www.DeepakPublishing.com

Sorgenfrei, M., et al. (2013): JoSS, Vol. 2, No. 1, pp. 105-118
(Peer-reviewed Article available at www.jossonline.com)

www.JoSSonline.com

Controller Gain-Tuning for a Small Spacecraft Attitude Tracking Maneuver Using a Genetic Algorithm

Matthew C. Sorgenfrei[1], Sanjay S. Joshi[1], and Amit K. Sanyal[2]

[1] *Department of Mechanical and Aerospace Engineering, University of California, Davis, CA, USA*
[2] *Department of Mechanical and Aerospace Engineering, New Mexico State University, Las Cruces, NM, USA*

Abstract

This article considers the problem of tuning control law gain parameters to optimize the performance of a small spacecraft in low Earth orbit (LEO). The control law under consideration has previously been shown to almost globally stabilize the attitude dynamics of a spacecraft under various operating conditions. In the present work the gain parameters are optimized offline for a small spacecraft performing a tracking maneuver in which the desired attitude is time-varying. The spacecraft is subject to both known actuator saturation limits and multiple external moments. Because of the complexity of both the control law and the problem under consideration, a genetic algorithm is used to optimize the gains of the controller. The genetic algorithm is designed to accommodate for actuator saturation constraints while still generating desired system performance by means of a user-defined fitness function. The controller designs selected by the genetic algorithm are compared to those found by tuning the gains manually using an "informed" trial-and-error search, and it is shown that the genetic algorithm-derived controller solutions generally yield better system performance. Furthermore, a preliminary investigation into the impact of spacecraft parameter variation indicates that the controller design selected by the genetic algorithm is robust to parameter uncertainty. This suggests that automated gain tuning by means of a genetic algorithm could substitute for a human engineer tuning gains in certain applications.

1. Introduction

The problem of tuning controller gain parameters to optimize the performance of a spacecraft in low Earth orbit (LEO) is of practical interest to the aerospace community. This paper focuses on the a priori tuning of gains for a recent small spacecraft science mission. In general, if the system to be controlled is linear or can be linearized about an operating point, then it is possible to apply classical root locus techniques for pole

Corresponding Author: Matthew Sorgenfrei, mcsorgenfrei@ucdavis.edu
Publication History: Submitted – 10/25/12; Revision Accepted – 04/24/13; Published – 08/07/13

placement (e.g., Abdel-Motagly, et al., 2003) or more advanced linear optimal control techniques (e.g., Romano, et al., 2007). If the system is nonlinear, however, the techniques for tuning controller performance become more complicated. Previously-demonstrated approaches include application of physics-based control strategies to a simplified version of the system (Sand, 2009) or nonlinear programming techniques (Crespo, et al., 2010; Fisher, et al., 2007). In this work, we present a new methodology for offline optimization of the gain parameters of a nonlinear control law using a genetic algorithm (GA). Implementation of a GA has advantages over these other strategies in that no reduction or simplification of the underlying equations of motion is required, a large swath of the design space can be rapidly searched, and the algorithm is less likely to settle on local optima (Negnevitsky, 2002; Haupt, 2004).

Previous results have shown that the controller under study is capable of almost globally stabilizing the attitude dynamics of a spacecraft when the control actuators are subject to known saturation limits and the spacecraft is acted upon by environmental moments (Sanyal and Chartuvedi, 2008; Sanyal and Lee-Ho, 2009). With stability established, the GA must determine what gain values result in the best performance of the spacecraft when performing an attitude tracking maneuver. Genetic algorithms have been previously used in the aerospace industry for complex multivariable optimization problems, such as launch vehicle design (Bayley, et al., 2008), actuator placement (Hebrard and Hentrot, 2003), and control of flexible structures (Peng, et al., 2006). In this study we adapt a GA that has been used to optimize the control system of a CubeSat-class spacecraft (Sorgenfrei and Joshi, 2011) for the present problem of tuning a complex control law applied to a small spacecraft. Past applications of GAs to controller design have relied on linearization of the system equations (Krishnakumar and Goldberg, 1992), simplification of the contributing external moments (Qi, et al., 2006), or coupling of a GA with gradient search methods to tune the gains (Seywald, et al., 1995). In contrast, this research utilizes a stand-alone genetic algorithm to optimize the gains of a nonlinear control law applied to a small spacecraft subject to multiple external moments and actuator saturation.

The present work uses as an example the Hawai'iSat-1 spacecraft, a small (75 kg) spacecraft that was designed to operate in LEO. A collaborative effort between the University of Hawai'i and NASA Ames Research Center, Hawai'iSat was tasked with observing certain characteristics of the Earth's oceans by means of a hyperspectral imager. The overall spacecraft design matured as far as the Critical Design Review, and in this work the commercially-available sensors and actuators from that design inform the spacecraft simulation. In particular, the spacecraft was equipped with three magnetic torque rods and one reaction wheel for control, as well as a three-axis gyroscope, a digital magnetometer, and analog sun sensors for attitude determination. This is a very typical suite of sensors and actuators for small spacecraft, and the methodology presented herein is applicable to a wide range of missions. For the purposes of this research, only the attitude control problem will be considered; it is assumed that full state feedback is available.

Using the control algorithm of Sanyal and Chartuvedi (2008), the spacecraft must track a time-varying attitude and maintain a constant angular velocity. This is a typical maneuver for a spacecraft operating in LEO, in which it can be desirable for a body-fixed reference frame to track a local vertical, local horizontal (LVLH) orbit frame, which itself rotates about an inertially fixed Earth-centered frame. By tracking the attitude of the LVLH frame, it is possible for nadir-facing instruments on the spacecraft to continuously observe the surface of the Earth. This was the requirement for the Hawai'iSat mission, in which the hyperspectral imaging payload had to be constantly nadir-pointing. The challenge for the control algorithm is to drive the spacecraft from an arbitrary initial state to tracking the LVLH frame in the presence of actuator saturation constraints and external environmental moments.

In this work, we compare the performance of the genetic algorithm to that of an "informed" trial-and-error approach. The space of candidate design solutions is extremely large; thus, an exhaustive test of every possible design is intractable. As such, in order to achieve desirable performance with the candidate control law,

a control systems engineer would need to apply their intuition, through a trial-and-error search of a portion of the design space. This trial-and-error strategy would likely be guided by performance metrics that are relevant to the needs of the application at hand. In a genetic algorithm, such metrics are captured in a "fitness function." For this research, a fitness function is created that seeks a balance between settling time and the amount of control effort expended. Prior to implementation of the fitness function, a so-called "death-penalty" is applied to any design for which the commanded control torque exceeds the maximum available torque from the control actuators. In order to better understand the efficacy of the genetic algorithm for gain optimization, a comparison is made between the fitness of controller designs found using the algorithm and those found via trial-and-error tuning performed by a control systems engineer.

