

Frequency Relaying based on Genetic Algorithm using FPGAs

D. V. Coury, M. Oleskovicz, A. C. B. Delbem, E. V. Simões, T. V. Silva, J. R. de Carvalho, D. Barbosa

University of São Paulo- Av. Trabalhador Sancarlense, 400, São Carlos, São Paulo, Brazil

Abstract—This work presents an accurate and precise Genetic Algorithm (GA) for frequency estimation of Electrical Power System (EPS) signals. The problem of estimating the frequency of a distorted electrical signal is modeled as an optimization problem. The advantages of GAs in this approach include the use of coding for a number of solutions which facilitates computer implementation, as well as the search for an appropriate solution from a population of possible solutions. The GA is programmed in a FPGA (*Field-Programmable Gate Array*) device and the estimation procedure is performed in real-time. This is made possible due to (a) the implicit parallelism of FPGAs in computing their instructions, (b) the suitable choice of steps of GAs to explore this parallelism and (c) the appropriate choice of FPGA for good performance. To evaluate the performance of the proposed method, an EPS was simulated having typical operation conditions. The resulting signals were analyzed by the proposed GA-FPGA approach and promising results were compared to a commercial relay.

Index Terms—Frequency estimation, Genetic algorithms, Field-programmable gate arrays, Digital relaying, Power system protection.

I. INTRODUCTION

REAL-TIME estimation of Electrical Power System (EPS) frequency is an important task in many fields of Electrical Engineering. Digital frequency relays use the frequency estimation to make decisions and protect the EPS components against loss of synchronism, under- and over-frequency. Accurate frequency estimation is also essential for stability of the EPS, since the dynamic balance between generation and load, a prerequisite for stable operation, has become more difficult to maintain considering the large expansion of electrical systems. The increasing interest in Power Quality (PQ) has also stimulated researchers to find new tools and methods to estimate the instantaneous frequency accurately.

Several researchers have proposed different techniques to solve the problem of frequency estimation. Two algorithms based on phase errors using Discrete-Fourier Transform (DFT)

are presented in [1] and [2]. References [3] and [4] present complex frequency estimation methods based on the Kalman filtering approach. In [5], [6] and [7] the authors present methods obtained by modeling the estimation task as optimization problems. The Newton algorithm is used to solve the problem in [5]. The steepest descent method is used in [6] and [7] and the resulting sets of non-linear equations are called EPLL (*Enhanced Phase-Locked Loop*). An algorithm based on the Least Error Squares (LES) is derived in [8] taking into account some coefficients of the Taylor's expansion series for the model of the input signal. An adaptive filtering technique, the Least Mean Square (LMS) algorithm, is presented in [9]. Intelligent techniques have also been used for frequency estimation of EPS signals [10]-[13]. Reference [10] presents an Artificial Neural Network (ANNs) based approach. In [11]-[13], Genetic Algorithms (GAs) are the tools used to solve the optimization problem considered.

This paper presents an efficient method based on GAs for frequency estimation in an electrical signal. A sliding window with the samples from an input signal is used to adjust to a pure sinusoid wave. The main contribution of this paper, compared to previous work [12]-[13], is the development of the prototype relay working in real-time. This experience was possible by optimizing the GA operations for hardware implementation, as described later. The paper presents the following structure. The proposed GA-FPGA approach is presented in Section II. The simulated EPS to obtain the waveforms for analysis is described in Section III. Section IV. presents the simulated cases and results of frequency estimation compared to results of a commercial frequency relay (function 81). Finally, Section V. presents the concluding notes.

II. THE PROPOSED GA-FPGA APPROACH

A. Genetics Algorithms

Genetic algorithms are adaptive search algorithms for optimization problems. Their mechanism of searching for the optimum solution is heuristic, inspired by natural selection and population genetics [14]. These algorithms operate in a population of possible solutions (the individuals) for the problem, with random initialization in most cases. As the population evolves, its characteristics change using genetic operators and the improvement of possible solutions can be reached. The individuals are encoded as strings (the chromosomes), normally as sequences of the binary digits "0" and "1".

This work was supported by CNPq from Brazil.

D. V. Coury, M. Oleskovicz, J. Carvalho and D. Barbosa are with Department of Electrical Engineering, Engineering School of São Carlos, from University of São Paulo (USP), São Carlos – SP, Brazil, CEP 13.566-590. Phone: +55 (16) 3373-8133, (emails: coury@sc.usp.br, olesk@sc.usp.br, janison@sc.usp.br, dbarbosa@usp.br).

