A Solely Magnetic Genetic–Fuzzy Attitude Control Algorithm for a CubeSat

Article *in* Journal of Spacecraft and Rockets · September 2015

DOI: 10.2514/1.A33294

3 authors, including:

Kelly Cohen
University of Cincinnati
214 PUBLICATIONS **1,035** CITATIONS

Solely Magnetic Genetic/Fuzzy-Attitude-Control Algorithm for a CubeSat

Alex R. Walker* and Philip T. Putman†
Sierra Lobo, Inc., Milan, Ohio 44846
and
Kelly Cohen‡
University of Cincinnati, Cincinnati, Ohio 45219

DOI: 10.2514/1.A33294

A genetic algorithm was used to optimize performance of a fuzzy inference system acting as a controller for a magnetically actuated CubeSat. A solely magnetically controlled satellite is a nonlinear, underactuated system for which the uncontrollable axis varies as a function of orbit position and attitude; variation is approximately periodic with orbit position. Therefore, controllability is not guaranteed, making solely magnetic control a less than ideal option for spacecraft requiring a high degree of pointing accuracy or spacecraft subject to relatively large disturbances. However, for small spacecraft, such as CubeSats, with modest pointing and disturbance rejection requirements, solely magnetic actuation is a good option. The genetic-algorithm-tuned fuzzy controller solution was compared to a similar linear quadratic regulator solution that was tuned to minimize the cost function used by the genetic algorithm. Both were optimized with respect to a single set of initial conditions. The genetic-algorithm-tuned fuzzy controller was found to be a lower-cost solution than the linear quadratic regulator for the optimized set of initial conditions. Additionally, a Monte Carlo analysis showed the genetic-algorithm-tuned fuzzy controller tended to settle faster than the linear quadratic regulator over a variety of initial conditions.

Nomenclature

A	=	submatrix of state dynamics matrix
a_D	=	gain multiplying derivative fuzzy inference system output
$a_{D,\theta}$	=	gain multiplying derivative fuzzy inference system angular error input
$a_{D,\omega}$	=	gain multiplying derivative fuzzy inference system angular velocity error input
a_P	=	gain multiplying proportional fuzzy inference system output
$a_{P,\theta}$	=	gain multiplying proportional fuzzy inference system angular error input
$a_{P,\omega}$	=	gain multiplying proportional fuzzy inference system angular velocity error input
$\bar{\boldsymbol{B}}$	=	local magnetic field vector, T
$\boldsymbol{b}$	=	component of magnetic field vector, T
C_D	=	spacecraft drag coefficient
D	=	derivative feedback gain
$E_{3\times3}$	=	3×3 identity matrix
F	=	linearized dynamics matrix multiplying states
G	=	linearized dynamics matrix multiplying control
$\hat{\boldsymbol{H}}$	=	orbit angular velocity unit vector
I	=	moment of inertia, $\text{kg} \cdot \text{m}^2$
i	=	orbit inclination, deg
J	=	cost function value
K	=	linear quadratic regulator feedback gain matrix
K_D	=	derivative fuzzy inference system
K_P	=	proportional fuzzy inference system
$\bar{\boldsymbol{M}}$	=	magnetic moment vector, $\text{A} \cdot \text{m}^2$
$\boldsymbol{m}$	=	magnetic moment vector component, $\text{A} \cdot \text{m}^2$
$\bar{\boldsymbol{m}}$	=	Earth magnetic dipole vector, $\text{A} \cdot \text{m}^2$
P	=	proportional feedback gain
Q	=	state weighting matrix for linear quadratic regulator design
q	=	unit quaternion
R	=	control weighting matrix for linear quadratic regulator design
$\boldsymbol{r}$	=	component of position vector of spacecraft with respect to Earth center of mass, m
$\bar{\boldsymbol{r}}$	=	position vector of spacecraft with respect to Earth center of mass, m
S	=	planform area, m^2
t	=	time, s
$\boldsymbol{u}$	=	input control vector of linearized system dynamics
$\boldsymbol{V}$	=	velocity vector component, m/s
$\bar{\boldsymbol{V}}$	=	velocity vector, m/s
$\boldsymbol{v}$	=	eigenaxis or Euler axis of axis-angle attitude representation
$\boldsymbol{x}$	=	linearized state vector
$\tilde{\boldsymbol{x}}$	=	first linearized component of quaternion
Y	=	matrix solution of linear quadratic regulator
$\tilde{y}$	=	second linearized component of quaternion
$\tilde{z}$	=	third linearized component of quaternion
δ	=	moment arm distance of force to center of mass, m
ε	=	obliquity of the ecliptic, deg
θ	=	angle of axis-angle attitude representation, deg
μ_E	=	specific gravitational constant (geopotential) of Earth, m^3/s^2
μ_0	=	permeability of free space, $\text{kg}\cdot\text{m}/\text{A}^2\cdot\text{s}^2$
ρ	=	atmospheric density, kg/m^3
σ	=	angular velocity of sun-pointing reference frame with respect to inertial frame, rad/s
τ	=	torque, $\text{N} \cdot \text{m}$
φ	=	angular velocity of spacecraft orbit with respect to inertial frame, rad/s
ψ	=	angle between spacecraft body-fixed frame axis and corresponding desired frame axis, deg
Ω	=	orbit right ascension of the ascending node, deg
ω	=	angular rate vector component, rad/s
$\bar{\boldsymbol{\omega}}$	=	angular rate vector, rad/s

Received 10 March 2015; revision received 22 July 2015; accepted for publication 26 July 2015; published online 21 September 2015.

*Aerospace Engineer, Department of Research and Technology, 11401 Hoover Road. Professional Member AIAA.

†Senior Research Scientist, Department of Research and Technology, 11401 Hoover Road.

‡Professor of Aerospace Engineering, School of Aerospace Systems, P.O. Box 210070. Associate Fellow AIAA.

$\tilde{\omega}$ = linearized angular rate, rad/s

Subscripts

ave = average quantity
d = drag
err = error
f = final condition
g = gravity gradient
M = magnetic
max, sim = maximum of the simulation
x = x component of three-dimensional vector
y = y component of three-dimensional vector
z = z component of three-dimensional vector

I. Introduction

THIS paper presents the use of a genetic algorithm to perform an offline optimization of a fuzzy logic-based gain-scheduled proportional-derivative (PD) feedback attitude control algorithm for a solely magnetically actuated CubeSat. The work in this paper is motivated by the design of the attitude control system for CryoCube™-1, which is a CubeSat designed and built by Sierra Lobo, Inc., and the NASA Kennedy Space Center. The purpose of CryoCube-1 is to study condensation and steady-state behavior of liquid oxygen in a reduced-gravity environment. Thermal and communications analyses showed that CryoCube-1 could achieve its mission objectives if it maintained a sun-pointing orientation. A preliminary comparison of the magnitude of disturbance torques with the magnitude of available magnetic control torque suggested that magnetic control could be sufficient to maintain the desired attitude. However, the magnetic control torque can only be exerted in the plane perpendicular to the local magnetic field, making a solely magnetically actuated satellite an underactuated system that lacks guaranteed controllability. Numerous techniques have been applied to develop stable feedback controllers for solely magnetically actuated satellites, and the literature shows generally positive results [1–29].

One of the key advantages of solely magnetic attitude control is that magnetic actuators generally consume less power and require less volume than other typical attitude control hardware. Magnetic actuators are essentially electromagnets that allow the spacecraft to exert a torque on the Earth via Earth's magnetic field. The spacecraft exchanges its momentum with Earth rather than exchanging its momentum with onboard wheels, as is done with reaction wheel and momentum wheel attitude control systems. Additionally, wheeled control systems generate torque through acceleration; friction and coupled rotational dynamics tend to accelerate the wheels undesirably. Thus, wheels must be powered even when not generating useful torque. The electromagnets of a magnetic control system, however, generate torque only when powered. Furthermore, wheeled attitude control systems can accumulate a relatively large amount of stored momentum, which is another consequence of accelerating to produce control torque. This stored momentum usually must be dissipated to allow the wheels to provide any more useful control torque. Therefore, spacecraft that use wheeled attitude control systems generally use another form of attitude control, such as cold gas thrusters, to provide a wheel momentum dumping mechanism. In fact, magnetic actuation was first used as a means to dump excess momentum from momentum wheels [1].

Martel et al. were the first to suggest magnetic actuators as the sole means of active satellite attitude control for improving the stability of a passively gravity gradient stabilized spacecraft [1]. They used an Euler angle and angular rate feedback, took a classical transfer function root analysis of the linearized system in order to show stability, and simulated a spacecraft using the designed control law along with a previously presented Kalman filter to demonstrate expected performance [1].

Almost a decade later, Wiśniewski published his Ph.D. thesis, which outlined the techniques he had applied to solely magnetic actuation for attitude control of the Ørsted satellite [2]. The thesis demonstrated design of an infinite-horizon linear quadratic regulator (LQR), a finite-horizon LQR, and a constant-gain LQR, but the main results were the nonlinear, PD feedback sliding mode and energy-based Lyapunov-stable controllers designed for mission-specific maneuvers. Wiśniewski also suggested possibly using fuzzy logic with the energy-based Lyapunov-stable controllers to design a fuzzy PD magnetic attitude controller [2]. Wiśniewski [3] and Wiśniewski and Markley [4] presented work on periodic LQRs, Wiśniewski and Blanke [5] presented work on energy-based Lyapunov-stable PD controllers, Wiśniewski [6] presented work on sliding mode PD controllers, and Wiśniewski and Stoustrup presented new work on periodic $\boldsymbol{H}_2$ control in [7].

Over the last 15 years, many have followed Wiśniewski's lead [2], developing periodic LQRs [8–13], Lyapunov-stable PD controllers [14–19], and $\boldsymbol{H}_\infty/\boldsymbol{H}_2$ controllers [20], whereas some have investigated classical PD design [8,21–24], model-predictive [11,25], time-optimal [26,27], and fuzzy logic methods [28,29]. For the most part, the classical PD design focuses on pole placement [21] or iteratively hand-tuning design parameters to achieve desired performance [8,23]. Likewise, much of the work done with LQRs is very similar, although [10,11] have examined discrete-time models and [12] examined the performance of an LQR with an extended Kalman filter. Lovera and Astolfi presented many different Lyapunov-stable PD controllers for the magnetically actuated satellite attitude control problem. Most notably, they presented general scaling conditions for local exponential stability with quaternion and angular rate feedback and conditions for local exponential stability with saturated PD feedback in [15], and they presented a locally exponentially stable proportional feedback controller in [16]. $\boldsymbol{H}_\infty/\boldsymbol{H}_2$, model predictive, and fuzzy logic controllers, although less investigated for magnetically actuated satellite attitude control, have shown results consistent with those of the other feedback control methods, although $\boldsymbol{H}_\infty/\boldsymbol{H}_2$ and model predictive controllers can be used to achieve greater robustness [29].

The time-optimal controllers investigated for the magnetic actuation problem are unique, in that it has been shown that an attitude error on the order of 40 to 60 deg can be reduced to zero in several minutes [26,27], whereas all aforementioned feedback controllers have settling times on the order of orbital period(s) [1–27]. Long settling times are not surprising, given the fact that a magnetically induced torque can only be exerted perpendicular to the local geomagnetic field; taken at every instant in time, the system is uncontrollable. Some analysis has been done to quantify the degree to which magnetically actuated satellites may be controlled. The most general analysis, performed by Bhat, showed that the magnetically actuated satellite system is guaranteed to be strongly accessible if the magnetic field and its time derivative are linearly independent at every instant [30]. Furthermore, controllability was found to be guaranteed if, in addition to the magnetic field and its time derivative being linearly independent at every instant, the magnetic field is periodic in time [30]. However, the near-linearly independent, near-periodic nature of more complex orbit models and models of Earth's magnetic field have not been rigorously analyzed, although Bhat suggested that there was some evidence that the near-ideal true magnetic field experienced in an actual low Earth orbit might be enough to guarantee controllability [30].

