

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com



Parameter Optimization for Spacecraft Attitude Stabilization Using Magnetorquers

Renato Bruni and Fabio Celani

Abstract

The attitude stabilization of a spacecraft that uses magnetorquers as torque actuators is a very important task. Depending on the availability of angular rate sensors on the spacecraft, control laws can be designed either by using measurements of both attitude and attitude rate or by using measurements of attitude only. The parameters of both types of control laws are usually determined by means of a simple trial-and-error approach. Evidently, such an approach has several drawbacks. This chapter describes recently developed systematic approaches for determining the parameters using derivative-free optimization techniques. These approaches allow to find the parameter values that minimize the settling time of the attitude error or an indirect measure of this error. However, such cost indices depend also on initial conditions of the spacecraft, which are not known in advance. Thus, a min-max optimization problem is formulated, whose solution provides values of the parameters minimizing the chosen cost index under the worst initial conditions. The chapter also provides numerical results showing the effectiveness of the described approaches.

Keywords: derivative-free optimization, spacecraft attitude control, robust optimization, min-max problems, magnetic actuators

1. Introduction

A *magnetorquer* (or magnetic torquer) is a torque actuator widely used for attitude control in satellites, especially those flying in low Earth orbits. The magnetorquer generates a magnetic dipole that interacts with the Earth magnetic field, thus generating a torque used to control the spacecraft attitude. The control torque generated by magnetorquers is constrained to belong to a plane orthogonal to the Earth magnetic field; hence, magnetorquers may be supported by additional torque actuators to achieve full three-axis control (see [1], Chapter 7). However, attitude control systems using only magnetorquers represent a feasible option especially for low-cost satellites or for satellites with a fault in the main attitude control system. Therefore, attitude control of spacecraft using magnetorquers is a very important topic in aerospace engineering.

Many control laws have been designed in such a setting, and an overview can be found in [2]. In particular, Celani [3] shows that attitude stabilization using only magnetorquers can be achieved by proportional derivative-like (PD-like) control

which requires measurements of both attitude and attitude rate. The same work shows that attitude stabilization can also be obtained by using attitude-only feedback, which has the advantage of not requiring the installation of angular rate sensors (rate gyros), thus saving in cost, volume, and weight. Those control laws contain parameters. Thus, numerical values must be assigned to those parameters to practically implement the laws. A common way to find those values is by using a so-called trial-and-error approach, consisting in trying several values and rejecting those which are not acceptable, and eventually choosing the best among the acceptable ones. This approach is affected by two important drawbacks: (a) it is often very time-consuming, and (b) it is not systematic. Thus, even if an obtained solution is satisfactory, it is not known if protracting the search could lead to better solutions and which would be the magnitude of the possible improvement.

On the other hand, a systematic approach for determining the parameters should aim at finding the values which minimize the settling time of the attitude error. Such an approach has been recently proposed in [4–6]. However, this is not an easy objective to pursue. The settling time depends not only on the parameters but also on the initial conditions of the spacecraft. To overcome this issue, the above works propose to compute the values of the parameters that minimize the settling time obtained under the worst initial conditions, so as to provide averagely good results and to set as bound the worst-case behavior. Hence, in the above works, the problem is modeled as a min-max problem, and the obtained parameters' values are called *robust optimal* values.

This min-max problem is considerably challenging, since solving the main minimization problem (upper-level) requires solving a maximization problem (lower-level) at every evaluation of its objective function. A decomposition is not possible because the worst initial conditions are not determined in general: they, in turn, depend on the adopted parameters. Optimizing this min-max problem involves other two mathematical issues:

1. Settling time cannot be expressed as an analytical function of parameters and initial conditions; therefore the specification of an explicit optimization model is not possible.
2. Settling time is not even continuous with respect to both parameters and initial conditions, in the sense that small variations of the latter may result in a substantial gap in the variation of the settling time.

To overcome these issues, the approach proposed in [4–6] relies on the use of derivative-free optimization algorithms as building blocks. These techniques do not need first-order information on the objective function nor do they need its analytical expression. They only need to compute the objective function over a number of points by means of simulations.

For the design of the specific optimization algorithm, a distinction must be done between spacecraft equipped with angular rate sensors, studied in [5], and spacecraft not equipped with those sensors, studied in [6]. This because, in the first case, the control is based on the availability of measurements of both attitude and attitude rates and is realized through a *PD-like control*, while in the second case, the control can rely only on attitude feedback, and it is called *attitude-only feedback control*.

The first case, though presenting the mentioned difficulties, turns out to be the easier among the two, since the PD-like control law has only two parameters, and their robust optimal values can be successfully determined by means of a global search optimization procedure of the type of DIRECT algorithm [7], as proposed in [5]. In the second case, the control law contains four parameters, and their range of

variation is wider than the first case. Thus, the determination of the robust optimal parameters becomes even harder, and a very complex derivative-free optimization algorithm based on a combination of both local and global search had to be developed. Moreover, numerical experience has shown that dealing with a discontinuous objective function makes the determination of the optimal solution very challenging even when using derivative-free algorithms. Thus, in Ref. [4, 6], the objective function is changed to the so-called integral time absolute error (ITAE) which is continuous with respect to both the parameters and the initial conditions. Such a change is acceptable since it has been shown that minimizing the ITAE is approximately equivalent to minimizing the settling time (see [8]).

This chapter describes in detail the abovementioned approaches to the determination of robust optimal values for the parameters in order to minimize the settling time or the ITAE obtained under the worst initial conditions. The exposition will highlight in particular the following main contributions: (i) the definition of a new systematic approach for the determination of the parameters for the PD-like control algorithm and for the attitude-only feedback, (ii) the formulation of a min-max optimization model to find a robust optimal solution to both problems, and (iii) the development of derivative-free optimization strategies to tackle the min-max problems.

The chapter is organized as follows: Section 2 defines the spacecraft model and the control algorithms for the two cases mentioned above; Section 3 describes the optimization model and the solution algorithm for the case of PD-like control; Section 4 provides some computational experience for this first case; Section 5 describes the optimization model and the solution algorithm for the case of attitude-only feedback control; Section 6 provides again computational experience for this second case.

2. Spacecraft model and control algorithms

The following coordinate frames are introduced to describe the attitude dynamics of an Earth-orbiting spacecraft and the Earth magnetic field:

Earth-centered inertial frame \mathcal{F}_i . The origin of the frame is the center of the Earth, the x_i axis coincides with the vernal equinox direction, the z_i axis is the axis of rotation of the Earth and points northward, and the y_i axis completes the frame (see [1], Chapter 2.6.1).

Spacecraft body frame \mathcal{F}_b . Its origin is in the spacecraft mass center. The axes are attached to the spacecraft and are selected so that the (inertial) pointing objective is having \mathcal{F}_b aligned with \mathcal{F}_i .

