Design of Linear Quadratic Regulator for the Three-Axis Attitude Control System Stabilization of Microsatellites.

Eze, Charles U., Dr Mbaocha, Christian C., Dr Onojo, James O.

ABSTRACT

In recent years, there have been increased global interests in space-related activities. Attitude determination and control are required in nearly all space missions. In satellites, it is one of the most important subsystems since the accuracy of the satellite mission depends on this subsystem. Mission objectives of satellites may be severely disrupted without correct attitude control. This paper describes the design, analysis and models of attitude control systems (ACS) for the Low-Earth Orbit (LEO) satellites. The mathematical models for dynamic and kinematics associated with these satellites and the designed optimal controller are briefly presented and linearized. With the aid of the powerful Computational tool of MATLAB, a program is developed for the design of the Linear Quadratic Regulator (LQR). The LQR is applied to the 3-axis stabilization and control of Microsatellite using 3-reaction wheels each placed on one axis. Parameters of real Microsatellite are used to test and run the designed LQR controller with MATLAB software which accurately stabilized the attitude of the satellite system. The effects of various control design parameters on the overall system are analyzed and optimum control parameters, Q= diag([8, 8, 8, 8, 8, 8]), R = 0.1*diag([1, 1, 1]) and gain (K), that minimizes the performance index, are obtained. The Satellite system control design specifications of settling time ≤ 10 seconds, power consumption ≤ 0.3 watts and zero steady-state errors (0) are achieved.

Keywords: Attitude Dynamics, Control System, Linear Quadratic Regulator, MATLAB, Microsatellites, Optimal Controller, Reaction Wheel, Three-axis Stabilization.

1. Introduction

In the 21sth century, the use of Low-Earth Orbit (LEO) satellites has increased with the great developments in space technologies. These satellites, ranging from Micro to Nano types, are deployed in orbit for various missions such as telecommunication, weather forecasting, taking images of Earth, ship movement surveillance, obtaining digital elevation maps of disaster areas, environmental tracking of some animals for scientific research and so on, [1]. LEO satellites fly between 600 km. and 1000 km. above the Earth. Satellites are deployed for various missions; hence, the attitude control system of satellites (ACS) is some of the most important subsystems of a satellite. This is because the accuracy of its space mission depends on this subsystem, [2]. The orientation in space with respect to different coordinate systems is referred to as the satellite attitude. Real-time or post-facto knowledge, and maintenance of a desired, specified attitude within a given tolerance in a satellite system is known as the attitude determination and control of satellite system (ADCS), [2]. It is the satellite's visual sense and feeling in space especially in Microsatellites. ADCS is also needed for the pointing of solar arrays in proper direction relative to solar rays in order to absorb maximum energy for the satellite's mission.

The ACS system must maintain attitude control in the presence of constant disturbance torques on the satellite. Disturbance torques are functions of the inertial properties of a satellite and it's orbital location. For Microsatellites in Low-Earth Orbit (LEO), the most common disturbance torques are caused by solar radiation pressure, interaction with Earth's local magnetic field, aerodynamic drag due to Earth's atmosphere, and gravity-gradient torque. These disturbances exert external torques, which build up angular momentum within the satellite, [3].

As a result of all these disturbance torques mentioned above, in time, the satellite tends to drift away from the desired attitude, hence, a suitable control system must be designed to offset these disturbances, [4].

In this paper, a linear quadratic regulator (LQR) is designed and implemented as controllers for the attitude control system (ACS). The actuators used for control are brushless DC three-reaction wheels (3-RWs). The computational tool of MATLAB is used to develop a program for the designed LQR. The effects of various control design parameters on the overall system are analyzed and optimum control parameters (Q and R) that minimizes the performance index, are obtained. The Satellite system control design specifications of settling time ≤ 10 sec, power consumption ≤ 0.3 watts and zero steady-state errors (0) are achieved. The developed program for the designed LQR can be applied to other industrial processes especially in the spacecrafts, navigational and missile control systems. Finally, the stability analysis is conducted using the Nyquist stability criterion to ensure that the designed controller and satellite system are stable.

2.0 Design Objectives.

- Design of a Linear Quadratic Regulator (LQR) controller for 3-axis (ACS) of Microsatellite.
- Analysis of the effects of different Weighing Matrices, (Q and R) on the performance of the designed ACS.

2.1 Performance Index

A performance index is a quantitative measure of the performance of a system and is chosen so that emphasis is given to the important system specifications, [5]. For this design, the following performance indices are given,

1. $J(x,u) = \frac{1}{2}\int_0^\infty (x^T Q x + u^T R u)dt$
2. The settling time is to be ≤ 10 seconds,
3. Power consumption ≤ 0.3 Watts and
4. Zero (0) steady-state error / final value.

3.0 Satellite Attitude Control Kinematics and Dynamic Models.

Mathematical models of physical systems are key elements in the design and analysis of control systems. To understand and control the complex satellite system, a quantitative mathematical model of the system must be derived from basic relationship between system variables.

3.1 Kinematic Equations for Satellite

Kinematics of the satellite describes the orientation of the satellite. The differential equations are given in equations (1 - 2). Detailed information about them can be found in [4, 3, and 1].

$$\dot{\eta} = -\frac{1}{2}\varepsilon^T \omega_{bo}^b \quad (1)$$

$$\dot{\varepsilon} = \frac{1}{2}\eta\omega_{bo}^b - \frac{1}{2}\omega_{bo}^b \times \varepsilon \quad (2)$$

Where $\varepsilon =$ unit Quaternions, $\omega_{bo}^b =$ the angular velocity between the body and reference frames decomposed in the body frame. When equations (1) and (2) are combined, they can be represented as equation (3), where S is the cross product, [1].

$$\dot{q} = \begin{bmatrix} \dot{\eta} \\ \dot{\varepsilon} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} -\varepsilon^T \\ \eta I_{3X3} + S(\varepsilon) \end{bmatrix} \omega_{bo}^b \quad (3)$$