A. Delbem, E. Simoes and T. Silva are with Institute of Mathematical and Computer Sciences, from University of São Paulo (USP) (email: acbd@icmc.usp.br, simoes@icmc.usp.br, tiagovs@grad.icmc.usp.br).

Basically, the fundamental principle of a GA is that “the fittest member of a population has the highest probability for survival”. Thus, it is important to evaluate each possible solution. This evaluation is made by associating a *fitness* value to each individual, computed with the cost function defined for the particular problem. Higher values for fitness indicate better chromosomes. The corresponding individuals have a larger probability to survive and produce offspring for a new generation.

The main genetic operators used to produce the new generation are crossover and mutation. These operators are responsible for establishing how individuals will exchange or simply change their genetic features in order to produce new individuals. A general structure of a genetic algorithm is shown below, where t is the generation index and $P(t)$ is the population at generation t .

- $t \leftarrow 0$;
- Initialize $P(t)$ randomly
- Evaluate $P(t)$
- while (stopping criteria is false)
 - Apply selection to $P(t)$ to generate P_{AUX}
 - Apply crossover and mutation to P_{AUX}
 - $P(t+1) \leftarrow P_{AUX}$
 - Evaluate $P(t+1)$
 - $t \leftarrow t+1$

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

B. The Optimization Problem and the Proposed GA

The optimization problem using GAs to estimate the frequency of an EPS is defined considering a sinusoidal model for the voltage signal. The analysis of the electric signal can be performed considering a sliding window as shown in Fig. 1, to determine system dynamics. This figure exemplifies the process with the use of a window of a length equal to one-cycle. Mathematically, the cost function value at time instant n is defined as:

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

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 approximate the input signal [11].

The process of finding an optimum solution to minimize (1) is exemplified in Fig. 2. Here, a workstation executed the GA using a one-cycle window of an electrical signal with

amplitude, frequency and phase equal to 1.0 pu, 60 Hz and 4.1888 radians, respectively. Fig. 2(a) shows the initial random population and, after 20 generations, the process of convergence of the algorithm can be observed in Fig. 2(b). The best individual of each population is indicated by arrows and the approximation to the actual values of parameters in proportion to the increase of generations can be observed.

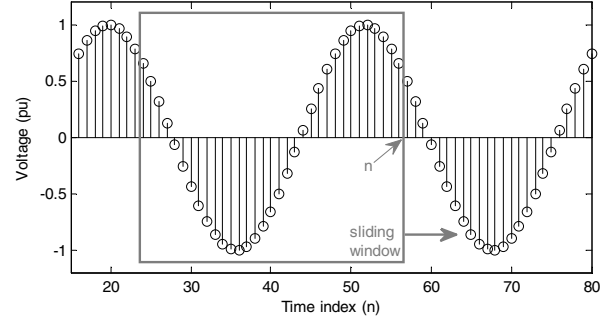


Fig. 1 Sliding window under analysis at time n

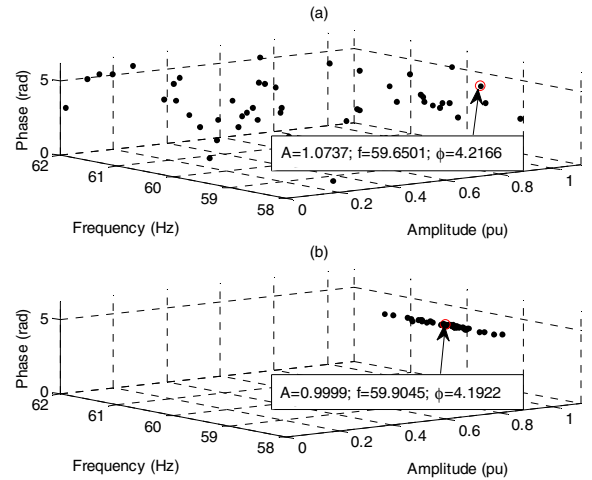


Fig. 2. Population of a GA at (a) first generation and, (b) twentieth generation

1) Encoding

A binary code scheme is used to represent the parameters of the set $\{A, f, \theta\}$. Each individual has three sequences of bits corresponding to three integer numbers as shown in Fig. 3.

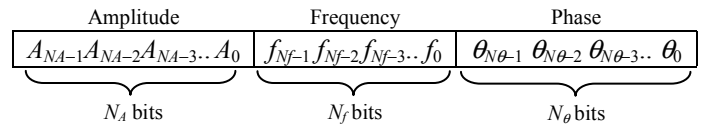


Fig. 3. Individual representation

Assuming a normalized input signal and a limit of ± 2 Hz for the variation of frequency, the search space is shown in Table I, as well as the number of bits to represent each parameter.

