# Efficient multi-objective optimizers by meta-heuristics for power system control

GHOURAF DJAMEL EDDINE[1] , NACERI ABDELLATIF[2]
Department of Electrical Engineering,
National Polytechnic School
[1]SCAMRE Laboratory, BP1523 EL M'naour, Oran 31000, ALGERIA
[2]IRECOM Laboratory, BP 98 22000 ALGERIA

*Abstract:* - This paper proposes the Meta-heuristics approaches using genetic algorithms (GA) and particle swarm optimization (PSO) for tuning power system stabilizer PSS parameters. In this work we have proposed a multi-objective function based on two objectives: first maximize the stability margin by increasing the damping factors and second minimize the eigenvalues real parts. For the effectiveness function proposed check, we compared it with mono-objective function. The simulation results, by comparative study between genetic algorithms and particle swarm optimizations techniques via multi objective and mono objective functions proved the efficiency of the PSS adapted by multi-objective function based genetic algorithms in comparison with particle swarm optimization, it's enhanced stability of power system works under different operating modes and different network configurations. The simulation results obtained under developed graphical user interface (GUI)

*Keywords-* Turbo-Alternator, Genetic Algorithms GA, Particle Swarm Optimization PSO, multi-objective function, mono-objective function, robustness, graphical interface GUI.

Received: November 23, 2021. Revised: October 11, 2022. Accepted: November 13, 2022. Published: December 5, 2022.

## 1. Introduction

The electrical energy has become the major form of energy for end use consumption in today's world. There is always a need to make electric energy generation and transmission, both more economic and reliable. The voltages throughout the system are also controlled to be within ±5% of their rated values by automatic voltage regulators acting on the generator field exciters, and by the sources of reactive power in the network, [1].

Stability and robustness are considered essential requirements for friability and continuity of electrical energy production this latter produced by a series of systems with very complex mathematical models called power systems. Since these systems are installed in complex environmental conditions they are exposed to a variation of uncertainty which is affected directly in the operation of these systems and therefore the stability of the energy production, the power system stabilizer PSS plays an important role to improve the power systems stability, [2].

The parameters of CPSS are determined based on the linearized model of the power system. Providing good damping over a wide operating range, the CPSS parameters should be fine-tuned in response to both types of oscillations. Since power systems are highly non-linear systems, with configurations and parameters which alter through time, the CPSS design based on the linearized model of the power system cannot guarantee its performance in a practical operating environment, [3]. Therefore, an adaptive PSS which considers the nonlinear nature of the plant and adapts to the changes in the environment is required for the power system, [3]. In order to improve the performance of CPSSs, numerous techniques have been proposed for designing them, such as intelligent optimization methods and fuzzy logic method [7, 8].

Meta-heuristic techniques are a new family of stochastic algorithms which aim to solve difficult optimization problems. Used to solve various applicative problems, these methods have the advantage to be generally efficient on a large number of problems.GA and PSO belong to population approaches. Meta-heuristics are generally used to solve a simplified OPF (Optimal Power Flow) problem such as the classic economic dispatch, security - constrained economic power dispatch, and reactive optimization problem, as well as optimal reconfiguration of an electric distribution network. [4],[6].

Genetic algorithms (GAs) were invented by John Holland in the 1960s and were developed by Holland and his students and colleagues at the University of Michigan in the 1960s and the 1970s. In contrast with evolution strategies and evolutionary programming, Holland's original goal was not to design algorithms to solve specific problems, but rather to formally study the phenomenon of adaptation as it occurs in nature and to develop ways in which the mechanisms of natural adaptation might be imported into computer systems, [5].

The Particle Swarm Optimization (PSO) strategy is a new class of algorithms proposed to solve continuous optimization problems . The Particle Swarm Optimizer was introduced by James Kennedy and Russell Eberhart in 1995. Inspired by social behavior and movement dynamics of insects, birds and fish, it is also related, however, to evolutionary computation, and has links to both genetic algorithms and evolution strategies, [4], [5].

In this paper, the robust PSS design is realized using multi-objective function optimization GA and PSO applied in the automatic excitation regulator of powerful synchronous generators

## 2. Power Systems Model

The dynamic performance study and stability analysis of power systems requires faithful mathematical models, we used in our work permeances networks modeling based on the PARK-GARIVE model of powerful synchronous generators for simplifying hypotheses and testing the control algorithm. The PSG model defined by the following equations [2, 15]:

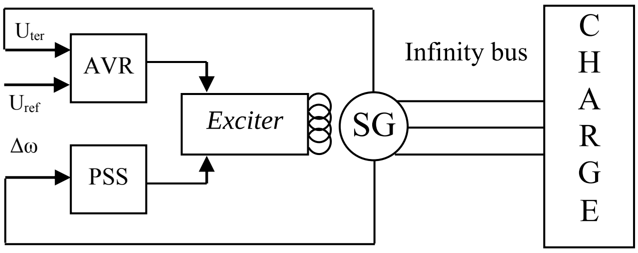



**Figure 1** Standard system IEEE type SMIB with excitation control of powerful synchronous generators

- Currents equations

$$i_d = \frac{U_q - E_q^{"}}{X_d^{"}} \qquad i_q = \frac{-U_d + E_d^{"}}{X_q^{"}} \qquad i_f = \frac{\phi_f - \phi_{ad}}{X_{sr}}$$
$$i_{1d} = \frac{\phi_{1d} - \phi_{ad}}{X_{srd}} \qquad i_{1q} = \frac{\phi_{1q} - \phi_{aq}}{X_{sr1q}} \qquad i_{2q} = \frac{\phi_{2q} - \phi_{aq}}{X_{sr2q}} \tag{1}$$

- Voltage equations

$$U_d = X_q^{"} i_q - E_d^{"} - r i_d$$
$$U_q = X_d^{"} i_d + E_q^{"} - r i_q$$
$$E_q^{"} = \frac{\dfrac{1}{\frac{X_{sf} X_f}{X_{ad}}} E_q^{'} + \dfrac{1}{\frac{X_{sfd} X_{fd}}{X_{ad}}} E_{fq}^{'}}{\dfrac{1}{X_{ad}} + \dfrac{1}{X_{sf}} + \dfrac{1}{X_{sfd}}} \qquad E_q^{"} = \frac{\dfrac{1}{\frac{X_{sfq} X_{fq}}{X_{aq}}} E_{fd}^{'}}{\dfrac{1}{X_{ad}} + \dfrac{1}{X_{sfq}}} \tag{2}$$

- Flow equations:

$$\phi_{ad} = E_q^{"} + (X_d^{"} - X_s) i_d \qquad \phi_{aq} = E_d^{"} + (X_q^{"} - X_s) i_q$$
$$\frac{1}{\omega_s}\frac{d\phi_f}{dt} = U_{f0} - R_f i_f \qquad \frac{1}{\omega_s}\frac{\phi_{1d}}{dt} = -R_{1d} i_{1d} \tag{3}$$
$$\frac{1}{\omega_s}\frac{\phi_{1q}}{dt} = -R_{1q} i_{1q} \qquad \frac{1}{\omega_s}\frac{\phi_{2q}}{dt} = -R_{2q} i_{2q}$$

- Mechanical equations

$$T_j \frac{d}{dt} s + \left(\Phi_{ad}.I_q - \Phi_{aq}.I_d\right) = M_T \quad \text{ou} \quad T_j \frac{d}{dt} s = M_T - M_e \tag{4}$$

- Automatic Voltage Regulator model (AVR)

$$V_R = \frac{K_A V_E - V_R}{T_A} \quad , \quad V_E = V_{ref} - V_F \tag{5}$$

- Power system stabilizer model (PSS)

$$V_{PSS} = K_{PSS} \frac{pT_\omega}{1 + pT_\omega} \frac{1 + pT_1}{1 + pT_2} \frac{1 + pT_3}{1 + pT_4} \Delta input \tag{6}$$

## 3. Meta-Heuristics

The new paradigms were called meta-heuristics and were first introduced in the mid-80s as a family of searching algorithms able to approach and solve complex optimization problems, using a set of several general heuristics. The term meta-heuristic was proposed in [16], to define a high level heuristic used to guide other heuristics for a better evolution in the search space. Although traditional stochastic search methods are mainly guided by chance (solutions change randomly from one step to another), they can be used in combination with meta-heuristic algorithms to guide the search process and to accelerate the convergence.

