We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,300

Open access books available

172,000

International authors and editors

190M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Modern Control System Learning

Brendon Smeresky and Alex Rizzo

Abstract

This manuscript will explore and analyze the effects of different controllers in an overall spacecraft's attitude determination and control system (ADCS). The experimental setup will include comparing an ideal nonlinear feedforward controller, a feedback controller, and a combined ideal nonlinear feedforward + feedback controller within a Simulink simulation. A custom proportional, derivative, integral controller was implemented in the feedback control, adding an additional term to account for the nonlinear coupled motion. Consistent proportional, derivative, and integral gains were used throughout the duration of the experiment. The simulated results will show that the ideal nonlinear feedforward controller lacked an error correction mechanism and took extra time to execute, the feedback controller fared only slightly better, but the combined ideal nonlinear feedforward controller with feedback correction yielded the highest accuracy with the lowest execution time. This highlights the potential effectiveness for a learning control system.

Keywords: control systems, feedforward, feedback, learning systems

1. Introduction

The goal of a spacecraft's attitude determination and control system (ADCS) is to have a system that can move to and hold a specific orientation in three-dimensional space, relative to an inertial frame. This project can be viewed through three different lenses: classical control, modern control, and/or machine learning. These lenses explain the same control theory in three different contexts. For all the control theories, the ADCS considers the kinetics, kinematics, disturbances, controls, and actuators that dictate the system's motion [1]. Specifically, with regard to classical control, both feedforward and feedback controller are implemented in order to eliminate the error between a desired and commanded signal [1]. With regard to modern control, a similar estimation and correction method is implemented using either an extended Kalman controller, which relies upon on a nonlinear control estimator coupled with a linear control corrector, or an unscented Kalman controller, which uses both a nonlinear control estimator and nonlinear control corrector, in order to reduce error [2–6]. The third context relates control systems to deterministic artificial intelligence and machine learning. The control estimate derives from self-awareness of its own attributes that update every time-step with new information. This self-updating learning mechanism can be viewed akin to the update cycle used by supervised learning algorithms to model a system's performance. As an example, the updating mechanism is either a linear or nonlinear method to update an unknown inertia matrix for a spacecraft [7–13].

Figure 1 depicts the topology of the computational steps that take desired angle inputs and calculates Euler angle outputs: ϕ , θ , and ψ . The desired angle inputs are

Figure 1.
The overview of the control system from the desired ϕ , θ , and ψ inputs (white box), through calculations steps (light gray boxes) to ϕ , θ , and ψ Euler angle outputs (dark gray box).

processed through the trajectory, controls, actuators, dynamics, and disturbances blocks. Section II will explain the theory behind the overall control system, Section III will detail the experimental setup, Section IV will show the results, and Section V will conclude this paper.

2. Background and theory

The rotation maneuver from one position to another is measured from the inertial reference frame or $[X_I, Y_I, Z_I]$ to the final position, the body reference frame or $[X_B, Y_B, Z_B]$, as depicted in **Figure 2**. For this simulation, a model was created to rotate from orientation A, $[X_A, Y_A, Z_A]$, to orientation B, $[X_B, Y_B, Z_B]$. The kinetics, kinematics, orbital frame, and disturbance calculations are explained in Ref. [1]. This paper focuses on a combined feedforward plus feedback controller as an error estimation and correction mechanism. This simulation utilizes three control moment gyroscopes (CMG) that are responsible for physically moving the system according to the inputted control signal. Additionally, this paper will focus on how the control system implementation affects the rotational maneuver and final orientation of a spacecraft.

Figure 2.
Execution of a rotation from X_I to X_B ; blue arrows denote angle rotations which can be seen to rotate around the quaternion axis, q_4 in red.

2.1 Control moment gyroscopes

A CMG acts like a large reaction wheel spinning at a constant speed that transfers energy by changing its orientation. It successfully induces motion by following the governing equations of rotational mechanics, detailed in Eqs. (1) and (2):

$$T = \dot{H}_s + \omega \times H_s = u^* = \dot{H} \quad (1)$$

$$H_s = J\omega + h \quad (2)$$

These governing equations apply to any rotating, rigid body of mass. T represents the torque which is equal to the optimal control, u^* ; H_s represents the total system's angular momentum; ω represents the angular velocity of the body; J is the inertia matrix for the entire body, including the CMGs; and h is the momentum produced by the CMGs. All these variables are expressed in the body frame of the spacecraft.