Parameter	Number of Bits	Range	Unit
A	8	[0.75, 1.0]	pu
F	24	[58.0 62.0]	Hz
θ	12	[0.0, 2π]	Rad

2) Selection

The selection process consists of randomly choosing individuals for reproduction. The tournament operator [14] is used at this stage. This procedure has been reported to give adequate results for several application domains. The selection of two parents is carried out according to the algorithm below. Four individuals $\{a,b,c,d\}$ are randomly chosen from the current population. The parents p_1 and p_2 are results of competition between the pairs $\{a,b\}$ and $\{c,d\}$, respectively.

- random selection of four individuals $\{a,b,c,d\}$;
- $p_1 \leftarrow$ best between $\{a,b\}$ (comparing their fitness)
- $p_2 \leftarrow$ best between $\{c,d\}$ (comparing their fitness)

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 p_1 and p_2 is performed as represented generically in Fig. 4, where $\psi = \{A, f, \theta\}$. Each parameter ψ is combined as shown in this figure that is, computing the mean value between ψ_{p_1} and ψ_{p_2} and distance δ between them. The chosen parameter for the offspring is randomly determined from the five possible values presented. This procedure leads to a 20% of probability in choosing each value and is followed three times (one time for each $\psi = \{A, f, \theta\}$).

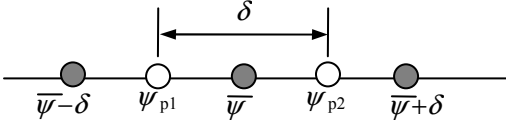


Fig. 4. The crossover operator

4) Mutation

The mutation is the genetic operator responsible to generate 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 1 (or subtract) to the parameter.

C. FPGA Implementation Details

From the previous GA theory presented, it can be easily concluded that the large number of mathematical operations required creates a barrier that makes the use of GAs difficult in real-time systems. For example, consider a monitored system with a sampling period of 1ms and a common FPGA device operating with 25 MHz. All steps of the genetic algorithm must be performed before acquiring the next sample. It means that 20,000 clock pulses are available to obtain the optimum solution. It is not difficult to conclude that pure sequential processing is not able to solve this problem. However, the intrinsic parallel processing capacity of FPGAs makes GA implementation possible in hardware in order to build real-time measurement systems.

Even though FPGAs are appropriate to implement computational complex methodologies, some strategies must be adopted to adapt the algorithm to the specific hardware limitations. More specifically, it is highly desirable that only simple algebraic (adding and subtracting) and logical (AND, OR and NOT) operations have to be executed. Another simple operation for hardware implementation is to shift the bits of a binary sequence to the left or the right. These operations are intrinsic of hardware architecture, using a few number of clock pulses to be executed. The next three topics explain aspects of hardware implementation for the system developed in this research.

1) The sine function

The sine function requires attention because of the variation of index k as observed in (1). Multiplications are highly undesirable and these operations can be reduced since a multiplication by a factor of 2 is equivalent to a shift of one bit to the left in a binary representation. Thus, the operations can be optimized and, for example, the term $2\pi k/T$ with $k = 6$ is quickly computed shifting the term with $k = 3$ one bit to the left.

The sine function is another important issue in the optimization of the algorithm. One solution is to consider a few terms of power series for this function. However, this strategy implies an increase in multiplications approximately in proportion to the square of the number of terms used. We adopted the strategy of using a look-up table, easily configured into the FPGA internal memory. The running time to evaluate the fitness function is largely reduced by computing the sine function *a priori* in a workstation for a large range of values: 1,024 points.

2) Random numbers

Another relevant aspect of the proposed GA design in a FPGA is the generation of enough adequate random numbers. These numbers are the running basis of selection, crossover and mutation steps of the GA, as seen before. The literature presents various random number generators developed for FPGAs. We have chosen the construction of a ROM-based table to store random numbers calculated off-line in a workstation. This ROM memory should have enough numbers for a complete run of the proposed GA, but this means it would need much more memory that is commonly available in FPGAs. The ROM was then implemented as a circular table in order to ensure a supply of random numbers for any number of generations. Moreover, different GA behaviors can be obtained by initializing a sequence of random numbers from different positions of the table.

3) Generating a population

The production of d new individuals from a population can be basically obtained by generating p copies of the circuit designed to produce one new individual. The parallelization of the generation of new individuals is relatively easy as each individual is an array of bits and the circuit blocks for bit modifications by the crossover and mutation operators are

relatively small.

