



# Kalman-Filter-Based Unconstrained and Constrained Extremum-Seeking Guidance on $SO(3)$

Alex Walsh\*

*University of Michigan, Ann Arbor, Michigan 48109*

James Richard Forbes<sup>†</sup> and Stephen Alexander Chee<sup>‡</sup>  
*McGill University, Montreal, Quebec H3A 0C3, Canada*

and

John Jens Ryan<sup>§</sup>

*NASA Langley Research Center, Hampton, Virginia 23681*

DOI: 10.2514/1.G002635

**Extremum-seeking guidance endeavors to drive the output of a system to the extremum of an unknown objective function. This paper proposes an extremum-seeking guidance algorithm on  $SO(3)$  for cases with and without inclusion and exclusion zones. The gradient of the unknown objective function is estimated via a Kalman filter so that the extremum of the objective function can be approximated. To satisfy inclusion and exclusion zone constraints, two different constrained Kalman filters are proposed. The first Kalman filter is a gain-projected Kalman filter, and the second is a novel linear matrix inequality based Kalman filter that is able to accommodate a larger class of constraints. The proposed extremum-seeking guidance algorithm is demonstrated using a performance objective that relates a spacecraft's attitude to received power of an unknown radiation source using a patch antenna.**

## I. Introduction

CONTROL algorithms are typically designed to minimize the error between a measured output and a desired output. The desired output of the system can be determined in a variety of different ways, such as by a human pilot, by a lookup table scheduled as a function of time, or by a guidance algorithm. Often guidance algorithms are optimal in some sense, where guidance commands are given by the solution of an optimization problem that takes a performance objective into account. The exact relation between a system's output and the performance objective may be unknown. In such a situation, an extremum-seeking guidance algorithm can be used to provide commands, such as a desired trajectory, to a system to maximize or minimize the unknown performance function. Examples of extremum seeking spans several industries, such as the automotive sector [1–3], the energy sector [4,5], biomedical engineering [6], and aviation [7,8].

Consider extremum-seeking algorithms that attempt to maximize an objective function  $J: D \rightarrow \mathbb{R}$ , where  $J = J(z)$ , and where  $z \in D$  is the performance variable of the plant. There are three main approaches to extremize the objective function  $J$  on the restricted domain  $D \subset \mathbb{R}^{n \times m}$ ,  $n, m \in \mathbb{N}$ , depending on the nature of the constraints. These methods include a Lagrangian approach [9], a barrier function approach [10–12], and an optimization on manifolds approach [13,14]. Each method has its own strengths depending on the nature of the given constraints. In this paper, extremum seeking on the manifold  $SO(3)$  and on a constrained subset of  $SO(3)$  is discussed. In particular, extremum seeking on  $SO(3)$  is enabled by

merging a Lagrangian-based approach and a manifold-based gradient optimization algorithm.

Gradient-based extremum-seeking guidance methods require estimation of the gradient, a process that usually falls in two categories: a parameter estimation approach [3,15,16], and a Kalman filter approach [8]. A Kalman filter is advantageous because it is able to filter measurement noise while providing the gradient estimate and is therefore the approach used in this paper. For constrained subsets of  $SO(3)$ , a constrained gradient optimization method is used, which can be enabled by a constrained filtering method. In this paper, we show how to write attitude inclusion and exclusion zones as linear inequality constraints, which are written as a function of the gradient of the performance function. The linear attitude constraints are constructed at each time step, and a constraint to ensure that the linearization remains valid can then be used. The idea of sequential linearization and then ensuring small step sizes are small is similar to that of [17]. A constrained gradient is then estimated using a gain-projected Kalman filter [18], which requires linear constraints. This filter projects the posteriori gradient estimate onto a constrained set so that the desired trajectory does not violate the inclusion or exclusion zones. Further discussion of filtering with equality an inequality state constraints can be found in [19].

In general, a Kalman filter determines the best state estimate in a minimum mean-square error sense. This is an optimization problem that can be formulated as a semidefinite programming (SDP) problem using linear matrix inequalities (LMIs), which can be solved efficiently using interior-point methods ([20] p. 1). Using an LMI approach to Kalman filtering, state constraints, or set membership can easily be introduced. An LMI-based approach is used for set membership filtering for equality, inequality, and linearized nonlinear constraints in [21,22]. Unlike the gain-projected Kalman filter, the LMI-based filter modifies the Kalman gain so that the state estimate is constrained during the correction step, and thus no extra constraining step is needed to ensure that the state estimates satisfy the constraints. Using [21,22] as inspiration, a novel LMI-based Kalman filter is introduced to estimate the constrained gradient for the extremum-seeking guidance problem. The filter has a different structure, different assumptions on the system's characteristics, and a different derivation from that in [21], and it can handle both the linear inequality and norm constraints placed on the gradient estimates. These attributes make the LMI-based filter well suited for estimating the constrained gradient in extremum-seeking applications.

There are several contributions in this paper that together realize extremum seeking on  $SO(3)$  and on a constrained subset of  $SO(3)$ .

Received 25 November 2016; revision received 12 March 2017; accepted for publication 13 March 2017; published online 8 June 2017. Copyright © 2017 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved. All requests for copying and permission to reprint should be submitted to CCC at [www.copyright.com](http://www.copyright.com); employ the ISSN 0731-5090 (print) or 1533-3884 (online) to initiate your request. See also AIAA Rights and Permissions [www.aiaa.org/randp](http://www.aiaa.org/randp).

\*Ph.D. Candidate, Department of Aerospace Engineering, François Xavier Bagnoud Building, 1320 Beal Avenue; [aexwalsh@umich.edu](mailto:aexwalsh@umich.edu).

<sup>†</sup>Assistant Professor, Department of Mechanical Engineering, Macdonald Engineering Building, 817 Sherbrooke Street West; [james.richard.forbes@mcgill.ca](mailto:james.richard.forbes@mcgill.ca).

<sup>‡</sup>Ph.D. Candidate, Department of Mechanical Engineering, Macdonald Engineering Building, 817 Sherbrooke Street West; [stephen.chee@mail.mcgill.ca](mailto:stephen.chee@mail.mcgill.ca).

<sup>§</sup>Branch Head, Dynamic Systems and Controls; [john.j.ryan@nasa.gov](mailto:john.j.ryan@nasa.gov).

First, we elucidate the relationship between the gradient of a function on  $SO(3)$  and a Taylor series expansion of the same function. The extremum-seeking algorithm hinges on this relationship. Second, we demonstrate how to reformulate inclusion and exclusion zones on  $SO(3)$  as linear inequality constraints that are easily embedded into a constrained optimization problem. To be clear, it is the manner that we reformulate these constraints into a form suitable for constrained optimization that is the contribution, not the fashion that the constraints are initially posed. The initial statement of the attitude constraints is identical to other work, such as in [23,24]. Third, we demonstrate how to estimate the gradient of an unknown constrained performance function using a gain-projected Kalman filter and a novel LMI-based Kalman filter. The estimated gradient is used in a gradient ascent optimization algorithm. Fourth, we demonstrate the proposed extremum-seeking technique on a spacecraft attitude guidance problem. Specifically, the proposed extremum-seeking guidance law is used to determine a desired attitude that maximizes an objective function, and a feedback control law minimizes the error between the actual attitude and the desired attitude of the spacecraft. For purposes of this paper, the spacecraft's attitude is assumed to be known exactly, but similar control techniques can be applied if the attitude were estimated from sun sensors, star trackers, magnetometers, etc.

The first contribution is developed in Sec. II once pertinent notation and concepts from differential geometry are reviewed. Section III gives an overview of gradient and projected gradient ascent optimization on  $SO(3)$ . Unconstrained and constrained extremum-seeking on  $SO(3)$  using gradient and projected gradient ascent are shown in Secs. IV and V, respectively. Application of the developed extremum-seeking guidance algorithms to a spacecraft equipped with a patch antenna and three reaction wheels is in Sec. VI, and closing remarks are in Sec. VII.

## II. Mathematical Preliminaries

This section briefly introduces necessary notation and then reviews important concepts from differential geometry. The tangent space of  $SO(3)$  and the gradient of a function  $f: SO(3) \rightarrow \mathbb{R}$  are explained as well as the relationship between a first-order Taylor series expansion of  $f$  and the gradient of  $f$  that lies in the tangent space of  $SO(3)$ . The explanation of the gradient is necessary because a gradient ascent optimization method is used for the extremum-seeking guidance algorithm.

### A. Notation

A frame of reference  $\mathcal{F}_a$  is defined by a set of three orthonormal dextral basis vectors  $\mathbf{a}^1, \mathbf{a}^2,$  and  $\mathbf{a}^3$  ([25] p. 523). The matrix  $\mathbf{C}_{ba} \in SO(3)$  is a direction cosine matrix (DCM) relating the orientation of  $\mathcal{F}_b$  relative to  $\mathcal{F}_a$ , where

$$SO(3) = \left\{ \mathbf{C} \in \mathbb{R}^{3 \times 3} \mid \mathbf{C}^T \mathbf{C} = \mathbf{1}, \det \mathbf{C} = +1 \right\}$$

where  $\mathbf{1}$  is the identity matrix. The  $i$ th column of the identity matrix is denoted as  $\mathbf{1}_i$ . A principal DCM about the  $i$ th basis vector of  $\mathcal{F}_b$  or  $\mathcal{F}_a$  by an angle  $\alpha$  is denoted by  $\mathbf{C}_{ba} = \mathbf{C}_i(\alpha)$ . The vector  $\mathbf{r}_a$  is resolved in  $\mathcal{F}_a$  and can be resolved in  $\mathcal{F}_b$  using the relation  $\mathbf{r}_b = \mathbf{C}_{ba} \mathbf{r}_a$ . The angular velocity of  $\mathcal{F}_a$  relative to  $\mathcal{F}_b$  resolved in  $\mathcal{F}_c$  is given by  $\boldsymbol{\omega}_c^{ab}$ .

The symmetric projection operator  $\mathcal{P}_s: \mathbb{R}^{n \times n} \rightarrow \mathbb{S}^n$  projects the matrix  $\mathbf{U} \in \mathbb{R}^{n \times n}$  to the set of symmetric matrices,  $\mathbb{S}^n = \{ \mathbf{U} \in \mathbb{R}^{n \times n} \mid \mathbf{U} = \mathbf{U}^T \}$ , where  $\mathcal{P}_s(\mathbf{U}) = (1/2)(\mathbf{U} + \mathbf{U}^T)$ . The antisymmetric projection operator  $\mathcal{P}_a: \mathbb{R}^{n \times n} \rightarrow \mathfrak{so}(n)$  projects a matrix  $\mathbf{U} \in \mathbb{R}^{n \times n}$  to the set of antisymmetric matrices,  $\mathfrak{so}(n) = \{ \mathbf{U} \in \mathbb{R}^{n \times n} \mid \mathbf{U} = -\mathbf{U}^T \}$ , where  $\mathcal{P}_a(\mathbf{U}) = (1/2)(\mathbf{U} - \mathbf{U}^T)$ . The operator  $(\cdot)^\times$  maps  $\mathbb{R}^3 \rightarrow \mathfrak{so}(3)$ . For example, for  $\mathbf{v} = [v_1 \ v_2 \ v_3]^T$ ,  $\mathbf{v}^\times$  is given by

$$\mathbf{v}^\times = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}$$

The operator  $(\cdot)^\vee$  maps  $\mathfrak{so}(3) \rightarrow \mathbb{R}^3$ , such that  $(\mathbf{v}^\times)^\vee = \mathbf{v}$ . For example,

$$\begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}^\vee = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \mathbf{v}$$

The DCM  $\mathbf{C}_{ba}$  can be parameterized by the rotation vector  $\boldsymbol{\phi}^{ba}$ . Specifically,  $\mathbf{C}_{ba} = e^{-\boldsymbol{\phi}^{ba \times}}$ , where  $e^{-\boldsymbol{\phi}^{ba \times}}$  is the matrix exponential given by

$$e^{-\boldsymbol{\phi}^{ba \times}} = \cos(\|\boldsymbol{\phi}^{ba}\|) \mathbf{1} + (1 - \cos(\|\boldsymbol{\phi}^{ba}\|)) \left( \frac{\boldsymbol{\phi}^{ba}}{\|\boldsymbol{\phi}^{ba}\|} \right) \left( \frac{\boldsymbol{\phi}^{ba}}{\|\boldsymbol{\phi}^{ba}\|} \right)^T - \sin(\|\boldsymbol{\phi}^{ba}\|) \left( \frac{\boldsymbol{\phi}^{ba}}{\|\boldsymbol{\phi}^{ba}\|} \right)^\times \quad (1)$$

where  $\boldsymbol{\phi}^{ba} = (\boldsymbol{\phi}^{ba^T} \boldsymbol{\phi}^{ba})^{1/2}$  [26]. Computation of  $\boldsymbol{\phi}^{ba}$  is given by  $\boldsymbol{\phi}^{ba} = -(\ln(\mathbf{C}_{ba}))^\vee$ . For a small rotation, that is when  $\|\boldsymbol{\phi}^{ba}\| \ll 1$ ,  $\mathbf{C}_{ba}$  can be approximated as

$$\mathbf{C}_{ba} \approx \mathbf{1} - \boldsymbol{\phi}^{ba \times} \quad (2)$$

which can be derived by using small-angle approximations in Eq. (1). The derivative of  $e^{t\mathbf{A}}$ ,  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , with respect to  $t$ , is  $de^{t\mathbf{A}}/dt = \mathbf{A}e^{t\mathbf{A}} = e^{t\mathbf{A}}\mathbf{A}$ .