Most meta-heuristics algorithms are only approximation algorithms, because they cannot always find the global optimal solution, [9]. But the most attractive feature of a meta-heuristic is that its application requires no special knowledge on the optimization problem to be solved, hence it can be used to define the concept of a general problem solving model for optimization problems or other related problems, [17], [18]. Since their introduction in the mid-80s till now, meta-heuristic methods for solving optimization problems have been continuously developed, allowing addressing and solving a growing number of such problems, previously considered difficult or even impossible to solve. These methods include simulated annealing, tabu search, evolutionary computation techniques, artificial immune systems, genetic algorithms, particle swarm optimization, ant colony algorithm, differential evolution, harmony search, honey-bee colony optimization etc. The next section presents a brief review of basic issues for the most commonly used meta-heuristics cited above. Several applications of these methods in the field of power systems, [10].

In this work we are based on genetic algorithms, particle swarm optimization techniques.

### III.1.Genetic algorithms

Genetic Algorithm (GA) is a search technique that mimics the mechanisms of natural selection, discovered by John Holland in 1970, [11], [19].Cell is the building unit of all living organisms. In each cell there is a set of chromosomes which are strings of DNA. Every chromosome consists of genes which encode a particular protein. During reproduction, crossover first occurs. Genes from parents form in some way the whole new chromosome. However, the new created offspring can be mutated. Mutation occurs when the elements of DNA are a bit changed.

These changes are mainly caused by errors in copying genes from parents. The fitness of an organism is measured by success of the organism in its life. With generations, the good characteristics remain and the bad ones died which represents "The survival of the fittest".

Much work have been done on optimization by genetic algorithms to tune power system stabilizer parameters for adaptation and reliability of these techniques to power systems.

### III.2. Particle swarm optimization

Particle swarm optimization is a population based stochastic optimization method, [12].Explores for the optimal solution from a population swarm of moving particle vectors, based on a fitness function. Each ith particle vector represents a potential answer and has a position ($X_{ik}$) and a velocity ($V_{ik}$) at the kth iteration in the problem space. Each ith vector keeps a record of its individual best position ($P_{ik}$), which is associated with its own best fitness it has achieved so far, at any $k_{th}$ step in the iteration process. This value is known as pbesti. Moreover, the optimum position among all the particles obtained so far in the swarm is stored as the global best position ($P_{gk}$). This location is called gbest. The new velocity of particle will be updated according to the following equation, [13]:

$$v_l^{k+1} = wv_l^k + c_1 r_1 + \left(P_i^k - X_i^k\right) + c_2 r_2 + \left(P_g^k - X_i^k\right) \qquad (7)$$

where w is an inertia weight in the first part that represents the memory of a particle during a search, c1 and c2 are positive numbers illustrating the weights of the acceleration terms that guide each particle toward the individual best and the swarm best positions respectively, r1 and r2 are uniformly distributed random numbers in (0, 1), and N is the number of particles in the swarm. The second and the third parts of (8) represent cognitive and social parts respectively. The inertia weighting function in (7) is usually calculated using the following equation:

$$W = \frac{W_{\max} - (W_{\max} - W_{\min})iter}{iter_{\max}} \qquad (8)$$

Where $w_{max}$ and $w_{min}$ are the maximum and minimum values of w respectively, $iter_{max}$ is the maximum number of iterations and iter is the current iteration number. The first term in (7) enables each particle to perform a global search by exploring a new search space. The last two terms in (7) enable each particle to perform a local search around its individual best position and the swarm best position. Each particle changes its position based on the updated velocity according to the following equation:

$$X_i^{k+1} = X_i^k + V_i^{k+1} \qquad (9)$$

### III.3.The difference between GA and PSO

The PSO algorithm shares many common points with the genetic algorithm (GA). Both algorithms start with a population of individuals randomly generated; all both have objective function values for evaluating the population. Both algorithms start with the population and seek optimum random techniques. The two systems do not guarantee success. They also have the memory, which is important for the algorithm. Such as genetic algorithms, PSO is based on populations that slowly converge to one or more solutions. However, with PSO, the particles are preserved throughout the entire process; they do not die. Contrary to the genetic algorithm, this is based on competition for the best chance of survival and reproduction. PSO uses a type of cooperation between the molecules; this is realized by exchanging the coordinates of the best solutions which have been produced up to this point. PSO traditionally has no crossover between individuals, and has no mutation and the particles are never replaced by other individuals during execution. Instead of that PSO refines its research by attracting the particles [14, 20].

Table 1 gives us the difference between GA and PSO, [21].

**Table 1** a comparative between GA and PSO

| | GA | PSO |
|---|---|---|
| Base | Nature | Nature |
| Principle | Algorithm | Algorithm |
| Invidious | Chromosome | Bird, insect ... |
| selection | Utilizable | No utilizable |
| crossing | Utilizable | No utilizable |
| mutation | Utilizable | No utilizable |
| Number of individuals generated each iteration (example 30 individuals in a population) | 60 individuals (30 individuals of crossing and 30 individuals of mutation) | 30 individuals |
| Excursion Time | Court | Average |

## I. TUNING POWER SYSTEM STABILIZER PARAMETERS BASED GA AND PSO .

### IV.1 Objectives functions

The objective functions choice based on the needs of our controlled system, [21].

To study of the influence choice of objective functions in the controlled system performances we have realized a comparative study between two objective functions:

- Mono objective function.
- Multi objective function

### IV.1.1 Mono-objective function.

The aim of using PSS is to ensure a satisfactory damping of the oscillations and to guarantee the overall stability of the system for different operating points.

To meet this goal, we have used for the first time a mono objective function to minimize the real parts of the system eigenvalues. Therefore, all eigenvalues will be in D area of stability (figure 2)

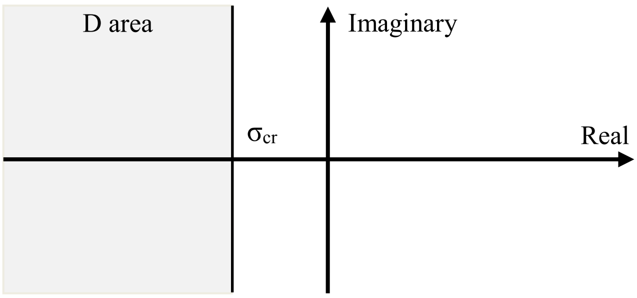



**Figure 2** Stability areas.

To understand this notion, we consider two systems with same imaginary parts $\omega_{s1} = \omega_{s2}$ and the deferent real part σ:

- System 1 : $P_{1,2}$=-6±6j
- System 2 : $P_{1,2}$=-1±6j

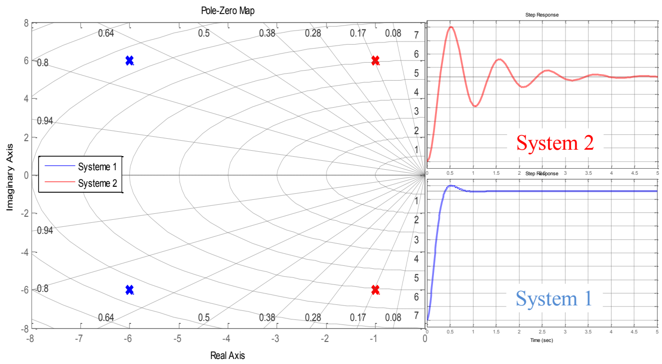



**Figure 3** The σ influence in the controlled system stability

In this result we can see that the decrease of the real part improved the dynamic performance and system stability.

Depending on this notion we proposed the flowing mono objective function which must minimize the real parts of the eigenvalues system.

$$F_{obj} = \min(\sigma) \tag{10}$$

### IV.1.2.Optimization results

To optimize and study of power system we created a graphical user interface GUI (figure 4) under MATLAB allows to:

- Optimize controller parameters using genetic algorithms and particle swarm optimization by mono and multi objective.
- View system regulation results and simulation.
- Calculate the system dynamics parameters.
- Test system stability and robustness.

### A. GA optimization method

To run optimization by genetic algorithms under GUI we use: **optimization /GA /PSS/ mono objective**

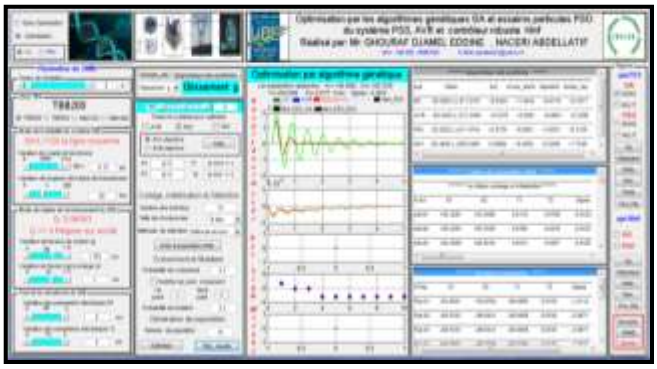


**Figure 4** PSS parameters syntheses using GA mono objective under GUI MATLAB

The below optimization result for: 10 generations and 10 individuals obtained using realized graphical interface.

