JOTA manuscript No.
(will be inserted by the editor)

Combining Global and Local Strategies to Optimize Parameters in Magnetic Spacecraft Control via Attitude Feedback

Renato Bruni · Fabio Celani

Received: date / Accepted: date

Abstract The attitude control of a spacecraft using magnetorquers can be obtained by using attitude feedback, instead of state feedback, with the advantage of not requiring the installation of attitude rate sensors, thus saving in cost, volume, and weight. In this work an attitude feedback with four design parameters is considered. The practical determination of appropriate values for these parameters is a critical open issue. We propose here to search for the parameters' values which minimize the convergence time to reach the desired attitude. Such a systematic approach has several advantages but requires to overcome a number of difficulties to be realized. First, convergence time cannot

Renato Bruni, corresponding author

Dep. of Computer Control and Management Engineering "Antonio Ruberti",

Sapienza University of Rome, Roma, Italy. e-mail: bruni@dis.uniroma1.it

Fabio Celani

School of Aerospace Engineering, Sapienza University of Rome, Roma, Italy.

e-mail: fabio.celani@uniroma1.it

be expressed in analytical form as a function of these parameters. Therefore, we develop a solution approach based on derivative-free optimization algorithms. Secondly, design parameters may range over very wide intervals. As a consequence, the feasible set cannot be explored densely in reasonable time. Thus, we propose a fast probing technique based on local search to identify which regions of the search space have to be explored densely. Thirdly, convergence time depends also on the initial conditions of the spacecraft, which are not known in advance. Hence, we formulate a min-max model to find robust parameters, namely parameters aiming at minimizing convergence time under the worst initial conditions.

Keywords Derivative-free Optimization · Attitude Control · Min-Max Formulations · Strategy Integration · Magnetorquers

Mathematics Subject Classification (2000) 90C26 · 90C90 · 93D15

1 Introduction

The attitude control of a spacecraft that uses magnetorquers as torque actuators is a very important task in astronautics. Many control laws have been designed for this task, and a survey of various approaches is in [1]. Most of them require measures of attitude, attitude rate, and geomagnetic field; however, control algorithms that do not require measures of attitude rate have been proposed, too. Such algorithms have the important practical advantage of not requiring the installation of attitude rate sensors on the spacecraft, thus saving in cost, volume, and weight. In particular, [2] proposes a feedback control law that requires only measures of spacecraft attitude and of the geomagnetic field, which is inspired by a similar control law presented in [3]. The first work contains a proof that attitude stabilization is achieved when the design parameters are positive. However, numerically different (positive) values for the design parameters produce very different behaviors, and the practical determination of appropriate values for these parameters is a critical open issue. Note that this is not a classical optimal control problem, since the design parameters are not control inputs. On the contrary, this is an example of the many problems with the following features: 1) a control algorithm is available, but it contains some parameters that should be determined before applying the algorithm; 2) these parameters have a deep influence on the performances of the control system; 3) no easy guidelines for choosing those parameters are available.

In this work, we propose an innovative and systematic approach for determining the mentioned design parameters: they should be those which optimize an appropriate control objective. In this case, we search for the design parameters that minimize the time needed to converge to the desired attitude. We measure this time by considering the Integral Time Absolute Error (ITAE) of the attitude, to avoid possible discontinuity issues given by other measures.

However, reaching this aim requires to overcome a number of difficulties, as explained below. The relation between the above ITAE and the design parameters cannot be expressed in analytical form, but only sampled by using software simulations. For this reason, we propose a solution approach based on the use of derivative-free optimization algorithms. These algorithms use no first order information on the objective function. In practice, they work without the analytical expression of the objective function; they only need to compute it in a number of points by using the above simulations.

Moreover, design parameters may range over very wide real intervals; hence, the search space of the optimization problem is numerically wide. Consequently, derivative-free optimization algorithms which rely on the dense exploration of the search space would require excessive run times. Thus, we develop a combination of global search strategy of the type of DIRECT [4,5] and of local search strategy of the type of SDBOX [6,7]. In particular, the local strategy is innovatively used in the first part of the procedure as a fast probing technique to identify the 'promising' region(s) of the search space, which are then explored densely. The advantages of combining and correctly balancing local and global information has been studied for global optimization in several previous works; for instance using: local tuning on the behavior of the objective [8]; the explicit definition of two phases, global and local [9]; acceleration techniques based on local improvements [10,11]. The DIRECT method has been widely used in applications, either in its original form or modified to take into account specific features of the application, see, e.g., [12,13].

Furthermore, the convergence time depends also on the initial conditions of the spacecraft. This implies that control parameters that are efficient for specific initial conditions are not able to guarantee efficiency for different initial conditions. Clearly, there exist a large variety of possible initial conditions for a spacecraft, and they are not known *a priori*. Thus, we initially solve the problem for a particularly meaningful tuple of initial conditions. Subsequently, we define a set of all reasonably possible initial conditions, and we search for the values of the parameters that minimize the ITAE obtained under the worst initial conditions within this set. Such worst initial conditions are not defined in general, but they in turn depend on the adopted design parameters. Therefore, we formulate a min-max problem, whose solutions are robust optimal values for the design parameters, in the sense that they provide the best possible upper bound on the value of ITAE for variations of the initial conditions. This problem is quite difficult, since the solution of the main minimization problem (upper-level) needs the solution of a maximization problem (lower-level) at every evaluation of its objective function, and a decomposition is not possible.

The problem of the optimal determination of the design parameters, in the case of a spacecraft equipped also with angular rate sensors, has been studied in [14]. However, in the case of [14], the control law is substantially simpler than the control law analyzed here, and it contains only two design parameters instead of the four parameters considered in the present work. Due to the complexity of the control law used in this work, the time needed to compute the objective function through simulation can become up to one hundred times longer than the time required in [14]. In addition, since four parameters instead of two must be optimized, the search space here is larger by several orders of magnitude. As a consequence, the optimization problem studied here is much more challenging than the problem solved in [14]. Indeed, the solution approach adopted in [14] is not able to solve the optimization problem considered here, as showed by our experiments. Thus, we develop in this work a new and more complex solution approach, which provides encouraging results. Finally, we also integrate this approach with the innovative computation of a lower bound to the optimization problem based on physical considerations, in order to better evaluate the quality of the obtained solution.

To sum up, the main contributions of this work are: *i*) the definition of a new systematic approach for the determination of the design parameters for a magnetic attitude control algorithm not using attitude rate; *ii*) the development of a combination of derivative free global and local search strategies to tackle these very computationally demanding problems. Therefore, this work constitutes a substantial advancement with respect to [14]. A preliminary version of this paper appeared in [15].

