Journal of
Applied Remote Sensing

RemoteSensing.SPIEDigitalLibrary.org

State-dependent Riccati equation controller using Java in remote sensing CubeSats

Alessandro G. Romero
Luiz C. G. de Souza

SPIE.

Alessandro G. Romero, Luiz C. G. de Souza, "State-dependent Riccati equation controller using Java in remote sensing CubeSats," *J. Appl. Remote Sens.* **13**(3), 032509 (2019), doi: 10.1117/1.JRS.13.032509.

State-dependent Riccati equation controller using Java in remote sensing CubeSats

Alessandro G. Romero[a,*] **and Luiz C. G. de Souza**[b]
[a]Brazilian National Institute for Space Research, Space Mechanics and Control,
São José dos Campos, Brazil
[b]Federal University of ABC, São Bernardo do Campo, Brazil

Abstract. STRaND and PhoneSat programs have attracted the attention of the aerospace community by applying, in CubeSats, commercial off-the-shelf smartphones based on Google's Android. In Android, the development commonly applies Java hence this language is a natural candidate for the attitude and orbit control subsystem (AOCS). Moreover, such AOCS can be designed with success by linear control theory; however, the linearized models are not able to represent all the effects of the nonlinear terms present in the dynamics. Therefore, nonlinear control techniques can yield better performance. An example is the Nano-Satellite Constellation for Environmental Data Collection, used as the reference in this work, a set of remote sensing CubeSats from the Brazilian National Institute for Space Research, in which the AOCS must stabilize the satellite in three-axes. We present the investigation of a state-dependent Riccati equation (SDRE) controller, a nonlinear controller, based on attitude errors given by quaternions. The investigation uses Java, accordingly, it can run on an Android operating system in a CubeSat, and it has low cost. Two controllers (linear and SDRE) were evaluated using a Monte Carlo perturbation model. The initial results show that the SDRE controller provides better performance. © *2019 Society of Photo-Optical Instrumentation Engineers (SPIE)* [DOI: 10.1117/1.JRS.13.032509]

Keywords: CubeSat; state-dependent Riccati equation; Java; nonlinear; attitude.

Paper 190111SS received Feb. 16, 2019; accepted for publication Jun. 25, 2019; published online Jul. 17, 2019.

1 Introduction

In 2013, the STRaND (University of Surrey and Surrey Satellite Technology Ltd)[1] and the PhoneSat (NASA)[2] programs attracted attention of the aerospace community by applying commercial off-the-shelf smartphones in CubeSats. Both programs deployed CubeSats using smartphones based on Google's Android, in which application development is mainly based on Java programming language. Some of these CubeSats had actuators, e.g., STRaND-1 had three reaction wheels mounted in an orthogonal configuration to provide three-axis control, whereas PhoneSat 2.0 beta had magnetorquers to de-tumble the spacecraft. Taking into account a CubeSat that runs an Android operating system (based on a smartphone), it is natural to evaluate the attitude and orbit control subsystem (AOCS) based on Java.

Furthermore, the design of a satellite AOCS that involves plant uncertainties, large angle maneuvers, and fast attitude control following a stringent pointing, requires nonlinear control methods in order to satisfy performance and robustness requirements.

An example is the Nano-satellite Constellation for Environmental Data Collection (CONASAT),[3,4] a set of remote sensing CubeSats from the Brazilian National Institute for Space Research (INPE), in which the AOCS must stabilize the satellite in three-axes in order to maximize the receiving of environment data sent by platforms in the Brazilian territory.

One candidate method for a nonlinear AOCS controller is the state-dependent Riccati equation (SDRE) method, originally proposed in Ref. 5 and then explored in detail in Refs. 6–8. SDRE provides an effective algorithm for synthesizing nonlinear feedback control by allowing

*Address all correspondence to Alessandro G. Romero, E-mail: romgerale@yahoo.com.br

1931-3195/2019/$28.00 © 2019 SPIE

nonlinearities in the system states while offering great design flexibility through state-dependent weighting matrices. The SDRE can be considered as the nonlinear counterpart of linear-quadratic regulator (LQR) control method.[6–8] SDRE is based on the arrangement of the system model in a form known as state-dependent coefficient (SDC) matrices.

Accordingly, a suboptimal control law is carried out by a real-time solution of an algebraic Riccati equation (ARE) using the SDC matrices by means of a numerical algorithm. Therefore, SDRE linearizes the plant about the instantaneous point of operation and produces a constant state-space model of the system. The process is repeated in the next sampling steps, producing and controlling several state-dependent linear models out of a nonlinear one.