II. Modeling

CubeSats are small satellites, about 10 cm cubed per unit (referred to as a U). CryoCube™ is a 3U CubeSat, measuring about $10 \times 10 \times 30$ cm in its stowed configuration. All parts of the spacecraft are rigidly attached to the structure, and the vibrational modes of the spacecraft are very high frequency relative to the rotational dynamics modes. Therefore, as is the case with many satellites, it is reasonable to model CryoCube as a rigid body with only rotational degrees of freedom. Generally, translational degrees of freedom are governed by orbital equations of motion and are assumed unaffected by rotational degrees of freedom. The translational and rotational degrees of freedom are not decoupled, however, because disturbance torques influence satellite rotational motion and are dependent on translational degrees of freedom.

The well-known Euler moment equation [Eq. (1)] describes the time evolution of the angular rates of a rotating rigid body:

$$I\dot{\bar{\omega}} = -\bar{\omega} \times (I\bar{\omega}) + \bar{\tau} \tag{1}$$

Here, I is the moment of inertia tensor; $\bar{\omega}$ is the angular velocity expressed in body-fixed coordinates; and $\bar{\tau}$ is the torque exerted on the body, which is also expressed in body-fixed coordinates.

Unit quaternion parameterization of the attitude is used for its simplicity. Unit quaternion attitude parameterization is related to the axis-angle attitude representation, which parameterizes a coordinate transformation by a unit vector and an angle, by Eq. (2):

$$\bar{q} = \begin{bmatrix} \cos(\theta/2) \\ \sin(\theta/2)v \end{bmatrix} \tag{2}$$

Here, $\bar{q}$ is the quaternion four vector, θ is the angle of the axis-angle attitude representation, and $\boldsymbol{v}$ is the unit vector of the axis-angle attitude representation. Euler's rotation theorem describes the significance of the axis and the angle. According to Euler's rotation theorem, every rotation matrix in three dimensions has at least one real eigenvector. That is, every rotation corresponds to at least one direction that is unaffected by the rotation. For the axis-angle parameterization, $\boldsymbol{v}$ is the unaffected direction and is known as an eigenaxis or an Euler axis, and θ is the angle of rotation about the eigenaxis that yields the coordinate transformation.

The time derivative of the unit quaternion parameter representation of attitude is quadratic with respect to the states (i.e., attitude parameters and attitude rates), as shown in Eq. (3), whereas Euler angle parameterization requires using transcendental functions:

$$\dot{\bar{q}} = \frac{1}{2}\begin{bmatrix} 0 & -\boldsymbol{\omega}_x & -\boldsymbol{\omega}_y & -\boldsymbol{\omega}_z \\ \boldsymbol{\omega}_x & 0 & \boldsymbol{\omega}_z & -\boldsymbol{\omega}_y \\ \boldsymbol{\omega}_y & -\boldsymbol{\omega}_z & 0 & \boldsymbol{\omega}_x \\ \boldsymbol{\omega}_z & \boldsymbol{\omega}_y & -\boldsymbol{\omega}_x & 0 \end{bmatrix}\bar{q} \tag{3}$$

Here, $\boldsymbol{\omega}_x$, $\boldsymbol{\omega}_y$, and $\boldsymbol{\omega}_z$ are components of the angular velocity vector $\bar{\omega}$ expressed in body-fixed coordinates.

Unit quaternion parameterization is not a minimal representation, in that two different parameterizations represent one attitude. However, unit quaternion parameterization does not suffer from the existence of the singularities that befall other parameterizations, such as Euler angles [31].

In addition to the general rigid-body kinematics and dynamics, a magnetically actuated CubeSat is subject to additional forces, which appear as torque terms in the Euler moment equation. Potential sources of torque external to the satellite itself are gravity gradient torque, thermal radiation torques, solar wind torque, aerodynamic torque, and, of course, magnetic torques [32]. All external torques result from an associated force and a difference in location of the center of mass of the spacecraft and the point through which the associated force acts. Gravity gradient torques result from the attractive gravity force between the spacecraft and other massive objects, most notably the Earth; this force acts through the spacecraft's center of gravity, which is distinct from its center of mass. Thermal radiation torques are phenomena resulting from interactions of photons with the spacecraft. Solar wind and other ionizing radiation torques result from the interaction of the spacecraft with charged particles, such as protons, electrons, and alpha particles, as well as high-energy electromagnetic radiation produced by the sun and other sources of ionizing radiation. Aerodynamic torques result from the interaction of the spacecraft with gasses in the uppermost portions of the atmosphere. Lastly, magnetic torques result from the interaction of residual, operational, or intentionally induced magnetic fields of the spacecraft with the magnetic field of the Earth. The three most prominent torques to which CryoCube is subjected are aerodynamic, gravity gradient, and magnetic torques.

Aerodynamic torque is commonly modeled using a continuum model for fluid dynamics, although the density of the upper atmosphere and the speed of the spacecraft may place aerodynamic forces closer to a free molecular flow regime [33]. For a continuum fluid model, the aerodynamic drag is proportional to the dynamic pressure, the surface area perpendicular to the flow, and the drag coefficient of the spacecraft. Aerodynamic torque is calculated by taking the cross product of the vector from center of mass to the center of pressure with the aerodynamic force. In general, the drag coefficient and center of pressure vary with the orientation of the spacecraft with respect to the velocity vector. However, for ease of modeling, it is assumed that the drag coefficient and center of pressure are fixed. With this in mind, Eq. (4) models a continuum drag torque for a satellite:

$$\tau_D = \frac{\rho C_D}{2|\bar{V}|}\begin{bmatrix} S_y\delta_z V_y^3 - S_x\delta_y V_z^3 \\ S_z\delta_x V_z^3 - S_x\delta_z V_x^3 \\ S_x\delta_y V_x^3 - S_y\delta_x V_y^3 \end{bmatrix} \tag{4}$$

Here, ρ is the atmospheric density, C_D is the spacecraft drag coefficient, $\bar{V} = [V_x \quad V_y \quad V_z]^T$ is the spacecraft velocity in body-fixed coordinates, S refers to planform areas perpendicular to the body-fixed axes denoted by their respective subscripts, and $\delta = [\delta_x \quad \delta_y \quad \delta_z]^T$ is the moment arm distance of the drag force to the spacecraft center of mass.

Typical values for the spacecraft drag coefficient range from 2.0 to 4.0, but, as suggested in [32]: a value of 2.25 was chosen to model CryoCube.

Gravity gradient torque can be calculated using Eq. (5). Derivation of this expression can be found in [34]:

$$\tau_G = 3\frac{\mu_E}{|\bar{r}|^5}\begin{bmatrix} \boldsymbol{r}_y\boldsymbol{r}_z(I_z - I_y) \\ \boldsymbol{r}_x\boldsymbol{r}_z(I_x - I_z) \\ \boldsymbol{r}_x\boldsymbol{r}_y(I_y - I_x) \end{bmatrix} \tag{5}$$

Here, μ_E is the specific gravitational constant (sometimes called geopotential) of Earth; $\bar{r} = [\boldsymbol{r}_x \quad \boldsymbol{r}_y \quad \boldsymbol{r}_z]^T$ is the position vector (in body-fixed coordinates) of the spacecraft with respect to Earth's center of mass; and I_x, I_y, and I_z are the principal moments of inertia in the $\boldsymbol{x}$, y, and z axes of the spacecraft. It is assumed that the moment of inertia tensor is diagonal.

Magnetic torque can be calculated using Eq. (6):

$$\tau_M = \bar{\boldsymbol{M}} \times \bar{\boldsymbol{B}} \tag{6}$$

Here, $\bar{\boldsymbol{M}}$ is the spacecraft magnetic moment vector and $\bar{\boldsymbol{B}}$ is the local magnetic field vector. This equation assumes the spacecraft's magnetic moment is known, which is a generally good assumption for magnetically actuated satellites, because the controlled magnetic moment is proportional to the current flowing through the electromagnetic actuators.

Additional models are required to simulate attitude dynamics of a rigid-body magnetically actuated spacecraft with relatively high fidelity. Obviously, a magnetic field model is required. For the present paper, the International Geomagnetic Reference Field 2010 with secular perturbations [34,35] was used to simulate the magnetic field model for a period in early 2013. In addition to the magnetic field model, an orbit model is also required. Vallado's SGP4 orbit propagation algorithm [36] was used in conjunction with the two line element set (TLE) of a CubeSat in an orbit similar to the predicted orbit of CryoCube to provide a realistic simulation. The TLE of the Raiko CubeSat was chosen as a close approximation to that of CryoCube because Raiko was deployed from the International Space Station (ISS), and CryoCube is expected to operate in an orbit with a similar altitude and inclination to that of a CubeSat deployed from the ISS. Additionally, Raiko's TLE contains a drag coefficient term (B star), which tends to decrease orbit altitude over time. The Raiko and CryoCube have similar sizes, shapes, and masses, so the B-star term given in Raiko's TLE is likely similar to that which CryoCube may exhibit. The TLE for the Raiko CubeSat from February 2013[§] is given in Fig. 1.

Orbit parameters for this TLE are given in Table 1.

[§]Data available online at http://www.celestrak.com/NORAD/elements/cubesat.txt [retrieved 19 February 2013].

```
1 38852U 98067CN  13050.88445773  .00042037  00000-0  48588-3 0  1478
2 38852  51.6455 326.2357 0012049 280.6377  79.3267 15.61365943 21461
```

Fig. 1 Raiko TLE used to simulate CryoCube orbit.

The desired attitude of the CryoCube, shown in Fig. 2, is a sun-pointing orientation in which the X axis points toward the sun and the Z axis remains perpendicular to the ecliptic with the same sense as the Celestial North Pole, which is the Z axis of the Earth-centered inertial coordinate system.

III. LQR Design

A constant-gain infinite-horizon linear quadratic regulator was developed as a baseline solution with which to compare the performance of the genetic algorithm (GA)-tuned fuzzy system. An LQR was used as a baseline because, of all the techniques that have been previously applied to the magnetically actuated satellite problem, the LQR is the most widely demonstrated, and formulation of the LQR problem is most similar to formulation of the GA. In both LQR design and the GA-tuned fuzzy controller design, an algorithm is used to minimize a cost function subject to system dynamics. A constant-gain infinite-horizon LQR was chosen, in particular, because the computational expense of such controllers is minimal compared with that of time-varying-gain infinite-horizon and finite-horizon LQRs [2–4,10,11].

The LQR design requires a set of linear differential equations describing the system dynamics. Therefore, the equations of motion were linearized about the desired attitude, the state around which the system will ideally spend all of its time. The resulting linearized equations of motion [Eqs. (7–9)] reduce to a relatively complex matrix expression with terms that are a function of orbit position and velocity, which alone are periodic functions of time. The complexity of this expression is an artifact of the choice of reference frame. A simpler set of equations results if a nadir-pointing frame is chosen [9]:

$$\begin{bmatrix} I & 0_{3\times3} \\ 0_{3\times3} & E_{3\times3} \end{bmatrix} \begin{bmatrix} \dot{\tilde{\omega}}_x \\ \dot{\tilde{\omega}}_y \\ \dot{\tilde{\omega}}_z \\ \dot{\tilde{x}} \\ \dot{\tilde{y}} \\ \dot{\tilde{z}} \end{bmatrix} = \begin{bmatrix} 0_{3\times3} & A_D + A_G \\ \frac{1}{2}E_{3\times3} & 0_{3\times3} \end{bmatrix} \begin{bmatrix} \tilde{\omega}_x \\ \tilde{\omega}_y \\ \tilde{\omega}_z \\ \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix} + \begin{bmatrix} 0 & \boldsymbol{b}_z & -\boldsymbol{b}_y \\ -\boldsymbol{b}_z & 0 & \boldsymbol{b}_x \\ \boldsymbol{b}_y & -\boldsymbol{b}_x & 0 \\ & 0_{3\times3} & \end{bmatrix} \begin{bmatrix} \boldsymbol{m}_x \\ \boldsymbol{m}_y \\ \boldsymbol{m}_z \end{bmatrix} \tag{7}$$

Here, $E_{3\times3}$ is the 3×3 identity matrix, $\tilde{\boldsymbol{\omega}} = [\tilde{\omega}_x \quad \tilde{\omega}_y \quad \tilde{\omega}_z]^T$ is the linearized spacecraft angular velocity, $\tilde{\boldsymbol{v}} = [\tilde{x} \quad \tilde{y} \quad \tilde{z}]^T$ is the linearized quaternion attitude representation, $\bar{\boldsymbol{B}} = [\boldsymbol{b}_x \quad \boldsymbol{b}_y \quad \boldsymbol{b}_z]^T$ is the local magnetic field vector, and $\bar{\boldsymbol{M}} = [\boldsymbol{m}_x \quad \boldsymbol{m}_y \quad \boldsymbol{m}_z]^T$ is the spacecraft magnetic moment vector.