Physically on a spacecraft, the CMGs are typically one of several cylinders located at different orientations within the spacecraft's bus, used to perform reorientation maneuvers. In **Figure 1**, the CMGs are represented within the actuators block, which is directly before the dynamics block. The actuator input is a commanded torque $u = \dot{H}$, and the output is an actual torque that is fed to the system using a sinusoidal trajectory described in [14].

The method of transforming a commanded torque, T , into an actual torque using the CMGs is described in Eqs. (3), (4), and (5). In these equations, the variable θ_i is the rotation about the gimbal axis, and β is the skew angle orientation of the respective CMG. The notations have also been shortened so that sine and cosine trig functions are "s" and "c," respectively:

$$[A] = \frac{\delta H}{\delta \theta} = \begin{bmatrix} \sin \theta_1 & \cos \beta_2 \cos \theta_2 & -\sin \theta_3 \\ -\cos \beta_1 \cos \theta_1 & \sin \theta_2 & \cos \beta_3 \cos \theta_3 \\ \sin \beta_1 \cos \theta_1 & \sin \beta_2 \cos \theta_2 & \sin \beta_3 \cos \theta_3 \end{bmatrix} \quad (3)$$

$$[A]^{-1} \dot{H} = [A]^{-1} [A] \{\dot{\theta}\} = \{\dot{\theta}\} \quad (4)$$

$$[A]^{-1} = \begin{bmatrix} \frac{c(\theta_3)}{c\theta_1 s\theta_3 + s\theta_1 \theta_3} & \frac{-s(\theta_3)}{(c\theta_1 s\theta_3 + s\theta_1 \theta_3)t\theta_2} & \frac{s(\theta_3)}{c\theta_1 s\theta_3 + s\theta_1 \theta_3} \\ 0 & \frac{1}{s\theta_2} & 0 \\ \frac{-c(\theta_1)}{c\theta_1 s\theta_3 + s\theta_1 \theta_3} & \frac{-s(\theta_1)}{(c\theta_1 s\theta_3 + s\theta_1 \theta_3)t\theta_2} & \frac{s(\theta_1)}{c\theta_1 s\theta_3 + s\theta_1 \theta_3} \end{bmatrix} \quad (5)$$

The first step is to build $[A]$, commonly referred to as the Jacobian matrix or the spatial matrix. This matrix takes the angular changes in H and compares them to the changes in θ_i . This concept is visually represented in **Figure 3**, given a fixed β and fixed θ_i from (3).

The second step is based on the "so-called" inverse steering law, which is represented in (4). This equation relates the commanded torque to the gimbal rate, where the commanded torque is also known as the rate of change of the angular momentum. This calculation can be done with a variety of methods. A few potential methods for inversion include singular value decomposition, matrix inversion for square matrices, matrix pseudo-inversion for non-square matrices, or element by element in $[A]^{-1}$ as defined in (5). Each of these methods yields a commanded gimbal rate, $\dot{\theta}$. For simplification, the actuator is assumed to be perfect and produces the actual gimbal rate commanded without error or loss.

Figure 3. CMG diagram with blue for axes and β angles, red for angular momentum vectors, green for θ rotation axes, and purple for the direction of θ rotation.

The last step is to invoke the inverse steering law in (6) to relate the Jacobian matrix to the actual gimbal rate and determine the actual torque produced. This can be completed using decoupled equations, as shown in [15]:

$$T = [A]\{\dot{\theta}\} = \dot{H} \quad (6)$$

Alternatively, Eqs. (4), (5), and (6) can be combined algorithmically and solved for using Eq. (7):

$$\begin{aligned} \det[A] = & s\theta_1(s\theta_2s\beta c\theta_3 - c\beta c\theta_3s\beta c\theta_2) + c\beta c\theta_2(c\beta c\theta_1s\beta c\theta_3 + c\beta c\theta_3s\beta c\theta_1) \\ & + s\theta_3(c\beta c\theta_1s\beta c\theta_2 + s\theta_2s\beta c\theta_1) \text{ where } \beta_1 \neq \beta_2 \neq \beta_3 \end{aligned} \quad (7)$$

In one calculated step, an output torque is yielded. This torque will feed into the dynamics as described in (1), using the process detailed above and in [1].