In this paper, we present the investigation of an SDRE controller performance by simulations. The simulator is implemented based on Java and related open-source software libraries (Hipparchus—linear algebra library and Orekit—flight dynamics library); therefore, it can run in a variety of platforms, including an Android operating system in a CubeSat, and it has low cost. The Hipparchus open-source library was extended in order to solve the ARE equation that is the cornerstone of the SDRE method, a major contribution of the simulator.

The following related topics are beyond the scope of this paper: orbital dynamics simulation[9–11] and attitude estimation based on noisy sensor measurements.[4,9,10]

A good survey of the SDRE method can be found in Ref. 6 and its systematic application to deal with a nonlinear plant in Ref. 7. The SDRE method was applied in Refs. 12–18 for controlling a nonlinear system similar to the six-degree of freedom satellite model considered in this paper. Reference 12 defined a simulator using Euler angles based on commercial software, whereas Ref. 15 applied quaternions on commercial software. Reference 17 extended an opensource project and defined an opensource simulator based on Euler angles. Finally, regarding CONASAT, Ref. 4 applied the SDRE as a filter technique together with a proportional-integral-derivative (PID) controller while Ref. 18 compared performance and robustness of an SDRE with a PID controller.

The application of SDRE method and, consequently, the ARE problem that arises, have already been studied in the available literature, e.g., Ref. 14 investigated the approaches for the ARE solving as well as the resource requirements for such online solving. In fact, the intensive resource requirements for the online ARE solving is the major drawback of SDRE. Nonetheless, the SDRE method has three major advantages: (a) simplicity, (b) numerical tractability, and (c) flexibility for the designer, being comparable to the flexibility in the LQR.[15]

To the best of our knowledge, neither a Java implementation for an ARE solver nor a Java open-source option for simulation of SDRE controllers are available. Hence, those products constitute major contributions of this work, which can be deployed in an Android operating system in a CubeSat focused on remote sensing. An additional contribution of this work is the explicit modeling of the state-space model for a three-axes stabilized attitude-maneuvering satellite using quaternions for synthesizing feedback control.

The simulator is evaluated using the parameters of CONASAT[4] and two AOCS controllers (PID and SDRE); the results of the Monte Carlo perturbation model shown that SDRE-based controller provides better performance in the set of scenarios considered.

This paper is organized as follows. In Sec. 2, the problem description is presented. In Sec. 3, the satellite physical modeling in the simulator is reviewed. In Sec. 4, we explore the state-space models and the controllers. In Sec. 5, we share simulation results. Finally, the conclusions are shared in Sec. 6.

2 State-Dependent Riccati Equation Technique

The SDRE technique entails factorization (that is, parametrization) of the nonlinear dynamics into the state vector and the product of a matrix-valued function that depends on the state itself. In doing so, SDRE brings the nonlinear system to a (nonunique) linear structure having SDC matrices given as

$$\dot{\vec{x}} = A(\vec{x})\vec{x} + B(\vec{x})\vec{u} \quad \vec{y} = C\vec{x}, \tag{1}$$

where $\vec{x} \in \mathbb{R}^n$ is the state vector and $\vec{u} \in \mathbb{R}^m$ is the control vector. Notice that the SDC form has the same structure as a linear system but with the system matrices, A and B, being functions of the state vector. The nonuniqueness of the SDC matrices creates extra degrees of freedom, which can be used to enhance controller performance; however, it poses challenges since not all SDC matrices fulfill the SDRE requirements, e.g., the pair (A, B) must be pointwise stabilizable.

The system model in Eq. (1) is subject of the cost functional described as

$$J(\vec{x}_0, \vec{u}) = \frac{1}{2}\int_0^\infty [\vec{x}^T Q(\vec{x})\vec{x} + \vec{u}^T R(\vec{x})\vec{u}]\mathrm{d}t, \tag{2}$$

where $Q(\vec{x}) \in \mathbb{R}^{n\times n}$ and $R(\vec{x}) \in \mathbb{R}^{m\times m}$ are the state-dependent weighting matrices. In order to ensure local stability, $Q(\vec{x})$ is required to be positive semidefinite for all $\vec{x}$, and $R(\vec{x})$ is required to be positive for all $\vec{x}$.[14]