The remainder of this article is organized as follows. First, the rotational dynamics of a spacecraft in LEO is briefly reviewed and the control law under study is presented in Section 2. The genetic algorithm used to optimize the gains of this control law is presented in Section 3, along with a description of the fitness function and death penalty employed. Simulation results for the gain optimization effort are presented in Section 4, and a discussion of these results is offered in Section 5. Finally, conclusions and recommendations for future work are made in Section 6.

2. Spacecraft Dynamics and Control

2.1 Spacecraft Rotational Dynamics

The problem of controlling the attitude dynamics of a small spacecraft in LEO subject to external moments was addressed in Sanyal and Chartuvedi (2008). Prior to formulating this problem, it is important to first define a set of reference frames that are appropriate for the given application. In this work, we assume the spacecraft is operating in a circular, equatorial orbit. As such, the inertial reference frame N is defined with the $\mathbf{n}_2$ unit vector normal to the orbit plane, the $\mathbf{n}_1$ unit vector pointing towards vernal equinox, and the $\mathbf{n}_3$ unit vector completing the right-handed system. The control task to be performed is for the spacecraft body frame to track the LVLH reference frame, which is defined such that the $\mathbf{l}_2$ unit vector is normal to the orbit plane, the l1 vector points in the spacecraft velocity direction, and the $\mathbf{l}_3$ vector points towards zenith. The LVLH frame rotates about the inertial frame at a rate equal to the spacecraft orbital rate Ω_0, which gives rise to the tracking problem under study. The origin of the body-fixed reference frame B is located at the geometric center of the spacecraft, and its axes are defined such that the B frame and the LVLH frame are aligned when the rotation matrix (to be defined shortly) from the B frame to the LVLH frame equals identity. A visualization of the N={ $\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3$} and LVLH ={ $\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3$} frames, as well as the Hawai'iSat spacecraft and the B frame (with axes $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$), are given in Figure 1.

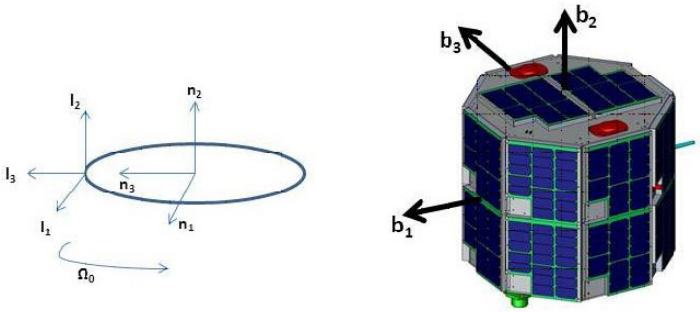

Figure 1. Inertial reference frame N and local vertical, local horizontal LVLH reference frame (left) and body-fixed frame (right) for the problem under study.

As outlined in Wertz and Larson (1978), typical sources of external moments in LEO include aerodynamic drag, gravity gradient effects, solar pressure drag, and residual magnetism. While the control law considered herein can stabilize a system in the most general case in which all of these moments are present (Sanyal and Chartuvedi, 2008), in this work we will consider only the moments contributed by aerodynamic drag and the gravity gradient of the spacecraft. For typical LEO missions, the moment contributed by aerodynamic drag is the largest external contributor by at least an order of magnitude (Wertz and Larson, 1978), and this is also the case for Hawai'iSat. As such, it is a reasonable simplification to only consider these two external moments. In this case, the equations of

motion for the rotation of the spacecraft are given by Euler's equation (Kane, et al., 1983) as

$$J\dot{\omega} + \omega \times J\omega = \tau + M_{ad} + M_g, \quad (1)$$

where **J** is the inertia tensor for the spacecraft in kg m^2, **ω** is the angular velocity (in rad s^{-1}) of the spacecraft defined in the B frame, **τ** is the control torque, M_{ad} is the aerodynamic drag moment, and M_g is the gravity gradient moment, all given in N m. Note that both environmental moments are dependent upon the orientation of the spacecraft with respect to LVLH reference frame (Wertz and Larson, 1978), such that

$$M_{ad} = 0.5\rho C_d A V^2 (\bar{v} \times \overline{S_{CP}}) \quad (2)$$

$$M_g = 3\Omega_0 (-{}^B R^L \hat{l}_3 \times -J\,{}^B R^L \hat{l}_3). \quad (3)$$

In Equation (2), ρ is the atmospheric density at the orbit altitude in kg m^{-3}, C_d is the drag coefficient of the spacecraft, A is the spacecraft area normal to the velocity direction in m^2, and V is the linear velocity of the spacecraft in m s^{-1}. The unit vector $\bar{v}$ is in the velocity direction, while $\overline{S_{CP}}$ is the vector from the center of mass to the center of pressure of the spacecraft (defined in the B frame). The rotation matrix ${}^B R^L$ in (3) defines the rotation from the LVLH frame to the B frame, and the unit vector $-\mathbf{l}_3$ is used because we require the nadir vector direction to define the gravity gradient moment. While the controller to be optimized uses a feed-forward term to eliminate the environmental moments, it is important to note that the presence of these external moments will still result in complex spacecraft motion. Referring back to Equation (1), the moments could impart general three-axis rotation, a behavior which could not be accurately analyzed using a linear approximation of (1). As such, a gain-tuning strategy that can accommodate the full nonlinear motion of the system is desirable.

Following the notation of Sanyal and Chartuvedi (2008) and Sanyal and Lee-Ho (2009), rotation matrices are used to describe the orientation of the spacecraft body frame with respect to an inertial reference frame. Rotation matrices (denoted R) are members of the special orthogonal group SO(3), defined in Bullo and Lewis (2005) as:

$$SO(3) = \{R \in \mathbb{R}^{3\times 3} : R^T R = I = RR^T, \det(R) = 1\}$$

and the evolution of a given rotation matrix R in time is governed by the equation

$$\dot{R} = R\omega^{\times}, \quad (4)$$

where ω^x is the skew-symmetric operator that carries a 3x1 vector into a 3x3 matrix (Bullo and Lewis, 2005). Equations (1) and (4) fully describe the rotational motion of a spacecraft in LEO. We further parameterize this motion by introducing expressions for the error in attitude and angular velocity, denoted by **Q** and $\boldsymbol{\omega}_e$ respectively:

$$Q = R_d^T R \quad (5)$$

$$\omega_e = \omega - Q^T \omega_d, \quad (6)$$

where $\mathbf{R}_d$ and $\boldsymbol{\omega}_d$ are the desired spacecraft attitude and angular velocity, both of which are defined with respect to the inertial frame N. The rotational motion of the spacecraft (Equations (1) and (4)), parameterized by the rotational and angular velocity error (Equations (5) and (6)) serves as the foundation for the development of the control law presented in the next section.

2.2 Candidate Control Law

The control law to asymptotically track the desired attitude and angular velocity of a spacecraft is first proposed in Sanyal and Chartuvedi (2008) as:

$$\tau = -L\omega_e + JQ^T\dot{\omega}_d + (Q^T\omega_d)^{\times} JQ^T\omega_d + trace(K - KQ) \left[k_1 n_1^{\times} Q^T n_1 + k_2 n_2^{\times} Q^T n_2 + k_3 n_3^{\times} Q^T n_3\right] - M_{ad}(R) - M_g(R) \quad (7)$$

While it is beyond the scope of this article to fully replicate the proof of almost global stability for this controller, some interesting properties of Equation (7) are discussed here.