To solve the constant-gain infinite-horizon LQR problem, which is to minimize Eq. (10) subject to the linear system dynamics of Eq. (11), the algebraic Riccati equation (ARE) [Eq. (12)] must be solved. The constant-gain feedback matrix and resulting feedback control are then given by Eqs. (13) and (14), respectively:

$$J = \int_0^\infty (\boldsymbol{x}^T Q \boldsymbol{x} + \boldsymbol{u}^T R \boldsymbol{u})\,\mathrm{d}t \tag{10}$$

$$\dot{\boldsymbol{x}} = F\boldsymbol{x} + G\boldsymbol{u} \tag{11}$$

$$0 = YGR^{-1}G^TY - Q - YF - F^TY \tag{12}$$

$$K = R^{-1}G^TY \tag{13}$$

$$\boldsymbol{u} = K\boldsymbol{x} \tag{14}$$

Here, J is the LQR cost function, $\boldsymbol{x}$ is the state vector, $\boldsymbol{u}$ is the control vector, Q is the matrix that weights quadratic state terms in the LQR cost function, R is the matrix that weights quadratic control terms in the LQR cost function, t is time, F is the matrix that multiplies the state vector in the linear dynamic system of equations, G is the matrix that multiplies the control vector in the linear dynamic system of equations, Y is the solution of the Riccati equation, and K is the LQR feedback gain matrix that premultiplies the state vector to yield the control effort (i.e., the spacecraft magnetic moment). The columns of the K matrix that multiply the quaternion states are proportional gains, and the columns of the K matrix that multiply the angular velocity states are the derivative gains. The larger the values in the R matrix, the smaller will be the control effort and vice versa. The larger the values in the Q matrix, the more emphasis will be placed on maintaining smaller state error, and vice versa. The diagonal elements of the Q matrix weight squares of the states, whereas offdiagonal entries weight quadratic products of different states. Similarly, diagonal elements of the R matrix weight squares of the control inputs (magnetic moment components), whereas offdiagonal entries weight quadratic products of different control inputs. Q and R were chosen to be diagonal matrices because it is desired to minimize all states and all input control, and not specific

$$A_D = \frac{3\rho C_D}{|V|} \begin{bmatrix} S_y\delta_z V_y^2 V_z + S_z\delta_y V_y V_z^2 & -S_z\delta_y V_x V_z^2 & -S_y\delta_z V_x V_y^2 \\ -S_z\delta_x V_y V_z^2 & S_z\delta_x V_x V_z^2 + S_x\delta_z V_x^2 V_z & -S_x\delta_z V_x^2 V_y \\ -S_y\delta_x V_y^2 V_z & -S_x\delta_y V_x^2 V_z & S_x\delta_y V_x^2 V_y + S_y\delta_x V_x V_y^2 \end{bmatrix} \tag{8}$$

$$A_G = \frac{6\mu_E}{|\bar{\boldsymbol{r}}|^5} \begin{bmatrix} (\boldsymbol{r}_z^2 - \boldsymbol{r}_y^2)(I_z - I_y) & \boldsymbol{r}_x\boldsymbol{r}_y(I_z - I_y) & -\boldsymbol{r}_x\boldsymbol{r}_z(I_z - I_y) \\ -\boldsymbol{r}_x\boldsymbol{r}_y(I_x - I_z) & (\boldsymbol{r}_x^2 - \boldsymbol{r}_z^2)(I_x - I_z) & \boldsymbol{r}_y\boldsymbol{r}_z(I_x - I_z) \\ \boldsymbol{r}_x\boldsymbol{r}_z(I_y - I_x) & -\boldsymbol{r}_y\boldsymbol{r}_z(I_y - I_x) & (\boldsymbol{r}_y^2 - \boldsymbol{r}_x^2)(I_y - I_x) \end{bmatrix} \tag{9}$$

Table 1 Orbit parameters for Raiko TLE

Orbit parameter	value
Inclination, deg	51.6455
Eccentricity	0.0012049
Semimajor axis, km	6761.8
Mean anomaly (at epoch), deg	79.3267
Argument of perigee, deg	280.6377
Right ascension, deg	326.2357

combinations of states and input control. Furthermore, all proportional states were weighted equally, all derivative states were weighted equally, and all input control components were weighted equally, so that only two weight values were used to define the Q matrix and only one weight value was used to define the R matrix.

Average F and GG^T matrices were calculated to solve the ARE. The GG^T matrix was used instead of the G matrix because the average G matrix would be zero, which would result in no feedback control. The resulting GG^T matrix was used to calculate a positive definite G matrix, which was used to calculate the feedback gain matrix. A similar technique was presented in [9]. A simplified model was used to calculate the time-varying F and G matrices of Eq. (7). This simplified model was derived from an Euler angle representation of the position and velocity vectors of the spacecraft in the sun-pointing reference frame. Equation (15) gives the Euler angle representation of spacecraft position and velocity vectors with respect to the desired sun-pointing orientation:

$$\begin{bmatrix}\hat{\boldsymbol{r}}\\ \hat{\boldsymbol{V}}\\ \hat{\boldsymbol{H}}\end{bmatrix} = \begin{bmatrix}\cos(\phi t) & \sin(\phi t) & 0\\ -\sin(\phi t) & \cos(\phi t) & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & \cos(i) & \sin(i)\\ 0 & -\sin(i) & \cos(i)\end{bmatrix} \times \begin{bmatrix}\cos(\Omega) & \sin(\Omega) & 0\\ -\sin(\Omega) & \cos(\Omega) & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & \cos(\varepsilon) & -\sin(\varepsilon)\\ 0 & \sin(\varepsilon) & \cos(\varepsilon)\end{bmatrix} \times \begin{bmatrix}\cos(\sigma t) & -\sin(\sigma t) & 0\\ \sin(\sigma t) & \cos(\sigma t) & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\hat{\boldsymbol{x}}\\ \hat{y}\\ \hat{z}\end{bmatrix} \tag{15}$$

Here, $[\hat{\boldsymbol{r}} \quad \hat{\boldsymbol{V}} \quad \hat{\boldsymbol{H}}]^T$ is the orbital coordinate system defined by the set of spacecraft position, velocity, and orbital angular velocity unit vectors; $[\hat{\boldsymbol{x}} \quad \hat{y} \quad \hat{z}]^T$ is the set of unit vectors defining the sun-pointing coordinate system; t is time; φ is the angular velocity of the spacecraft orbit with respect to the inertial frame; i is the orbit inclination; Ω is the right ascension of the orbit; ε is the obliquity of the ecliptic; and σ is the angular rate of the sun-pointing reference frame with respect to the inertial frame. The first three matrices (from left to right) on the right-hand side of Eq. (15) transform from the inertial reference frame to the orbital reference frame. The last two matrices on the right-hand side of Eq. (15) transform from the sun-pointing reference frame to the inertial reference frame. In this instance, using these five coordinate transformation matrices is more intuitive than combining transformations to find a set of three Euler angles.

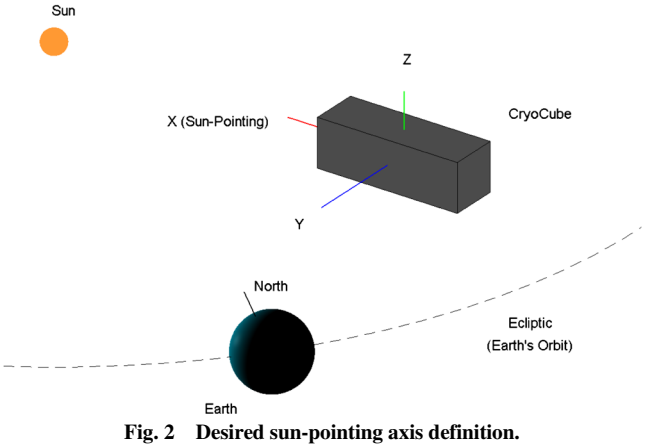


Fig. 2 Desired sun-pointing axis definition.

Earth's magnetic field was modeled as a dipole with a magnitude of 7.94×10^{22} A $\cdot$ m^2, for which the positive pole is opposite the direction of the North Magnetic Pole. The local strength and direction of the magnetic field at the spacecraft position are given by Eq. (16):

$$\bar{\boldsymbol{B}} = \frac{\mu_0}{4\pi|\bar{\boldsymbol{r}}|^5}(3\bar{\boldsymbol{r}}(\bar{\boldsymbol{m}} \cdot \bar{\boldsymbol{r}}) - \bar{\boldsymbol{m}}) \tag{16}$$

Here, $\bar{\boldsymbol{m}}$ is the magnetic dipole vector of Earth.

Components of the spacecraft position, spacecraft velocity, and local magnetic field vectors from the simplified model were substituted into the linearized equations of motion. The time-varying components of the F and GG^T matrices were averaged over all time through expansion of the expressions into a sum of first-order time-dependent sinusoidal functions and constant terms. It is assumed that the resulting expressions are periodic such that the average of the first-order time-dependent sinusoidal functions is zero. This treatment yields exact expressions for constant F and GG^T matrices. Specifically, the averaging technique yields $A_D = 0_{3\times3}$ and the exact expressions for the quadratic position terms of A_G are found in Eqs. (17–19). Additionally, Eq. (20) gives the expression for the GG^T matrix, and Eqs. (21–23) give expressions for the quadratic magnetic field terms:

$$\begin{aligned}\boldsymbol{r}_x^2 = \boldsymbol{r}_y^2 &= \frac{1}{4}|\bar{\boldsymbol{r}}|^2\cos^2(\Omega)\cos^2(\varepsilon)\cos^2(i) + \frac{1}{4}|\bar{\boldsymbol{r}}|^2\cos^2(\Omega)\\ &+ \frac{1}{2}|\bar{\boldsymbol{r}}|^2\cos(\Omega)\cos(\varepsilon)\cos(i)\sin(\varepsilon)\sin(i)\\ &+ \frac{1}{4}|\bar{\boldsymbol{r}}|^2\cos^2(\varepsilon)\sin^2(\Omega) + \frac{1}{4}|\bar{\boldsymbol{r}}|^2\cos^2(i)\sin^2(\Omega)\\ &+ \frac{1}{4}|\bar{\boldsymbol{r}}|^2\sin^2(\varepsilon)\sin^2(i)\end{aligned} \tag{17}$$

$$\begin{aligned}\boldsymbol{r}_z^2 &= \frac{1}{2}|\bar{\boldsymbol{r}}|^2\cos^2(\Omega)\cos^2(i)\sin^2(\varepsilon)\\ &- |\bar{\boldsymbol{r}}|^2\cos(\Omega)\cos(\varepsilon)\cos(i)\sin(\varepsilon)\sin(i)\\ &+ \frac{1}{2}|\bar{\boldsymbol{r}}|^2\cos^2(\varepsilon)\sin^2(i) + \frac{1}{2}|\bar{\boldsymbol{r}}|^2\sin^2(\Omega)\sin^2(\varepsilon)\end{aligned} \tag{18}$$