The SDRE controller linearizes the plant about the current operating point and creates constant state space matrices so that the LQR method can be used. This process is repeated in all samplings steps, resulting in a pointwise linear model from a nonlinear model, so that an ARE is solved and a control law is computed also in each step. Therefore, according to LQR theory and Eqs. (1) and (2), the state-feedback control law in each sampling step is $\vec{u} = -K(\vec{x})\vec{x}$ and the state-dependent gain $K(\vec{x})$ is obtained as[7]

$$K(\vec{x}) = R^{-1}(\vec{x})B^T(\vec{x})P(\vec{x}), \tag{3}$$

where $P(\vec{x})$ is the unique, symmetric, positive-definite solution of the algebraic SDRE given as[7]

$$P(\vec{x})A(\vec{x}) + A^T(\vec{x})P(\vec{x}) - P(\vec{x})B(\vec{x})R^{-1}(\vec{x})B^T(\vec{x})P(\vec{x}) + Q(\vec{x}) = 0. \tag{4}$$

Considering that Eq. (4) is solved in each sampling step, it is reduced to an ARE. Finally, the conditions for the application of the SDRE technique in a given system model are:[7]

1. $A(\vec{x}) \in C^1(\mathbb{R}^w)$;
2. $B(\vec{x})$, $C(\vec{x})$, $Q(\vec{x})$, $R(\vec{x}) \in C^0(\mathbb{R}^w)$;
3. $Q(\vec{x})$ is a positive semidefinite and $R(\vec{x})$ is a positive definite;
4. $A(\vec{x})x \Rightarrow A(0)0 = 0$, i.e., the origin is an equilibrium point;
5. pair(A, B) is pointwise stabilizable (a sufficient test for stabilizability is to check the rank of controllability matrix);
6. pair$(A, Q^{\frac{1}{2}})$ is pointwise detectable (a sufficient test for detectability is to check the rank of observability matrix).

3 Satellite Physical Modeling

The simulator is designed based on a typical mission developed by INPE, in which the AOCS must stabilize a satellite in three-axes so that the optical payload can point to the desired target.

Since an active control system is required, the simulator uses momentum exchange actuators. The type of momentum exchange actuator used is reaction wheel, a rotating machine that is commonly applied for very accurate control and for moderately fast maneuvers since it allows continuous and smooth control with the lowest possible disturbing torques.[10] The basic technical features of a reaction wheel are maximum torque, maximum momentum storage capacity (or maximum angular velocity), torque noise, and friction.[10] The reaction wheel provided in the simulator models the first two technical features.

Focusing on the sensors, there are two principal types of attitude determination hardware: attitude sensors and angular velocity sensors.[10] The simulator models these two types of sensors: (1) a set of attitude sensors, the set of Sun sensors (quite-common on Earth-orbiting satellites[10]), and (2) an angular velocity sensor, a gyroscope. The sensors, available in the simulator, are ideal and simplified, in the sense that, they can read the physical quantities at any moment with perfect accuracy and no noise. In addition, the set of Sun sensors provides through the entire simulation the measure of the Sun versor in the body-fixed frame ($\hat{s}_b$) so there is no eclipse, the Sun is not moving in the Earth-centered inertial (ECI) reference frame

($\hat{s}_i = [0.323116\ 0.868285\ 0.376401]^T$) and the Sun is always visible by each individual sensor. Indeed, ECI is a quasiinertial reference frame generally used in AOCS.[9,10]

Next sections explore the kinematics and the rotational dynamics of the satellite attitude available in the simulator.

3.1 *Kinematics*

Given the ECI reference frame ($\mathfrak{F}_i$) and the frame defined in the satellite with origin in its center of mass (the body-fixed frame, $\mathfrak{F}_b$), then a rotation $R \in SO(3)$ ($SO(3)$ is the set of all attitudes of a rigid body described by 3×3 orthogonal matrices whose determinant is one) represented by an unit quaternion $Q = [q_1\, q_2\, q_3 | q_4]^T$ can define the attitude of the satellite.

Defining the angular velocity $\vec{\omega} = [\omega_1\, \omega_2\, \omega_3]^T$ of $\mathfrak{F}_b$ with respect to $\mathfrak{F}_i$ measured in the $\mathfrak{F}_b$, the kinematics can be described as[9]

$$\dot{Q} = \frac{1}{2}\Omega(\vec{\omega})Q = \frac{1}{2}\Xi(Q)\vec{\omega} \quad \Omega(\vec{\omega}) \triangleq \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix}, \tag{5}$$

where the unit quaternion Q satisfies the following identity:

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1. \tag{6}$$

Equation (5) describes the satellite's attitude based on the initial attitude and the history of the change in the angular velocity [$\dot{Q} = F(\omega, t)$]. Nonetheless, it is worthy to mention that although the definition of the unit quaternion is global in the sense that it can represent all attitudes, each physical attitude $R \in SO(3)$ is represented by a pair of unit quaternions $\pm Q \in \mathbf{S}^3$.[19] This characteristic can produce undesirable effects as unwind, in which the trajectories of the closed-loop system start close to the desired attitude and yet travel a large distance before returning to the desired attitude.[19]

