

Falsification-based Verification for Multi-Mode Spacecraft Attitude Control Systems

John Matthew Brewer Panagiotis Tsiotras Kendra Lang Sean Phillips

Abstract—In this paper, an approach to the verification and validation of satellite controllers is proposed and shown to be capable of invalidating black-box controllers against prescribed system constraints. The system being tested is a rigid-body satellite featuring a hybrid controller and operating within an environment that introduces disturbances that are not perfectly known. A falsification-based approach to the verification and validation process is taken within an optimization framework. The optimizer attempts to minimize a cost function designed from the constraints given for a particular maneuver by searching over a set of initial conditions that may result in problematic trajectories. As a result of the approach taken in this paper, the optimizer is able to find counter-example initial conditions of the satellite with a hybrid controller that result in violation of the imposed constraints.

I. INTRODUCTION

As technology advances, systems involved in safety-critical applications become increasingly more complicated and difficult to predict. Because of this, the burden of analyzing system capabilities and comparing their behaviors against a set of criteria also increases, often making the practice of *formal verification* via a rigorous mathematical proof-certificate intractable. Such is the case when dealing with systems too complicated to model or which are considered to be “black-box”.

Complex critical systems such as satellites require guarantees and extremely high confidence levels capable of dealing with varying operational modes since unintended behaviors can lead to catastrophic failures, as in the case for the ASTROS-H observatory [1]. This is one of the reasons why the practical aspect of satellite control has been slow to adopt newer control techniques and, instead, tends to rely on legacy flight-proven controllers. For satellite development, verification and validation (V&V) techniques can encourage more complete and expedited approaches to the testing of higher-level control schemes while also maintaining acceptable confidence levels for the overall mission.

J. M. Brewer is a Ph.D. Candidate at the School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332. jbrewer30@gatech.edu

P. Tsiotras is the David and Andrew Lewis Chair Professor at the School of Aerospace Engineering and the Institute for Robotics & Intelligent Machines, Georgia Institute of Technology, Atlanta, GA 30332. tsiotras@gatech.edu

K. Lang is the Space and Autonomy Technical Director at Verus Research, 6100 Uptown Blvd NE, Ste 260, Albuquerque, NM 87110. Kendra.Lang@verusresearch.net

S. Phillips is a Research Mechanical Engineer at the Air Force Research Laboratory, Space Vehicles Directorate in Kirtland AFB, 87117.

Support for this work was made possible through the Air Force Small Business Technology Transfer Research (STTR) program, contract number FA9453-19-C-0621.

Approved for public release: distribution unlimited. PA: AFRL-2020-0176.

A general framework for the spacecraft control V&V problem comes with many challenges such as environmental effects which are difficult to predict, multi-mode controllers for varying operational tasks, and subsystems which are not open for inspection. There are a number of approaches to V&V in the literature, many of which are provided in the survey [2] and relate to the approach of *falsification* (i.e., the search for scenarios that lead to a violation of some safety criteria).

Optimization-based approaches have gained considerable attention in “black-box” V&V. Works such as [3], [4], [5], [6], [7], [8] approach falsification by minimizing a robustness metric generated from temporal logic-based formulations of the system constraints using Monte-Carlo, Ant Colony, Tabu, and other optimization search techniques. Falsification of hybrid flight control laws was demonstrated in [9] using Monte-Carlo Tree Search. References [10], [11] utilize a Bayesian optimization process with [10] including an approach that directly deals with high-dimensional systems using random embeddings. In [12], an RRT-inspired approach to efficient search space sampling was shown to reasonably handle high-dimensional search spaces, while [13] formulated a thorough treatment to the advantages of adaptive sampling techniques to finding falsifying scenarios with minimal sampling. Similarly to the previous works, [14] demonstrated the success of the Adaptive Las Vegas Tree Search algorithm for intelligently sampling the search-space for falsification tasks. Reference [15] presented a two-layer optimizer for falsifying hybrid systems in which local searches are performed using simple hill-climbing, while an outer optimizer uses Monte-Carlo Tree Search to guide the local search. In addition to the previous references, optimization-based academic tools for system falsification are also available. Two of the most popular tools are S-TaLiRo [16] and Breach [17].

Several other recent frameworks based on machine learning ideas also exist, but since this work focuses on an optimization-based approach, only a few of these methodologies will be mentioned here. In [18], [19], the effectiveness of *Asynchronous Advantage Actor-Critic* and *Double Deep Q Network* algorithms was utilized to find falsifying conditions, while [20] demonstrated that *Deep Deterministic Policy Gradient* could successfully train agents to generate adversarial conditions leading to collisions.

In this paper, we demonstrate the successful application of falsification towards the validation of high-level hybrid control schemes in a realistic satellite mission scenario using optimization-based ideas. Specifically, the control scheme is a multi-mode attitude controller dealing with uncertain environmental torques — a difficult controller to study tradi-

tionally. We also compare various optimization algorithms, including established ones (e.g., S-TaLiRo), when applied towards falsifying a satellite benchmark model.

