of errors: magnitude, phase and timing. On the other hand, the standard does not specify the method of measurement or other factors such as sampling rates, algorithms or synchronization methods. It mandates the TVE should remain below 1% in various conditions, enabling

manufacturers to choose different measurement methods while assuring conformance with the result under a range of basic performance.

The standard does not specify PMU performance requirements in transient conditions. Taking this into account, PMUs having different estimation algorithms implemented may differ considering their outcomes. The standard suggests benchmark tests in order to test the influence of the transients. Annex G of the IEEE C37.118 Standard describes the benchmark tests to evaluate transient effects concerning the estimation precision of the algorithms implemented by a PMU. In the tests described, there is a 10% step in magnitude, a 90° step in phase and a 5 Hz step in frequency in a pure sine wave form.

### 3.2. Results using synthetic data

Some results using the described benchmark tests as an input for the implemented algorithms are shown in this section. The algorithms have a sampling rate of 32 samples per cycle of the fundamental frequency. Additionally, a Gaussian white noise with zero mean and Signal Noise Ratio (SNR) of 60 dB was added to the input signals to make it closer to actual data.



3.2.1. Magnitude step

The definition of the waveform for the magnitude step test is presented in the following equation:

$$\begin{aligned} X(t < 0) &= X_{m1} \cos(\omega_0 t) \\ X(0) &= (X_{m1} + X_{m2})/2 \cos(\omega_0 t) \\ X(t > 0) &= X_{m2} \cos(\omega_0 t) \end{aligned} \tag{3}$$

where  $X_{m1} = 0,9X_{m2}$ .

Fig. 3 shows the outputs for the implemented algorithms (GA, PLL and DFT) considering the magnitude step test. Fig. 3a shows the TVE for the three algorithms with the magnitude step at  $t = 0$ . Fig. 3b and c shows the magnitude and frequency estimation respectively for the 10% magnitude step test.

The results show that the three algorithms worked adequately concerning the steady state output. However, the GA holds the fastest recovering time (period that the TVE remained in a value above 1%) after the magnitude step. The DFT based algorithm had the second fastest recovering time, followed by the PLL based algorithm.

3.2.2. Phase step

The definition of the waveform for the phase step test is presented in the following equation:

$$\begin{aligned} X(t < 0) &= X_m \cos(\omega_0 t) \\ X(0) &= X_m \cos(\omega_0 t + \omega/4) \\ X(0) &= X_m \cos(\omega_0 t + \omega/2) \end{aligned} \tag{4}$$

Fig. 4 shows the outputs for the implemented algorithms (GA, PLL and DFT) considering the phase step test. Fig. 4a shows the TVE for the three algorithms with the phase step at  $t = 0$ . Fig. 4b and c shows the magnitude and frequency estimation, respectively, for the 90° phase step test.

The results are similar to the ones presented before and show that the genetic algorithm holds the fastest recovering time after the phase step.

3.2.3. Frequency step

The definition of the waveform for the frequency step test is presented in the following equation:

$$\begin{aligned} X(t < 0) &= X_m \cos(\omega_0 t) \\ X(0) &= X_m \\ X(0) &= X_m \cos[2\pi(f_0 + 5 \text{ Hz})t] \end{aligned} \tag{5}$$

Fig. 5 shows the outputs for the implemented algorithms (GA, PLL and DFT) considering the frequency step test. Fig. 5a shows the TVE for the three algorithms with the 5 Hz frequency step at  $t = 0$ . Fig. 5b and c shows the magnitude and frequency estimation, respectively, for the frequency step test. Similarly to the other cases, the GA has the fastest recovering time. The DFT based algorithm holds the second fastest (the first crossing to the 1% zone was considered concerning recovering time). It must be emphasized that a simple averaging output filter was used in this case to improve the DFT behavior when facing non-nominal frequency input waveforms.