### B. Useful Identities

*Proposition 1:* Let  $\mathbf{M} \in \mathbb{R}^{n \times n}$ , and  $\boldsymbol{\Omega} \in \mathfrak{so}(n)$ , then

$$\text{tr}(\mathbf{M}\boldsymbol{\Omega}) = \text{tr}(\mathcal{P}_a(\mathbf{M})\boldsymbol{\Omega}) \quad (3)$$

*Proof:* Expand the right side using the definition of  $\mathcal{P}_a(\mathbf{M})$  to obtain the left side.

*Proposition 2:* Let  $\mathbf{u}, \mathbf{v} \in \mathbb{R}^3$ , then

$$\text{tr}(-\mathbf{u}^\times \mathbf{v}^\times) = 2\mathbf{u}^T \mathbf{v} \quad (4)$$

*Proof:* Use the identity defined by Eq. (2.56b) in [26] in Eq. (4).

*Proposition 3 (chain rule):* Given a function  $f(\mathbf{X}): \mathbb{R}^{n \times n} \rightarrow \mathbb{R}$  and a function  $\mathbf{X}(y): \mathbb{R} \rightarrow \mathbb{R}^{n \times n}$ , the derivative of  $f(\mathbf{X}(y))$  with respect to  $y$  is given by

$$\frac{df}{dy} = \text{tr} \left[ \left( \frac{df(\mathbf{X})}{d\mathbf{X}} \right) \left( \frac{d\mathbf{X}(y)}{dy} \right) \right] \quad (5)$$

*Proof:* See the Appendix.

*Theorem 1 (Weierstrass):* If a function  $f(x): D \rightarrow \mathbb{R}$  is continuous and all  $x \in D$ , where  $D$  is compact, then an extremum exists on  $D$  [27].

In Sec. III, continuous functions of the form  $f: SO(3) \rightarrow \mathbb{R}$  are considered. The manifold  $SO(3)$  is compact, and as such, the function  $f$  admits an extremum.

### C. Manifolds and Tangent Spaces

The manifold  $SO(3)$  is an embedded submanifold of  $\mathbb{R}^{3 \times 3}$  [28]. The embedding space of  $SO(3)$ , denoted as  $\overline{SO(3)}$ , is the manifold  $\mathbb{R}^{3 \times 3}$ . The tangent space to  $SO(3)$  at  $\mathbf{C} \in SO(3)$  is [28]

$$T_{\mathbf{C}}SO(3) = \{ \boldsymbol{\Omega} \mathbf{C} : \boldsymbol{\Omega} \in \mathfrak{so}(3) \} \quad (6)$$

A manifold whose tangent spaces are equipped with a smoothly varying inner product is called a Riemannian manifold. The manifold  $\mathbb{R}^{3 \times 3}$  has the inner product given by

$$\langle \mathbf{A}, \mathbf{B} \rangle = \text{tr}(\mathbf{A}^T \mathbf{B}) \quad (7)$$

where  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{3 \times 3}$ . The submanifolds and tangent manifolds of  $\mathbb{R}^{3 \times 3}$  inherit this inner product. Thus,  $SO(3)$  becomes a

Riemannian manifold when  $T_C SO(3)$  is equipped with the inner product defined by Eq. (7). The inner product is important because it is used to define the gradient of a function.

The orthogonal complement of  $T_C SO(3)$  is denoted as  $(T_C SO(3))^\perp$ . An element of  $Y \in (T_C SO(3))^\perp$  is defined such that, for all  $X \in T_C SO(3)$ ,  $\langle X, Y \rangle = 0$ . The space  $(T_C SO(3))^\perp$  is given by [28]

$$(T_C SO(3))^\perp = \{S\mathbf{C} : S \in \mathbb{S}^3\} \tag{8}$$

*Proposition 4:* Let  $T_C \overline{SO(3)}$  be the tangent space of  $\overline{SO(3)}$  at  $C$ . An element  $\mathbf{Z} \in T_C \overline{SO(3)}$  can be decomposed into the sum of an element in  $T_C SO(3)$  and in  $(T_C SO(3))^\perp$  by

$$\mathbf{Z} = \mathcal{P}_C(\mathbf{Z}) + \mathcal{P}_C^\perp(\mathbf{Z}) \tag{9}$$

where  $\mathcal{P}_C$  and  $\mathcal{P}_C^\perp$  are projection operators that define the mapping  $\mathcal{P}_C: T_C \overline{SO(3)} \rightarrow T_C SO(3)$  and  $\mathcal{P}_C^\perp: T_C \overline{SO(3)} \rightarrow (T_C SO(3))^\perp$ . These projections are given by [28]

$$\mathcal{P}_C(\mathbf{Z}) = \mathcal{P}_a(\mathbf{Z}\mathbf{C}^T)\mathbf{C} \tag{10}$$

$$\mathcal{P}_C^\perp(\mathbf{Z}) = \mathcal{P}_s(\mathbf{Z}\mathbf{C}^T)\mathbf{C} \tag{11}$$

*Proof:* Comparing Eqs. (10) and (11) to Eqs. (6) and (8) shows that  $\mathcal{P}_C(\mathbf{Z}) \in T_C SO(3)$  and  $\mathcal{P}_C^\perp(\mathbf{Z}) \in (T_C SO(3))^\perp$ . Evaluating Eq. (9) using Eqs. (10) and (11) completes the proof.

**D. Gradients on  $SO(3)$  from Definition**

Let  $f: SO(3) \rightarrow \mathbb{R}$  be a continuous function, where  $SO(3)$  is a Riemannian manifold equipped with the inner product given by Eq. (7). The gradient of  $f$  at  $C$ , denoted by  $\nabla f(C)$ , is defined as the unique element of  $T_C SO(3)$  that, for all  $\Xi \in T_C SO(3)$ , satisfies

$$\langle \nabla f(C), \Xi \rangle = \left. \frac{d(f(\Gamma(\epsilon)))}{d\epsilon} \right|_{\epsilon=0} \tag{12}$$

where  $\Gamma(\epsilon) \in SO(3)$ ,  $\Gamma(0) = C$ , and where  $\Gamma'(0) = [d\Gamma(\epsilon)/d\epsilon]_{\epsilon=0} = \Xi$  [28]. An example of a function  $\Gamma(\epsilon)$  that satisfies this criteria is

$$\Gamma(\epsilon) = e^{-\epsilon\mathbf{g}^\times}\mathbf{C} \tag{13}$$

$$\Gamma'(0) = -\mathbf{g}^\times\mathbf{C} \tag{14}$$

where  $\mathbf{g}$  is any element of  $\mathbb{R}^3$ .

*Example 1:* Consider the function  $f(C) = \text{tr}(\mathbf{B}^T C)$ , which maps  $SO(3)$  to  $\mathbb{R}$ , where  $\mathbf{B} \in \mathbb{R}^{3 \times 3}$ . Evaluating Eq. (12) using Eqs. (13) and (14) yields

$$\begin{aligned} \langle \nabla f(C), -\mathbf{g}^\times\mathbf{C} \rangle &= \left. \frac{d}{d\epsilon} \text{tr}(\mathbf{B}^T e^{-\epsilon\mathbf{g}^\times}\mathbf{C}) \right|_{\epsilon=0} \\ \text{tr}[-(\nabla f(C))^T \mathbf{g}^\times\mathbf{C}] &= \text{tr}(-\mathbf{B}^T \mathbf{g}^\times\mathbf{C}) \end{aligned}$$

Using identity (3) yields

$$\text{tr}[-C(\nabla f(C))^T \mathbf{g}^\times] = \text{tr}(-\mathbf{g}^\times \mathcal{P}_a(\mathbf{C}\mathbf{B}^T))$$

Because  $\nabla f(C) = \mathbf{\Omega}C$  for some  $\mathbf{\Omega} \in \mathfrak{so}(3)$ ,  $C(\nabla f(C))^T = -C\mathbf{\Omega}^T = -\mathbf{\Omega}$  is antisymmetric. Thus, using identity (4) yields

$$\begin{aligned} 2\mathbf{g}^T[C(\nabla f(C))^T]^\vee &= 2\mathbf{g}^T \mathcal{P}_a(\mathbf{C}\mathbf{B}^T)^\vee \\ \mathbf{g}^T[C(\nabla f(C))^T - \mathcal{P}_a(\mathbf{C}\mathbf{B}^T)]^\vee &= 0 \end{aligned} \tag{15}$$

Because Eq. (15) must hold for all  $\mathbf{g}$ , the gradient is given by

$$\nabla f(C) = -\mathcal{P}_a(\mathbf{C}\mathbf{B}^T)\mathbf{C} \tag{16}$$

**E. Gradients on  $SO(3)$  Using Projection**

A different method to derive a gradient on  $SO(3)$  can be found using the projections discussed in Sec. II.C. Let  $\bar{f}$  be defined on  $\mathbb{R}^{3 \times 3} \rightarrow \mathbb{R}$ , and let  $f$  be the restriction of  $\bar{f}$  defined on  $SO(3) \rightarrow \mathbb{R}$ . The gradient of  $f$  at  $C$  is denoted as  $\nabla f(C)$ , and the gradient of  $\bar{f}$  at  $C$  is denoted as  $\nabla \bar{f}(C)$ . The gradient  $\nabla \bar{f}(C)$  is given by [29]

$$\nabla \bar{f}(C) = \left( \frac{d\bar{f}}{dC} \right)^T \tag{17}$$

The gradient  $\nabla f(C)$  can be expressed in terms of  $\nabla \bar{f}(C)$  using Eq. (10), by projecting  $\nabla \bar{f}(C)$  onto  $T_C SO(3)$ , that is [28]

$$\nabla f(C) = \mathcal{P}_C(\nabla \bar{f}(C)) \tag{18}$$

*Example 2:* Let  $\bar{f}(C) = \text{tr}(\mathbf{B}^T C)$ ,  $\mathbf{B}, C \in \mathbb{R}^{3 \times 3}$ , and let  $f$  be the restriction of  $\bar{f}$  such that  $C \in SO(3)$ . The gradient  $\nabla \bar{f}(C)$  is given by [30]

$$\nabla \bar{f}(C) = \left( \frac{d}{dC} \text{tr}(\mathbf{B}^T C) \right)^T = \mathbf{B}$$

Using Eqs. (10) and (18), the gradient  $\nabla f(C)$  is given by

$$\nabla f(C) = \mathcal{P}_a(\mathbf{B}\mathbf{C}^T)\mathbf{C} = -\mathcal{P}_a(\mathbf{C}\mathbf{B}^T)\mathbf{C} \tag{19}$$

which is the same result as in Eq. (16).

**F. Gradient from Taylor Series**

Although the previous two sections have provided an analytic method to determine the gradient of a function on  $SO(3)$ , a numerical method is needed in this paper for implementation purposes. In this section, we show the relationship between a Taylor series expansion of  $f(C): SO(3) \rightarrow \mathbb{R}$  and the gradient  $\nabla f(C)$ . This relationship is critical for the extremum-seeking guidance algorithm presented in Sec. IV.

Let  $C = e^{-\phi^\times}\hat{C}$  be a parameterization of  $C$ , where  $\hat{C} \in SO(3)$  is constant, and where  $\phi = [\phi_1 \ \phi_2 \ \phi_3]^T \in \mathbb{R}^3$  represents a rotation from the nominal  $\hat{C}$ . The first-order Taylor series expansion of a function  $f(C(\phi))$  at  $\hat{C}$ , that is around the point  $\phi = \mathbf{0} + \delta\phi$ , is given by

$$f(C(\phi)) \approx f(C(\mathbf{0})) + \mathbf{b}^T \delta\phi \tag{20}$$

where  $\mathbf{b} = [b_1 \ b_2 \ b_3]^T$  is the gradient of Eq. (20), and where

$$\mathbf{b}^T = \left. \frac{d}{d\phi} f(C(\phi)) \right|_{\phi=\mathbf{0}} \tag{21}$$

Because  $\mathbf{b} \in \mathbb{R}^3$  is the gradient of  $f(C(\phi)): \mathbb{R}^3 \rightarrow \mathbb{R}$  at  $\phi = \mathbf{0}$ , and  $\nabla f(\hat{C}) \in T_{\hat{C}} SO(3)$  is the gradient of  $f(C): SO(3) \rightarrow \mathbb{R}$  at  $\hat{C}$ , the matrices  $\mathbf{b}$  and  $\nabla f(\hat{C})$  are different. However, for brevity, we refer to both  $\mathbf{b}$  and  $\nabla f(\hat{C})$  as the gradient. We feel that this practice is acceptable in this paper because  $\mathbf{b}$  can uniquely identify  $\nabla f(\hat{C})$  and vice versa, via the relation given in Proposition 5.

