Automatic Satellite Sun Sensors Placement Using Multi-Objective Genetic Algorithm

Amanda Leonel Nascimento
and André Rodrigues Simões
Fundação de Ciência, Aplicações e Tecnologia
Espaciais (FUNCATE)
São José dos Campos, Brasil
Emails: amandaleonel@gmail.com
andre.simoes@inpe.br

Fabrício Galende Marques de Carvalho
and Paulo Nolberto dos Santos Alarcón
Instituto Nacional de Pesquisas Espaciais (INPE)
ETE-Divisão de Sistemas Espaciais
São José dos Campos, Brasil
Emails: fabricio.galende@inpe.br
paulo_ alarcon13@yahoo.com.br

***Abstract*—This work deals with satellite sun sensor placement using genetic algorithms. For a simplified but realistic problem scenario, this problem is solved and many simulation results are shown. The proposed methodology, which relies on numerical shadow analysis and multi-objective optimization, is discussed and a typical satellite design problem, that is frequently solved by a try-and-error approach, is converted into an optimization problem that can be solved automatically.**

Keywords—Genetic algorithms, satellite design, attitude control subsystem.

I. Introduction

During almost all space missions, the design team is faced with many trade studies and very challenging optimization problems. These problems can be experienced in early mission phases, such as in feasibility study, or in late stages like mission operations and support. Moreover, as the time progresses and the mission life cycle evolves, the nature of these problems gradually changes from the need for a general conceptual description to a detailed system/subsystem design, mission operation strategy and spacecraft disposal [1]. A feasible approach to deal with these problems is to use genetic algorithms (GAs), which are global randomized optimization tools that are inspired by the mechanics of natural selection [2], [3].

The idea of using GAs during space mission life cycle is not new and can be found in many papers. In [4], GAs are used in spacecraft conceptual design and launch vehicle selection during feasibility study, moreover, this paper prescribes an iterative preliminary design process where tasks like evaluation and design changing are replaced by a GA search tool. In [5], the problem of spacecraft sun sensor (SS) placement, which can be regarded as a detailed design problem, is treated in a simplified scenario where the optimizer searches for some sensor set configuration in a way to ensure a full sky coverage by at least four SS. Concerning operation mission phases, such as satellite constellation operations, in [6] GAs were applied to constellation maneuver planning, where satellite trajectories are optimized in a way to avoid collisions, save and equalize fuel consumption and reduce maneuver duration. Regarding attitude maneuvers involving a single spacecraft, [7] proposes the utilization of GAs in attitude path planing when the spacecraft has to deal with geometric, timed and dynamic constraints.

In this paper, we use genetic algorithms as tools to automatically select the number, position and orientation of SS used in a simplified satellite Attitude and Orbit Control Subsystem (AOCS) design. When compared to [5] and [8] (which is based on traditional constrained optimization methods), our paper has the following original contributions:

- The shadow analysis, which is based on a realistic spacecraft operation environment, is explicitly treated (i.e. the shadow caused by any spacecraft part, such as payloads and solar panels, are not ignored);
- Not only the position but also the sensor orientation problem is treated;
- Instead of using a fixed number of SS per algorithm run, our work also deals with the number of sensors optimization, which characterizes an multi-objective optimization problem with conflicting goals.

The following textual structure applies to this paper: Section II states the sun sensor placement problem based on the need for sun direction determination and taking into account fault tolerance, moreover, the shadow analysis process is briefly described; Section III describes the specific genetic algorithm design that is applicable to this work; Section IV shows simulation results for two different problem scenarios; The paper is finished in Section V, where overall conclusions are made and directions for further works are discussed.

II. Problem Description

Many artificial satellites require some kind of attitude determination and control. This characteristic is typically due to power generation needs (e.g. to point solar arrays towards the Sun), payload data labeling, thermal constraints and so on [9], [10]. To perform such a task, which is required by the attitude controller, the satellite can rely on measurements from equipments like star sensors (that are very expensive, complex and generally require some sort of initial attitude determination), from cheaper and simpler equipments, such as magnetometers and sun sensors, or from a combination of both. Moreover, once the architecture has been chosen, the designer

978-1-4799-3194-1/13 $31.00 © 2013 IEEE
DOI 10.1109/BRICS-CCI-CBIC.2013.73

shall perform a careful analysis, concerning the sensor set configuration, to ensure that attitude determination is always possible provided that the correct operation condition for the sensor set is satisfied.