Since the objective is having \mathcal{F}_b aligned to \mathcal{F}_i , consider the relative kinematics and dynamics of the satellite with respect to the inertial frame. Let the attitude of \mathcal{F}_b with respect to \mathcal{F}_i be represented by quaternion $q = [q_1 \ q_2 \ q_3 \ q_4]^T = [q_v^T \ q_4]^T$. The corresponding direction cosine matrix is equal to

$$C(q) = (q_4^2 - q_v^T q_v)I + 2q_v q_v^T - 2q_4 q_v^\times, \quad (1)$$

where I is the identity matrix (see [9], Section 5.4). Moreover, given $a \in \mathbb{R}^3$ symbol a^\times denotes the skew symmetric matrix

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

so that the cross product $a \times b$ can be expressed as the matrix multiplication $a^\times b$. The attitude kinematics equation is equal to $\dot{q} = W(q)\omega$ (see [9], Section 5.5.3), where $\omega \in \mathbb{R}^3$ is the spacecraft angular velocity resolved in \mathcal{F}_b and

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

The attitude dynamics equation resolved in body frame is equal to $J\dot{\omega} = -\omega^\times J\omega + T$, where $J \in \mathbb{R}^{3 \times 3}$ is the inertia matrix of the spacecraft and T is the control torque resolved in \mathcal{F}_b (see [9]). Three magnetic coils aligned with the \mathcal{F}_b axes equip the spacecraft. Thus, the following magnetic attitude control torque is created

$$T = m_{coils} \times B^b = -B^{b \times} m_{coils}. \quad (4)$$

In the previous expression, $m_{coils} \in \mathbb{R}^3$ is the column matrix of the magnetic dipole moments for the three coils, and B^b is the Earth magnetic field at spacecraft resolved in body frame \mathcal{F}_b (see [1], Section 7.4.1). Let B^i be the Earth magnetic field at spacecraft expressed in inertial frame \mathcal{F}_i . Observe that B^i changes in time, at least because of the motion of the spacecraft along the orbit. Hence,

$$B^b(q, t) = C(q)B^i(t) \quad (5)$$

showing the explicit dependence of B^b on both q and t . The previous equations grouped together lead to the following system

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

where m_{coils} is the control variable. Let us characterize the dependence on time of $B^b(q, t)$, which is equivalent to characterize the time dependence of $B^i(t)$. Assume a circular orbit with radius R . Through the adoption of the so-called dipole model of the Earth magnetic field (see [10], Appendix H), obtain:

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

In the previous equation, μ_m is the total dipole strength, $r^i(t)$ is the position of the spacecraft resolved in \mathcal{F}_i , and $\hat{r}^i(t)$ is the column matrix of the direction cosines of $r^i(t)$. The coordinates of vector $\hat{m}^i(t)$ are the direction cosines of the Earth magnetic dipole expressed in \mathcal{F}_i which are written as follows:

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

where θ_m is the coelevation of the dipole, $\omega_e = 360.99$ deg/day is the average angular velocity of the Earth, and α_0 is the dipole right ascension at $t = 0$. Set $\mu_m = 7.746 \cdot 10^{15}$ Wb m and $\theta_m = 170.0^\circ$ as reported in [11].

Eq. (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 the orbital plane coordinate system a_p, b_p whose origin is at the Earth center and with a_p axis coinciding with the line of nodes. Then, the center of mass of the satellite is positioned in

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

where n is the orbital rate and ψ is spacecraft argument at time $t = 0$. It is possible to determine the mass center coordinates in the inertial frame \mathcal{F}_i from (9) by means of a rotation matrix which depends on the inclination *incl* of the orbit and on the right ascension of the ascending node (RAAN) Ω (see [1], Section 2.6.2). By inserting the expressions of the latter coordinates into (7), we obtain an explicit expression for $B^i(t)$.

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 $q_v \rightarrow 0$ and $\omega \rightarrow 0$. Reference [3] proposes the following stabilizing proportional derivative (PD)-like control law, which is a modification of those in [12, 13]:

$$m_{coils} = -m_{coils}^* \text{sat} \left(\frac{1}{m_{coils}^*} B^b(q, t) \times (\kappa_p q_v + \kappa_d \omega) \right). \quad (10)$$

In the previous equation, m_{coils}^* is the saturation limit on each magnetic dipole moment, “sat” denotes the standard saturation function, and κ_p and κ_d are parameters. As shown in [3], if the orbit’s inclination *incl* is not too low, there are large ranges for the values of $\kappa_p > 0$ and $\kappa_d > 0$ which lead to local exponential stability of equilibrium $(q, \omega) = (\bar{q}, 0)$ for the closed-loop systems (6) and (10).

The following attitude-only feedback, obtained as modification of one described in [12], is also proposed in [3]

$$\begin{aligned} \dot{\delta} &= \alpha(q - \beta\delta) \\ m_{coils} &= -m_{coils}^* \text{sat} \left(\frac{1}{m_{coils}^*} B^b(q, t) \times \left(\kappa_1 q_v + \kappa_2 \alpha \beta W(q)^T (q - \beta\delta) \right) \right). \end{aligned} \quad (11)$$

In (11), $\delta \in \mathbb{R}^4$ is an internal state of the controller, $\kappa_1 \ \kappa_2 \ \alpha \ \beta$ are all parameters, 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 . It has the advantage of not requiring the installation of rate gyros obtaining savings in cost, volume, and weight. As shown in [3], if the orbit’s inclination *incl* is not too low, there are large ranges for the values of $\kappa_1 > 0, \kappa_2 > 0, \alpha > 0, \beta > 0$ which lead to local exponential stability of equilibrium $(q, \omega, \delta) = (\bar{q}, 0, \frac{1}{\alpha} \bar{q})$ achieved for the closed-loop system (6) and (11).

For both control laws, there are no precise indications for choosing the parameters. In practice, they are usually computed recurring to a trial-and-error approach. This causes the two main limitations: (1) the computation is quite time-consuming, and (2) it is not systematic. This means that when a satisfactory set of parameter values is finally obtained, it is unknown whether continuing the search could allow to discover new parameter values producing a better performance of the closed-loop system. And, if the search is continued, it is unknown when it should be stopped and what could be the possible improvements obtained with this additional search. In any case, it can easily happen to neglect values providing an overall better performance, unless all possible values are tried. However, such an exhaustive

search is generally impracticable, because the search space is way too vast to be completely explored. Therefore, we describe here a different approach to find the values of the feedback parameters. Since the desired attitude is obtained when $q_v = 0$, we define the settling time t_{si} for each component q_i , with $i \in \{1, 2, 3\}$, as:

$$t_{si} := \min t \quad \text{s.t.} |q_i(t)| \leq \nu \quad \forall t \geq t_{si}. \quad (12)$$

In other words, this is the time needed for $|q_i|$ to permanently remain under ν . Value $0 < \nu < 1$ can be set depending on the desired final value of q_i . Finally, we can define the settling time t_s for the whole quaternion q as the time required by the slowest component of q_v , hence

$$t_s := \max_{i=1,2,3} t_{si} \quad (13)$$

Now, having set the spacecraft initial conditions to specific values, one can determine the values of the parameters that minimize the settling time t_s for each control law (10) and (11).