*Proposition 5:* The gradient  $\mathbf{b}$  and the gradient  $\nabla f(\hat{C})$  are related via

$$\nabla f(\hat{C}) = -\frac{1}{2}\mathbf{b}^\times\hat{C} \tag{22}$$

*Proof:* Because the parameterization of  $C = e^{-\phi^\times}\hat{C}$  results in  $C \in SO(3)$  for all  $\phi \in \mathbb{R}^3$ , we are able to relax  $f$  to  $\bar{f}$ . Using Eqs. (17) and (21), and the chain rule given by Eq. (5), each element of  $\mathbf{b}$  is given by

$$\begin{aligned}
 b_i &= \left. \frac{d}{d\phi_i} f(\mathbf{C}(\boldsymbol{\phi})) \right|_{\phi=0} = \text{tr} \left[ \left( \frac{d\bar{f}(\mathbf{C})}{d\mathbf{C}} \right) \left( \frac{d\mathbf{C}}{d\phi_i} \right) \right]_{\phi=0} \\
 &= \text{tr} \left[ (\nabla \bar{f}(\mathbf{C}))^T \left( \frac{d}{d\phi_i} \exp \left( - \sum_{j=1}^3 \phi_j \mathbf{1}_j^\times \right) \hat{\mathbf{C}} \right) \right]_{\phi=0} \\
 &= \text{tr} \left[ (\nabla \bar{f}(\mathbf{C}))^T \left( -\mathbf{1}_i^\times \exp \left( - \sum_{j=1}^3 \phi_j \mathbf{1}_j^\times \right) \hat{\mathbf{C}} \right) \right]_{\phi=0} \\
 &= \text{tr} [ -(\nabla \bar{f}(\hat{\mathbf{C}}))^T \mathbf{1}_i^\times \hat{\mathbf{C}} ], \\
 &= \text{tr} [ -\hat{\mathbf{C}} (\nabla \bar{f}(\hat{\mathbf{C}}))^T \mathbf{1}_i^\times ]
 \end{aligned} \tag{23}$$

where  $\nabla \bar{f}(\hat{\mathbf{C}})$  is the gradient of  $\bar{f}$  at  $\hat{\mathbf{C}}$ . Using Eqs. (3) and (4), Eq. (23) simplifies to

$$b_i = 2[\mathcal{P}_a(\hat{\mathbf{C}}(\nabla \bar{f}(\hat{\mathbf{C}}))^T)]^v \mathbf{1}_i$$

and thus

$$\begin{aligned}
 \mathbf{b} &= 2\mathcal{P}_a(\hat{\mathbf{C}}(\nabla \bar{f}(\hat{\mathbf{C}}))^T)^v \\
 -\frac{1}{2}\mathbf{b}^\times &= \mathcal{P}_a(\nabla \bar{f}(\hat{\mathbf{C}})\hat{\mathbf{C}}^T)
 \end{aligned} \tag{24}$$

Right multiplying Eq. (24) by  $\hat{\mathbf{C}}$  and substituting in Eq. (18) yields Eq. (22).

### III. Maximization of a Function $J: SO(3) \rightarrow \mathbb{R}$

In this section, we review the gradient ascent and constrained gradient ascent methods to solve the problem

$$\max_{C_{ba} \in SO(3)} J = J(C_{ba}) \tag{25}$$

$$s.t. \mathbf{x}_b^{iT} C_{ba} \mathbf{y}_a^i \geq \cos(\alpha_i), \quad i = 1, \dots, n_c \tag{26}$$

where  $J: SO(3) \rightarrow \mathbb{R}$  is continuous,  $\alpha_i \in \mathbb{R}$ , and  $n_c$  is the number of constraints. Because the manifold  $SO(3)$  is compact [28], and because  $J$  is continuous,  $J$  has an extremum in  $SO(3)$  by Theorem 1. Equation (26) can be used to define both inclusion zones and exclusion zones, where  $\mathbf{x}_b^i$  and  $C_{ba}\mathbf{y}_a^i$  must have a separation angle no larger than  $\alpha_i$ . To conserve the “ $\geq$ ” sign, the exclusion zone between  $\mathbf{x}_b^i$  and  $C_{ba}\mathbf{y}_a^i$  with angle  $\alpha_i$  is written as

$$\mathbf{x}_b^{iT} C_{ba} (-\mathbf{y}_a^i) \geq \cos(\pi - \alpha_i) \tag{27}$$

In Sec. III.A, the algorithm to solve the unconstrained problem [Eq. (25)] is considered. In Sec. III.B, the algorithm to solve the constrained problem defined by Eqs. (25) and (26) is shown.

#### A. Gradient Ascent

A retraction on a manifold is a mapping from the tangent space to the manifold, that is  $T_C SO(3) \rightarrow SO(3)$ . The retraction of  $\nabla f(\mathbf{C}) = \boldsymbol{\Omega}\mathbf{C}$  is given by a Riemannian exponential mapping on  $SO(3)$  and is defined as  $R_C(\boldsymbol{\Omega}\mathbf{C}) = e^{\boldsymbol{\Omega}\mathbf{C}}$  [28].

*Example 3:* Consider the gradient from examples 1 and 2. The retraction of  $\nabla f(\mathbf{C}) = -\mathcal{P}_a(\mathbf{C}\mathbf{B}^T)\mathbf{C}$  is given by

$$R_C(\nabla f(\mathbf{C})) = \exp[-\mathcal{P}_a(\mathbf{C}\mathbf{B}^T)]\mathbf{C} \tag{28}$$

and Eq. (1) can be used to evaluate the exponential in Eq. (28), with  $\boldsymbol{\phi} = \mathcal{P}_a(\mathbf{C}\mathbf{B}^T)^v$ . ■

Let the optimizer of  $J = J(C_{ba})$  be  $C_{b^*a}$ . The frame  $\mathcal{F}_{b_k}$  is a frame that is defined at the  $k$ th step of the gradient ascent algorithm. The gradient ascent algorithm determines successive  $C_{b_{k+1}a}$  using the gradient of  $J$  at  $C_{b_ka}$ , given by  $\nabla J(C_{b_ka})$ . The matrix  $C_{b_{k+1}a}$  is given by

$$C_{b_{k+1}a} = R_{C_{b_ka}}(2\kappa_k \nabla J(C_{b_ka})) \tag{29}$$

where  $2\kappa_k > 0$  is the step size [28]. When the gradient  $\nabla J(C_{b_ka})$  is determined from a Taylor series such as in Eq. (20), the update law is given by

$$C_{b_{k+1}a} = e^{-d_k^\times} C_{b_ka} \tag{30}$$

where  $d_k = \kappa_k \mathbf{b}_k$ . Equation (30) is obtained by substituting Eq. (22) into Eq. (29), which is the critical step that enables the extremum-seeking algorithm. This gradient ascent algorithm is shown in Fig. 1. The step in the tangent space  $T_{C_{b_ka}} SO(3)$  is  $-\kappa_k \mathbf{b}_k^\times C_{b_ka}$ , depicted by the dark arrow (or blue arrow online). The retraction from  $T_{C_{b_ka}} SO(3)$  to  $SO(3)$  is represented by the light arrow (red arrow online). The shading represents values of  $J$ , with the maximum value at the star (red shading online), and the minimum values at the shading of the edges of the sphere (blue shading online).

If the objective function is known and can be evaluated,  $\kappa_k$  can be chosen via an exact line search to ensure that there is a sufficient decay in  $J$ . However, the exact line search can be computationally expensive, and thus the line search can be approximated via Armijo’s rule to provide a maximum step size [31]. Wolfe’s conditions also provide a provision for sufficient decrease by providing a lower bound on the step size [31]. In extremum seeking, the objective function cannot be evaluated, and neither of these conditions can be used. Instead, the step size  $\kappa_k$  can be chosen to satisfy

$$d_{\min} \leq \kappa_k \|\mathbf{b}_k\|_2 \leq d_{\max} \tag{31}$$

where  $d_{\min}$  is a minimum step size, and  $d_{\max}$  is a maximum step size. These conditions do not directly ensure a sufficient decrease but allow for quicker convergence of  $J$  when the curvature of  $J$  becomes small.

#### B. Constrained-Gradient Ascent

Consider the optimization problem given by Eqs. (25) and (26). This problem is solved by incorporating constraints to the gradient ascent method presented in Sec. III.A. The updated DCM  $C_{b_{k+1}a}$  must satisfy Eq. (26), and thus Eq. (26) is rewritten as

$$\mathbf{x}_{b_{k+1}}^{iT} C_{b_{k+1}a} \mathbf{y}_a^i \geq \cos(\alpha_i) \tag{32}$$

The constraint defined by Eq. (32) can be transformed to a linear inequality constraint if the step  $d_k$  is small. One method to obtain a small  $d_k$  is by considering the norm constraint

$$d_k^T d_k \leq d_{\max}^2 \tag{33}$$

With a small step  $d_k$ , the small-angle approximation [Eq. (2)] can be used, and Eq. (30) becomes

$$C_{b_{k+1}a} = (\mathbf{1} - d_k^\times) C_{b_ka} \tag{34}$$

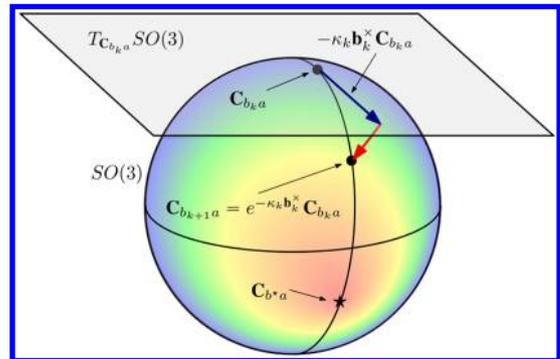


Fig. 1 Gradient ascent for  $J = J(C_{ba})$ , with maximum  $C_{b^*a}$ . The sphere is a visualization of  $SO(3)$ , and the plane is a visualization of the tangent space  $T_{C_{b_ka}} SO(3)$ .

Substituting Eq. (34) into Eq. (32) yields

$$\mathbf{x}_{b_{k+1}}^{iT} (\mathbf{1} - \mathbf{d}_k^*) \mathbf{C}_{b_{k+1}a} \mathbf{y}_a^i \geq \cos(\alpha_i) \quad (35)$$

Simplifying and rearranging Eq. (35) yields

$$\mathbf{A}_{\text{con},k} \mathbf{d}_k \geq \boldsymbol{\xi}_{\text{con},k} \quad (36)$$

where

$$\mathbf{A}_{\text{con},k} = \begin{bmatrix} \mathbf{x}_{b_{k+1}}^{1T} (\mathbf{C}_{b_{k+1}a} \mathbf{y}_a^1)^\times \\ \vdots \\ \mathbf{x}_{b_{k+1}}^{n_c T} (\mathbf{C}_{b_{k+1}a} \mathbf{y}_a^{n_c})^\times \end{bmatrix}, \quad \boldsymbol{\xi}_{\text{con},k} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_{n_c} \end{bmatrix}$$

and where  $\beta_i = \cos(\alpha_i) - \mathbf{x}_{b_{k+1}}^{iT} \mathbf{C}_{b_{k+1}a} \mathbf{y}_a^i$ . The linearization of the constraint is the second contribution. The linearization is possible because we are looking for  $\mathbf{C}_{b_{k+1}a}$  in the neighborhood of  $\mathbf{C}_{b_{k+1}a}$  rather than looking for any  $\mathbf{C}_{b_{k+1}a}$  that satisfies Eq. (32). Other methods to determine attitude based on inclusion and exclusion zones can be found in [23], but this method uses quaternions and quadratic constraints. In [24], admissible sets are used, and this method does not fit within this framework. Equation (36) is a linear constraint, which works well given the nature of the application, which is a projected gradient optimization.

The DCM  $\mathbf{C}_{b_{k+1}a}$  is still given by Eq. (30), but  $\mathbf{d}_k$  is chosen as the minimizer of

$$\min \frac{1}{2} (\mathbf{d}_k - \kappa_k \mathbf{b}_k)^T (\mathbf{d}_k - \kappa_k \mathbf{b}_k) \quad (37)$$

such that Eqs. (33) and (36) are satisfied. Ignoring Eq. (33) and choosing a small  $\kappa_k$ , the solution to Eq. (37) subject to Eq. (36) is given by constructing the matrices of active constraints  $\mathbf{A}_{\text{con},k}^{\text{active}}$  and  $\boldsymbol{\xi}_{\text{con},k}^{\text{active}}$  to yield

$$\mathbf{A}_{\text{con},k}^{\text{active}} \mathbf{d}_k = \boldsymbol{\xi}_{\text{con},k}^{\text{active}} \quad (38)$$

where the minimizing  $\mathbf{d}_k$  is given by

$$\mathbf{d}_k = \kappa_k \mathbf{b}_k - \mathbf{A}_{\text{con},k}^{\text{active}T} (\mathbf{A}_{\text{con},k}^{\text{active}} \mathbf{A}_{\text{con},k}^{\text{active}T})^{-1} (\mathbf{A}_{\text{con},k}^{\text{active}} \kappa_k \mathbf{b}_k - \boldsymbol{\xi}_{\text{con},k}^{\text{active}}) \quad (39)$$

Equation (39) projects  $\mathbf{d}_k$  onto the constrained set [Eq. (36)].