Another possible derivation of Eq. (5) is using the vector g (Gibbs vector or Rodrigues parameter) as $Q = [g^T | q_4]$:

$$\dot{Q} = -\frac{1}{2}\begin{bmatrix} \omega^\times \\ \omega^T \end{bmatrix}\begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} + \frac{1}{2}q_4\begin{bmatrix} 1_{3\times 3} \\ 0 \end{bmatrix}\vec{\omega}, \tag{7}$$

where $\omega^\times$ is the cross-product skew-symetric matrix of $\vec{\omega}$ and 1 is the identity matrix. Note that the Gibbs vector is geometrically singular since it is not defined for 180 deg of rotation,[19] nonetheless, Eq. (7) is global.

3.2 *Rotational Dynamics*

In order to know the history of the change in the angular velocity, it is necessary to understand the history of the change in the angular acceleration [$\dot{\vec{\omega}} = G(\tau, t)$] of the satellite. According to the Euler–Newton formulation of the rotational motion, angular acceleration is caused by torques, in other words, the change in the angular momentum $\dot{\vec{h}}$ is equal to the net torques $\vec{g}$ applied in the satellite, see Eq. (8) (the present section is derived based on the center of mass of the satellite; for the general case, see Ref. 9):

$$\dot{\vec{h}} = \vec{g}. \tag{8}$$

Taking into account the motion of the body-fixed frame $\mathfrak{F}_b$ with respect to the ECI $\mathfrak{F}_i$ and the angular velocity $\vec{\omega}$, the derivative of the angular momentum in $\mathfrak{F}_b$ is defined as

$$\dot{\vec{h}} = \vec{g} - \vec{\omega} \times \vec{h}. \tag{9}$$

Furthermore, $\dot{\vec{h}} = I.\dot{\vec{\omega}}$ and $\vec{h} = I.\vec{\omega}$, where I is the time-invariant inertia tensor. The combination of this definition and Eq. (9) results in

$$I.\dot{\vec{\omega}} = \vec{g} - \vec{\omega} \times (I.\vec{\omega}). \tag{10}$$

Recall the satellite has a set of three reaction wheels, each one aligned with its principal axes of inertia, moreover, such type of actuator, momentum exchange actuators, does not change the angular momentum of the satellite. Consequently, it is mandatory to model their influence in the satellite, in particular, the angular momentum of the satellite is defined by

$$\vec{h} = \left(I - \sum_{n=1}^{3} I_{n,s} a_n a_n^T\right)\vec{\omega} + \sum_{n=1}^{3} h_{w,n}\vec{a}_n, \tag{11}$$

where $I_{n,s}$ is the inertia moment of the reaction wheels in their symmetry axis $\vec{a}_n$, $h_{w,n}$ is the angular momentum of the n reaction wheel about its center of mass ($h_{w,n} = I_{n,s} a_n^T \omega + I_{n,s}\omega_n$), and ω_n is the angular velocity of the n reaction wheel.

One can define I_b using

$$I_b = I - \sum_{n=1}^{3} I_{n,s} a_n a_n^T. \tag{12}$$

Using I_b, the motion of the satellite is described as

$$\dot{\vec{\omega}}^b = I_b^{-1}\vec{g_{cm}} - I_b^{-1}\omega^{\times} I_b \vec{\omega} - I_b^{-1}\omega^{\times}\sum_{n=1}^{3} h_{w,n}\vec{a}_n - I_b^{-1}\sum_{n=1}^{3} g_n \vec{a}_n, \tag{13}$$

where g_{cm} is the net external torque and g_n are the torques generated by the reactions wheels ($\dot{h}_{w,n} = g_n$).

4 Controller Design

In a zero-bias-momentum system, there are two dynamics states that must be controlled: (1) the attitude (perhaps described by unit quaternions Q) and (2) its stability ($\dot{Q}$, in other words, the angular velocity ω of the satellite). Taking into account the simulator, the following high-level requirements are derived: (1) is refined in "the attitude must be stabilized and must follow the Sun according to a given Sun versor in the satellite" and (2) is refined in "the angular velocity read by the gyroscope must be as close as possible of 0." These are typical requirements of the launch and early orbit phase (LEOP), in the majority of missions the most critical phase, in which the demand for the reaction wheels is the highest one.[20] These high-level requirements lead to the control loop shown in Fig. 1.