Let **L** and **K** be positive definite control gain ma-

trices, with the restriction that the diagonal elements of **K** must be non-equal (e.g., k1 ≠ k2 ≠ k3). As seen in Equation (7), the gain matrix **L** operates on the angular velocity error ωe and the gain matrix **K** operates on the angular position error **Q**. Note also that the output control law is dependent upon the desired angular velocity $\boldsymbol{\omega}_d$ and angular acceleration $\boldsymbol{\omega}_d$, and that the attitude-dependent aerodynamic drag moment $\mathbf{M}_{ad}$ and the gravity gradient torque $\mathbf{M}_g$ are included directly in the control algorithm. This implies that the attitude determination and control subsystem (ADCS) is capable of resolving the attitude of the spacecraft and feeding that attitude forward to the controller in order to calculate estimated values of the two environmental moments. This is not an unreasonable assumption, given that current small spacecraft microprocessors are capable of generating state estimates at rates of 4 Hz and higher. Based on the proof presented in Sanyal and Chartuvedi (2008), it is known that $(\mathbf{Q}, \boldsymbol{\omega}_e) = (\mathbf{I},0)$ is an asymptotically stable equilibrium of the closed loop system consisting of Equations (1) – (4) and (7), with a domain of convergence that is almost global on the state space of attitude motion. We seek gain combinations to optimize the performance of this controller for tracking a constant angular velocity and a time-varying attitude with respect to the inertial frame.

In this research, we restrict both **K** and **L** to being diagonal positive definite matrices. While the original stability proof requires that **K** be diagonal (Sanyal and Chartuvedi, 2008), no such restriction exists for **L**. However, it is possible to use a diagonal version of **L** without loss of generality by a change of body coordinate frames. For example, let $\mathbf{R}_1$ be a rotation matrix from the body frame B to a body frame B_1. Then, for a general torque vector applied in each body frame, we have the relationship $\boldsymbol{\tau}_1 = \mathbf{R}_1\boldsymbol{\tau}$ and $\mathbf{R}_1\mathbf{L}\,\boldsymbol{\omega e} = \mathbf{R}_1\mathbf{L}\mathbf{R}_1^T(\mathrm{R1}\ \omega 1) = \mathbf{L0}(\mathbf{R1}\ \omega 1)$, where $\mathbf{L0} = \mathbf{R1LR1T}$ is diagonal. Reduction of the gain matrix **L** to a diagonal 3x3 matrix results in a total of six gain terms that need to be optimized for the maneuver under consideration. Note that while the nonlinear controller is "PD-like" in structure, one cannot assume that the algorithm will behave exactly like the linear counterpart, making the task of tuning the six gain terms a formidable one.

3. The Genetic Algorithm

3.1 Algorithm Overview

Genetic algorithms have previously been used for optimization of complex design problems in the aerospace industry, including control law gain tuning for a linear system (Krishnakumar and Goldberg, 1992). As outlined in Negnevitsky (2002), GAs use fundamental concepts from evolutionary biology to "evolve" a population of solutions for a given design problem by assessing the fitness of each candidate solution. Past results (Holland, 1975) have shown that GAs yield better and better design solutions as more generations are evolved. Design solutions are encoded as a string of binary digits, with one such string of bits making up a "chromosome" (a design). Over the life of the algorithm, the population of chromosomes is subjected to the probabilistic genetic operators of crossover and mutation. The probability of crossover and/or mutation occurring is controllable by the user, and a detailed study on the ideal combinations of these attributes is provided in Grefenstette (1986). Past work on gain tuning via GAs has been limited to linear systems (Qi et al., 2006) or control laws with relatively few gain parameters (Omatu and Yoshioka, 1997). In the current research, we consider the general rotation of a small spacecraft as governed by a nonlinear control law with six gain terms total. This results in a large, complex space of designs that is not easily searchable using traditional methods. The basic genetic algorithm used in this work was first presented in Sorgenfrei et al. (2010), where it was applied to linear controller design for nanosatellites. Extensions have since been made to accommodate a broader range of control algorithms used for small spacecraft (Sorgenfrei and Joshi, 2011; Sorgenfrei et al., 2012). The present research expands upon this foundation, using the same algorithm structure to tune the gains of a nonlinear controller applied to a small spacecraft subject to environmental moments and actuator saturation constraints.

A standard approach to solving complex optimization problems such as that considered here would be to apply a convex optimization routine (Robinett et al., 2005). Unfortunately, one downside to implementing

traditional gradient search techniques is their propensity to settle on local optima when applied to non-convex performance indices, a problem that is particularly relevant to the complex design space considered here. While there is not a mathematical proof for the ability of genetic algorithms to yield the globally optimal solution to a particular problem, the probabilistic genetic operators employed have a tendency to drive the GA to avoid such local optima by "pushing" it into different regions of the design space. Instead of using a convex optimization approach, this article compares the performance of the genetic algorithm to an informed trial-and-error strategy, which is described in greater detail in Section 4.

The central element of any GA implementation is the chromosome structure used to encode the members of the design population under study. Referring back to the control law of Equation (7), there are six gain terms to be tuned, three each in the gain matrices **K** and **L**. Based on preliminary testing of the controller in simulation, the GA will test gain values between 0.0001 and 0.1. The magnitude of these gains might seem relatively low, but it was found that gains in this range gave rise to controller solutions less likely to violate the actuator saturation death penalty. The implementation of this penalty will be described further in the next subsection. If six bits are used to encode each gain term (resulting in $2^6 = 64$ increments), a resolution of 0.00016 can be achieved by evenly dividing the gain interval of 0.0001 - 0.1 into 64 steps. Using six bits per gain term results in a gene that is 36 bits long, where each bit (gene) represents a part of the binary representation of a gain value. There are 64 possible gain values for the six different gain terms, resulting in more than 68 *billion* (64^6) design solutions. A visualization of a representative chromosome for the current problem is given in Figure 2.

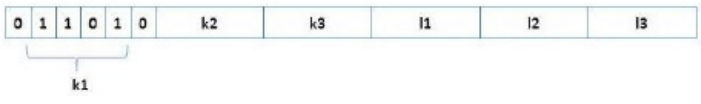

Figure 2. Visualization of the chromosome structure used to optimize the nonlinear controller under study.

At the outset of the algorithm, a random population of designs is created and each one is tested in a spacecraft maneuver simulation environment constructed using MATLAB. After the simulation is complete, the fitness of the design is evaluated using the fitness function. Once all of the chromosomes have been tested, the most-fit solutions are selected for reproduction via application of tournament selection of size two. A new generation of designs is created, using the most-fit designs from the previous generation as parents, and crossover and mutation are applied during the reproduction process. The size of the design population, the number of generations for which the algorithm is run, the crossover probability, and the mutation probability are all selectable by the user. Using Grefenstette (1986) as a guide, a number of different GA trials are undertaken in this work in order to find the best combination of variables for the problem under study.