With a small step size  $\kappa_k$ , ignoring Eq. (33) is acceptable if it is either assumed or known that  $\kappa_k \mathbf{b}_k$  is always small. When estimating the constrained gradient in Sec. V, the gain-projected Kalman filter of Sec. V.A requires linear constraints, and norm constraints must be reformulated or ignored. However, the Kalman filter in Sec. V.B that uses LMIs to determine the filter gain is capable of using Eq. (33) directly in the filter formulation.

#### IV. Unconstrained Guidance on $SO(3)$ with an Unknown Objective Function

Consider the system shown in Fig. 2. The attitude of  $\mathcal{F}_b$  relative to  $\mathcal{F}_a$  is parameterized by  $\mathbf{C}_{ba}$ . The function  $J: SO(3) \rightarrow \mathbb{R}$  is a performance function. The outputs of the plant are  $\mathbf{C}_{ba}$  and the measurement of  $J = J(\mathbf{C}_{ba})$ , described in Sec. IV.A. The extremum-seeking guidance law determines a desired frame  $\mathcal{F}_d$  and the DCM  $\mathbf{C}_{da}$ . The attitude error  $\mathbf{C}_{bd} = \mathbf{C}_{ba} \mathbf{C}_{da}^T$  is used for feedback in the controller to drive  $\mathbf{C}_{ba}$  to  $\mathbf{C}_{da}$ . Ideally, the desired attitude  $\mathbf{C}_{da}$

converges to  $\mathbf{C}_{b^*a}$ , where  $\mathbf{C}_{b^*a}$  is an unknown extremum of  $J(\mathbf{C}_{ba})$ . Therefore, because  $\mathbf{C}_{ba}$  converges to  $\mathbf{C}_{da}$  through feedback control,  $\mathbf{C}_{ba}$  converges to  $\mathbf{C}_{b^*a}$ . When the mapping of  $SO(3) \rightarrow \mathbb{R}$  of  $J$  is unknown, the gradient must be estimated to use the gradient ascent algorithm from Sec. III.A.

Using the gradient ascent optimization method and Eq. (30), the desired attitude is given by

$$\mathbf{C}_{d_{k+1}a} = e^{-d_k^*} \mathbf{C}_{b_{k+1}a} \quad (40)$$

where  $d_k$  is the step. The purpose of this section is to explain how to determine  $d_k$  for unconstrained extremum seeking on  $SO(3)$ . Because  $d_k$  is determined from the gradient, the gradient must be estimated. The gradient is estimated using measurements described in Sec. IV.A, and the filter described in Sec. IV.B.

##### A. Performance Measurements

The estimation of the gradient of  $J$  is enabled by taking a first-order Taylor series expansion of  $J$ , using Eq. (20), at  $\mathbf{C}_{ba}(t_k) = \mathbf{C}_{b_{k+1}a}$ . The parameterization  $\mathbf{C}_{ba} = \exp(-\boldsymbol{\phi}^{bb^*}) \mathbf{C}_{b_{k+1}a}$  in the Taylor series expansion yields

$$J(\mathbf{C}_{ba}) \approx J(\mathbf{C}_{b_{k+1}a}) + \mathbf{b}_k^T \boldsymbol{\phi}^{bb^*} \quad (41)$$

where  $\mathbf{b}_k$  is the gradient of  $J$  at  $\mathbf{C}_{b_{k+1}a}$ . The units of  $\mathbf{b}_k$  is typically the unit of  $J$  per radian. When using a first-order Taylor series expansion, it is implied that the Taylor series is a valid approximation of  $J$  in the neighborhood of  $\mathbf{C}_{b_{k+1}a}$ . Therefore, if we knew the exact value of  $\mathbf{b}_k$ ,  $J(\mathbf{C}_{b_{k+1}a})$  could be approximated by evaluating Eq. (41) at  $\mathbf{C}_{ba} = \mathbf{C}_{b_{k+1}a}$  to obtain

$$J(\mathbf{C}_{b_{k+1}a}) \approx J(\mathbf{C}_{b_{k+1}a}) + \mathbf{b}_k^T \boldsymbol{\phi}^{b_{k+1}b_{k+1}}$$

However, in extremum seeking,  $\mathbf{b}_k$  is unknown, and the measurement of the performance function  $J(\mathbf{C}_{b_{k+1}a})$  and  $J(\mathbf{C}_{b_{k+1}a})$  is known. Thus, instead of using the Taylor series to approximate  $J$  in the neighborhood of  $\mathbf{C}_{b_{k+1}a}$ , the Taylor series is used to approximate  $\mathbf{b}_k$  in the neighborhood of  $\mathbf{C}_{b_{k+1}a}$ , using measurements of  $J$  at attitudes  $\mathbf{C}_{b_{k-\ell}a}$ , for  $\ell = 0, \dots, N$ . The linear case is detailed in [8], but for clarity, an example for a function  $J: \mathbb{R} \rightarrow \mathbb{R}$  is shown in Fig. 3. On the left of Fig. 3, the Taylor series at  $x_k$  can be used to approximate  $J(x_{k-1})$ . On the right, multiple measurements of  $x$  and  $J$  can be used to approximate the Taylor series at  $x_k$ . The exact first-order Taylor series cannot be obtained due to the sampling, but if measurements are made close enough together, the difference between the approximated Taylor series and the actual Taylor series is negligible. Returning to  $J: SO(3) \rightarrow \mathbb{R}$ , Eq. (41) evaluated at  $\mathbf{C}_{ba} = \mathbf{C}_{b_{k-\ell}a}$ ,  $\ell = 1, \dots, N$ , and rearranged results in

$$\begin{bmatrix} \Delta J_k \\ \vdots \\ \Delta J_{k-N+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\phi}^{b_k b_{k-1}^T} \\ \vdots \\ \boldsymbol{\phi}^{b_k b_{k-\ell}^T} \end{bmatrix} \mathbf{b}_k \quad (42)$$

where

$$\Delta J_{k-\ell+1} = J(\mathbf{C}_{b_{k+1}a}) - J(\mathbf{C}_{b_{k-\ell}a}), \quad \boldsymbol{\phi}^{b_k b_{k-\ell}^T} = -(\ln(\mathbf{C}_{b_{k+1}a} \mathbf{C}_{b_{k-\ell}a}^T))^\vee$$

Equation (42) is written more compactly as

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{b}_k \quad (43)$$

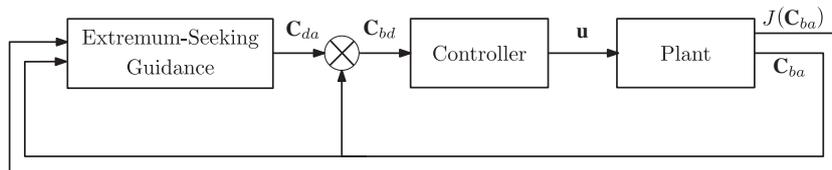


Fig. 2 Extremum-seeking guidance block diagram. The attitude error is given by  $\mathbf{C}_{bd} = \mathbf{C}_{ba} \mathbf{C}_{da}^T$ .

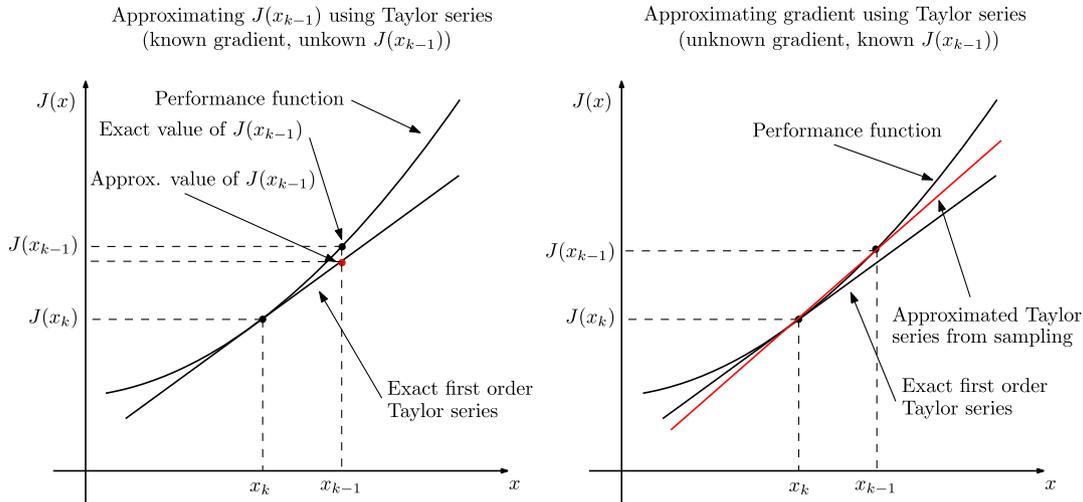


Fig. 3 Sampling of the performance function  $J(x)$ , where  $J: \mathbb{R} \rightarrow \mathbb{R}$ , to approximate either  $J(x_{k-1})$  or  $\mathbf{b}_k$ .

where

$$\mathbf{y}_k = \begin{bmatrix} \Delta J_k \\ \vdots \\ \Delta J_{k-N+1} \end{bmatrix}, \quad \mathbf{H}_k = \begin{bmatrix} \boldsymbol{\phi}^{b_k b_{k-1}^T} \\ \vdots \\ \boldsymbol{\phi}^{b_k b_{k-\ell}^T} \end{bmatrix} \quad (44)$$

Both  $\mathbf{y}_k$  and  $\mathbf{H}_k$  can be determined from the current and previous attitude estimates of the plant.

Determining  $\mathbf{b}_k$  using Eq. (43) is a parameter estimation approach to extremum seeking, where  $\mathbf{b}_k$  contains the parameters, and  $\mathbf{H}_k$  is the regressor. As such, persistence of excitation is required for observability of  $\mathbf{b}_k$  [8,32]. In this paper, a Kalman filter formulation is used to estimate  $\mathbf{b}_k$ . When augmented with measurement noise, Eq. (43) forms the basis of the measurement model for the Kalman filter approach, as explained in Sec. IV.B. The Kalman filter approach also requires persistence of excitation [8].

### B. Kalman Filter Formulation

A linear time-varying Kalman filter is used to estimate  $\mathbf{b}_k$ , by specifying a process model and by modifying Eq. (43) to be the measurement model, where both are given by

$$\mathbf{b}_k = \mathbf{b}_{k-1} + \mathbf{w}_{k-1} \quad (45)$$

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{b}_k + \mathbf{v}_k \quad (46)$$

where  $\mathbf{w}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ ,  $\mathbf{Q}_k \geq 0$ ,  $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ ,  $\mathbf{R}_k > 0$ , and where  $\mathbf{y}_k$  and  $\mathbf{H}_k$  are defined in Eq. (44). The notation  $\mathbf{w} \sim \mathcal{N}(\bar{\mathbf{w}}, \mathbf{Q})$ ,  $\mathbf{w}, \bar{\mathbf{w}} \in \mathbb{R}^n$ ,  $\mathbf{Q} \in \mathbb{R}^{n \times n}$ , indicates that  $\mathbf{w}$  is Gaussian with a mean of  $\bar{\mathbf{w}}$  and covariance  $\mathbf{Q}$ .

The Kalman filter estimates  $\mathbf{b}_k$  using a prediction step

$$\hat{\mathbf{b}}_k^- = \hat{\mathbf{b}}_{k-1}, \quad \hat{\mathbf{y}}_k^- = \mathbf{H}_k \hat{\mathbf{b}}_k^-, \quad \mathbf{P}_k^- = \mathbf{P}_{k-1} + \mathbf{Q}_{k-1}$$

and a correction step

$$\hat{\mathbf{b}}_k = \hat{\mathbf{b}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \hat{\mathbf{y}}_k^-) \quad (47)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T \quad (48)$$

where  $\hat{\mathbf{b}}_k$  and  $\hat{\mathbf{b}}_k^-$  are the predicted and estimated gradient, and  $\mathbf{P}_k^-$  and  $\mathbf{P}_k$  are the predicted and estimated error covariance. Equation (48) is known as the ‘‘Joseph form’’ and is more numerically stable than other forms, while still being valid for any gain  $\mathbf{K}_k$  [26]. This is useful because, in Sec. V.B, Eq. (48) can be used to correct  $\mathbf{P}_k^-$  to obtain  $\mathbf{P}_k$ .