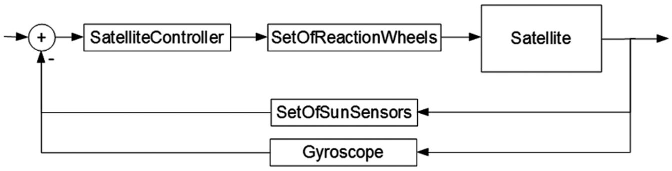


Fig. 1 Satellite control.

The state and the control vectors, for the control loop, can be defined as

$$\begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \end{bmatrix} = \begin{bmatrix} Q \\ \vec{\omega} \end{bmatrix} \quad [\vec{u}_1] = \left[\vec{T}_c = \sum_{n=1}^{3} g_n \vec{a}_n \right]. \tag{14}$$

The control regulator problem requires that the errors in the attitude and angular velocity must be obtained. The error in the angular velocity is directly obtained from the gyroscope readings, nonetheless, the error in the attitude must be computed. The applied approach to computation of the error in the attitude is: given two versors, namely (a) the actual Sun versor $\hat{s}_b$ in the satellite frame obtained by the proper transformation of the Sun versor in the ECI ($\hat{s}_i$); and (b) the reference versor in the satellite frame; to compute a rotation (there are many) from the actual Sun versor to the reference versor. The computed rotation can be described by an unit quaternion Q.

The following sections explore the state-space modeling and the controllers' synthesis. Table 1 shows the satellite characteristics[4] and references used in the linearization process and in the simulation results presented in Sec. 5.

4.1 *Linear Control Based on Proportional-Integral-Derivative Controller*

The satellite physical model previous explored is a nonlinear time invariant system. In order to fit such model in a state space model for a linear time invariant (LTI) system described by Eq. (15), the model must be linearized:

$$\dot{\vec{x}} = A\vec{x} + B\vec{u}, \quad \vec{y} = C\vec{x}. \tag{15}$$

The assumptions that the angular displacement is small ($\dot{\omega} = 0$ e $\dot{\theta} = \omega$, linearization) and that there are no net external torques ($g_{cm} = 0$) lead to the linearized model in the following equation:

$$\begin{bmatrix} \dot{\vec{x}}_1 \\ \dot{\vec{x}}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -I_b^{-1} \end{bmatrix} [\vec{u}_1], \quad [\vec{y}] = 1 \begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \end{bmatrix}. \tag{16}$$

Using control theory, the first alternative for the control of such model is the usage of a PID controller based on the error of Sun versor ($\vec{s}_{be}$) as well as the error of angular velocity ($\vec{\omega}_e$) in the satellite frame. At INPE, a typical controller for the LEOP is shown in Eq. (17), in which the Sun pointing error is $s_{pe} = \arccos \hat{s}_{b1}$ regarding the reference in Table 1.

Table 1 Satellite characteristics and references.

Name	Value
Satellite characteristics	
Inertia tensor (kg.m^2)	$\begin{bmatrix} 0.0547 & 0 & 0 \\ 0 & 0.0519 & 0 \\ 0 & 0 & 0.0574 \end{bmatrix}$
Actuators characteristics: reaction wheels	
Inertia (kg.m^2)	0.00015
Maximum torque (N.m)	0.000625
Maximum angular velocity (RPM)	750
References for the controller	
Sun versor in the body (XYZ)	$[0 \quad 0]^T$
Angular velocity (rad/s, XYZ)	$[0 \quad 0 \quad 0]^T$

$$\vec{u}_1 = -(K_p \vec{s_{be}} + K_{pd} \vec{\omega}_e + K_{pi} \dot{\vec{\omega}}_e), \quad \vec{s_{be}} = [0 - \hat{s}_{b3} \hat{s}_{b2}]^T, \quad \vec{\omega}_e = \vec{\omega} - \vec{\omega}_r. \tag{17}$$

Note that this type of controller neither has difficulties in its simulation nor in its numerical real-time implementation.