3. Determination of optimal parameters for the case of PD-like control

In the case of PD-like control (10), the minimization of the settling time can be formulated as follows

$$\min_{\kappa_p \geq 0, \kappa_d \geq 0} t_s. \quad (14)$$

In order to practically solve the problem, the feasible set should be reasonably bounded and not infinite. Hence, we define two upper bounds $\widehat{\kappa}_p$ and $\widehat{\kappa}_d$ for the gains κ_p and κ_d . These values can usually be determined for the specific problem; we do this for our case study in Section 3.3. Thus, the problem becomes as follows, where the dependence of t_s on the initial conditions $q_0 := q(0)$, $\omega_0 := \omega(0)$, and ψ (see (9)) is explicitly indicated,

$$\min_{(\kappa_p, \kappa_d) \in K} t_s(\kappa_p, \kappa_d, q_0, \omega_0, \psi), \quad (15)$$

and the feasible set is $K = \{(\kappa_p, \kappa_d) : 0 \leq \kappa_p \leq \widehat{\kappa}_p, 0 < \kappa_d \leq \widehat{\kappa}_d\}$. Even though t_s obviously depends on κ_p and κ_d and on initial conditions, it is not possible in practice to express this relation in analytical form. Moreover, t_s is discontinuous with respect to κ_p , κ_d and initial conditions. See for example in ([14], p. 233) the prove that the settling time of the step response is not continuous with respect to the system parameters.

Now, problem (15) has the following features: (i) an analytic expression of the objective function cannot be given, and (ii) the objective function is not continuous with respect to the decision variables. Therefore, we cannot use standard optimization techniques. Instead, we need to use derivative-free optimization.

3.1 From the simple min to the min-max problem

Given the initial conditions (q_0, ω_0, ψ) , problem (15) can be solved to optimality by using the global optimization derivative-free technique described in Section 3.2.

However, when the initial conditions change, then that solution may be no longer optimal. An example of this is given in Section 4.1. Therefore, a robust approach consists in searching for the optimal solution to (15) under the *worst* initial condition. Such a worst-case approach is commonly used in similar cases. In other words, we accept to pay something, in terms of objective function values, in the easy cases, but we obtain in return advantages in the more difficult situations. However, the worst initial condition is not a priori known, since it depends on the chosen values of κ_p and κ_d . Therefore, the problem cannot be decomposed and should be solved as a whole.

The set of values describing the initial conditions is given by:

$$S = \left\{ (q_0, \omega_0, \psi) : \|q_{0v}\| \leq 1, q_{04} = (1 - q_{0v}^T q_{0v})^{1/2}, \right. \\ \left. |\omega_{01}| \leq \hat{\omega}_{01}, |\omega_{02}| \leq \hat{\omega}_{02}, |\omega_{03}| \leq \hat{\omega}_{03}, 0 \leq \psi < 2\pi \right\} \quad (16)$$

Set S includes any possible initial attitude and any possible initial argument ψ for the spacecraft; only the magnitude of the initial angular rate is limited. The worst-case minimization of t_s can be modeled as follows:

$$\min_{(\kappa_p, \kappa_d) \in K} \max_{(q_0, \omega_0, \psi) \in S} t_s(\kappa_p, \kappa_d, q_0, \omega_0, \psi). \quad (17)$$

To use the derivative-free algorithm explained in Section 3.2, the feasible set of each optimization problem must be now converted into a hyperrectangle. This can be obtained by expressing $\|q_{0v}\| \leq 1$ in spherical coordinates (ρ, ϕ, θ) :

$$S = \left\{ (q_0, \omega_0, \psi) : q_{01} = \rho \sin \theta \cos \phi, q_{02} = \rho \sin \theta \sin \phi, q_{03} = \rho \cos \theta, \right. \\ \left. q_{04} = (1 - q_{0v}^T q_{0v})^{1/2}, 0 \leq \rho \leq 1, 0 \leq \phi < 2\pi, 0 \leq \theta \leq \pi, \right. \\ \left. |\omega_{01}| \leq \hat{\omega}_{01}, |\omega_{02}| \leq \hat{\omega}_{02}, |\omega_{03}| \leq \hat{\omega}_{03}, 0 \leq \psi < 2\pi \right\}. \quad (18)$$

The dependence of t_s on q_0 can be expressed as dependence on the variables (ρ, ϕ, θ) , as follows. We introduce the hyperrectangle:

$$H = \left\{ (\rho, \phi, \theta, \omega_0, \psi) : 0 \leq \rho \leq 1, 0 \leq \phi < 2\pi, 0 \leq \theta \leq \pi, \right. \\ \left. |\omega_{01}| \leq \hat{\omega}_{01}, |\omega_{02}| \leq \hat{\omega}_{02}, |\omega_{03}| \leq \hat{\omega}_{03}, 0 \leq \psi < 2\pi \right\}. \quad (19)$$

Now, the min-max problem (17) can be equivalently reformulated as follows:

$$\min_{(\kappa_p, \kappa_d) \in K} \max_{(\rho, \phi, \theta, \omega_0, \psi) \in H} t_s(\kappa_p, \kappa_d, \rho, \phi, \theta, \omega_0, \psi). \quad (20)$$

3.2 The whole derivative-free optimization approach

This section explains the min-max procedure to solve problem (20). Its building blocks are given by the following derivative-free algorithms:

3.2.1 The global strategy

Lipschitzian methods constitute a main approach in non-differentiable optimization. However, they are limited by their requirement of knowing the value of the Lipschitz constant. On the other hand, the search may be conducted without knowing the Lipschitz constant if we use DIRECT-type algorithms [15]. These techniques are based on the partitioning of the feasible region into hyperrectangles

and on their examination in a specific order. The feasible region starts as a single hyperrectangle that is internally normalized to a unit hyperrectangle. At each iteration k , the algorithm partitions the hyperrectangles identified in the previous $(k-1)$ th iteration, obtaining a collection of smaller hyperrectangles $\mathcal{H}^{(k)} = \{H_1, \dots, H_k\}$, and evaluates the objective function in their central points. The computation of the objective value allows to identify potentially optimal hyperrectangles within $\mathcal{H}^{(k)}$. These hyperrectangles are further partitioned and investigated in the next $(k+1)$ th iteration of the algorithm, while the rest of them is simply discarded. The algorithm stops when the hyperrectangle size becomes very small or when the maximum number of iterations is reached. This algorithm guarantees to converge to the global optimum of the function if sampling is dense enough. However, a large number of function evaluations might be needed to obtain dense sampling, and consequently large computational times could be required.

3.2.2 The local strategy

When the function under optimization is sufficiently regular, it can be optimized by using gradient-based methods, which require the computation of the first-order derivatives of the objective function. However, the basic strategy underlying these methods could be replicated even if no information on the derivatives is available. This is the basic idea of the SDBOX algorithm [16], which can be described as follows. After the selection of a starting point, it cyclically determines a feasible and *good* descent direction for the objective function, and then it performs a *sufficiently* large step along such direction. To find the good feasible descent direction, at each iteration k , the algorithm computes the local behavior of the objective function at the variation of the i -th variable. If a move of length α along d_i gives a feasible point where the function is reduced enough, the algorithm applies a linesearch technique d_i to find the stepsize α_k . Otherwise, the algorithm tries the same operations in the opposite direction $-d_i$. If both d_i and $-d_i$ are not able to obtain a sufficient decrease, the stepsize α is decreased. The linesearch technique does not require information on the slope of the objective function. Note that the convergence of the algorithm was proved in [16] for minimization of a continuously differentiable function; however, the same technique is often used to optimize different types of functions. This approach is considerably faster than the global strategy; however it is well known that a poor choice of the initial point leads to poor solutions.