3.2 Dynamic Model of a Satellite

Dynamic equations describe how velocity changes for a given force. According to Newton-Euler formulation presented in [4], [1], angular momentum, H changes according to applied torque. The total angular momentum of the spacecraft is, [2].

$$H_b = I\omega_b + h_w \quad (4)$$

Where I is the inertia matrix of the satellite, ω_b is the angular velocity of satellite in body frame and h_w is the angular momentum of the reaction wheels, which can be described in body frame as follows:

$$h_w = LI_w\omega_w \quad (5)$$

where $I_w = diag(I_{w1}, I_{w2}, \dots I_{wn})$ is the RW diagonal inertia matrix, $[L]_{3\times n}$ is the RW distribution matrix, and $\omega_w = [\omega_{w1} \quad \omega_{w2} \dots \quad \omega_{wn}]^T$ is the angular speed of the wheels. The superscript n, refers to the number of RWs used, [2]. In this paper, 3-RWs are employed at the three-body axis to supply the control torques required to stabilize the satellite. Hence, n=3.

From the Euler's moment equations represented by the angular momentum rate with respect to the body frame, the following dynamics of the satellite can be described, (Sidi, 1997).

$$\frac{d}{dt}H_b = -\omega_b(t) \times H_b(t) + T_d(t) \quad (6)$$

where T_d, represents torques generated on the satellite as a result of disturbances.

From equation (4), the following can be derived;

ISSN 2229-5518

$$\dot{H}_b = \left[\frac{dH}{dt}\right]_b = I\frac{d\omega_b}{dt} + \dot{h}_w \tag{7}$$

Equating equations (6) and (7):

$$I\frac{d\omega_b}{dt} = -\omega_b(t) \times H_b(t) + T_d(t) - \dot{h}_w \tag{8}$$

By substituting equation (4) into (8):

$$\frac{d\omega_b}{dt} = I^{-1}\left[-\omega_b(t) \times (I\omega_b + h_w) + T_d(t) - \dot{h}_w\right] \tag{9}$$

The reaction wheels work on the principle of momentum exchange, therefore the angular momentum produced by reaction wheels are transferred to the satellite with opposite sign, i.e.

$$\dot{h}_w = -T_c \tag{10}$$

where T_c is the command torque, which is determined by the controllers. By substituting (10) into (9), the following dynamic equation can be obtained:

$$\frac{d\omega_b}{dt} = I^{-1}\left[-\omega_b(t) \times (I\omega_b + h_w) + T_d(t) + T_c\right] \tag{11}$$

3.3 Reaction Wheel Torque

The torques generated by the 3-RWs can be defined alongside with the mathematical model of the satellite. Hence, for a 3-axis RW configuration, x, y and z, the torques are generally modeled by the following equation, [1].

$$\tau_r^b = \left(\frac{dL_r}{dt}\right)^b + \omega_{bi}^b \times L_r - \tau_{friction}^b \tag{12}$$

τ_r^b is the torque caused by reaction wheel,

$L_r = [L_{rx} \quad L_{ry} \quad L_{rz}]^T = I_r\omega_r$ is the total moment vector of reaction wheel,

$\tau_{friction}^b$ is the frictional torque caused by wheels and usually assumed to be zero. Then equation (12) yields;

$$\tau_r^b = \left(\frac{dL_r}{dt}\right)^b + \omega_{bi}^b \times L_r = \begin{bmatrix}\tau_{rx}\\ \tau_{ry}\\ \tau_{rz}\end{bmatrix} = \begin{bmatrix}\dot{L}_{rx} + L_{rz}\omega_y - L_{ry}\omega_z\\ \dot{L}_{ry} + L_{rx}\omega_z - L_{rz}\omega_x\\ \dot{L}_{rz} + L_{ry}\omega_x - L_{rx}\omega_y\end{bmatrix} \tag{13}$$

$$\omega_{bi}^b = [\omega_x \quad \omega_y \quad \omega_z]^T \tag{14}$$

The 3-RWs mounted each on one axis are used to apply control torques to rotate and maintain the satellite to the desired orientation. The task here is to develop a robust, rapid and appropriate controller. The Standard Orthogonal 3-wheel configuration is applied in this design and its configurations with mathematical distribution matrices are given in equation (15), [1]. Each column vector represents the distribution of the reaction wheel torques on to the axis of rotation of the satellite. If T_1, T_2, T_3 represent the torques by each RW, then the moments acting on the satellite can be defined as, [1];

$$[T_x \quad T_y \quad T_z]^T = L_{3\times3}[T_1 \quad T_2 \quad T_3]^T \tag{15}$$

Following the kinematics and dynamic mathematical models, the satellite rotation matrix is defined. Rotation matrix can behave as a transformation of a vector represented in one coordinate frame to another frame or as a rotation of a vector within the same frame or as a description of mutual orientation between two frames. The rotation matrix R, from frame $\boldsymbol{a}$ to $\boldsymbol{b}$ is denoted R_a^b. The rotation of a vector from one frame is written with the following notation:

$$x^{to} = R_{from}^{to}x^{from} \tag{16}$$

Angle-axis parameterization is a way of parameterization of the rotation matrix, given in equation (17) as $R_{from}^{to}, R_{\vartheta,\theta}$ corresponding to a rotation θ about the ϑ-axis as defined in [3], [1].

$$R_{\vartheta,\theta} = I + S(\vartheta)\sin\theta + (1-\cos\theta)S^2(\vartheta) \tag{17}$$

where S is the skew-symmetric operator.