Concerning attitude determination, a simple geometric method called TRIAD, which relies on the sun vector determination, is typically used by many satellites [9]. As a consequence, sun vector calculation is directly related to attitude determination.

The easiest and cheapest way to determine the Sun direction in space is to perform the reading of three coarse sun sensors measurements and to execute a change of basis to compute the sun direction in the spacecraft coordinate frame. So, all the three sensors shall be illuminated simultaneously and their orientation vector shall form a basis of a three-dimensional (3D) space. Moreover, because of the fact that no single point of failure shall exist, the configuration design must be performed in a way that at each time instant there are at least four illuminated sensors. However, the need for a single point of failure fault tolerance increases the potential number of sensors to assure a full sky coverage and, consequently, the spacecraft cost (i.e. number of computer interfaces, harnesses, geometrical dimensions, launcher costs, etc.).

The scope of our work is the utilization of GAs in the process of finding possible sensor set configurations that assures full sky coverage in a two-dimensional (2D) space. In spite of being a simplified scenario, it is a realistic one and is equivalent of having a spacecraft in a sun-synchronous orbit with sun sensors placed on one plane.

Figure 1 shows a schematic diagram which maps the 3D problem to the simplified 2D problem and Fig. 2 shows the notation used throughout the paper to represent a simple sun sensor orientation on the satellite surface. It must be pointed out that, throughout this work, each sun sensor is assumed to be a cosine detector with 180 deg field of view (FOV).

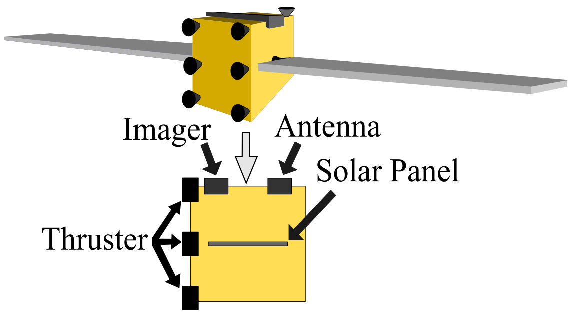

Fig. 1. Mapping the 3D problem to 2D simplified problem.

A. Sensor Placement and Shadow Analysis

To ensure a full sky coverage in a plane, at least 2 sensors must be illuminated at any time. This is the primary objective of the 2D optimization problem. Secondly, a single point of failure is not desired, so, the number of simultaneously illuminated sensors is increased by one (3 total). The last problem is to reduce the total number of sensors, placed on allowed spacecraft surfaces, required to achieve these goals.

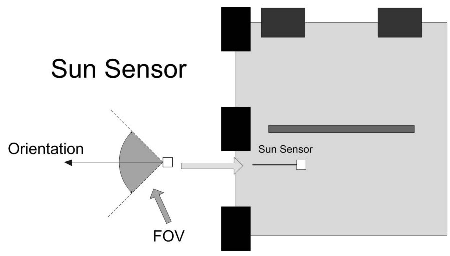

Fig. 2. Sun sensor representation.

The analysis which is performed to ensure full sky coverage is done through the shadow analysis. To do so, it was assumed that the Sun was a point source of light at infinity (angular radius almost equal to zero degrees), a sensor placed outside the spacecraft is not allowed and any sensor inside or exactly on the surface of a spacecraft object is completely obscured. In regard to computational implementation, the main inputs to this analysis are:

- Geometric characteristics of the satellite, including all possible objects placed on the spacecraft body that can cause a sun ray obstruction;
- Number of sun sensors used;
- Sun sensor parameters: Position, orientation, FOV, resolution and type;
- Angular step used to vary the Sun position.

After a candidate configuration solution is proposed, the analysis returns a matrix that relates the number of sensors to the sky coverage. For instance, the first row in Table I means that 100 degrees are uncovered by sensors due to lack of field of view, shadow or placement in a forbidden region (e.g. inside or on the surface of some instrument); the fourth row means that 106 degrees of the sky are simultaneously covered by 3 sensors. It can be noted that addition of all elements in the second column must always result in 360 degrees.

