Shock and Vibration 19 (2012) 1–10
DOI 10.3233/SAV-2012-0702
IOS Press

Design of a nonlinear controller for a rigid-flexible satellite using multi-objective Generalized Extremal Optimization with real codification

I. Mainenti-Lopes, L.C.G. Souza* and F.L. De Sousa
National Institute for Space Research, Space Mechanics and Control Division, SP, Brasil

Abstract. The Attitude Control System (ACS) for Flexible Space Structures (FSS) like rigid-flexible satellite and solar sails demands great reliability, autonomy and robustness. The association of flexible motion and large angle maneuver imply that the FSS dynamics is only captured by complex non-linear mathematical model. As a result, FSS controller performance designed by linear control technique under the hypothesis of rigid dynamic can be degraded. Although vibrations can be suppressed rapidly, the flexibility effect can introduce a tracking error resulting in a minimum attitude acquisition time. On the other hand, faster manoeuvres can excite flexible modes in such a way to make the FSS lose the required pointing accuracy. In the present work, it is shown that a new multi-objective optimization algorithm, called M-GEO$_{real}$ (Multi-objective Generalized Extremal Optimization with real codification), is a good tool to be used in such kind of problems. The M-GEO$_{real}$ is a real coded version of the M-GEO evolutionary algorithm. Its performance on finding the gains of a non linear control law is evaluated through its application to the problem of controlling a large angle attitude manoeuvre of a rigid-flexible satellite.. The satellite non-linear model consists of a rigid central hub with a clamped free flexible beam. The multi-objective approach allows optimizing conflicting objective functions like time and energy. As a result, one can find a trade-off solution (non-dominated solutions). These solutions become available to the designer for posterior choice of an individual solution to be implemented. The non-dominated solutions are represented in the design space (Pareto set) and in the objective functions space (Pareto front). Having in mind the complexity of implementing a control algorithm in onboard satellite computer, this preliminary investigation has shown that the non-linear controller based on the M-GEO$_{real}$ algorithm is a promising technique, since it has satisfied all the ACS requirements. A great advantage of the M-GEO$_{real}$ procedure is its capacity to deal with non-linear system and designing non-linear controller with constant gains facilitating the on board computer implementation.

Keywords: Nonlinear controller, flexible satellite, multi-objective optimization, Generalized Extremal Optimization

1. Introduction

Evolutionary algorithms (EAs) are stochastic methods of optimization that are based on natural process and they are widely used to tackle engineering and scientific optimization problems [1]. This kind of stochastic method employs a population of candidate solutions that is “evolved” during the search as better individuals (new solutions) which are generated from previous ones in the sense that they are closer to the global minimum [2]. The main advantage of the evolutionary algorithms is the capacity to avoid local optimal solutions, allowing searching for the global optimum. In fact, evolutionary algorithms are very robust methods and they are capable to tackle problems

*Corresponding author: L.C.G. Souza, National Institute for Space Research, Space Mechanics and Control Division, Av. dos Astronautas, 1758, S J dos Campos, SP, Brasil. E-mail: gadelha@dem.inpe.br.

ISSN 1070-9622/12/$27.50

with non-linearities in the objective functions. They can easily deal with constrains and their non-linearities, and also deal well with problems that have different kinds of design variables.

The Generalized Extremal Optimization (GEO) algorithm [4], is one of such evolutionary algorithms that have been applied successfully to different kinds of engineering optimization problems, including multi-objective ones [3].

Although GEO and its original multi-objective version, M-GEO have showed good performance to tackle optimization problems, they codify the variables with strings of bits. This characteristic imposes a precision to the variables and this can lead to a sub-optimal solution if the bit coding does not capture the variable optimal values. In order to avoid this limitation, a real code version of GEO, called GEO_{real} , was developed [5]. This new version showed to have better performance than previously versions of GEO when tackling test functions, although the GEO_{real} cannot tackle multi-objective problems.
In this context, a real coded version of M-GEO, called M-GEO_{real}, was developed and its performance compared to M-GEO and NSGA-II [6] algorithms for two multi-objective test functions, ZDT1 and TNK. The results for this performance comparison shown that M-GEO_{real} had better or similar performance than M-GEO and a competitive performance compared to NSGA-II.