From the derived mathematical models of equations (1-13), it can be observed that they are still in their non-linear forms. For the purpose of this design, they must be linearized in order to apply the LQR controller that will be discussed in the subsequent section 4.4. Hence, linearized systems of matrix are defined. The linearization points for angular velocities (ω_{ib}^b), are selected as, [6], [7];

$$\omega_{ib}^b = [\dot{\phi} \quad \dot{\theta} \quad \dot{\psi}]^T + \omega_0[-\psi \quad -1 \quad \phi]^T \tag{18}$$

ω_{ib}^b, which is angular velocities of satellites in body axis ($\omega_{ib}^b = 2\dot{\varepsilon}$) is linearized as, [6], [7];

$$\omega_{ib}^b = \begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix} = \begin{bmatrix}2\dot{\varepsilon}_1 - 2\omega_0\varepsilon_3\\ 2\varepsilon_2 - \omega_0\\ 2\dot{\varepsilon}_3 + 2\omega_0\varepsilon_1\end{bmatrix} = \begin{bmatrix}\dot{\phi} - \psi\omega_0\\ \dot{\theta} - \omega_0\\ \dot{\psi} + \phi\omega_0\end{bmatrix} \tag{19}$$

Hence its time derivation is given as;

$$\dot{\omega}_{ib}^b = \begin{bmatrix}\dot{\omega}_x\\ \dot{\omega}_y\\ \dot{\omega}_z\end{bmatrix} = \begin{bmatrix}2\ddot{\varepsilon}_1 - 2\omega_0\varepsilon_3\\ 2\ddot{\varepsilon}_2\\ 2\ddot{\varepsilon}_3 + 2\omega_0\dot{\varepsilon}_1\end{bmatrix} = \begin{bmatrix}\ddot{\phi} - \dot{\psi}\omega_0\\ \ddot{\theta}\\ \ddot{\psi} + \dot{\phi}\omega_0\end{bmatrix} \tag{20}$$

From equations (19 - 20) it can be observed that the following relations hold between the quanternions, $(\varepsilon_1, \varepsilon_2, \varepsilon_3)$ and the Euler angles, (ϕ, θ, ψ) as,

$$[\phi \quad \theta \quad \psi]^T = [2\varepsilon_1 \quad 2\varepsilon_2 \quad 2\varepsilon_3]^T \tag{21}$$

The Euler angles ϕ, θ, and ψ are defined as the rotational angles about the satellite body axes:Roll ϕ, about the $\boldsymbol{x}$ −axis; Pitch θ, about the $\boldsymbol{y}-$ axis; and Yaw ψ, about the $\boldsymbol{z}-$ axis. The term ω_0 represents the initial orbital angular velocity of the satellite. Hence, applying equations (18 – 20 to equation (11), the following linearized mathematical models are obtained, [6], [7];

$$I\frac{d\omega_{bi}^b}{dt} = \left[-\omega_{bi}^b(t) \times (I\omega_{bi}^b) + \tau_g^b + \left(\frac{dL}{dt}\right)^b\right] \tag{22}$$

Applying the skew symmetric to equation (22) and arranging in component form yields;

$$I_x\ddot{\phi} = \phi\left[4\omega_0^2(I_z - I_y) - \omega_0\dot{\theta}(I_z - I_y)\right] + \dot{\theta}\dot{\psi}\left(\left(I_y - I_z\right)\right) + \dot{\psi}\omega_0\left(I_z - I_y + I_z\right) + \left(\dot{L}_{rx}\right) \tag{23a}$$

$$I_y\ddot{\theta} = 3\omega_0^2(I_x - I_z)\theta + \phi[\psi\omega_0^2(I_x - I_z) + \dot{\phi}\omega_0\left(I_z - I_x\right)] + \dot{\psi}\psi\omega_0\left(I_x - I_z\right) + \dot{\psi}\dot{\phi}(I_z - I_x) + \left(\dot{L}_{ry}\right) \tag{23b}$$

$$I_z\ddot{\psi} = \psi\left[\omega_0^2\left(I_x - I_y\right) + \dot{\theta}\omega_0\left(I_y - I_x\right)\right] + \dot{\phi}[\omega_0\left(I_y - I_x - I_z\right)] + \dot{\theta}\left(I_x - I_y\right)] + \left(\dot{L}_{rz}\right) \tag{23c}$$

The state-space equation for this system represented in linear form of equation (24) can be derived by defining the following states;

$$x = [\dot{\phi} \quad \dot{\theta} \quad \dot{\psi} \quad \ddot{\phi} \quad \ddot{\theta} \quad \ddot{\psi}] \text{ and } u = [\dot{L}_{rx} \quad \dot{L}_{ry} \quad \dot{L}_{rz}]^T \tag{24a}$$

$$\dot{x}(t) = Ax(t) + B(t)u(t) \quad , \quad y(t) = Cx(t) + Du(t) \tag{24b}$$

As a result, equation (23) results to equations (25) and (26);

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{4\omega_0^2(I_z-I_y)}{I_x} & 0 & 0 & 0 & 0 & \frac{\omega_0(I_z-I_y+I_z)}{I_x} \\ 0 & \frac{3\omega_0^2(I_x-I_z)}{I_y} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\omega_0^2(I_x-I_y)}{I_z} & \frac{\omega_0(I_y-I_x-I_z)}{I_z} & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \theta \\ \psi \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{I_x} & 0 & 0 \\ 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & \frac{1}{I_z} \end{bmatrix} \begin{bmatrix} \dot{L}_{rx} \\ \dot{L}_{ry} \\ \dot{L}_{rz} \end{bmatrix} \tag{25}$$

Output equation,

$$\begin{bmatrix} \phi \\ \theta \\ \psi \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \phi \\ \theta \\ \psi \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + 0 \tag{26}$$

Where $A =$

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{4\omega_0^2(I_z-I_y)}{I_x} & 0 & 0 & 0 & 0 & \frac{\omega_0(I_z-I_y+I_z)}{I_x} \\ 0 & \frac{3\omega_0^2(I_x-I_z)}{I_y} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\omega_0^2(I_x-I_y)}{I_z} & \frac{\omega_0(I_y-I_x-I_z)}{I_z} & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{I_x} & 0 & 0 \\ 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & \frac{1}{I_z} \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad D = 0$$

3.4 Linear Quadratic Regulator Controller Design

In this paper, LQR is the control technique used in attitude stabilization. It is a powerful control technique for designing linear controllers for complex systems that have stringent performance requirements. The main idea of the control system is to find a cost function and minimize this cost function. After the cost function is minimized, the system states are fed back by a gain-matrix. In (Wisniewski, et al, 1996; Hall, 2002; Lewis, 1998) [8], [6], and [9], LQR technique is further explained in detail.

