MEASUREMENT SCIENCE REVIEW, **23**, (2023), No. 4, 146-153

sciendo

ISSN 1335-8871

MEASUREMENT SCIENCE REVIEW

Journal homepage: https://content.sciendo.com

# Attitude Control and Parameter Optimization: A Study on Hubble Space Telescope

Emre Sayin, Rahman Bitirgen, Ismail Bayezit*

*Faculty of Aeronautics and Astronautics, Istanbul Technical University, Maslak, 34469, Istanbul, Turkiye,*
*bayezit@itu.edu.tr*

Abstract: In this work, we build a satellite attitude Proportional-Integral-Derivative (PID) controlled system by using the Hubble Space Telescope (HST) parameters as a reference and tune its controller parameters using various tuning methods. First, we give the equations for the motion of a satellite. We elaborate the control structure as controller, actuator, dynamics, and kinematics subsystems and construct an external disturbance model. We use a reaction wheel assembly used in the HST with the same configuration as the actuator. We evaluate the performance of the linearization by comparing it with the nonlinear model output. By working on the linearized model, we tune the PID controller parameters using two different methods: "Model-Based Root Locus Tuning” and "Genetic Algorithm Based Tuning". First, we obtain the controller parameters by manipulating the poles on the root locus plot of the linearized system. In addition, we use genetic algorithms to find the optimized controller values of the system. Finally, we compare the performances of the two methods based on their cost function values and find that the Genetic Algorithm-based tuned parameters are more fruitful in terms of the cost function value than the parameters obtained by the Root Locus-based tuning. However, it is found that the Root Locus-based tuning performs better in disturbance rejection.

Keywords: Spacecraft attitude control, controller parameter optimization, reaction wheel actuation, quaternion error feedback, Hubble Space Telescope.

## 1. Introduction

The advancement of technology has undergone a significant transformation since the first satellite was launched into space. Presently, the Earth is surrounded by a multitude of over 2000 satellites [1]. Among them, the Hubble Space Telescope (HST) stands as a notable orbiting spacecraft, devised by NASA to serve as the pioneering space telescope for observing the cosmos from Earth's orbital vantage point [2].

The HST has a crucial component known as the pointing system, which facilitates the alignment of the satellite with a fixed position in space. Since the primary objective of the HST is to observe celestial objects located at great distances, achieving exceptional pointing accuracy is essential. Consequently, the HST was engineered to have a pointing accuracy exceeding 0.01 arc seconds. This remarkable precision is achieved through the use of four Reaction Wheel Assemblies (RWA) as attitude control mechanisms, alongside four Magnetorquer Bars (MTBs), which are responsible for both momentum management and deceleration of the reaction wheels. For this work, the MTBs are ignored because they contribute little to the momentum change of the HST. The reaction wheels of the HST produce a torque of 0.8 Nm at a maximum speed of 3000 rpm and a rotor moment of inertia of 0.84 kgm$^2$ per wheel [3]. The moment of inertia of the HST can be found in [4].

As with any satellite that has an attitude control system, a control structure must be built to align the spacecraft to a desired angle. To build the model-based control structure, a mathematical model of the spacecraft must be created and simulated in the computational environment.

Proportional-Integral-Derivative (PID) control is still the most popular control strategy, used in 90-95% of industrial applications [5]. One of the critical issues in the development of PID controllers is the selection of the proper controller parameters ($K_p$, $K_i$ and $K_d$). Numerous studies have been conducted on controller parameter tuning, such as classical Ziegler-Nichols Tuning [6], Cohen-Coon Method [7], Relay Based Tuning [8], Self-Tuning [9], Reinforcement Learning Neural Network Based Tuning. [10], Particle Swarm Optimization (PSO) Based Tuning [11], and Genetic Algorithm Based Tuning [12]-[14].

In this paper, we use both model-based root locus tuning and genetic algorithm to find the proper controller gains. Several studies have been conducted in the literature that covers satellite attitude control with PID tuning methods

DOI: 10.2478/msr-2023-0019

*Corresponding author: bayezit@itu.edu.tr (I. Bayezit) using various methods similar to ours. Wang et al. designs a flexible spacecraft attitude control simulation using a PID control algorithm. They optimize the control parameters using PSO and compare them with the trial-and-error method, which can be described as a time-consuming method for complex systems [11]. Khoshrooz et al. design a reaction wheel actuated satellite attitude control system. They design a PD controller and optimize the PD parameters using Genetic Algorithm and PSO methods. They compare two optimized variants with an LQR controller and verify them with a Hardware-in-the-Loop testbed [13]. Daw et al. design a geostationary satellite attitude control system using PID controllers. They tune the parameters of the controller using experimental Ziegler-Nichols, a fairly generic tuning technique, and a genetic algorithm. At the end of their study they compare the performance of the two techniques [14]. Jia and Yang design a nonlinear adaptive PD controller for attitude control of a rigid spacecraft to compensate for spacecraft uncertainties such as inertia variation. An adaptive part is added to the nonlinear PD controller and the parameters are determined using a genetic algorithm. However, no comparison has been made with other tuning methods [12].

In this work, we present a structure for satellite attitude control using PID controllers. We elaborate the design using the physical parameters of the HST. Then we tune the controller parameters using the model-based root locus method and the genetic algorithm. We simulate and compare the results in a commercial software environment.