In Sec. V.B, the derived  $\mathbf{K}_k$  is different from the traditional Kalman gain, which is found by minimizing  $\text{tr}(\mathbf{P}_k)$  [26]. The Kalman gain  $\mathbf{K}_k$  is given by

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (49)$$

The step is determined using the estimated gradient,

$$\mathbf{d}_k = \kappa_k \hat{\mathbf{b}}_k + \mathbf{w}_k^{\text{pe}} \quad (50)$$

where  $\mathbf{w}_k^{\text{pe}}$  is persistent excitation required to ensure observability of the performance function [8]. The desired attitude is given by substituting Eq. (50) into Eq. (40).

## V. Constrained Guidance on $SO(3)$ with an Unknown Objective Function

Constrained extremum-seeking guidance is similar to the unconstrained version in that the step  $\mathbf{d}_k$  from Eq. (40) must be determined. Two different constrained extremum-seeking guidance problems are considered in this section. First, the gain-projected Kalman filter solves an extremum-seeking problem to maximize Eq. (25), subject to the inclusion and exclusion zones [Eq. (26)]. The gain-projected Kalman filter estimates the constrained gradient that can be used with the constrained gradient ascent method from Sec. III.B. As the name implies, the gain-projected Kalman filter estimates the gradient and then projects it to a constrained set. Second, the LMI-based Kalman filter solves almost the same problem as the gain-projected Kalman filter. Unlike the gain-projected Kalman filter, which only handles linear equality and inequality constraints, the LMI-based Kalman filter can handle any type of LMI-based constraints. The disadvantage is that an SDP must be solved at each guidance step, but norm constraints can be efficiently handled with this formulation. Therefore, the norm constraint given by Eq. (33) is considered in addition to the inclusion and exclusion zones [Eq. (26)].

### A. Gain-Projected Kalman Filter

The projected gradient method can be incorporated in the Kalman filter by using the gain-projected Kalman filter [18], which is a simple modification to the filter in Sec. IV.B. The gradient is first estimated as in the unconstrained case to obtain  $\hat{\mathbf{b}}_k$ . The constrained gradient estimate  $\tilde{\mathbf{b}}_k$  is then obtained by solving

$$\min \frac{1}{2} (\tilde{\mathbf{b}}_k - \hat{\mathbf{b}}_k)^T (\tilde{\mathbf{b}}_k - \hat{\mathbf{b}}_k)$$

subject to

$$\kappa_k \mathbf{A}_{\text{con},k} \tilde{\mathbf{b}}_k \geq \xi_{\text{con},k} \quad (51)$$

where Eq. (51) is obtained by substituting  $\mathbf{d}_k = \kappa_k \tilde{\mathbf{b}}_k$  in Eq. (36). The solution to this minimization problem is given by

$$\tilde{\mathbf{b}}_k = \hat{\mathbf{b}}_k - \kappa_k^{-1} \mathbf{A}_{\text{con},k}^{\text{active}T} (\mathbf{A}_{\text{con},k}^{\text{active}} \mathbf{A}_{\text{con},k}^{\text{active}T})^{-1} (\mathbf{A}_{\text{con},k}^{\text{active}} \kappa_k \hat{\mathbf{b}}_k - \xi_{\text{con},k}^{\text{active}}) \quad (52)$$

At the next iteration, where  $k$  has been incremented by 1, the new prediction step is given by

$$\hat{\mathbf{b}}_k^- = \tilde{\mathbf{b}}_{k-1}$$

There is no change to the predicted covariance matrix  $\mathbf{P}_k$ , given by Eq. (48). The gain-projected method does not bias the Kalman filter and is simple to implement, which makes this method effective for scenarios with limited computational resources [18]. The desired attitude  $\mathbf{C}_{d_{k+1}a}$  is given by substituting

$$\mathbf{d}_k = \kappa_k \tilde{\mathbf{b}}_k + \mathbf{w}_k^{\text{pe}}$$

into Eq. (40).

### B. Kalman Filter with Linear Matrix Inequality Constraints

The LMI-based Kalman filter does not require a step to constrain  $\hat{\mathbf{b}}_k$  to  $\tilde{\mathbf{b}}_k$ . Rather, the gain  $\mathbf{K}_k$  is determined so that the constraints are imposed directly on the estimate  $\hat{\mathbf{b}}_k$ . The constraints are given by

$$\kappa_k \mathbf{A}_{\text{con},k} \hat{\mathbf{b}}_k \geq \xi_{\text{con},k} - \zeta_k \quad (53)$$

$$\kappa_k^2 \hat{\mathbf{b}}_k^T \hat{\mathbf{b}}_k \leq d_{\text{max}}^2 \quad (54)$$

where Eq. (54) is obtained from Eq. (33) by substituting  $\mathbf{d}_k = \kappa_k \hat{\mathbf{b}}_k$ . The matrix  $\zeta_k \geq \mathbf{0}$  is a slack variable, which transforms Eq. (53) to a soft constraint. This slack variable is critical because, if  $\mathbf{C}_{b_k a}$  is not feasible, a large  $\hat{\mathbf{b}}_k$  might be necessary to satisfy Eq. (53), which could violate Eq. (54). In this situation, without  $\zeta_k$ , it would be impossible to simultaneously satisfy both sets of constraints. Any optimization problem posed with these constraints would be infeasible.

The gain matrix  $\mathbf{K}_k$  is chosen to minimize the objective function

$$\mathcal{J}_k(\mathbf{K}_k, \zeta_k) = \frac{1}{2} [(\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{b}}_k)^T \mathbf{R}_k^{-1} (\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{b}}_k) + (\hat{\mathbf{b}}_k - \hat{\mathbf{b}}_k^-)^T (\mathbf{P}_k^-)^{-1} (\hat{\mathbf{b}}_k - \hat{\mathbf{b}}_k^-)] + s_k^T \zeta_k \quad (55)$$

which is inspired by the recursive least-squares [33] and maximum-likelihood approach to deriving the Kalman filter [34]. This objective function weighs the term  $\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{b}}_k$  with respect to the measurement covariance  $\mathbf{R}_k$  and weighs the residual between the prior and posterior estimates with respect to the prior covariance  $\mathbf{P}_k^-$ . The matrix  $s_k \geq \mathbf{0}$  is a weighting column matrix that penalizes a nonzero slack variable  $\zeta_k$ . The matrix  $s_k$  is chosen to be sufficiently large to render  $\mathbf{C}_{b_k a}$  feasible by driving  $\zeta_k$  to zero. When the constraints are infeasible, the term  $s_k^T \zeta_k$  is the dominant term of  $\mathcal{J}_k$ , and thus the primary objective is to satisfy Eq. (53) with a zero slack variable. Once this occurs, the first term of  $\mathcal{J}_k$  is the dominant term, and normal Kalman filtering behavior ensues.

Using the correction of the state estimate  $\hat{\mathbf{b}}_k = \hat{\mathbf{b}}_k^- + \mathbf{K}_k \mathbf{r}_k$  from Eq. (47), where  $\mathbf{r}_k = \mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{b}}_k^-$  is the innovation term, the objective function can be rewritten as

$$\begin{aligned} \mathcal{J}_k(\mathbf{K}_k, \zeta_k) &= \frac{1}{2} [(\mathbf{y}_k - \mathbf{H}_k (\hat{\mathbf{b}}_k^- + \mathbf{K}_k \mathbf{r}_k))^T \mathbf{R}_k^{-1} (\mathbf{y}_k - \mathbf{H}_k (\hat{\mathbf{b}}_k^- + \mathbf{K}_k \mathbf{r}_k)) \\ &\quad + ((\hat{\mathbf{b}}_k^- + \mathbf{K}_k \mathbf{r}_k) - \hat{\mathbf{b}}_k^-)^T (\mathbf{P}_k^-)^{-1} ((\hat{\mathbf{b}}_k^- + \mathbf{K}_k \mathbf{r}_k) - \hat{\mathbf{b}}_k^-)] + s_k^T \zeta_k \\ &= \frac{1}{2} [(\mathbf{r}_k - \mathbf{H}_k \mathbf{K}_k \mathbf{r}_k)^T \mathbf{R}_k^{-1} (\mathbf{r}_k - \mathbf{H}_k \mathbf{K}_k \mathbf{r}_k) \\ &\quad + (\mathbf{K}_k \mathbf{r}_k)^T (\mathbf{P}_k^-)^{-1} (\mathbf{K}_k \mathbf{r}_k)] + s_k^T \zeta_k \\ &= \frac{1}{2} \mathbf{r}_k^T [(\mathbf{1} - \mathbf{H}_k \mathbf{K}_k)^T \mathbf{R}_k^{-1} (\mathbf{1} - \mathbf{H}_k \mathbf{K}_k) + \mathbf{K}_k^T (\mathbf{P}_k^-)^{-1} \mathbf{K}_k] \mathbf{r}_k \\ &\quad + s_k^T \zeta_k \end{aligned} \quad (56)$$

Note that, without the norm constraint,  $\zeta_k$  is ignored, and the Kalman gain given by Eq. (49) can be recovered by differentiating Eq. (55) with respect to  $\mathbf{K}_k$  and solving for  $\mathbf{K}_k$ .

A new matrix variable  $\mathbf{Z}_k$  is introduced to transform Eq. (56) to a linear convex objective function given by [29]

$$\hat{\mathcal{J}}_k(\mathbf{K}_k, \mathbf{Z}_k, \zeta_k) = \frac{1}{2} \mathbf{r}_k^T \mathbf{Z}_k \mathbf{r}_k + s_k^T \zeta_k \quad (57)$$

where  $\mathbf{Z}_k$  is subject to the constraint

$$\begin{aligned} \mathbf{Z}_k &\geq (\mathbf{1} - \mathbf{H}_k \mathbf{K}_k)^T \mathbf{R}_k^{-1} (\mathbf{1} - \mathbf{H}_k \mathbf{K}_k) + \mathbf{K}_k^T (\mathbf{P}_k^-)^{-1} \mathbf{K}_k^T, \\ \mathbf{Z}_k - (\mathbf{1} - \mathbf{H}_k \mathbf{K}_k)^T \mathbf{R}_k^{-1} (\mathbf{1} - \mathbf{H}_k \mathbf{K}_k) - \mathbf{K}_k^T (\mathbf{P}_k^-)^{-1} \mathbf{K}_k^T &\geq \mathbf{0} \end{aligned} \quad (58)$$

Using the Schur complement ([20] pp. 7–8), Eq. (58) can be converted to an LMI in terms of  $\mathbf{Z}_k$  and  $\mathbf{K}_k$ :

$$\begin{bmatrix} \mathbf{Z}_k & (\mathbf{1} - \mathbf{H}_k \mathbf{K}_k)^T & \mathbf{K}_k^T \\ (\mathbf{1} - \mathbf{H}_k \mathbf{K}_k) & \mathbf{R}_k & \mathbf{0} \\ \mathbf{K}_k & \mathbf{0} & \mathbf{P}_k^- \end{bmatrix} \geq \mathbf{0} \quad (59)$$

In addition,  $\hat{\mathbf{b}}_k = \hat{\mathbf{b}}_k^- + \mathbf{K}_k \mathbf{r}_k$  is substituted into Eqs. (51) and (54) to obtain

$$\begin{bmatrix} d_{\text{max}}^2 & \kappa_k (\hat{\mathbf{b}}_k^- + \mathbf{K}_k \mathbf{r}_k)^T \\ \kappa_k (\hat{\mathbf{b}}_k^- + \mathbf{K}_k \mathbf{r}_k) & \mathbf{1} \end{bmatrix} \geq \mathbf{0} \quad (60)$$

$$\kappa_k \mathbf{A}_{\text{con},k} (\hat{\mathbf{b}}_k^- + \mathbf{K}_k \mathbf{r}_k) \geq \xi_{\text{con},k} - \zeta_k \quad (61)$$

The gain  $\mathbf{K}_k$  is found by minimizing Eq. (57) subject to Eqs. (59–61). The correction for  $\mathbf{P}_k^-$  is still given by Eq. (48). Using  $\hat{\mathbf{b}}_k$  from this section,  $\mathbf{d}_k$  is determined using Eq. (50), and the desired attitude  $\mathbf{C}_{d_{k+1}a}$  is once again found using Eq. (40).

## VI. Numerical Example

Three different extremum-seeking guidance algorithms are used for guidance of a spacecraft with a patch antenna. The guidance algorithms attempt to align the antenna with an unknown source, which could be from another spacecraft or from a ground station, to maximize received power, while satisfying inclusion and exclusion zones. For simplicity, in this numerical example, the position of the radiation source relative to the spacecraft does not change over time. The three Kalman-filter-based guidance algorithms are 1) the unconstrained Kalman filter, 2) the gain-projected Kalman filter with inclusion and exclusion zones, and 3) the LMI-based Kalman filter with norm and attitude constraints.