4.2 *Nonlinear Control Based on State-Dependent Riccati Equation Controller*

For small and slow maneuvers, a linear controller can be used, however, for large and fast maneuvers, the linearized equations do not hold and discontinuities compromise the system (e.g., saturation of the actuators).[10] In order to avoid linearization, the SDRE method is applied. SDRE is based on the arrangement of the system model in a pseudolinear form given by Eq. (1) and then solving an ARE in each sampling step.[6]

Assuming that there are no net external torques ($g_{cm} = 0$), the rotational dynamics defined in Eq. (13) can be rearranged as defined by Eq. (18) (applying the property $v^\times w = -w^\times v$):

$$\dot{\vec{\omega}} = \left[-I_b^{-1} \omega^\times I_b + I_b^{-1} \left(\sum_{n=1}^{3} h_{w,n} a_n \right)^\times \right] \vec{\omega} - I_b^{-1} \sum_{n=1}^{3} g_n a_n. \tag{18}$$

Taking into account the state and control vectors defined in Eq. (14), the state space model can be defined using Eqs. (5) (Ω) and (18). However, the SDC matrices do not fulfill the SDRE requirements, in particular, the pair (A, B) is not pointwise stabilizable. An alternative option for the definition of the SDC matrices is to use Eq. (7), which leads to

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} \begin{bmatrix} \omega^\times \\ \omega^T \end{bmatrix} & 0 & \begin{bmatrix} \frac{1}{2} q_4 I_{3\times 3} \\ 0 \end{bmatrix} \\ 0 & 0 & -I_b^{-1} \omega^\times I_b + I_b^{-1} \left(\sum_{n=1}^{3} h_{w,n} a_n \right)^\times \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -I_b^{-1} \end{bmatrix} [u_1]$$

$$[y] = 1 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \tag{19}$$

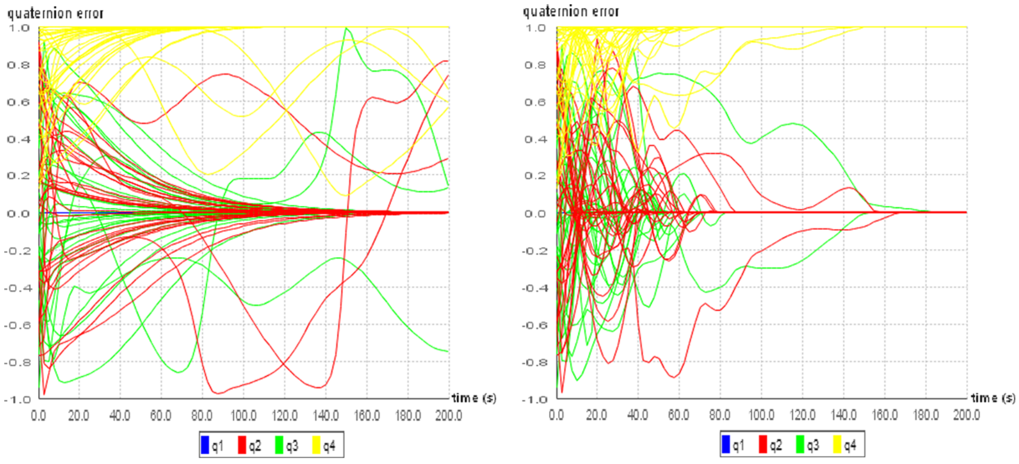
Equation (19) has been shown to satisfy SDRE conditions described in Sec. 2. Therefore, the state-dependent gain K is computed using Eqs. (3) and (4), which in turn supports the computation of the control vector $\vec{u}_1 = -K(\vec{x})\vec{x}$. Next section shares the simulation results for the controllers.

5 Simulation Results

In order to compare the performance and robustness of the controllers, a simulation test was conducted with the full Monte Carlo perturbation model described as follows: (1) the initial Euler angle errors of the nonlinear spacecraft system are randomly selected using three independent normal distributions ($\mu = 0$, $\sigma = 1$) multiplied by 90 deg; and (2) the initial angular velocity errors are randomly selected using three independent normal distributions ($\mu = 0$, $\sigma = 1$) multiplied by 0.1 rad/s.

The Monte Carlo model ran 30 times and in each time one simulation of the two controllers was executed. Such executions used simulation time 200 s, fixed step 0.005 s, the data presented in Table 1 and the following controllers: (1) PID controller [$K_p = 1$, $K_{pd} = \text{diag}(-24, -26, -32)$, $K_{pi} = 0$] defined by Eqs. (16) and (17); and (2) SDRE controller ($R = 1$ and $Q = 1$) defined by Eqs. (19) and (3).

Each graph in Fig. 2 shows the respective collection of all quaternion errors computed during simulations for a given controller. Figure 2 shows that the scalar part of quaternion (q_4) converges to 1 ($q_4 \rightarrow 1$) as well as that the vectorial part of the quaternion converges to 0 ($[q_1\, q_2\, q_3]^T \rightarrow [0\,0\,0]^T$) in the sense that the target attitude of the satellite was reached. The PID controller, in Fig. 2(a), is not able to control all runs in the specified simulation time.