********* Creating the initial population ********

********* 1st step coding and initialization ********

| N ind | K1 | K2 | T1 | T2 | Segma |
|---|---|---|---|---|---|
| Individu:01 | +05.3255 | +03.4588 | 0.0118 | 0.0726 | -0.9124 |
| Individu:02 | +05.3255 | +05.3529 | 0.0785 | 0.0216 | -0.9127 |
| Individu:03 | +02.3059 | +00.5216 | 0.0313 | 0.0397 | -0.9197 |
| Individu:04 | +06.6431 | +02.7176 | 0.0633 | 0.0138 | -0.9099 |
| Individu:05 | +06.4784 | +00.6039 | 0.0064 | 0.0698 | -0.9101 |
| Individu:06 | +00.2471 | +02.9373 | 0.0169 | 0.0028 | -0.5860 |
| Individu:07 | +00.4392 | +05.4902 | 0.0559 | 0.0616 | -0.5478 |
| Individu:08 | +03.6235 | +02.0588 | 0.0528 | 0.0236 | -0.9167 |
| Individu:09 | +06.8078 | +01.0431 | 0.0906 | 0.0040 | -0.9089 |
| Individu:10 | +05.4902 | +04.5843 | 0.0988 | 0.0514 | -0.9112 |

********* 2nd step selection ********

| N ind | K1 | K2 | T1 | T2 | Segma |
|---|---|---|---|---|---|
| Individu:01 | +05.3255 | +05.3529 | 0.0785 | 0.0216 | -00.9127 |
| Individu:02 | +02.3059 | +00.5216 | 0.0313 | 0.0397 | -00.9197 |
| Individu:03 | +02.3059 | +00.5216 | 0.0313 | 0.0397 | -00.9197 |
| Individu:04 | +06.4784 | +00.6039 | 0.0064 | 0.0698 | -00.9101 |
| Individu:05 | +06.4784 | +00.6039 | 0.0064 | 0.0698 | -00.9101 |
| Individu:06 | +00.2471 | +02.9373 | 0.0169 | 0.0028 | -00.5860 |
| Individu:07 | +03.6235 | +02.0588 | 0.0528 | 0.0236 | -00.9167 |
| Individu:08 | +03.6235 | +02.0588 | 0.0528 | 0.0236 | -00.9167 |
| Individu:09 | +05.4902 | +04.5843 | 0.0988 | 0.0514 | -00.9112 |
| Individu:10 | +02.3059 | +00.5216 | 0.0313 | 0.0397 | -00.9197 |

********* 3rd step Crossing ********

| N ind | K1 | K2 | T1 | T2 | Segma |
|---|---|---|---|---|---|
| Individu:01 | +05.8745 | +05.7922 | 0.0801 | 0.0154 | -00.9116 |
| Individu:02 | +01.7569 | +00.0824 | 0.0298 | 0.0459 | -00.9209 |
| Individu:03 | +02.9647 | +00.6314 | 0.0313 | 0.0444 | -00.9182 |
| Individu:04 | +05.8196 | +00.4941 | 0.0064 | 0.0651 | -00.9118 |
| Individu:05 | +05.4902 | +01.1529 | 0.0173 | 0.0526 | -00.9126 |
| Individu:06 | +01.2353 | +02.3882 | 0.0060 | 0.0201 | -00.9966 |
| Individu:07 | +03.6235 | +02.0588 | 0.0528 | 0.0236 | -00.9167 |
| Individu:08 | +03.6235 | +02.0588 | 0.0528 | 0.0236 | -00.9167 |
| Individu:09 | +05.8196 | +04.0353 | 0.0801 | 0.0655 | -00.9102 |
| Individu:10 | +01.9765 | +01.0706 | 0.0501 | 0.0256 | -00.9203 |

********* 4st Step Mutation ********

| N ind | K1 | K2 | T1 | T2 | Segma |
|---|---|---|---|---|---|
| Individu:01 | +04.0078 | +01.8392 | 0.0902 | 0.0044 | -00.9157 |
| Individu:02 | +01.7569 | +03.6784 | 0.0793 | 0.0271 | -00.9282 |
| Individu:03 | +00.3294 | +00.4941 | 0.0095 | 0.0444 | -00.5863 |
| Individu:04 | +02.3059 | +00.7137 | 0.0048 | 0.0710 | -00.9196 |
| Individu:05 | +03.8980 | +01.5373 | 0.0485 | 0.0655 | -00.9155 |
| Individu:06 | +05.2431 | +02.3608 | 0.0294 | 0.0099 | -00.9138 |
| Individu:07 | +05.7647 | +02.6078 | 0.0520 | 0.0691 | -00.9107 |
| Individu:08 | +03.2941 | +03.6784 | 0.0863 | 0.0150 | -1.41140 |
| Individu:09 | +00.7412 | +04.0902 | 0.0563 | 0.0628 | -00.6215 |
| Individu:10 | +05.9294 | +02.8275 | 0.0926 | 0.0256 | -00.9107 |

********* Optimization Results ********

| N Pop | K1 | K2 | T1 | T2 | Segma |
|---|---|---|---|---|---|
| Population:01 | +03.2941 | +03.6784 | +00.0863 | 0.0150 | -1.4114 |
| Population:02 | +06.5333 | +06.6431 | +00.0856 | 0.0330 | -2.0971 |
| Population:03 | +06.5333 | +06.6431 | +00.0856 | 0.0330 | -2.0971 |
| Population:04 | +06.5882 | +06.7529 | +00.0294 | 0.0177 | -3.2822 |
| Population:05 | +06.5882 | +06.7529 | +00.0294 | 0.0177 | -3.2822 |
| Population:06 | +06.5882 | +06.7529 | +00.0294 | 0.0177 | -3.2822 |
| Population:07 | +06.5882 | +06.7529 | +00.0294 | 0.0177 | -3.2822 |
| Population:08 | +06.5882 | +06.7529 | +00.0294 | 0.0177 | -3.2822 |
| Population:09 | +06.5882 | +06.7529 | +00.0294 | 0.0177 | -3.2822 |
| Population:10 | +06.5882 | +06.7529 | +00.0294 | 0.0177 | -3.2822 |

Optimization is completed .......
The optimized parameters K1= +06.5882 K2= +06.7529 T1=+00.0294 T2= 0.0177 Segma= -3.2822

DOI: 10.37394/23205.2022.21.38


## B. PSO optimization method

To run particle swarm optimization under GUI we use: **optimization /PSO /PSS/ mono objective**

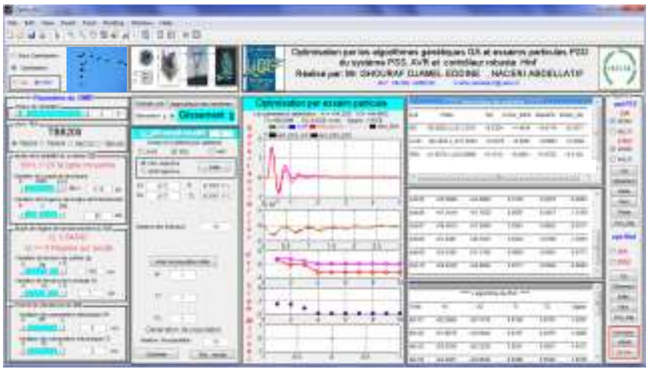


**Figure 5** PSS parameters syntheses using PSO mono objective under GUI MATLAB

************ PSO initialization ***********

| N ind | K1 | K2 | T1 | T2 | Segma |
|---|---|---|---|---|---|
| Individu:01 | +02.2669 | +03.7478 | 0.0196 | 0.0735 | -1.2297 |
| Individu:02 | +01.8199 | +03.4169 | 0.0887 | 0.0380 | -0.8399 |
| Individu:03 | +03.5118 | +01.0559 | 0.0077 | 0.0258 | -0.9175 |
| Individu:04 | +00.9394 | +02.0283 | 0.0995 | 0.0431 | -0.6079 |
| Individu:05 | +00.9999 | +04.9862 | 0.0136 | 0.0070 | -0.9263 |
| Individu:06 | +01.3419 | +01.7628 | 0.0087 | 0.0457 | -1.0159 |
| Individu:07 | +05.4810 | +01.0900 | 0.0381 | 0.0693 | -0.9117 |
| Individu:08 | +01.9461 | +06.7947 | 0.0322 | 0.0798 | -1.0203 |
| Individu:09 | +03.0305 | +01.3165 | 0.0950 | 0.0563 | -0.9171 |
| Individu:10 | +05.9236 | +04.6694 | 0.0731 | 0.0946 | -0.9093 |