The outline of the paper is as follows: in Section A, the satellite equations of motion are constructed as dynamics, kinematics, and actuator subsystems. Also, the linearized equations of motion are shared. The attitude control structure is constructed in Section B, and a comparison between nonlinear and linear models is performed to show that the linearized model is reliable to use instead of the nonlinear model in terms of linear model-based tuning and faster computation process. In Section C, the controller parameters are tuned to meet the time domain requirements using Root Locus and Genetic Algorithm-based methods. Section 3 compares the performance of tuning with respect to cost function values. In Section 4, conclusions and the advantages and disadvantages of both methods are discussed and future research aspects are explored.

## 2. METHODS

### *A. Satellite equations of motion*

In this section, the equations of motion of the HST are given with dynamics, kinematics, and actuator subsystem equations.

#### *Dynamics*

Spacecraft dynamics with reaction wheels as actuators can be calculated using the dynamics equations based on Newton's second law as given below [15]:

$$\dot{h}_{tot} = N_e - \omega \times h_{tot} \tag{1}$$

where $h_{tot}$ is the total angular momentum acting on the spacecraft, $\omega$ is the angular velocity vector of the spacecraft, and $N_e$ is the external torque acting on the spacecraft.

Equation (1) can be divided into two parts: the spacecraft and the reaction wheel. For the spacecraft, the angular momentum can be defined as $I_s\omega$. Then (1) becomes:

$$\frac{d}{dt}(I_s\omega) + \dot{h}_w = N_e - \omega \times I_s\omega - \omega \times h_w \tag{2}$$

where $h_w$ is the angular momentum generated by the reaction wheels and $I_s$ is the moment of inertia matrix of the spacecraft. It is also known that the control torque vector $N_c$ is related to the rate of change of the angular momentum vector of the satellite as:

$$\dot{h}_w = -N_c \tag{3}$$

than (2) becomes [16]:

$$\dot{\omega} = I_s^{-1}(-S(\omega)\boldsymbol{I_s\omega} - S(\omega)h_w + N_c + N_e) \tag{4}$$

where $S$ is the skew-symmetric matrix given as:

$$\begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{5}$$

#### *Kinematics*

To create kinematic equations, a quaternion representation can be used. A quaternion is a representation with three vector and one scalar element as [17]:

$$q = iq_1 + jq_2 + kq_3 + q_4 \tag{6}$$

Equation (6) can be rewritten in terms of its vector and scalar elements as:

$$q \equiv (g, q_4) \tag{7}$$

It is also possible to indicate the kinematic equations by using the Euler angles, which can be defined as rotations around the body axis of the spacecraft:
$\phi$: Roll angle (rotation around the x-axis)
$\theta$: Pitch angle (rotation around the y-axis)
$\psi$: Yaw angle (rotation around the z-axis)

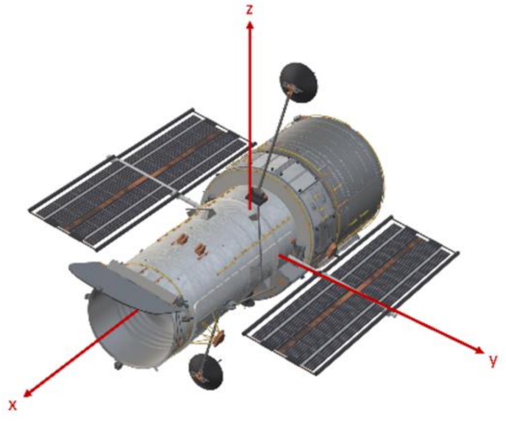

Fig. 1. An illustration of the HST with defined axes.

The defined x, y, and z axes for the HST are shown in Fig. 1. In addition, Euler angles can be converted to quaternions using the following transformation:

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} s\left(\frac{\phi}{2}\right)c\left(\frac{\theta}{2}\right)c\left(\frac{\psi}{2}\right) - c\left(\frac{\phi}{2}\right)s\left(\frac{\theta}{2}\right)s\left(\frac{\psi}{2}\right) \\ c\left(\frac{\phi}{2}\right)s\left(\frac{\theta}{2}\right)c\left(\frac{\psi}{2}\right) + s\left(\frac{\phi}{2}\right)c\left(\frac{\theta}{2}\right)s\left(\frac{\psi}{2}\right) \\ c\left(\frac{\phi}{2}\right)c\left(\frac{\theta}{2}\right)s\left(\frac{\psi}{2}\right) - s\left(\frac{\phi}{2}\right)s\left(\frac{\theta}{2}\right)c\left(\frac{\psi}{2}\right) \\ c\left(\frac{\phi}{2}\right)c\left(\frac{\theta}{2}\right)c\left(\frac{\psi}{2}\right) + s\left(\frac{\phi}{2}\right)s\left(\frac{\theta}{2}\right)s\left(\frac{\psi}{2}\right) \end{bmatrix} \tag{8}$$

where $c$ stands for cosine and $s$ stands for sine. Moreover, quaternions can be converted to Euler angles using (9):

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \tan^{-1}\frac{2q_4q_1+2q_2q_3}{1-2(q_1^2+q_2^2)} \\ \sin^{-1}(2q_4q_2 - 2q_3q_1) \\ \tan^{-1}\frac{2q_4q_3+2q_1q_2}{1-2(q_2^2+q_3^2)} \end{bmatrix} \tag{9}$$

In this paper, the kinematic equations are represented using quaternions because singularities occur in high-angle maneuvers, called gimbal lock, when Euler angles are used to represent the kinematics. The use of quaternions is also advantageous in terms of fast computational power. Furthermore, the kinematic equations can be represented as [16]:

$$\dot{q} = \frac{1}{2}\Omega(\omega)q \tag{10}$$

where:

$$\Omega(\omega) = \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \tag{11}$$

By using kinematic equations, it is possible to obtain the quaternion elements of a spacecraft from its angular velocity elements. The angular velocity vector can be obtained by integrating the rate of change of the angular velocity vector into the dynamic equations.