The exposition is organized as follows: Section 2 defines the spacecraft model and the control algorithm and contains the formulations of the problem of determining optimal design parameters; Sections 3 describes the proposed solution approach which integrates global and local strategies; Section 4 reports computational results of the determination of the optimal parameters of the magnetic attitude feedback presented in [2].

2 Control Algorithm and Optimal Design Parameters

To describe the attitude dynamics of an Earth-orbiting rigid spacecraft and to represent the geomagnetic field, we use the following reference frames

Earth-centered inertial frame $\mathcal{F}_i$. It is a standard inertial frame for Earth orbits. Its origin is in the centre of the Earth, its x_i axis is the vernal equinox direction, its z_i axis coincides with the axis of rotation of the Earth and points northward, and its y_i axis completes an orthogonal right-handed frame (see [16, Chapter 2.6.1]).

Spacecraft body frame $\mathcal{F}_b$. The origin of this frame is in the centre of mass of the spacecraft. Its axes are attached to the spacecraft and are picked so that the (inertial) pointing objective is aligning $\mathcal{F}_b$ with $\mathcal{F}_i$.

The goal is aligning $\mathcal{F}_b$ to $\mathcal{F}_i$; thus, the focus will be on the relative kinematics and dynamics of the satellite with respect to the inertial frame. Let $q = [q_1\ q_2\ q_3\ q_4]^T = [q_v^T\ q_4]^T$ be the unit quaternion representing attitude of $\mathcal{F}_b$ with respect to $\mathcal{F}_i$. The corresponding attitude matrix is given by

$$C(q) = (q_4^2 - q_v^{\mathrm{T}} q_v)I + 2q_v q_v^{\mathrm{T}} - 2q_4 q_v^{\times}, \tag{1}$$

where I is the identity matrix (see [17, Section 5.4]). Moreover, a superscript $\times$ applied to any $a \in \mathbb{R}^3$ denotes the skew symmetric matrix

$$a^{\times} := \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{2}$$

that allows to express the cross product $a \times b$ as as the matrix multiplication $a^{\times} b$. The attitude kinematics is given by $\dot{q} = W(q)\omega$ (see [17, Section 5.5.3]), where $\omega \in \mathbb{R}^3$ is the angular velocity of $\mathcal{F}_b$ w.r.t. $\mathcal{F}_i$ resolved in $\mathcal{F}_b$ and

$$W(q) := \frac{1}{2} \begin{bmatrix} q_4 I + q_v^{\times} \\ -q_v^T \end{bmatrix}. \tag{3}$$

The attitude dynamics in body frame is given by $J\dot{\omega} = -\omega^{\times} J\omega + T$, where $J \in \mathbb{R}^{3\times 3}$ is the spacecraft inertia matrix, and T is the control torque expressed in $\mathcal{F}_b$ (see [17]).The spacecraft is equipped with three magnetic coils aligned with the $\mathcal{F}_b$ axes, which generate the magnetic attitude control torque

$$T = m_{coils} \times B^b = -B^{b\times}\ m_{coils} \tag{4}$$

In this expression, $m_{coils} \in \mathbb{R}^3$ is the vector of magnetic dipole moments for the three coils, and B^b is the geomagnetic field at spacecraft expressed in body frame $\mathcal{F}_b$ (see [16, Section 7.4.1]). Let B^i be the geomagnetic field at spacecraft resolved in inertial frame $\mathcal{F}_i$. Note that B^i varies with time, at least because of the spacecraft motion along the orbit. Then,

$$B^b(q,t) = C(q)B^i(t) \tag{5}$$

which shows explicitly the dependence of B^b on both q and t. By grouping together the previous equations, the following system is obtained

$$\begin{aligned} \dot{q} &= W(q)\omega \\ J\dot{\omega} &= -\omega^{\times} J\omega - B^b(q,t)^{\times}\ m_{coils} \end{aligned} \tag{6}$$

in which m_{coils} is the control input. We need to characterize the time-dependence of $B^b(q,t)$, which is the same as characterizing the time-dependence of $B^i(t)$. Assume that the orbit is circular of radius R; then, adopting the so called dipole model of the geomagnetic field (see [18, Appendix H]), we obtain:

$$B^i(t) = \frac{\mu_m}{R^3}[3((\hat{m}^i(t))^T \hat{r}^i(t))\hat{r}^i(t) - \hat{m}^i(t)]. \tag{7}$$

In equation (7), μ_m is the total dipole strength, $r^i(t)$ is the spacecraft position vector resolved in $\mathcal{F}_i$, and $\hat{r}^i(t)$ is the vector of the direction cosines of $r^i(t)$. The components of vector $\hat{m}^i(t)$ are the direction cosines of the Earth's magnetic dipole resolved in $\mathcal{F}_i$ which can be expressed as follows

$$\hat{m}^i(t) = \begin{bmatrix} \sin(\theta_m)\cos(\omega_e t + \alpha_0) \\ \sin(\theta_m)\sin(\omega_e t + \alpha_0) \\ \cos(\theta_m) \end{bmatrix} \tag{8}$$

where θ_m is the dipole's coelevation, $\omega_e = 360.99$ deg/day is the Earth's average rotation rate, and α_0 is the right ascension of the dipole at time $t = 0$. We use $\mu_m = 7.746\ 10^{15}$ Wb m and $\theta_m = 170.0°$ as reported in [19].

Equation (7) shows that, to characterize the time dependence of $B^i(t)$, one needs to determine an expression for $r^i(t)$ which is the spacecraft position vector resolved in $\mathcal{F}_i$. Define a coordinate system a_p, b_p in the orbital plane whose origin is at the center of the Earth, and with a_p axis coinciding with the line of nodes. Then, the position of the center of mass of the satellite is given by

$$\begin{aligned} a^p(t) &= R\cos(nt + \psi) \\ b^p(t) &= R\sin(nt + \psi) \end{aligned} \tag{9}$$

where n is the orbital rate, and ψ is the argument of the spacecraft at time $t = 0$. The coordinates of the center of mass of the satellite in inertial frame $\mathcal{F}_i$ can be easily obtained from (9) by using an appropriate rotation matrix which depends on the orbital inclination *incl* and on the value Ω of the Right Ascension of the Ascending Node (RAAN) (see [16, Section 2.6.2]). Plugging into (7) the equations of the latter coordinates, an explicit expression for $B^i(t)$ can be obtained.