******* PSO algorithm *******

| N Pop | K1 | K2 | T1 | T2 | Segma |
|---|---|---|---|---|---|
| Iteration:01 | +02.2669 | +03.7478 | 0.0196 | 0.0735 | -1.2297 |
| Iteration:02 | +01.9678 | +03.6840 | 0.0197 | 0.0479 | -1.2695 |
| Iteration:03 | +02.8359 | +05.3230 | 0.0241 | 0.0581 | -1.5411 |
| Iteration:04 | +04.2027 | +04.9842 | 0.0396 | 0.0528 | -1.9376 |
| Iteration:05 | +04.2027 | +04.9842 | 0.0396 | 0.0528 | -1.9376 |
| Iteration:06 | +04.2027 | +04.9842 | 0.0396 | 0.0528 | -1.9376 |
| Iteration:07 | +04.2027 | +04.9842 | 0.0396 | 0.0528 | -1.9376 |
| Iteration:08 | +04.2027 | +04.9842 | 0.0396 | 0.0528 | -1.9376 |
| Iteration:09 | +04.2027 | +04.9842 | 0.0396 | 0.0528 | -1.9376 |
| Iteration:10 | +04.2027 | +04.9842 | 0.0396 | 0.0528 | -1.9376 |

Optimization is completed .......
The optimized parameters K1= +04.2027 K2= +04.9842 T1=+00.0396 T2= 0.0528 Segma= -1.9376

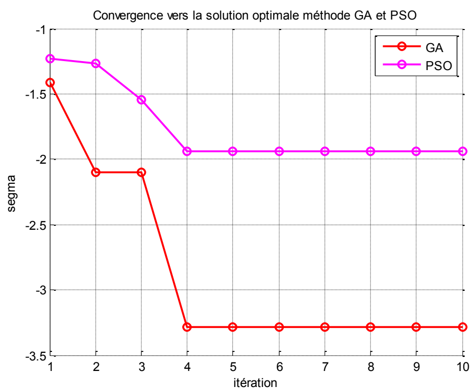



**Figure 6** Optimization result of GA and PSO using mono objective function

The optimization results obtained show that the GA (σ = -3.2822) more reliable compared to PSO (σ = -1.9376)

### IV.1.3.Simulation results

For SMIB system stability study we have performed perturbation in turbine torque ($\Delta T_m$ =15% at 0.5 second)

We simulated SMIB system under

- Different operations regimes: under-excited, the nominal and the over-excited.
- Different electrical network: long, court and average
- Different synchronous generators: TBB 200, 500, 1000 and BBC720.

We optimized the controller parameters by GA and PSO under different conditions cited above.

The following results were obtained by SMIB studied with following cases: closed loop System with PSS_GA_ mono objective and PSS_PSO_ mono objective

Figures 7 and 8 show simulation results of power system under critical regime (under excited and long transmission line network)

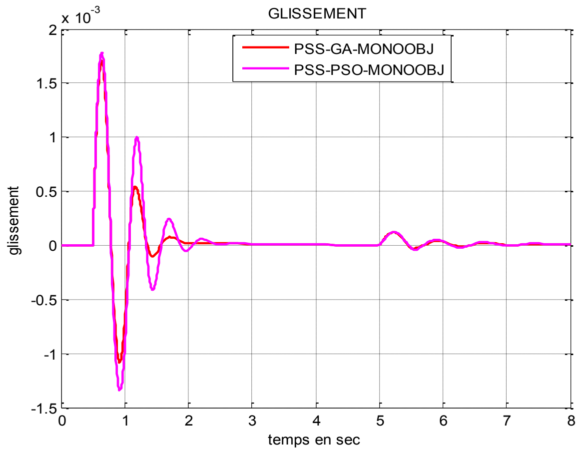



**Figure 7** Variable speed

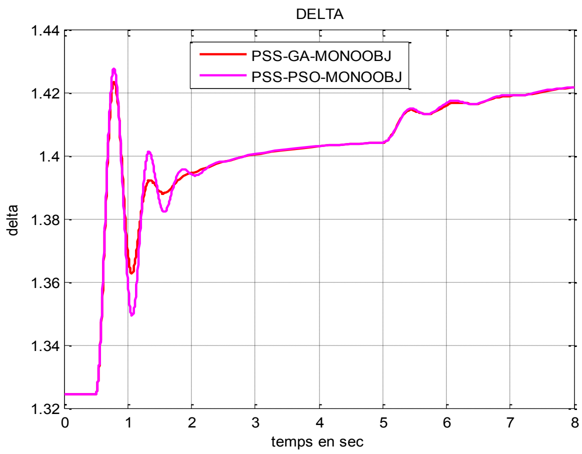



**Figure 8** Internal angle

From the obtained results it can be seen that:

- The parameters optimization of power system stabilizer PSS using mono objective GA and PSO gives the SMIB system a considerable improvements in the stability and dynamics performances
- Concerning the optimization method, the GA is well adapted with the system SMIB compared to the PSO.

### IV.2.1 Multi objective function.

The system SMIB is stable based on mono objective function, but it contains a disadvantage especially if the two factors $\sigma$ and damping coefficient $\zeta$ are minimal simultaneously. The dynamic behavior of such a system depends on two values: $\sigma$ and especially the damping coefficient $\zeta$. To study the influence of damping coefficient $\zeta$ on the controlled system we consider two systems with real part $\sigma_{s1}=\sigma_{s2}$ and $\omega_{s1} \neq \omega_{s2}$ ($\omega$ imaginary part):

System 21: $P_{21,2}=-2\pm j$ with $\zeta = 0.8944$

System 22: $P_{22,2}=-2\pm 8j$ with $\zeta = 0.2425$

The systems poles and step responses match each system shown in figure 9.

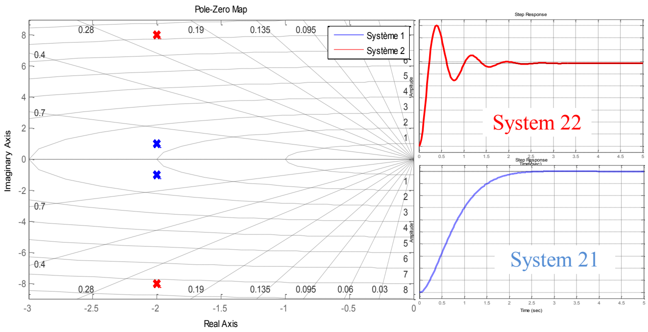



**Figure 9** the $\zeta$ influence to controlled system

From the obtained results, it can be seen that:

The increase of the damping coefficient $\zeta$ improves system stability. Based on these results we propose a new objective function composed by two functions. This function must maximize stability margin by increasing the damping factors while minimizing the real parts of the system eigenvalues, and second function must maximize the set of two objective functions.

$$\max(\zeta)-\min(\sigma) \tag{11}$$

$$F_{Mult\_obj}=\max\left(\max(\zeta)-\min(\sigma)\right) \tag{12}$$

### IV.2.2.Optimization results

### A. GA optimization method

To run GA multi objective optimization under GUI we use: **optimization /GA /PSS/ multiobjective**

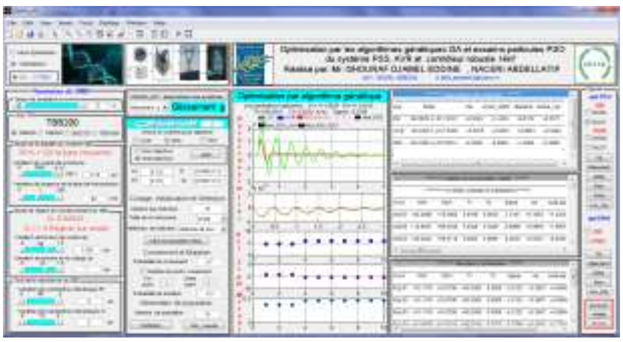


**Figure 10** PSS parameters syntheses using GA multi objective under GUI MATLAB

Optimization example using GA technique with Number of individuals=10 , Number of population =10

*********** Creating the initial population **********

*********** 1st step coding and initialization **********

| N ind | K1 | K2 | T1 | T2 | Sigma | ksi | multi-obj |
|---|---|---|---|---|---|---|---|
| Individu:01 | +02.2588 | +10.5882 | 0.0106 | 0.0843 | -1.3181 | +0.1054 | +1.4234 |
| Individu:02 | +10.9647 | +09.2706 | 0.0329 | 0.0459 | -0.8998 | +0.9959 | +1.8957 |
| Individu:03 | +00.2824 | +09.4118 | 0.0622 | 0.0890 | -0.4643 | +0.0381 | +0.5023 |
| Individu:04 | +02.0706 | +08.0941 | 0.0473 | 0.0432 | -1.1622 | +0.0932 | +1.2554 |
| Individu:05 | +09.1765 | +07.8118 | 0.0711 | 0.0659 | -0.9022 | +0.9960 | +1.8982 |
| Individu:06 | +05.9765 | +11.4353 | 0.0602 | 0.0792 | -1.4711 | +0.1082 | +1.5794 |
| Individu:07 | +00.5647 | +03.6706 | 0.0294 | 0.0702 | -0.6460 | +0.0535 | +0.6995 |
| Individu:08 | +02.6353 | +02.4471 | 0.0906 | 0.0154 | -0.9187 | +0.9989 | +1.9176 |
| Individu:09 | +03.8118 | +02.5412 | 0.0501 | 0.0095 | -0.9165 | +0.9964 | +1.9129 |
| Individu:10 | +05.9765 | +11.1529 | 0.0828 | 0.0706 | -1.2001 | +0.0891 | +1.2892 |