By combining dynamic and kinematic equations, the orientation of a satellite can be determined from the forces acting on it, i.e., the external torque and the torque generated by the reaction wheels. In this work, it is assumed that there are no external torques acting on the HST. Also, an actuator model must be built to calculate the torque generated by the RWA.

### *Actuator*

The HST uses four reaction wheels and magnetorquers to stabilize the attitude of the spacecraft. The RWA consists of two pairs. The pairs are spaced 90 degrees apart. Also, the elements of each pair are tilted ±20 degrees from the defined yz plane [18].

Since the Satellite Body Coordinate (SBC) axis and the reaction wheel axes in the HST are not identical, the torques generated by the RWA must be converted to the SBC system. To convert the RWA to the SBC system, a transformation matrix $A$ must be created. Multiplying the transformation matrix by the torque vector of the RWA ($\boldsymbol{T_{RWA}}$) results in the total torque vector in the SBC system ($\boldsymbol{T_{SBC}}$):

$$\mathrm{T_{SBC}} = [\mathrm{A}]\mathrm{T_{RWA}} \tag{12}$$

The transformation matrix $A$ can be written as:

$$\Omega(\omega) \equiv \begin{bmatrix} -s(20)c(45) & s(20)c(45) & s(20)c(45) & -s(20)c(45) \\ -c(20)c(45) & -c(20)c(45) & c(20)c(45) & c(20)c(45) \\ s(45) & s(45) & s(45) & s(45) \end{bmatrix} \tag{13}$$

For the RWA configuration of the HST, the total torque vector in the SBC system can be obtained by substituting the obtained transformation matrix into (12).

Moreover, the torque vector in the SBC system can also be converted to a RWA torque vector. To do this, the pseudo-inverse of the transformation matrix, $A^{\dagger}$, must be substituted into the equation[19]:

$$T_{RWA} = [\boldsymbol{A}^{\dagger}]T_{SBC} \tag{14}$$

For simplicity, a reaction wheel can be assumed to be a Permanent Magnet DC (PMDC) motor. Since not all parameters of the HST RWA are publicly available, a commercial PMDC motor was chosen (PBLH60AS115-430) that meets the requirements of the HST RWA, which are 0.8 Nm torque capability and 3000 rpm speed [3].

It is possible to explain the response of a PMDC motor by using a set of linear equations [20]:

$$\frac{di_a(t)}{dt} = \frac{1}{L_a}[e_a(t) - R_a i_a(t) - K_b \omega_m(t)] \tag{15}$$

$$\frac{d\omega_m(t)}{dt} = \frac{1}{J_m}[K_i i_a(t) - T_L(t) - B_m \omega_m(t)] \tag{16}$$

where $i_a$ is the armature current, $e_a$ is the input voltage, $\omega_m$ is the motor angular velocity, $T_L$ is the load torque, $R_a$ is the armature resistance, $L_a$ is the armature inductance, $K_i$ is the torque constant, $K_b$ is the back-emf constant, $J_m$ is rotor inertia, and $B_m$ is the viscous-friction coefficient.

From (15) and (16), the resulting angular velocity of a PMDC motor can be derived as an output by using the voltage as an input. The angular velocity of the RWA in the body frame can be converted to the torque generated by the RWA using the ideal reaction wheel equation [16]:

$$\dot{\mathrm{h}} = \frac{\mathrm{d}}{\mathrm{dt}}\mathrm{J}\omega_{\mathrm{w}} = \mathrm{T_w} \tag{17}$$

where $h$ is the angular momentum vector, $J$ is the moment of inertia vector, $\omega_w$ is the angular velocity vector of the RWA, and $T_w$ is the torque vector generated by the RWA.

### *External disturbance*

It is known that there are two dominant external disturbances acting on the HST: the gravity gradient and the aerodynamic torques.

The gravity gradient torque is the main external force acting on the HST [21]. Since the HST is not symmetrical, the gravitational force acting on different points of the spacecraft changes and generates a disturbance torque. This torque can be approximated as a linear equation [22]:

$$G_x = \frac{3\mu}{R^3}\left(I_{zz} - I_{yy}\right)\phi$$
$$G_y = \frac{3\mu}{R^3}\left(I_{zz} - I_{xx}\right)\theta \tag{18}$$
$$G_z = 0$$

where $\mu$ is the gravitational parameter of the Earth and $R$ is the orbital radius for circular orbits. The HST has a nearly circular orbit with an eccentricity of 0.0003 [23].

The other disturbance torque is the aerodynamic torque. Since the HST is orbiting in the Low Earth Orbit (LEO), it is subjected to a drag force due to the air present at these altitudes. The drag force $F_d$ acting on the spacecraft can be calculated using a well-known equation [22]:

$$\mathrm{F_d} = 0.5\rho v^2 c_D A \tag{19}$$

where $\rho$ is the air density, $v$ is the linear velocity vector of the spacecraft, $c_D$ is the drag coefficient, and $A$ is the reference area facing the air. The total disturbance torque can be constructed using the gravity gradient and the aerodynamic torques acting on the HST, which are the dominant disturbances. For the HST, the total disturbance torque varies between 0-0.25 Nm, as given in [24]. To represent the total torque, which varies between 0-0.25 Nm, a band-limited white noise with a noise power of 5e$^{-6}$ and sampling time of 0.001 seconds was used. Fig. 2 shows the generated white noise. The values greater than 0.25 Nm represent the remaining external torques.