3.2.3 The whole procedure

To simplify the description of the robust optimization approach, we now rename the set of variables (κ_p, κ_d) as x belonging to a feasible set $F_x = \{lb_i \leq x_i \leq ub_i, i = 1, \dots, n\} \subset \mathbb{R}^n$ (in our case $F_x = K$ and $n = 2$) and the set of initial conditions $(\rho, \phi, \theta, \omega_0, \psi)$ as y belonging to a feasible set $F_y = \{lb_j \leq y_j \leq ub_j, j = 1, \dots, m\} \subset \mathbb{R}^m$ (in our case $F_y = H$ and $m = 7$, since ω_0 has three components). Moreover, we now use $g(x, y)$ to denote the objective function (in our case t_s). Its analytical expression is not known; however its value can be computed via a software simulation.

The upper-level minimization problem $\min_{x \in F_x} g(x, y)$ must be solved by means of a global strategy applied one time: no starting points can be considered here; hence, there is no ground for a local strategy. Thus, the procedure will contain an external loop implementing the global strategy. This loop essentially computes the value of g

obtained for a number $maxeval_ext$ of points x . Let \bar{x} be one of them. The evaluation of \bar{x} requires however to solve one lower-level maximization problem $\max_{y \in F_y} g(\bar{x}, y)$.

Hence, the lower-level problem should be solved many times, and its corresponding computation time must be reduced to a few seconds. We can solve it in three ways: (1) with the global strategy but using a necessarily small maximum number of function evaluations $maxeval_int$; (2) with the local strategy, allowing the same number of function evaluations from a single start; and (3) with the local strategy, using a multistart from different starting points, still allowing $maxeval_int$ total function evaluations. This last option allows to use information that happens to be available in the choice of the starting points. In our case, from the physics of the problem, we suppose that good solutions are in the vicinity of extremal values of angular velocity, so we take them as starting points. We report the whole procedure:

Procedure 1: Solve min-max

Input: A $g(x, y)$ computable by means of a software simulation for $x_i \in [lb_i, ub_i]$ with $i = 1, \dots, n$ and $y_j \in [lb_j, ub_j]$ with $j = 1, \dots, m$.

Output: A robust optimum value $x^* = \arg \min_{x \in F_x} \left(\max_{y \in F_y} g(x, y) \right)$.

External loop:

Solve the upper-level problem $\min_{x \in F_x} f(x)$, with $f(x) = \max_{y \in F_y} g(x, y)$ by using

DIRECT for $maxeval_ext$ evaluations of f , and return x^* .

The k -th evaluation works with point $x^{(k)}$

Internal loop:

Solve the lower-level problem $\max_{y \in F_y} g(x^{(k)}, y)$

by using

$\left\{ \begin{array}{l} \text{DIRECT performing } maxeval_int \text{ evaluations of } g \\ \text{SDBOX single start performing } maxeval_int \text{ evaluations of } g \\ \text{SDBOX multistart with } s \text{ starting points, performing} \\ \frac{maxeval_int}{s} \text{ evaluations of } g \text{ for each of them} \end{array} \right.$

4. Computational results for the case of PD-like control

We apply the above approach to solve the numerical example studied in [3]. The inertia matrix of the spacecraft is equal to $J = \text{diag}[27, 17, 25]$ kgm²; the saturation level for each magnetic dipole moment is given by $m_{coils}^* = 10$ Am². The orbit has an inclination of $incl = 87^\circ$ and an altitude of 450 km. The orbital period in these conditions is about 5609 s. The value Ω of RAAN is 0. Given these values, we can compute an upper bound $\hat{\kappa}_p$ for κ_p . Simulation shows that $\|B^i(t)\| \geq B_{min} = 2.4 \times 10^{-5}$ T for the considered orbit; since $\|B^b(q, t)\| = \|B^i(t)\|$, it follows that $\|B^b(q, t)\| \geq B_{min}$. Assume that the resolution of the attitude sensor is $q_s = 0.04$ in terms of quaternion component. In this case, an upper bound for κ_p can be computed by enforcing that each component of the term $B^b(q, t) \times \kappa_p q_v$ (see (10)) does

not exceed the saturation limit $m_{coils}^* = 10 \text{ Am}^2$ when $B^b(q, t)$ and q_v are orthogonal, $\|B^b(q, t)\| = B_{min}$ and $|q_i| = q_r$ for one index $i \in 1, 2, 3$, and $q_i = 0$ for the other ones. This is achieved if κ_p is smaller than $m_{coils}^*/(B_{min}q_r) = 1.0417 \times 10^8$; thus, we set $\hat{\kappa}_p = 10^8$. Assuming that the attitude rate sensor has a resolution of $\omega_r = 4 \times 10^{-4}$ rad/s, by a parallel argument it follows that an upper bound for κ_d is given by $m_{coils}^*/(B_{min}\omega_r) = 1.0417 \times 10^9$; thus, we set $\hat{\kappa}_d = 10^9$.

4.1 Fixed initial conditions

Consider an initial state characterized by attitude equal to the target attitude $q_0 = \bar{q}$ (which corresponds to having $\rho = 0$ and any value for ϕ and θ) and by the following initial angular rate:

$$\omega_0 = [0.02 \quad 0.02 \quad -0.03]^T \text{ rad/s.} \quad (21)$$

This example corresponds to a spacecraft possessing the desired attitude and no angular momentum, which received an impact from a small object leading to an instantaneous change in the spacecraft angular rate. We assign to ψ a random value over the interval $[0, 2\pi[$ by setting $\psi = 0.332$ rad.

Gains κ_p and κ_d have been found by trial-and-error in [3] as $\kappa_p = 2 \times 10^5$ and $\kappa_d = 3 \times 10^8$. The settling time corresponding to these values of the gains is $t_s = 4.002$ orbital periods, while the vast majority of possible gain values would produce settling times larger than 10 orbital periods. On the contrary, if we use the optimization algorithm described in Section 3.2, we obtain the following fixed initial condition optimal gains after only 32,821 iterations (i.e., 251 s of computation):

$$\begin{aligned} \kappa_p &= 205761.316872 \\ \kappa_d &= 99382716.049383 \end{aligned} \quad (22)$$

The corresponding optimal settling time is $t_s = 1.775$ orbital periods. Hence, the optimization algorithm can provide a significant improvement in convergence speed.

However, the above gains might become no longer optimal if we assume different initial conditions. Consider, for example, the case of initial conditions with the same attitude $q_0 = \bar{q}$ (i.e., $\rho = 0$ and any ϕ and θ), the same argument $\psi = 0.332$ rad, but a different initial angular rate

$$\omega_0 = [0.1 \quad 0.1 \quad 0.1]^T \text{ rad/s.} \quad (23)$$

The above gains yield a settling time $t_s = 3.562$, and they are no longer optimal, as shown in Section 4.2. Indeed, the optimal values of the gains would need to be determined for every possible initial condition, which is clearly impossible in practice. As seen in Section 3.1, we can instead search for the gain values providing the best performance under the worst initial condition. It has been found that the worst initial conditions corresponding to gains (22) are

$$\begin{aligned} \rho &= 0.5, \quad \phi = 0.0, \quad \theta = 2.356194490192345, \quad \psi = 1.570796326794897, \\ \omega_0 &= [-0.1 \quad 0.1 \quad 0.1]^T. \end{aligned} \quad (24)$$

	DIRECT + DIRECT	DIRECT + SDBOX single start	DIRECT + SDBOX multistart
κ_p	205493.82716049382	203930.04115226338	209581.61865569273
κ_d	117283950.61728397	154320987.65432101	117283950.61728397
t_s	3.359	3.142	3.742
ext eval	10,641	11,127	10,011
Time (s)	172,242.6	57,255.1	89,640.8

Table 1.
 Results of Solve min-max with $maxiter_{ext} = 10,000$ and $maxiter_{int} = 2200$.