Since $C(q) = I$ for $q = [q_v^T \ q_4]^T = \pm\bar{q}$ where $\bar{q} = [0\ 0\ 0\ 1]^T$ (see (1)), then the goal is designing control strategies for m_{coils} so that $q_v \to 0$ and $\omega \to 0$. The following stabilizing control law, obtained as modification of one described in [3], is proposed in [2]

$$\begin{aligned} \dot{\delta} &= \alpha(q - \epsilon\lambda\delta) \\ m_{coils} &= -m^\star_{coils}\mathrm{sat}\left(\frac{1}{m^\star_{coils}}B^{b\times}\epsilon^2\left(k_1 q_v + k_2\alpha\lambda W(q)^T(q-\epsilon\lambda\delta)\right)\right). \end{aligned} \tag{10}$$

In (10), $\delta \in \mathbb{R}^4$ is an internal state of the controller, $k_1\ k_2\ \alpha\ \lambda\ \epsilon$ are all design parameters, $m^\star_{coils}$ is the saturation limit on each magnetic dipole moment, "sat" denotes the standard saturation function, $B^{b\times}$ is defined through (2), and $W(q)$ was introduced in (3). Note that the previous equation describes an attitude feedback, since it requires only the measure of attitude q and not of attitude rate ω. Thus, it is much more complex than the PD-like feedback considered in [14] which uses measures of ω and contains only two design parameters. As shown in [2], selecting $k_1 > 0$, $k_2 > 0$, $\alpha > 0$, $\lambda > 0$, and choosing $\epsilon > 0$ small enough, local exponentially stability of equilibrium $(q, \omega, \delta) = (\bar{q}, 0, \frac{1}{\epsilon\lambda}\bar{q})$ is achieved for the closed-loop system (6) and (10) if the orbit's inclination *incl* is not too low.

However, there are no indications for choosing the feedback parameters k_1, k_2, α, λ, and the scaling factor ϵ. We only know that k_1, k_2, α, λ have to be greater than zero, and ϵ has to be greater than zero and smaller than an upper bound $\epsilon^* > 0$ which is very difficult to compute. In practice, they are mostly determined by a trial-and-error search, which suffers from the following limitations. First, this approach is quite time-consuming. Second, and more important, it is not systematic. This means that, when satisfactory values of the gains are finally obtained, it is not known whether extending the search could lead to new values of the gains that provide an overall better performance of the closed-loop system. Moreover, in case the search is extended, it is not known when it is appropriate to stop it, and not even the amount of the possible improvements that this additional work could produce. In any case, unless performing an exhaustive search for all the possible values of the gains, it can easily happen that we neglect values providing an overall better performance. On the other hand, an exhaustive search is almost always impossible to perform, because the search space is too large to be explored in practice. Consequently, we propose in this work the following approach to determine the feedback parameters. Since the desired attitude is reached when $q_v = 0$, we define the settling time t_{si} for each component q_i, with $i \in \{1, 2, 3\}$, as:

$$t_{si} := \min\ t \quad \text{ s.t. } |q_i(t)| \leq \nu \quad \forall t \geq t_{si} \tag{11}$$

which represents the time needed for $|q_i|$ to become eventually smaller or equal than ν. Value $0 < \nu < 1$ depends on how small we wish to keep q_i. Then, we define the settling-time t_s for the whole quaternion q as that corresponding to the slowest component of q_v, hence

$$t_s := \max_{i=1,2,3} t_{si} \tag{12}$$

Furthermore, rather then expressing feedback (10) by using five design parameters k_1, k_2, α, λ and ϵ, we rewrite it in terms of only four design parameters

$\kappa_1 = \epsilon^2 k_1 > 0$, $\kappa_2 = \epsilon k_2 > 0$, $\alpha > 0$, $\beta = \epsilon\lambda > 0$, obtaining

$$\begin{aligned} \dot{\delta} &= \alpha(q - \beta\delta) \\ m_{coils} &= -m^*_{coils}\mathrm{sat}\left(\frac{1}{m^*_{coils}} B^{b\times}\left(\kappa_1 q_v + \kappa_2 \alpha\beta W(q)^T (q - \beta\delta)\right)\right). \end{aligned} \tag{13}$$

Now, having set the spacecraft initial conditions to specific values, one can determine the values of κ_1, κ_2, α, and β that minimize the settling time t_s.

However, the objective function t_s is not continuous with respect to the design parameters, and this introduces numerical difficulties in solving the optimization problem. Thus, as a novelty with respect to [14], here we consider, as alternative objective function, the so called Integral Time Absolute Error (ITAE) [20], denoted by Γ:

$$\Gamma = \int_0^{T_f} t\|q_v(t)\|dt \tag{14}$$

where $\|\cdot\|$ denote the Eucledian norm, and T_f is a time chosen sufficiently large. The ITAE has the advantage of being continuous with respect to the design parameters. Continuous differentiability cannot be theoretically ensured. However, since $(q, \omega, \delta) = (\pm\bar{q}, 0, \frac{1}{\beta}\bar{q})$ are equilibria of the closed-loop system (6) and (13), the occurring of $q_v = [0\ 0\ 0]^T$ at some finite time is very unlikely. Note that the latter event would make Γ not continuously differentiable with respect to the design parameters since $\|q_v\|$ is not continuously differentiable at $q_v = [0\ 0\ 0]^T$.

It is known that minimizing the ITAE leads to nearly-optimal solutions w.r.t. the settling time objective. Indeed, in very simple cases, it is proved analytically that minimizing the ITAE leads to solutions that are optimal in terms of settling time minimization. In more complex situations, it was empirically shown that minimizing the ITAE gives solutions that are very close to the optimal ones in terms of settling time minimization (see [20]).