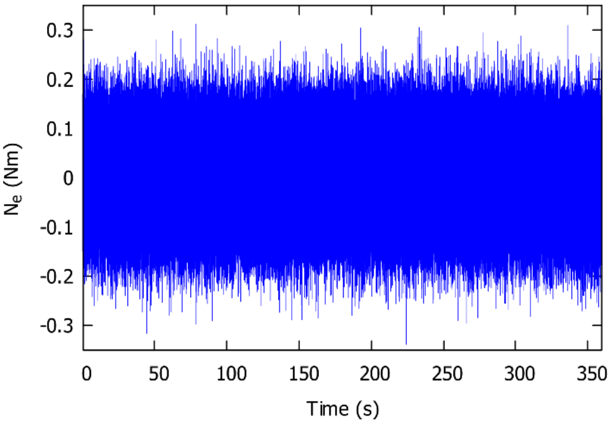

Fig. 2. Generated noise as external torque.

*Model validation*

The model was created in a commercial software environment. To validate the model, 3 volts are applied to the roll DC motor in the SBC system, which represents the reaction wheel of the spacecraft, neglecting the external disturbance. Since the voltage is given in the SBC, the spacecraft is expected to perform only roll motion. The output of the Euler angles is shown in Fig. 3. As shown in Fig. 3, the HST begins to rotate continuously around the roll axis. This is expected because the reaction wheel generates a control torque at the beginning of the movement, then there is no torque produced to stop the spacecraft.

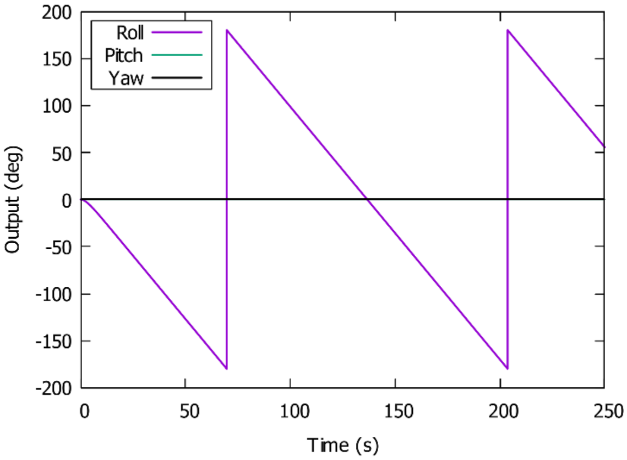

Fig. 3. Model output for 3 volts input of the reaction wheel.

*Linearization of equations of motion*

For a linear controller, it is important to have a linear system in order to adjust the controller parameters to the behavior of the system. Since the attitude control model of the HST is nonlinear, the equations of motion must be linearized.

Assuming that the external torque is zero, the linear state space equation of a spacecraft can be expressed as [16]:

$$\dot{x}(t) = A(t)x(t) + B_u(t)N_c(t) \tag{20}$$

where as,

$$\mathrm{A(t)} = \begin{bmatrix} I_S^{-1}A_{\omega,\omega} & 0 & I_S^{-1}A_{\omega,h} \\ \frac{1}{2}I_{3\times3} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{21}$$

Elements of matrix $A(t)$ are:

$$\mathrm{A_{\omega,\omega}} = \begin{bmatrix} A_{\omega,1} & A_{\omega,2} & A_{\omega,3} \end{bmatrix} \tag{22}$$

$$\mathrm{A_{\omega,1}} = \begin{bmatrix} 0 \\ -\omega_3 I_{zz} + \omega_3 I_{xx} + h_3 \\ \omega_2 I_{yy} - \omega_2 I_{xx} - h_2 \end{bmatrix} \tag{23}$$

$$\mathrm{A_{\omega,2}} = \begin{bmatrix} \omega_3 I_{zz} - \omega_3 I_{yy} - h_3 \\ 0 \\ -\omega_1 I_{xx} - \omega_1 I_{yy} + h_1 \end{bmatrix} \tag{24}$$

$$\mathrm{A_{\omega,3}} = \begin{bmatrix} -\omega_2 I_{yy} + \omega_2 I_{zz} + h_2 \\ \omega_1 I_{xx} - \omega_1 I_{zz} - h_1 \\ 0 \end{bmatrix} \tag{25}$$

$$\mathrm{A_{\omega,h}} = \begin{bmatrix} 0 & \omega_3 & -\omega_2 \\ -\omega_3 & 0 & \omega_1 \\ \omega_2 & -\omega_1 & 0 \end{bmatrix} \tag{26}$$

$$B_u = \begin{bmatrix} I_s^{-1} \\ 0 \\ -I_{3\times3} \end{bmatrix} \tag{27}$$

The moment of inertia vector of the satellite is taken as $I_s = diag\ (I_{xx}, I_{yy}, I_{zz}) = diag$ (31.046, 77.217, 78.754) [4] and the state space vector is:

$$x = \left(\widetilde{\omega}_1, \widetilde{\omega}_2, \widetilde{\omega}_3, \tilde{g}_1, \tilde{g}_2, \tilde{g}_3, \tilde{h}_1, \tilde{h}_2, \tilde{h}_3\right)^T \tag{28}$$

with the elements of the angular velocity vector of the spacecraft as $\widetilde{\omega}_1$, $\widetilde{\omega}_2$, and $\widetilde{\omega}_3$; the first three (vector) elements of the quaternion vector as $\tilde{g}_1$, $\tilde{g}_2$, and $\tilde{g}_3$; the elements of the angular momentum vector as $\tilde{h}_1$, $\tilde{h}_2$, and $\tilde{h}_3$.