**Table 2**  
Recovering time for the implemented algorithms.

Benchmark test	Algorithm recovering time (s)		
	GA	PLL	DFT
Magnitude step	0.001042	0.026563	0.019271
Phase step	0.011929	0.064583	0.020833
Frequency step	0.008854	0.078646	0.021751

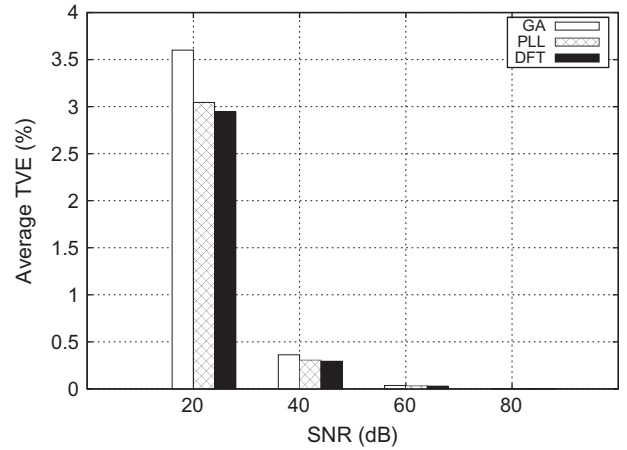


Fig. 6. Noise influence in a steady state signal.

3.2.4. Recovering time

Table 2 summarizes the results of the performed tests, presenting the time required by each algorithm to return to a reliable level (TVE below 1%) after undergoing the presented transient events.

Analyzing Table 2, one can note that the severest event was the frequency step test, with the highest recovering time. The GA presented the best performance among the tested algorithms, with the shortest recovering times for all the tests.

3.2.5. Noise influence

In order to test the noise influence in the estimation precision of the implemented algorithms, tests were performed adding other levels of white Gaussian noise (SNR of 20, 40, 60 and 80 dB) to pure sinusoids with the duration of 2 s, without the presence of transient events.

Fig. 6 presents the input signal noise's influence on the implemented algorithms (GA, PLL, DFT) concerning the estimation precision using TVE for the different levels of noise mentioned before. As expected, Fig. 6 shows that input signals with a lower SNR causes a higher precision error. However, it can be seen that only signals with a 20 dB SNR have a TVE higher than 1%, regardless of the algorithm being used. For higher SNR values, the precisions of the three algorithms are below 1% of the TVE.

3.3. GA implemented in FPGA for real-time estimations

As detailed in Section 2, GAs are known as computationally complex techniques since they require the generation, evaluation

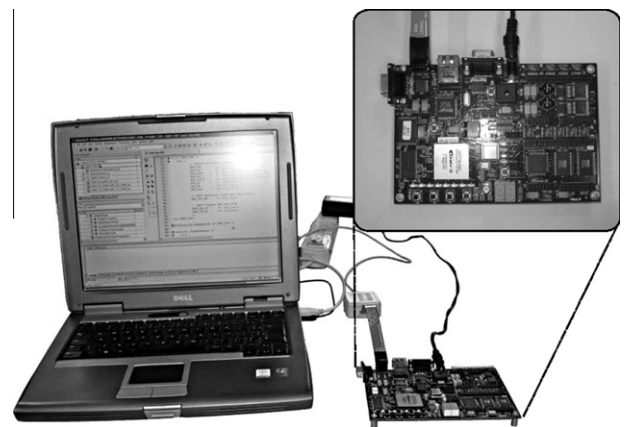


Fig. 7. Implemented FPGA hardware for the architecture of the proposed GA.



and comparison of several potential solutions at each iteration. On the other hand, GAs have an inherent parallelism that is often hidden by codes for sequential processing developed for the usual computer architectures. The hardware designers, using the FPGAs available nowadays, can implement a relatively large variety of architectures. This generates flexibility to explore parallel computer solutions using FPGAs.

In this paper, the parallelism of GAs was explored for frequency estimation using an FPGA [23]. The proposed GA effectively generates several individuals (potential solutions) in parallel, significantly reducing the computing time. Moreover, the processing of

evaluation of a solution is also parallelized. Note that the evaluation is in general the most critical point in terms of efficiency of a GA. For frequency estimation, the computing time for the evaluation increases linearly with the number of signal samples. The running time is reduced to a logarithmic rate when the evaluation is parallelized.