2.2 Singularities

A singularity occurs when an element in the Jacobian matrix, from (3), is resolved to an undefined value. This is caused by a zero existing within the denominator of (5), which complicates the calculation when solving for the determinant of $[A]$. Eq. (7) shows that singularities depend upon θ_1 , θ_2 , θ_3 , β_1 , β_2 , and β_3 values. Conceptually, this occurs at certain θ and β combinations because infinite torque is required to move the CMG when the torque vector is orthogonal to the gimbal axis. Trying to do so is impossible and yields unstable behavior as the CMG acts erroneously and tries to resolve a $1/0$ calculation and command an undefined torque. Furthermore, when all the β angles are equal, $\beta_1 = \beta_2 = \beta_3$, (7) simplifies further but is depicted as is for thoroughness.

This analysis is completed by taking (7) and stepping through θ and β combinations to verify which yield a zero determinant. For example, with $\sin \theta_1$, $\cos \beta_2$, $\cos \theta_2$, and $\sin \theta_3$ all equal 0, the determinant of $[A]$ is zero, yielding a possible solution

of $\theta_1 = \theta_2 = \theta_3 = 0^\circ$. However, this is a long process with the many permutations of solutions, which are omitted in this manuscript [15].

2.3 Controllers and observers

The input to a CMG is a torque vector, $[T_x, T_y, T_z]$, which is a signal generated by the trajectory block in **Figure 1**. However, this signal is not tuned to adjust to real world influences, where mechanical hardware can introduce errors due to incorrectly or un-modeled attributes, noise, etc. In order to overcome these losses, either a feedforward controller, a feedback controller, or a combination of both controllers can be used to counter introduce errors. More specifically, proportional, integral, and derivative (PID) gains are correlated to the position error generated when moving from one position to another position to correct the errors. However, using only the position error e_θ , its integral $\int e_\theta dt$ and its derivative $\frac{d}{dt} e_\theta$ result in inaccuracies. This is due to the derivative calculation, which is both inefficient and inaccurate as a result of the virtual-zero reference created in the cascaded topology of a PID controller when the computation is initialized [16–19]. This inaccuracy can be prevented by sending in both the position error and the velocity error, which has been done in this experiment via an enhanced Luenberger proportional, derivative,

Figure 4.
Topology of a feedback controller with two methods of control: A classic PID controller or an enhanced Luenberger PDI controller.

Figure 5.
PDI controller with ω input to remove virtual-zero reference with K_p , K_d , and K_i gains.

Figure 6. Topology of an observer based upon a PID controller and two sets of differentiation to yield full-state knowledge.

integral (PDI) controller [16–19]. Additionally, the enhanced Luenberger controller differs from the conventional PID controller which only receives a position error. The result is a controller that outputs a commanded torque to the actuator block in **Figure 1**. Topologies are shown of the overall feedback controller in **Figure 4**, and the enhanced Luenberger PDI controller in **Figure 5**.

The simulation in this experiment utilized an observer to eliminate the virtual-zero reference described above. An observer is comparable to a controller as they both take in a position via system sensors and output a twice differentiated term to produce full-state feedback knowledge. However, it provides an advantage: because it is not implemented on a specific piece of hardware, much higher gains can be utilized than can be used otherwise. A topology of the observer is shown in **Figure 6** in the controller/observer block.

2.4 Control system learning

A control system is capable of learning by estimating its desired position and then updating the control to correct for errors. In a learning control system, the control estimator is the feedforward controller defined by (8), and the corrector or learning mechanism is the feedback controller defined by (9), where Φ and Θ are defined by (10) and (11), respectively. The feedback controller can also be written in terms of the gains and the position error, e_θ , as shown in (12). Combining (8) and (9) yields a learning system that develops a more accurate control over time through (13):

$$u_{ff} = \sum T = J\dot{\omega}_d + \omega_d \times J\omega_d = \Phi\Theta \quad (8)$$

$$u_{fb} = [\Phi]\{\hat{\Theta}\} = \Phi(\Phi^T\Phi)^{-1}\Phi^T\delta u \quad (9)$$