3.2 Fitness Function Implementation

The fitness function used to assess the performance of each controller design was first introduced in Sorgenfrei and Joshi (2011). This function rewards controller designs that result in the minimum possible combination of settling time (T_s) and the amount of control effort expended (Ce), such that

$$f = -(T_s + \beta Ce)\,. \tag{8}$$

The desire to balance minimal settling time against minimal control effort is a benchmark problem in spacecraft engineering (Wie, 2005); however, for multi-axis rotation problems, the standard definition of settling time cannot be applied. In this work we define settling time in terms of the principal axis of rotational error ζ which can be extracted directly from the attitude error rotation matrix **Q** using Rodrigues' formula (Bullo and Lewis, 2005). A system is considered to have settled when the norm of the attitude error vector ζ falls below a certain reference level $||\zeta|| < ||\zeta_{ref}||$, where ζ_{ref} is determined from an attitude error reference matrix **Qref**. This rotation is defined as the composition of three principal rotations about each axis in an amount of 0.1°, e.g., $Q_{ref} = R_1(0.1°)R_2(0.1°)R_3(0.1°)$.

Note that this compound rotation is equivalent to $||\zeta_{ref}|| = 0.003$. The total control effort expended during a maneuver is calculated as the L_1 norm of the total torque produced by the actuators, with Ce defined as

$$Ce = \sum_i ||\tau_i|| \, \Delta t_i. \quad (9)$$

The *i* subscript in Equation (9) denotes the *i*th time step of the simulation, and the overall value of Ce is given in units of N m s. A non-zero weight factor β is included in Equation (8) to allow for tuning of the relative importance of the two performance metrics, as well as to roughly account for unit-based differences in the magnitude of these metrics. Thus, if a spacecraft designer was more concerned with minimizing the amount of control effort expended, it would be possible to increase β to penalize designs requiring a higher amount of control effort. Regardless of the value of β, the most desirable controller design will be that which results in the largest (least negative) value of *f*. This results from the smallest possible combination of T_s and Ce. Genetic algorithm search results will be presented both for β = 1 and for a value of β that has been selected to more closely balance the magnitude of the two performance metrics.

Prior to assessing the quality of a given design by applying the fitness function, each candidate controller solution must pass through a death penalty. This death penalty is used to ensure that the controller design never requires more than the maximum available control torque that the actuators can produce. For this research, the Hawai'iSat spacecraft model was simplified slightly by only considering the magnetic torque rods as available actuators, and treating the Earth's magnetic field as constant for a given orbit altitude. While it is a well-known fact that the Earth's field varies throughout a standard LEO orbit, inclusion of a detailed magnetic field model is beyond the scope of the current work. The selected torque rods are capable of producing a magnetic moment of 25 A m^2, which is equivalent to approximately 1.2 mN m of torque about each axis in a circular LEO orbit of 600 km. After a controller design is implemented in our MATLAB spacecraft simulator, the death penalty checks to see if a control torque greater than the saturation limit was ever commanded. If an issued control command violates the saturation limit, the death penalty discards that design from the design population. If not, that design is passed to the fitness function for evaluation. In this way, the genetic algorithm is capable of not only ensuring that the actuator saturation limit is not violated, but also selecting a controller that will yield ideal system performance by way of the fitness function. Simulation results using the GA are presented in the next section.

4. Numerical Simulation

4.1 Simulation Description

The gain parameters for the control law given in Equation (7) are tuned to generate the best possible response for a small spacecraft attitude tracking maneuver. Initially, the spacecraft is rotated 1° about an arbitrary principal axis of rotation, and has zero angular velocity in the B frame. This initial state is analogous to that which may be achieved after a spacecraft has completed the "detumble" phase of operations, in which the initial angular velocity of the system is damped out, often using magnetic torquers for actuation (Wertz and Larson, 1978). In this research, the principal axis of the initial rotational error is defined as

$$\hat{\lambda} = \frac{1}{\sqrt{3}}[-1 \quad 1 \quad -1]^T$$

and it is the job of the controller to drive the spacecraft from this initial error state to the desired state. The desired angular velocity of the spacecraft is $\boldsymbol{\omega}_d = [\,0 \quad \Omega_0 \quad 0\,]^T$ (rad/s), which results in a time-varying desired rotation matrix $\mathbf{R}_d$. When this desired state has been achieved, the spacecraft will be holding its alignment with the LVLH frame as that frame orbits the Earth. Gain parameters will be tuned using the fitness function of (8) and the aforementioned death penalty as a guide.

The stability proof presented in Sanyal and Chartuvedi (2008), levies no requirements as to the inertia properties of the spacecraft, and the controller in question has been applied to a wide range of spacecraft sizes (Sanyal and Chartuvedi, 2008; Sanyal and Nordkvist, 2010). In this research, we use the inertia tensor for the

Hawai'iSat spacecraft as modeled by the design team, with $\mathbf{J}$ = diag (5.69 5.69 4.26) kg m^2. This is the inertia tensor for a cylindrically shaped small spacecraft measuring 0.66 m in diameter and 0.77 m tall. Recall that, as stated earlier, the maximum allowable control torque that can be applied about any of the three body axes is 1.2 mN m. Numerical simulation of the spacecraft system is accomplished using a previously-reported Lie group variational integrator (LGVI) (Sanyal and Nordkvist, 2012; Lee, et al., 2005).

Two different strategies are used to determine the best possible *a priori* combination of gains for the control task described above. The first method is to apply the genetic algorithm to the gain tuning problem using the fitness function (8) combined with a death penalty as a guide for desired performance. The second method, referred to as informed trial-and-error, relies upon the intuition of a control systems engineer to tune the elements of the gain matrices based on the system behavior observed in simulation. An engineer undertaking a trial-and-error approach to gain tuning necessarily has access to the same information provided by the fitness function; however, given the large design space, it is unlikely that they would be able to explore an appreciable sample of candidate solutions. In contrast, the randomized crossover and mutation operators within the GA allow for rapid search of a broad swath of the design space.

In the trial-and-error tuning approach a control systems engineer is given three hours to tune the gain variables by hand using the same constraints as described in the preceding sections. Three hours were allotted to hand-tuning because that was approximately the amount of time required for the GA to execute a standard search trial on a desktop PC, and it was desired to create a one-to-one comparison between the methods. For this approach, both gain matrices are initialized to 0.05 $\mathbf{I}_{3x3}$ (essentially the middle of the admissible gain range) and then numerous simulation iterations are undertaken. After each simulation run, gain variables are either increased or decreased depending on the observed time-response of the spacecraft. The proportional gain matrix $\mathbf{K}$ is adjusted first, and then the derivative gain matrix $\mathbf{L}$ is adjusted as appropriate. While the nonlinear controller is "PD-like", it would be incorrect to assume that this controller will demonstrate the same behavior as its linear counterpart. The approach of adjusting the $\mathbf{K}$ matrix first and then the $\mathbf{L}$ matrix is an attempt to tune the proportional response prior to the derivative response, but other strategies may be equally valid, given that there is only a loose analogy to a PD controller. In contrast to this directed gain tuning approach, the GA will randomly change any gain within the controller structure as dictated by the crossover and mutation probability rates.