In this paper it is shown an application of M-GEO_{real} to a multiobjective non-linear control problem: The optimization of the gains of a non-linear attitude control law, to perform a large angle maneuver of a rigid-flexible satellite [8]. The M-GEO_{real} controller performance and robustness was tested simulating a large angle maneuver, where the non-linear terms of the plant work like perturbations [9]. Results has shown that, not only trade-off solutions for minimizing time and fuel are found for the problem using M-GEO_{real} but it is an algorithm very promising to be implemented in a satellite on board computer, since its non linear control law gains are simple and constant.

2. The M-GEO_{real} algorithms

Multi-objective optimizations problems consist in optimize simultaneously two or more conflicting objectives. Because the objectives are conflicting, it is impossible to obtain one solution that optimizes all objectives. Therefore, in the set of solutions each solution will not optimize one objective without losing optimality in the others. This set of solution in the design space is called Pareto Set and in the objective space it is called Pareto Front. The main goal of an algorithm capable to tackle multi-objective problems is to obtain the Pareto Set and the Pareto Front. Originally the M-GEO_{real} algorithm was developed based on the second algorithm presented in [5], called GEO_{real2}. The main difference between GEO_{real2} and M-GEO_{real} is how each one deals with the best solution. As a mono-objective algorithm, GEO_{real2} stores the best solution along the run and returns only one solution, while M-GEO_{real} stores the non-dominated solutions along the run and, for each new solution, a test is made to determine which solution will be kept and which will ones be discarded.

The following steps show how this test works and it will be called in this paper as Pareto Front Test:

(i) Test if the new solution is dominated by any solution in the stored Pareto Front. It means, if any solution in Pareto Front is at least equal in all objective functions except for one that is better than the new solution. If the new solution is dominated, keep the Pareto Front and go to the step (iii). Otherwise, include the new solution and go to the next step;
(ii) Determine all solutions that are dominated by the new solutions, discarding then from the Pareto Front;
(iii) Finish the Pareto Front Test.

The M-GEO_{real} algorithm was developed to recover the Pareto Front and the Pareto Set maintaining the main characteristic of the GEO algorithm. The following steps describe the M-GEO_{real} algorithm:

(i) Initialize randomly a string of N design variables; calculate the value of all functions $F_m(\boldsymbol{x})$ with this set of variables, where m is the number of objective functions. Store $F_m(\boldsymbol{x})$ in Pareto Front and $\boldsymbol{x}$ in Pareto Set;
(ii) Set the value of the index i to 1;
(iii) Set the value of the index j to 1;

(iv) Generate randomly m weight w_m between 0 and 1, each one associated to each objective function and calculate the adaptability of $\boldsymbol{x}$ given by

$$A_j = \frac{\sum_{k=1}^{m} w_k F_k(x)}{\sum_{k=1}^{m} w_k} \tag{1}$$

where A_j is the adaptability of the j-th variable, and when $j = 0$, A_0 represents the adaptability of the variable value unchanged. That is, $x'_{i0} = x_i$. Therefore, there is a chance to keep the variable value if it is a good value. This is one of the differences between this version and the mono-objective version;

(v) Change the value of the variable x_i using an equation given by

$$x'_{ij} = x_i + N_j(0,\sigma_j)x_i \tag{2}$$

Calculate $F_m(\boldsymbol{x})$ using the value of x'_{ij} instead of x_i and run the Pareto Front Test. Calculate the adaptability of x'_{ij} using Eq. (1), where $N_j(0,\sigma_j)$ is a random number with Gaussian distribution and σ_j is the standard deviation;

(vi) Return value of x_i to the vector $\boldsymbol{x}$, increment value of j, return to step (iv). Repeat this sequence until $j > P$;
(vii) Assign a rank k_j to each x'_{ij} according to the A_j value with j = 0, 1, …, P , where $k_j = 1$ to the best value and $k_j = P + 1$ to the worst value;
(viii) Choose, with uniform probability one, x'_{ij} (including the solution unchanged x'_{i0}), accept this choice with probability equal to $k_j^{-\tau}$. If the choice was accepted, store the chosen x'_{ij}, but do not change the value of x_i yet, and continue to next step. Otherwise, go back to step (viii);
(ix) Increment the index i and go back to step (iii). Repeat this process until $i > N$.
(x) Change each element x_i of the vector $\boldsymbol{x}$ according to the value x'_{ij} chosen in step (vii). Calculate $F_m(\boldsymbol{x})$ using the new vector $\boldsymbol{x}$ and run the Pareto Front Test. Test a stopping criterion. If it is accepted, go to step (xii). Otherwise, test a population restart criterion. If it is accepted, go to step (xi). Otherwise, go back to step (ii);
(xi) Initialize randomly a string of N design variables, calculate all objective function value $F_m(\boldsymbol{x})$ with this set of variables and run the Pareto Front Test. Go back to step (ii);
(xii) Return the Pareto Front and the Pareto Set.