(a) (b)

Fig. 2 Quaternion error comparison between controllers: (a) PID and (b) SDRE.

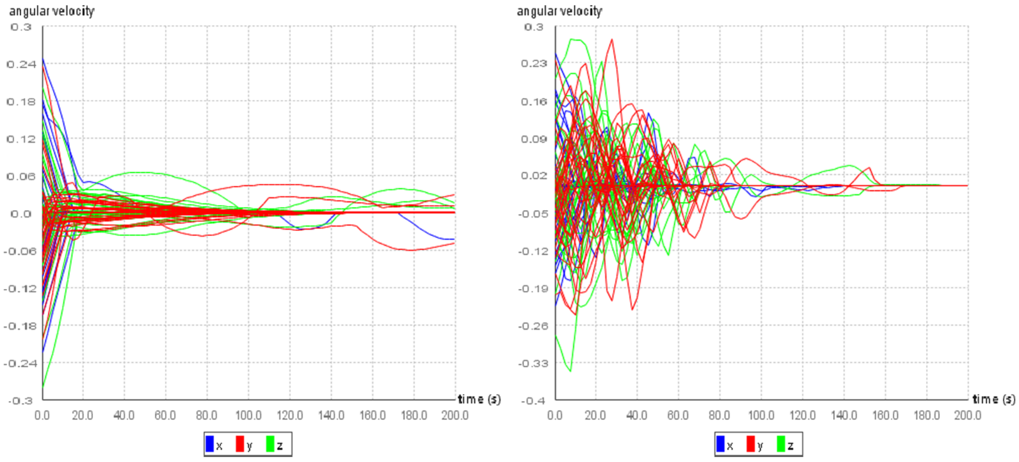
On the other hand, the SDRE controller, in Fig. 2(b), has the best performance since is exhibits asymptotically stability for all runs in at most 180 s.

The simulation results are in accordance with the literature[6–8,12] since an SDRE controller has better performance and robustness than the linear controllers.

Regarding satellite's angular velocity, each graph in Fig. 3 shows the respective collection of angular velocities (one for each axis) computed during simulations for a given controller. The PID controller, in Fig. 3(a), results in lower angular velocities, however, it does not stabilize the attitude and it has a residual angular velocity. On the other hand, the SDRE controller, in Fig. 3(b), results in initial higher angular velocities but it stabilizes the attitude and do not have residual angular velocity.

Once again, the simulation results are in accordance with the literature[6–8,12] since an SDRE controller has better performance and robustness than the linear controller albeit higher control effort and, consequently, higher energy consumption.

Such simulations results are neither valid for the general case nor for initial conditions out of the range of the Monte Carlo perturbation model due to the underlining nonlinear dynamics.



(a) (b)

Fig. 3 Satellite's angular velocity comparison between controllers: (a) PID and (b) SDRE.

6 Conclusion

To the best of our knowledge, the proof of concept of the open-source satellite simulator for SDRE controllers is original.[17,18,21] Moreover, the simulator can run in a variety of platforms—including an Android operating system in a remote sensing CubeSat—and it has low cost. It is worth mentioning that 39.33% of the cost of CONASAT was selected for the acquisition of software license (page 12, Ref. 3), whereas the open-source options, such as the one proposed here, would reduce such cost to zero.

Some capabilities of the simulator were not explored in this paper due to lack of space. For example, the simulator has flight dynamics capabilities (based on Orekit[11]), capabilities for the unloading of the angular momentum of the reaction wheels (based on a magnetometer and a set of magnetorques), runtime checkings, and other options to define the SDC matrices. In the mid-term, the simulator itself will be an open-source software under GNU General Public License (GPL).[21]

Regarding the discussion whether the application of SDRE technique in AOCS can yield gains in the missions developed by INPE since performance in the LEOP is critical to the success of a mission and the simulation results show that the performance of the SDRE controller is better; SDRE can yield gains. Nonetheless, its implementation requires more computing resources and tends to exhibit difficulties for verification. Therefore, it is too early to draw a definitive conclusion.

In conclusion, the open-source satellite simulator fills a gap in the open-source community, the availability of options to run SDRE controllers and filters in Java—including an Android operating system in a remote sensing CubeSat. Moreover, the simulation results indicate that SDRE can yield gains in the performance of INPE missions, in particular, CONASAT.