Consequently, with knowledge of the satellite's angular velocity, angular momentum, and moment of inertia vector, it is possible to obtain a linear approximation of the attitude of the spacecraft using the linearized equations.

### *B. Attitude control structure*

A general attitude control diagram consists of a controller, an actuator, and the satellite system, which consists of the equations for the dynamics and kinematics of the spacecraft. The block diagram for the simulation of the attitude control system of the HST is shown in Fig. 4. The HST is exposed to gravity gradient, and aerodynamic and solar-array disturbances in orbit [24], [25]. For simplicity, it was assumed that there are no external and internal disturbances in the system and no errors in the attitude determination sensors.

In the case of the HST, PID controllers were used as controllers, a RWA consisting of four reaction wheels of the HST as actuators, and equations for the dynamics and kinematics of the satellite as the plant. The main purpose is to dampen the error between the desired angle and the output angle with a stable response. This purpose is achieved by changing the voltage input of the actuators using PID controllers.

In the work, a quaternion error vector between the output quaternion and the desired quaternion was calculated as an error vector to be used as input to the controller. Then, the voltage command generated by the controller is used for the PMDC motors of the actuator. The angular momentum and torque vector generated by the RWA is used by the satellite dynamics in order to calculate the corresponding satellite angular velocity. Finally, the satellite angular velocity is used by the kinematics subsystem to calculate the resulting quaternion, which is sent for error calculation. To visualize the angle response, the quaternion vector is converted to Euler angles using (9).

A linear control structure was also built using the linearized equations of motion. The linearized model also uses 3 PID controllers to control movement in the yaw, roll, and pitch axes. They generate voltage commands for the actuators and finally the torque vector generated by the actuators is converted to vector elements of the quaternion. Since the scalar element of the quaternion vector can be assumed to be '1' for small-angle maneuvers, it is assumed to be 1 for the linearized model.

*Comparison of nonlinear and linear models*

To test the precision of the linearization, the same random PID parameters were used for both the nonlinear and linearized models. First, the PID parameters and filter coefficient were set to 52.995, 2.488, 132.442, and 4.748, respectively. Then, for the roll, pitch, and yaw angles, the responses of the nonlinear and linearized models were examined for different angle commands. To find the difference between the nonlinear and linearized models, the root mean square error (RMSE) method was used. RMSE values for different inputs for the roll, pitch, and yaw angle commands are given in Table 1.

In addition, responses of the nonlinear and linear models were compared for corresponding angle inputs. An angle command with 3 degrees of roll and 15 degrees of yaw was input to the system for arbitrary PID coefficients (same for nonlinear and linear systems). The angle responses of the system are shown in Fig. 5 and Fig. 6, with closer views inside. As shown in Fig. 5 and Fig. 6, it is evident that for 3 degrees of roll input, the response of the nonlinear system is nearly identical to that of the linearized system. However, at 15 degrees of yaw input, it can be seen that the difference between the nonlinear and linearized models becomes larger, which is to be expected since linear models are mostly valid for small angle maneuvers. Consequently, it can be said that the linearized model is fairly reliable and can be used instead of the nonlinear model in determining the optimal controller parameters for small-angle maneuvers.

Table 1. RMSE values for different angle commands (in degrees).

| $Angle_{cmd}$ | $RMSE_{roll}$ | $RMSE_{pitch}$ | $RMSE_{yaw}$ |
|---|---|---|---|
| 03 | 6.365e$^{-4}$ | 7.893e$^{-4}$ | 7.878e$^{-4}$ |
| 08 | 0.014 | 0.016 | 0.0153 |
| 15 | 0.089 | 0.113 | 0.102 |
| 25 | 0.396 | 0.656 | 0.499 |
| 35 | 1.0219 | 2.696 | 1.434 |

### *C. Controller tuning*

Developing the controller for a system involves two tasks: determining the structure of the controller and tuning the controller parameter/s. The behavior of the system can be changed by manipulating its controller parameters. The controller structure can guarantee stability and convergence, but without proper tuning of the controller parameters, the system may not be in the best shape to meet the required or desired time-domain characteristics. For a PID-controlled system, there are several methods to obtain optimal PID values. In this paper, we present a tuning method based on genetic algorithm optimization and compare it with a well-known, well-established tuning method: Root Locus-based tuning.

We tune the parameters to control the roll axis of the spacecraft. The time domain requirements for the system are determined as the desired time constant is 1.5 seconds and the desired percentage overshoot for one degree of angle input is 2%. This robust behavior is preferred because the spacecraft components are very sensitive to sudden motion.