TABLE I. EXAMPLE OF SHADOW ANALYSIS OUTPUT

Number of Sensors	Angular Coverage
0	100
1	30
2	54
3	106
4	70
5	0
6	0
7	0

Still concerning the computational aspects of the shadow analysis problem, it must be emphasized that the following parameters have a significant impact on the algorithm performance:

- Sun step: This is directly related to the time required to perform the analysis and with the numerical error that arises when an object border or sensor FOV boundary is encountered. In our implementation, if δ_{Sun} denotes the Sun step ($\delta_{Sun} > 0$), n_{oi} represents the number of objects that can cause sun obstruction for the sensor i

and N_{SS} represents the total number of sensors for a given configuration, the analysis output will have an uncertainty less than or equal to $\sum_{i=1}^{N_{SS}} \delta_{Sun} * (n_{oi}+1)$. Therefore, the smaller δ_{Sun} the smaller the error and the slower the analysis. Moreover, in spite of being an analysis design parameter, the choice of a value for δ_{Sun} is driven by the sensor resolution (i.e. $\delta_{Sun} \leq$ (sensor resolution))

- Object shapes: The more complex the object shapes are, the slower the analysis is performed. A very practical and feasible approach is to use an envelope composed by simple shapes (i.e. rectangles, triangles and circles) to represent an object with an irregular shape. For the case of moving parts (i.e. a solar panel that spins around some spacecraft axis), the envelope can circumscribe the region where the movement takes place.

In the next section it will be explained how genetic algorithms can be used to optimize the number, the position and the orientation of the sun sensors.

III. Optimization Using Genetic Algorithms

Every optimization problem that makes use of genetic algorithms is characterized mainly by: a) The way the solutions are represented in the search space (coding); b)The objective function (fitness); c)The number of solutions that are evaluated at each step (population size); d) Reproduction, crossover and mutation parameters. In the following subsections, these design parameters are defined and explained for the case where the satellite side is represented by a square (1.45m x 1.45m) and for two obstacle configuration cases: A) Simple satellite with a locked solar panel (1.41m x 0.05m) and no thrusters and other objects (see Fig. 3); B) A satellite with a moving solar panel (1.00m x 0.05m), three thrusters in one side (0.20m x 0.30m), a imager (0.30m x 0.20m) and one antenna (0.30m x 0.20m) see Fig. 4).

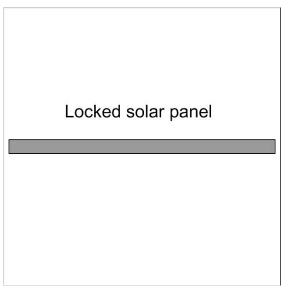

Fig. 3. Satellite face with a locked solar panel.

A. Representation

In this work a binary code scheme was used and the following parameters where used to make up the chromosome: a) Sensor position (x_{ss} and z_{ss}, for cartesian coordinates, or θ_{ss} and ρ_{ss}, for polar form); b) sensor orientation in degrees (Θ_{ss}); c) Activation flag (single bit for the multi-objective optimization case).

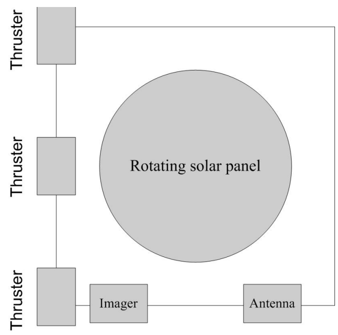

Fig. 4. Satellite with thrusters, rotating solar panels, imager and antenna.

Cartesian coordinates were used in the single-objective optimization. For this case, the chromosome length per sensor was equal to 24bits (7 for x_{ss}, 7 for z_{ss} and 10 bits for Θ_{ss}). This design assures an spacial resolution of 1.1cm and an angular resolution of $0.35\,\mathrm{deg}$. Moreover, the number of sensors was equal to seven, so the chromosome length was equal to 168 bits.