### A. Simulation Parameters

#### 1. Spacecraft with Three Reaction Wheels

Consider a spacecraft with moment of inertia  $\mathbf{J}_b^{\text{bc}}$ , equipped with three orthonormal reaction wheels that have a combined moment of inertia  $\mathbf{J}_b^{\text{vw}}$  and rotation speed  $\dot{\boldsymbol{\gamma}}$ . The control input to the wheels is given by  $\boldsymbol{\eta}$ , a wheel acceleration. Frame  $\mathcal{F}_a$  is inertial,  $\mathcal{F}_b$  is the body frame of the spacecraft, and  $\mathcal{F}_d$  is the desired body frame

representing the desired spacecraft attitude. The DCM  $C_{ba}$  parameterizes the attitude of the spacecraft relative to  $\mathcal{F}_a$  and is assumed to be known deterministically. In practice,  $C_{ba}$  can be estimated using a suite of sensors such as a magnetometer, a sun sensor, a horizon sensor, and others. The spacecraft's dynamics are given by [35,36]

$$J_b^{Bc} \dot{\omega}_b^{ba} + \dot{\omega}_b^{ba \times} (J_b^{Bc} \omega_b^{ba} + J_b^{WV} \dot{\gamma}) + J_b^{WV} \ddot{\gamma} = \mathbf{0} \quad (62)$$

$$\ddot{\gamma} = -\eta \quad (63)$$

$$\dot{C}_{ba} = -\omega_b^{ba \times} C_{ba} \quad (64)$$

In particular, the spacecraft's moment of inertia is  $J_b^{Bc} = \text{diag}(4, 4, 1) \text{ kg} \cdot \text{m}^2$ , and the reaction wheel array's moment of inertia is  $J_b^{WV} = 0.041 \text{ kg} \cdot \text{m}^2$ . The initial conditions of the spacecraft are  $C_{ba}(0) = C_2(60 \text{ deg})C_3(20 \text{ deg})$ ,  $\omega_b^{ba}(0) = \mathbf{0} \text{ rad/s}$ , and  $\dot{\gamma}(0) = \mathbf{0} \text{ rad/s}$ .

2. Patch Antenna and Performance Function

The received power of an antenna, denoted by  $C$ , is given by [37]

$$C = P_{\text{EIRP}} L_s L_a G_r \quad (65)$$

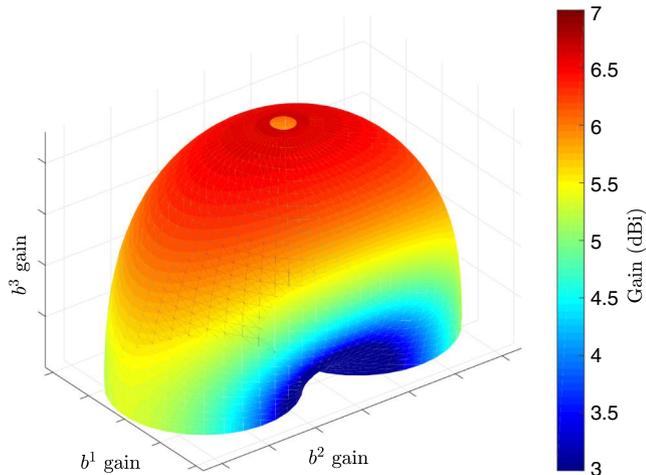


Fig. 4 Radiation pattern of a patch antenna for the  $m = 1$  and  $n = 0$  mode [38]. Gain below 3 dBi not pictured for clarity.

where  $P_{\text{EIRP}}$  is effective isotropic radiated power,  $L_s$  is the space loss, and  $L_a$  is the transmission path loss. For simplicity, let  $P_{\text{EIRP}} L_s L_a = 1 \text{ W}$ , and thus  $C = G_r \cdot 1 \text{ W}$ . Let the patch antenna be mounted on the spacecraft such that  $\mathbf{b}^3$  associated with the body frame is normal to the patch antenna. The radiation pattern of the patch antenna is given by the function  $G_r(\theta, \phi)$ , where  $\theta$  and  $\phi$  can be found from

$$\cos(\theta) = \mathbf{b}_a^{3T} \mathbf{a}_a^3, \quad \cos(\phi) = \mathbf{b}_a^{1T} \mathbf{a}_a^1$$

where  $\mathbf{a}_a^1 = \mathbf{1}_1$ ,  $\mathbf{a}_a^3 = \mathbf{1}_3$ , and  $C_{ba}^T = [\mathbf{b}_a^1 \ \mathbf{b}_a^2 \ \mathbf{b}_a^3]$ . The radiation pattern is shown in Fig. 4 [38]. The objective at hand is to maximize  $J(C_{ba}) = C(C_{ba}) = G_r(C_{ba}) \cdot 1 \text{ W}$ , where  $J$  is all that is available to the guidance algorithm. To be clear, the guidance algorithm uses measurements of received power but does not know the explicit relationship between received power and attitude. The attitude that maximizes the antenna gain is given by  $C_{b^*a}(\theta^*, \phi^*)$ . This attitude corresponds to when  $\mathbf{b}^3$  is aligned with  $\mathbf{a}^3$  and, as such,  $\theta^* = 0 \text{ deg}$  and  $\phi^* \in \mathbb{R}$ . Because the guidance algorithm has no knowledge of where the radiation source is, the guidance algorithm has no knowledge of  $C_{b^*a}$ . The maximum performance is  $J(C_{b^*a}) = 4.61 \text{ W}$ . The actual gradient of  $J(C_{ba})$  is determined numerically in simulation to determine the gradient estimate error. Received power is measured at 10 Hz, and white noise with standard deviation of 0.05 W is added to each measurement of received power.

3. Attitude Constraints

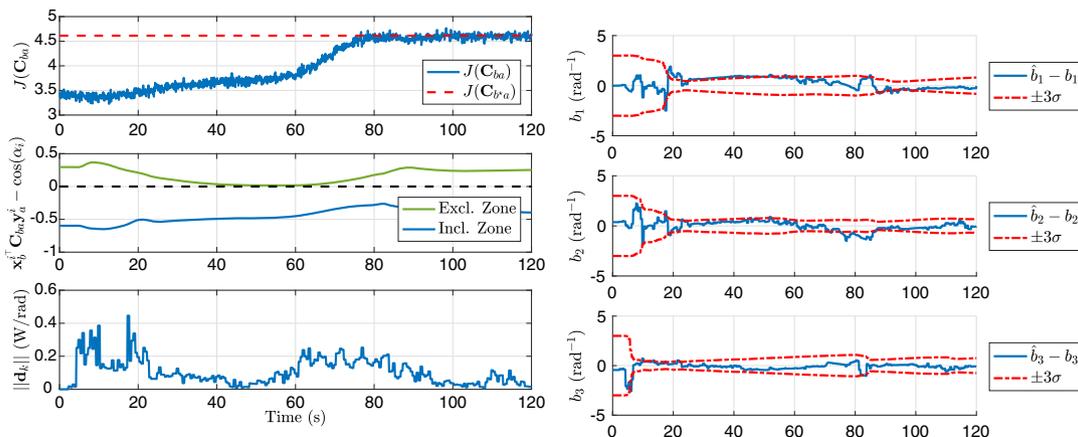
Inclusion zones and exclusion zones often arise constraints on sensors or scientific payloads onboard the spacecraft. For example, a star tracker cannot point toward the sun, an Earth surveillance sensor must always point toward the Earth, and solar cells should point toward the sun. The exclusion and inclusion zones are given by

$$\mathbf{x}_b^1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{y}_a^1 = - \begin{bmatrix} 0.3 \\ 1 \\ -0.5 \end{bmatrix} \cdot \left\| \begin{bmatrix} 0.3 \\ 1 \\ -0.5 \end{bmatrix} \right\|^{-1}, \quad (66)$$

$$\alpha_1 = 180 \text{ deg} - 20 \text{ deg}$$

$$\mathbf{x}_b^2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{y}_a^2 = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}, \quad \alpha_2 = 20 \text{ deg} \quad (67)$$

The constraint associated with  $\mathbf{x}_b^1$ ,  $\mathbf{y}_a^1$ , and  $\alpha_1$  is the exclusion zone.



a) Objective function, constraints, and step size. In middle plot, constraints are satisfied when they are above the black dashed line (i.e., above 0)

b) Gradient estimate error with  $\pm 3\sigma$  bounds

Fig. 5 Extremum-seeking enabled by the unconstrained Kalman filter, with no attitude constraints.

#### 4. Guidance Algorithms

The guidance algorithms run at 2 Hz have  $N = 5$  measurements, with one measurement taken at a frequency of 5 Hz. These values are chosen so that the spacecraft has time to react to a change in desired attitude and so that enough new measurements of  $J$  can be made. The weight  $s_k$  is chosen as  $s_k = 10^3 [1 \ 1 \ 1]^T$ , and the LMIs are solved using the MOSEK solver [39] with the YALMIP interface [40] in Matlab. The persistent excitation  $w_k^{pe}$  is set to  $w_k^{pe} = 0.001 \sin(0.05\pi k) [1 \ 1 \ 1]^T$  rad. As in [8], to ensure smooth commands to the attitude controller, the gradient estimate is filtered with a first-order low-pass filter with a cutoff frequency of 0.08 Hz.

The Kalman filter is used in a similar manner to filters used for parameter estimation. As such, the matrix  $R_k$  can be characterized using the noise of the received power measurements. A single element of  $y_k$  is the difference between two measurements of the objective function. Because the noise added to a measurement of the objective function has a standard deviation of 0.05 W, the standard deviation of noise of the difference of two measurements is  $0.05\sqrt{2}$  W. Using this information and the update rate of the Kalman filter,  $R_k$  is set to  $R_k = 0.0011$  W. It is more difficult to characterize the  $Q_{k-1}$  matrix. Various methods exist to tune  $Q_{k-1}$ , such as a retrospective optimization [41], but trial and error is used in this paper, where  $Q_{k-1} = 0.0021$  W/rad. The initial error covariance estimate is  $P_0 = 1$  W/rad, and the initial gradient estimate is  $\hat{b}_0 = 0$  W/rad.

The Kalman filters use a  $\kappa_k$  of 0.15, and the LMI-based Kalman filter uses the norm constraint [Eq. (54)] with  $d_{\max} = 0.25$  W/rad. Considering the interpretation that the norm of  $d_k$  represents an angle of rotation in radians,  $d_{\max} = 0.25$  W/rad corresponds to a maximum rotation of 14 deg for each iteration of the guidance algorithm, ensuring the small-angle approximation used for the attitude constraints is always valid.

#### 5. Control Algorithm

The desired attitude  $C_{d_k a}$  is fed to the discrete-time attitude controller, given by

$$\eta_k = -J_b^{W-1} (k_d \omega_{b_k a}^{b_k a} - k_p \mathcal{P}_a(C_{b_k d_k})^\vee) \quad (68)$$

where  $C_{b_k d_k} = C_{b_k a} C_{d_k a}^T$  is the attitude error [35]. The attitude controller runs at 10 Hz, and thus  $T = 0.1$  s. The gains of Eq. (68) are  $k_d = 1.21$  N · m and  $k_p = 0.961$  (N · m)/s. The controller is simulated in discrete time, and the dynamics of the spacecraft are simulated in continuous time.

### B. Simulation Results

#### 1. Unconstrained Kalman Filter

The simulation results with the unconstrained Kalman filter are shown in Fig. 5. Figure 5a shows that  $J$  converges to 4.61 W, which corresponds to the maximum possible performance. Near  $J = 4.61$  W, the magnitude of the gradient becomes quite small, and without  $d_{\min}$ , the convergence time would be much longer. The middle plot in Fig. 5a shows the values of the constraints. For the constraints to be satisfied, both lines must be greater than zero. A dotted black line is shown to emphasize the zero line on the plot. Figure 5b shows that the gradient estimate error converges to zero and that the errors mostly remain within the  $\pm 3\sigma$  bounds.