*********** 2nd step selection **********

| N ind | K1 | K2 | T1 | T2 | Sigma | ksi | multi-obj |
|---|---|---|---|---|---|---|---|
| Individu:01 | +10.9647 | +09.2706 | 0.0329 | 0.0459 | -00.8998 | +0.9959 | +1.8957 |
| Individu:02 | +10.9647 | +09.2706 | 0.0329 | 0.0459 | -00.8998 | +0.9959 | +1.8957 |
| Individu:03 | +02.0706 | +08.0941 | 0.0473 | 0.0432 | -01.1622 | +0.0932 | +1.2554 |
| Individu:04 | +09.1765 | +07.8118 | 0.0711 | 0.0659 | -00.9022 | +0.9960 | +1.8982 |
| Individu:05 | +09.1765 | +07.8118 | 0.0711 | 0.0659 | -00.9022 | +0.9960 | +1.8982 |
| Individu:06 | +05.9765 | +11.4353 | 0.0602 | 0.0792 | -01.4711 | +0.1082 | +1.5794 |
| Individu:07 | +02.6353 | +02.4471 | 0.0906 | 0.0154 | -00.9187 | +0.9989 | +1.9176 |
| Individu:08 | +02.6353 | +02.4471 | 0.0906 | 0.0154 | -00.9187 | +0.9989 | +1.9176 |
| Individu:09 | +03.8118 | +02.5412 | 0.0501 | 0.0095 | -00.9165 | +0.9964 | +1.9129 |
| Individu:10 | +02.6353 | +02.4471 | 0.0906 | 0.0154 | -00.9187 | +0.9989 | +1.9176 |

*********** 3rd step Crossing **********

**croissement state**

Pc = 0.267 **0 1 1 1 1 0 0 1** 0 0 1 1 0 1 1 0 **0 1 1 0 0 1 1 1** 0 0 1 0 0 1 0 0 -----> Pc < PC: There is a crossing
Pc = 0.267 **0 1 1 1 1 0 0 1** 0 0 1 1 0 1 1 0 **0 1 1 0 0 1 1 1** 0 0 1 0 0 1 0 0 -----> Pc < PC: There is a crossing
Pc = 0.521 **0 0 1 1 1 0 0 0** 0 0 1 1 0 1 0 0 **1 1 0 0 0 1 1 1** 0 0 1 0 0 1 1 1 -----> Pc < PC: There is a crossing
Pc = 0.521 **0 0 1 1 1 0 0 0** 0 0 1 1 0 1 0 0 **1 1 1 0 0 0 1 1** 0 0 1 0 0 1 1 1 -----> Pc < PC: There is a crossing
Pc = 0.766 **0 0 1 1 1 0 0 0** 0 0 1 1 0 1 0 0 **1 1 1 0 0 1 1 1** 0 0 1 0 0 1 1 1 -----> Pc > PC: no crossing ……..
Pc = 0.766 **1 1 0 0 0 0 1 1** 1 0 1 0 0 1 1 0 **1 0 1 1 0 1 0 1** 1 0 1 0 1 0 0 0 -----> Pc > PC: no crossing….......
Pc = 0.571 **0 0 1 0 1 1 0 0** 1 0 1 0 1 1 0 0 **0 1 1 1 1 0 0 0** 0 1 1 0 1 1 1 0 -----> Pc < PC: There is a crossing
Pc = 0.571 **0 0 1 0 1 1 0 0** 1 0 1 0 1 1 0 0 **0 1 1 1 1 0 0 0** 0 1 1 0 1 1 1 0 -----> Pc < PC: There is a crossing
Pc = 0.765 **1 1 1 0 1 0 0 1** 1 1 0 0 0 1 0 1 **0 1 0 1 0 0 1 1** 0 1 1 1 0 1 0 1 -----> Pc > PC: no crossing ……..
Pc = 0.765 **1 1 1 0 1 0 0 1** 1 1 0 0 0 1 0 1 **0 1 0 1 0 0 1 1** 0 1 1 1 0 1 0 1 -----> Pc > PC: no crossing …… ..

| N ind | K1 | K2 | T1 | T2 | Sigma | ksi | multi-obj |
|---|---|---|---|---|---|---|---|
| Individu:01 | +10.9647 | +09.2706 | 0.0329 | 0.0459 | -00.8998 | +0.9959 | +1.8957 |
| Individu:02 | +10.9647 | +09.2706 | 0.0329 | 0.0459 | -00.8998 | +0.9959 | +1.8957 |
| Individu:03 | +02.0706 | +08.0941 | 0.0473 | 0.0432 | -01.1622 | +0.0932 | +1.2554 |
| Individu:04 | +09.1765 | +07.8118 | 0.0711 | 0.0659 | -00.9022 | +0.9960 | +1.8982 |
| Individu:05 | +09.1765 | +07.8118 | 0.0711 | 0.0659 | -00.9022 | +0.9960 | +1.8982 |
| Individu:06 | +05.9765 | +11.4353 | 0.0602 | 0.0792 | -01.4711 | +0.1082 | +1.5794 |
| Individu:07 | +02.6353 | +02.4471 | 0.0906 | 0.0154 | -00.9187 | +0.9989 | +1.9176 |
| Individu:08 | +02.6353 | +02.4471 | 0.0906 | 0.0154 | -00.9187 | +0.9989 | +1.9176 |
| Individu:09 | +05.6941 | +02.5412 | 0.0407 | 0.0142 | -00.9125 | +0.9920 | +1.9045 |
| Individu:10 | +00.7529 | +02.4471 | 0.1000 | 0.0107 | -00.6279 | +0.0523 | +0.6802 |

*********** 4st Step Mutation **********

**mutation probabilities used**

0.03 0.51 0.510.770.470.370.920.640.650.330.240.820.420.240.560.200.620.610.380.440.530.050.810.35 0.420.750.330.460.730.640.750.8
0.76 0.83 0.830.410.780.420.080.580.060.750.510.200.040.420.100.110.220.490.340.840.950.750.460.240.110.370.790.270.190.040.540.61
0.46 0.20 0.560.970.880.120.560.340.740.650.680.800.490.040.740.130.050.270.990.870.360.620.360.930.480.760.830.530.740.620.1 70.19
0.29 0.39 0.110.030.250.820.030.290.340.670.790.730.100.250.410.830.040.650.040.550.380.430.310.410.770.920.560.960.020.480.210.26
0.84 0.01 0.670.210.600.260.410.560.310.230.050.540.060.860.170.770.070.140.430.620.960.060.110.550.580.900.050.980.280.800.0 70.98
0.65 0.69 0.440.020.780.350.051.000.810.630.990.930.800.520.850.270.860.640.430.330.410.300.210.130.720.610.020.210.310.810.070.96
0.46 0.76 0.460.210.150.150.630.440.120.120.220.380.780.590.280.520.530.590.110.960.770.200.750.680.480.830.420.680.210.730.3 90.14
0.03 0.64 0.270.930.500.890.910.800.750.080.600.331.000.630.910.690.250.130.800.060.950.180.410.470.580.930.810.000.220.540.430.13
0.29 0.88 0.490.680.440.630.440.570.320.200.200.640.660.590.230.320.940.380.270.670.770.940.520.390.110.520.820.910.050.020.8 30.34
0.63 0.81 0.040.690.850.830.600.110.170.440.690.060.390.660.070.830.810.030.590.440.601.000.890.670.520.910.370.830.920.840.4 80.37