$$\Phi = \begin{bmatrix} \dot{\omega}_x & \dot{\omega}_y & \dot{\omega}_z & -\omega_y\omega_z & 0 & \omega_z\omega_y \\ \omega_x\omega_z & \dot{\omega}_x & 0 & \dot{\omega}_y & \dot{\omega}_z & -\omega_z\omega_x \\ -\omega_x\omega_y & 0 & \dot{\omega}_x & \omega_y\omega_x & \dot{\omega}_y & \dot{\omega}_z \end{bmatrix} \quad (10)$$

$$\Theta = \{J_{xx}, J_{xy}, J_{xz}, J_{yy}, J_{yz}, J_{zz}\}^T \quad (11)$$

$$u_{fb} = k_p e_\theta + k_d \dot{e}_\theta + k_i \int e_\theta \quad (12)$$

$$\begin{aligned} u_{tot} &= u_{ff} + u_{fb} \\ &= J\dot{\omega}_d + \omega_d \times J\omega_d + \Phi(\Phi^T\Phi)^{-1}\Phi^T\delta u \\ &= \Phi\Theta + \Phi(\Phi^T\Phi)^{-1}\Phi^Tdu = \Phi\left[\Theta + (\Phi^T\Phi)^{-1}\Phi^T\delta u\right] \end{aligned} \tag{13}$$

Overall, the term “ $[\Phi]\{\hat{\Theta}\}$ ” in (9) represents the self-awareness statement. The nonlinear state transition matrix, $[\Phi]$, was built by knowing the dynamics of the system, and $\{\hat{\Theta}\}$ is the vector of unknown variables. Another application includes analyzing a changing inertia matrix, where it is assumed that the mass of the system is varying. The vector of unknowns, $\{\hat{\Theta}\}$, is the learned moment of inertia that is recalculated at every iteration of the model and determining its new mass [9].

3. Experimental setup

This manuscript documents the implementation of three different control algorithm combinations to induce a yawing motion on a spacecraft by generating the commanded torque to a three-gimballed spacecraft ACS. The three cases are a nonlinear feedforward control (case 1), a linear feedback control (case 2), and a combination of both nonlinear feedforward + linear feedback (case 3). The gains for these controllers are found in **Table 1**.

The model in this manuscript was built in MATLAB and Simulink, where integrations were calculated using the ode45 with the Runge–Kutta solver and a fixed time-step. Euler angles were resolved using a 3-2-1 rotation sequence with the atan2 trigonometry function.

As per [1], initialized values include torque = $[0,0,0]$ and quaternion = $[0,0,0,1]$. The spacecraft’s inertia matrix is $J = [10,0.1,0.1; 0.1,10,0.1; 0.1,0.1,10]$. The disturbances are defined in [1]. The orbital altitude was set at 150 km with a drag coefficient of 2.5 and orbital motion off. Each simulation executed over a five-second quiescent period, five-second maneuver time, and five-second post-maneuver observation period, totaling 60 seconds and yielding a $\omega_f = \pi/2$ and $\varphi = \pi/2$ for the sinusoidal trajectory of the controller.

	K_p gain	K_d gain	K_i gain
PDI controller	1000	10	0.1
Observer	100,000	500	0.1

Table 1.
Tuned gain values for the PDI controller and observer.

4. Experimental results and analysis

4.1 Time-step analysis

Time-step analysis was completed to determine whether reducing the time-step would help minimize the body frame to the inertial frame error deviations. The results of executing a maneuver with a feedforward + feedback controller utilizing two different time-steps are depicted in **Figure 7**. Expectations were that a smaller time-step would result in more precise results, meaning a smaller deviation

between the commanded and executed Euler angles. However, comparing the red and blue trajectories within each of the three plots in **Figure 7** shows that although some refinement is gained by decreasing the time-step, the gain is minimal. Therefore, a larger time-step can be used without losing much accuracy.

Comparing the $\theta_{\text{actual}} - \theta_{\text{desired}}$ and $\omega_{\text{actual}} - \omega_{\text{desired}}$ errors for time-steps of 0.01 and 0.001 in **Figure 8** yielded a similar result. The θ_z channel did receive the greatest amount of refinement with a smaller time-step, but no order of magnitude increase resulted. Therefore, these results confirm that varying the time-step has limited impact on the trajectories. Therefore, for the gains in **Table 1**, a minimum time-step of 0.01 is recommended, and decreasing it provides no additional benefit.