Gains	DIRECT	SDBOX multistart	Grid search
	<i>eval = 40,000</i>	<i>eval = 40,000 (5000 × 8)</i>	<i>eval = 78,125 (5⁷)</i>
$\kappa_p = 205493.82716049382$ $\kappa_d = 117283950.61728397$	3.876	3.872	3.990
$\kappa_p = 203930.04115226338$ $\kappa_d = 154320987.65432101$	4.223	4.149	4.362
$\kappa_p = 209581.61865569273$ $\kappa_d = 117283950.61728397$	3.895	3.744	4.016

Table 2.
 Accurate evaluation of the previously obtained gains.

Under conditions (24), the above values of κ_p and κ_d yield a settling time $t_s = 4.338$.

4.2 Variable initial conditions

We now search for κ_p and κ_d using the robust optimization approach. We allow initial conditions to vary as follows: $0 \leq \rho \leq 1$, $0 \leq \phi \leq 2\pi$, $0 \leq \theta \leq \pi$, $|\omega_{01}| \leq 0.1$, $|\omega_{02}| \leq 0.1$, $|\omega_{03}| \leq 0.1$, and $0 \leq \psi \leq 2\pi$. We allow a maximum of 2,200 function evaluations in each internal loop, in order to practically solve the problem. **Table 1** reports the results of Procedure Solve min-max with $maxeval_{ext} = 10,000$ and $maxeval_{int} = 2200$.

When using SDBOX multistart, the starting points are $\rho = 0.5$, $\phi = \pi$, $\theta = \pi/2$, $\psi = \pi$, and all the eight combinations of extreme angular velocities

$$[0.1 \ 0.1 \ 0.1]^T, [0.1 \ 0.1 \ -0.1]^T, \dots, [-0.1 \ -0.1 \ -0.1]^T. \quad (25)$$

These results show that (i) SDBOX multistart is able to reach the highest value of the objective t_s even with the small number of computations allowed in the internal loop and (ii) the different results in the solution of the lower-level problems cause a different evolution of the upper-level search. To understand which among the three points above is the best choice, we continue the analysis in **Table 2**. We fix parameters κ_p and κ_d , and we solve the maximization problem with increased accuracy by allowing more function evaluations. We do this by means of global strategy, local multistart strategy (which proved to dominate the single start one), and an exhaustive grid search, which is much slower but used here as reference.

This analysis shows that the first point has the smallest maximum settling time. In conclusion, the robust optimal solution is:

$$\kappa_p = 205493.82716049382, \quad \kappa_d = 117283950.61728397. \quad (26)$$

To evaluate the performance of the above solution (26), we try it with the three different initial conditions reported above. If the initial state is (21), the settling time becomes $t_s = 1.945$ (instead of 1.775). Hence, there is a small worsening. Clearly, no improvement was possible because gains (22) are optimal for state (21). If the initial state is given by (23), we obtain $t_s = 2.958$ (instead of 3.562). Hence, there is an improvement. This could be expected, though not guaranteed, since gains (22) have a higher worst-case result than gains (26).

Finally, if the initial state is given by (24), then we obtain $t_s = 3.370$ (instead of 4.338). Hence, there is a substantial improvement. Indeed, this was certain, since gains (26) have a worst-case result considerably better than gains (22). In conclusion, the gains computed with the proposed robust optimization approach offer improvements in the difficult situations. Moreover, even if they may be suboptimal in the easier cases, the balance appears beneficial.

5. Determination of optimal parameters for the case of attitude-only feedback control

As observed, the objective function t_s is not continuous with respect to the parameters, and this introduces numerical difficulties in solving the optimization problem. Thus, one can consider an alternative objective function named integral time absolute error (ITAE) [8], indicated by Γ :

$$\Gamma = \int_0^{T_f} t \|q_v(t)\| dt \quad (27)$$

where $\|\cdot\|$ represents the Euclidean norm and T_f is a time which is selected large enough. The ITAE carries the benefit of being continuous with respect to the design parameters. Continuous differentiability cannot be guaranteed. However, since $(q, \omega, \delta) = (\pm\bar{q}, 0, \frac{1}{\beta}\bar{q})$ are equilibria of the closed-loop system (6) and (11), it is very unlikely that q_v becomes $[0 \ 0 \ 0]^T$ at some finite time. Note that this would make Γ not continuously differentiable with respect to the design parameters because $\|q_v\|$ is not continuously differentiable at $q_v = [0 \ 0 \ 0]^T$.

Minimizing the ITAE is known to lead to nearly optimal solutions with regard to the settling time. In fact, in very simple situations, it is shown analytically that minimizing the ITAE leads to solutions that minimize the settling time. In more complex scenarios, it was numerically shown that minimizing the ITAE gives solutions that are very close to the optimal ones in terms of settling time (see [8]).

By introducing physically reasonable upper bounds $\hat{\kappa}_1, \hat{\kappa}_2, \hat{\alpha}, \hat{\beta}$ for the design parameters, we obtain the feasible set $K = \{(\kappa_1, \kappa_2, \alpha, \beta) : 0 \leq \kappa_1 \leq \hat{\kappa}_1, 0 \leq \kappa_2 \leq \hat{\kappa}_2, 0 \leq \alpha \leq \hat{\alpha}, 0 \leq \beta \leq \hat{\beta}\}$. Now, our optimization problem is:

$$\min_{(\kappa_1, \kappa_2, \alpha, \beta) \in K} \Gamma. \quad (28)$$

Given specific initial conditions of the spacecraft, problem (28) can be solved by a suitable use of derivative-free techniques. However, if the initial conditions of the

spacecraft change, that solution might be no longer optimal. Since several different initial conditions for the spacecraft are possible in practical situations, a robust solution is an optimal solution to problem (28) under the *worst* spacecraft initial conditions. Such a worst-case optimization is widely employed in such scenarios, because by adopting the latter approach, we can give an efficient bound on the objective value in spite of the uncertainty on the spacecraft initial conditions. However, the worst initial conditions for the spacecraft cannot be determined a priori, since they depend on the selected values of $\kappa_1, \kappa_2, \alpha, \beta$. 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:

$$S = \left\{ (q_0, \omega_0, \psi, \alpha_0) : \|q_{0v}\| \leq 1, q_{04} = (1 - q_{0v}^T q_{0v})^{1/2}, \right. \\ \left. |\omega_{01}| \leq \hat{\omega}_{01}, |\omega_{02}| \leq \hat{\omega}_{02}, |\omega_{03}| \leq \hat{\omega}_{03}, 0 \leq \psi < 2\pi, 0 \leq \alpha_0 < 2\pi, \right\}. \quad (29)$$