The population restart test is made to increase the algorithm capacity to recover all Pareto Front. In this work, the criterion to restart the population was given by the free parameter rt , which represents the number of restarts along the search. A disadvantage of M-GEO$_{real}$ compared with M-GEO is the increase of free parameters. In M-GEO algorithm, there are only two parameters; the value of τ and rt. In the M-GEO$_{real}$, there are P+3 new free parameters (P standard deviations, the P, τ and rt values). The intention of using several values for the standard deviation for the same variable is to allow the algorithm to search in a greater range of values in a single iteration. Therefore, it is interesting to select high and low values of σ_j. To reduce the amount of free parameters, the following rule was adopted.

$$\sigma_{i+1} = \frac{\sigma_i}{s.i} \tag{3}$$

where $i = 1, 2, \ldots P$ and s is an arbitrary number greater than one. In this work, it was chosen $s = 2$. In that way, it is enough to define σ_1 and all the other values of σ_j will be automatically defined. Therefore, there are as many high values as low values of σ. Now, it is needed to define four free parameters: σ_1, P, τ and rt.

However, M-GEO$_{real}$ algorithm can change all variables per iteration. While M-GEO changes only one bit, that is, it can change just one variable per iteration. Besides, M-GEO chooses one function $F_m(x)$ to calculate the adaptability per iteration. This procedure may lead the algorithm to find solutions at the edge of the Pareto Front. The M-GEO$_{real}$ approach uses a weight sum of the functions. The flowchart of M-GEO$_{real}$ is shown Fig. 1.