Figure 6 shows a three-dimensional plot of the results from the unconstrained and the gain-projected Kalman filters. The left cone (red cone online) is the exclusion zone, and the right cone (green cone online) is the inclusion zone. The darker trace (red trace online) is the locus of points created by the tip of  $x^1$  from Eq. (66), and the lighter trace (green trace online) is the locus of points created by  $x^2$  from Eq. (67). In the unconstrained formulation shown in Fig. 6a, the lighter line remains clear of the inclusion zone. As it happens, the darker line does not venture into the exclusion zone, but there is no guarantee that this will happen given different initial conditions or noise characteristics. The black vectors form the basis for  $\mathcal{F}_a$ , and the

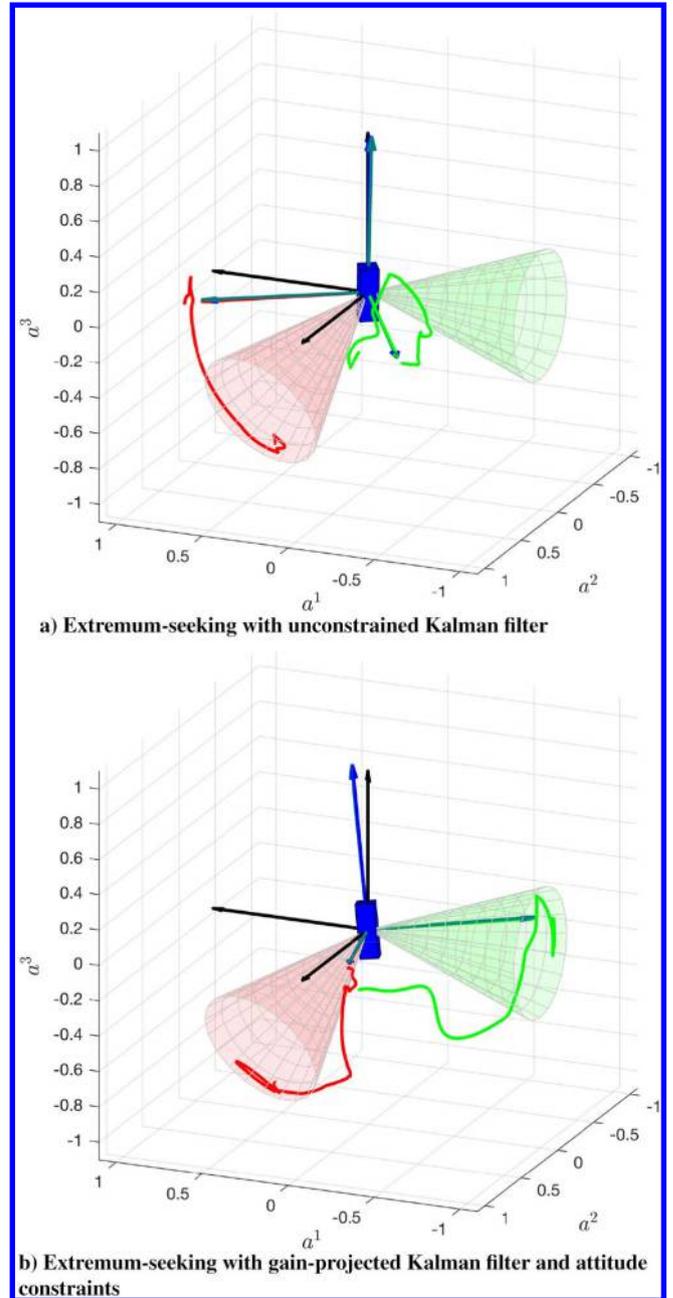
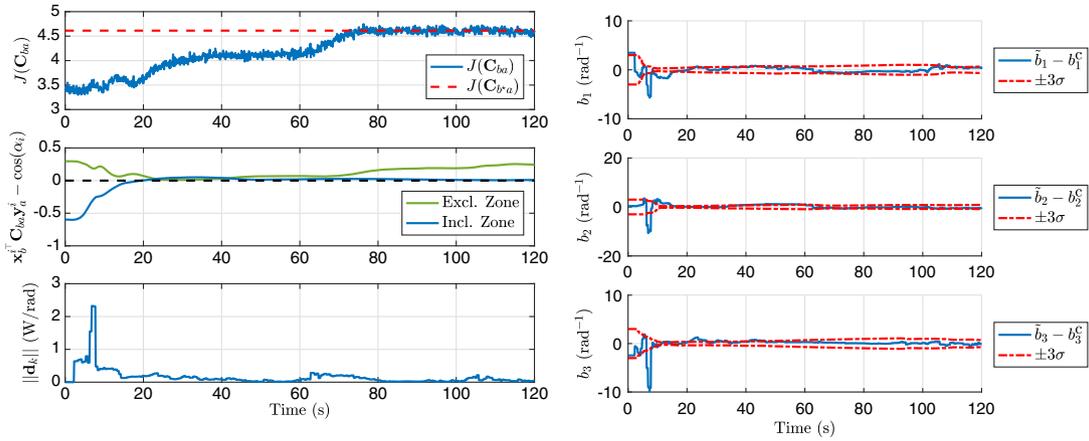


Fig. 6 Three-dimensional representation of extremum-seeking algorithms. The exclusion zone is the left cone (red cone online), and the inclusion zone is the right cone (green cone online).

grey (dark-green/blue online) vectors are the bases for  $\mathcal{F}_d$  and  $\mathcal{F}_b$ , respectively. However, because the controller drives the error  $C_{bd}$  to 1,  $\mathcal{F}_d$  and  $\mathcal{F}_b$  are virtually indistinguishable in this plot.

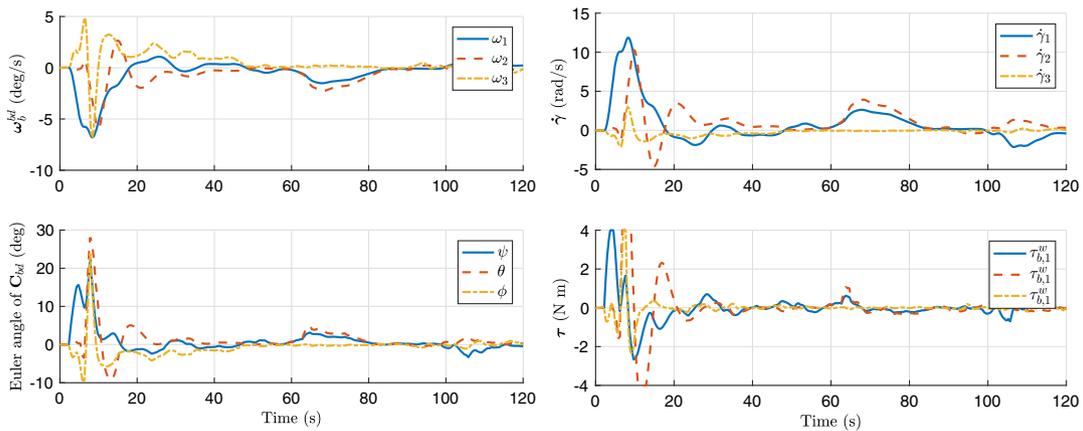
#### 2. Gain-Projected Kalman Filter with Attitude Constraints

The gain-projected Kalman filter considers the inclusion and exclusion zones from Eqs. (66) and (67), and the results of this simulation are shown in Fig. 7. Figure 7a shows that  $J$  converges to  $J(C_{b^* a}) = 4.61$  W, the theoretical maximum of the constrained problem. In addition, the initial reaction of the extremum-seeking guidance algorithm is to ensure that the inclusion zone constraint is satisfied. As a result,  $\hat{b}$  and  $d_k$  become large, and estimation of  $\hat{b}$  is poor, as shown by  $\hat{b}$  escaping the  $\pm 3\sigma$  bounds in Fig. 7b. Once the attitude constraints are satisfied, the gradient estimates improve, and the  $\pm 3\sigma$  bounds are satisfied. Note that the gradient estimate errors are the difference between the constrained gradient estimate  $\hat{b}_k$  and the constrained gradient  $b_k^c$ . The constrained gradient is obtained by



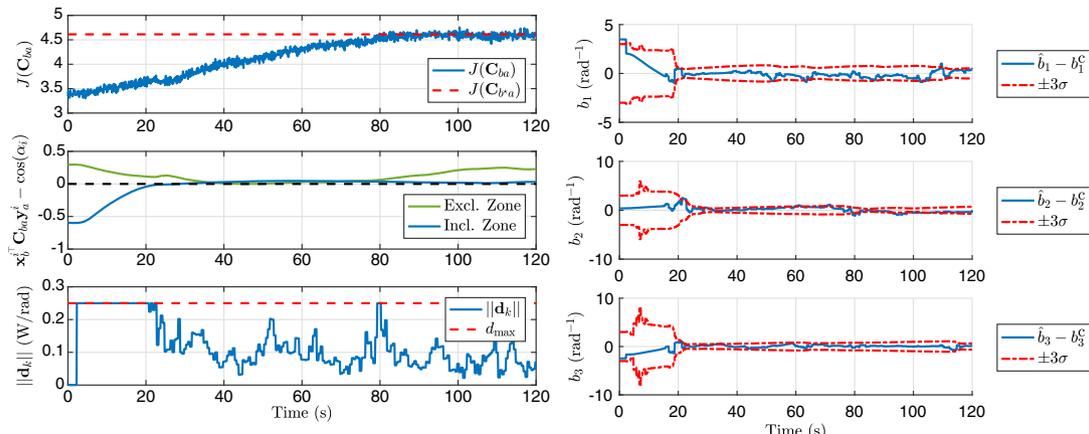
a) Objective function, constraints, and step size. In middle plot, constraints are satisfied when they are above the black dashed line (i.e., above 0) b) Constrained gradient estimate error with  $\pm 3\sigma$  bounds

Fig. 7 Extremum-seeking enabled by the gain-projected Kalman filter, with attitude inclusion and exclusion zones.



a) Attitude and angular velocity errors b) Reaction wheel speeds and control torques

Fig. 8 Spacecraft state errors and control for gain-projected Kalman filter extremum-seeking simulation.



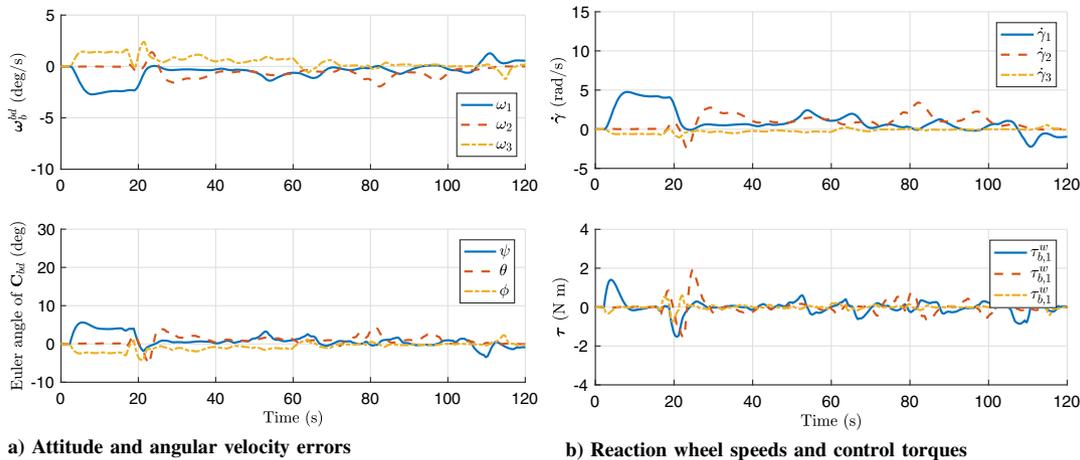
a) Objective function, constraints, and step size. In middle plot, constraints are satisfied when they are above the black dashed line (i.e., above 0) b) Constrained gradient estimate error with  $\pm 3\sigma$  bounds

Fig. 9 Extremum-seeking enabled by the LMI-based Kalman filter, with attitude inclusion and exclusion zones, and with norm constraint on  $d_k$ .

constraining  $b_k$  using the same equation that is used for  $\hat{b}_k$ , that is Eq. (52).

The three-dimensional plot of the gain-projected Kalman filter is shown in Fig. 6b. The light grey line (green line online) begins

outside the inclusion zone cone (green cone online) and initially follows a trajectory perpendicular to the inclusion zone. This part of the trajectory is when the guidance algorithm seeks to satisfy the attitude pointing constraints. The second part of the light grey line is



**Fig. 10** Spacecraft state errors and control for LMI-based Kalman filter extremum-seeking simulation.

when the guidance algorithm seeks to maximize  $J(C_{ba})$ . In this plot,  $\mathbf{b}^3$  is not exactly aligned with  $\mathbf{a}^3$ , but this is expected because received power is not sensitive near the extremum, which is to say the gradient near the optimum is quite shallow.

Figure 8 shows the performance of the attitude control law with the gain-projected Kalman filter simulation. Specifically, Fig. 8a shows that the angular velocity and attitude errors trend to zero, which indicates that  $C_{bd}$  does in fact trend to  $\mathbf{1}$ . Figure 8b shows the reaction wheel rates and control torques, which also go to zero because there are no external torques acting on the spacecraft.

### 3. Linear-Matrix-Inequality-Based Kalman Filter with Norm and Attitude Constraints

The third simulation uses the LMI-based Kalman filter, which considers both the attitude constraints [Eqs. (66) and (67)] and the norm constraint [Eq. (54)]. The results of this simulation are shown in Fig. 9. The objective function converges to its constrained extremum in a slightly longer time than the gain-projected Kalman filter. In addition, the attitude constraints are not satisfied as quickly in this example because  $\mathbf{d}_k$  is constrained. This is not a shortcoming of the LMI-based Kalman filter but rather an advantage. The effect of constraining  $\mathbf{d}_k$  is shown by comparing Figs. 8 and 10. Constraining  $\mathbf{d}_k$  results in a smoother commanded attitude, gradient estimates that remain within the  $\pm 3\sigma$  bounds, and smoother and smaller control effort than with the projected-gain Kalman filter.

## VII. Conclusions

This paper presents several useful results for unconstrained and constrained extremum-seeking guidance on  $SO(3)$ . First, the relationship between the gradient of a function on  $SO(3)$  and the gradient of its Taylor series expansion is given. Second, it is shown how to transform inclusion and exclusion zone attitude constraints to a linear inequality constraint. Third, three different Kalman-filter-based extremum-seeking guidance algorithms are presented to estimate the gradient of an unknown function on  $SO(3)$  to find an extremizing attitude. Each filter has its unique advantages and can be chosen based on the level of complexity of the application. Numerical simulations are presented to demonstrate the effectiveness of the extremum-seeking guidance algorithms.