In fact, it is worth mentioning that operations related to crossover and mutation are optimized from a practical point of view. The crossover is carried out by some additions and a shift to the left to execute the division by 2 and obtain the mean value represented in Fig. 4. The mutation is carried out with a few additions, according to the mutation rate. Moreover, in terms of hardware implementation, the selection using the tournament is clearly advantageous compared with the roulette.

Additionally, optimization of the FPGA memory is obtained by using only 2 populations, the current one and the next one. The GA operations are carried out according to the random selection of individuals (possible parents) from the current population. Only the offspring is stored in the next population. This population then becomes the current population. This process is repeated until the maximum numbers of generations are reached. Finally, an elitism criterion is adopted to accelerate the process of convergence. That is, the first individual of the next population is always the best one of the current population.

D. Determining an Appropriate FPGA

An efficient GA in FPGA for frequency estimation requires a balance between estimation quality and number of logic gates to support parallelization of the genetic operators. A direct parallelization of all the operators, as usually programmed in software, would require a number of gates that is not available in any FPGA. The viability of the proposed approach for FPGAs in market was then investigated. Table II shows the required characteristics of the developed GA. Table III presents the characteristics of FPGAs, F1, F2 and F3. The number of individuals created in parallel is p . From the running time point of view, the ideal solution is $p=30$, i.e. all individuals from the population generated in parallel. However, this strategy requires a very large FPGA and, consequently, it increases the price of the proposed approach. From Table II, one can conclude that the construction of a FPGA-based relay that is fast enough for frequency estimation purposes is possible: FPGA F1 requires less than $0.2ms$ to process one data window (see column for $p=2$ in Table II) and it uses an inexpensive FPGA.

TABLE II
GA REQUIREMENTS AND ADEQUATE FPGA CIRCUITS

Characteristic	GA Requirements		
	$p = 1$	$p = 2$	$p = 4$
Number of combinatorial functions	5,339	9,205	9,192
Memory bits	404,736	408,064	305,664
Registers	2,989	3,032	3,031
DSPs	16	32	32
GA performance (ms)*	0.355	0,177	0.089
Chosen FPGA**	F1	F1	F2
FPGA price	US\$995	US\$995	US\$ 1,995

* – Milli-seconds to produce an adequate estimation of frequency for one data window, where adequate means percentage of frequency error below 0.089.

** – F2 is used if F1 can not support the parallelism indicated by p .

TABLE III
CHARACTERISTICS OF FPGAs F1, F2 AND F3

FPGA	F1 / ALTERA ep2S60f672c3	F2 / ALTERA ep2s180f1508c5	F3 / ALTERA ep4sgx230kf40c2n
LE	60k	180K	680k
Pins	493	1,171	1,517
Memory bits	2,5M	9,3M	22,4M
DSP	288 (9x9 bits)	768 (9x9 bits)	1,360 (18x18 bits)
Price	US\$995	US\$ \$1,995	US\$ 4,495

III. THE SIMULATED POWER SYSTEM

The electrical power system of Fig. 5 is used to generate typical electrical signals with frequency variations, such as in the case of sudden load connections (disconnections). The computer simulations were performed using the ATP (Alternative Transient Program) software [15].

The EPS consists of a 13.8 kV and 76 MVA synchronous generator, 13.8:138 kV and 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 $f_p=0.92$ inductive. The connections of power transformers are delta and star, respectively, for the high and low voltage winding. The synchronous generator and power transformers parameters used for simulation using the ATP are shown, respectively, in Table II and Table III.

For the synchronous generator (Table II), S is the total-phase volt-ampère rating of the machine, N_p is the number of poles which characterizes the machine rotor, V_L is the rate line-to-line voltage of the machine, f is the electrical frequency of the generator, IFD is the field current, R_a is the armature resistance, X_l is the armature leakage reactance, X_0 is the zero-sequence reactance, X_d is the direct-axis synchronous reactance, X_q is the quadrature-axis synchronous reactance, X_d' is the direct-axis transient reactance, X_d'' is the direct-axis subtransient reactance, X_q'' is the quadrature-axis subtransient reactance, τ_{do}' is the direct-axis open-circuit transient time constant, τ_{do}'' is the direct-axis open-circuit subtransient time constant and τ_{qo}' is the quadrature-axis open-circuit transient time constant.

It must be emphasized that the transmission line model used was JMARTI from ATP. This model makes the variation of the line parameters in function of the frequency possible. Consequently, a better representation is obtained for the system's behavior when facing disturbances resulted from unbalance between generation and load.