II. SATELLITE BENCHMARK MODEL

A. Satellite Dynamical Model

We consider a rigid satellite with reaction wheel and thruster actuators, with a hybrid controller (serving as a benchmark case), and influenced by environmental disturbance torques. The complete dynamics of the satellite are [21]

$$\begin{aligned}\dot{\mathbf{q}} &= \mathbf{Q}\boldsymbol{\omega}, \\ J\dot{\boldsymbol{\omega}} &= -\boldsymbol{\omega} \times J\boldsymbol{\omega} - \boldsymbol{\omega} \times S\mathbf{J}_r\mathbf{w} - S\boldsymbol{\tau}_w + \boldsymbol{\tau}_e + \boldsymbol{\tau}_d, \\ \mathbf{J}_r\dot{\mathbf{w}} &= \boldsymbol{\tau}_w,\end{aligned}\quad (1)$$

where where $J \in \mathbb{R}^{3 \times 3}$ is the satellite's inertia tensor, $\boldsymbol{\omega} \in \mathbb{R}^3$ refers to the satellite's inertial angular velocity expressed in the frame of the satellite body, $\boldsymbol{\tau}_e \in \mathbb{R}^3$ refers to the external torque inputs (such as those from thrusters), $\boldsymbol{\tau}_d \in \mathbb{R}^3$ refers to the disturbance torques on the satellite, $\mathbf{q} \in \mathbb{R}^4$ represents the attitude quaternion $\mathbf{q} = q_1 + iq_2 + jq_3 + kq_4$, where $\mathbf{q} = [q_1, q_2, q_3, q_4]^T$, and $S \in \mathbb{R}^{3 \times 4}$ is a matrix mapping the four reaction wheel axes to the satellite body frame, and the quaternion kinematic matrix $\mathbf{Q}$ is given by

$$\mathbf{Q} = \frac{1}{2} \begin{bmatrix} -q_2 & -q_3 & -q_4 \\ q_1 & -q_4 & q_3 \\ q_4 & q_1 & -q_2 \\ -q_3 & q_2 & q_1 \end{bmatrix}. \quad (2)$$

Furthermore, $\mathbf{J}_r \in \mathbb{R}^{4 \times 4}$ is a diagonal matrix where the elements on the diagonal are the inertia values of each reaction wheel along its own axis, $\mathbf{w} \in \mathbb{R}^4$ is a vector containing the angular velocities of each reaction wheel with respect to the spacecraft, and $\boldsymbol{\tau}_w \in \mathbb{R}^4$ is the vector containing the torque from each reaction wheel.

B. Satellite Benchmark Controller

The controller to be tested is a hybrid controller featuring four distinct modes that handle typical operational tasks: attitude stabilization, attitude slew, satellite detumbling, and reaction wheel momentum dumping. Each mode is entered autonomously depending on the satellite state via the inter-mode transitions depicted in Figure 1. In the subsequent discussions, the controller mode will also be indicated by σ with the corresponding torques indicated using the notation $\boldsymbol{\tau}_{w\sigma}$ and $\boldsymbol{\tau}_{e\sigma}$ to refer to torques $\boldsymbol{\tau}_w$ and $\boldsymbol{\tau}_e$, respectively.

1) *Desaturation* ($\sigma = 0$): The condition to enter this mode is $\|\mathbf{w}\|_\infty \geq w_{\text{sat}}$ and the exit criterion is $\|\mathbf{w}\|_\infty \leq w_{\text{exit}}$, where w_{sat} is the saturation speed of the wheels and w_{exit} is some specified exit wheel velocity. The controller takes the form

$$\boldsymbol{\tau}_{w0} = -k_d^{\text{dump}}\mathbf{w}, \quad \boldsymbol{\tau}_{e0} = -S\mathbf{k}_d^{\text{dump}}\mathbf{w}, \quad (3)$$

with $k_d^{\text{dump}} > 0$.

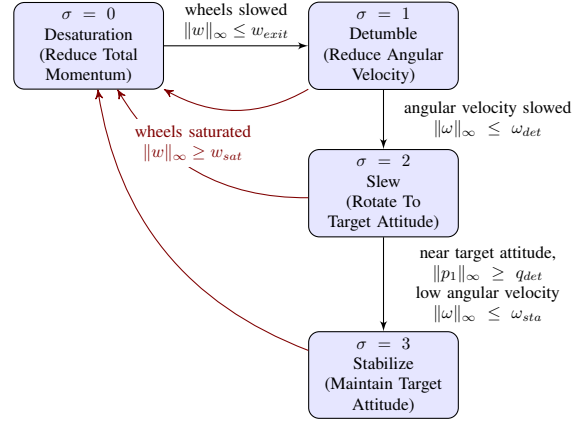


Fig. 1: Flow diagram of hybrid controller mode transitions.

2) *Detumble* ($\sigma = 1$): Detumble mode is initiated following an exit from desaturation mode. The switch from detumble to slew mode occurs once $\|\boldsymbol{\omega}\|_\infty \leq \omega_{\text{det}}$, where ω_{det} is the maximum velocity desired when entering the slew mode. The following control is used in detumble mode

$$\boldsymbol{\tau}_{w1} = S^+ k_d^{\text{det}} \boldsymbol{\omega}, \quad \boldsymbol{\tau}_{e1} = [0 \ 0 \ 0]^T, \quad (4)$$

where $S^+ \in \mathbb{R}^{3 \times 3}$ is the pseudo-inverse of S , and the control gain satisfies $k_d^{\text{det}} > 0$.