The linearized and time invariant systems in equations (25 and 26) are applied in the control design. The optimization problem consists of finding a linear control law of the type, [8], [6];

$$u(t) = -Kx(t) \tag{27}$$

where k is the feedback gain-matrix. To find the control signal 'u' that minimizes the cost function, the performance index (PI) is defined:

$$J(x, u) = \frac{1}{2}\int_0^\infty (x^T Qx + u^T Ru)dt \tag{28}$$

Substituting (27) into equation (28) yields;

ISSN 2229-5518

$$J = \frac{1}{2}\int_0^{\infty} x^T(Q + K^T RK)dt \qquad (29)$$

The feedback gain matrix K has the form,

$$K = R^{-1}B^T P \qquad (30)$$

where P is the solution of algebraic Riccati equation given in (31):

$$A^T P + PA + Q - PBR^{-1}B^T P = 0 \qquad (31)$$

where $Q \geq 0, R > 0, P \geq 0$ are symmetric, positive definite and semi-positive matrices respectively defined as state and control Weighting matrices,

$$Q = diag[Q_1, \quad Q_2, \quad \dots \quad Q_{ns},] \qquad (32a)$$

$$R = diag[R_1, \quad R_2, \quad \dots \quad R_{na},] \qquad (32b)$$

Where n_s, is the number of the states and n_a, is the number of actuators, [8].

The selection of Q and R matrices are performed by adjusting them in the developed MATLAB code until the desired performance is achieved. There are several procedures for solving algebraic Riccati equation. The feedback gain matrix, K is calculated in MATLAB using the syntax command in (33);

$$[K, P, E] = lqr(A, B, Q, R) \qquad (33)$$

$lqr(A, B, Q, P)$ calculates the optimal gain matrix K such that the state-feedback law, $u = -Kx$, minimizes the quadratic cost function in equation (29) for the state-space model. T is taken as the simulation time and t_0 is taken as zero in equation (29). The MATLAB syntax; *lqr* also returns the solution P of the Ricatti equation given in equation (31) and the closed-loop eigenvalues *E=eig(A-B*K).* The block diagram showing the optimal configuration of the designed system is shown in Figure 3.1.

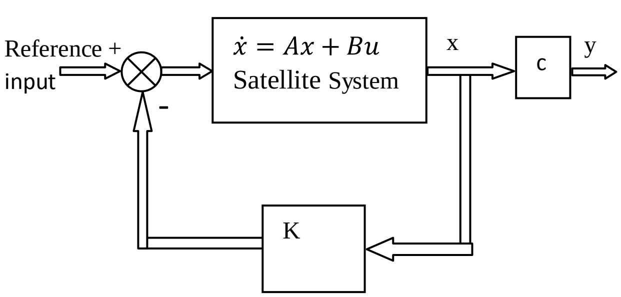

Fig 3.1. Block Diagram of LQR optimal controller for the satellite system.

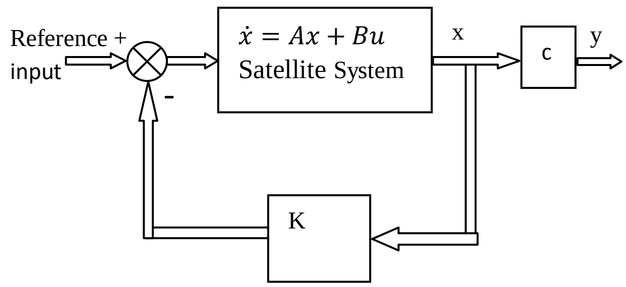

Fig 3.1. Block Diagram of LQR optimal controller for the satellite system.

Table 3.1 presents the parameters for satellite system simulation.

TABLE 3.1. Parameters for Satellite Simulations, [1].

Satellite parameters	**Values**	**Units**
Satellite weight	120	Kg
Satellite Inertia Matrix	$I_x = 9.8194$, $I_y = 9.7030$, $I_z = 9.7309$	Kgm^2
Orbit	686 (LEO orbit)	Km
Orbit angular velocity, ω_0	0.0010764	Rads/sec
Initial Roll angle, $[\phi_0]$	3	Degrees
Initial Pitch angle, $[\theta_0]$	1	Degrees
Initial Yaw angle, $[\psi_0]$	1	Degrees
Initial angular velocities $[\phi, \quad \theta, \quad \psi]$	[0 0 0]	Rads/sec

Figure 3.2 presents the algorithm adopted for the LQR controller design.

4.0 Results and Discussion

The choices of Weighting Matrices affect the Control system performance. Figure 4.1 show the effects of different Weighting Matrices, Q and R on the control system performance. Firstly, the Q-Matrix is kept constant at Q1= diag([1, 1, 1, 1, 1, 1]) while the R-Matrix is varied between R1a = 0.1*diag([1, 1, 1,]) to R1d = diag([10, 10, 10]). The resulting responses of the satellite angular velocities for the 3-axis are show in figures 4.1a to 4.1c. .

MATLAB programs developed for the design of the LQR are given in Appendix A.