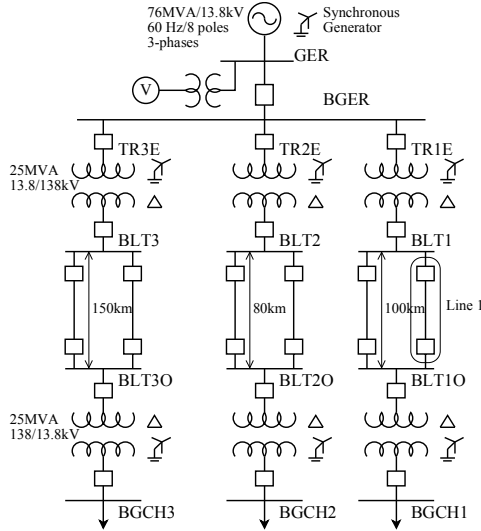


Fig. 5. The power system simulated

Another important emphasis is placed on the simulation of the synchronous generator, with an automatic speed control for hydraulic systems [16] and automatic voltage regulation (AVR) [17]-[19], considering the electrical and mechanical parameters of the generator. The transfer function of the speed regulator is given by

$$\frac{\eta(s)}{\Delta F(s)} = \frac{1}{R} \cdot \frac{1 + sT_r}{(1 + sT_g) \left(1 + s\frac{r}{R}T_r\right)} \quad (2)$$

where $\eta(s)$ is the servomotor position, $\Delta F(s)$ is the frequency deviation, R is the steady-state speed drop, T_g is the main gate servomotor time constant and T_r is the reset time. Table IV presents the parameters concerning the speed regulator.

The excitation control system of the generator is shown in Fig. 6 in flow chart representation. The basic function of the excitation control system automatically adjusts the magnitude of the DC field current of the synchronous generator to maintain the terminal voltage constant as the output varies according to the generator capacity [20].

The field voltage control can improve the transient stability of the power system after a major disturbance. However, the extent of the field voltage output is limited by the exciter's ceiling voltage, which is restricted by generator rotor insulation [20]-[21].

TABLE II
THE SYNCHRONOUS GENERATOR SIMULATION DATA

Parameter	Value (Unit)	Parameter	Value (Unit)
S	76 (MVA)	N_p	8
V_L	13.8 (kV _{RMS})	f	60 (Hz)
IFD	250 (A)	R_a	0.004 (p.u)
X_l	0.175 (p.u)	X_0	0.132 (p.u)
X_d	1.150 (p.u)	X_q	0.685 (p.u)
X_d'	0.310 (p.u)	X_d''	0.210 (p.u)
X_q''	0.182 (p.u)	τ_{do}'	5.850 (p.u)
τ_{do}''	0.036 (p.u)	τ_{qo}'	0.073 (p.u)

TABLE III
THE POWER TRANSFORMER SIMULATION DATA

Element	$R_+(\Omega)$	$L_+(mH)$
Primary Impedance	1.7462	151.37
Secondary Impedance	0.0175	1.514

TABLE IV
PARAMETERS CONCERNING THE SPEED REGULATOR

Description	Value (Unit)
Main gate servomotor time constant (T_g)	0.600 (sec)
Reset time (T_r)	0.838 (sec)
Transient speed drop (r)	0.279
Steady-state speed drop (R)	0.100
Moment of inertia (M)	1.344 (sec)
Water starting constant (T_w)	0.150 (sec)

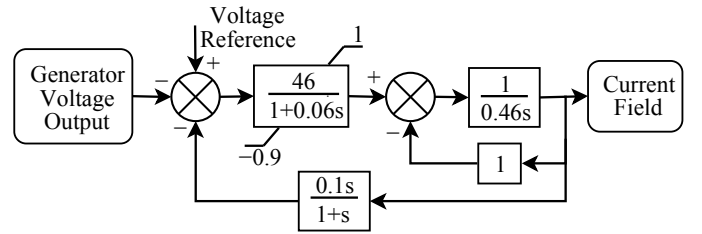


Fig. 6. Flow chart of the excitation control system.