The details of how to parallelize a GA for frequency estimation is described in [22–24]. The resulting hardware is much faster than the sequential GA so that a frequency estimation requires less than a millisecond, enabling its application in real-time. Fig. 7 shows the FPGA infrastructure employed to develop the parallelized GA,

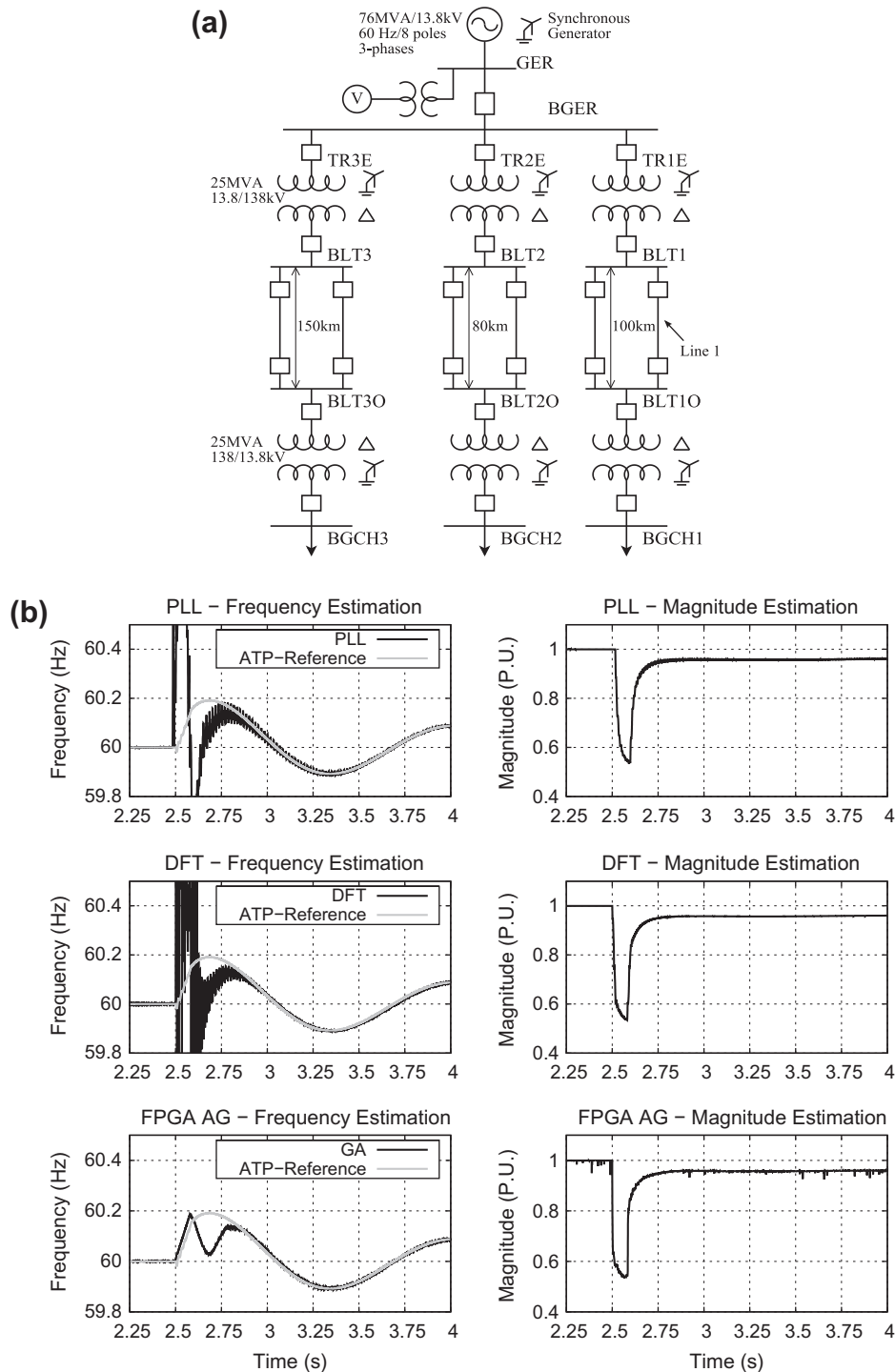


Fig. 8. Example of an event in a power system and the corresponding estimations performed by the DFT, PLL and GA techniques: (a) Power system simulated. (b) Magnitude and frequency estimations for a permanent fault at 50% of line 1.

using Quartus II software from Altera [31]. Moreover, this GA is tested in real-time using signals generated from a simulated electrical power system during transient events.