4.1 Simulation Results

Four different genetic algorithm trials were initially undertaken, the results for which are presented in Tables (1) and (2). For these preliminary tests the weight factor within the fitness function was held at $\beta = 1$ in order to see what the "natural" behavior of the fitness function would be. In these trials the variables of interest are the design population size Pop, the number of generations for which the algorithm was run Num, and the crossover probability Xover. The values of these variables, as seen in Table 1, follow the basic suggestions provided in Grefenstette (1986). For all of these simulations the mutation rate was held constant at 0.001%, also replicating the work of Grefenstette (1986). The outputs that are recorded for these GA trials are the maximum fitness value achieved f_{max}, the settling time in seconds (T_s), the control effort in N m s (Ce), and the gain values that resulted in these performance attributes (given in Table 2).

Table 1. GA performance for four trials with $\beta = 1$

Pop Size	Num	Xover	f_{max}	Ts (s)	Ce (Nms)
40	20	80	-205.4	205.4	0.12
40	40	70	-187.2	187.1	0.12
40	50	70%	-179.5	179.4	0.1
50	40	70%	-176.2	176.1	0.1

Table 2. Gain values corresponding to four GA trials with β = 1

k_1	k_2	k_3	l_1	l_2	l_3
0.0445	0.0397	0.0207	0.0715	0.0905	0.0905
0.0619	0.0255	0.0175	0.0937	0.0921	0.0715
0.0651	0.0112	0.0207	0.0905	0.0984	0.0937
0.0604	0.0017	0.0302	0.0968	0.0968	0.0984

As can be seen in Table 1, the best-performing controller solution was achieved using a population of 50 designs evolved over 40 generations. The maximum fitness score is f = -176.2, corresponding to T_s = 176.1 s and Ce = 0.1 N m s. The gain values that resulted in this performance are **K** = diag (0.0604 0.0017 0.0302) and **L** = diag (0.0968 0.0968 0.0984). A component-wise time history of the spacecraft angular velocity and the commanded control torque for this best-performing controller design are presented in Figure 3.

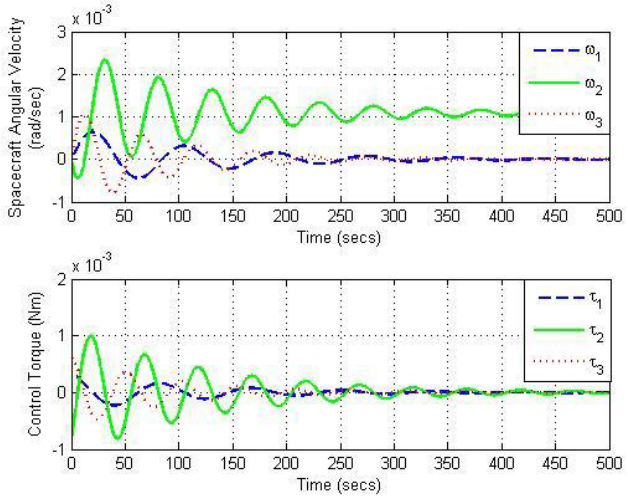

Figure 3. Angular velocity (top plot) and control torque (bottom plot) as a function of time for the best controller design found using the GA with β = 1.

The efficacy of the GA for gain tuning is compared to that of a human control systems engineer using the informed trial-and-error approach described in the preceding section, with the weight factor set to β = 1. Starting with both gain matrices initialized at 0.05 $\mathbf{I}_{3x3}$, different gain combinations were tested in simulation and then adjusted, based on the observed performance. Using this approach, the best-performing gain values found were **K** = diag (0.0175 0.02 0.025) and **L** = 0.1 $\mathbf{I}_{3x3}$. For this gain combination, a fitness score of f = -186.96 was achieved, resulting from a settling time of T_s = 186.9 s and a control effort of Ce = 0.0626 N m s. It is interesting to note that the hand-tuned gains resulted in the system taking 10 seconds longer to settle, but only required approximately 60% of the control effort needed by the GA-derived solution. The extent to which settling time and control effort must be balanced is up to the engineer concerned with a specific application, and a strength of the GA is that it can be quickly re-run for different performance metric weightings as desired.

Referring back to the best-performing design found using the GA, there is a large difference in the numerical value of the settling time and the amount of control effort expended. The differences in magnitude between T_s and Ce (176.1 vs. 0.1) indicate that it might be worthwhile to increase the weight factor β in order to force the fitness function to more evenly balance between the contributions from the two performance metrics during the evolution of the design population. In the preceding test case, the magnitude of the settling time was large enough that the GA selected a design that minimized settling time as much as possible, at the consequence of a greater amount of control effort being expended. If one were to increase the weight factor to β = 1000, a typical value of βCe (as calculated within the fitness function) might be on the order of 100 N m s, as opposed to 0.1 N m s. If the fitness function aims to minimize the combination of settling time and control effort, this should have the effect of causing the GA to favor designs with a longer settling time that expend less control effort. To test this idea, the same four GA trials were re-run with β = 1000, the results of which are reported in Tables 3 and 4. Note that in Table 3 the value of β is included in the value of βCe to reflect the balance that the GA is being asked to make via the fitness function. As such, one must divide by 1000 to see the value of Ce associated with a specific design solution.

Table 3. GA performance for four trials with β = 1000

Pop Size	Num	Xover	f_{max}	Ts (s)	βCe (Nms)
40	20	80	-270.38	237.5	32.88
40	40	70	-403.8	268.87	135.1
40	50	70	-289.71	245.9	43.81
50	40	70	-270.36	230.8	39.56

Table 4. Gain values corresponding to the four GA trials with β = 1000

k_1	k_2	k_3	l_1	l_2	l_3
0.0049	0.0112	0.0096	0.0952	0.0937	0.0826
0.0017	0.035	0.0604	0.0794	0.0715	0.0889
0.0001	0.0033	0.0223	0.0952	0.081	0.0921
0.0096	0.0175	0.0064	0.0984	0.0889	0.0905

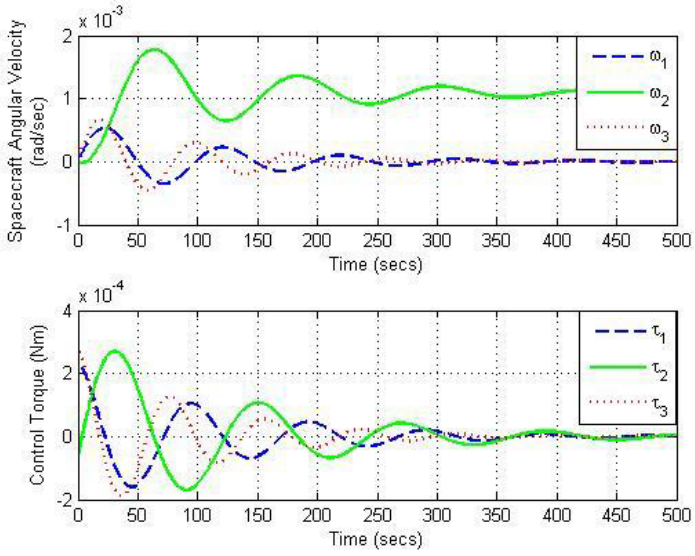

Figure 4. Angular velocity (top plot) and control torque (bottom plot) as a function of time for the best controller design found using the GA with β = 1000.