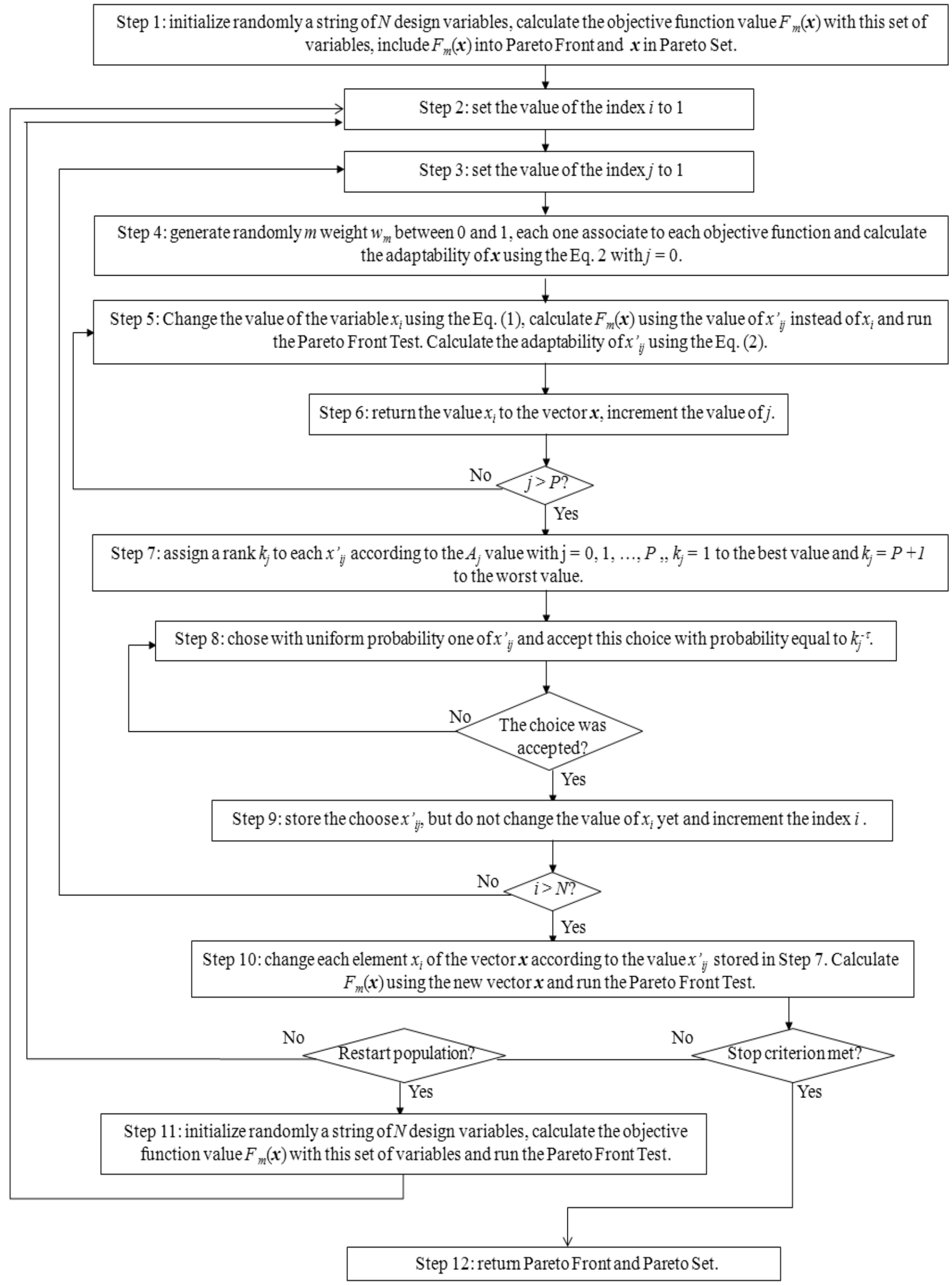

Fig. 1. M-GEO_{real} algorithm flowchart [7].

3. Rigid-flexible satellite model

In this test, the M-GEO_{real} algorithm is applied to design the non-linear attitude control law that optimizes, simultaneously, the time and the energy of the rigid flexile satellite control system to perform an attitude maneuver. The rigid-flexible satellite model consists of rigid central hub with one clamped beam [7], see Fig. 2. The satellite attitude is given by the angular rotation θ of its reference system x_1y_1 with respect the inertial reference system XY.

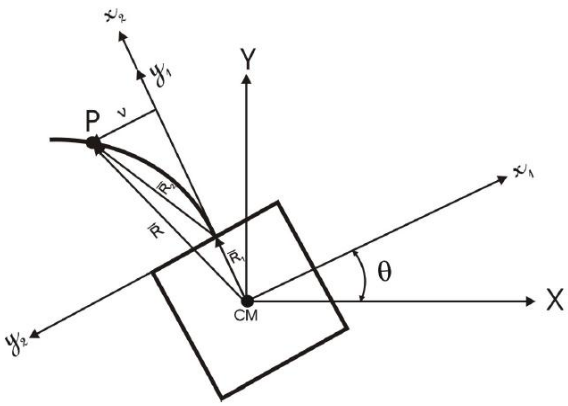

Fig. 2. Rigid-flexible satellite model [7].

In order to model the flexibility of the beam, it was used the Euler-Bernoulli formulation, where ν(y, t) represents the elastic displacement of the beam. Considering only the first vibration mode of the beam ω, the equations of motion that describe the satellite central body rotation angle θ and the elastic displacement of the beam p= ν(y, t) are given by

$$\begin{cases} 2C_3\ddot{\theta} + C_2\ddot{p} + 2C_1 p^2\ddot{\theta} + 4C_1\dot{\theta}p\dot{p} = \xi \\ 2C_1\ddot{p} + C_2\ddot{\theta} - 2C_1\dot{\theta}^2 p + 2C_1\omega^2 p = 0 \end{cases} \tag{4}$$

where C_1, C_2 and C_3 are the following constants

$$C_1 = \frac{\rho A_h}{2}; \qquad C_2 = \rho A_h R_1 \lambda + \rho A_h \mu; \qquad C_3 = \frac{1}{2}\left[\rho A_h \left(R_1^2 l + R_1 l^2 + \frac{l^3}{3}\right) + I_o\right]$$

After some manipulations, the satellite equations of motion in the state space [6] is given by

$$\dot{x}_1 = x_2 \tag{5}$$

$$\dot{x}_2 = \frac{2C_1C_2\omega_1^2 x_3 - 2C_1C_2x_2^2x_3 - 8C_1^2x_2x_3x_4 + 2C_1\xi}{4C_1^2x_3^2 + C_4} \tag{6}$$

$$\dot{x}_3 = x_4 \tag{7}$$

$$\dot{x}_4 = \frac{4C_1C_3x_2^2x_3 + 4C_1^2x_2^2x_3 - 4C_1C_3\omega_1^2x_3 - 4C_1^2x_3^3 + 4C_1C_2x_2x_3x_4 - C_2\xi}{4C_1^2x_3^2 + C_4} \tag{8}$$

Where $C_4 = 4C_1C_3 - C_2^2$, $x_1 = \theta$, $x_2 = d\theta/dt$, $x_3 = p$ and $x_4 = dp/dt$.