$$\boldsymbol{r}_x\boldsymbol{r}_y = \boldsymbol{r}_x\boldsymbol{r}_z = \boldsymbol{r}_y\boldsymbol{r}_z = 0 \tag{19}$$

$$GG^T = \frac{|\bar{\boldsymbol{m}}|\mu_0}{4\pi|\bar{\boldsymbol{r}}|^3}\begin{bmatrix}\frac{\boldsymbol{b}_y^2}{I_x^2} + \frac{\boldsymbol{b}_z^2}{I_x^2} & \frac{-\boldsymbol{b}_x\boldsymbol{b}_y}{I_xI_y} & \frac{-\boldsymbol{b}_x\boldsymbol{b}_z}{I_xI_z}\\ \frac{-\boldsymbol{b}_x\boldsymbol{b}_y}{I_xI_y} & \frac{\boldsymbol{b}_x^2}{I_y^2} + \frac{\boldsymbol{b}_z^2}{I_y^2} & \frac{-\boldsymbol{b}_y\boldsymbol{b}_z}{I_yI_z}\\ \frac{-\boldsymbol{b}_x\boldsymbol{b}_z}{I_xI_z} & \frac{-\boldsymbol{b}_y\boldsymbol{b}_z}{I_yI_z} & \frac{\boldsymbol{b}_x^2}{I_z^2} + \frac{\boldsymbol{b}_y^2}{I_z^2}\end{bmatrix} \tag{20}$$

$$\begin{aligned}\boldsymbol{b}_x^2 = \boldsymbol{b}_y^2 &= \frac{1}{2}\sin^2(\varepsilon) - \frac{3}{2}\sin^2(\varepsilon)\sin^2(i) + \frac{27}{16}\sin^2(\varepsilon)\sin^4(i)\\ &+ \frac{27}{16}\sin^2(\Omega)\cos^2(i)\sin^2(i) + \frac{9}{16}\cos^2(\Omega)\sin^2(i)\\ &+ \frac{27}{16}\cos^2(\varepsilon)\cos^2(\Omega)\cos^2(i)\sin^2(i) + \frac{9}{16}\cos^2(\varepsilon)\sin^2(\Omega)\sin^2(i)\\ &- \frac{3}{2}\cos(\varepsilon)\cos(\Omega)\sin(\varepsilon)\cos(i)\sin(i)\\ &+ \frac{27}{8}\cos(\varepsilon)\cos(\Omega)\sin(\varepsilon)\cos(i)\sin^3(i)\end{aligned} \tag{21}$$

$$\boldsymbol{b}_z^2 = \cos^2(\varepsilon) - 3\cos^2(\varepsilon)\sin^2(i) + \frac{9}{4}\cos^2(\varepsilon)\sin^4(i) + \frac{9}{4}\sin^2(\varepsilon)\sin^2(\Omega)\sin^2(i) + \frac{9}{4}\cos^2(\Omega)\sin^2(\varepsilon)\cos^2(i)\sin^2(i) + 3\cos(\varepsilon)\cos(\Omega)\sin(\varepsilon)\cos(i)\sin(i) - \frac{9}{2}\cos(\varepsilon)\cos(\Omega)\sin(\varepsilon)\cos(i)\sin^3(i) \quad (22)$$

$$\boldsymbol{b}_x\boldsymbol{b}_y = \boldsymbol{b}_x\boldsymbol{b}_z = \boldsymbol{b}_y\boldsymbol{b}_z = 0 \quad (23)$$

Notice that the GG^T matrix is diagonal, so an average (root mean square) G matrix can be simply calculated by taking the positive roots of the diagonal entries of the GG^T matrix.

A constant-gain infinite-horizon LQR can be designed using these constant matrices and a suitable choice of Q and R matrices. Bryson's rule was used as a first-cut approach to define Q and R. The Q and R matrices were then updated by hand to minimize the cost function used in the genetic algorithm optimization problem. This cost function [Eq. (27)] will be discussed in more detail in the next section.

IV. Genetic/Fuzzy System

The controller developed for this effort is a Mamdani fuzzy inference system for which the membership functions and rule base were tuned using a genetic algorithm. The fuzzy inference system acts as a gain scheduler, which is a function that changes feedback gains for different system operational regimes, because it dynamically outputs a scalar proportional gain and a scalar derivative gain, which are both functions of proportional error magnitude and derivative error magnitude. This section provides a brief overview of Mamdani fuzzy inference systems and genetic algorithms. After each is introduced, the method of implementation is briefly discussed.

A. Fuzzy Inference System

Fuzzy logic is a multivalued logic system that allows digital computers a means by which to emulate subjective decision making. Specifically, fuzzy logic can be used to infer a set of outputs based on a set of inputs and a set of logic-based rules that map inputs to outputs in what is termed a fuzzy inference system (FIS). Various methods, including the Mamdani and Sugeno methods, exist for performing the mapping, but all methods have been shown to allow fuzzy inference systems to act as universal approximators [37]. The property of universal approximation means that FISs (pronounced fis-es) can be used to approximate any function to any arbitrary degree of accuracy. However, with increasing accuracy, a FIS becomes more computationally complex.

In general, FISs perform three operations to transform inputs to outputs: fuzzification, inference/composition, and defuzzification. A set of if–then rules and the type of FIS determines how this transformation occurs. The if–then rules are posed as follows:

If *input* 1 is *adjective* 1 AND ... AND *input i* is *adjective j*, then *output k* is *adjective m*

Though this rule specifies only one output, one rule can include multiple outputs because multiple outputs do not affect one another; however, each output may be affected by all inputs. The adjectives in the rules, used to linguistically describe the inputs and outputs, are known as membership functions, and the inputs and outputs, or nouns, are known as linguistic variables [37]. Membership functions map a quantity of interest, such as a linguistic variable, to the range [01]. For a given value of the linguistic variable, the mapping represents the degree of truth of the statement:

linguistic variable n IS *adjective p*

Membership functions may overlap so that linguistic variables may "belong" to two or more, sometimes contradictorily named, membership functions. Fuzzification, represented by the vertical lines in Fig. 3, is the process of mapping all inputs of all rules to the range [0 1] via the membership functions [37]. Inference and composition combine all fuzzified values of each rule into single numbers, and then they use those numbers with the corresponding output variables' membership functions to produce modified output membership functions [37]. Minimum–maximum (min–max) composition, used for the satellite control FISs, calculates the minimum (intersection) of the antecedents for each rule, and then it uses these minima to truncate the corresponding output variables' membership functions such that, for a given output variable of one rule, the effective output membership function is the minimum (intersection) of the output membership function specified by the rule and the intersection of the antecedents of the rule over the range of all possible output values (usually limited to some finite range for simplicity) [37]. This process is shown in Fig. 3. For each output variable, composition also produces a composite output membership function by combining the effective output membership functions of all rules. For min–max composition, the composite output membership function of a given output is the maximum (union) of the effective output membership functions for the given output for all rules [37]. This process is shown in Fig. 4. The total effective output membership function for the output of interest is converted to a single numerical output by a process called defuzzification [37], represented by the vertical line in Fig. 4. The defuzzification method used by the satellite control FISs is a centroid method in which the

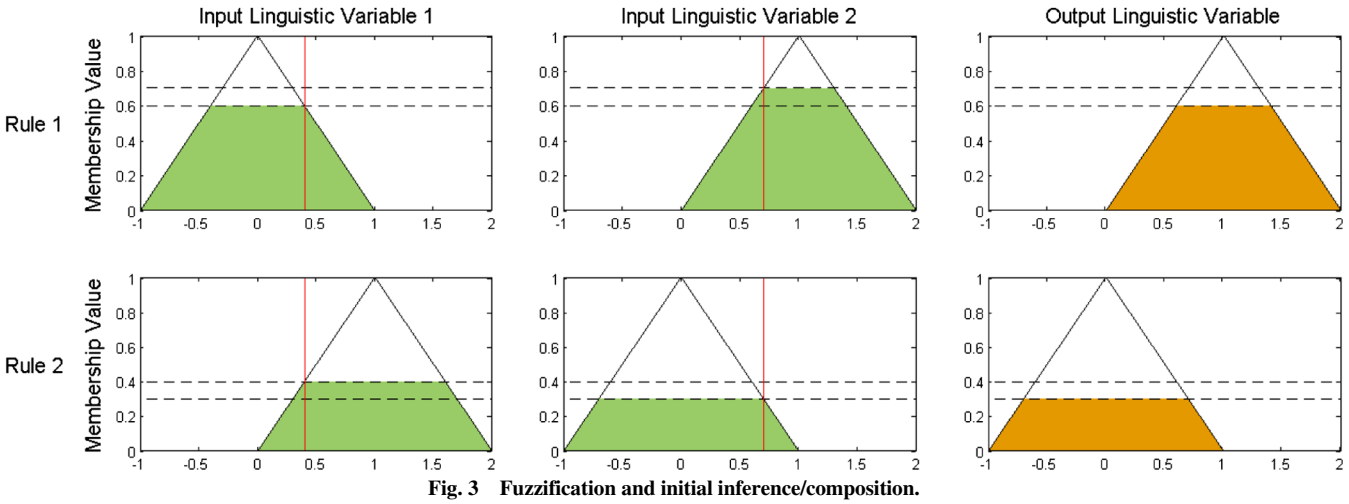


Fig. 3 Fuzzification and initial inference/composition.

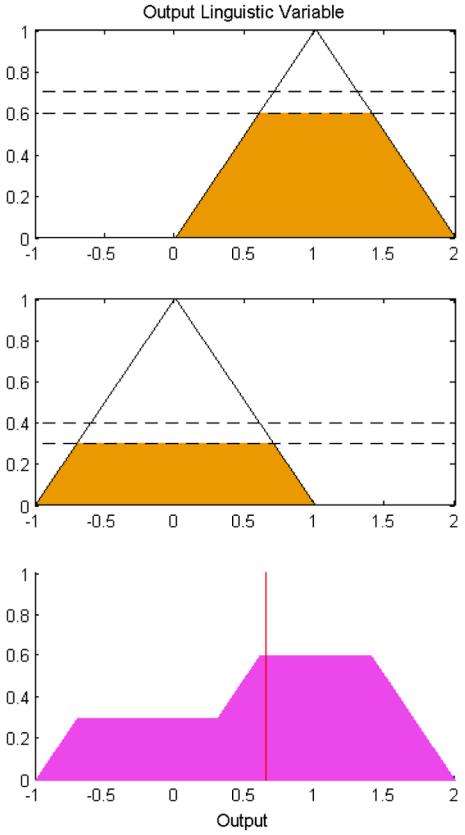


Fig. 4 Final inference/composition and defuzzification.

centroid of the total effective output membership function is used as the numerical output [37].

The gain scheduler for the magnetically actuated CubeSat attitude controller is a set of two Mamdani-type FISs; one outputs the proportional gain, described by Eq. (24), and the other outputs the derivative gain, described by Eq. (25):

$$P = a_P \cdot K_P \left(\min\left\{ a_{P,\theta}\left[1 - \left| 1 - \frac{2}{\pi}\theta_{\text{err}} \right| \right], 1 \right\}; \quad \min\{ a_{P,\omega} |\boldsymbol{\omega}_{\text{err}}|, 1 \} \right) \tag{24}$$

$$D = a_D \cdot K_D \left(\min\left\{ a_{D,\theta}\left[1 - \left| 1 - \frac{2}{\pi}\theta_{\text{err}} \right| \right], 1 \right\}; \min\{ a_{D,\omega} |\boldsymbol{\omega}_{\text{err}}|, 1 \} \right) \tag{25}$$

Here, P is the scalar proportional gain, a_P is a scalar gain that multiplies the proportional FIS output, K_P is the proportional gain FIS, $a_{P,\theta}$ is a scalar gain that multiplies the proportional FIS angular error input, $a_{P,\omega}$ is a scalar gain that multiplies the proportional FIS angular velocity error input, D is the scalar derivative gain, a_D is a scalar gain that multiplies the derivative FIS output, K_D is the derivative gain FIS, $a_{D,\theta}$ is a scalar gain that multiplies the derivative FIS angular error input, and $a_{D,\omega}$ is a scalar gain that multiplies the derivative FIS angular velocity error input.