By defining physically reasonable upper bounds $\widehat{\kappa_1}, \widehat{\kappa_2}, \widehat{\alpha}, \widehat{\beta}$ for the design parameters, we obtain the feasible set $K = \{(\kappa_1, \kappa_2, \alpha, \beta) : \ 0 \leq \kappa_1 \leq \widehat{\kappa_1}, 0 \leq \kappa_2 \leq \widehat{\kappa_2},\ 0 \leq \alpha \leq \widehat{\alpha},\ 0 \leq \beta \leq \widehat{\beta}\}$. Now, our optimization problem is:

$$\min_{(\kappa_1, \kappa_2, \alpha, \beta) \in K} \quad \mathit{\Gamma}. \tag{15}$$

Given specific initial conditions of the spacecraft, problem (15) can be solved by a suitable use of derivative-free techniques, as described in the following section. However, when changing the initial conditions of the spacecraft, that solution may be no longer optimal. Since many different initial conditions for the spacecraft may occur in practice, a robust approach would be to search for the optimal solution to problem (15) under the *worst* initial conditions for the spacecraft. Such a worst case optimization is widely used in similar scenarios, because in this manner we can provide an efficient bound on the objective value notwithstanding the uncertainty regarding the initial conditions of the spacecraft. However, the worst initial conditions for the spacecraft is not *a priori* computable, since it depends on the chosen values of $\kappa_1, \kappa_2, \alpha, \beta$, so the problem cannot be decomposed and should be solved as a whole. The initial conditions of the spacecraft are given by $q_0 = q(0)$, $\omega_0 = \omega(0)$, $0 \leq \psi < 2\pi$, and $0 \leq \alpha_0 < 2\pi$. We chose the set of their possible values as:

$$\begin{aligned} S = \Big\{ &(q_0, \omega_0, \psi, \alpha_0) : \ \|q_{0v}\| \leq 1,\ q_{04} = (1 - q_{0v}^T q_{0v})^{1/2}, \\ &|\omega_{01}| \leq \widehat{\omega_{01}},\ |\omega_{02}| \leq \widehat{\omega_{02}},\ |\omega_{03}| \leq \widehat{\omega_{03}},\ 0 \leq \psi < 2\pi,\ 0 \leq \alpha_0 < 2\pi \Big\}. \end{aligned}$$

Note that S includes any possible initial attitude, any possible initial argument ψ for the spacecraft, and any possible right ascension of the Earth's magnetic dipole at time $t = 0$. It only limits the magnitude of the initial angular rate. Now, minimizing Γ under the worst initial conditions for the spacecraft corresponds to the following min-max problem:

$$\min_{(\kappa_1, \kappa_2, \alpha, \beta) \in K} \quad \max_{(q_0, \omega_0, \psi, \alpha_0) \in S} \quad \Gamma. \tag{16}$$

To apply the techniques described in the next section, we convert the feasible set of each optimization problem into a hyperrectangle. Since S has not that shape, we express the set $\|q_{0v}\| \leq 1$ in spherical coordinates (ρ, ϕ, θ):

$$\begin{aligned} S = \{(q_0, \omega_0, \psi, \alpha_0) : q_{01} = \rho \sin\theta \cos\phi,\ q_{02} = \rho \sin\theta \sin\phi,\ q_{03} = \rho\cos\theta, \\ q_{40} = (1 - q_{0v}^T q_{0v})^{1/2},\ 0 \leq \rho \leq 1,\ 0 \leq \phi < 2\pi,\ 0 \leq \theta \leq \pi, \\ |\omega_{01}| \leq \widehat{\omega_{01}},\ |\omega_{02}| \leq \widehat{\omega_{02}},\ |\omega_{03}| \leq \widehat{\omega_{03}},\ 0 \leq \psi < 2\pi,\ 0 \leq \alpha_0 < 2\pi\}\,. \end{aligned}$$

The dependence of Γ on q_0 can now be expressed as dependence on the variables (ρ, ϕ, θ). Consequently, after having introduced the hyperrectangle

$$\begin{aligned} H = \{(\rho, \phi, \theta, \omega_0, \psi, \alpha_0) :\ 0 \leq \rho \leq 1,\ 0 \leq \phi < 2\pi,\ 0 \leq \theta \leq \pi, \\ |\omega_{01}| \leq \widehat{\omega_{01}},\ |\omega_{02}| \leq \widehat{\omega_{02}},\ |\omega_{03}| \leq \widehat{\omega_{03}},\ 0 \leq \psi < 2\pi,\ 0 \leq \alpha_0 < 2\pi\}\,, \end{aligned}$$

the min-max problem (16) can be equivalently reformulated as follows

$$\min_{(\kappa_1, \kappa_2, \alpha, \beta) \in K} \quad \max_{(\rho, \phi, \theta, \omega_0, \psi, \alpha_0) \in H} \quad \Gamma. \tag{17}$$

3 Proposed Solution Approach

To simplify and generalize the description of the proposed approach, that could also be applied to different problems sharing the same structure, we now rename the set of design parameters $(\kappa_1, \kappa_2, \alpha, \beta)$ as x belonging to a feasible set $F_x = \{lbx_i \leq x_i \leq ubx_i,\ i = 1, \ldots, n\} \subset \mathbb{R}^n$ (in our case $F_x = K$ and $n = 4$), and the set of initial conditions $(\rho, \phi, \theta, \omega_0, \psi, \alpha_0)$ as y belonging to a feasible set $F_y = \{lby_j \leq y_j \leq uby_j,\ j = 1, \ldots, m\} \subset \mathbb{R}^m$ (in our case $F_y = H$ and $m = 8$, since ω_0 has 3 components). Denote by $f(x, y)$ the function providing the objective value (in our case $\varGamma$). When y is fixed (initial conditions assigned) we simply write $\bar{y}$ in it, when x is fixed (design parameters assigned) we write $\bar{x}$ in it. Problem (15) is expressed as

$$\min_{x \in F_x} f(x, \bar{y})$$

and may be tackled by a global derivative-free optimization algorithm of the type of DIRECT [4]. Those methods work without the need for analytically writing the objective function; they only need to compute it in a number of points by using simulations. In more detail, Direct-type algorithms work as follows. The feasible region starts as a single hyperrectangle that is normalized to a unit hyperrectangle. At the generic iteration k, the algorithm partitions the hyperrectangles obtained from the previous $(k-1)^{th}$ iteration to form a collection of smaller hyperrectangles $\mathcal{H}^{(k)} = \{H_1, \ldots, H_{p^{(k)}}\}$, and evaluates each of them. Potentially optimal hyperrectangles within $\mathcal{H}^{(k)}$ are identified, and only these hyperrectangles are further partitioned and investigated in the next $(k+1)^{th}$ iteration of the algorithm.

Usually, each hyperrectangle H_h is evaluated by sampling the objective function at its central point. This choice, combined with the subsequent tripartition of H_h in case H_h is among the potentially optimal ones, aims at reducing the number of function evaluations. The algorithm stops when the size of the hyperrectangles becomes too small, or when it reaches the maximum number of iterations. Due to the so-called *everywhere dense* property, such an algorithm converges to a global optimum of the function if the sampling is dense enough. However, a dense search may require a very large number of function evaluations.