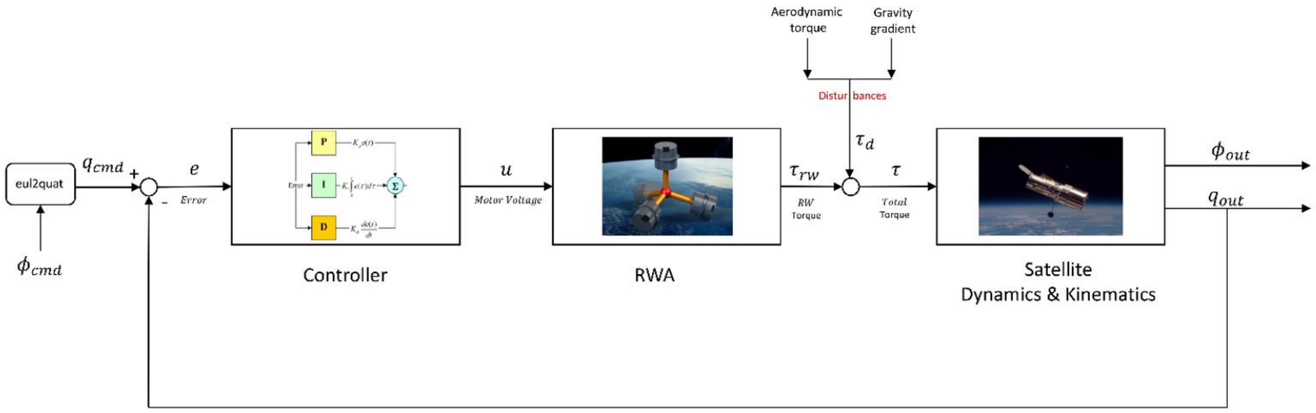

Fig. 4. Satellite attitude control system block diagram.

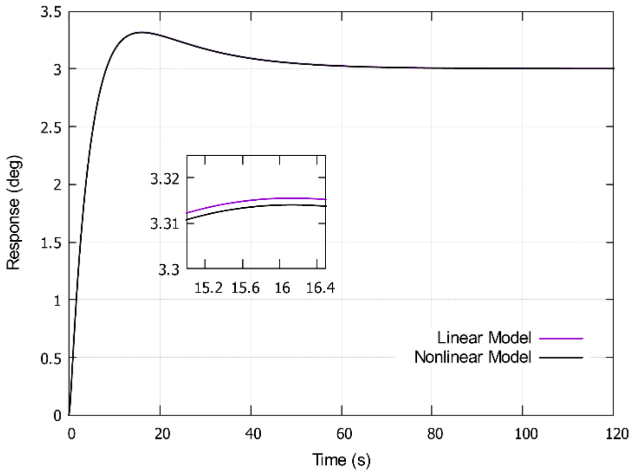

Fig. 5. Responses of the models to 3 degrees of roll input.

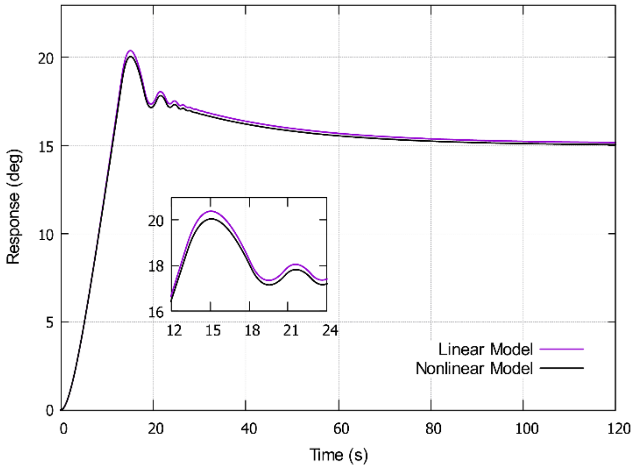

Fig. 6. Responses of the models to 15 degrees of yaw input.

### *Root locus-based tuning*

To execute this method, the root locus of the linearized system was plotted by using the "Control System Designer" tool in the MATLAB/Simulink environment. This Simulink method updates the response of the system at each time for the corresponding root locus change and is able to restrict the manipulation area for the given time domain requirements. The settling time restriction is chosen to be lower than 6 seconds and the percent overshoot is lower than 2%. By manipulating the poles on the root locus to the desired points, a satisfactory output was obtained for the given requirements. The optimal PID parameters for the roll angle determined by the root locus method are 113.554, 0.006, 260.942, and 14.626 for P, I, D, and N coefficients, respectively.

### *Genetic algorithm based tuning*

Genetic Algorithm (GA) is an optimization-based method that can be used for tuning controller parameters. The tuning process of the GA is as follows:

The objective function is chosen as:

$$J = |\tau - \tau_d| + w_{OS}(OS) \tag{19}$$

where $\tau$ is the time constant and $w_{os}$ is the weight of overshoot in the cost function. The function is optimized for the PID controller coefficients $K_p$, $K_i$, $K_d$, and $N$. The PID structure is as follows:

$$K_p + K_i\left(\frac{1}{s}\right) + K_d\left(\frac{Ns}{s+N}\right) \tag{19}$$

The optimization procedure is shown in Fig. 7. It can be described as follows:

1. The first generation is created, i.e., values for $K_p$, $K_i$, $K_d$, and $N$ (search variables) are selected. Initialization of the population.
2. The cost function value for each search variable is calculated.
3. The next generation is created based on crossover mutation functions.
4. Step 2 and Step 3 are repeated until the minimum value for the cost function is reached.

The optimization procedure is run several times with different $w_{os}$ ranging from 100 to 1. Since the overshoot condition is less than 2% and the time constant part of the cost function takes values close to 1, it is decided that the best suited value for $w_{os}$ is 1.

The genetic algorithm is a global optimization method with benefits of being derivative free, suitable for complex, non-convex problems. However, this does not necessarily mean that whenever an optimization problem is constructed with GA, the algorithm will yield the global optimum values. By its nature, GA has the ability not to converge to the local optimum, but this is possible with crossover and especially mutation parameters. In several runs of this specific PID tuning problem, some of the results did not yield a better cost function value than the Root Locus-based tuning method. To overcome this situation, the population number is increased so that the coverage of each generation is increased. The mutation rate is adjusted using the trial-and-error method. When the mutation rate is increased, the population deviates from the previously known minima, which preferably leads to finding new minima. The final tuned values of the controller using GA are 48.894, 0.071, 106.046, and 1.427 for P, I, D, and N coefficients, respectively.