The computer simulations were performed using the ATP software. Fig. 8a presents the power system used. The EPS consists of a 13.8 kV and 76 MVA synchronous generator, 138:13.8 kV three phase power transformers of 25 MVA, transmission lines between 80 and 150 km in length and loads between 5 and 25 MVA with a power factor  $\text{fp} = 0.92$  inductive. The connections of power transformers are delta and star, respectively, for the high and low voltage winding. Detailed specifications of the electrical power system simulated and the modeling of its elements can be found in [24].

The sampled voltage signal at bus BLT1 of the power system is the input for the proposed GA. Fig. 8b presents estimations for magnitude and frequency, respectively, found by the GA considering a permanent three phase-to-ground fault at 50% of line 1 at  $t = 2.5$  s. The ATP-Reference frequency is shown to guide the evaluation of the frequency estimations. Fig. 8b also provides DFT and PLL estimations for the same input signal. The figure shows a very precise magnitude and frequency estimation considering the GA approach during transient events.

#### 4. Conclusion

This work presented a methodology based on GAs applied to real-time phasor estimation and frequency tracking. The precision and response time of the GAs were evaluated using synthetic data generated according to the proposed benchmark tests described in the IEEE C37.118 standard. In order to compare the performance with traditional methods, DFT and PLL based methods were implemented. Results showed that GAs have the fastest response time when compared to other implemented methods.

In addition, a GA was implemented in FPGA in order to take advantage of the intrinsic parallelism of this kind of algorithm, enabling the real-time processing of the signal. An EPS was simulated in the ATP software to provide data very close to real situations. The results were compared to a frequency reference provided by the ATP software. They show that GAs have a good precision concerning the estimations provided with the fastest response time when compared to the conventional algorithms for the same purpose. It is well known that the DFT is a widely used method with vast application in PMU equipment. This paper offers an alternative method that is faster facing transient events, it is immune against small frequency variations and responds well in the presence of noise.

The paper also demonstrates that although GAs demand a bigger computational burden if compared to the other methods and involves a convergence process, they can be designed and implemented in an adequate FPGA hardware prototype for PMUs considering real-time measurement purposes.

#### Acknowledgments

The authors would like to acknowledge CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) and FAPESP (Fundação para Amparo a Pesquisa do Estado de São Paulo) for financial support as well as the University of São Paulo (USP) in São Carlos, SP, Brazil, for the research facilities.