In our case, all $lbx_i = 0$, since design parameters cannot be negative. Values ubx_i can be set after considerations on the physical problem, typically at very high values. For instance, $\widehat{\kappa_1}$ may easily reach 10^9, since its maximum feasible value can be determined knowing the saturation level of the dipole moment of the coils, the minimum amplitude of the geomagnetic field, and the attitude sensor's resolution. However, by using these bounds, a sufficiently dense exploration of the feasible set F_x requires a number of function evaluations such that the corresponding run time is impracticable. Conversely, a sampling that uses a practically affordable number of function evaluations does not reach solutions sensibly better than random solutions. Indeed, our problem has a large number of local minima, which strongly affect the difficulty of the overall optimization task. In this case, the evaluation of the generic H_h using only one point may be very inaccurate at the first iterations of the algorithm, because the initial size of those hyperrectangles is too large. By proceeding with the iterations, their size becomes smaller, but their number, and consequently the run time, increases exceedingly.

Therefore, we would need a fast but more effective probing technique to early identify the 'promising' H_h. Then, the problem could be tackled if the span of such promising regions is small enough to be explored densely in reasonable time. Hence, we develop a probing technique based on the use of the local derivative-free optimization algorithm SDBOX. This algorithm was initially proposed in [6] as a globally convergent algorithm for the minimization of a continuously differentiable function, but it can be practically used to optimize different types of functions as a good compromise between efficiency and convergence properties (see also [21]). It is a derivative-free algorithm inspired by the strategy underlying gradient-based methods: finding a *good* feasible descent direction for the objective function, and performing a *sufficiently* large step along such direction. It needs no information on the first order derivatives, because a good feasible descent direction is determined by investigating the local behavior of the objective along different directions. Most interestingly for our case, given a starting point $(\kappa_1', \kappa_2', \alpha', \beta') = x'$, this algorithm is aimed at quickly finding good solutions in the vicinity of x'.

A solution approach combining the above described global and local strategies to solve problem (15) is described below.

Procedure 1: Solve min combining Global and Local search

Input A vector $\bar{y} \in F_y$ and a function $f(x, \bar{y})$ computable by means of a software simulation for any $x \in F_x$. Values for the parameters p, *maxeval*, *maxsubsets*, *maxiter*, *maxpost*.

Output A solution x^{**} approximating one vector in $\arg\min_{x \in F_x} f(x, \bar{y})$.

1. Normalize and grid partition the whole feasible set F_x into a collection of hyperrectangles $\mathcal{H}^{(1)} = \{H_1, \ldots, H_p\}$ similarly to the initial phase of DIRECT.
2. For each $H_h \in \mathcal{H}^{(1)}$, compute the value f_h of the solution obtained by *maxeval* iterations of SDBOX in H_h starting from its central point. This is an upper bound on the value of the best solution in H_h and constitutes our 'evaluation' of H_h.
3. Take a number *maxsubsets* of hyperrectangles corresponding to the smallest of the above f_h values.
4. Take the region given by the union of those subsets, and 'convexify' it by including also the additional subsets required to convert it into an hyperrectangle ${F_x}^*$.
5. Switch to DIRECT algorithm to continue the search in ${F_x}^*$ allowing *maxiter* function evaluations. This search can now be dense using reasonable time and it gives a solution x^*.
6. Try to improve x^* by using *maxpost* iterations of a local search method, finally obtaining a solution x^{**} to problem (15).

The number of hyperrectangles p in the partitioning phase (step 1) and the number of iterations *maxeval* in the evaluation phase (step 2) should be calibrated on the considered practical case, so that the evaluation phase is executed for a given amount of time. Indeed, there is a trade off between speed and effectiveness of this phase. Also, such parameters should be balanced by considering that the accuracy of the evaluation of each H_h depends not only on *maxeval*, but also on the span of each H_h, which in turn depends on p . Note, however, that the hyperrectangles in the initial partition $\mathcal{H}^{(1)}$ do not need to have the same size; their size can be increasing with the absolute values of the coordinates so that p is kept smaller. The number *maxsubsets* in the selection phase (step 3) should be chosen so that ${F_x}^*$ is smaller than F_x by several orders of magnitude, otherwise the advantages of the proposed procedure lessen. The number of function evaluations *maxiter* in the standard DIRECT phase (step 5) is selected so that the search in ${F_x}^*$ is dense enough; this search can now be accomplished in practice because of the size reduction in F_x. The post-optimization phase (step 6) can be performed with SDBOX or with CS-DFN [7], another linesearch based method which uses a dense set of search directions, instead of only the coordinates ones. For this reason, CS-DFN does not require the hypothesis of f continuously differentiable, but may be more computationally expensive on smooth problems. In the case of our application, for instance, Γ is at least continuous. There are no theoretical arguments to ensure its continuous differentiability, even if this may often occur in practice. In conclusion, by choosing between SDBOX or

CS-DFN depending on the presence or absence of the continuous differentiability of f, and by setting *maxpost* sufficiently large, we find a solution x^{**} which satisfies necessary conditions for a local optimum. Moreover, x^{**} should approximate one of the global optima, because, given the dense search in F^*, it should provide one of the global minima of F^*, and the regions of $F \setminus F^*$, which were less ‘promising’, should not contain better solutions. Clearly, depending on the computational demand of the practical case, the accuracy and the properties of the evaluation phase can be modified, either by changing the number of iterations, or even by using a different evaluation algorithm, still remaining within the same algorithmic framework.

On the other hand, problem (17) is composed of an upper-level minimization problem and a lower-level maximization one. By renaming the set of variables as described at the beginning of the section, the problem has the following form,

$$\min_{x \in F_x} \left(\max_{y \in F_y} f(x,y) \right) = \min_{x \in F_x} g(x)$$

with function g such that its value on the generic point $\bar{x}$ is given by the solution of the lower-level problem:

$$g(\bar{x}) = \max_{y \in F_y} f(\bar{x}, y).$$

We solve the upper-level problem by means of an external loop applying the combination of local and global search described as Procedure 1. This loop computes, using a parameter *maxeval_ext* to define the overall maximum number of function evaluations, the value of g corresponding to different points of F_x. Let $\bar{x}$ be one of them; then the evaluation of $g(\bar{x})$ needs the solution of one lower-level maximization problem $\max_{y \in F_y} f(\bar{x}, y)$. Thus, the lower-level problem must be solved up to *maxeval_ext* times. Consequently, solving it in a few seconds is crucial.