3) *Slew* ($\sigma = 2$): The attitude slew mode utilizes an eigenaxis slew controller extended from the one-dimensional slew controller described in [21]

$$\boldsymbol{\tau}_{w2} = S^+ (-\boldsymbol{\omega} \times J\boldsymbol{\omega} - \boldsymbol{\omega} \times S\mathbf{J}_r\mathbf{w} + \boldsymbol{\tau}_s), \quad (5)$$

where $\boldsymbol{\tau}_{e2} = [0 \ 0 \ 0]^T$. Here, $\boldsymbol{\tau}_s \in \mathbb{R}^3$ is the commanded slew torque in the body frame. The slew torque is created similarly to the bang-bang type of control input given in [22] and is

$$\boldsymbol{\tau}_s = \begin{cases} J\mathbf{q}_e \gamma \text{sign}(\theta) & \text{if } \theta > -\dot{\theta}|\dot{\theta}|/(2\gamma), \\ -J\mathbf{q}_e \gamma \text{sign}(\theta) & \text{otherwise,} \end{cases} \quad (6)$$

where $\mathbf{q}_e \in \mathbb{R}^3$ is the eigenaxis of rotation, $\theta \in [-\pi, \pi]$ is the angle about the eigenaxis necessary to achieve the desired rotation, and $\gamma \in \mathbb{R}$ is the commanded acceleration about $\mathbf{q}_e$. If the desired attitude is the quaternion $\mathbf{q}_d$, then if $\mathbf{p} = \mathbf{q}_d \mathbf{q}^{-1}$ (quaternion multiplication), $\mathbf{q}_e$ is the normalized vector part of $\mathbf{p}$ (i.e. $\mathbf{q}_e = \hat{\mathbf{p}}_v$, where $\mathbf{p}_v = [p_2, p_3, p_4]^T$ and $\hat{\mathbf{z}} = \mathbf{z}/\|\mathbf{z}\|$) and $\theta = 2 \cos^{-1}(p_1)$.

4) *Stabilization* ($\sigma = 3$): Stabilization mode is entered after the slew mode and once the attitude and velocity meet the criteria $p_1 \geq q_{\text{sta}}$ and $\|\boldsymbol{\omega}\|_\infty \leq \omega_{\text{sta}}$, with $1 > q_{\text{sta}} \gg 0$ and $1 \gg \omega_{\text{sta}} > 0$. The attitude stabilization mode utilizes a control law of the form

$$\boldsymbol{\tau}_{w3} = S^+ (-\boldsymbol{\omega} \times S\mathbf{J}_r\mathbf{w} + k_p \mathbf{p}_v + k_d \boldsymbol{\omega}), \quad (7)$$

with $\boldsymbol{\tau}_{e3} = [0 \ 0 \ 0]^T$ and where $k_d, k_p > 0$.

C. Disturbance Torques

Three sources of environmental disturbance torques were considered for $\boldsymbol{\tau}_d$ such that $\boldsymbol{\tau}_d = \boldsymbol{\tau}_{da} + \boldsymbol{\tau}_{dg} + \boldsymbol{\tau}_{ds}$. These are aerodynamic, gravitational, and solar radiation torques.

1) *Aerodynamic Disturbances*: The aerodynamic torque model is given as in [23]

$$\tau_{da} = \frac{\rho c_d A \|v_a\|^2}{2} \hat{v}_a \times (c_{pa} - c_g), \quad (8)$$

where $\rho \in \mathbb{R}$ is the local air density, $c_d \in \mathbb{R}$ is the satellite's coefficient of friction, $v_a \in \mathbb{R}^3$ is the relative velocity vector between the Earth's atmosphere around the satellite and the satellite, $A \in \mathbb{R}$ is the cross sectional area of the satellite perpendicular to v_a , $c_g \in \mathbb{R}^3$ is the satellite's center of gravity, and $c_{pa} \in \mathbb{R}^3$ is the satellite's center of aerodynamic pressure.

2) *Solar Radiation Pressure Disturbances*: The solar radiation pressure torque model is given as [23]

$$\tau_{ds} = \frac{F_s}{c} A_s (1 + r_q) \hat{v}_s \times (c_{ps} - c_g), \quad (9)$$

where $F_s \in \mathbb{R}$ is the Total Solar Irradiance (TSI) local to the satellite, $c \in \mathbb{R}$ is the speed of light, $v_s \in \mathbb{R}^3$ is the vector from the satellite to the Sun, $A_s \in \mathbb{R}$ is the cross sectional area of the satellite perpendicular to v_s , $r_q \in [0, 1]$ is the reflectivity of the satellite body exposed to the Sun, and $c_{ps} \in \mathbb{R}^3$ is the center of solar pressure of the satellite.

3) *Gravity-Gradient Disturbances*: The gravity-gradient torque model is given as [24]

$$\tau_{dg} = \frac{3\mu}{\|r\|^3} \hat{r} \times (Jr), \quad (10)$$

where $\mu \in \mathbb{R}$ is the gravitational parameter of Earth and $r \in \mathbb{R}^3$ is the vector from the satellite to the Earth's center.