4. Simulation of the satellite control law using the M-GEO$_{real}$ algoritm

The non-linear control law [8] is given by

$$\xi = -K_1x_1 - K_2x_2 - K_3x_1x_2 \tag{9}$$

where ξ is the control torque and K_1, K_2 , K_3 are the nonlinear control law gains.

The M-GEO$_{real}$ algorithm obtains the gains K1, K2 and K3 of the nonlinear control law aiming to minimize time and energy during a satellite rotation maneuvers. For each set of gains tried by M-GEO$_{real}$, the integration algorithm is called to calculate the value of time F_1 and energy F_2 using the following equations

$$F_1 = T \tag{10}$$

$$F_2 = \sum_{i=0}^{hT} \left| \left(K_1 x_{1i} + K_2 x_{2i} + K_3 x_{1i} x_{2i} \right) \left(x_{1i} - x_{1i+1} \right) \right| \tag{11}$$

The equations of motion are integrated using a Runge-Kutta algorithm, which step is h, x_{1i} and x_{2i} are the angle and angular velocity of each i-th iteration. The controller optimization is done considering $0 < K_1 < 20000$, $0 < K_2 < 20000$ and $0 < K_3 < 20000$. The satellite parameters values used in the simulation are presented in Table 1.

Table 1

Satellite parameters used in the simulations

Parameters	Description	Value	Unit
A_h	Cross section of the beam	7.5×10^{-4}	m^2
ρ	Alumininum density	2700	kg/m^2
l	Beam length	2.0	M
E	Young's modulus	7×10^{10}	N/m^2
$a_1 l$	Eigen value associated to the beam first mode of vibration	1.878	–
I_h	Moment t of inertia of the beam	1.5625×10^{-9}	m^4
I_o	Satellite's Moment of inertia	1125	kg.m^2
ω	Beam first mode of vibration	18,0001	rad/s
R_1	Half of the central body edge	0.75	M

In order to stress the nonlinear terns of the plant at the end of the control action, the maneuver simulated is from an initial angle of 0^o to a final angle of −90^o. The idea of stressing the non-linear terns of the plant [8] permits to investigate the performance and robustness of the M-GEO$_{real}$ controller. Figure 3 shows the Pareto Front, where it is possible to see the non-dominated solutions found by the algorithm.

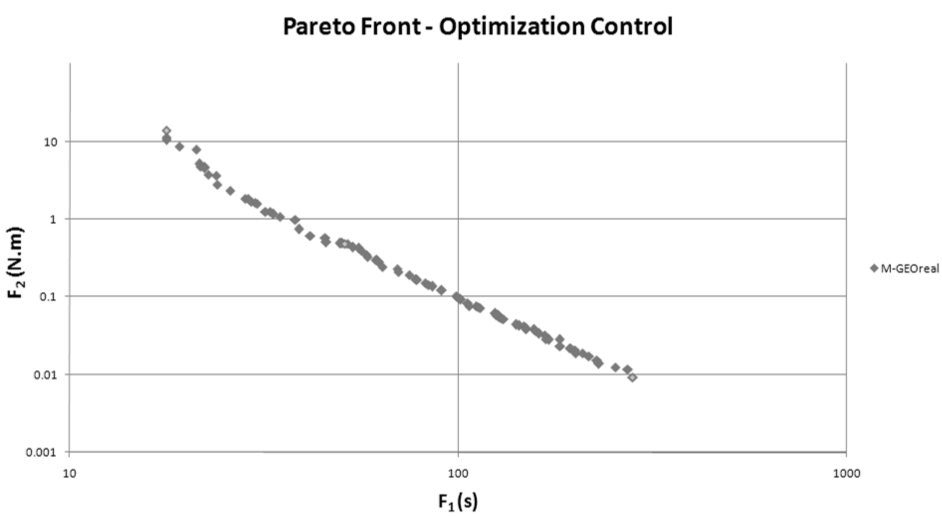

Fig. 3. Pareto front for an attitude maneuver from 0^o to −90^o.