IV. RESULTS AND DISCUSSION

This section presents results of the proposed scheme. Three cases of abnormal operation are simulated for the electrical system of Fig. 5. Each condition imposes a particular dynamic behavior in the power balance and, consequently, in the variation of the power system frequency. Measurements from a commercial relay (function 81) were obtained by using the simulated voltage signals from ATP with a signal emulator, in order to compare the results. Moreover, the actual frequency of the EPS, was measured directly from the angular speed of the synchronous generator. Concerning the proposed methodology, the experiment was done by connecting the FPGA device to a computer by a serial port. A dedicated PC sends a sample of the normalized voltage signal under analysis with a sampling interval of 1.3 ms, simulating the system connected to a real EPS by A/D converters, potential transformers and analog filters (anti-aliasing and others). In fact, the simulations performed by the ATP for the EPS (Fig. 5) used a sampling rate of 10 kHz and a digital filter was necessary to prevent aliasing. Thus, a 2nd-order Butterworth filter, with cutoff frequency of 200 Hz, is included in the simulation before the down-sampling operation with factor 13. Another 2nd-order Butterworth filter, with cutoff frequency of 5 Hz, is inserted at the output to smooth the frequency estimation. The GA runs with a data window of 15 samples, 30 individuals per population and the stop criteria is the number of 300 generations. The FPGA device is an *Altera Stratix II*. Some of the situations analyzed are presented in the next topics.

A. A sudden connection of load Blocks

In this case, load blocks are connected in the BCGH3 busbar at time $t=2$ s. The frequency before the switching is stabilized at the nominal frequency of 60 Hz. After the load insertion, the frequency oscillates as shown in Fig. 7. This figure shows the reference and the results of the AG-FPGA proposed structure and the commercial relay. Estimations are within the 0.2% error band and they follow the variation of frequency occurred in the test. The mean-squared errors in this case are 3.75×10^{-4} and 2.43×10^{-4} , for the AG-FPGA method and commercial relay, respectively.

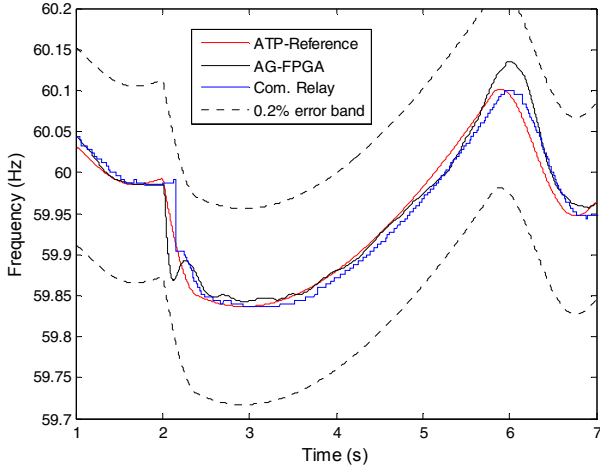


Fig. 7. Frequency estimation: the case of a sudden connection of load blocks.

B. Energization of TR1E Transformer

In this case, the analyzed signal is obtained during the energization of TR1E transformer, initiated at time $t=1$ s. The frequency decreases as shown by the ATP reference in Fig. 8. Furthermore, this figure shows the results of the AG-FPGA approach as well as the commercial relay. Both estimations follow the decrease in frequency with time delays (mean values) of 142.2 and 104.5 milliseconds (or approximately 8 and 6 cycles). The mean squared errors are 5.77×10^{-3} and 1.87×10^{-3} , for the AG-FPGA approach and the commercial relay, respectively.

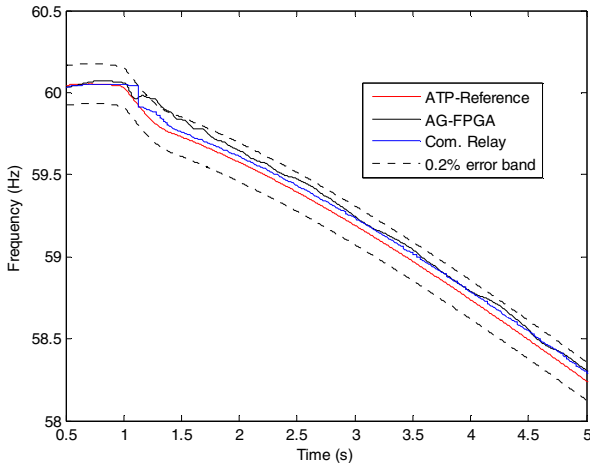


Fig. 8. Frequency estimation: the case of energization of the TR1E transformer

C. A sudden disconnection of TR1E and TR3E transformers

This case analyzes the impact of disconnecting TR1E and TR3E transformers, occurred at 1.0 s. From Fig. 9, the increase in the power system frequency can be seen. The AG-FPGA and the commercial relay estimations follow this increase with time delays (mean values) of 82.4 and 113.7 milliseconds (or approximately 5 and 7 cycles), respectively. The AG-FPGA estimation does not extrapolate the 0.2% error band and the mean squared errors are 2.56×10^{-3} and 4.15×10^{-3} , for the proposed structure and the commercial relay, respectively.