D. Satellite Trajectory Generation and Dependencies

The satellite system dependencies consist of both the satellite's attitude and angular velocities and a set of parameters which may be considered uncertain or variable depending upon the scenario. To this end, the satellite system evolves according to the adopted notation for the dynamics

$$\dot{x}(t) = f_\sigma(x(t)), \quad (11)$$

in which $x(t) = [x_s^T(t), x_p^T]^T \in \mathbb{R}^n$ is a vector containing variables which affect the satellite trajectory, where $x_s(t) = [q^T(t), \omega^T(t), w^T(t)]^T \in \mathbb{R}^{11}$ is the state from (1) and x_p is a vector containing all uncertain parameters. Note that $\dot{x}_p = 0$ so that the uncertain parameters are assumed to be constant. In the benchmark system used here, the uncertain parameters are assumed to be $x_p = [\rho, c_d, A, A_s, F_s, r_q]^T \in \mathbb{R}^6$ and have a known minimum and maximum value. The scenario in this paper therefore corresponds to $n = 17$.

The system (11) can be modeled as a switching hybrid system as in [25], where the subscript σ is the discrete switching signal state indicating the current mode of the controller. This hybrid system is depicted in Figure 1.

E. Satellite Constraints

A field-of-view constraint is considered in which the z -body axis of the satellite (v_c as expressed in the body frame) must not come within a specified angle θ_f of some inertial vector v_s . Practically speaking, v_c may represent the bore-axis of a camera, v_s a vector pointing from the satellite to

the Sun, and θ_f is related to half the field-of-view of the camera. This constraint may be expressed at any time t as

$$\Delta\theta(x(t)) = \cos^{-1}(v_s^T R(x(t)) v_c) - \theta_f \geq 0, \quad (12)$$

where $R(x(t))$ is the rotation matrix that takes elements from the body frame to the inertial frame and depends upon q . The constraint (12) represents the smallest angle between v_s and the edge of the keep-out zone. When $\Delta\theta(x(t)) = 0$, the angle between v_s and v_c at t is exactly θ_f . For $\Delta\theta(x(t)) < 0$, $\angle(v_s, v_c) < \theta_f$ and the constraint is violated.

III. FALSIFICATION APPROACH TO VERIFICATION AND VALIDATION

The verification and validation of the hybrid controller is performed using falsification, where counterexamples are used to disprove the controller's ability to prevent the satellite from violating constraints. Because the satellite controller is intended to provide autonomous capabilities, the search space to find these worst-case scenarios is limited to the satellite's initial conditions and parameters which may not be well known. Accordingly, constraint violations are evaluated along the resulting trajectory of the satellite.

A. Optimization Formulation

The optimization problem is formulated as follows.

$$\begin{aligned} \min_y \quad & c(y) \\ \text{s.t.} \quad & g(y) \geq 0, \\ & h(y) = 0. \end{aligned} \quad (13)$$

Here, the point $y \in \mathbb{R}^m$ in the optimization search space is mapped into a point $x \in \mathbb{R}^n$ in the state-space of the satellite by a user-defined function $u : \mathbb{R}^m \rightarrow \mathbb{R}^n$ which may be beneficial depending upon x or the optimizer's search behavior. Letting $\varphi(x(0), t) = x(t)$ represent the flow of $f(\cdot)$ in (11) starting at $x(0)$, the objective function is defined as

$$c(y) = \min_{0 \leq t \leq T} C(\varphi(u(y), t)), \quad (14)$$

and is the minimum value of $C(\cdot)$ along the satellite's trajectory beginning at the initial condition $x(0) = u(y)$ and ending at $\varphi(x(0), T)$. In (14), $C(\cdot)$ is a generic scalar function that measures how close the satellite is to violating the constraints at time t . The inequality constraint(s) $g(y) \geq 0$ (and equality constraint(s) $h(y) = 0$) limit the optimization space to correspondingly feasible values for $x(0)$.

For the purposes of finding counterexample trajectories, $c(y)$ is defined such that the $c(y) < 0$ corresponds to a violation. In this case, any early-termination of the optimizer when $c(y) < 0$ is sufficient for system invalidation.

B. Objective Function Formulation

Since the falsification-based approach seeks *any* constraint violation in order to determine whether the controller is safe to use, optimization over the worst objective function is appropriate when considering multiple satellite constraints. Therefore, if $C_i(\cdot)$, $i = 1, 2, \dots, M$, is a collection of functions to optimize over, then $C(\cdot)$ may be chosen as

$$C(x(t)) = \min_{i \in \{1, 2, \dots, M\}} C_i(x(t)). \quad (15)$$

Hence C is taken to consist of the satellite's constraints, for example, $C_1(x(t)) = \Delta\theta(x(t))$, etc. In the presence of multiple satellite constraints, because we are in search of any violation, the optimization may alternatively be performed over each satellite constraint separately.

In the V&V community, the use of metric temporal logic (MTL) statements to formalize constraints has become popular. Certain metrics, such as those described in [26] and [27], allow the conversion of MTL statements into *robustness* values that can then be used as the optimization variable $c(y)$ in our approach, as done in [28]. Therefore, the proposed approach can incorporate MTL statements as long as they have been converted into robustness statements. While adhering to certain robustness metrics appears to limit the functions that can be used for $c(y)$ in (14), the approach works well in practice.