The FISs have the same two inputs: the unit magnitude of θ_{err} and the magnitude of $\boldsymbol{\omega}_{\text{err}}$. Both inputs are restricted to the range [0 1]. Each normalized input and output range is traversed by three triangular membership functions. Constants tuned by the genetic algorithm multiply the inputs, thereby setting the effective input ranges in terms of true angular velocity and angular error. Two distinct FISs are used to allow the effective input ranges to differ, thereby allowing different regimes where one output may be relatively active while the other output may behave as if inputs are saturated.

The outputs of the FISs are proportional and derivative gains, which are multiplied by the proportional error and derivative error, respectively, and summed to calculate a control torque. The desired torque is then used to calculate a magnetic moment that can generate the component of the desired torque that is perpendicular to the local magnetic field, as described in Eq. (26). Recall that magnetic actuation can only exert torques perpendicular to the magnetic field, so it is impossible to exert desired torques parallel to the magnetic field with only magnetic actuation:

$$\bar{\boldsymbol{M}} = \frac{\bar{\boldsymbol{B}} \times (-I(P v_{\text{err}} \cos(\theta) + D\boldsymbol{\omega}_{\text{err}}))}{|\bar{\boldsymbol{B}}|^2} \tag{26}$$

Note that the proportional feedback is $v_{\text{err}} \cos(\theta)$, whereas the derivative feedback is the angular velocity error vector. When scaled by feedback gains and summed, these feedback terms prescribe the torque the controller should attempt to exert to correct angular rate error and attitude error, because a torque generated about the error axes described previously will cause the spacecraft to angularly accelerate about these axes, thus reducing the errors.

B. Genetic Algorithm

A genetic algorithm is a metaheuristic optimization procedure. GA operation is based on the theory of evolution via natural selection. In a GA, a set of potential solutions is encoded in some manner, analogous to the way DNA encodes instructions in living organisms [38]. The solutions can mutate, via random changes of random location(s) in the encoding, or reproduce with one another, via swapping of various location(s) in encodings [38]. A fitness function determines which solutions are most fit, and therefore have a better chance of reproducing [38]. The fitness function is the function that the GA tries to maximize. Usually, mutation occurs at a slower rate than crossover (the reproduction operation), but some problems, especially those with many local minima and large solution spaces, benefit from a higher mutation rate because the crossover process tends to converge relatively quickly to local minima [39]. Mutation allows more of the solution space to be explored because it closely resembles a random search. Generally, each generation completely replaces the previous generation, which can potentially remove good solutions from the population. However, schemes such as elitist selection, which copies the best member(s) of the previous generation to the new generation, help overcome this difficulty.

The GA used to find the best fuzzy controller used Eq. (27) as a cost function:

$$J = \boldsymbol{\omega}_{\text{ave}} + 5 \times 10^{-5}\theta_{\text{ave}} + 5 \times 10^{-8} t_f \tag{27}$$

Here, J is the GA cost function, $\boldsymbol{\omega}_{\text{ave}}$ is the average angular velocity error over the simulation period, θ_{ave} is the average axis-angle representation angular error over the simulation period, $t_{\text{max,sim}}$ is the maximum time simulated, and t_f is the final time the simulated axis-angle representation angular error is greater than 10 deg. The coefficients in Eq. (27) weight each term of the cost function approximately equally for a spacecraft that is capable of settling from a 60 deg error in approximately one orbit with a long-term average angular error of about 10 deg. This particular cost function was chosen because it is desired to find an attitude control algorithm that minimizes the time required for the spacecraft attitude error to settle to within 10 deg; a 10 deg pointing accuracy was deemed acceptable for mission success based on spacecraft thermal and communication analyses. Minimum time settling is desired because quick maneuvers off the nominal sun-pointing orientation were being considered as part of the mission profile. Attitude error and attitude rate error terms are included in the cost function so that the initial GA population, which may not have any solutions that settle, contains members with a variety of costs such that the GA does not initially act as a random search algorithm. The weights were chosen with an initial attitude error of 60 deg in mind because 60 deg was the size of the maneuver

Table 2 GA string parameterization scheme with example decoding

Parameter	Base	No. of digits	Example	Example decoded
Actuator on/off time	Binary	6	100,111	39
Input MF free vertex	Binary	5	00010	$2/31 = 0.0645$
Center output MF vertex	Binary	6	10,000	$16/31 = 0.516$
Output MF half-width	Binary	5	10,000	$8/31 = 0.258$
Input/output gain multiple	Decimal	2	26	2.6
Input/output gain exponent	Binary	3	010	2

MF, membership function.

Table 3 Input/output gain example calculations

Input/output gain	Multiple	Exponent	Exponent bias	Value
a_P	5.8	0	−5	5.80×10^{-5}
$a_{P,\theta}$	4.4	-3	0	4.40×10^{-3}
$a_{P,\omega}$	8.4	2	2	8.40×10^{4}
a_D	1.7	2	−3	1.70×10^{-1}
$a_{D,\theta}$	2.1	0	0	2.1
$a_{D,\omega}$	9.0	0	2	9.00×10^{2}

Table 4 FIS rule base for string 321132203

FIS		θ_{err}		
		Small	Medium	Large
h_{er}	Small	Large	Medium	Small
	Medium	Small	Large	Medium
	Large	Medium	—	Large

being contemplated, and one set of initial conditions was used to evaluate all controllers so that solution fitness would be consistent from generation to generation.

The fitness value of each member is found by subtracting the cost of each member from the maximum of all member costs such that the least fit member always has a fitness of zero. Members are selected for crossover via a roulette wheel selection, in which a member's probability of being selected is equal to its fitness value normalized by the sum of all member fitness values. The first generation starts with 50 randomly generated members. In subsequent generations, elitist selection keeps the best two members of the previous generation, whereas 15 population members are generated by mutation and 35 population members are generated by crossover.

The encoding is a 222-digit definition of FISs and other controller parameters. The entire encoding is not shown in order to conserve space, but the following description and Tables 2–4 demonstrate how portions of the encoding define fuzzy controller parameters. Two six-digit binary numbers define the amount of time the magnetic actuators are turned on and turned off; they must be turned off to take magnetic field measurements (note that simulations for both fuzzy and LQR controllers include cycling actuation on and off). The first row of Table 2 provides an example of the actuator time encoding. Five six-digit binary numbers define discrete locations on the range [0 1], where the free vertices of the input triangular membership functions of each input lie (recall there are two inputs for each of two FISs). The second row of Table 2 provides an example of the input membership function free vertex encoding. The leftmost input membership function is centered at zero, and the rightmost input membership function is centered at one; the vertices lying outside the input range do not require definition, as they cannot affect the output. One six-digit binary number defines the discrete location on the range [0 1] where the center of the output triangular membership function lies (recall there is one output for each of two FISs). The third row of Table 2 provides an example of the output center membership function center vertex encoding. The leftmost output membership function is centered at zero, and the rightmost output membership function is centered at one. All three output membership functions are symmetric, and their half-widths are defined by three five-digit binary numbers. The fourth row of Table 2 provides an example of the output membership half-width encoding. For each FIS, three numbers define FIS input and output gains, with each expressed in scientific notation. A two-digit decimal number defines the gain multiple in 10ths, and a three-digit binary number defines the power of 10 that multiplies this gain multiple. The fifth and sixth rows of Table 2 provide examples of the input/output gain encoding.

The power of 10 is biased for each input/output gain based on the authors' previous experience; however, the biases could be avoided if the size of the solution space was increased. Complete decoding of representative multiples and exponents for each of the FIS input/output gains is shown in Table 3 along with the specific exponent bias for each FIS input/output gain.

For each FIS, a nine-digit quaternary number defines the rule base, because each input has three possible membership functions for a three-by-three matrix of input combinations, and there are three possible outputs or no output that could be activated by a specific rule combination. Table 4 shows how this portion of the encoding generates a FIS rule base. The columns and rows of Table 4 define the antecedents of each rule, and the corresponding entry defines the output membership function that is activated by that particular rule. For example, row 1 and column 1 of Table 4 define the rule

"IF θ_{err} is Small AND $\boldsymbol{\omega}_{\text{err}}$ is Small, THEN output is Large"

V. Results

Controller design was accomplished by taking into consideration one set of initial conditions, termed the design initial conditions, and finding the controller that minimized the cost function described in the previous section over a period of 10 h. The initial error quaternion with respect to the desired quaternion was $[0.8660 \;\; 0.2811 \;\; -0.2008 \;\; 0.3614]^T$, corresponding to an angular error of 60 deg. The initial angular rate was $[0.001 \;\; 0.005 \;\; 0.002]^T$ rad/s, and the desired attitude was the sun-pointing reference frame described in Sec. II, which was nearly an inertial frame over the 10 h simulation period. The initial time of simulation was 19 February 2013 at 21:13:37.147872 p.m. Coordinated Universal Time, and the orbit is defined by the TLE given in Fig. 1. The CubeSat is taken to have principal moments of inertia: $I_x = 2.300 \times 10^{-2}$ kg · m^2, $I_y = 2.594 \times 10^{-2}$ kg · m^2, and $I_z = 2.600 \times 10^{-2}$ kg · m^2. Note that mass distribution of the spacecraft components is such that the principal moments are very similar. The faces perpendicular to the X axis are taken to be 100 cm^2, and the faces perpendicular to the Y axis and Z axis are taken to be 300 cm^2. A value of 2.25 [32] was used as the drag coefficient of the CubeSat, and the center of pressure is fixed at $[-6.711 \;\; 5.816 \;\; -8.142 \times 10^{-3}]^T$ mm from the center of mass as expressed in spacecraft body-fixed coordinates, aligned with the principal inertia axes.

All simulations were conducted using MATLAB with the default double precision floating-point arithmetic. MATLAB's Fuzzy Toolbox was also used to generate and evaluate FISs. The GA was written in MATLAB without using MATLAB's available GA functionality (part of the Global Optimization Toolbox).

LQR and fuzzy controller performance were analyzed and compared for the design initial conditions and for a random set of initial conditions evaluated as part of a Monte Carlo analysis. The following four subsections present the results of these analyses and comparisons. First, a high-level comparison of the performance with respect to the GA cost function is presented for controller responses to the design initial conditions. Next, the LQR response to the design initial conditions is analyzed. Then, the fuzzy controller response to the design initial conditions is analyzed. Finally, the Monte Carlo analysis and comparison of the performance of the two controllers with respect to the GA cost function is presented.

Table 5 Cost function comparison of LQR and genetic/fuzzy controllers for design initial conditions

Controller	Cost	Settling time, s	Average pointing error, deg	Average angular rate error, rad/s
LQR	1.2962×10^{-3}	4234	15.469	3.110×10^{-4}
Genetic/fuzzy	1.2285×10^{-3}	4123	14.020	3.214×10^{-4}
Genetic/fuzzy with Respect to LQR	5.2% smaller	2.62% faster	9.37% smaller	3.34% larger

A. Controller Comparison for Design Initial Conditions

A fuzzy controller for which the feedback gains vary with spacecraft attitude state and a constant-gain LQR were designed. The performance of both controllers will be discussed in more detail in the following sections; however, for clarity of purpose of the present work, Table 5 is given as an overview and a comparison of the performance of the two controllers with respect to the design initial conditions. It can be seen that, for the design initial conditions, the fuzzy gain scheduler outperforms the LQR in terms of the cost function given by Eq. (27). Table 5 also gives the complete breakdown of the cost function components, comparing the performance of the constant-gain LQR with the performance of the GA-optimized fuzzy controller. It can be seen that the fuzzy gain scheduler has a faster settling time and lower average pointing error but a larger average angular rate error for the design initial conditions.