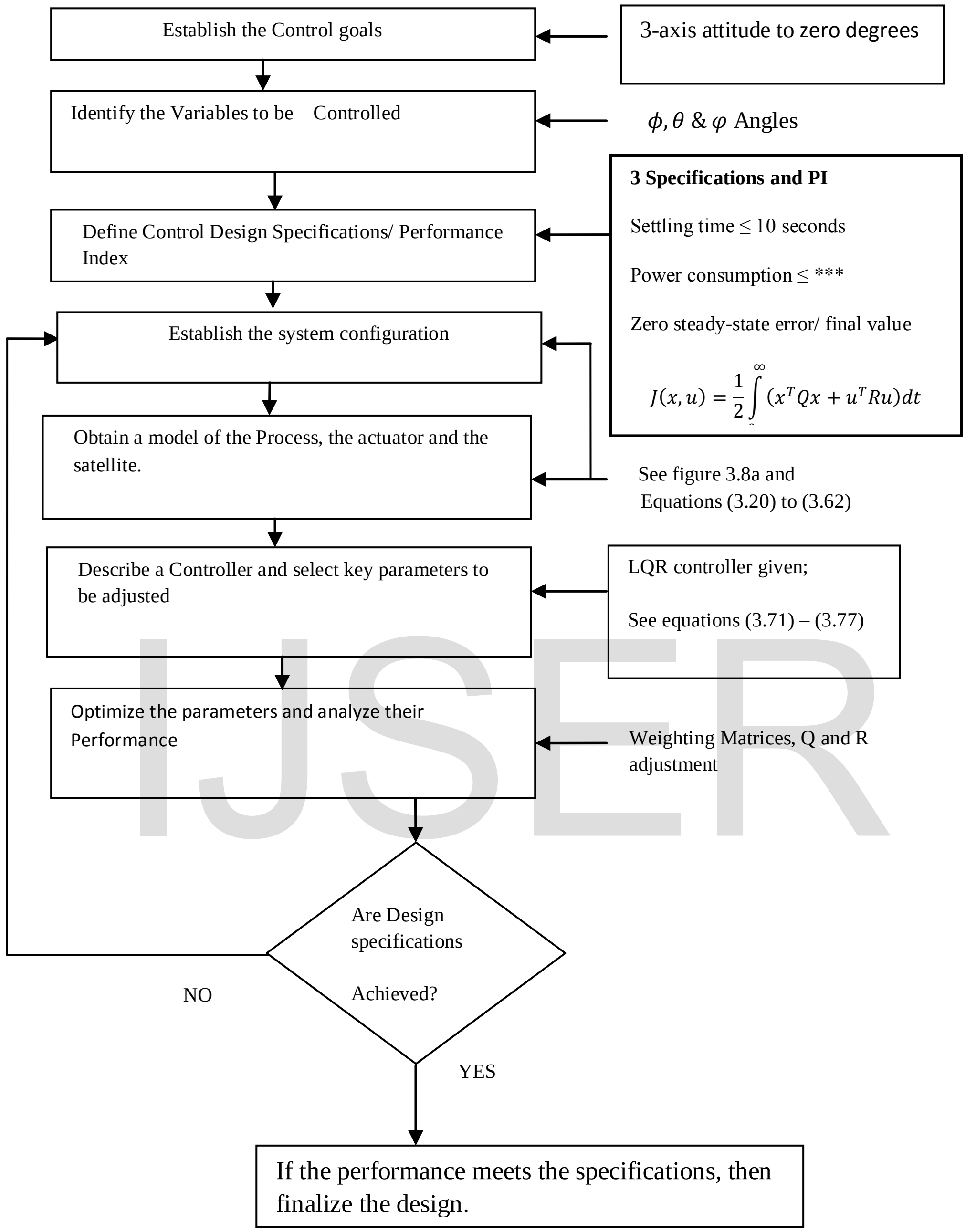

Fig 3.2. LQR controller design algorithm for the ACS.

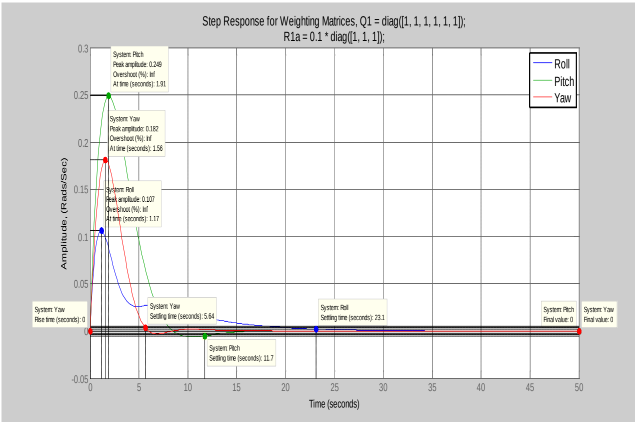

Fig 4.1a. Step response with Weighting Matrices Q1= diag([1, 1, 1, 1, 1, 1]) and R1a = 0.1*diag([1, 1, 1,])

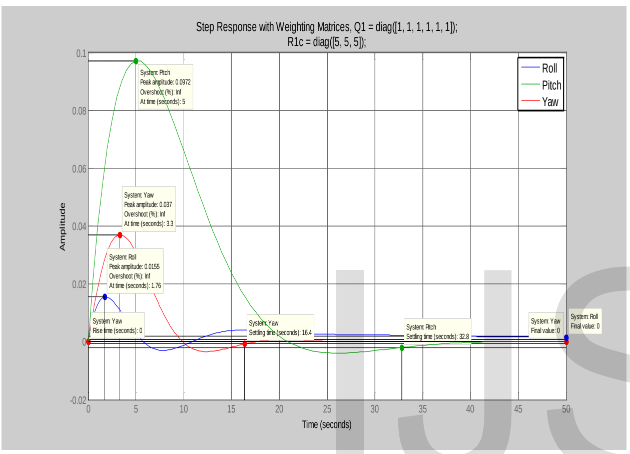

Fig 4.1b. Step response with Weighting Matrices Q1= diag([1, 1, 1, 1, 1, 1]) and R1c = diag([5, 5, 5]).