As can be seen in Table 3, the design solution achieved using a population of 50 designs evolved over 40 generations is just barely the best overall performer. For this trial, in which f = -270.36, the value of settling time is about 44 seconds greater than in the original GA trials, and the control effort has been reduced by about 2/3. Thus, the weight factor β has the intended effect, in that the fitness function now favors design solutions that provide a more even balance of the two performance metrics. It is important to note that it is not appropriate to compare the values of f_{max} for β = 1 and β = 1000. By adjusting the weight factor β, the basic structure of the fitness function has now changed, and the GA is trying to optimize something different in each test case. Instead, one should compare values of T_s and Ce across the two trials, as these do not change, regardless of the structure of the fitness function. The time response of the best-performing design solution with β = 1000 can be seen in Figure 4. Note the scale of the control torque plot is 10 times smaller than in Figure 3.

The design space under study in this research is obviously quite complex. While it is essentially infeasible to exhaustively test every admissible design solution in this problem, it is possible to test cross-sections of the design space to better understand the relationship between certain gain terms and the resulting fitness value. The approach taken was to set β = 1 and hold all gain values constant except for two, and then to vary these two gains across all admissible values. In the first test, all gain values were held to their "optimal" values as dictated by the best-performing GA design (with β = 1), except the first two entries of the proportional gain matrix **K** (k_1 and k_2). This results in a controller with gains **K** = diag (k_1 k_2 0.0302) and **L** = diag (0.0968 0.0968 0.0984), where k_1 and k_2 are varied between 0.0001 and 0.1 using all 64 possible values. By calculating the fitness value for each possible gain combination, it is possible to generate a surface in 3D space (seen in the left-hand image of Figure 5 below), showing how the fitness value varies with changing values of k_1 and k_2. The highest fitness score found in this cross-sectional optimization test is f = -165.53, corresponding to T_s = 165.4 s and Ce = 0.13 N m s. This performance was achieved using k_1 = 0.0449 and k_2 = 0.0641. It is important to notice that there is one point (k_1 = k_2= 0.0001) in Figure 5 which has an extremely low fitness value. Referring back to Section 2, one requirement introduced in the stability proof of Sanyal and Chartuvedi (2008), was that all values of the gain matrix **K** be non-equal. Given that this gain combination violates that requirement, the extremely poor fitness value may not be surprising. Logic was written into the genetic algo-

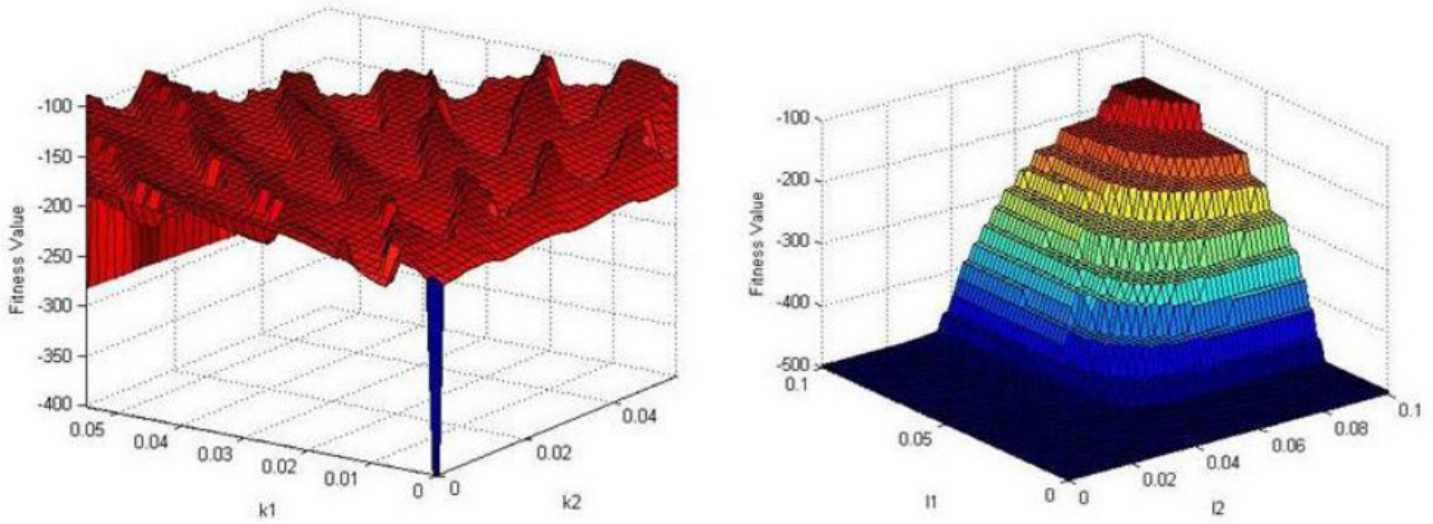

Figure 5. Fitness value as a function of gain combination when all gains are held constant except k_1 and k_2 (left plot) and a similar plot for l_1 and l_2 (right plot). The gains under study are uniformly varied between 0.0001 and 0.1.

rithm to discard randomly generated values of **K** that violated the stability requirements of (7), thus avoiding any such outliers.

A similar test to determine the relationship between the fitness value and certain gain terms was performed for the derivative gain matrix **L**, in which the gain matrices were set to **K** = diag (0.0604 0.0017 0.0302) and **L** = diag (l_1 l_2 0.0984) for all possible values of l_1 and l_2. Seen in the right-hand image of Figure 5, there are no discontinuities in this fitness surface because the only requirement imposed on the gain matrix **L** is that it be positive definite. Thus, all gain combinations within the gain range being tested are valid solutions for the controller design problem, in that all values tested are nonzero. For this exhaustive search test, the maximum fitness value achieved is f = -175.5, corresponding to a settling time of T_s = 175.4 s and Ce = 0.1 N m s. This performance was accomplished with the two derivative-type gains set to their maximum value, such that $l_1 = l_2 = 0.1$.