Concerning the multi-objective optimization, a polar coordinate representation was used. The reason to do this is because it is straightforward to deal with the constraint of not putting any sensor inside the region where the solar panels are moving. In this case, chromosome length per sensor was equal to 21 bits (4 bits for ρ_{ss}, 8 bits for θ_{ss}, 8 bits for Θ_{ss} and 1 bit for the activation flag). Such configuration provided a position resolution equal to of 2.8 cm and an angular resolution equal to $1.4\,\mathrm{deg}$. The number of sensors among solutions could vary from 0 to 15, so the chromosome length was equal to 315 bits.

B. Fitness Function

For the single-objective optimization case, the fitness function consisted of a weighted sum involving the output from the shadow analysis. Mathematically, it can be expressed as follows:

$$\mathcal{F}_{\mathcal{S}} = \sum_{i=0}^{N_{SS}} (w_{si} \times \varphi_i) \tag{1}$$

Where:

- $\mathcal{F}_{\mathcal{S}}$: Fitness function for single-objective optimization;
- N_{SS}: Maximum number of available sensors;
- w_{si}: Weight related to i simultaneously illuminated sensors (coverage);
- φ_i: Total angular interval with i illuminated sensors.

The fitness function used in the multi-objective optimization was a weighted average of two functions [11], being the first one equal to that used in the simple optimization, and the second one related to the number of sensors in the solution in a way to penalize configurations with a large number of sensors (i.e. more than 8 sensors) or that will never assure a full sky coverage or single point of failure fault tolerance (e.g. solutions with 0 to 5 sensors). Mathematically, the fitness function for this case is given by.

$$\mathcal{F}_{\mathcal{M}} = \frac{W_S \times \mathcal{F}_{\mathcal{S}} + W_N \times \mathcal{F}_N}{W_S + W_N} \quad (2)$$

$$\mathcal{F}_{\mathcal{N}} = w_{Ni} \quad (3)$$

Where:

- $\mathcal{F}_{\mathcal{M}}$: Fitness function for multi-objective optimization;
- $\mathcal{F}_{\mathcal{N}}$: Fitness function related to the number of sensors.
- W_S: Weight related to the sky coverage fitness function component;
- W_N: Weight related to the number of sensors fitness function;
- w_{Ni}: Weight related to the number of sensors for a configuration with i sensors.

Because the fact that ensure a full sky coverage is considered more important than reducing the total number of sensors $W_S = 2 * W_N$.

It is also important to point out that the choice of the weights, which are related to sky coverage and maximum number of sensors, is driven by AOCS requirements and design constraints. For example, suppose that the AOCS computer of the spacecraft has at most 8 interfaces that are to be used to collect data from sun sensors and that double point of failure tolerance is a primary subsystem requirement. For this single optimization problem, a candidate vector for $W_{si} = [w_{s0} \ w_{s1} \ w_{s2} \ \ldots \ w_{s8}]^T$ is $W_{si} = [0 \ 1 \ 5 \ 10 \ 15 \ 10 \ 5 \ 1 \ 0]^T$. If the AOCS computer was not yet designed and if there are some flexibility to the maximum number of sensors (e.g. at most 11), multi-optimization can be used with $W_{si} = [0 \ 1 \ 5 \ 10 \ 15 \ 15 \ 15 \ 15 \ 15 \ 15 \ 15 \ 15]^T$ and w_{Ni} given by

$$w_{Ni} = \begin{cases} 900 \times i & \text{if } 0 \le i \le 6 \\ 5400 & \text{if } 7 \le i \le 8 \\ (11 - x) \times 1800 & \text{if } 9 \le i \le 11 \end{cases} . \quad (4)$$

Regarding the shadow analysis output, since coarse sun sensors typically present resolution of a couple of degrees [9], we considered $\delta_{Sun} = 1deg$ a good trade-off between the expected numerical error and the computational time needed to achieve the stop condition.

C. Population Size

In this paper, the population size was fixed to be equal to 100. Some preliminary tests were performed with population size varying from 100 to 300 chromosomes but no relevant improvement was observed.

D. Reproduction, Crossover and Mutation

Two selection methods were used to implement reproduction. For the case of single-objective optimization and locked solar panel case, a simple roulette wheel scheme (with and without elitism) was used. For the multi-objective optimization and the satellite with many obstacles, a roulette wheel with fitness scaling was applied.