#### References

- [1] Phadke A. Synchronized phasor measurements – a historical overview. IEEE/PES transmission and distribution conference and exhibition, vol. 1. Asia Pacific: IEEE; 2002. p. 476–9.
- [2] Martin K, Hamai D, Adamiak M, Anderson S, Begovic M, Benmouyal G, et al. Exploring the IEEE standard C37.118-2005 synchrophasors for power systems. IEEE Trans Power Delivery 2008;23(4):1805–11.
- [3] Hart D, Uy D, Gharpure V, Novosel D, Karlsson D, Kaba M. PMUs – a new approach to power network monitoring. ABB Rev 2001:58–61.
- [4] Burnett Jr R, Butts M, Sterlina P. Power system applications for phasor measurement units. IEEE Comput Appl Power 1994;7(1):8–13.
- [5] IEEE-Std.1344. IEEE standard for synchrophasors for power systems.
- [6] IEEE-Std.C37.118. IEEE standard for synchrophasors for power systems.
- [7] Phadke A, Kasztenny B. Synchronized phasor and frequency measurement under transient conditions. IEEE Trans Power Delivery 2009;24(1):89–95.
- [8] Wang M, Sun Y. A practical method to improve phasor and power measurement accuracy of DFT algorithm. IEEE Trans Power Delivery 2006;21(3):1054–62.
- [9] Wang M, Sun Y. A practical, precise method for frequency tracking and phasor estimation. IEEE Trans Power Deliv 2004;19(4):1547–52.
- [10] Phadke A, Thorp J, Adamiak M. A new measurement technique for tracking voltage phasors, local system frequency, and rate of change of frequency. IEEE Trans Power Apparatus Syst 1983;5:1025–38.
- [11] Hart D, Novosel D, Hu Y, Smith B, Egolf M. A new frequency tracking and phasor estimation algorithm for generator protection. IEEE Trans Power Deliv 1997;12(3):1064–73.
- [12] Benmouyal G. An adaptive sampling-interval generator for digital relaying. IEEE Trans Power Delivery 1989;4(3):1602–9.
- [13] Begovic M, Djuric P, Dunlap S, Phadke A. Frequency tracking in power networks in the presence of harmonics. IEEE Trans Power Delivery 1993;8(2):480–6.
- [14] Sachdev M, Giray M. A least error squares technique for determining power system frequency. IEEE Trans Power Apparatus Syst 1985;2:437–44.
- [15] Sachdev M, Baribeau M. A new algorithm for digital impedance relays. IEEE Trans Power Apparatus Syst 1979;6:2232–40.
- [16] Ziarani A, Konrad A. A method of extraction of nonstationary sinusoids. Signal Process 2004;84(8):1323–46.
- [17] Karimi-Ghartemani M, Iravani M. Robust and frequency-adaptive measurement of peak value. IEEE Trans Power Deliv 2004;19(2):481–9.
- [18] Dash P, Jena R, Panda G, Routray A. An extended complex Kalman filter for frequency measurement of distorted signals. IEEE Trans Instrum Meas 2000;49(4):746–53.
- [19] Routray A, Pradhan A, Rao K. A novel Kalman filter for frequency estimation of distorted signals in power systems. IEEE Trans Instrum Meas 2002;51(3):469–79.
- [20] Burnett Jr R, Butts M, Cease T, Centeno V, Michel G, Murphy R, et al. Synchronized phasor measurements of a power system event. IEEE Trans Power Syst 1994;9(3):1643–50.
- [21] El-Naggar K, Youssef H. A genetic based algorithm for frequency-relaying applications. Electr Power Syst Res 2000;55(3):173–8.
- [22] Souza S, Oleskovicz M, Coury D, Silva T, Delbem A, Simoes E. FPGA implementation of genetic algorithms for frequency estimation in power systems. In: Power and Energy Society general meeting. Pittsburgh (USA): IEEE; 2008. p. 1–6.
- [23] Souza S, Oleskovicz M, Coury D, Silva T, Delbem A, Simoes E. An efficient frequency estimation methodology using genetic algorithms in FPGA. In: IEEE international conference on industrial electronics – IECON. Taipei (Taiwan): IEEE; 2007. p. 2020–5.
- [24] Coury D, Oleskovicz M, Delbem A, Simões E, Silva T, de Carvalho J, et al. A genetic based algorithm for frequency relaying using FPGAs. In: Power and Energy Society general meeting. Calgary (Canada): IEEE; 2009. p. 1–8.
- [25] Back T, Fogel DB, Michalewicz Z, editors. Handbook of evolutionary computation. Bristol (UK): IOP Publishing Ltd.; 1997.
- [26] Jong KAD. Evolutionary computation: a unified approach. MIT Press; 2006.
- [27] Holland JH. Genetic algorithms and the optimal allocation of trials. SIAM J Comput 1973;2(2):88–105.
- [28] Goldberg DE. Genetic algorithms in search, optimization and machine learning. 1st ed. Boston (MA, USA): Addison-Wesley Publishing Company; 1989.
- [29] Alternative transients program rule book. Leuven EMTP Center (LEC); 1987.
- [30] Brown S, Vranesic Z, editors. Fundamentals of digital logic with vhdl design. Boston (MA, USA): McGraw-Hill; 2008.
- [31] Altera. Stratix IV device handbook: volume 1. 101 Innovation Drive, San Jose, CA 95134, 2011. <<http://www.altera.com>>.