IV. VERIFICATION AND VALIDATION RESULTS

A. Example Use of the Falsification Approach

The optimizer chosen to solve (13) in this example is the Nelder-Mead algorithm in [29]. The algorithm was modified to handle constrained optimization problems using *Box's method* [30], and a simple random restart of the initial guess is performed if it appears stuck in a local minimum.

The satellite parameters for the example were $J = \text{diag}(1000, 700, 500) \text{ kg m}^2$, $J_r = \text{diag}(0.2, 0.2, 0.2, 0.2) \text{ kg m}^2$, $S = \begin{bmatrix} -1 & 0 & 0 & 1/\sqrt{3} \\ 0 & -1 & 0 & 1/\sqrt{3} \\ 0 & 0 & -1 & 1/\sqrt{3} \end{bmatrix}$, $c_g = [1.25, 1.75, 2.5]^T \text{ m}$, $c_{ps} = [1.5, 2.1, 3]^T \text{ m}$, $c_{pa} = [1.5, 2.1, 3]^T \text{ m}$, $k_d^{det} = 50$, $\gamma = 0.004$, $k_p = 5$, $k_d = 50$, $k_d^{dimp} = 0.012$, $\omega_{det} = 0.0002 \text{ rad/s}$, $q_{sta} = 0.98$, $\omega_{sta} = 0.04 \text{ rad/s}$, $w_{sat} = 600 \text{ rad/s}$, $w_{exit} = 30 \text{ rad/s}$, v_a is considered the same as the satellite velocity's magnitude, $v_c = [0, 0, 1]^T$, and $\theta_f = 5^\circ$. The satellite is in a circular orbit around Earth. The semi-major axis is 6978.1 km, eccentricity is 0, inclination is 35° , longitude of ascending node is 230° , the argument of periapsis is 270° , and the true anomaly is 135° . Additionally, the epoch at $t = 0$ is the date February 7th, 2020 and time 7pm GMT.

The desired quaternion in this example is updated in real time so that v_c is pointed towards a point fixed on the Earth's surface (denoted with r_s) when the relative quaternion p becomes identity. The desired attitude q_d is calculated onboard the satellite as $q_d(t) = [\cos(\theta_d(t)/2), \sin(\theta_d(t)/2)r_d(t)]^T$ where $\theta_d(t) = \cos^{-1}(v_c^T \hat{r}_b(t))$, $r_d(t) = \hat{r}_b(t) \times \hat{v}_c$, and $r_b(t) = r_s(t) - r(t)$. The intended target on Earth for v_c located by r_s is the Sequoia National Forest, with latitude 36.3° North and longitude 118.5° West.

The elements in $x(t)$ were given the initial guess of $q(0) = [0.651, 0, -0.759, 0]^T$, $\omega(0) = [0, 0, 0]^T$, $w = [0, 0, 0, 0]^T$, $\rho = 2.778 \times 10^{-13} \text{ kg/m}^3$, $c_d = 2$, $A = 17.5 \text{ m}^2$, $A_s = 17.5 \text{ m}^2$, $F_s = 1362 \text{ W/m}^2$, and $r_q = 0.6$.

Each element of $x(t)$ is considered independent with the exception of $q(t)$. As an attitude quaternion, $q(t)$ must obey $\|q(t)\| \equiv 1$, or equivalently, $q_v^T q_v \leq 1$, where $q_v = [q_2, q_3, q_4]^T$. Incorporating the first expression into $u(y)$ and the second expression into $g(y) \geq 0$, we can omit searching

over q_1 and reduce the dimensionality of the search space. Thus, in letting $y_q = [y_1, y_2, y_3]^T$, we define the following for the optimization problem formulation

$$u(y) = x(0) = \begin{bmatrix} \sqrt{(1 - y_q^T y_q)} \\ y \end{bmatrix}. \quad (16)$$

This choice of $u(y)$ then defines the search space y as the elements of $x(0)$ excluding the scalar part of the attitude quaternion $q(0)$, that is, $y = [q_v^T(0), \omega^T(0), w^T(0), \rho, c_d, A, A_s, F_s, r_q]^T$. We also define the inequality constraints $g(y)$ so that the following limitations apply: $c(y) \geq 0$, $y_q^T y_q \leq 1$, $\| [y_4 \ y_5 \ y_6]^T \|_\infty \leq 0.1$, $\| [y_7 \ y_8 \ y_9 \ y_{10}]^T \|_\infty \leq 400$, $6.55 \times 10^{-14} \leq y_{11} \leq 4.9 \times 10^{-13}$, $1 \leq y_{12} \leq 2.5$, $10 \leq y_{13} \leq 18$, $10 \leq y_{14} \leq 18$, $1360 \leq y_{15} \leq 1367$, and $0.2 \leq y_{16} \leq 0.7$ in order to bound the feasible space and ensure no violation occurs at $t = 0$. The constraints $h(y) = 0$ are not used.