The performance and robustness of the non-linear control law designed by the M-GEO$_{real}$ algorithm is demonstrated getting three sets of F_1 , F_2 , K1, K2 and K3 that are associated with three Pareto Front position, called solution 1, 2 and 3, located at the right, medium and left positions.

From Table 2, which shows the values of the three sets, one observes that time is inversely proportional to the energy, as expected. However, the gains associated with the linear part of the control law increase for quick maneuver, and the gains associated with the nonlinear term of the control law decreases. From these results, one observes that, for quick maneuvers the non linear terms of the equations of motion needs to be tackle by high nonlinear terms of the control law.

Table 2
The Pareto front three set associated with solutions 1, 2 and 3

	Solution 1	Solution 2	Solution 3
F_1	281.12 s	50.94 s	17.73 s
F_2	0.00926041 N.m	0.48416 N.m	13.8841 N.m
K1	1.64053	41.142	53.3337
K2	24.5172	155.887	175.031
K3	214.214	753.913	40.212

Figures 4 and 5 show the angular displacement and the angular velocity associated with the three solutions 1, 2 and 3. From these figures, one observes that the three control law actions reflects the values of the three sets showed in Table 2, that is, more time less energy.

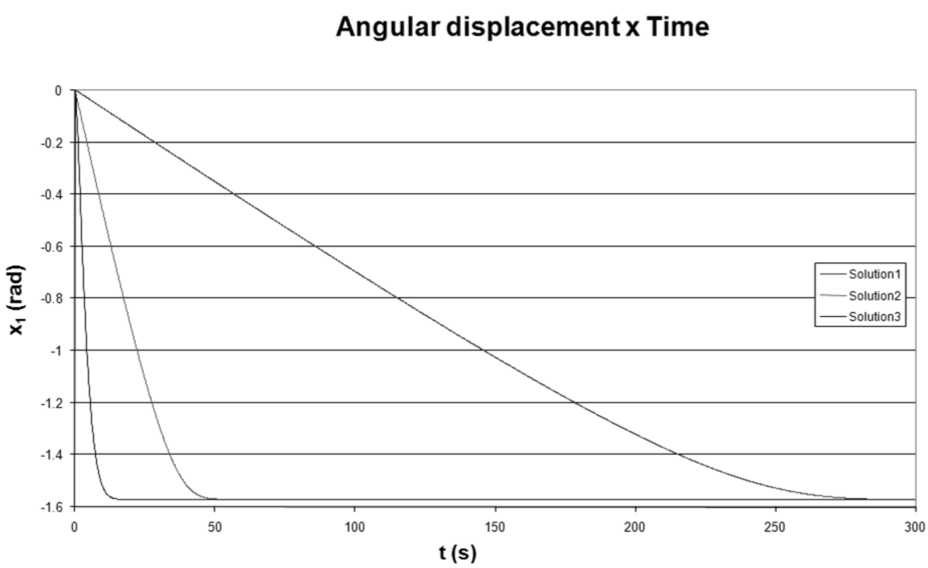

Fig. 4. Angular displacement for the three solutions 1, 2 and 3.

Figure 6 shows the flexible beans deformations for the three solutions 1, 2 and 3. One observes that the maneuver performed slowly (more time) excited less the panels flexibility. This result shows that the M-GEO$_{real}$ algorithm permits to display the Pareto front, from where one can get the appropriated values of the gains that will satisfies the pointing requirements of the maneuvers. For instances, rigid flexible satellite mission with large attitude maneuvers and high pointing requirements must be performed by the non linear control law with gains of the solution 1. If the satellite can be considered as a rigid structure, one can use the gains of solution 3.