A single point crossover was used and the crossover rate, which determines the generation gap, was fixed equal to 0.98, while mutation rate was fixed to be equal to 0.002.

In all the simulations the stop condition was the maximum number of generations equal to 1000.

The following section presents some simulation results for both single and multi-optimization problems.

IV. Simulation Results

A. Satellite With Locked Solar Panel

Regarding the single-objective optimization problem, in 20 simulation runs (10 with and 10 without elitism) the algorithm was able to find solutions that ensure a full sky coverage with single point of failure fault tolerance. Table II shows the average value for 10 simulations runs for the case were elitism was used and with a weight set $W_{si} = [5 \ 10 \ 19 \ 35 \ 19 \ 10 \ 5 \ 0]^T$ (the case without elitism performed almost the same).

TABLE II. Average sky coverage for 10 runs for the case with locked solar panel, single objective and elitism.

Number of illuminated sun sensors	Angular Coverage
0	0
1	$0.3°$
2	$2.8°$
3	$355.6°$
4	$1.3°$
5	0
6	0
7	0

REMARK: When the shadow analysis was repeated with $\delta_{Sun} = 0.1deg$ for the configurations that resulted from the optimization process, the uncovered area almost disappeared, so the remaining uncovered area in the original results were due to the shadow analysis and not to the GA optimization process itself.

For the multi-optimization problem, 10 runs were performed and elitism was used. Similar results were achieved but with a reduced number of sensors (6 in all simulation runs). Figure 5 illustrates the boundaries of the multi-objective function while Table III illustrates the average result concerning sky coverage.

TABLE III. Average sky coverage for 10 runs for the case with locked solar panel, multiobjective and elitism.

Number of illuminated sun sensors	Angular Coverage
0	$0.1°$
1	$0.3°$
2	$4.8°$
3	$354.0°$
4	$0.8°$
5	0
6	0

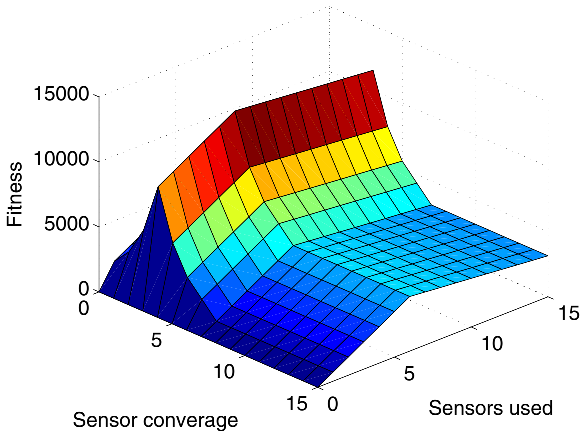

Fig. 5. Fitness function for the satellite with locked panel and multi-optimization.

REMARK: The uncovered 0.1deg shown in Tab. III was due to a singe run that presented both 1deg uncoverd neither by 1 nor 2 sensors. After the simulation was repeated with $\delta_{Sun} = 0.1deg$, these uncovered area almost disapeared.

B. Satellite with thrusters, imager, antenna and rotating solar panel

The previous section showed that the adopted approach was quite effective in dealing with a simple problem that could be solved manually. In this section simulation results are shown for the scenario illustrated by Fig. 4, which constitutes a problem that can not be solved easily by a designer without using too many sensors.

For this problem, in many runs tournament selection and a elitism caused premature algorithm convergence without having any solution with a tolerable single point of failure (i.e. with an uncovered region bellow the expected numerical error limit). When elitism was eliminated and a simple roulette wheel scheme was used, the algorithm experienced frequent losses of the best solutions and, in contrast with the previous simulation scenarios, did not converged at all. This was noted in both simple and multi-objective optimization problem. Moreover, it should be pointed out that selecting the number of sensors for the single optimization problem to be executed was not a convenient task.