4.2 Control implementation

The performance of the three control system implementations is depicted in **Figure 9**. Comparing the three cases allowed further analysis on the differences between feedforward, feedback, and the combined feedforward + feedback control system. The feedforward and feedforward + feedback controllers are more precise than the feedback method. This is because it is based off an exact control equation, Eq. (8). Conversely, the feedback controller is based off a PDI controller that has one time-step of induced lag and is therefore less precise. Additionally, the gains in a PDI or PID controller must be finely tuned with predetermined gain values, which can be an iterative and time-consuming task because controller performance varies greatly depending on the values. Lastly, the combined controller configuration represents an error combination of both the feedforward and feedback plots. It allows the analytical accuracy of the feedforward equation to be updated with the responsiveness of the feedback correction, but too little error exists in this case for the results to be visible.

Figure 10 shows the error of the commanded Euler angle over time. The two left-most plots in **Figure 10** show that the error is different for each controller. The

Figure 7.
Time-step analysis for the ϕ , θ , and ψ Euler angles for two time-steps.

Figure 8.
Time-step analysis comparing $\theta_{actual}-\theta_{desired}$ and $\omega_{actual}-\omega_{desired}$ errors.

Figure 9.
Controller error over time for the three configurations.

feedback controller fluctuates initially as it corrects to minimize error over time. The feedforward controller is excellent initially but slowly deviates as error accrues without correction. Lastly, the combined controller is the best of both and starts with minimal error but then corrects that error over time. However, note that the error is minimal because no movement was commanded in either the φ or θ channels; the residual error exists because of cross-chatter between channels resultant from Eqs. (3) and (5).

Figure 10.
Euler angle error for the three controller configurations.

	$\Phi_{\text{actual}} - \Phi_{\text{desired}}$	$\theta_{\text{actual}} - \theta_{\text{desired}}$	$\Psi_{\text{actual}} - \Psi_{\text{desired}}$	Run time (seconds)
u_{ff}	$-3.93\text{e-}04^\circ$	$4.18\text{e-}04^\circ$	$-1.48\text{e-}02^\circ$	20.7
u_{fb}	$6.03\text{e-}09^\circ$	$-9.01\text{e-}08^\circ$	$4.66\text{e-}08^\circ$	20.5
u_{ff+fb}	$1.06\text{e-}08^\circ$	$-7.94\text{e-}09^\circ$	$4.38\text{e-}09^\circ$	20.3

Table 2.
The actual-desired Euler angle errors for the three cases using a 0.001 time-step and their associated run times.

The third plot from the left shows the position error in the ψ channel, which is the channel in which a 30-degree maneuver was commanded. Before the maneuver was started, the desired versus commanded error starts at 30°. At each time-step, the position is updated and the error decreases. Due to the scale of the maneuver, all three controllers appear to do well during the maneuver. To better understand the performance, the fourth and right-most plot is a zoom-in of the third plot, illustrating the accuracies of the feedforward controller due to the forward propagation accuracy of (8), the lag induced error and imperfect gain tuning of the feedback controller, and the results of the combined feedforward + feedback controllers. It is a better example of how each controller operates: the feedforward controller again starts off with minimal error before deviating over time, the feedback controller starts with the greatest amount of error that is overshoot and damped, and the combined controller, which is the best of both the minimal starting error and correction capability of the other controllers.

Lastly, **Table 2** compares the boundary condition satisfaction at the final time of the maneuver. The results show that the feedforward + feedback controller has both the least amount of error and the shortest runtime. Conversely, the feedforward controller is the worst in both accuracy and runtime but only by a small margin compared to the feedback controller. The feedforward controller is hypothesized to perform worse because (12) tries to model (1) but can only poorly approximate it, yielding inaccuracies, which accrue over time.

Figure 11 is a revisualization of the data in **Figure 10**. This depiction is more intuitive and breaks down the change in angular position over time for each

Figure 11.
Change in angular position for all three controller configurations.

controller, as well as magnifying the post-maneuver oscillations and damping. Commanding $[0;0;30]$, we see that all controller configurations are responsive to this input, with the expected differences. The accuracy of the feedforward control in the left-most plots, combined with the undamped response of feedback control of the middle plots, gives dampened response of the combined controller in the right-most plots.