Finally, the objective function is set such that

$$C(x(t)) = \Delta\theta(x(t)). \quad (17)$$

The falsification approach was able to find an $x(0)_s$ and x_p for which the use of the controller led the spacecraft into a violating state. The initial condition values found were $q(0) = [0 \ -0.249 \ -0.9803 \ 0.1957]^T$, $\omega(0) = [0.0051 \ -0.0065 \ 0.0134]^T \text{ rad/s}$, $w(0) = [0.0442 \ -0.0086 \ 0.0536 \ -0.0068]^T \text{ rad/s}$, and the uncertain parameters found were $\rho = 4.733 \times 10^{-13} \text{ kg/m}^3$, $c_d = 2.265$, $A = 10 \text{ m}^2$, $A_s = 10 \text{ m}^2$, $F_s = 1360.4 \text{ W/m}^2$, $r_q = 0.226$, with the violation constraint (17) occurring during the stabilization mode of the controller as is visible in Figure 2 and depicted in the simulation snapshots shown in Figure 3.

B. Performance Survey of the Falsification Approach

To gain an idea of the proposed approach's success as it relates to the satellite model, a comparison of 100 different runs of the algorithm was performed using various optimization tools. The optimizers implemented for this comparison are the Nelder-Mead, Particle Swarm, and Bayesian Optimization algorithms, as well as the tool S-TaLiRo. Using the preceding simulation and values, each run was initialized with randomized values for y .

The number of simulated trajectories allowed for the falsification task was limited to a maximum number of 200, which if reached before the discovery of a violation, the run was considered unsuccessful. This limit was imposed due to both the practical time constraint that comes with computationally heavier models or interacting with hardware and because the falsification problem naturally cannot validate systems with absolute certainty, and instead, can only invalidate them (there is no way to know whether after 1000 iterations if the 1001 iteration will reveal a violation).

Although falsification can only invalidate a system, there is a general correlation between the number of trajectories checked and the level of trust in the correct operation of the system if the falsifier is skilled at finding violations. When

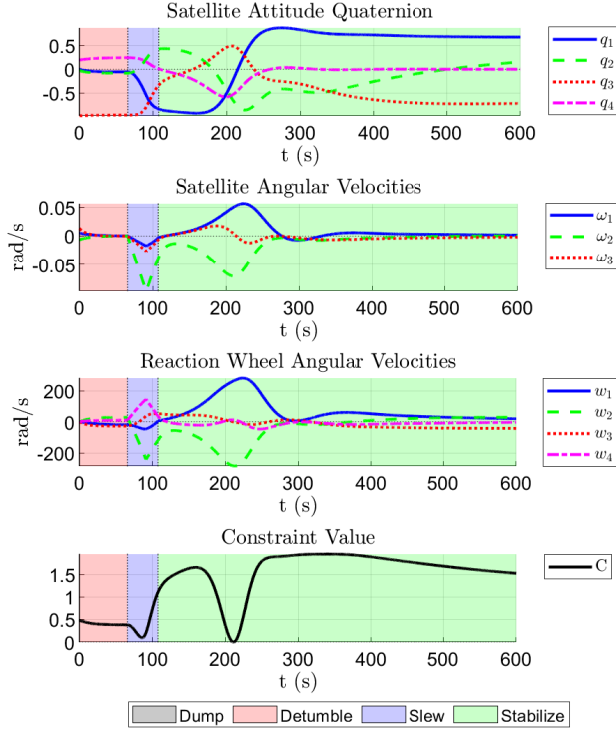


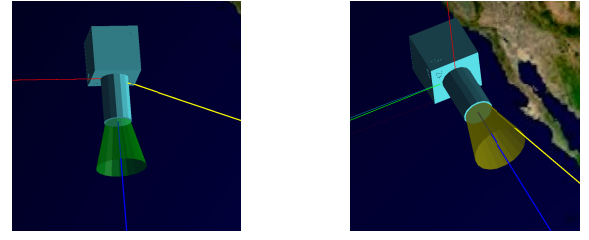
Fig. 2: Trajectory satellite starting at violating initial conditions with violating uncertain parameters found from the optimizer. A $\Delta\theta$ violation is reached at $t = 211$ s where $c(y) = C(x(211)) = -0.005$.

TABLE I: Comparison results showing the percentage of successful falsifications of each algorithm within the allotted 200 simulations.

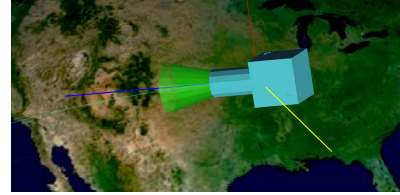
Algorithm/Tool	Falsification rate
Nelder-Mead Optimization	87%
Particle Swarm Optimization	99%
Bayesian Optimization	100%
S-TaLiRo	99%

the number of available trajectories is limited, a falsifier that consistently discovers violations with a relatively small number of system simulations leads to more confidence in the system than using a falsifier that varies widely in the number of simulations needed.

The success rate of the different optimizers to falsify the 17-dimensional satellite model is shown in Table I. A histogram of the number of satellite trajectories generated before finding the falsification during each run is given in Figure 4 and illustrates the relative efficiency of each algorithm. In this case, the proposed framework with a Bayesian Optimization scheme outperformed the other optimizers tested as well as the tool S-TaLiRo in terms of iteration count and success rate. This is demonstrated by the majority of runs requiring comparatively fewer trajectories and having lower variability in the number of trajectories needed, and this appears to indicate that Bayesian optimization in this framework is more reliable in finding violating conditions.



(a) $t = 0$ at violating initial (b) $t = 211$ when FOV violation was found by optimizer.



(c) $t = 600$ when satellite is pointing towards Sequoia National Forest.