To overcome the experienced difficulties, the selection algorithm was changed to include a linear fitness scaling scheme [3]. So, the roulette wheel slots were filled according to the following scaled fitness values:

$$\mathcal{F}_{\mathcal{MS}i} = 2 \times \left(1 + \frac{\mathcal{F}_{\mathcal{M}i} - \mathcal{F}_{\mathcal{M}MAX}}{\mathcal{F}_{\mathcal{M}MAX} - \mathcal{F}_{\mathcal{M}MIN}}\right) + \epsilon \qquad (5)$$

where $\mathcal{F}_{\mathcal{M}i}$ is the original fitness of the solution i, $\mathcal{F}_{\mathcal{MS}i}$ is the resulting scaled fitness value, $\mathcal{F}_{\mathcal{M}MAX}$ and $\mathcal{F}_{\mathcal{M}MIN}$ are the maximum and the minimum unscaled fitness at the generation, respectively, and $\epsilon = 0.01$ was used to avoid numerical problems during selection.

Figure 6 illustrates the surface for the unscaled fitness function that was used in the multi-objective optimization. Note that the maximum values are achieved by solutions with 6 to 8 sensors and with a full sky coverage by three or more sensors.

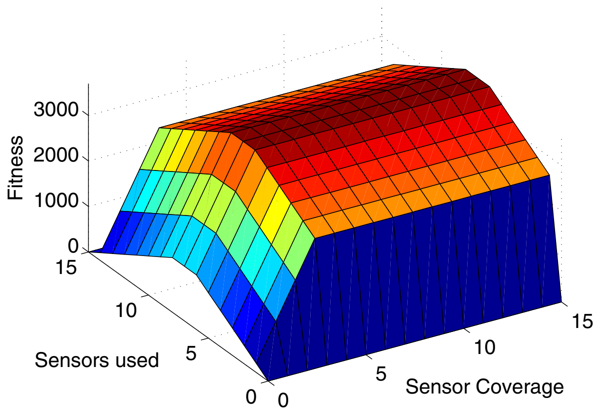

Fig. 6. Multi-objective fitness function for the satellite with rotating solar panels, thrusters, imager and antenna.

Table IV illustrates the coverage of the best solution in 20 runs, Table V illustrates the average values and Fig. 7 shows the fitness function evolution for the best solution. Satellite configuration at the beginning and by the end of the best run is also shown in Figs. 8 and 9.

TABLE IV. BEST SKY COVERAGE FOR THE MULTI-OBJECTIVE OPTIMIZATION REGARDING THE SATELLITE WITH ROTATING SOLAR PANELS, IMAGER, THRUSTERS AND ANTENNA.

Number of illuminated sun sensors	Angular Coverage
0	0°
1	0°
2	0°
3	281°
4	79°
5	0°
6	0°

TABLE V. AVERAGE SKY COVERAGE OF 20 RUNS FOR THE MULTI-OBJECTIVE OPTIMIZATION REGARDING THE SATELLITE WITH ROTATING SOLAR PANELS, IMAGER, THRUSTERS AND ANTENNA.

Number of illuminated sun sensors	Angular Coverage
0	0°
1	0°
2	18.57°
3	252.57°
4	82.58°
5	6.27°
6	0°

Note that the proposed modification allowed us to achieve a full sky coverage with 2 sun sensors in all the simulations. Moreover the remaining area uncovered by at least 3 SS lied within the acceptable numerical error boundaries. In almost all the cases, when the shadow analysis was performed with a reduced step in the obtained solution, the uncovered area

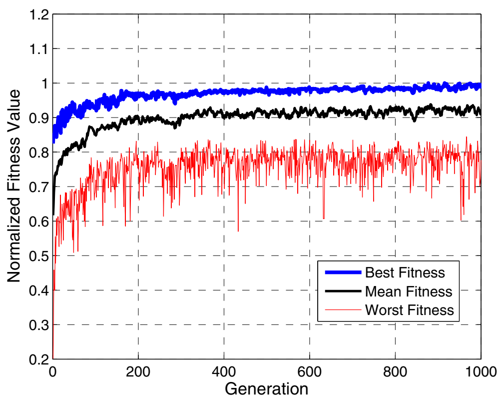

Fig. 7. Evolution of the fitness function for the best solution.

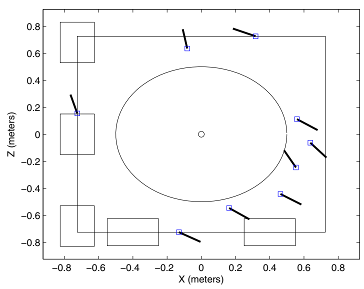

Fig. 8. Best satellite configuration by the beginning of the simulation.