5. Conclusion

This experiment implemented and compared the effects of a feedforward, feedback, and a combined forward + feedback control system. A yaw maneuver was commanded, and the response was measured to show that an ADCS can estimate and then update its control over time, similar to a feedforward and feedback learning mechanism. The results showed that the feedforward controller lacks a correction mechanism that accrues error and takes time; the feedback system is slightly better in both metrics, but the combined feedforward + feedback system combines the best of both systems for superior accuracy in the shortest time. Therefore, the combined system is the best choice for its accuracy and adaptability. However, this combined system needs to be further researched by subjecting the system to noise and induced disturbances to validate the combined system’s responsiveness.

Acknowledgements

We would like to thank our teacher, Dr. Timothy Sands, for his guidance in developing this work, as well as our families for their support while we spent our time away from them developing this research, so thank you.

Conflict of interest

The authors declare no conflict of interest.

IntechOpen

IntechOpen

Author details

Brendon Smeresky* and Alex Rizzo
Naval Postgraduate School Monterey, California, United States

*Address all correspondence to: bpsmeres@nps.edu

IntechOpen

© 2020 The Author(s). Licensee IntechOpen. Distributed under the terms of the Creative Commons Attribution - NonCommercial 4.0 License (<https://creativecommons.org/licenses/by-nc/4.0/>), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited.

References

- [1] Smeresky B, Rizzo A, Sands T. Kinematics in the information age. *Mathematics*. 2018;**6**(9):148
- [2] Kalman RE. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*. 1960;**82**: 35-45
- [3] McElhoe BA. An assessment of the navigation and course corrections for a manned flyby of Mars or Venus. *IEEE Transactions on Aerospace and Electronic Systems*. 1966;**2**(4):613-623
- [4] Julier SJ, Uhlmann JK, UKF. New extension of the Kalman filter to nonlinear systems. In: *Signal Processing, Sensor Fusion, and Target Recognition VI*. Proceedings of SPIE. 3. pp. 182-193
- [5] Sands T. Nonlinear-adaptive mathematical system identification. *Computation*. 2017;**5**:47
- [6] Kalman RE. Contributions to the theory of optimal control. In: *Boletín de la Sociedad Matemática Mexicana*. 1960; **5**(2). pp. 102-119
- [7] Nakatani S, Sands T. Battle-damage tolerant automatic controls. *Electrical and Electronic Engineering*. 2018; **8**(1):23
- [8] Nakatani S, Sands T. Simulation of spacecraft damage tolerance and adaptive controls. *IEEE Aerospace*. 2014;**1-16**:2014
- [9] S Nakatani T. Sands, nonlinear adaptive control: Autonomous damage recovery in space. *International Journal of Control, Automation and Systems*. 2016;**2**:23-36
- [10] Sands T, Kim JJ, Agrawal BN. Spacecraft fine tracking pointing using adaptive control. In: *Proceedings of 52th International Astronautical Congress*; 2007
- [11] Sands T, Kim JJ, Agrawal B. Spacecraft Adaptive Control Evaluation. *Infotech@ Aerospace* 2012, 2476
- [12] Sands T, Kim JJ, Agrawal BN. Improved Hamiltonian adaptive control of spacecraft. In: *2009 IEEE Aerospace Conference*. 1-10; 2009
- [13] Cooper M, Heidlauf P, Sands T. Controlling chaos – Forced van der pol equation. *Mathematics*. 2017;**5**(4):70
- [14] Baker K, Cooper M, Heidlauf P, Sands T. Autonomous trajectory generation for deterministic artificial intelligence. *Electrical and Electronic Engineering*. 2018;**8**:59-68. DOI: 10.5923/j.eee.20180803.01
- [15] Sands TA, Kim JJ, Agrawal B. Control moment gyroscope singularity reduction via decoupled control. In: *IEEE SEC* 2009, 388-391; 2009
- [16] Sands T., Lorenz R. Physics-based automated control of spacecraft. In: *Proceedings of the AIAA Space Conference & Exposition*; Pasadena, CA, USA; 14–17 September 2009
- [17] Sands TA. Physics-based control methods. In: *Advances in Spacecraft Systems and Orbit Determination*. London, UK: InTech Publishers; 2012. pp. 29-54
- [18] Sands T. Improved magnetic levitation via online disturbance decoupling. *Physics Journal*. 2015;**1**: 272-280
- [19] Sands T. Phase lag elimination at all frequencies for full state estimation of spacecraft attitude. *Physics Journal*. 2017;**3**(1):1-12