## 3. Results

The controllers obtained by two different methods can be compared in terms of their performance. For this purpose, we use the cost function values of two different tuning methods. By inserting PID values obtained by two methods, it is possible to get a cost function value that is used as a performance criterion to be minimized.

For the Root Locus-based tuning method, we obtained a cost function value of 0.7215, while the cost function value for the GA-based tuning is 0.5129. Since the cost function value obtained from the GA is lower than Root Locus-Based Tuning, it can be said that the PID values calculated by the GA are suitable for the control system. To obtain more realistic results, it is also crucial to compare the results in the presence of a disturbance. To achieve this, the disturbance explained in Section 2.A. was included in the system roll angle. The output of the roll angle for a custom signal input is shown in Fig. 7.

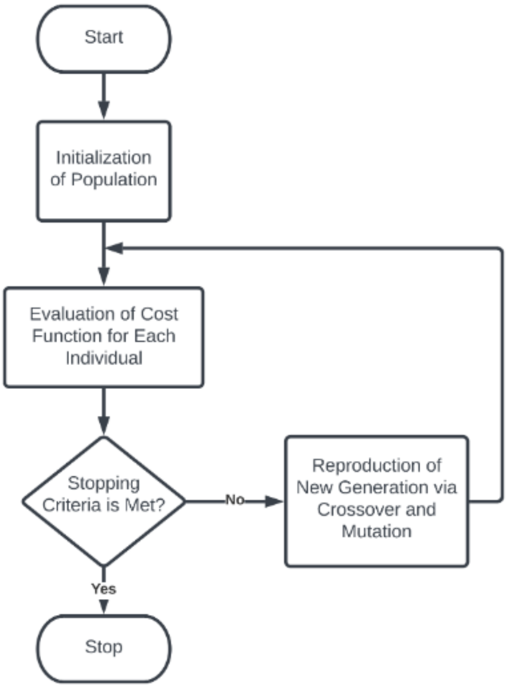
Start
Initialization of Population
Evaluation of Cost Function for Each Individual
Stopping Criteria is Met?
No
Reproduction of New Generation via Crossover and Mutation
Yes
Stop

Fig. 7. Genetic algorithm based PID tuning chart.

## 4. Conclusions and Discussion

All subsystems are built as controller, actuator, and plant using equations of motion. Linearization has been performed and it has been shown that the linearized model can be used to determine the controller parameters both mathematically and visually. The PID parameters are obtained by two methods and their performance is compared in terms of cost function values. It is shown that the GA parameters perform better in terms of the corresponding cost function values. However, it is found that in the presence of disturbance, the root locus-tuned parameters perform better in terms of disturbance rejection as depicted in Fig. 8.

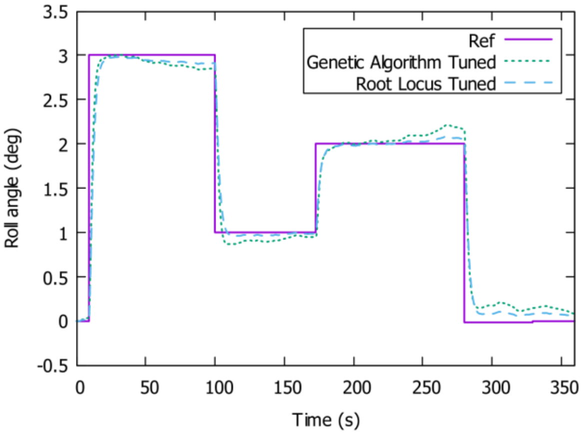

Fig. 8. Comparing tuning methods including external disturbance.

The genetic algorithm is able to optimize complex cost functions, considering the saturations of actuating signals, nonlinear, non-convex phenomena acting in the problem itself. If desired, there is no need to linearize or simplify the system, unlike linear or modern control theory applications. This comes with the cost of being computationally expensive and time-consuming. When optimized tuning is advantageous and even required for a task such as spacecraft control, where pointing accuracy is quite significant, the time and computation burden can be neglected.

## Acknowledgment

This project was financially supported by the Istanbul Technical University (ITU) Scientific Research Projects Coordination Unit Research Start Fund with Project ID: ITUBAP 40957 and Project Code: MAB-2017-40957.

## References

[1] Lohani, S., Joshi, R. (2020). Satellite network security. In *2020 International Conference on Emerging Trends in Communication, Control and Computing (ICONC3)*. IEEE, 1-5. https://doi.org/10.1109/ICONC345789.2020.9117553

[2] Shiga, D. (2010). Happy birthday Hubble: The telescope that almost wasn't. *New Scientist*, 206 (2756), 26-27. https://doi.org/10.1016/S0262-4079(10)60938-4

[3] Dougherty, H., Rodoni, C., Tompetrini, K., Nakashima, A. (1983). Space telescope pointing control. *IFAC Proceedings Volumes*, 16 (11), 15-24. https://doi.org/10.1016/S1474-6670(17)62184-0

[4] Nurre, G. S., Sharkey, J. P., Nelson, J. D., Bradley, A. J. (1995). Preservicing mission, on-orbit modifications to Hubble Space Telescope pointing control system. *Journal of Guidance Control Dynamics*, 18 (2), 222–229. https://doi.org/10.2514/3.21373

[5] Borase, R. P., Maghade, D., Sondkar, S., Pawar, S. (2021). A review of PID control, tuning methods and applications. *International Journal of Dynamics and Control*, 9 (2), 818-827. https://doi.org/10.1007/s40435-020-00665-4

[6] Mohammed, L., Ahmed, M. M. (2014). Spacecraft pitch pid controller tunning using Ziegler Nichols method. *IOSR Journal of Electrical and Electronics Engineering*, 9 (6), 62-67. https://www.iosrjournals.org/iosr-jeee/Papers/Vol9-issue6/Version-1/I09616267.pdf

[7] Joseph, E. A., Olaiya, O. O. (2017). Cohen-Coon PID tuning method: A better option to Ziegler-Nichols PID tuning method. *International Journal of Recent Engineering Research and Development (IJRERD)*, 2 (11), 141-145. http://www.ijrerd.com/papers/v2-i11/29-IJRERD-B576.pdf

[8] Wilson, D. I. (2005). Relay-based PID tuning. *Automation and Control*, 10-12.