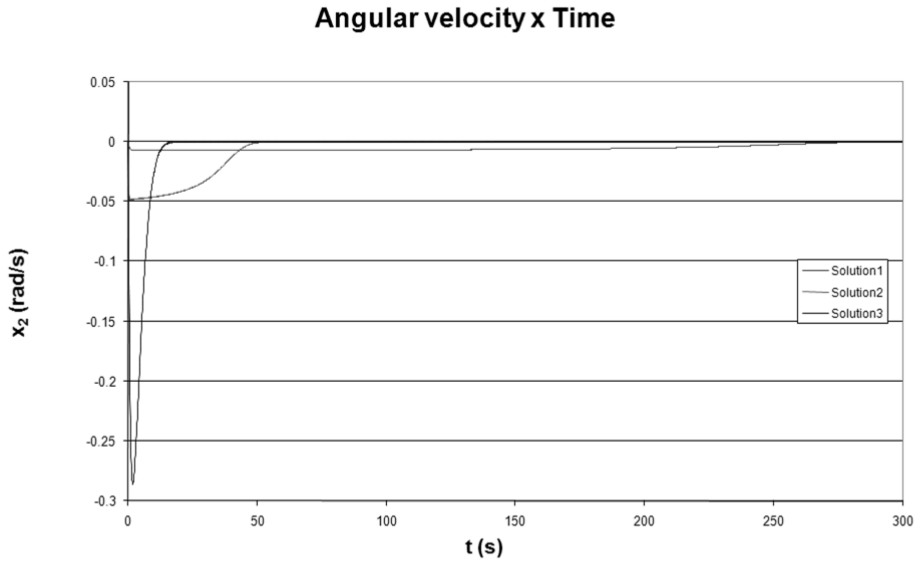

Fig. 5. Angular displacement for the three solutions 1, 2 and 3.

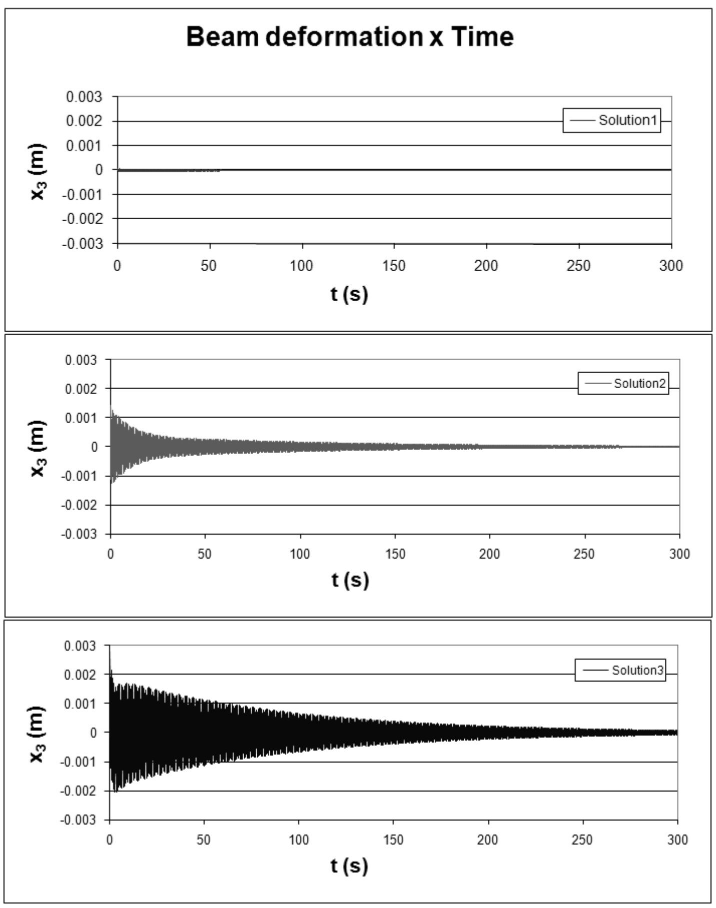

Fig. 6. Flexible beam displacement for the three solutions 1, 2 and 3.

Finally, from Fig. 7, which shows the torques for solutions 1, 2 and 3, one observes that the energy spent decreases from solution 1 to solution 3, as expected.