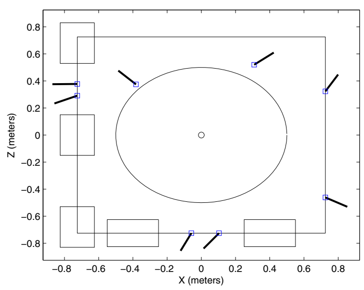

Fig. 9. Best satellite configuration by the end of the simulation.

decreased significantly, showing that the optimization process worked well and that the remaining uncovered spaces were mainly due to numerical errors that can be controlled.

It is also important to point out that for many cases where the reduction of δ_{Sun} did not eliminated the uncovered sky region, a single manual changing of the resulting final best solution sufficed to provide a full sky coverage with single point of failure fault tolerance.

V. Conclusion

This paper treated the satellite sun sensor placement problem as a multi-objective optimization problem that is possible to be solved by genetic algorithms in an automatic or almost automatic fashion.

The proposed methodology is well suited to be applied during the AOCS detailed design phase, when the designer looks for some sun sensor set configuration in a way to ensure the sun vector direction determination. Therefore, it becomes easy to map functional requirements (e.g. full sky attitude determination) and design constraints (e.g. single point of failure fault tolerance) to some specific design choice through an automatic optimization process. For any specific satellite design, the shape of multi-objective fitness function (e.g. maximum value for solutions containing between 6 to 8 sensors) and other design parameters that serve as inputs to the shadow analysis (e.g. sensors with a smaller FOV) completely drive the behavior of the automatic optimization. Moreover, this approach is quite effective in reducing the time required by such an analysis and also the spacecraft total costs related to the number of sensors, electrical interfaces, and so on.

The application of this methodology in a more complex simulation scenario (3D problem with an increased number of obstacles) and comparisons with other optimization approaches, such as particle swarm optimization [12], are currently under work.

Acknowledgment

This work was partially supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) through the project grant number 560131/2010-0.

The authors also would like to thank the CNPq project coordinator, Doctor Adenilson Roberto da Silva by his valuable support.

References

[1] J. R. WERTZ and W. J. LARSON, *Space Mission Analysis and Design*, 3rd ed. Dordrecht, Netherlands: Kluwer Academic, 1999.

[2] J. H. Holland, *Adaptation in natural and artificial systems*. Cambridge, MA, USA: MIT Press, 1992.

[3] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[4] T. Mosher, "Spacecraft design using a genetic algorithm optimization approach," *Proc. of the 1998 IEEE Aerospace Conference*, pp. 123–134, 1998.

[5] B. Jackson and B. Carpenter, "Optimal placement of spacecraft sun sensors using stochastic optimization," in *Proceedings of the 2004 IEEE Aerospace Conference*, vol. 6, 2004, pp. 3916–3923 Vol.6.

[6] S. Li, R. Mehra, R. Smith, and R. Beard, "Multi-spacecraft trajectory optimization and control using genetic algorithm techniques," *Proc. of the 2000 IEEE Aerospace Conference*, vol. 16, pp. 99–108, 2000.

[7] R. P. Kornfeld, "On-board autonomous attitude maneuver planning for planetary spacecraftt using genetic algorithms," *Proc. of the 2003 AIAA Guidance, Navigation and Control Conference*, 2003.

[8] J. C. Springmann and J. W. Cutler, "Optimization of directional sensor orientation with application to photodiodes for spacecraft attitude determination," in *Proceedings of the 23rd AAS/AIAA Spaceflight Mechanics Meeting*, 2013.

[9] J. R. Wertz, *Spacecraft Attitude Determination and Control*, 1st ed. Torrance, CA: Microcosm Inc., 1978.

[10] M. Sidi, *Spacecraft Dynamics and Control: A Practical Engineering Approach*. Cambridge University Press, 2000.

[11] A. Konak, D. W. Coit, and A. E. Smith, “Multi-objective optimization using genetic algorithms: A tutorial,” *Reliability Engineering and System Safety*, pp. 992–1007, 2006.

[12] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of the 1995 IEEE International Conference on Neural Networks*, vol. 4, November 1995, pp. 1942–1948.