Note that S includes all possible initial attitudes, all possible right ascensions of the Earth magnetic dipole at time $t = 0$, and all possible initial arguments ψ for the spacecraft. It only constrains the amplitude of the initial angular velocity. The minimization of Γ under the worst spacecraft initial conditions is equivalent to formulating the following min-max problem:

$$\min_{(\kappa_1, \kappa_2, \alpha, \beta) \in K} \max_{(q_0, \omega_0, \psi, \alpha_0) \in S} \Gamma. \quad (30)$$

To apply the optimization techniques, we convert the feasible set of each optimization problem into a hyperrectangle by expressing the set $\|q_{0v}\| \leq 1$ in spherical coordinates (ρ, ϕ, θ) :

$$S = \left\{ (q_0, \omega_0, \psi, \alpha_0) : q_{01} = \rho \sin \theta \cos \phi, q_{02} = \rho \sin \theta \sin \phi, q_{03} = \rho \cos \theta, \right. \\ \left. q_{40} = (1 - q_{0v}^T q_{0v})^{1/2}, 0 \leq \rho \leq 1, 0 \leq \phi < 2\pi, 0 \leq \theta \leq \pi, \right. \\ \left. |\omega_{01}| \leq \hat{\omega}_{01}, |\omega_{02}| \leq \hat{\omega}_{02}, |\omega_{03}| \leq \hat{\omega}_{03}, 0 \leq \psi < 2\pi, 0 \leq \alpha_0 < 2\pi, \right\}. \quad (31)$$

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

$$H = \left\{ (\rho, \phi, \theta, \omega_0, \psi, \alpha_0) : 0 \leq \rho \leq 1, 0 \leq \phi < 2\pi, 0 \leq \theta \leq \pi, \right. \\ \left. |\omega_{01}| \leq \hat{\omega}_{01}, |\omega_{02}| \leq \hat{\omega}_{02}, |\omega_{03}| \leq \hat{\omega}_{03}, 0 \leq \psi < 2\pi, 0 \leq \alpha_0 < 2\pi, \right\}, \quad (32)$$

the min-max problem (30) can be equivalently reformulated as follows:

$$\min_{(\kappa_1, \kappa_2, \alpha, \beta) \in K} \max_{(\rho, \phi, \theta, \omega_0, \psi, \alpha_0) \in H} \Gamma. \quad (33)$$

5.1 Combining global and local search

To simplify the description of the proposed approach, we now rename the set of design parameters $(\kappa_1, \kappa_2, \alpha, \beta)$ as x belonging to a feasible set $F_x = \{l b x_i \leq x_i \leq u b x_i, i = 1, \dots, n\} \subset \mathbb{R}^n$ (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 $F_y =$

$\{l b y_j \leq y_j \leq u b y_j, j = 1, \dots, m\} \subset \mathbb{R}^m$ (in our case $F_y = H$ and $m = 8$, since ω_0

possesses three components). Indicate by $f(x, y)$ the function giving the objective value (in our case Γ). When y is fixed (the initial conditions are assigned), we simply write \bar{y} in it; when x is fixed (the design parameters are assigned), we write \bar{x} in it. Problem (28) can be written as

$$\min_{x \in F_x} f(x, \bar{y}) \quad (34)$$

and may be tackled by a global derivative-free optimization algorithm of the type of DIRECT [7], already described in Section 3. 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. Due to the so-called *everywhere dense* property, such an algorithm reaches a global optimum if the sampling is dense enough. However, a dense search may require that the function is evaluated many times.

In our case, all $lbx_i = 0$, since design parameters have to be greater than or equal to zero. Values ubx_i can typically be set to very large values based on some physical considerations. For instance, κ_1 can easily become equal to 10^9 , since its maximum feasible value can be determined knowing the saturation level of the coil moments, the minimum amplitude of the geomagnetic field, and the attitude sensor resolution. However, when considering these values, a sufficiently dense exploration of the feasible set F_x requires a number of function evaluations such that the corresponding run time is impracticable. On the other hand, a sampling that uses a practically sustainable number of function evaluations does not produce solutions substantially better than random solutions. The presence of a large number of local minima makes the optimization task particularly difficult. In such conditions, the evaluation of the generic H_h using only one point may be very inaccurate at the first iterations of the algorithm, because the dimension of the hyperrectangles is too large. By proceeding with the iterations, the hyperrectangles becomes smaller, but their number, and consequently the run time, increases exceedingly.

On the other hand, if a fast but effective probing technique to early identify the “promising” H_h would be available, then one could explore densely only such promising regions in reasonable times. Hence, a probing technique based on the use of the local derivative-free optimization algorithm SDBOX was proposed in [6]. This algorithm was originally presented in [16] as a globally convergent algorithm for the minimization of a continuously differentiable function, but in practice it can be employed to optimize different types of functions as a good trade-off between efficiency and convergence properties. Most interestingly for our case, given an initial guess $(\kappa'_1, \kappa'_2, \alpha', \beta') = x'$, this algorithm should be able to find good solutions in short times in the neighborhood of x' . We describe below a solution approach combining these global and local strategies to solve problem (28).

Procedure 2: 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 $\mathcal{H}^{(1)} = \{H_1, \dots, H_p\}$ 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 (28).

Depending on the practical case, the number of hyperrectangles p in the partitioning phase (step 1) and the number of iterations *maxeval* in the evaluation phase (step 2) must be selected in order to allow a fixed computational time to the evaluation phase. Indeed, we search for a compromise between speed and effectiveness at this stage. Those parameters must be tuned by also 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 . However, there is no need that all the hyperrectangles in the initial partition $H^{(1)}$ have the same size. Their size could be growing with the absolute values of the coordinates so that p is kept smaller. The number *maxsubsets* in the selection phase (step 3) must be chosen so that F_x^* remains much smaller than F_x ; otherwise the benefits of the proposed procedure is reduced. The number of function evaluations *maxiter* in the standard DIRECT phase (step 5) is chosen so that the search in F_x^* is dense enough. This is now possible because of the size reduction in F_x . The post-optimization phase (step 6) can be executed with SDBOX or with CS-DFN [17], which is another linesearch-based method which uses a dense set of search directions and not only the coordinates ones. CS-DFN does not require f to be continuously differentiable; however, it could be more computationally expensive on smooth problems. For example, in our case, Γ is at least continuous, and even if there are no theoretical arguments to ensure its continuous differentiability, in practice this property may often occur. In conclusion, we can choose between SDBOX and CS-DFN depending on the presence or absence of the continuous differentiability of f , and if we set *maxpost* to be sufficiently large, we obtain 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. Note that the accuracy and the properties of the evaluation phase can be modified, depending on the computational request of the practical case, either by modifying the number of iterations or even by employing an alternative evaluation algorithm, still keeping the same algorithmic framework.