B. LQR Response to Design Initial Conditions

The constant-gain infinite-horizon LQR was designed by tuning the Q and R matrices of Eq. (10), as well as the actuator on and actuator off times to minimize the cost function in Eq. (27). For simplicity, the Q and R matrices were kept diagonal and only three different weights were optimized: the weight on proportional error states, the weight on derivative error states, and the weight on control effort. Tuning was performed via an automated search of the five variables (two time parameters and three weight parameters). One at a time, variables were increased or decreased and, for each change, a corresponding controller was generated and its cost evaluated via Eq. (27). If increasing/decreasing a variable resulted in improved cost, the variable was increased/decreased until no improvement was realized; the lowest cost solution found thus far was saved to move forward. The algorithm then moved on to the next variable. This process was repeated until altering any one of the variables independent of the others only resulted in an increase in cost. This algorithm mimicked a hand-tuning strategy for cost minimization. It is important to note that the algorithm is not guaranteed to find the global minimum; however, a GA is not guaranteed to find the global minimum of a cost function either. The resulting Q and R matrices are shown in Eqs. (28) and (29), respectively; and the resulting feedback gain matrix, found by substituting orbit values obtained from the TLE and other physical constants into the average linearized F and G matrices, is shown in Eq. (30). For each actuation/measurement cycle, the optimal time for actuators to be turned on was found to be 43 s and the optimal time for actuators to be turned off was found to be 20 s:

$$Q = \begin{bmatrix} \frac{E_{3\times3}}{1.21\times10^{-8}} & 0_{3\times3} \\ 0_{3\times3} & \frac{E_{3\times3}}{1.289\times10^{-3}} \end{bmatrix} \tag{28}$$

$$R = \frac{E_{3\times3}}{5.76 \times 10^{-8}} \tag{29}$$

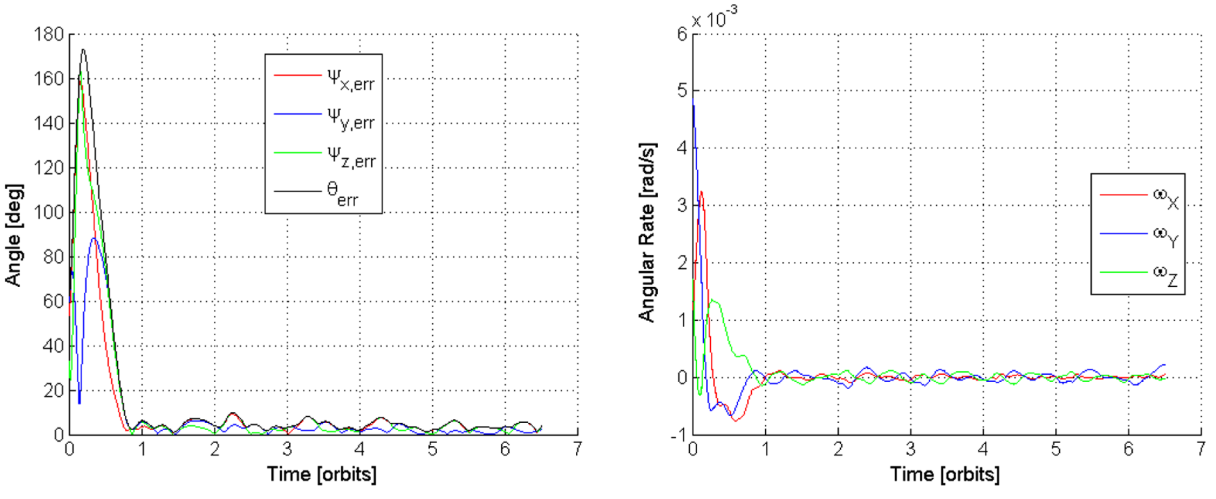
$$K = \begin{bmatrix} 6.680\times10^{-3} & 0 & 0 & 3.292 & 0 & 0 \\ 0 & 6.447\times10^{-3} & 0 & 0 & 3.373 & 0 \\ 0 & 0 & 6.685\times10^{-3} & 0 & 0 & 3.178 \end{bmatrix} \tag{30}$$

The LQR performs well, stabilizing the satellite to within a 10 deg angular error in the steady state, as shown in Fig. 5a. Note that ψ is the angle between the spacecraft body-fixed frame axis and the corresponding desired frame axis, whereas θ_{err} is the angle of the axis-angle representation of the transformation between the body-fixed frame and the desired frame. From the design initial conditions, it takes 4864 s, or just under one orbit, to settle to this steady state. The angular velocity error settles in approximately the same amount of time, as shown in Fig. 5b.

Figure 6 shows that control effort is initially relatively large, on the order of 10^{-3} A · m^2, to correct the set of initial conditions far from the steady state, but in the long term, control effort is on the order of 10^{-4} A · m^2. Actuator saturation is assumed to occur at 0.02 A · m^2, so with this control method and the scenario under investigation, there is no actuator saturation and a margin about four times larger than the largest observed dipole moment in this simulation.

C. GA-Tuned Fuzzy Response to Design Initial Conditions

The GA-automated design of the gain-scheduled fuzzy controller was run for 90 generations to converge on a solution. The plot of



a) Angular error b) Angular rate error

Fig. 5 Satellite state response to LQR for design initial conditions.

solution convergence in Fig. 7 shows that the initial population was composed of solutions far from optimal. Convergence slowed down significantly after a few generations, but the GA ultimately found a gain-scheduled fuzzy controller capable of outperforming the constant-gain infinite-horizon LQR solution for the design initial conditions. Performing the same optimization by hand would have been a laborious task, because the solution space for the fuzzy gain scheduler is very large and, before running the GA, it was unknown what gain scheduling strategy could improve upon both the settling time and steady-state accuracy.

The optimal fuzzy controller found by the GA is shown in the following series of figures and tables. These data describe how the fuzzy gain scheduler alters the feedback gains as a function of attitude and rate errors. Table 6 gives the values for the input and output gains for the two FISs. Figure 8 shows the proportional gain FIS membership functions, and Table 7 gives the proportional gain FIS rule base. Figure 9 shows the derivative gain FIS membership functions, and Table 8 gives the derivative gain FIS rule base. Additionally, the GA found that, for each actuation/measurement cycle, the optimal time for actuators to be turned on was found to be 20 s and optimal time for actuators to be turned off was found to be 50 s.

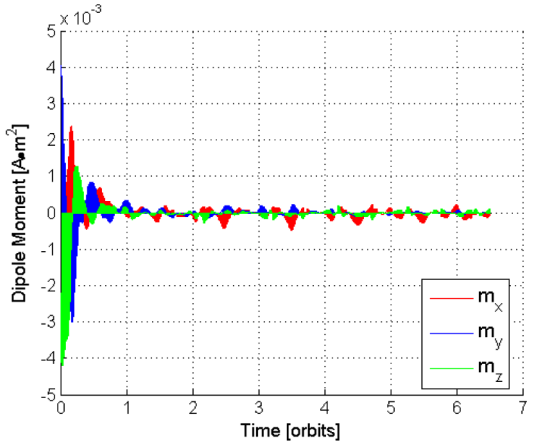


Fig. 6 Magnetic dipole control with LQR for design initial conditions.

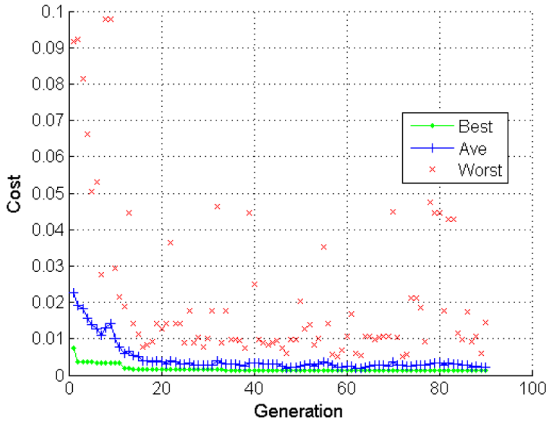


Fig. 7 Best, average (Ave), and worst costs of each generation.

Table 6 Fuzzy gain scheduler constants used in Eqs. (24) and (25)

Variable	Value
a_P	5.80×10^{-5}
$a_{P,\theta}$	4.40×10^{-3}
$a_{P,\omega}$	8.40×10^{4}
a_D	1.70×10^{-1}
$a_{D,\theta}$	2.1
$a_{D,\omega}$	9.00×10^{2}

Table 7 Proportional gain FIS rule base

		θ_{err}		
	P FIS	Small	Medium	Large
ω_{err}	Small	Small	Medium	Medium
	Medium	Large	—	Small
	Large	—	—	—

Table 8 Derivative gain FIS rule base

		θ_{err}		
	D FIS	Small	Medium	Large
ω_{err}	Small	—	—	Medium
	Medium	—	—	—
	Large	—	Medium	Medium

The optimized fuzzy solution state response is shown in Fig. 10, and the control effort is shown in Fig. 11. From the design initial conditions, the genetic/fuzzy solution takes 4123 s to settle to within a 10 deg pointing error, which is about 2.62% faster than the constant-gain LQR. The fuzzy controller also maintains an average angular error about 9.37% smaller than that of the LQR. The tradeoff of the faster settling and smaller pointing errors, however, is an increase in angular velocity error of 3.34% over the LQR.

Figure 12 shows that the proportional and derivative feedback gains do indeed change over the course of simulation runtime, suggesting that a gain-scheduled approach can outperform constant-gain feedback control in a magnetically actuated satellite. The proportional gain remains nearly constant for the duration of the simulated scenario, dropping to zero or near zero for brief instants when the derivative error becomes very small. The derivative gain is decidedly more active, increasing approximately an order of magnitude when derivative error is small.

Examination of the final FISs corroborates the behavior of the fuzzy gain scheduler and provides additional insight into the optimal strategy uncovered by the GA. The very small value the GA reached

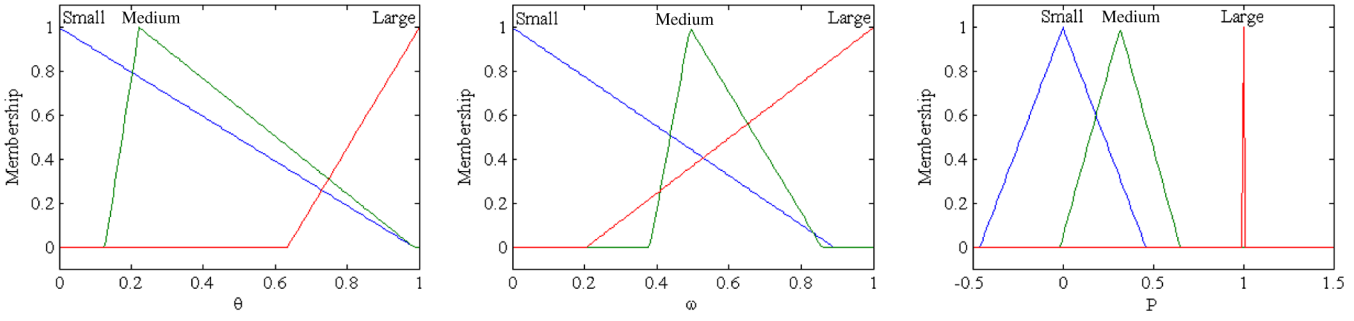


Fig. 8 Proportional gain FIS membership functions.

for $a_{P,\theta}$, shown in Table 6, drives all of the input angular errors toward zero according to Eq. (24). This, in turn, limits all possible input angular errors to the "small" membership function of the input labeled θ in Fig. 8, thus limiting the scope of rules that can be activated in the FIS to the small column of θ_{err} in Table 7. The relatively large value for $a_{P,\omega}$, shown in Table 6, drives the majority of the input angular velocity errors toward one, according to Eq. (24). This, in turn, drives most of the input angular velocity errors to the "large" membership function of the input labeled $\boldsymbol{\omega}$ in Fig. 8. Very small angular velocity errors occasionally activate the "medium" and small membership functions. Thus, most of the time, the small-angular-error/large-angular-velocity regime activates no output membership functions, forcing the FIS to output the default average of the output range: 0.5. Occasionally, the small angular error and medium or small angular velocity error regimes activate the large P output membership function and/or the small P output membership function. However, because the area of the large P output membership function is so small, the small P output membership function tends to dominate the response at smaller values of the angular velocity error. This explains the occasional drop of the proportional gain from about 2.9×10^{-5}, which is $0.5 \times 5.80 \times 10^{-5}$, to zero or near zero when the angular velocity error becomes very small.