5. Discussion

The results presented in the preceding section indicate that the GA-based *a priori* gain tuning approach yields comparable performance to a hand-tuning approach, and has a greater likelihood of outperforming the hand-tuning strategy. For the study in which β = 1, the GA was able to find a design with a better maximum fitness score, and was also able to search a broader swath of the overall design space. As a point of reference, during the roughly three hour execution time of the GA, 2000 design solutions were tested, while during the informed trial-and-error approach, approximately 120 designs were tested. Furthermore, during the three hours in which the GA was executing, no human involvement was necessary, which means that on a real-world project, the attending engineer could be performing other tasks during the GA design process. It is noteworthy that the hand-tuned solution yielded a design that expended less control effort. The need to minimize either settling time or control effort expended is obviously a choice of the control systems engineer, and depends on the application. For the Hawai'iSat mission, it was necessary to slew the spacecraft around to point the hyperspectral imager to the required location in a relatively short amount of time, and thus settling time was typically prioritized higher than control effort expended. Depending on the specific application in question, a reverse set of priorities may in fact hold, but the fitness function can easily re-tuned to reflect a different balance.

The relative importance of settling time and the amount of control effort expended also inform the selection of the weight factor β. Since the numerical values of settling time for the tracking maneuver are so

much larger than the numerical values of control effort expended across that maneuver, the impact of changes in control effort expended from one set of gains to another is much less observable. Increasing the value of β makes the impact of the control effort metric more observable, causing the GA to favor different gain combinations. For Hawai'iSat, the controller designs resulting from using β = 1 were appropriate for the specific needs of the tracking maneuver, but the same might not be true for other maneuvers or different spacecraft. The weight factor within the fitness function is a powerful tool for balancing multiple performance metrics, but the specifics of that balance must be the responsibility of the design engineer.

An important question for any controller design scheme is that of sensitivity of the solutions to changes in system parameters. One system model was used to test all of the controller design solutions within the GA population, and it is certain that some system parameters will differ between the spacecraft simulation and the real world hardware. To better understand the impact of parameter variations on the GA-derived 'optimal' controller, the entries of the diagonal inertia tensor **J** were varied ±2.5%, ±5%, ±7.5%, ±10%, and ±12.5%. The spacecraft simulation was then re-run for the tracking maneuver, all using the same fixed gains found via the GA optimization, but now for each of the perturbed values of **J**. A plot of the maneuver settling time and amount of control effort expended for the different changes in **J** can be seen in Figure 6.

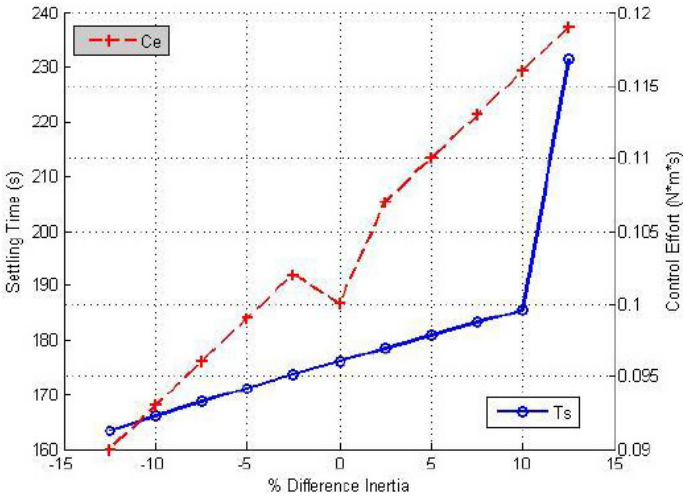

Figure 6. Settling time and amount of control effort expended as a function of percentage difference in the inertia tensor values.

As seen in Figure 6, both Ts and Ce tend to get better for decreasing values of the inertia tensor **J**, while they get worse for increasing values of **J**. This follows intuition, in that the same amount of control torque is being applied to a system that now has less mass, and thus the spacecraft can be driven more quickly to the desired final state without any "cost" to the amount of control effort expended. It is, however, interesting to see the large jump in the value of Ts when the inertia perturbation is increased from 10% to 12.5%. Apparently, when this much of a mass increase is seen, the selected controller design is no longer effective for the tracking maneuver. To better understand the phenomenon of changing the inertia tensor by 12.5%, the time history of the norm of the principal error direction ζ and the norm of the total control torque **τ** are plotted for the nominal inertia tensor ($\mathbf{J_0}$), the inertia tensor for a +12.5% change in inertia ($\mathbf{J_p}$), and the inertia tensor for a -12.5% change in inertia ($\mathbf{J_m}$). As seen in these plots, the controller performs worst when the inertia terms have been increased by a full 12.5%.

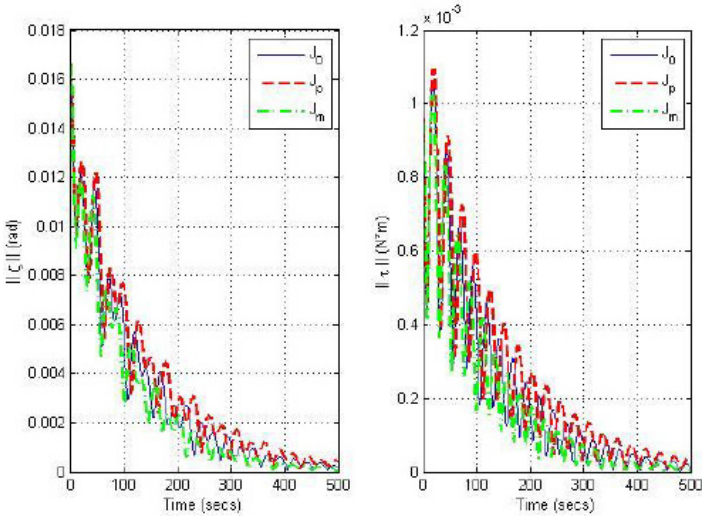

Figure 7. Comparison in the norm of the principal error angle ζ (left plot) and norm of control torque (right plot) for the nominal inertia tensor (J_0), the +12.5% inertia tensor (J_p) and the -12.5% inertia tensor (J_m).

Another aspect of parameter sensitivity that is of interest to a control systems designer is that of sensitivity to changes in the gain values themselves. Referring back to Figure 5, it is important to note first of all that there are no gain combinations that result in the system

becoming unstable. As such, one can use this figure to study how changing gain values will affect the overall fitness of a controller solution. For example, in the left plot, one sees that there are plateaus of constant fitness value whereby changes in l_1 or l_2 do not impact the performance of the spacecraft system. In contrast, the plot of variations in k_1 and k_2 show ridges of approximately constant fitness value. Thus, varying k_1 or k_2 might result in a decrease in the spacecraft system performance, although none of these ridges are extremely steep. This gain sensitivity is not a feature of the GA-based approach; it is a function of the design problem itself. A human control systems engineer would also encounter sensitivity to changes in the values of the **K** gain terms after selecting the best-performing gain values for implementation. This sensitivity to gain changes arises from the physical system and the controller under study, not the gain tuning methodology. It is very important to remember that the surfaces seen in Figure 5 are only one cross-section of a much more complex multi-dimensional design space, but studying these plots does provide some insight into the sensitivity of the controller gain terms.