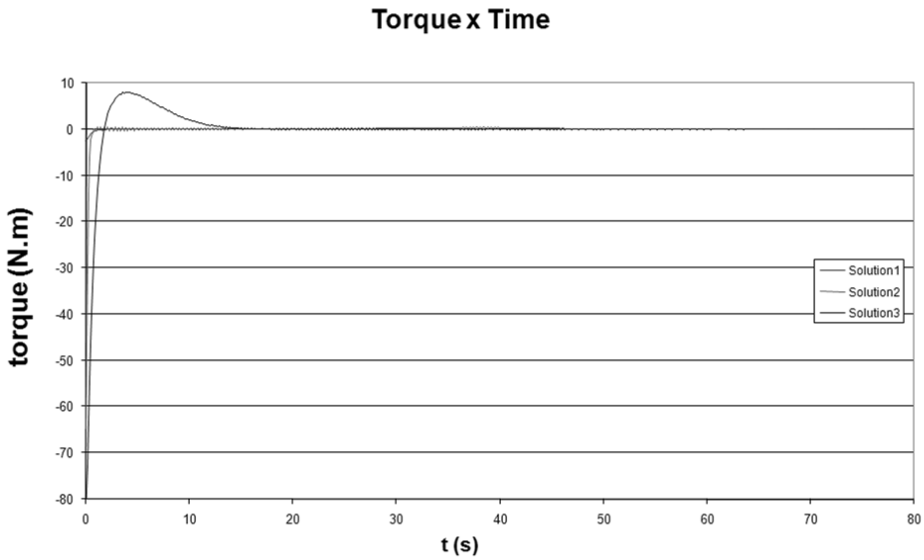

Fig. 7. Torques of the three solutions 1, 2 and 3.

5. Final remarks

The M-GEO$_{real}$ multiobjective evolutionary algorithm was applied to design a nonlinear attitude control law, to perform a large angle maneuver of a rigid-flexible satellite, optimizing time and energy, simultaneously.

The performance and robustness of the M-GEO$_{real}$ controller was demonstrated using three solutions of the Pareto Front to perform the big maneuver control. The non-linear term of the rigid-flexible satellite model were stressed in order to investigate the non linear controller performance and robustness.

From the Pareto front sets, one observes that time is inversely proportional to the energy, as expected. However, the gains associated with the linear part of the control law increase for quick maneuver, and the gains associated with the nonlinear term of the control law decreases. As a result, for quick maneuvers, the non linear terms of the equations of motions needs to be tackle by high nonlinear terns of the control law.

The simulations have show that the angular displacement and velocity reflect the control law gains actions associated with the Pareto front set, that is, more time less energy. From the flexible beans deformations for the three solutions, one observes that the maneuver performed slowly (more time) excited less the panels flexibility. Therefore, quick vibrations reductions introduce a tracking error resulting in a minimum attitude acquisition time. On the other hand, faster maneuvers can excite flexible modes in such way to lose pointing accuracy.

As for satellite on board compute implementation, the M-GEO$_{real}$ procedure is very promising, since the non linear control law gains designed, besides optimizing simultaneously the time and energy, they keep simplicity of constant gains.

References

[1] L.D. Davis, K. De Jong, M.D. Vose and L.D. Whitley, *Evolutionary Algorithms*, Mathematics and Its Applications, Vol. 111, Springer-Verlag, Berlin, Germany, 1999.

[2] T. Bäck and H.P. Schwefel, An overview of evolutionary algorithms for parameter optimization, *Evolutionary Computation* **1**(1) (1993), 1−23.

[3] R.L. Galski, F.L. De Sousa and F.M. Ramos, *Application of a New Multiobjective Evolutionary Algorithm to the Optimum Design of a Remote Sensing Satellite Constellation*, Proceedings of the 5th International Conference on Inverse Problems in Engineering: Theory and Practice, Cambridge, UK, 2005.
[4] F.L. De Sousa, F.M. Ramos, P. Paglione and R.M. Girardi, New stochastic algorithm for design optimization, *AIAA Journal* **41**(9) (2003), 1808 –1818.
[5] I. Mainenti-Lopes, F.L. De Sousa and L.C.G. Souza, *The Generalized Extremal Optimization with Real Codification*, Proceedings of the EngOpt − International Conference on Engineering Optimization, Rio de Janeiro, Brazil, 2008.
[6] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, *A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, April, 2002.
[7] I. Mainenti-Lopes, L.C.G. Souza, F.L. De Sousa and A.P.C. Cuco, *Multi-objective Generalized Extremal Optimization with real codification and its application in satellite attitude control*, in: 19th International Congress of Mechanical Engineering − COBEM, Gramado − RS. Brasil, 2009.
[8] R.W. Rietz and D.J. Inman, Comparison of linear and nonlinear control of a slewing beam, *Journal of vibration and control* **6**(2) (2000), 309−322.
[9] L.C.G. Souza and R.G. Gonzales, Satellite simulator control system design using SDRE method, *Journal of Mechanics Engineering and Automation* **1** (2011), 123−132.