On the other hand, problem (33) is made of an upper-level minimization problem and a lower-level maximization one. By changing the names of the set of variables as explained at the beginning of the section, the problem becomes

$$\min_{x \in F_x} \left(\max_{y \in F_y} f(x, y) \right) = \min_{x \in F_x} g(x) \quad (35)$$

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

$$g(\bar{x}) = \max_{y \in F_y} f(\bar{x}, y). \quad (36)$$

The upper-level problem is solved through an external loop applying the combination of local and global search described as Procedure 2. This loop computes, using a parameter $maxeval_ext$, to define the overall maximum number of function evaluations and the value of g corresponding to different points of F_x . Let \bar{x} be one of them; then the evaluation of $g(\bar{x})$ requires 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 need excessive time. Therefore, the local strategy appears to be the only feasible choice for the lower-level problem. However, taking as initial guess 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, based on the problem physics, it can be assumed that good solutions of the maximization problem are in the neighborhood of extreme values of angular velocity. Therefore, we solve the lower-level problem in a nested loop by using a local search with multistart. We take as initial guesses the eight combinations of extreme values for the three components of the angular velocity $\pm\omega_{\hat{0}1}$, $\pm\omega_{\hat{0}2}$, and $\pm\omega_{\hat{0}3}$. This can be done with SDBOX or CS-DFN, depending on the presence or absence of the continuous differentiability of f . In our case, Γ is at least continuous also with regard to the initial conditions of the spacecraft. Again, there are no theoretical arguments to ensure its continuous differentiability; however in practice this may often happen. For each solution of the lower-level problem, we allow a necessarily small maximum number of function evaluations $maxeval_int$. In conclusion, this produces a solution x_R^* for problem (33). We report here the whole procedure:

Procedure 3: Solve min-max combining global and local search

Input: $Af(x, y)$ 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

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

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 Procedure 2 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.

6. Computational results for the case of attitude-only feedback control

We apply the above method to solve the numerical example presented in [3]. The spacecraft inertia matrix is $J = \text{diag}[27, 17, 25]$ kg m², and the saturation level for each dipole moment is $m_{coils}^* = 10$ A m². The orbit inclination is $incl = 87^\circ$, and the orbit altitude is 450 km; the right ascension of the ascending node Ω is equal to 0. Upper bounds $\kappa_1, \kappa_2, \hat{\alpha}, \hat{\beta}$ are selected as $(10^9, 10^9, 10^4, 10^{-3})$.

At the beginning we consider the easier situation of known spacecraft initial conditions, and we present, in Section 6.1, the results of Procedure 2 in solving this problem. We report also a comparison with the classical DIRECT method. Subsequently, in Section 6.2, the more realistic case of a spacecraft having variable initial conditions is considered.

6.1 Fixed initial conditions

We consider here the case of the above described spacecraft with known fixed initial conditions; thus we deal with problem (28) using the following values:

$$(\rho, \phi, \theta, \omega_0, \psi, \alpha_0) = (0, 0, 0, 0.02, 0.02, -0.03, 0.9416, 4.5392). \quad (37)$$

The best solution obtained in [3] by trial and error search achieves a value of ITAE = 3.7×10^7 , while the vast majority of the solutions have the ITAE limit value of about 1.2×10^9 . This upper limit is related to the value of T_f in the definition of ITAE (27), which is chosen equal to 56,009 s corresponding to 10 orbital periods. Practically this means that when we reach the limit value of ITAE, the corresponding settling time would be roughly larger than 10 orbital periods. Then, that solution is not an attractive 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 run the simulations that compute the ITAE in practice.

Table 3 reports two solution attempts carried on with the standard DIRECT algorithm on the whole feasible set K with, respectively, 50,000 and 100,000 iterations, followed by 1000 iterations of local search refinement using CS-DFN. In spite of the significant computational effort (the running times of these experiments, respectively, correspond to about 3 days and 1 week), the obtained solutions have values of ITAE greater than 1.1×10^9 that is not very different from the ITAE value of a random solution. Evidently, the search was not dense enough to explore

Algorithm	Solution	Obj. value	Time
DIRECT 50,000 + CS-DFN 1000	$\kappa_1 = 913405022.139$ $\kappa_2 = 195426826.870$ $\alpha = 9794.170752422$ $\beta = 0.0000000000000$	1,142,470,478.101	262,000 + 5460 s
DIRECT 100,000 + CS-DFN 1000	$\kappa_1 = 500798437.500$ $\kappa_2 = 159788790.177$ $\alpha = 9996.679486501$ $\beta = 0.0000000000003$	1,129,234,873.703	565,200 + 5040 s
Procedure 2: Combining global and local strategies	$\kappa_1 = 246494.579020$ $\kappa_2 = 233333315.349$ $\alpha = 92.5925925927$ $\beta = 0.000129629629$	8,021,573.4077	201,500 + 15 + 4 s

Table 3.
 Comparison of Procedure 2 and standard DIRECT with 50,000 or 100,000 iterations.

the feasible set as it would be needed. Other similar attempts with standard DIRECT do not achieve better results. The same table also reports Procedure 2 of Section 5.1. As observable, this procedure is able to reach a much better solution, with a value of ITAE of about 8×10^6 . Note that this solution is also considerably better than the best solution obtained by trial and error in days of work.

The promising region of the feasible set K has been identified with the previously described probing technique. The values of $maxelav$ and p are selected so that this step is executed in reasonable time, according to the following considerations. One single function evaluation takes a time which is very variable and goes from fractions of seconds to several tenths of seconds; but a very rough average function evaluation time can be assessed as equal to 1 s. By selecting $maxeval = 10$, the evaluation of each hyperrectangle would require roughly 10 s. Thus, to finish within 2 or 3 days of computation, we select $p = 40,960$, 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 do not have the same size; they increase with the absolute value of the coordinates. The following hyperrectangle is obtained as convexification of the collection of the most promising regions, as in Step 4 of Procedure 2:

$$K^* = \{(\kappa_1, \kappa_2, \alpha, \beta) : 200000 \leq \kappa_1 \leq 260000, 200000000 \leq \kappa_2 \leq 310000000, 0 \leq \alpha \leq 100, 0 \leq \beta \leq 0.001\}. \quad (38)$$

The determination of K^* actually needs 409,600 iterations of SDBOX and 201,500 s of computations (about 56 hours). Note that the time necessary for each function evaluation is generally much faster in K^* , where it can be less than 0.01 s than in the rest of K . Now, by applying standard DIRECT strategy over the feasible set K^* , as in Step 5 of Procedure 2, after 3007 iterations and only 15 s, we obtain the solution:

$$\kappa_1 = 200781.893004, \quad \kappa_2 = 226268861.454, \quad \alpha = 39.3004115226, \quad \beta = 0.0005 \quad (39)$$

whose value is $ITAE = 9.072 \times 10^6$. 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 . To better appreciate the difference, consider that an exploration of K with the same degree of density used on K^* would require $15 \times 15,151,515.15 = 227,272,727.25$ s 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 (39) can be further refined by using the local search strategy, as in Step 6 of Procedure 2. By performing 1000 iterations of the local search CS-DFN, which can move along a dense set of directions [17], the solution (39) is improved in 4 s to $ITAE = 8.021 \times 10^6$ with the solution:

$$\begin{aligned} \kappa_1 = 246494.579020, \quad \kappa_2 = 233333315.349, \quad \alpha = 92.5925925927, \\ \beta = 0.000129629629. \end{aligned} \quad (40)$$

We also tested in this post-optimization phase the local search SDBOX, which moves only along the coordinate directions [16]. With 5000 additional iterations of SDBOX that requires 11 s, solution (39) is improved to the following new solution, which has $ITAE = 9.062 \times 10^6$

$$\begin{aligned} \kappa_1 &= 200781.893004, \quad \kappa_2 = 226268812.597, \quad \alpha = 39.3004115226, \\ \beta &= 0.000500000024. \end{aligned} \quad (41)$$

Solution (40) corresponds to x^{**} of Procedure 2. Note that this is actually an approximation of a theoretical optimal solution. Since there is probably no optimal solutions available for comparison, we evaluate it by using the following considerations.