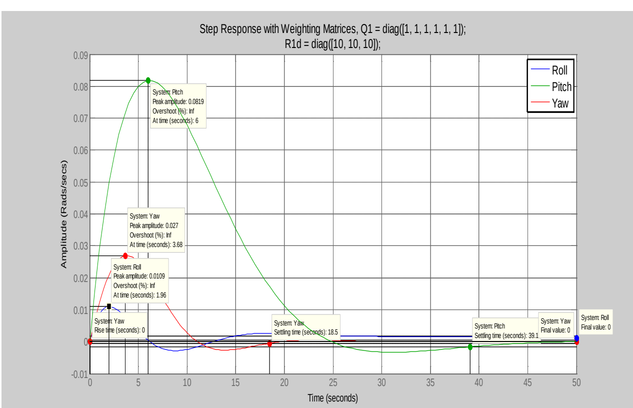

Fig 4.1c. Step response with Weighting Matrices Q1= diag([1, 1, 1, 1, 1, 1]) and R1d= diag([10, 10, 10]).

Again, the Weighting Matrix R is kept constant while the Q-Matrix is varied from Q2= diag([2, 2, 2, 2, 2, 2]) to Q5= diag([8, 8, 8, 8, 8, 8]). The effects are seen on the response of the system as shown in the plots of figures 4.2a to 4.2c.

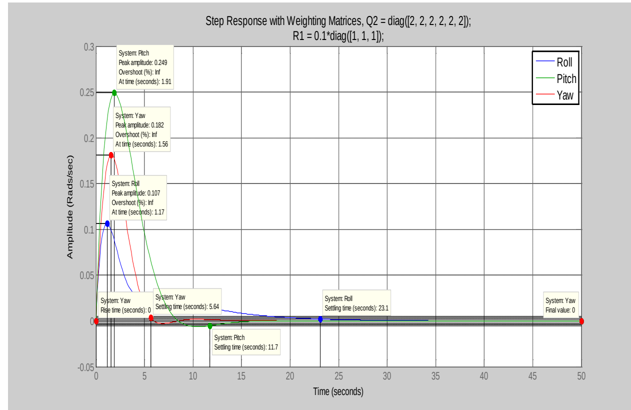

Fig 4.2a. Step response with Weighting Matrices Q2= diag([2, 2, 2, 2, 2, 2]) and R1 = 0.1*diag([1, 1, 1]).

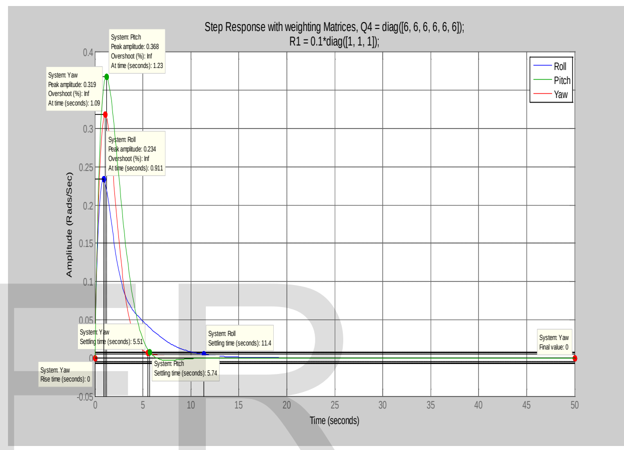

Fig 4.2b: Step response with Weighting Matrices Q4= diag([6, 6, 6, 6, 6, 6]) and R1 = 0.1*diag([1, 1, 1]).

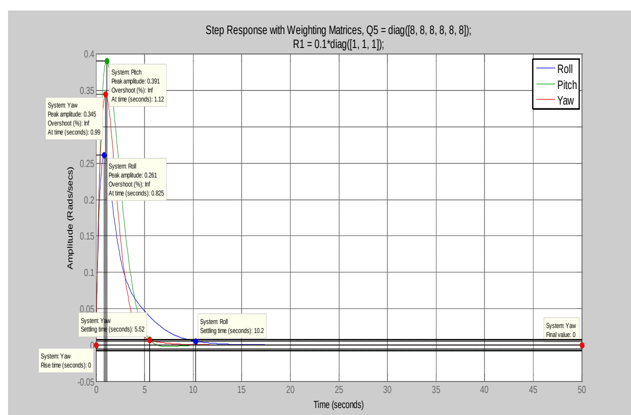

Fig4.2c. Step response with Weighting Matrices Q5= diag([8, 8, 8, 8, 8, 8]) and R = 0.1*diag([1, 1, 1]).

The resulting settling time, steady-state error/final value and amplitudes of the angular velocities for the 3-body axis; Roll ϕ, Pitch θ, and Yaw ψ, for different control parameters/Weighting Matrices are presented in table 4.1.

ISSN 2229-5518

From table 4.1, it can be observed that keeping the Weighting matrices Q, constant while increasing R results in a more robust system response. These are observed with the decreasing peak amplitude from 0.107 rads/sec to 0.0109 rads/sec, from 0.249 rads/sec to 0.0819 rads/sec and from 0.0819 rads/sec to 0.027 rads/sec for the Roll, Pitch and Yaw-axis respectively. However, there are poor transient responses of the system as the settling time of the satellite body axes increases. These are observed in the increase of the settling time from 23.1sec to 148.5sec, from 11.7sec to 39.1sec, and from 5.6sec to 18.5sec for the Roll, Pitch and Yaw-axis respectively.

Secondly, the effects of varying Q-matrix while keeping R-matrix constant are investigated and their various time responses presented in table 4.2. From table 5.2, it can be observed that increasing Q-matrix while keeping R-matrix constant results to better transient performance of the satellite system. Decreases in the 3-body axis settling time from (23.13 to 10.2) seconds, from (11.72 to 5.51) seconds and from (5.64 to 5.52) seconds for Roll, Pitch and Yaw axes respectively are observed.

TABLE 4.1. Control Parameters with Their Resulting Time Responses; Keeping Q1 constant at Q1 = diag([1, 1, 1, 1, 1, 1]); while varying R-matrix .