Fig. 3: Simulation snapshots depicting the violation. v_c and v_s are the blue and yellow vectors, respectively, and the green (yellow) cone represents an unviolated (violated) FOV cone.

V. CONCLUSIONS

In this paper, we demonstrated the use of an automated falsification framework for the validation of a multi-mode satellite system operating autonomously. The optimization-based falsification approach proposed was successful in finding satellite constraint violations and invalidated a test multi-mode hybrid controller by treating the satellite controller as a “black-box”. By preserving a generic “black-box” optimization-based approach, the proposed falsification approach allows significant expression of the cost function and optimization space for complicated states and validation criteria as well as free selection of an optimizer most appropriate for the search space and cost function used. Searching over the satellite’s initial state and a number of uncertain parameters, the method was able to invalidate a 17-dimensional satellite model within a generally reasonable number of simulations.

The proposed framework was also compared to the tool S-TaLiRo using the satellite model benchmark and various optimizers. Overall, a Bayesian Optimizer, when used within the proposed framework, demonstrated the most efficient and highest falsification rate out of the methods compared.

ACKNOWLEDGMENT

The authors would like to thank Kelsey Hawkins for helping develop parts of the spacecraft model and hybrid control law we use in this paper.

REFERENCES

- [1] “Hitomi experience report: Investigation of anomalies affecting the X-ray astronomy satellite “Hitomi” (ASTRO-H),” June 8 2016. [Online]. Available: https://global.jaxa.jp/projects/sat/astro_h/files/topics_20160608.pdf
- [2] A. Corso, R. J. Moss, M. Koren, R. Lee, and M. J. Kochenderfer, “A survey of algorithms for black-box safety validation,” 2020, *arXiv:abs/2005.02979*.

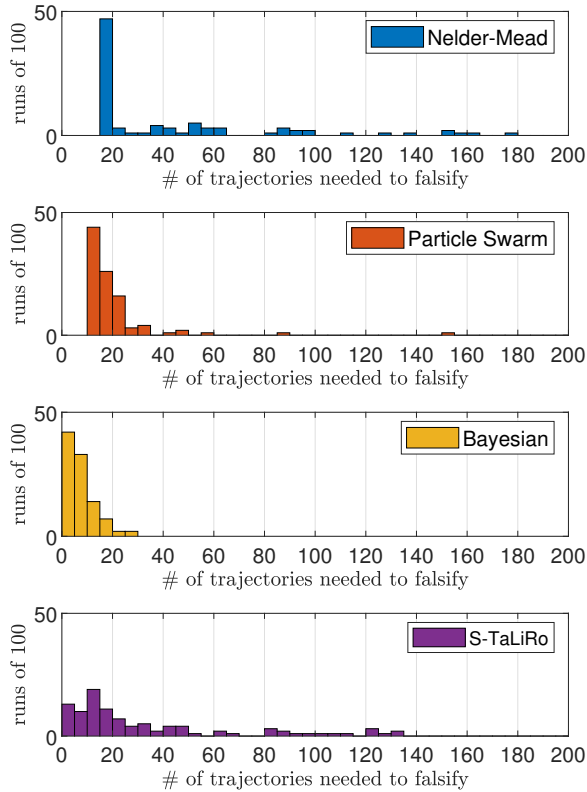


Fig. 4: Histograms of the number of simulated trajectories needed to falsify the satellite controller.