6. Conclusion

This paper presents a strategy for tuning the gains of an almost globally stable control law offline for application to a small spacecraft attitude tracking maneuver. The spacecraft inertia properties, environmental moments, and actuator saturation limits are all adapted from the real world Hawai'iSat-1 small spacecraft mission. The tracking maneuver under study is motivated by the science payload within Hawai'iSat-1, and is applicable to a wide range of missions in LEO. The results indicate that the genetic algorithm-based approach to gain tuning is more efficient than an informed trial-and-error approach, in that a better fitness value was achieved and a wider range of gain values were tested in the same period of time. Preliminary sensitivity studies show that changes in system properties (such as the spacecraft inertia tensor entries) do impact the performance of the GA-based design, but it is important to remember that the same effects would be felt by a controller design tuned by hand. An important extension of this research would be to perform the same studies using a spacecraft model in which both sensor and actuator noise are included or the effects of using an estimation scheme are simulated.

Acknowledgments

This research was supported in part by the NASA Ames Research Center University Affiliated Research Center contract NAS2-03144, TO.080.1.EA.REE. The authors would like to thank the NASA Ames Technical Manager, Elwood Agasid, as well as the members of the University of Hawai'i Hawai'iSat-1 project team for their support in developing this work.

References

Abdel-Motagly, et al. (2003): Active control of nonlinear panel flutter under yawed supersonic flow, in *Proc. AIAA Conf. Structures, Struct. Dyn., and Materials*, Norfolk, VA.

Bayley, D., et al. (2008): Design optimization of a space launch vehicle using a genetic algorithm, in *J. Spacecraft and Rockets*, Vol. 45 (4), pp. 733-740.

Bullo, F. and Lewis, A. (2005): *Geometric control of mechanical systems.* 1st ed., New York: Springer, pp. 300-320.

Crespo, L., Matsutani, M., Annaswamy, M. (2010): Verification and tuning of an adaptive controller for an unmanned air vehicle, in Proc. *AIAA Conf. on Guid., Navig., and Cntrl*, Toronto.

Fisher, J., et al. (2007): Spacecraft momentum management and attitude control using a receding horizon approach, in *Proc. AIAA Conf. on Guid., Navig., and Cntrl,* Hilton Head, SC.

Grefenstette, J. (1986): Optimization of control parameters for genetic algorithms, in *IEEE Trans. On Sys., Man, and Cyber*., Vol. 16 (1), pp. 122-128.

Haupt, R. and Haupt, S.(2004): *Practical Genetic Algorithms*. Hoboken, NJ: John Wiley and Sons, pp. 40-80.

Hebrard, P. and Hentrot, A. (2003): Optimal shape and position of the actuators for the stabilization of a string. *System and Control Letters*, Vol. 48 (1), pp.

199-209.

Holland, J. (1975): *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press, pp. 120-125.

Kane, T., et al. (1983): *Spacecraft dynamics*. 1st ed., New York: McGraw-Hill, pp. 120-122.

Krishnakumar, K. and Goldberg, D. (1992): Control system optimization using genetic algorithms, *J. Guid.,, Cntrl., and Dyn*., Vol. 15 (3), pp. 735-740.

Lee, T., McClamroch, N. H., and Leok, M. (2005): A lie group variational integrator for the attitude dynamics of a rigid body with applications to the 3D pendulum, in *Proc. IEEE Conf. on Cntrl Apps*, Toronto, Canada.

Negnevitsky, M. (2002): *Artificial intelligence: a guide to intelligent systems*. London:,Pearson Education Ltd, pp. 112-137.

Nordkvist, N. and Sanyal, A. (2010): A lie group variational integrator for rigid body motion in SE(3) with applications to underwater vehicle dynamics, in *Proc. IEEE Conf. on Decision and Cntrl*, Atlanta, GA.

Omatu, S. and Yoshioka, M. (1997): Self-tuning neuro-PID control and applications, in *Proc. IEEE Conf. Sys.,Man, and Cyber*., Orlando, FL.

Peng, F., et al. (2006): Testing of a membrane space structure shape control using genetic algorithm, in *J. Spacecraft and Rockets*, Vol. 43 (4), pp. 788-793.

Qi, W., et al. (2006): A design of nonlinear adaptive PID controller based on genetic algorithms, in *Proc. 25th Chinese Cntrl Conf*. Harbin, China.

Robinnett, R., et al. (2005): Applied dynamic programming for optimization of dynamical systems, *Soc. Ind. and Applied Mathematics*, pp. 37-42.

Romano, M., et al. (2007): Laboratory experimentation of autonomous spacecraft approach and docking to a collaborative target. *J. of Spacecraft and Rockets,* vol. 44 (1), pp. 164-173.

Sand, T. (2009): Physics-based automated control of a spacecraft, in *Proc. AIAA Space Conf, and Exhibition,* Pasadena, CA.

Sanyal, A. and Chartuvedi, N. (2008): Almost global robust attitude tracking control of a spacecraft in gravity, in *Proc. AIAA Conf. on Guid., Navig., and Cntrl*, Honolulu, HI.

Sanyal, A. and Lee-Ho, Z. (2009): Attitude tracking control of a small satellite in low earth orbit, in *Proc. AIAA Conf. on Guid., Navig., and Cntrl*, Chicago, IL.

Sanyal, A. and Nordkvist, N. (2012): Attitude state estimation with multi-rate measurements for almost global attitude feedback tracking, *J. Guid., Cntrl., and Dyn.*, Vol 35 (3), pp. 868-880.

Seywald, H., et al. (1995): Genetic algorithm approach for optimal control problems with linearly appearing controls, *J. Guid,, Cntrl., and Dyn.*, Vol. 18 (1), pp. 177-182.

Slotine, J.J. and Li, W.P. (2005): *Applied nonlinear control*. Upper Saddle River, NJ: Prentice Hall, pp. 37-42.

Sorgenfrei, M. and Joshi, S. (2011): Further results towards a location-scheduled control methodology, in *Proc. AIAA Conf. on Guid., Navig., and Cntrl,* Portland, Oregon.

Sorgenfrei, M. and Joshi, S. (2011): Reconfigurable spacecraft controller design via location-scheduled control, *J. Guid., Cntrl., and Dyn*., Vol. 34 (5), pp. 1164-1173.

Sorgenfrei, M., Joshi, S., Agasid, E., and Hines, J. (2010): Towards a location-scheduled controller for nano-satellites, in *Proc. AIAA Conf. on Guid., Navig., and Control,* Toronto, Canada.

Sorgenfrei, M., Sanyal, A., and Joshi, S. (2012): Preliminary optimization results for an almost globally stable control law using a genetic algorithm, in *Proc. AIAA Conf. on Guid., Navig., and Cntrl,* Minneapolis, MN.

Wertz, J., and Larson, W. (1978): *Space mission analysis and design*. 1st ed., El Segundo, CA: D. Riedel Publishing Co., pp. 410-412.

Wie, B. (2005): *Space Vehicle Dynamics and Control,* 2nd ed, AIAA, pp. 450-457.