Control Parameters (Q and R)	States/ Euler Angles	Settling Time (Sec)	Steady - State Error/ Final value	Peak Amplitude (Rads/ sec)
Q1 = diag([1, 1, 1, 1, 1, 1]); R1a = 0.1 * diag([1, 1, 1]);	Roll, $\ddot{\phi}$	23.1	0	0.107
	Pitch, $\ddot{\theta}$	11.7	0	0.249
	Yaw, $\ddot{\psi}$	5.6	0	0.182
Q1= diag([1, 1, 1, 1, 1, 1]) and R1c = diag([5, 5, 5]).	Roll, $\ddot{\phi}$	113.9	0	0.0156
	Pitch, $\ddot{\theta}$	32.8	0	0.0972
	Yaw, $\ddot{\psi}$	16.4	0	0.037
Q1= diag([1, 1, 1, 1, 1, 1]) and R1d = diag([10, 10, 10]).	Roll, $\ddot{\phi}$	148.5	0	0.0109
	Pitch, $\ddot{\theta}$	39.1	0	0.0819
	Yaw, $\ddot{\psi}$	18.5	0	0.027

From the results of the plots of figures4.1, and 4.2, and tables 4.1 and 4.2, it is observed that the optimum control parameters occur with the Weighting matrices; *Q5= diag([8, 8, 8, 8, 8, 8]) and R1 = 0.1*diag([1, 1, 1])* with the fastest settling time of 10.2sec, 5.51 sec and 5.52sec for Roll, Pitch and Yaw axes respectively. The peaks are 0.284rads/sec, 0.409rads/sec and 0.365rads/sec for Roll, Pitch and Yaw axes respectively. Equivalent torques generated on the satellite 3-body axes are shown in figure 4.3.

TABLE 4.2. Control Parameters with Their Resulting Time Responses; Keeping R constant at R = 0.1*diag([1, 1, 1]) while varying Q-matrix.

Control Parameters (Q and R)	States/ Euler Angles	Settling Time (Sec)	Steady-States Error/ Final value	Peak Amplitude (Rads/ sec)
Q2= diag([2, 2, 2, 2, 2, 2]) and R1 = 0.1*diag([1, 1, 1]).	Roll, $\ddot{\phi}$	23.13	0	0.107
	Pitch, $\ddot{\theta}$	11.72	0	0.249
	Yaw, $\ddot{\psi}$	5.64	0	0.182
Q4= diag([6, 6, 6, 6, 6, 6]) and R1 = 0.1*diag([1, 1, 1]).	Roll, $\ddot{\phi}$	11.43	0	0.235
	Pitch, $\ddot{\theta}$	5.74	0	0.368
	Yaw, $\ddot{\psi}$	5.51	0	0.319
Q5= diag([8, 8, 8, 8, 8, 8]) and R1 = 0.1*diag([1, 1, 1]).	***Roll, $\ddot{\phi}$***	***10.2***	***0***	***0.261***
	Pitch, $\ddot{\theta}$	***5.51***	***0***	***0.391***
	Yaw, $\ddot{\psi}$	***5.52***	***0***	***0.345***

As can be observed from figure 4.3, the torques on the satellite 3-body axes generated by the reactions are; 0.275Nm, 0.613Nm and 0.474Nm for the Roll, Pitch and Yaw axes respectively.

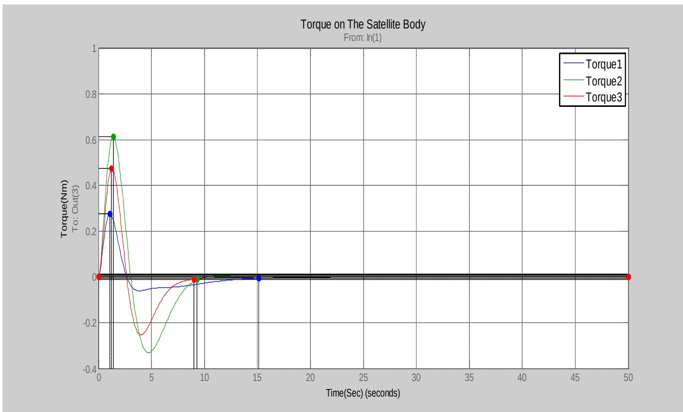

Fig 4.3. Torque generated on the satellite 3-body axis.

The equivalent power consumption corresponding to the maximum torque, (0.613Nm) is,

$$Power = Torque*Angular\ velocity\ (Watts) \qquad (34)$$

$$= 0.613*0.391 = 0.239\ Watts.$$

The corresponding optimal gain matrix, *[K]* from the state feedback law as defined in equation (27), *u=-kx,* that minimizes the quadratic cost function in equation (29) is obtained with the MATLAB code of equation (40);

$$>>\ [K, P, E] = lqr(A, B, Q5, R) \qquad (35)$$

The following results are obtained for the gain, *[K]* the solution of Ricatti equation, *(S),* and the Eigenvalues, of the closed – loop system, *(E=eig(A-B*K))* as ;

K =

$$\begin{matrix} 7.9092 & 0.0000 & -4.1766 & 21.1629 & 0.0000 & -7.1351 \\ 0.0000 & 8.9443 & 0.0000 & 0.0000 & 15.9272 & -0.0000 \\ 4.1766 & -0.0000 & 7.9092 & -7.2050 & -0.0000 & 13.5099 \end{matrix} \qquad (36)$$

P =

$$\begin{matrix} 21.4009 & 0.0000 & -0.0091 & 7.7542 & 0.0000 & 4.0549 \\ 0.0000 & 14.2457 & 0.0000 & 0.0000 & 8.6838 & -0.0000 \\ -0.0091 & 0.0000 & 13.6697 & -4.0947 & 0.0000 & 7.6789 \\ 7.7542 & 0.0000 & -4.0947 & 20.7479 & 0.0000 & -6.9952 \\ 0.0000 & 8.6838 & 0.0000 & 0.0000 & 15.4633 & -0.0000 \\ 4.0549 & -0.0000 & 7.6789 & -6.9952 & -0.0000 & 13.1164 \end{matrix} \qquad (37)$$

E =

-1.1160 + 0.9383i, -1.1160 - 0.9383i, -0.8565 + 0.0000i

-0.4616 + 0.0000i, -0.8202 + 0.4984i, -0.8202 - 0.4984i (38)

To further ascertain the validity and stability of the designed controller, the Nyquist plots of the entire satellite system with the designed controller are plotted in figures 4.4.