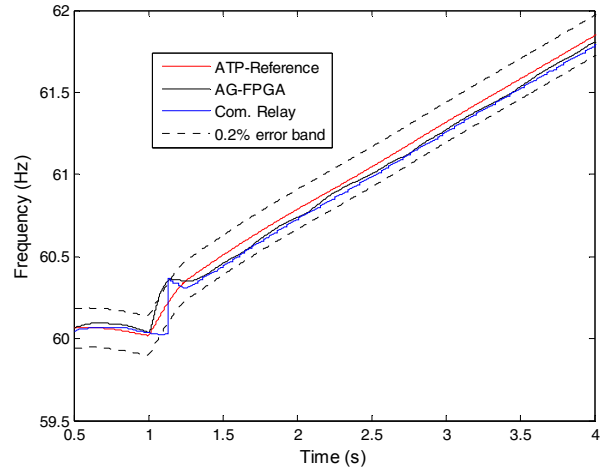


Fig. 9. Frequency estimation: the case of sudden disconnection of TR1E and TR3E transformers

V. CONCLUSIONS

This paper presented a real-time hardware implementation of GAs for frequency estimation using FPGA devices. The algorithm developed was carefully studied in order to optimize all mathematical operations for the hardware architecture. A complete Electrical Power System was simulated using ATP software in order to test the proposed scheme. Four cases of abnormal situations were used to test the relay performance. In general, it can be said that the performance obtained is equivalent to the commercial relay used for the comparison. The main contribution of this proposal is the development of a GA approach for frequency estimation implemented in hardware (FPGA) at a very small cost.

VI. REFERENCES

- [1] A. G. Phadke, J. S. Thorp, and M. G. Adamiak, "A new measurement technique for tracking voltage phasors, local system frequency, and rate of change of frequency," *IEEE Trans. Power App. Syst.*, vol. PAS-102, pp. 1025-1038, May 1983.
- [2] M. Wang, and Y. Sun, "A practical, precise method for frequency tracking and phasor estimation," *IEEE Trans. Power Del.*, vol. 19, pp. 1547-1552, Oct. 2004.
- [3] P. K. Dash, A. K. Pradhan, and G. Panda, "Frequency estimation of distorted power system signals using extended complex Kalman filter," *IEEE Trans. Power Del.*, vol. 14, pp. 761-766, July 1999.
- [4] A. Routray, S. K. Pradhan, and K. P. Rao, "A novel Kalman filter for frequency estimation of distorted signals in power systems," *IEEE Trans. Instrum. Meas.*, vol. 51, pp. 469-479, June 2002.

- [5] V. V. Terzija, M. B. Djuric, and B. D. Kovacevic, "Voltage phasor and local system frequency estimation using Newton type algorithm," *IEEE Trans. Power Del.*, vol. 9, pp. 1368-1374, July 1994.
- [6] A. K. Ziarani, and A. Konrad, "A method of extraction of nonstationary sinusoids," *Signal Processing*, vol. 84, pp. 1323-1346, Aug. 2004.
- [7] H. Karimi, M. Karimi-Ghartemani, and M. R. Iravani, "Estimation of frequency and its rate of change for applications in Power Systems," *IEEE Trans. Power Del.*, vol. 19, pp. 472-480, April 2004.
- [8] M. S. Sachdev, and M. M. Giray, "A least error squares technique for determining power system frequency," *IEEE Trans. Power App. Syst.*, vol. PAS-104, pp. 437-444, Feb. 1985.
- [9] A. K. Pradhan, A. Routray, and A. Basak, "Power system frequency estimation using least mean square techniques," *IEEE Trans. Power Del.*, vol. 20, pp. 1812-1816, July 2005.
- [10] P. K. Dash, D. P. Swain, A. Routray, and A. C. Liew, "An adaptive neural network approach for the estimation of power system frequency," *Electric Power Systems Research*, vol. 41, pp. 203-210, June 1997.
- [11] K. M. El-Naggar, and H. K. M. Youssed, "A genetic based algorithm for frequency-relaying applications," *Electric Power Systems Research*, vol. 55, pp. 173-178, 2000.
- [12] S. A. Souza, M. Oleskovicz, D. V. Coury, T. V. Silva, A. C. B. Delbem, and E. V. Simões, "FPGA implementation of genetic algorithms for frequency estimation in power systems," in *Proc. 2008 IEEE PES General Meeting*, pp. 1-6.
- [13] S. A. Souza, M. Oleskovicz, D. V. Coury, T. V. Silva, A. C. B. Delbem, and E. V. Simões, "An efficient frequency estimation methodology using genetic algorithms in FPGA," in *Proc. 2007 The 33rd Annual Conf. of the IEEE Industry Electronics Society*, pp. 2020-2025.
- [14] Goldberg D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.
- [15] EEUG, *Alternative Transients Program Rule Book*. LEC, 1987.
- [16] X. Vieira Filho, *Operation of Power System with Automatic Control*. Rio de Janeiro: Editora Campus Ltda, 1984.
- [17] D. C. Lee, Ed., *IEEE Recommended Practice for Excitation System Models for Power System Stability Studies (IEEE Std 421.5-1992)*. Energy Development and Power Generating Committee of the Power Engineering Society, 1992.
- [18] I. Boldea, *Synchronous Generators*. Boca Raton: CRC Press, 2006, ISBN: 084935725X.
- [19] V. Mukherjee and S. Ghoshal, "Intelligent particle swarm optimized fuzzy PID controller for AVR System," *Electric Power Systems Research*, vol. 77, pp. 1689-1698, Oct. 2007.
- [20] P. Kundur, *Power System Stability and Control*, ser. The EPRI power system engineering series, N. J. Balu and M. G. Lauby, Eds. New York: Mc Graw-Hill, 1994, ISBN: 007025958X.
- [21] J. S. K. Leung, D. J. Hill, and Y. Ni, "Global power system control using generator excitation, PSS, FACTS devices and capacitor switching," *Electric Power Systems Research*, vol. 27, pp. 448-464, 2005.