- [3] H. Abbas, G. Fainekos, S. Sankaranarayanan, F. Ivančić, and A. Gupta, "Probabilistic temporal logic falsification of cyber-physical systems," *ACM Trans. Embed. Comput. Syst.*, vol. 12, no. 2s, May 2013. [Online]. Available: <https://doi.org/10.1145/2465787.2465797>
- [4] A. Adimoolam, T. Dang, A. Donzé, J. Kapinski, and X. Jin, "Classification and coverage-based falsification for embedded control systems," in *Computer Aided Verification*, R. Majumdar and V. Kunčak, Eds. Springer International Publishing, 2017, pp. 483–503.
- [5] A. Aerts, B. Tong Minh, M. R. Mousavi, and M. A. Reniers, "Temporal logic falsification of cyber-physical systems: An input-signal-space optimization approach," in *IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, Vasteras, Sweden, 2018, pp. 214–223.
- [6] Y. S. R. Annapureddy and G. E. Fainekos, "Ant colonies for temporal logic falsification of hybrid systems," in *36th Annual Conference on IEEE Industrial Electronics Society*, Glendale, Arizona, USA, 2010, pp. 91–96.
- [7] J. Deshmukh, X. Jin, J. Kapinski, and O. Maler, "Stochastic local search for falsification of hybrid systems," in *Automated Technology for Verification and Analysis*, B. Finkbeiner, G. Pu, and L. Zhang, Eds. Springer International Publishing, 2015, pp. 500–517.
- [8] L. Mathesen, S. Yaghoubi, G. Pedrielli, and G. Fainekos, "Falsification of cyber-physical systems with robustness uncertainty quantification through stochastic optimization with adaptive restart," in *IEEE 15th International Conference on Automation Science and Engineering (CASE)*, Vancouver, BC, Canada, 2019, pp. 991–997.
- [9] R. Delmas, T. Loquen, J. Boada-Bauxell, and M. Carton, "An evaluation of monte-carlo tree search for property falsification on hybrid flight control laws," in *Numerical Software Verification*, M. Zamani and D. Zufferey, Eds. Springer International Publishing, 2019, pp. 45–59.
- [10] J. Deshmukh, M. Horvat, X. Jin, R. Majumdar, and V. S. Prabhu, "Testing cyber-physical systems through bayesian optimization," *ACM Transactions on Embedded Computing Systems*, vol. 16, no. 5s, Sept. 2017. [Online]. Available: <https://doi.org/10.1145/3126521>
- [11] Y. Abeyirigoonawardena, F. Shkurti, and G. Dudek, "Generating adversarial driving scenarios in high-fidelity simulators," in *International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada, 2019, pp. 8271–8277.
- [12] T. Nahhal and T. Dang, "Test coverage for continuous and hybrid systems," in *Computer Aided Verification*, W. Damm and H. Hermanns, Eds. Springer Berlin Heidelberg, 2007, pp. 449–462.
- [13] G. E. Mullins, P. G. Stankiewicz, R. C. Hawthorne, and S. K. Gupta, "Adaptive generation of challenging scenarios for testing and evaluation of autonomous vehicles," *Journal of Systems and Software*, vol. 137, pp. 197 – 215, 2018.
- [14] G. Ernst, S. Sedwards, Z. Zhang, and I. Hasuo, "Fast falsification of hybrid systems using probabilistically adaptive input," in *Quantitative Evaluation of Systems*, D. Parker and V. Wolf, Eds. Springer International Publishing, 2019, pp. 165–181.
- [15] Z. Zhang, G. Ernst, S. Sedwards, P. Arcaini, and I. Hasuo, "Two-layered falsification of hybrid systems guided by Monte Carlo tree search," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 11, pp. 2894–2905, 2018.
- [16] Y. Annapureddy, C. Liu, G. Fainekos, and S. Sankaranarayanan, "S-taliro: A tool for temporal logic falsification for hybrid systems," in *Tools and Algorithms for the Construction and Analysis of Systems*, P. A. Abdulla and K. R. M. Leino, Eds. Springer Berlin Heidelberg, 2011, pp. 254–257.
- [17] A. Donzé, "Breach, a toolbox for verification and parameter synthesis of hybrid systems," in *Computer Aided Verification*, T. Touili, B. Cook, and P. Jackson, Eds. Springer Berlin Heidelberg, 2010, pp. 167–170.
- [18] T. Akazaki, S. Liu, Y. Yamagata, Y. Duan, and J. Hao, "Falsification of cyber-physical systems using deep reinforcement learning," *Lecture Notes in Computer Science*, pp. 456–465, 2018. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-95582-7_27
- [19] Y. Yamagata, S. Liu, T. Akazaki, Y. Duan, and J. Hao, "Falsification of cyber-physical systems using deep reinforcement learning," *IEEE Transactions on Software Engineering*, 2020, doi:10.1109/TSE.2020.2969178.
- [20] V. Behzadan and A. Munir, "Adversarial reinforcement learning framework for benchmarking collision avoidance mechanisms in autonomous vehicles," *IEEE Intelligent Transportation Systems Magazine*, 2019, doi: 10.1109/ITS.2019.2898964.
- [21] J. L. Junkins and J. D. Turner, *Optimal Spacecraft Rotational Maneuvers*. Amsterdam: Elsevier, 1986.
- [22] K. Bilimora and B. Wie, "Time-optimal three-axis reorientation of a rigid spacecraft," *Journal of Guidance, Control, and Dynamics*, vol. 16, no. 3, pp. 446–460, 1993.
- [23] J. R. Wertz and W. J. Larson, *Space Mission Analysis and Design*, 3rd ed. 4940 W. 147th Street, Hawthorn, CA 90250 USA: Microcosm Press and Springer, 1999.
- [24] L. Markley and J. Crassidis, *Fundamentals of Spacecraft Attitude Determination and Control*. 233 Spring Street, New York, NY 10013 USA: Springer, 2014.
- [25] R. Goebel, R. G. Sanfelice, and A. R. Teel, *Hybrid Dynamical Systems*. 41 William Street, Princeton, New Jersey: Princeton University Press, 2012.
- [26] G. E. Fainekos and G. J. Pappas, "Robustness of temporal logic specifications for continuous-time signals," *Theoretical Computer Science*, vol. 410, no. 42, pp. 4262–4291, 2009. [Online]. Available: <https://doi.org/10.1016/j.tcs.2009.06.021>
- [27] N. Mehdipour, C. Vasile, and C. Belta, "Average-based robustness for continuous-time signal temporal logic," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 5312–5317.
- [28] S. Yaghoubi and G. Fainekos, "Local descent for temporal logic falsification of cyber-physical systems," in *Cyber Physical Systems. Design, Modeling, and Evaluation*, R. Chamberlain, W. Taha, and M. Törnren, Eds. Springer International Publishing, 2019, pp. 11–26.
- [29] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the Nelder-Mead simplex method in low dimensions," *SIAM Journal of Optimization*, vol. 9, no. 1, pp. 112–147, 1998.
- [30] F. Le Floch, "Issues of Nelder-Mead simplex optimisation with constraints," *SSRN Electronic Journal*, 01 2012.