**Coding after mutation**

**1 1 1 0 1 0 1 1** 0 1 0 1 1 1 1 0 **1 1 0 1 0 0 1 0** 1 1 1 0 1 0 0 1
**0 1 1 0 1 0 0 1** 1 1 1 0 0 0 0 0 **0 1 0 1 0 1 1 1** 0 1 1 1 0 1 0 1
**0 1 1 0 1 0 0 0** 1 0 1 0 1 0 0 1 **1 0 1 1 1 0 0 0** 0 1 1 0 1 1 0 1
**1 0 0 1 0 1 1 1** 1 1 0 0 1 1 0 0 **0 1 1 1 0 0 1 1** 1 0 0 0 0 0 1 0
**0 1 1 1 1 0 0 0** 1 0 1 0 1 0 1 0 **0 0 0 1 0 1 0 1** 1 0 1 0 0 0 1 1
**0 1 1 0 1 1 0 1** 1 1 1 1 0 0 1 0 **1 0 0 1 1 0 1 0** 1 1 1 1 1 0 0 0
**0 0 1 0 0 1 0 0** 1 1 0 1 0 1 1 0 **1 1 0 0 0 0 1 1** 0 0 1 0 1 1 1 0
**1 0 0 1 1 0 0 0** 0 1 1 1 0 1 0 0 **0 0 1 1 0 0 1 1** 0 0 1 1 1 1 1 0
**1 1 1 1 1 0 0 1** 0 1 0 1 0 1 0 0 **0 1 0 0 0 1 1 1** 1 0 1 0 1 0 0 0
**0 0 1 1 0 0 0 1** 1 0 1 0 0 1 1 0 **1 0 1 1 1 1 1 1** 0 0 0 1 1 0 1 1

| N ind | K1 | K2 | T1 | T2 | Sigma | ksi | multi-obj |
|---|---|---|---|---|---|---|---|
| Individu:01 | +11.0588 | +04.4235 | 0.0824 | 0.0914 | -00.8944 | +0.9811 | +01.8755 |
| Individu:02 | +04.9412 | +10.5412 | 0.0344 | 0.0459 | -02.4718 | +0.1902 | +02.6619 |
| Individu:03 | +04.8941 | +07.9529 | 0.0723 | 0.0428 | -01.7022 | +0.1293 | +01.8315 |
| Individu:04 | +07.1059 | +09.6000 | 0.0454 | 0.0510 | -02.9609 | +0.2143 | +03.1752 |
| Individu:05 | +05.6471 | +08.0000 | 0.0087 | 0.0640 | -02.7400 | +0.2169 | +02.9569 |
| Individu:06 | +05.1294 | +11.3882 | 0.0606 | 0.0973 | -01.1078 | +0.0833 | +01.1911 |
| Individu:07 | +01.6941 | +10.0706 | 0.0766 | 0.0181 | -00.9218 | +0.0746 | +00.9964 |
| Individu:08 | +07.1529 | +05.4588 | 0.0204 | 0.0244 | -00.9094 | +0.9958 | +01.9052 |
| Individu:09 | +11.7176 | +03.9529 | 0.0282 | 0.0659 | -00.8965 | +0.9798 | +01.8763 |
| Individu:10 | +02.3059 | +07.8118 | 0.0750 | 0.0107 | -01.1738 | +0.0952 | +01.2690 |

*********** Optimization Results **********

| N pob | K1 | K2 | T1 | T2 | Sigma | ksi | multi-obj |
|---|---|---|---|---|---|---|---|
| Population:01 | +10.1176 | +10.0706 | 0.0360 | 0.0009 | -4.2122 | +0.3687 | +4.5809 |
| Population:02 | +10.1176 | +10.0706 | 0.0360 | 0.0009 | -4.2122 | +0.3687 | +4.5809 |
| Population:03 | +10.1176 | +10.0706 | 0.0360 | 0.0009 | -4.2122 | +0.3687 | +4.5809 |
| Population:04 | +11.9529 | +11.8118 | 0.0216 | 0.0232 | -5.2799 | +0.4568 | +5.7367 |
| Population:05 | +11.9529 | +11.8118 | 0.0216 | 0.0232 | -5.2799 | +0.4568 | +5.7367 |
| Population:06 | +11.9529 | +11.8118 | 0.0216 | 0.0232 | -5.2799 | +0.4568 | +5.7367 |
| Population:07 | +11.9529 | +11.8118 | 0.0216 | 0.0232 | -5.2799 | +0.4568 | +5.7367 |
| Population:08 | +11.9529 | +11.8118 | 0.0216 | 0.0232 | -5.2799 | +0.4568 | +5.7367 |
| Population:09 | +11.9529 | +11.8118 | 0.0216 | 0.0232 | -5.2799 | +0.4568 | +5.7367 |

Population:10 +11.9529 +11.8118 0.0216 0.0232 -5.2799 +0.4568 +5.7367

Optimization is completed .......
The optimized parameters:K1= +11.9529 K2= +11.8118 T1=+00.0216 T2= 0.0232 Sigma= -5.2799 Ksi= +0.4568 multi-obj =5.7367

### B. PSO optimization method

To run GUI for optimization by particle swarm we use **optimization /PSO /PSS/ multiobjective**

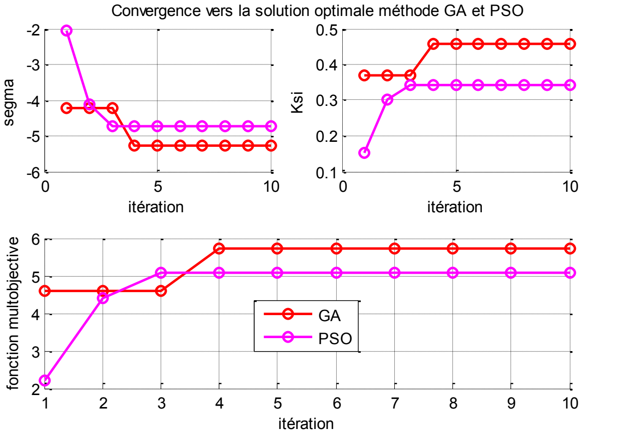



**Figure 11** PSS parameters syntheses using PSO multi objective under GUI MATLAB

Optimization example using PSO technique with Number of individuals=10 , Number of population =10

********* PSO initialization ********

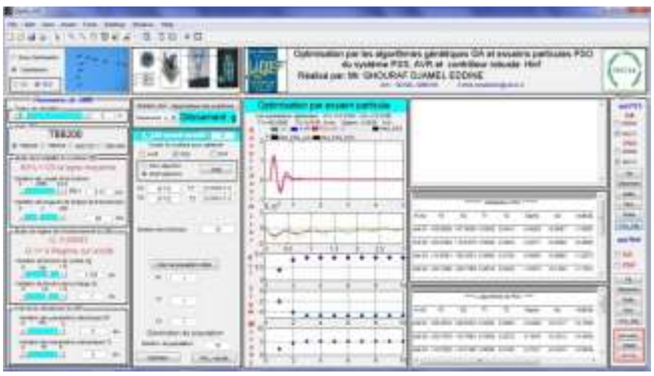
| N ind | K1 | K2 | T1 | T2 | Sigma | ksi | multi-obj |
|---|---|---|---|---|---|---|---|
| Individu:01 | +09.6808 | +07.9646 | 0.0268 | 0.0441 | -0.9029 | +0.9957 | +1.8987 |
| Individu:02 | +00.6388 | +10.6478 | 0.0098 | 0.0804 | -0.8217 | +0.0668 | +0.8885 |
| Individu:03 | +10.9561 | +09.3053 | 0.0686 | 0.0126 | -0.9005 | +0.9968 | +1.8973 |
| Individu:04 | +05.3306 | +08.7789 | 0.0816 | 0.0432 | -1.6275 | +0.1226 | +1.7501 |
| Individu:05 | +06.4605 | +11.5067 | 0.0345 | 0.0986 | -1.8937 | +0.1387 | +2.0324 |
| Individu:06 | +01.9972 | +09.8907 | 0.0077 | 0.0755 | -1.2895 | +0.1040 | +1.3936 |
| Individu:07 | +07.6214 | +07.1632 | 0.0062 | 0.0689 | -0.9071 | +0.9982 | +1.9053 |
| Individu:08 | +08.1054 | +05.5867 | 0.0723 | 0.0080 | -0.9067 | +0.9940 | +1.9007 |
| Individu:09 | +07.6468 | +06.0865 | 0.0152 | 0.0033 | -0.9090 | +0.9966 | +1.9056 |
| Individu:10 | +03.6881 | +08.7959 | 0.0277 | 0.0887 | -1.4943 | +0.1161 | +1.6104 |