VII. BIOGRAPHIES

Denis V. Coury was born in Brazil in 1960. He received a B.Sc. degree in Electrical Engineering from the Federal University of Uberlândia, Brazil, in 1983, an M.Sc degree from EESC - University of São Paulo, Brazil in 1986

and a Ph.D. degree from Bath University, England in 1992. He worked for the Technological Research Institute (IPT), São Paulo, Brazil from 1985 to 1986. He joined the Department of Electrical Engineering, University of São Paulo, São Carlos, Brazil in 1986, where he is presently a Full Professor in the Power Systems Group. Prof. Coury spent his Sabbatical at Cornell University (USA) in 1999-2000. His areas of research interest are power system protection, power quality, as well as new techniques for power system control and protection, including the use of intelligent tools.

Mário Oleskovicz received a B.Sc. degree in Electrical Engineering from the Santa Catarina Federal University, Brazil in 1995. He received an M.Sc. and a Ph.D. degree in Power Systems from São Carlos Engineering School, University of São Paulo, Brazil in 1997 and 2001, respectively. Nowadays, he is an Assistant Professor at the Electrical Engineering Department, São Carlos Engineering School, University of São Paulo, São Carlos, SP, Brazil. His areas of research interest are Digital Power System Protection, Power Quality and Intelligent Systems Applied to Power Systems.

Alexandre C. B. Delbem received a Ph.D. degree from the University of Sao Paulo, Sao Paulo, Brazil in 2002. He became an Assistant Professor at the University of Sao Paulo in 2003. His main research interest is efficiency of computational solutions for real-world problems. His main areas of publication have been power systems, bioinformatics, routing and scheduling problems.

Eduardo V. Simões received a B.Sc. degree in Electrical Engineering from the Rio Grande do Sul Federal University, Brazil in 1990. He received an M.Sc. in Computer Science at the same University in 1994. He finished his Ph.D. degree in Electronic Engineering at Kent University, U.K. in 2001. Nowadays, he is an Assistant Professor at the University of São Paulo, São Carlos, SP, Brazil. His areas of research interest are: robotics, genetic algorithms and embedded systems.

Tiago V. da Silva is an undergraduate student in Computer Science at the University of Sao Paulo, Brazil. He is involved in a research program implementing Genetic Algorithms to FPGAs for real-time solutions.

Janison Rodrigues de Carvalho was born in Brazil on July 27, 1980. He received a B.Sc. degree from the Federal University of Viçosa in Electrical Engineering on May, 2006. He received an M.Sc. degree from Federal University of Juiz de Fora also in Electrical Engineering in February, 2008. He is now working towards the Ph.D. degree at the University of São Paulo (USP), Engineering School of São Carlos- SP. His interests include digital signal processing, and power quality.

Daniel Barbosa was born in Brazil in 1981. He received a B.Sc. degree in Electrical Engineering from Bahia Federal University, Brazil, in 2005, an M.Sc degree from EESC - University of São Paulo, Brazil. He is currently a Ph. D. student at the same institution. His main research interests are power system control and protection, frequency estimation, transformer protection and power quality.