As noted previously, the derivative gain is much more dynamic than the proportional gain, and examination of the derivative gain FIS explains the underlying reasons. The scaling factors $a_{D,\theta}$ and $a_{D,\omega}$, shown in Table 6, are large enough to allow the full range of both the angular error and angular velocity error input to Eq. (25) to contribute to the FIS output. However, Table 8 shows that, over all rules, there exist only two possible output options: activate the medium output membership function or output no output membership functions. This explains the steps observed in Fig. 12b from near 0.01 to 0.085. The lower bound value of 0.01 arises from the multiplication of the value corresponding to the location of the centroid of the medium D output membership function, shown in Fig. 9 at about 0.07, by a_D, shown in Table 6. The upper bound is the product of the average of the output range and a_D, or 0.5×0.17. Small deviations near 0.01 can be attributed to errors in the finite summation of areas used to evaluate the FIS.

One of the benefits of fuzzy logic is that its linguistic formulation allows humans to interpret what a FIS is doing without actually knowing how the "fuzzy math" works. In this sense, the GA generated a control strategy that the optimized fuzzy controller executes, and this strategy can be interpreted using common language. The overall strategy of the GA-optimized fuzzy gain scheduler can be summarized as follows: maintain constant nominal gains, decreasing the proportional gain to zero when the angular velocity error approaches zero and increasing the derivative gain about an order of magnitude

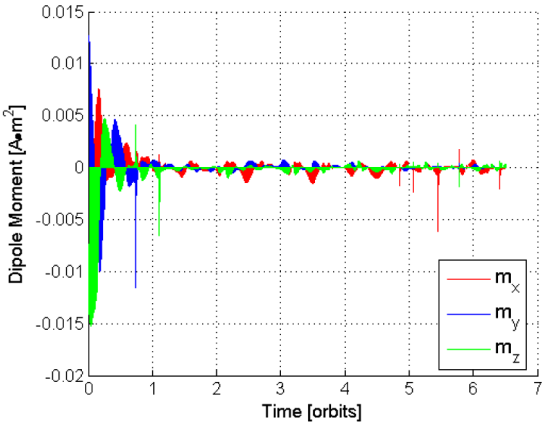


Fig. 11 Magnetic dipole control with GA-tuned fuzzy gain-scheduled PD controller for design initial conditions.

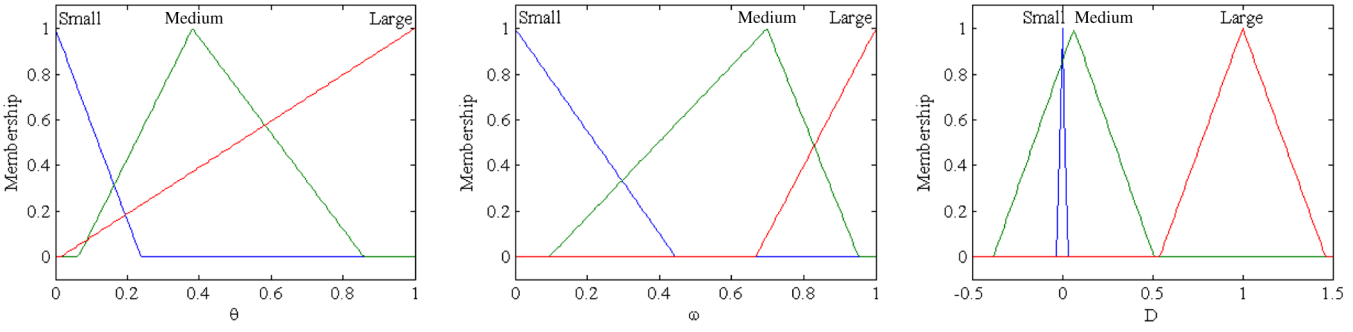
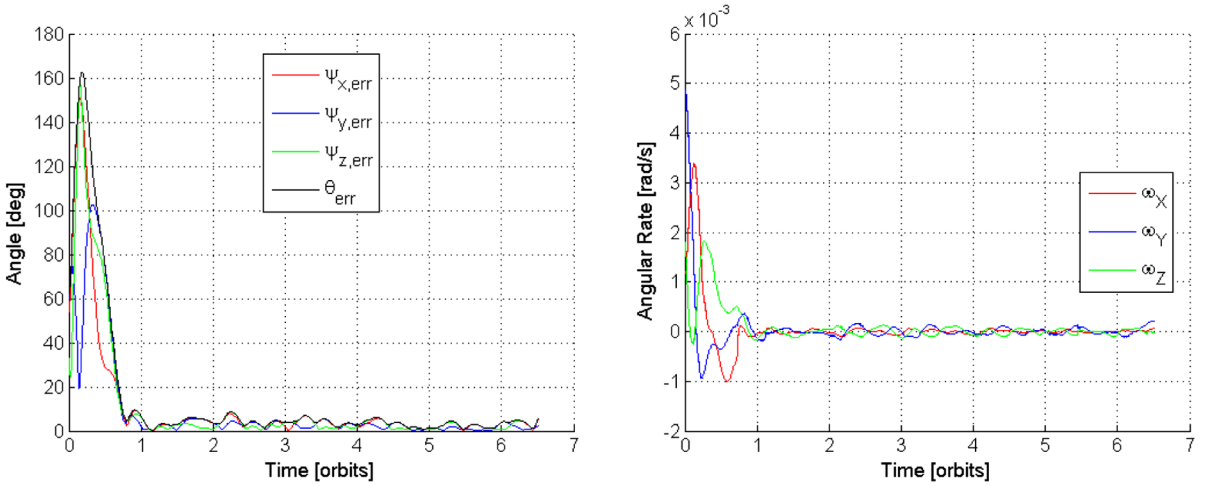
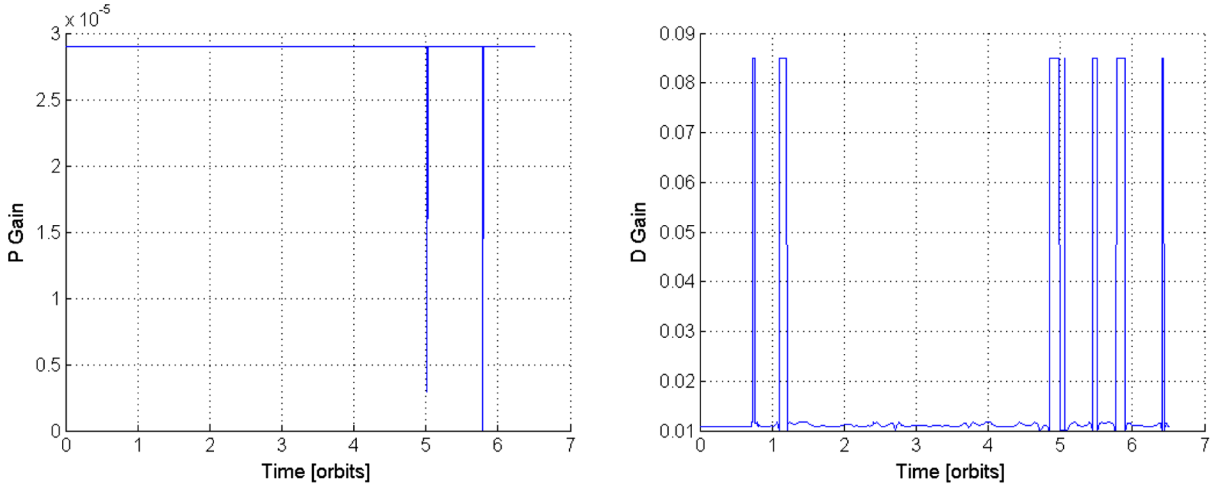


Fig. 9 Derivative gain FIS membership functions.



a) Angular error **b) Angular rate error**

Fig. 10 Satellite state response to GA-tuned fuzzy gain-scheduled PD controller for design initial conditions.



a) Proportional gain schedule b) Derivative gain schedule

Fig. 12 Fuzzy gain scheduler response for design initial conditions.

when the angular error approaches zero. Increasing the derivative gain under the conditions of a very small proportional error makes some sense intuitively, because at zero proportional error, any deviations from the desired attitude will first manifest as angular velocities. This is already accounted for, to a degree, in the PD feedback architecture; however, the fuzzy gain schedule tends to amplify this behavior near the steady state. On the other hand, decreasing the proportional gain to zero near the zero angular velocity error does not make much sense intuitively, because in an extreme case this could potentially stop feedback correction of the attitude so long as the angular velocity error remains bounded near zero. It was noted that this generally undesirable behavior was extremely brief in practice, so it may be that this behavior is in fact desirable for the few brief instances in the specific scenario used to train this fuzzy gain scheduler.

D. Monte Carlo Comparison of LQR and Fuzzy Controllers

A Monte Carlo analysis was run to determine the performance of both controllers for a variety of initial conditions. The number of trials was 1001, with random initial error quaternion, angular velocity, orbit right ascension, and orbit mean anomaly. All other simulation parameters were kept identical to the simulation used to design the LQR and fuzzy controllers. Random quaternions were generated by selecting points uniformly distributed on the unit sphere for the axis $\boldsymbol{v}$ and by selecting points uniformly distributed over the range [0 1] for $\cos(\theta/2)$; $\sin(\theta/2)$ was solved for by finding the positive root of $\sqrt{1-\cos^2(\theta/2)}$. Random angular velocity vectors, expressed in radians per second were generated by selecting three points uniformly distributed over the range $[-0.0025 \quad 0.0025]$. Random initial right ascensions and mean anomalies, expressed in degrees, were generated by selecting points uniformly distributed over the range $[0 \quad 360)$; these parameters were randomized to yield random magnetic field time histories possible for a spacecraft with the same semimajor axis, eccentricity, inclination, argument of perigee, and epoch as that given in Fig. 1 and Table 1.

Results of the Monte Carlo analysis, presented in Table 9, show that the fuzzy controller underperforms the LQR in terms of total cost, average angular error, and average angular rate error. All analyses of the statistical significance of population means were performed using the Student's t-distribution at the 99% confidence level. For both the LQR and the fuzzy controller, the average angular errors and the average angular rate errors reported in Table 5 are relatively close to the average angular errors and the average angular rate errors reported in Table 9. However, the average settling times reported in Table 9 are much larger (more than double) those reported in Table 5. The fuzzy controller does outperform the LQR in terms of settling time, on average settling to within a 10 deg error between 4.73 and 18.6 min faster than the LQR. Additionally, for the samples shown, the standard deviation of the fuzzy gain scheduler's settling time is also less than that of the LQR. Because the Monte Carlo analysis shows that both controllers do settle on average on the order of about 1.5 orbits, both controllers meet the 10 deg pointing requirement of the CubeSat for which they were designed.