[9] Gawthrop, P. (1986). Self-tuning PID controllers: Algorithms and implementation. *IEEE Transactions on Automatic Control*, 31 (3), 201-209. https://doi.org/10.1109/TAC.1986.1104241

[10] Park, D., Yu, H., Xuan-Mung, N., Lee, J., Hong, S. K. (2019). Multicopter pid attitude controller gain autotuning through reinforcement learning neural networks. In *Proceedings of the 2019 2nd International Conference on Control and Robot Technology*. New York, US: Association for Computing Machinery, 80-84. https://doi.org/10.1145/3387304.3387327

[11] Wang, Y., Xiong, J., Lu, Z., Zhang, Y. (2019). The optimization of spacecraft attitude control parameters based on improved particle swarm algorithm. In *2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*. IEEE, 1691-1694. https://doi.org/10.1109/IMCEC46724.2019.8983855

[12] Jia, Y., Yang, X. (2016). Optimization of control parameters based on genetic algorithms for spacecraft attitude tracking with input constraints. *Neurocomputing*, 177, 334-341. https://doi.org/10.1016/j.neucom.2015.11.022

[13] Khoshrooz, A. R., Vahid, D. M., Mirshams, M., Homaeinezhad, M. R., Ahadi, A. H. (2012). Novel method on using evolutionary algorithms for pd optimal tuning. *Applied Mechanics and Materials*, 110-116, 4977-4984. https://doi.org/10.4028/www.scientific.net/AMM.110-116.4977

[14] Daw, M. S., Zayed, A. S., Allafi, N. I., Husain, R. A. (2017). Genetic algorithm based PID controller for attitude control of geostationary satellite. In *International Conference & Exhibition for Geospatial Technologies – Libya GeoTec 2*, 318-330. https://www.lrsgis.org.ly/libyageotec2/ar/paper/34-21-B5-Eng-rev.pdf

[15] Sidi, M. J. (1997). *Spacecraft Dynamics and Control: A Practical Engineering Approach*, Cambridge Aerospace Series No. 7. Cambridge University Press, ISBN 978-0521550727.

[16] Blanke, M., Larsen, M. B. (2010). *Satellite Dynamics and Control in a Quaternion Formulation (2nd edition)*. Technical Report, Technical University of Denmark, Department of Electrical Engineering. https://backend.orbit.dtu.dk/ws/portalfiles/portal/98594729/Satdyn_mb_2010f.pdf

[17] Yang, Y. (2012). Spacecraft attitude determination and control: Quaternion based method. *Annual Reviews in Control*, 36 (2), 198-219. https://doi.org/10.1016/j.arcontrol.2012.09.003

[18] Hasha, M. D. (1987). Passive isolation/damping system for the Hubble Space Telescope reaction wheels. In *NASA-Lyndon B. Johnson Space Center, The 21st Aerospace Mechanisms Symposium*, 211-226. https://ntrs.nasa.gov/citations/19870020440

[19] Wertz, J. R. (ed.) (1978). *Spacecraft Attitude Determination and Control* (Astrophysics and Space Science Library, 73). ISBN 978-9027712042.

[20] Shahgholian, G., Shafaghi, P. (2010). State space modeling and eigenvalue analysis of the permanent magnet DC motor drive system. In *2010 2nd International Conference on Electronic Computer Technology*. IEEE, 63-67. https://doi.org/10.1109/ICECTECH.2010.5479987

[21] Thienel, J. K., Sanner, R. M. (2007). Hubble Space Telescope angular velocity estimation during the robotic servicing mission. *Journal of Guidance, Control, and Dynamics*, 30 (1), 29-34. https://doi.org/10.2514/1.20591

[22] De Guia, N. (2012). *Investigating various propulsion systems for an external attachment for a controlled manual de-orbit of the Hubble Space Telescope*. https://digitalcommons.calpoly.edu/aerosp/63/

[23] Lee, D., Springmann, J., Spangelo, S., Cutler, J. (2011). Satellite dynamics simulator development using lie group variational integrator. In *AIAA Modeling and Simulation Technologies Conference*. AIAA 2011-6430. https://doi.org/10.2514/6.2011-6430

[24] Hur-Diaz, S., Wirzburger, J., Smith, D. (2008). Three axis control of the Hubble Space Telescope using two reaction wheels and magnetic torquer bars for science observations. In *F. Landis Markley Astronautics Symposium*. https://ntrs.nasa.gov/citations/20080023343

[25] Foster, C. L., Tinker, M. L., Nurre, G. S., Till, W. A. (1995). Solar-array-induced disturbance of the Hubble Space Telescope pointing system. *Journal of Spacecraft and Rockets*, 32 (4), 634-644. https://doi.org/10.2514/3.26664

Received August 11, 2022
Accepted March 28, 2023