References

1. C. Bridges et al., “STRaND-1: the world’s first smartphone nanosatellite,” in *2nd Int. Conf. Space Technol.*, pp. 1–3 (2011).
2. J. Gozalvez, “Smartphones sent into space [mobile radio],” *IEEE Veh. Technol. Mag.* **8**(3), 13–18 (2013).
3. M. J. M. de Carvalho, “Estudo de uma missão espacial para coleta de dados ambientais baseada em nano satélites,” 2010, http://www.crn.inpe.br/conasat1/Documentos/gerais/Estudo%20de%20uma%20Missão%20Espacial%20para%20Coleta%20de%20Dados%20Ambientais%20Baseada%20em%20Nano%20Satélites.pdf (accessed 21 October 2018).
4. B. D. Reis de Mesquita, H. Koiti Kuga, and V. Carrara, “Estimation and attitude control in CONASAT nominal operation mode: an approach for SDRE filter and PID control,” *IEEE Lat. Am. Trans.* **15**(5), 835–842 (2017).
5. J. D. Pearson, “Approximation methods in optimal control I. Sub-optimal control†,” *J. Electron. Control* **13**(5), 453–469 (1962).
6. T. Çimen, “State-dependent Riccati equation (SDRE) control: a survey,” *IFAC Proc. Vol. (IFAC-PapersOnline)* **41**(2), 3761–3775 (2008).
7. T. Çimen, “Systematic and effective design of nonlinear feedback controllers via the state-dependent Riccati equation (SDRE) method,” *Ann. Rev. Control* **34**(1), 32–51 (2010).
8. J. R. Cloutier, C. N. D’souza, and C. P. Mracek, “Nonlinear regulation and nonlinear H-infinity control via the state-dependent Riccati equation technique,” in *Conf. Nonlinear Prob. Aviat.* (1996).
9. P. C. Hughes, *Spacecraft Attitude Dynamics*, John Wiley & Sons, Inc., New York (1986).
10. M. J. Sidi, *Spacecraft Dynamics and Control—A Practical Engineering Approach*, Cambridge University Press, England (2006).
11. “Orekit (version 8.0)” (2017) (accessed 1 April 2017).
12. R. G. Gonzales and L. C. G. de Souza, “Application of the SDRE method to design a attitude control system simulator,” *Adv. Astronaut. Sci.* **134**(Part 1–3), 2251–2258 (2009).
13. D. T. Stansbery and J. R. Cloutier, “Position and attitude control of a spacecraft using the state-dependent Riccati equation technique,” in *Proc. Am. Control Conf. ACC (IEEE Cat. No. 00CH36334)*, 3, 1867–1871 (2000).

14. P. K. Menon et al., "Real-time computational methods for SDRE nonlinear control of missiles," *Proc. Am. Control Conf.* 1, 232–237 (2002).
15. G. Di Mauro et al., "Nonlinear control for proximity operations based on differential algebra," *J. Guidance, Control Dyn.* **38**, 2173–2187 (2015).
16. G. Di Mauro, P. Di Lizia, and M. Lavagna, "Control of relative motion via state-dependent Riccati equation technique," in *AAS/IAAA Astrodyn. Spec. Conf.*, Univelt, San Diego, California, Vol. 142 (2011).
17. A. G. Romero and L. C. G. Souza, "Satellite controller system based on reaction wheels using the state-dependent Riccati equation (SDRE) on Java," in *Proc. 10th Int. Conf. Rotor Dyn.—IFToMM* (2019).
18. A. G. Romero and L. C. G. Souza, "Application of the SDRE technique based on java in a cubesat attitude and orbit control subsystem," in *IAA Proc. 3rd IAA Lat. Am. CubeSat—IAALACW* (2018).
19. P. W. Fortescue and G. G. Swinerd, "Attitude control," in *Spacecrafts System Engineering*, pp. 289–326, Wiley, New Jersey (2011).
20. J. R. Wertz and W. Larson, *Space Mission Analysis and Design*, Springer, Netherlands (1999).
21. A. G. Romero, "Satellite's simulation—developer guide (version 0.0.1)," (2019), to appear.

Alessandro G. Romero is a PhD student at the Brazilian National Institute for Space Research (INPE) in the concentration area of space mechanics and control. He obtained his PhD through a Brazil/Germany bilateral program at INPE and TU Kaiserslautern in 2014. Moreover, he obtained his master's degree at INPE in 2010 and his undergraduate degree at São Paulo State Technological College in 2000.

Luiz C. G. de Souza obtained his undergraduate degree in physics from the University of Brasilia (1978), his master's degree in space engineering and technology from the National Institute for Space Research (1987), and his PhD in space engineering from the College of Aeronautics at Cranfield University (1992), England. He has experience in the area of aerospace engineering focused on the rigid-flexible satellites. He is a professor at UFABC for the aerospace engineering course.