Varying the design initial conditions, which act as a training scenario for the controllers, or evaluating the controllers using multiple training scenarios could improve the steady-state robustness of the fuzzy gain scheduler by allowing the GA to optimize the controller for a larger portion of the state space. An improved training methodology would be required to avoid optimizing the controller to a particular set or even particular sets of initial conditions.

VI. Conclusions

A genetic algorithm was used to tune a fuzzy gain-scheduled proportional-derivative feedback controller for a solely magnetically actuated CubeSat. The performance of the genetic/fuzzy control algorithm was compared to a constant-gain infinite-horizon linear quadratic regulator (LQR) solution for the set of design initial conditions. The genetic/fuzzy solution was shown to outperform an LQR that was tuned to minimize the same cost function used by the genetic algorithm in response to the design initial conditions. However, a Monte Carlo analysis showed that the genetic/fuzzy solution outperformed the LQR in terms of settling time but underperformed the LQR in terms of average attitude and attitude rate errors. The optimization performed by the genetic algorithm was able to find a gain schedule that increased derivative feedback gain for

Table 9 Monte Carlo analysis of LQR and fuzzy controller

	LQR		Fuzzy		99% Confidence difference in means	
Value	Mean	Standard deviation	Mean	Standard deviation	At least	At most
J	1.288×10^{-3}	3.251×10^{-4}	1.413×10^{-3}	4.866×10^{-4}	7.710×10^{-5}	1.725×10^{-4}
t_f, s	9,181	3,784	8,482	3,407	−283.8	−1, 114
θ_{ave}, deg	12.07	3.184	14.80	5.788	2.189	3.266
$\boldsymbol{\omega}_{\text{ave}}$, rad/s	2.255×10^{-4}	5.182×10^{-5}	2.488×10^{-4}	7.068×10^{-5}	1.622×10^{-5}	3.051×10^{-5}

small magnitudes of proportional error and briefly decreased proportional feedback gain for very small magnitudes of derivative error. This gain schedule resulted in a 2.62% faster settling time, a 9.37% smaller average attitude angular error, and a 3.34% larger average angular rate error compared with the LQR solution for the set of design initial conditions. The Monte Carlo analysis showed that the average settling time of the fuzzy controller was 4.73 to 18.6 min faster than that of the LQR, but the average attitude error and the average angular rate error of the fuzzy controller were statistically significantly larger than those of the LQR. Future work will investigate alternate training methods to improve robustness of the controllers to various initial conditions, as well as the use of genetic/fuzzy systems for controlling fully actuated satellite systems.

References

[1] Martel, F., Pal, P., and Psiaki, M., "Active Magnetic Control System for Gravity Gradient Stabilized Spacecraft," *Proceedings of the 2nd Annual AIAA/USU Conference on Small Satellites*, Subsystems, 1988, http://digitalcommons.usu.edu/smallsat/1988/all1988/19/.

[2] Wiśniewski, R., "Satellite Attitude Control Using Only Electromagnetic Actuation," Ph.D. Dissertation, Dept. of Control Engineering, Aalborg Univ., Aalborg, Denmark, 1996.

[3] Wiśniewski, R., "Linear Time Varying Approach to Satellite Attitude Control Using Only Electromagnetic Actuation," *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 4, 2000, pp. 640–647. doi:10.2514/2.4609

[4] Wiśniewski, R., and Markley, F. L., "Optimal Magnetic Attitude Control," *IFAC World Congress*, Beijing, July 1999.

[5] Wiśniewski, R., and Blanke, M., "Fully Magnetic Attitude Control for Spacecraft Subject to Gravity Gradient," *Automatica*, Vol. 35, No. 7, 1999, pp. 1201–1214. doi:10.1016/S0005-1098(99)00021-7

[6] Wiśniewski, R., "Sliding Mode Attitude Control for Magnetic Actuated Satellite," *Proceedings of the 14th IFAC Symposium on Automatic Control in Aerospace*, Pergamon Press, Seoul, Korea, 1998.

[7] Wiśniewski, R., and Stoustrup, J., "Periodic H_2 Synthesis for Spacecraft Attitude Control with Magnetorquers," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 5, 2004, pp. 874–881. doi:10.2514/1.10457

[8] Makovec, K. L., "A Nonlinear Magnetic Controller for Three-Axis Stability of Nanosatellites," M.S. Thesis, Virginia Polytechnic Inst. and State Univ., Blacksburg, VA, 2001.

[9] Psiaki, M. L., "Magnetic Torquer Attitude Control via Asymptotic Periodic Linear Quadratic Regulation," AIAA Paper 2000-4043, 2000.

[10] Pulecchi, T., Lovera, M., and Verga, A., "Optimal Discrete-Time Design of Three-Axis Magnetic Attitude Control Laws," *IEEE Transactions on Control Systems Technology*, Vol. 18, No. 3, 2010, pp. 714–722. doi:10.1109/TCST.2009.2024757

[11] Lovera, M., and Varga, A., "Optimal Discrete-Time Magnetic Attitude Control of Satellites," *Proceedings of the 16th International Federation of Automatic Control World Congress*, Prague, 2005, pp. 157–162.

[12] Lovera, M., De Marchi, E., and Bittanti, S., "Periodic Attitude Control Techniques for Small Satellites with Magnetic Actuators," *IEEE Transactions on Control Systems Technology*, Vol. 10, No. 1, 2002, pp. 90–95. doi:10.1109/87.974341

[13] Reyhanoglu, M., and Hervas, J. R., "Three-Axis Magnetic Attitude Control Algorithms for Small Satellites," *Proceedings of 5th International Conference on Recent Advances in Space Technologies*, IEEE Publ., Piscataway, NJ, 2011, pp. 897–902.

[14] Lovera, M., and Astolfi, A., "Global Attitude Regulation Using Magnetic Control," *Proceedings of the 40th IEEE Conference on Decision and Control*, IEEE Publ., Piscataway, NJ, 2001, pp. 4604–4609.

[15] Lovera, M., and Astolfi, A., "Global Magnetic Attitude Control of Inertially Pointing Spacecraft," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 5, 2005, pp. 1065–1072. doi:10.2514/1.11844

[16] Lovera, M., and Astolfi, A., "Global Magnetic Attitude Control of Spacecraft," *Proceedings of the 43rd IEEE Conference on Decision and Control*, IEEE Publ., Piscataway, NJ, 2004, pp. 267–272.

[17] Gravdahl, J. T., "Magnetic Attitude Control for Satellites," *Proceedings of the 43rd IEEE Conference on Decision and Control*, IEEE Publ., Piscataway, NJ, 2004, pp. 261–266.

[18] Wang, P., Shtessel, Y. B., and Wang, Y., "Satellite Attitude Control Using Only Magnetorquers," *Proceedings of 30th Southeastern Symposium on System Theory*, IEEE Publ., Piscataway, NJ, 1998, pp. 500–504.

[19] Lovera, M., and Astolfi, A., "Spacecraft Attitude Control Using Magnetic Actuators," *Automatica*, Vol. 40, No. 8, 2004, pp. 1405–1414. doi:10.1016/j.automatica.2004.02.022

[20] Kulkarni, J., and Campbell, M., "An Approach to Magnetic Torque Attitude Control of Satellites via 'H_Infinity' Control for LTV Systems," *Proceedings of the 43rd IEEE Conference on Decision and Control*, IEEE Publ., Piscataway, NJ, 2004.

[21] Wood, M., and Chen, W., "Attitude Control of Magnetically Actuated Satellites with an Uneven Inertia Distribution," *Aerospace Science and Technology*, Vol. 25, No. 1, 2013, pp. 29–39. doi:10.1016/j.ast.2011.12.005

[22] Corti, A., and Lovera, M., "Attitude Regulation for Spacecraft with Magnetic Actuators: An LPV Approach," *Control of Linear Parameter Varying Systems with Applications*, Springer, New York, 2012, pp. 339–355.

[23] Guerrant, D. V., "Design and Analysis of Fully Magnetic Control for Picosatellite Stabilization," M.S. Thesis, California Polytechnic State Univ., San Luis Obispo, CA 2005.

[24] Mohammed, A., Boudjemai, A., Chouraqui, S., and Benyettou, M., "Magnetorquer Control for Orbital Manoeuvre of Low Earth Orbit Microsatellite," *WSEAS Transactions on Communications*, Vol. 5, No. 5, 2006, pp. 947–994.

[25] Wood, M., Chen, W., and Fertin, D., "Model Predictive Control of Low Earth Orbiting Spacecraft with Magneto-Torquers," *International Conference on Control Applications, Proceedings of the 2006 IEEE*, IEEE Publ., Piscataway, NJ, Oct. 2006, pp. 2908–2913.

[26] Liang, J., Fullmer, R., and Chen, Y., "Time-Oiptimal Magnetic Attitude Control for Small Spacecraft," *43rd IEEE Conference on Decision and Control*, IEEE Publ., Piscataway, NJ, 2004, pp. 255–260.

[27] Heydari, A., "Time-Optimal Closed-Loop Attitude Control of Magnetica Actuated Satellites Using Neural Networks," M.S. Thesis, Aerospace Engineering Dept., Sharif Univ. of Technology, Tehrān, Iran, 2008.

[28] Heydari, A., Pourtakdoust, S. H., and Heydari, H., "Magnetic Attitude Control Using Fuzzy Logic," *Proceedings of the IEEE International Symposium on Intelligent Control*, IEEE Publ., Piscataway, NJ, 2009, pp. 456–460.

[29] Silani, E., and Lovera, M., "Magnetic Spacecraft Attitude Control: A Survey and Some New Results," *Control Engineering Practice*, Vol. 13, No. 3, 2005, pp. 357–371. doi:10.1016/j.conengprac.2003.12.017

[30] Bhat, S. P., "Controllability of Nonlinear Time-Varying Systems: Applications to Spacecraft Attitude Control Using Magnetic Actuation," *IEEE Transactions on Automatic Control*, Vol. 50, No. 11, Nov. 2005, pp. 1725–1735. doi:10.1109/TAC.2005.858686

[31] Shuster, M. D., "A Survey of Attitude Representations," *Journal of the Astronautical Sciences*, Vol. 41, No. 4, 1993, pp. 439–517.

[32] Wertz, J. R., and Larson, J. W., *Space Mission Analysis and Design*, 3rd ed., Springer, New York, 1999, pp. 324, 366.

[33] Vallado, D. A., and Finkleman, D., "A Critical Assessment of Satellite Drag and Atmospheric Density Modeling," *AIAA/AAS Astrodynamics Specialist Conference*, AIAA Paper 2008-6442, 2008.

[34] Wertz, J. R., *Spacecraft Attitude Determination and Control*, Reidel, Dordrecht, The Netherlands, 1978, pp. 566–570, 779–786.

[35] Finlay, C., and Nair, M., "International Geomagnetic Reference Field IGRF-11," IAGA V-MOD Geomagnetic Field Modeling, Jan. 2010, http://www.ngdc.noaa.gov/IAGA/vmod/igrf.html [accessed 10 Feb. 2013.]

[36] Vallado, D. A., "CelesTrak: Astrodynamics Software," MATLAB Source Code, CelesTrak, Jan. 2013, http://celestrak.com/software/vallado/matlab.zip [retrieved 19 Feb. 2013].

[37] Ross, T. J., *Fuzzy Logic with Engineering Applications*, 3rd ed., Wiley, West Sussex, England, U.K., 2010, pp. 1–162.

[38] Goldberg, D. E., *Genetic Algorithms: In Search, Optimization, & Machine Learning*, Addison Wesley Longman, Reading, MA, 1989, pp. 1–25.

[39] Ernest, N. D., "UAV Swarm Cooperative Control Based on a Genetic-Fuzzy Approach," M.S. Thesis, Dept. of Aerospace Engineering, College of Engineering, Univ. of Cincinnati, Cincinnati, OH, 2012.

R. Haftka
Associate Editor