***** PSO algorithm *****

| N itération | K1 | K2 | T1 | T2 | Sigma | ksi | multi-obj |
|---|---|---|---|---|---|---|---|
| Itération:01 | +06.2910 | +08.8352 | 0.0285 | 0.0964 | -2.0428 | +0.1517 | +2.1945 |
| Itération:02 | +08.7976 | +10.7664 | 0.0568 | 0.0233 | -4.1044 | +0.3012 | +4.4056 |
| Itération:03 | +10.2320 | +10.3196 | 0.0606 | 0.0195 | -4.7231 | +0.3415 | +5.0645 |
| Itération:04 | +10.2320 | +10.3196 | 0.0606 | 0.0195 | -4.7231 | +0.3415 | +5.0645 |
| Itération:05 | +10.2320 | +10.3196 | 0.0606 | 0.0195 | -4.7231 | +0.3415 | +5.0645 |
| Itération:06 | +10.2320 | +10.3196 | 0.0606 | 0.0195 | -4.7231 | +0.3415 | +5.0645 |
| Itération:07 | +10.2320 | +10.3196 | 0.0606 | 0.0195 | -4.7231 | +0.3415 | +5.0645 |
| Itération:08 | +10.2320 | +10.3196 | 0.0606 | 0.0195 | -4.7231 | +0.3415 | +5.0645 |
| Itération:09 | +10.2320 | +10.3196 | 0.0606 | 0.0195 | -4.7231 | +0.3415 | +5.0645 |
| Itération:10 | +10.2320 | +10.3196 | 0.0606 | 0.0195 | -4.7231 | +0.3415 | +5.0645 |

Optimization is completed.......
The optimized parameters
K1= +10.2320 K2= +10.3196 T1=+00.0606 T2= 0.0195 Sigma= -2.0428 ksi= +0.1517 multiobj= +5.0645

**Figure 12** Optimization results of GA and PSO



The optimization results obtained (examples and figure 5) show that:

1. GA and PSO optimizations techniques well adapted to multi objective function:
   - Increase damping coefficient ζ.
   - Decrease of real part of the poles σ.
   - Increase multi objective function.
2. GA (GA_Multi = +5.7367) more reliable than PSO (PSO_Multi = +5.0645).

### IV.2.3.Simulation results

Figures 13, 14 and 15 show simulation results of power system studied under different regimes with: a:'s' variable speed, b:'delta' the power angle. System SMIB controlled using: PSS_GA_ mono objective, PSS_PSO_ mono objective, PSS_GA_ multi objective and PSS_PSO_ multi objective. Table 2 present the static and dynamics performances analyze of power system and PSS parameters optimized using GA and PSO calculated under GUI realized for long transmission line network and different values of reactive power (under excited, nominal, and over excited) for TBB 200.

With:

- $\varepsilon_s$ %: the static error.
- ts : the settling time for 5%.
- d%: the maximum overshoot.
- Poles.

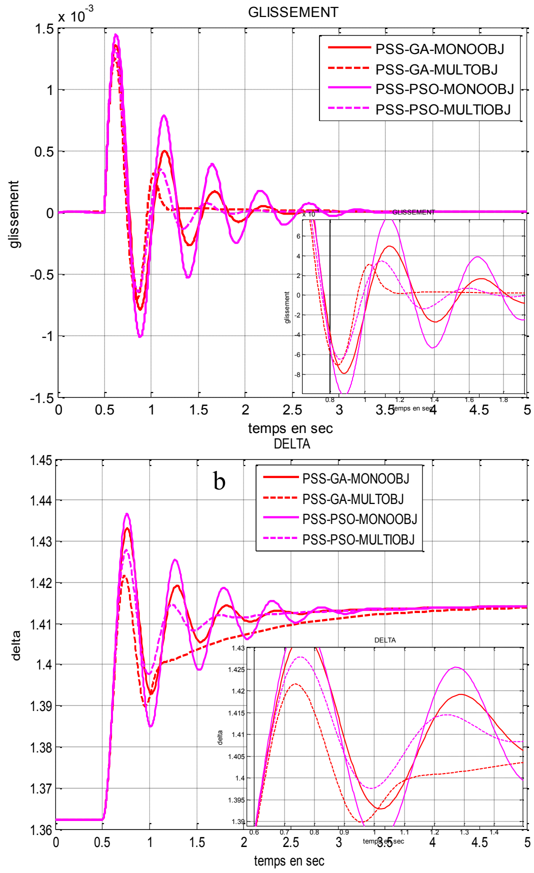



**Figure 13** over excited regime operation

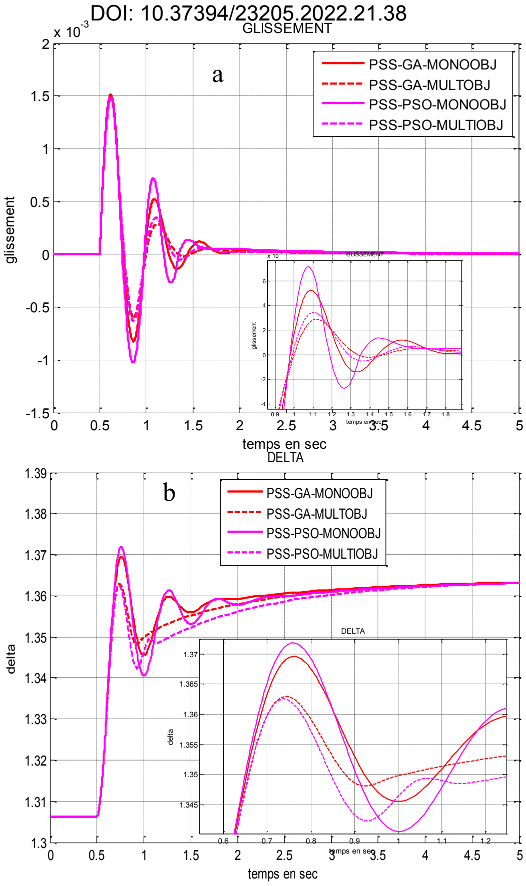



**Figure 14** under excited regime operation

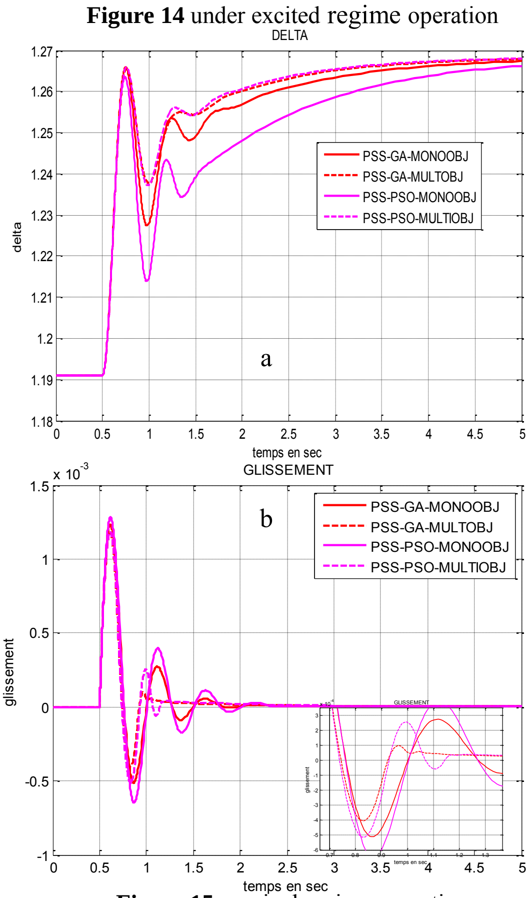



**Figure 15** nominal regime operation

**Table 2** static and dynamic performances of PSS optimized using GA and PSO mono objective and multi objective function

| | | $K_1$ | $K_2$ | $T_1$ | $T_2$ | Poles | $\zeta$ | D% | $\varepsilon$ | $t_r$ |
|---|---|---|---|---|---|---|---|---|---|---|
| **Over excited regime** | | | | | | | | | | |
| GA | MULTI | 04.3647 | 06.2039 | 0.0294 | 0.0354 | -5.9729 ± j 7.9196 | 0.6021 | 1.5042 | 0.0000 | 0.2628 |
| GA | MONO | 05.2157 | 06.3412 | 0.0091 | 0.0475 | -3.2284 ± j17.4565 | 0.1819 | 3.7240 | 0.0500 | 0.3023 |
| PSO | MULTI | 04.3562 | 06.3870 | 0.0309 | 0.0416 | -4.0989 ± j 8.8387 | 0.5679 | 1.6796 | 0.0139 | 0.2878 |
| PSO | MONO | 05.8831 | 06.4536 | 0.0367 | 0.0759 | -2.5594 ± j 19.656 | 0.1291 | 4.1776 | 0.0500 | 0.3037 |
| **Nominal regime** | | | | | | | | | | |
| GA | MULTI | 05.7922 | 06.6706 | 0.0614 | 0.0189 | -4.2363 ± j 6.4478 | 0.5491 | 2.9247 | 0.0000 | 0.2998 |
| GA | MONO | 04.7255 | 04.3137 | 0.0274 | 0.0600 | -3.7175 ± j 18.954 | 0.1925 | 4.4628 | 0.0129 | 0.3106 |
| PSO | MULTI | 07.2520 | 07.4527 | 0.0207 | 0.0331 | -3.9057 ± j 8.4649 | 0.4190 | 2.9207 | 0.0087 | 0.3084 |
| PSO | MONO | 03.9005 | 02.4449 | 0.0491 | 0.0259 | -3.1927 ± j 18.544 | 0.1697 | 4.8218 | 0.0130 | 0.3114 |
| **Under excited regime** | | | | | | | | | | |
| GA | MULTI | 05.7922 | 06.3686 | 0.0462 | 0.0013 | -2.7460 ± j 6.6338 | 0.3825 | 4.3941 | 0.0000 | 0.3001 |
| GA | MONO | 01.9529 | 00.1412 | 0.0458 | 0.0052 | -3.5238 ± j 21.876 | 0.1590 | 4.0305 | 0.0234 | 0.3981 |
| PSO | MULTI | 05.2003 | 06.3812 | 0.0750 | 0.0498 | -2.3075 ± j 8.5888 | 0.2595 | 4.6191 | 0.0123 | 0.3035 |
| PSO | MONO | 02.0196 | 01.0644 | 0.0088 | 0.0295 | -3.2726 ± j 20.973 | 0.1542 | 4.7322 | 0.0234 | 0.3027 |