For this problem, the global strategy either would perform a very poor search or would require excessive time. Therefore, the local strategy appears the only feasible choice for the lower-level problem. However, taking as starting point the center of the feasible set F_y does not lead to good solutions of the maximization problem within the limited available time.

In this case, by using our knowledge of the physics of the problem, we are able to suppose that good solutions of the maximization problem are in the vicinity of extremal values of angular velocity. Hence, we solve the lower-level problem in a nested loop by using a local search with multi-start, using as starting points the eight combinations of extreme values for the three components of the angular velocity $\pm\widehat{\omega_{01}}$, $\pm\widehat{\omega_{02}}$, and $\pm\widehat{\omega_{03}}$. This can be performed by means of SDBOX or CS-DFN, depending on the presence or absence of the continuous differentiability of f. In the case of our application, for instance, Γ is at least continuous also w.r.t. the initial conditions of the spacecraft. Again, there are no theoretical arguments to ensure its continuous differentiability, even if this may often occur in practice. For each solution of the lower-level problem, we allow a necessarily small maximum number of function evaluations *maxeval_int*. We obtain in this way a solution x^*_R to problem (17). The whole procedure is as follows:

Procedure 2: Solve min-max combining Global and Local search

Input A $f(x, y)$ computable by means of a software simulation for any $x \in F_x$ and $y \in F_y$. Values for the parameters *maxeval_ext*, *maxeval_int*, s.

Output A robust solution x^*_R approximating one vector in $\arg\min_{x \in F_x}(\max_{y \in F_y} f(x, y))$.

External loop:

Solve the upper-level problem $\min_{x \in F_x} g(x)$, with $g(\bar{x}) = \max_{y \in F_y} f(\bar{x}, y) \ \ \forall \bar{x} \in F_x$ by using Procedure1 with *maxeval_ext* total evaluations of g and return x^*_R.

Given $\bar{x}$, the evaluation of $g(\bar{x})$ is performed by the internal loop.

Internal loop:

Take $\bar{x}$ and solve the lower-level problem $\max_{y \in F_y} f(\bar{x}, y)$ by using multistart local search with s starting points, performing $\frac{maxeval\_int}{s}$ evaluations of f for each of them.

4 Computational Results

We apply our approach to solve the case study presented in [2]. The spacecraft inertia matrix is $J = \mathrm{diag}[27,\ 17,\ 25]$ kg m^2, and the saturation level for each magnetic dipole moment is $m^*_{coils} = 10$A m^2. The inclination of the orbit is $incl = 87°$, and the orbit's altitude is 450 km; the value Ω of Right Ascension of the Ascending Node is 0. Upper bounds $\widehat{\kappa_1}, \widehat{\kappa_2}, \widehat{\alpha}, \widehat{\beta}$ are set at $(10^9, 10^9, 10^4, 10^{-3})$.

We initially consider the easier case of known initial conditions for the spacecraft, and we present, in Section 4.1, the results of our Procedure 1 in solving this problem. We also provide there a comparison with the classical DIRECT method. Subsequently, in Section 4.2, we consider the more realistic case of a spacecraft having variable initial conditions. Since the standard DIRECT method could not solve the former case, which is much easier, even allotting quite large run times, we consider useless to apply it to this more demanding case, and we solve the problem only with Procedure 2.

4.1 Spacecraft operating from Fixed Initial Conditions

We consider here the case of the above described spacecraft with known fixed initial conditions, thus we deal with problem (15) using the following values:

$$(\rho, \phi, \theta, \omega_0, \psi, \alpha_0) = (0, 0, 0, 0.02, 0.02, -0.03, 0.9416, 4.5392) \qquad (18)$$

The best solution obtained by trial and error search in [2] has a value of ITAE=3.7×10^7, while the vast majority of the solutions reach a limit value for ITAE of about 1.2×10^9. This upper limit is due to the value of T_f in the definition of ITAE (14), which is chosen equal to 56,009 secs. that corresponds to 10 orbital periods. This means in practice that, when we reach the limit value of ITAE, the corresponding settling time would be greater than roughly 10 orbital periods. Then, that solution is not a good one and we are not interested in determining it with further precision. Even if this choice causes a flattening in the values of ITAE, the use of such a finite T_f is necessary to practically run the simulations that compute the ITAE.

As reported in Table 1, we preform two solution attempts by using the standard DIRECT algorithm on the feasible set K with respectively 50,000 and 100,000 iterations, followed by 1,000 iterations of local search refinement using CS-DFN. Notwithstanding the substantial computational effort (the running times of these experiments respectively correspond to about 3 days and 1 week), the solutions obtained have values of ITAE greater than 1.1×10^9, that is not much different from a random solution. Indeed, even with such a large number of iterations, the search was not dense enough to satisfactory explore the feasible set. Other similar attempts with standard DIRECT do not reach better results. On the other hand, by using Procedure 1 of Section 3, we obtain a much better solution, with a value of ITAE of about 8×10^6, which is also considerably better than the best solution obtained by trial-and-error in days of work. The evolution of Procedure 1 is described below in detail.

To identify the promising region of the feasible set K, we use the described probing technique. The values of $maxelav$ and p should be such that this step can be performed in reasonable time. One single function evaluation takes a time which is extremely variable, and varies from fractions of seconds to several tenths of seconds; a very rough average function evaluation time can be assessed as equal to 1 sec. We select $maxeval = 10$, and consequently we estimate that the evaluation of each hyperrectangle requires roughly 10 sec. Thus, to finish within 2 or 3 days of computation, we select $p = 40,960$,

Table 1 Comparison of Procedure 1 and standard DIRECT with 50,000 or 100,000 iterations.