Several avenues exist to improve the proposed extremum-seeking algorithm. In the present paper, the guidance law may cause the attitude controller to command unrealistic torques. To prevent this, the torques can either be saturated, or the guidance algorithm can be augmented with a model of the spacecraft dynamics. Using the spacecraft dynamics, torque limits can be written as a function of desired attitude using an LMI, which can be incorporated in the LMI-based Kalman filter. The resulting guidance algorithm will generate an attitude trajectory that will nominally keep control torques within acceptable limits. In addition, other optimization algorithms, such as conjugate-gradient methods, can improve the performance of the

extremum-seeking algorithm. Finally, the transformation of inclusion and exclusion zone constraints to linear inequality constraints may prove to be a useful transformation for other areas of research, such as optimal control or path planning.

## Appendix: Proof of Proposition 3

The derivatives of a function  $f(\mathbf{X}): \mathbb{R}^{n \times n} \rightarrow \mathbb{R}$  and a function  $\mathbf{X}(y): \mathbb{R} \rightarrow \mathbb{R}^{n \times n}$  are given by

$$\mathbf{A} = \frac{df}{d\mathbf{X}}, \quad \mathbf{B} = \frac{d\mathbf{X}}{dy} \quad (\text{A1})$$

where the elements of  $\mathbf{A}$  and  $\mathbf{B}$  are given by

$$a_{ij} = \frac{df(\mathbf{X})}{dx_{ji}}, \quad b_{ji} = \frac{dx_{ji}(y)}{dy} \quad (\text{A2})$$

where  $x_{ji}$  are elements of  $\mathbf{X}$  (notice the order of the indices). In addition, the chain rule for the function  $f(\mathbf{X}(y))$  is defined as

$$\frac{df}{dy} = \sum_{i=1}^n \sum_{j=1}^n \frac{df(\mathbf{X})}{dx_{ji}} \frac{dx_{ji}(y)}{dy} \quad (\text{A3})$$

Because the trace of two matrices  $\mathbf{A}$  and  $\mathbf{B}$  is

$$\text{tr}(\mathbf{AB}) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} b_{ji}$$

using Eqs. (A1–A3) yields

$$\begin{aligned} \frac{df}{dy} &= \sum_{i=1}^n \sum_{j=1}^n \frac{df(\mathbf{X})}{dx_{ji}} \frac{dx_{ji}(y)}{dy} \\ &= \sum_{i=1}^n \sum_{j=1}^n a_{ij} b_{ji} \\ &= \text{tr}(\mathbf{AB}) \\ &= \text{tr} \left[ \left( \frac{df(\mathbf{X})}{d\mathbf{X}} \right) \left( \frac{d\mathbf{X}(y)}{dy} \right) \right] \end{aligned}$$

## Acknowledgments

The authors would like to thank support from the NASA Armstrong Research Center for support and funding. In addition, we would like to thank David E. Zlotnik for useful discussions on differential geometry and Frantisek M. Sobolic for discussions on Kalman filtering.

## References

- [1] Popovic, D., Jankovic, M., Magner, S., and Teel, A. R., "Extremum Seeking Methods for Optimization of Variable Cam Timing Engine Operation," *IEEE Transactions on Control Systems Technology*, Vol. 14, No. 3, 2006, pp. 398–407.  
doi:10.1109/TCST.2005.863660
- [2] Drakunov, S., Ozguner, U., Dix, P., and Ashrafi, B., "ABS Control Using Optimum Search via Sliding Modes," *IEEE Transactions on Control Systems Technology*, Vol. 3, No. 1, 1995, pp. 79–85.  
doi:10.1109/87.370698
- [3] Zhang, C., and Ordóñez, R., *Extremum-Seeking Control and Applications: A Numerical Optimization-Based Approach*, Advances in Industrial Control, Springer-Verlag, London, 2012, pp. 121–131.
- [4] Ghaffari, A., Krstić, M., and Seshagiri, S., "Power Optimization and Control in Wind Energy Conversion Systems Using Extremum Seeking," *IEEE Transactions on Control Systems Technology*, Vol. 22, No. 5, 2014, pp. 1684–1695.  
doi:10.1109/TCST.2014.2303112
- [5] Ye, M., and Hu, G., "Distributed Extremum Seeking for Constrained Networked Optimization and Its Application to Energy Consumption Control in Smart Grid," *IEEE Transactions on Control Systems Technology*, Vol. 24, No. 6, 2016, pp. 2048–2058.  
doi:10.1109/TCST.2016.2517574
- [6] Guay, M., Dochain, D., Perrier, M., and Hudon, N., "Flatness-Based Extremum-Seeking Control over Periodic Orbits," *IEEE Transactions on Automatic Control*, Vol. 52, No. 10, 2007, pp. 2005–2012.  
doi:10.1109/TAC.2007.904255
- [7] Ariyur, K. B., and Krstić, M., *Real-Time Optimization by Extremum-Seeking Control*, Wiley, Hoboken, NJ, 2003, pp. 119–141.
- [8] Ryan, J. J., and Speyer, J. L., "Peak-Seeking Control Using Gradient and Hessian Estimates," *Proceedings of the 2010 American Control Conference*, IEEE Publ., Piscataway, NJ, 2010, pp. 611–616.
- [9] Coito, F., Lemos, J. M., and Alves, S. S., "Stochastic Extremum Seeking in the Presence of Constraints," *Proceedings of the 16th IFAC World Congress*, Vol. 38, Elsevier, Amsterdam, The Netherlands, 2005, pp. 276–281.
- [10] DeHaan, D., and Guay, M., "Extremum-Seeking Control of State-Constrained Nonlinear Systems," *Automatica*, Vol. 41, No. 9, 2005, pp. 1567–1574.  
doi:10.1016/j.automatica.2005.03.030
- [11] Tan, Y., Li, Y., and Mareels, I. M. Y., "Extremum Seeking for Constrained Inputs," *IEEE Transactions on Automatic Control*, Vol. 58, No. 9, 2013, pp. 2405–2410.  
doi:10.1109/TAC.2013.2254638
- [12] Guay, M., Moshksar, E., and Dochain, D., "A Constrained Extremum-Seeking Control Approach," *International Journal of Robust and Nonlinear Control*, Vol. 25, No. 16, 2015, pp. 3132–3153.  
doi:10.1002/mc.v25.16
- [13] Taringoo, F., Nešić, D., Tan, Y., and Dower, P. M., "Extremum Seeking Control for Nonlinear Systems on Compact Riemannian Manifolds," *Proceedings of the 53rd IEEE Conference on Decision and Control*, IEEE Publ., Piscataway, NJ, 2014, pp. 2667–2672.
- [14] Dürr, H.-B., Stanković, M., Johansson, K. H., and Ebenbauer, C., "Extremum Seeking on Submanifolds in the Euclidian Space," *Automatica*, Vol. 50, No. 10, 2014, pp. 2591–2596.  
doi:10.1016/j.automatica.2014.08.019
- [15] Guay, M., and Dochain, D., "A Minmax Extremum-Seeking Controller Design Technique," *IEEE Transactions on Automatic Control*, Vol. 59, No. 7, 2014, pp. 1874–1886.  
doi:10.1109/TAC.2014.2322992
- [16] Mills, G., and Krstić, M., "Constrained Extremum Seeking in 1 Dimension," *Proceedings of the 53rd IEEE Conference on Decision and Control*, IEEE Publ., Piscataway, NJ, 2014, pp. 2654–2659.
- [17] Liu, X., and Lu, P., "Solving Nonconvex Optimal Control Problems by Convex Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 3, 2014, pp. 750–765.  
doi:10.2514/1.62110
- [18] Simon, D., and Chia, T. L., "Kalman Filtering with State Equality Constraints," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 38, No. 1, 2002, pp. 128–136.  
doi:10.1109/7.993234
- [19] Simon, D., "Kalman Filtering with State Constraints: A Survey of Linear and Nonlinear Algorithms," *IET Control Theory & Applications*, Vol. 4, No. 8, 2010, pp. 1303–1318.  
doi:10.1049/iet-cta.2009.0032
- [20] Boyd, S. P., *Linear Matrix Inequalities in System and Control Theory*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1994, pp. 1, 7–8.
- [21] Yang, F., and Li, Y., "Set-Membership Filtering with State Constraints," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 45, No. 4, 2009, pp. 1619–1629.  
doi:10.1109/TAES.2009.5310323
- [22] Yang, F., and Li, Y., "Set-Membership Filtering for Discrete-Time Systems with Nonlinear Equality Constraints," *IEEE Transactions on Automatic Control*, Vol. 54, No. 10, 2009, pp. 2480–2486.  
doi:10.1109/TAC.2009.2029403
- [23] Kim, Y., and Mesbahi, M., "Quadratically Constrained Attitude Control via Semidefinite Programming," *IEEE Transactions on Automatic Control*, Vol. 49, No. 5, 2004, pp. 731–735.  
doi:10.1109/TAC.2004.825959
- [24] Weiss, A., Leve, F., Baldwin, M., Forbes, J. R., and Kolmanovsky, I., "Spacecraft Constrained Attitude Control Using Positively Invariant Constraint Admissible Sets on  $SO(3) \times \mathbb{R}^3$ ," *Proceedings of the 2014 American Control Conference*, IEEE Publ., Piscataway, NJ, 2014, pp. 4955–4960.
- [25] Hughes, P. C., *Spacecraft Attitude Dynamics*, Dover, Mineola, NY, 2004, p. 523.
- [26] Markley, F. L., and Crassidis, J. L., *Fundamentals of Spacecraft Attitude Determination and Control*, Springer, New York, 2014, pp. 29, 42–45, 464–465.
- [27] Luenberger, D. G., *Optimization by Vector Space Methods*, Wiley, New York, 1969, p. 39.
- [28] Absil, P.-A., Mahony, R., and Sepulchre, R., *Optimization Algorithms on Matrix Manifolds*, Princeton Univ. Press, Princeton, NJ, 2008, pp. 41–42, 46, 48, 59, 62.
- [29] Boyd, S., and Vandenberghe, L., *Convex Optimization*, Cambridge Univ. Press, Cambridge, England, U.K., 2004, pp. 67, 641.
- [30] Bemstein, D. S., *Matrix Mathematics: Theory, Facts, and Formulas*, 2nd ed., Princeton Univ. Press, Princeton, NJ, 2009, p. 691.
- [31] Nocedal, J., and Wright, S. J., *Numerical Optimization*, 2nd ed., Springer, New York, 2006, p. 33.
- [32] Ioannou, P. A., and Sun, J., *Robust Adaptive Control*, Dover, Mineola, NY, 2012, pp. 177, 220–245.
- [33] Jazwinski, A. H., *Stochastic Processes and Filtering Theory*, Academic Press, New York, 1970, pp. 205–207.
- [34] Rauch, H. E., Tung, F., and Striebel, C. T., "Maximum Likelihood Estimates of Linear Dynamic Systems," *AIAA Journal*, Vol. 3, No. 8, 1965, pp. 1445–1450.  
doi:10.2514/3.3166
- [35] Weiss, A., Kolmanovsky, I., Bernstein, D. S., and Sanyal, A., "Inertia-Free Spacecraft Attitude Control Using Reaction Wheels," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 5, 2013, pp. 1425–1439.  
doi:10.2514/1.58363
- [36] Walsh, A., Zlotnik, D. E., and Forbes, J. R., "Modelling of Spacecraft with N Reaction Wheels Using Arbitrary Attitude Parameterizations," *Advances in the Astronautical Sciences*, Vol. 157, 2016, pp. 945–956; also AAS Paper 16-138, Univelt, Inc., Escondido, CA.
- [37] Wertz, J. R., and Larson, W. J., (eds.), *Space Mission Analysis and Design*, 3rd ed., Microcosm Press, Hawthorne, CA, 1999, p. 554.
- [38] Lee, K. F., and Luk, K. M., *Microstrip Patch Antennas*, Imperial College Press, London, 2011, pp. 61–64.
- [39] Frenk, H., Roos, K., Terlaky, T., Zhang, S., Andersen, E., and Andersen, K., *High Performance Optimization*, Vol. 33, Applied Optimization, Springer, Boston, MA, 2000, pp. 197–232.
- [40] Löfberg, J., "YALMIP: A Toolbox for Modeling and Optimization in MATLAB," *Proceedings of the IEEE International Symposium on Computer Aided Control Systems Design*, IEEE Publ., Piscataway, NJ, 2004, pp. 284–289.
- [41] Sobolic, F. M., and Bernstein, D. S., "Kalman-Filter-Based Time-Varying Parameter Estimation via Retrospective Optimization of the Process Noise Covariance," *Proceedings of the 2016 American Control Conference*, IEEE Publ., Piscataway, NJ, 2016, pp. 4545–4550.