From table results, it can be observed that the use of PSS-GA and PSS-PSO improves considerably the dynamics performances by increasing damping coefficient $\zeta$ and improves stability by decreasing the real part of the poles $\sigma$ under different operating regimes. However optimization by the genetic algorithm in the majority of results obtained very effective compared to the use of particle swarms optimization.

The simulation results shown in figures 13,14 and 15 show the effectiveness of the use of GA mult-objective in comparison with GA mono objective, PSO mono objective, and PSO mult-objective, it can be observed static errors negligible so better precision, and very short setting time so very fast system, and we found that after a few oscillations, the system returns to its equilibrium state even in diffrent regimes operations.

The optimization and simulation results satisfy to show the reliability of the proposed optimization technique GA multi-objective.

## 4. Conclusion

In this article, the PSS parameters optimized using a genetic algorithm and particle swarm optimization applied to powerful synchronous generators exciter voltage control to improve static and dynamic performances of power system.

Genetic algorithm technique optimization allows us to obtain a considerable improvement in dynamics performances and robustness stability of the power system studied. The optimization and simulation results show that the optimization by the genetic algorithm very effective in comparison with the particle swarms optimization

All results are obtained by using our created GUI/MATLAB

### *References*

[1] T.K. Das, and G.K. Venayagamoorthy, "Optimal Design of Power System Stabilizers Using a Small Population Based PSO" IEEE Power Engineering Society General Meeting, 2006.

[2] GHOURAF Djamel Eddine and NACERI Abdellatif ‘’ An Advanced PID-PSS Based Genetic Algorithms Implemented using GUI - MATLAB ‘’,IEEE *Xplore* Proceedings of International Renewable and Sustainable Energy Conference (IRSEC’14),Page(s):411 - 418

[3] Sayed Mojtaba Shirvani Boroujeni, Reza Hemmati, Hamideh Delafkar and Amin Safarnezhad Boroujeni ‘Optimal PID power system stabilizer tuning based on particle swarm optimization’ Indian Journal of Science and Technology Vol. 4 No. 4,pp 379-383

[4] Sumathi N, Selvan MP and Kumaresan N (2007) A hybrid genetic algorithm based power system stabilizer. Int. conf. on intelligent & advanced systems. pp:876-881.

[5] Jiang P, Yan W and Weigu " PSS parameter optimization with genetic algorithms", DRPT 2008, Nanjing China. pp: 900-903.

[6] Yassami H, Darabi A and Rafiei SMR "Power system stabilizer design using strength pareto multiobjective optimization approach", Electric Power Systems Res. 80, 2010, pp838–846.

[7] Dubey M "Design of genetic algorithm based fuzzy logic power system stabilizers in multi machine power system", Int. Conf. on soft computing & intelligent systems, (2007) pp:214-219.

[8] Gi-Hyun Hwang; June-Ho Park; Hyeon Tae Kang; Sungshin,'Design of fuzzy power system stabilizer us ing adaptive evolutionary algorithm', IEEE *Xplore* Proceedings of the International Symposium on Industrial Electronics, 2000

[9] Zhu Jizhong. "Optimization of power system operation ", IEEE Press Editorial Board WILEY, 2009, p 470.

[10] Dhubkarya D.C., Deepak Nagariya, Jay Kumar, "Function Optimization Using Genetic Algorithm by VHDL", Global Journal of Computer Science and Technology, 2010, Vol. 1, Issue 9, pp 73-78.

[11] Melanie Mitchell. ‘'An Introduction to Genetic Algorithms’', Cambridge, Massachusetts London, England: A Bradford Book The MIT Press, 1999. 0−262−13316−4 (HB), 0−262−63185−7 (PB)

[12] Wei Zhang; Weifeng Shi; Jinbao Zhuo "Quantum-PSO based system stabilizer optimization for shipboard power system" ,IEEE *Xplore* Proceedings of 35th Chinese Control Conference (CCC), 2016

[13] F. Schutte, ‘'The Particle Swarm Optimization Algorithm", Structural Optimization, 2005. EGM 6365.

[14] Med. MEKHANET, L. MOKRANI and Med. LAHDEB " Comparison between Three Metaheuristics Applied to Robust Power System Stabilizer Design", ACTA ELECTROTEHNICA Vol51, N 1, 2012, pp 3-10

[15] S.V. SMOLOVIK ‘mathematical modeling Method of transient processes synchronous generators most usual and non-traditional in the electro-energy systems’ PhD Thesis State, Leningrad Polytechnic Institute, 1988, p 213, (translated from Russian).

[16] F. Glover, ‘' Future Paths for Integer Programming and Links to Artificial Intelligence'', Computers and Operations Research , vol 13 (5),pp 533–549, 1986.

[17] K. Y. Lee and M.A. El-Sharkawi, "Modern Heuristic Optimization Techniques with Applications to Power Systems'', IEEE Press Series on Power Engineering, John Wiley & Sons, 2008.

[18] Prasenjit Dey; Aniruddha Bhattacharya; Juhi Datta; Priyanath Das " Parameter tuning of power system stabilizer using a meta heuristic algorithm" ,IEEE *Xplore* Proceedings Second International Conference on Electrical, Computer and Communication Technologies (ICECCT), 2017

[19] J.H. Holland, ‘Adaptation in Natural and Artificial Systems’, University of Michigan Press, 1975.

[20] J. Kennedy, R. Eberhart. ‘Particle swarm optimization.’ In: Proceedings of the IEEE international conference of neural network (ICNN’95), vol. IV; 1995. p. 1942–8.

[21] Yosra Welhazi; Tawfik Guesmi; Chefai Dhifaoui; Hsan Hadj Abdallah " Robust design of multi machine power system stabilizers using multi objective PSO algorithm " 2014 5th International Renewable Energy Congress (IREC), IEEE *Xplore* Proceedings

## APPENDIX

1. Parameters of the used Turbo –Alternator

| *Parameters* | *TBB-200* | *TBB-500* | *BBC-720* | *TBB1000* | Units of measure |
|---|---|---|---|---|---|
| power nominal | 200 | 500 | 720 | 1000 | MW |
| Factor of power nominal. | 0.85 | 0.85 | 0.9 | 0.85 | p.u. |
| $X_d$ | 2.56 | 1.869 | 2.67 | 2.35 | p.u. |
| $X_q$ | 2.56 | 1.5 | 2.535 | 2.24 | p.u. |
| $X_s$ | 0.222 | 0.194 | 0.22 | 0.32 | p.u. |
| $X_f$ | 2.458 | 1.79 | 2.587 | 2.173 | p.u. |
| $X_{sf}$ | 0.12 | 0.115 | 0.137 | 0.143 | p.u. |
| $X_{sfd}$ | 0.0996 | 0.063 | 0.1114 | 0.148 | p.u. |
| $X_{sf1q}$ | 0.131 | 0.0407 | 0.944 | 0.263 | p.u. |
| $X_{sf2q}$ | 0.9415 | 0.0407 | 0.104 | 0.104 | p.u. |
| $R_a$ | 0.0055 | 0.0055 | 0.0055 | 0.005 | p.u. |
| $R_f$ | 0.000844 | 0.000844 | 0.00176 | 0.00132 | p.u. |
| $R_{1d}$ | 0.0481 | 0.0481 | 0.003688 | 0.002 | p.u. |
| $R_{1q}$ | 0.061 | 0.061 | 0.00277 | 0.023 | p.u. |
| $R_{2q}$ | 0.115 | 0.115 | 0.00277 | 0.023 | p.u. |