Algorithm	Solution	Obj Value	Time
DIRECT 50,000 + CS-DFN 1,000	κ_1= 913405022.139 κ_2=195426826.870 α= 9794.170752422 β= 0.000000000000	1,142,470,478.101	262,000 + 5,460 sec.
DIRECT 100,000 + CS-DFN 1,000	κ_1= 500798437.500 κ_2= 159788790.177 α= 9996.679486501 β= 0.000000000003	1,129,234,873.703	565,200 + 5,040 sec.
Procedure 1: combining global and local strategies	κ_1=246494.579020 κ_2= 233333315.349 α= 92.5925925927 β= 0.000129629629	**8,021,573.4077**	201,500 + 15 + 4 sec.

which is obtained by making 64 partitions on the domain of κ_1, 64 partitions on the domain of κ_2, and 10 partitions on the domain of α. The intervals corresponding to these partitions are not of the same size; they increase with the absolute value of the coordinates. We obtain the following hyperrectangle as convexification of the collection of the most promising regions, as in Step 4 of Procedure 1:

$$K^* = \{(\kappa_1, \kappa_2, \alpha, \beta) : 200000 \leq \kappa_1 \leq 260000,$$
$$200000000 \leq \kappa_2 \leq 310000000, 0 \leq \alpha \leq 100, 0 \leq \beta \leq 0.001\}.$$

The determination of K^* actually requires 409,600 iterations of SDBOX and 201,500 sec. of computations (about 56 hours). Note that the time required by each function evaluation is generally much faster in K^*, where it can be less than 0.01 sec., than in the rest of K. A complete analysis of the simulation run time is however out of the scope of this work. Now, by applying standard DIRECT strategy over the feasible set K^*, as in Step 5 of Procedure 1, after 3,007 iterations and only 15 sec., we obtain the solution:

$$\kappa_1 = 200781.893004,\ \kappa_2 = 226268861.454,\ \alpha = 39.3004115226,\ \beta = 0.0005 \tag{19}$$

whose value is ITAE = 9.072×10^6. Note that the volume of the set K^* is considerably smaller than that of K: it is only 1/15151515.15 of the volume of K. Consider that, as an example, an exploration of K with the same degree of density used on K^* would require $15 \times 15,151,515.15 = 227,272,727.25$ sec., that roughly corresponds to more than 7 years, if the simulation times on K were the same as on K^*. Since they are often much slower, the time needed would be even more.

Solution (19) can be further improved by using the local search strategy, as in Step 6 of Procedure 1. By performing 1000 iterations of the local search CS-DFN, which can move along a dense set of directions [7], the solution (19) is improved in 4 sec. to ITAE = 8.021×10^6 with the solution:

$$\kappa_1 = 246494.579020,\ \kappa_2 = 233333315.349,\ \alpha = 92.5925925927,\\ \beta = 0.000129629629. \tag{20}$$

As a comparison, we also tested the local search SDBOX, which moves only along the coordinate directions [6], for this post-optimization phase. When performing 5000 additional iterations of SDBOX, solution (19) is improved in 11 sec. with the solution reported below, which has ITAE = 9.062×10^6

$$\kappa_1 = 200781.893004,\ \kappa_2 = 226268812.597,\ \alpha = 39.3004115226,$$
$$\beta = 0.000500000024.$$

Solution (20) corresponds to x^{**} of Procedure 1. Since this is actually an approximation of an optimal solution, and there are no provably optimal solutions available for comparison, we further evaluate its quality as follows.

We compute the settling time (see (11) and (12)) corresponding to (20), that is $t_s = 6,280$ sec. Then, we compute a lower bound $\underline{t_s}$ on the the minimum settling time of the considered case, on the basis of physical considerations. We stress that this bound is extremely conservative, in the sense that the spacecraft evolution surely cannot require less time than that, though it could very easily require more. The initial conditions (18) correspond to having the spacecraft with the desired attitude but with a nonzero initial angular rate $\omega(0) = [0.02\ 0.02\ \ -0.03]^T$. Consider now simple rotations about each single body axis, and for each simple rotation compute lower bounds $\underline{t_{sx}}$, $\underline{t_{sy}}$, $\underline{t_{sz}}$ of the times necessary to move to the desired attitude with final zero angolar rate. Then, a rough lower bound for the settling time is given by $\underline{t_s} = \max\{\underline{t_{sx}}, \underline{t_{sy}}, \underline{t_{sz}}\}$. Value $\underline{t_{sx}}$ can be computed using the equation which describes rotation about the x body axis, which is given by

$$\ddot{\phi} = J_x T_x \tag{21}$$

where ϕ is the roll angle. The amplitude of torque T_x is limited by an upper limit T^*, which can be found using (4). Indeed, numerical simulations show that $\|B^i\| \leq B^* = 5 \cdot 10^{-5}$ T. Since $\|B^b\| = \|B^i\|$ (see (5)), then $\|B^b\| \leq B^*$. Moreover, each component of m_{coils} is bounded by $m^*_{coils} = 10$ A m^2 then $\|m_{coils}\| \leq \sqrt{3}\ m^*_{coils}$. Thus, $T^* = \sqrt{3}\ B^*\ m^*_{coils} = 5\sqrt{3} \cdot 10^{-4}$ N m. Next, the minimum time to bring the state of system (21) subject to the constraint $|T_x| \leq T^*$, from the initial state $\phi = 0\ \dot{\phi} = \omega_x(0)$ to the final state $\phi = 0\ \dot{\phi} = 0$, is given by (see [22, Section 7.2])

$$\underline{t_{sx}} = \frac{J_x}{T^*}(1+\sqrt{2})|\omega_x(0)| = 1,505 \text{ sec.}$$

Similar considerations hold for rotations about y and z axes leading to

$$\underline{t_{sy}} = \frac{J_y}{T^*}(1+\sqrt{2})|\omega_y(0)| = 948 \text{ sec.}$$

$$\underline{t_{sz}} = \frac{J_z}{T^*}(1+\sqrt{2})|\omega_z(0)| = 2,091 \text{ sec.}$$

Then, $\underline{t_s} = \max\{\underline{t_{sx}}, \underline{t_{sy}}, \underline{t_{sz}}\} = 2,091$ sec. Therefore, solution (20) only takes about 70 minutes more than the minimum time necessary to rotate the spacecraft about a single body axis at the maximum speed allowed by the available magnetorquers so that it goes to the desired rest position. Thus, solution (20) does not appear to be too far from an optimal solution.

4.2 **Spacecraft operating from Variable Initial Conditions**

We now solve problem (17) by using the above described Procedure 2. We select 20 partitions on the domain of κ_1, 20 on that of κ_2, and 10 on that of α. We therefore perform 40,000 evaluations of $g(x)$ for the identification of the new K^*, and then another 3000 + 1000 evaluations of $g(x)$ to solve the problem on this K^*, for a total of $maxeval\_ext = 44,000$.