The settling time (see (12) and (13)) corresponding to (40) is computed as $t_s = 6280$ s. Then, we search for a lower bound \underline{t}_s on the minimum settling time of the considered case, through physical considerations. The initial conditions (37) correspond to having the spacecraft with the desired attitude but with a nonzero initial angular velocity $\omega(0) = [0.02 \ 0.02 \ -0.03]^T$. 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 angular 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

$$\ddot{\phi} = J_x T_x \quad (42)$$

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 and 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 (42) 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 [18], Section 7.2)

$$\underline{t}_{sx} = \frac{J_x}{T^*} (1 + \sqrt{2}) |\omega_x(0)| = 1505 \text{ s.} \quad (43)$$

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

$$\begin{aligned} \underline{t}_{sy} &= \frac{J_y}{T^*} (1 + \sqrt{2}) |\omega_y(0)| = 948 \text{ s.} \\ \underline{t}_{sz} &= \frac{J_z}{T^*} (1 + \sqrt{2}) |\omega_z(0)| = 2091 \text{ s.} \end{aligned} \quad (44)$$

Then, $\underline{t}_s = \max \{ \underline{t}_{sx}, \underline{t}_{sy}, \underline{t}_{sz} \} = 2091$ s. Therefore, solution (40) takes just about 70 min 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 reaches the desired rest position. Thus, solution (40) does not appear to be too far from an optimal solution. Note also that the above computed bound is very conservative, in the sense that the minimum spacecraft evolution time surely cannot require less, though it could very easily require more.

6.2 Variable initial conditions

Now, problem (33) is solved by employing the above described Procedure 3. 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 described in Procedure 3, we employ as initial guesses the eight combinations of extreme values for the three components of the angular velocity $\pm\omega_{01}$, $\pm\omega_{02}$, and $\pm\omega_{03}$. We allow 64 iterations per point, for a total of 512 iterations per single lower-level problem, which requires slightly less than 3 s 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 s (about 95 h)

$$\begin{aligned} \kappa_1 &= 227777.777778, \quad \kappa_2 = 294444444.444, \\ \alpha &= 83.3333333333, \quad \beta = 0.00061095869532. \end{aligned} \tag{45}$$

The ITAE value of this solution is 2.176×10^7 for the initial conditions (37) instead of $ITAE = 8.021 \times 10^6$ of solution (40). However, this solution 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 achievable by varying the initial conditions in H for solution (40) is $ITAE = 1.103 \times 10^9$ that is very close 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 (33) by using standard DIRECT algorithm on the whole feasible set K allowing no more than 1 day of computation. More time-consuming attempts in solving problem (33) using standard DIRECT algorithm cannot be practically accomplished. This holds for the following motivations. In the case of fixed initial conditions (Section 6.1), 1 week of computation was not enough to explore the search space. In this case, there is an internal loop requiring 512 function evaluations instead of one single function evaluation. Hence, any serious attempt would need to allocate months of computation and would obtain results probably similar to those obtained in Section 6.1.

7. Conclusions

The attitude of a spacecraft can be controlled using only magnetorquers by means of a PD-like control or an attitude feedback. However, for both control laws, design parameters must be assigned. These parameters may be conveniently selected so that they minimize the spacecraft settling time, or an indirect measure of it, 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 objective obtainable by varying the initial conditions. This chapter has described solution approaches based on an innovative use of global and local derivative-free optimization techniques to practically solve these computationally demanding problems. This approach is able to provide robust solutions to the considered application in reasonable times.

IntechOpen

Author details

Renato Bruni^{1*†} and Fabio Celani^{2†}


1 Department of Computer Control and Management Engineering
“Antonio Ruberti”, Sapienza University of Rome, Roma, Italy

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

*Address all correspondence to: bruni@diag.uniroma1.it

† These authors contributed equally.

IntechOpen

© 2020 The Author(s). Licensee IntechOpen. Distributed under the terms of the Creative Commons Attribution - NonCommercial 4.0 License (<https://creativecommons.org/licenses/by-nc/4.0/>), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited. 

References

- [1] Sidi MJ. *Spacecraft Dynamics and control*. New York: Cambridge University Press; 1997
- [2] Silani E, Lovera M. Magnetic spacecraft attitude control: A survey and some new results. *Control Engineering Practice*. 2005;**13**(3):357-371
- [3] Celani F. Robust three-axis attitude stabilization for inertial pointing spacecraft using magnetorquers. *Acta Astronautica*. 2015;**107**:87-96
- [4] Bruni R, Celani F. Determining optimal parameters in magnetic spacecraft stabilization via attitude feedback. *AIP Proceedings*. 2016;**1776**: 090032
- [5] Bruni R, Celani F. A robust optimization approach for magnetic spacecraft attitude stabilization. *Journal of Optimization Theory and Applications*. 2017;**173**:994-1012
- [6] Bruni R, Celani F. Combining global and local strategies to optimize parameters in magnetic spacecraft control via attitude feedback. *Journal of Optimization Theory and Applications*. 2019;**181**:997-1014
- [7] Jones DR, Perttunen CD, Stuckman BE. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Application*. 1993;**79**(1):157-181
- [8] Graham D, Lawthrop RC. The synthesis of optimum response: Criteria and standard forms, part 2. *Transactions of the AIEE*. 1953;**72**:273-288
- [9] Wie B. *Space Vehicle Dynamics and Control*. Reston, VA: AIAA; 2008
- [10] Wertz JR, editor. *Spacecraft Attitude Determination and Control*. Norwell, MA: Kluwer Academic; 1978
- [11] Rodriguez-Vazquez AL, Martin-Prats MA, Bernelli-Zazzera F. Full magnetic satellite attitude control using ASRE method. In: 1st IAA Conference on Dynamics and Control of Space Systems. 2012
- [12] Lovera M, Astolfi A. Spacecraft attitude control using magnetic actuators. *Automatica*. 2004;**40**(8): 1405-1414
- [13] Lovera M, Astolfi A. Global magnetic attitude control of inertially pointing spacecraft. *Journal of Guidance, Control, and Dynamics*. 2005;**28**(5):1065-1067
- [14] Ogata K. *Modern Control Engineering*. 4th ed. Upper Saddle River, NJ: Prentice Hall; 2002
- [15] Jones DR. DIRECT global optimization. In: Floudas CA, Pardalos PM, editors. *Encyclopedia of Optimization*. Berlin: Springer; 2009. pp. 725-735
- [16] Lucidi S, Sciandrone M. A derivative-free algorithm for bound constrained optimization. *Computational Optimization and Applications*. 2002;**21**:119-142
- [17] Fasano G, Liuzzi G, Lucidi S, Rinaldi F. A linesearch-based derivative-free approach for nonsmooth constrained optimization. *SIAM Journal on Optimization*. 2014;**24**(3):959-992
- [18] Athans M, Falb PL. *Optimal Control: An Introduction to the Theory and Its Applications*. New York: Dover; 2007