For the internal loop, as explained in Procedure 2, we use as starting points the eight combinations of extreme values for the three components of the angular velocity $\pm\widehat{\omega_{01}}$, $\pm\widehat{\omega_{02}}$, and $\pm\widehat{\omega_{03}}$. We allow 64 iterations per point, for a total of 512 iterations per single lower-level problem, which requires slightly less than 3 sec. in average within K^*. Run times are greater in the rest of K, however we try to keep them under control by allowing to exit the internal loop when the value of $f(\bar{x}, y)$ is large enough to reach the limit value for ITAE of about 1.2×10^9. Then, the whole nested loop procedure provides the following solution in about 342,000 sec. (about 95 hours)

$$\kappa_1 = 227777.777778,\ \kappa_2 = 294444444.444,\ \alpha = 83.3333333333,$$
$$\beta = 0.00061095869532.$$

This solution has value ITAE= 2.176×10^7 for the initial conditions (18) instead of ITAE $= 8.021 \times 10^6$ of solution (20), but it is a robust solution: by varying the initial conditions in H, the worst value that can be obtained is ITAE $= 1.357 \times 10^8$, that is still considerably better than average solutions, whose vast majority has the limit value for ITAE of 1.2×10^9. As a comparison, the worst value obtainable by varying the initial conditions in H for solution (20) is ITAE $= 1.103 \times 10^9$, that is very near to the limit value for ITAE.

Indeed, the above limit value for ITAE is very easily obtainable for almost any generic tuple $(\bar{\kappa}_1, \bar{\kappa}_2, \bar{\alpha}, \bar{\beta})$ by simply searching for difficult initial conditions. Note also that this value of ITAE is obtained in some attempts in solving problem (17) by using standard DIRECT algorithm on the whole feasible set K allowing no more than 1 day of computation. We did not perform more time consuming attempts in solving problem (17) using standard DIRECT algorithm, since, considering the results already obtained in Section 4.1 allocating one week of computation, and that in this case we have an internal loop requiring 512 function evaluations instead of one single function evaluation, a serious attempt would need to allocate months of computation, and would obtain results probably similar to those obtained in Section 4.1.

5 Conclusions

The attitude control of a spacecraft using only magnetorquers can be achieved by an attitude feedback. However, four design parameters must be assigned. They have a deep influence on the spacecraft's behavior, and their determination is a critical open issue. Here, we formulate this problem as the selection of the parameters that minimize the Integral Time Absolute Error (ITAE), either for fixed initial conditions of the spacecraft or under the worst initial conditions. This latter choice gives an upper bound on the minimum value of the ITAE obtainable by varying the initial conditions. This formulation of the problem precisely represents the practical aims; however, it turns out to be a very difficult min-max problem. To practically solve these extremely computationally demanding problems, we have presented here a solution approach based on an innovative integration of global and local derivative-free optimization techniques. The proposed approach is able to provide robust solutions to the considered application in reasonable times.

Acknowledgements The authors would like to thank professor Stefano Lucidi for essential suggestions and discussions.

References

1. Silani, E., Lovera, M.: Magnetic spacecraft attitude control: A survey and some new results. Control Engineering Practice 13(3), 357-371 (2005)
2. Celani, F.: Robust three-axis attitude stabilization for inertial pointing spacecraft using magnetorquers. Acta Astronautica 107, 87-96 (2015)
3. Lovera, M., Astolfi, A.: Spacecraft Attitude Control Using Magnetic Actuators. Automatica, 40(8), 1405-1414 (2004)
4. Jones, D.R., Perttunen, C.D., Stuckman, B.E.: Lipschitzian optimization without the Lipschitz constant. Journal of Optimization Theory and Application 79(1), 157-181 (1993)
5. Jones, D.R.: DIRECT global optimization. In: C.A. Floudas and P.M. Pardalos (eds.), Encyclopedia of Optimization, 725-735, Springer, Berlin (2009)
6. Lucidi, S., Sciandrone, M.: A Derivative-Free Algorithm for Bound Constrained Optimization. Computational Optimization and Applications 21, 119-142 (2002)
7. Fasano, G., Liuzzi, G., Lucidi, S., Rinaldi, F.: A Linesearch-based Derivative-free Approach for Nonsmooth Constrained Optimization. SIAM Journal on Optimization 24(3), 959-992 (2014)
8. Sergeyev, Y.D.: An Information Global Optimization Algorithm with Local Tuning. SIAM Journal on Optimization 5(4), 858–870 (1995)

9. Sergeyev, Y.D., Kvasov, D.E.: Global Search Based on Efficient Diagonal Partitions and a Set of Lipschitz Constants. SIAM Journal on Optimization 16(3), 910–937 (2006)
10. Lera, D., Sergeyev, Y.D.: Acceleration of Univariate Global Optimization Algorithms Working with Lipschitz Functions and Lipschitz First Derivatives. SIAM Journal on Optimization 23(1), 508–529 (2013)
11. Sergeyev, Y.D., Mukhametzhanov, M.S., Kvasov, D.E., Lera, D.: Derivative-Free Local Tuning and Local Improvement Techniques Embedded in the Univariate Global Optimization. Journal of Optimization Theory and Applications 171, 186-208 (2016)
12. Bartholomew-Biggs, M.C., Parkhurst, S.C., and Wilson, S.P.: Using DIRECT to solve an aircraft routing problem. Computational Optim. and Applications 21(3), 311-323 (2002)
13. di Serafino, D., Liuzzi, G., Piccialli, V., Riccio, F., Toraldo, G.: A Modified DIviding RECTangles Algorithm for a Problem in Astrophysics. Journal of Optimization Theory and Applications 151, 175-190 (2011)
14. Bruni, R., Celani, F.: A Robust Optimization Approach for Magnetic Spacecraft Attitude Stabilization. Journal of Optimization Theory and Applications 173, 994-1012 (2017)
15. Bruni, R., Celani, F.: Determining optimal parameters in magnetic spacecraft stabilization via attitude feedback. AIP Proceedings 1776, 090032 (2016)
16. Sidi, M.J.: Spacecraft dynamics and control. Cambridge Univ. Press, New York (1997)
17. Wie, B.: Space vehicle dynamics and control. AIAA, Reston, VA (2008)
18. Wertz, J.R. (ed.): Spacecraft attitude determination and control. Kluwer Academic, Norwell, MA (1978)
19. Rodriguez-Vazquez, A.L., Martin-Prats, M.A., Bernelli-Zazzera, F.: Full magnetic satellite attitude control using ASRE method. In 1st IAA Conference on Dynamics and Control of Space Systems (2012)
20. Graham, D., Lawthrop, R.C.: The synthesis of optimum response: Criteria and standard forms, part 2. Transactions of the AIEE 72, 273-288 (1953)
21. Kolda, T.G., Lewis, R.M., Torczon, V.: Optimization by direct search: new perspectives on some classical and modern methods. SIAM Review 45(3), 385-482 (2003)
22. Athans, M., Falb, P. L..: Optimal control: an introduction to the theory and its applications. Dover, New York (2007)