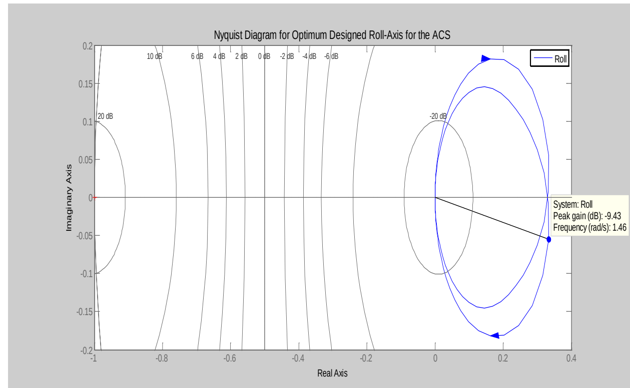

Fig 4.4a. Nyquist Plot for Optimum Designed Roll-Axis for the ACS.

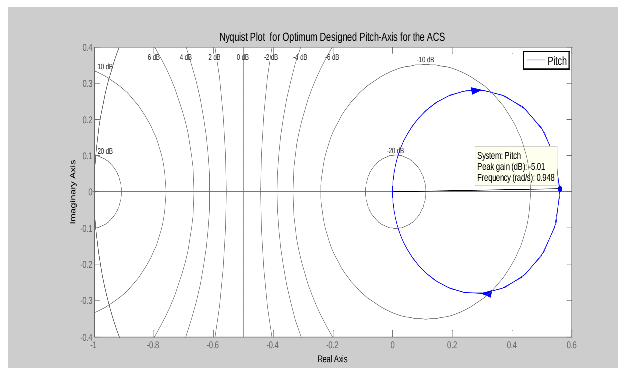

Fig 4.4b. Nyquist Plot for Optimum Designed Pitch-Axis for the ACS.

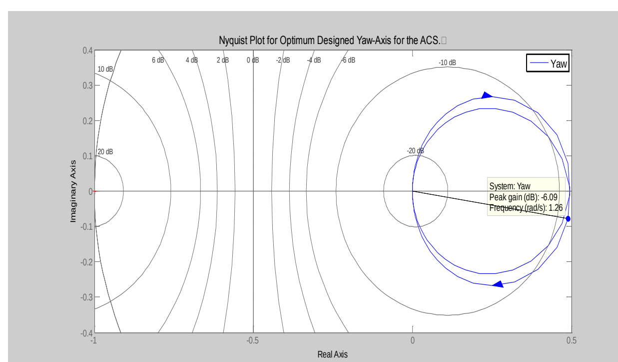

Figure 4.4c: Nyquist Plot for Optimum Designed Yaw-Axis for the ACS

From the Nyquist plots, it is observed that the -1 point on the Real-axis for the Roll, Pitch and Yaw-axis are not circled; hence the designed LQR controller and system are stable.

ISSN 2229-5518

5.0 Conclusion

In an effort to reduce settling time, steady-state error and power consumption of a LEO Microsatellite, an LQR controller is designed and analyzed for the attitude control system. For the designed LQR, it is demonstrated that by careful adjustment of the Weighting matrices, Q and R, the system performance (settling time, steady-state error, and power consumptions) can be modified to achieve design specifications. This agrees with previous researches on LQR by, [8], [6] and [9]. A MATLAB program is developed for the design of the LQR.

Finally, the designed Optimal LQR controller is able to meet the design goals; settling time ≤ 10 seconds, power consumption ≤ 0.3watts and zero steady-state error / final value. The LQR controller designed in this paper can be implemented in the control of other dynamic industrial plants processes, especially in Aircrafts, Navigation and Missile control systems.

6.0 Acknowledgement

The Authors wish to thank Sir $ Lady Abanulor, S. N. of Better life Zion Industry, Aba, Abia, Nigeria for their financial support in the publication of this research material.

References

[1] S. Karataş. "LEO Satellites: Attitude Dynamics and Some Nonlinear Control Techniques", M.Sc. Thesis in Department of Electrical and Electronics Engineering, Middle East Technical University, METU, 2006.

[2] M. J. Sidi . "Spacecraft Dynamics and Control: A Practical Engineering Approach", Cambridge Press, 2007.

[3] J. R. Wertz, & W. J. Larson. "Space Mission Analysis and Design; Space Technology Series", Hawthorne, CA: Microcosm Press, third edition, 1999.

[4] E. J. Overby. "Attitude Control for the Norwegian Student Satellite N-Cube", M.Sc. Thesis in NTNU, 2004.

[5] C. M. Close & D. K. Frederick. "Modeling and Analysis of Dynamic Systems", Boston: Houghton Mifflin, 2nd ed., 1993.

[6] C.D. Hall. "Spacecraft control and dynamics", *AOE4140 Lecture Notes, Aerospace and Ocean Engineering Virginia Tech,2002 .*

[7] M. R. Stefanescu & C. L. Prioroc . "Automatic Control Systems for ESMO Satellite Using LQR Control Strategy", *1st IAA Conference on Dynamics and Control of Space Systems, 2012.*

[8] R. Wisniewski & M. Blanke. "Satellite Attitude Control Using Only Electromagnetic Actuation", Ph.D. Thesis, Department Of Control Engineering, Aalborg University, Denmark, 1996.

[9] F.L. Lewis. "Linear Quadratic Regulator (LQR) State Feedback Design", Arlington: University of Texas, 1998.

Appendix

MATLAB Program Codes.

%% System matrices for the entire plant: 3 inputs ($\phi, \theta and\ \psi$), 3 outputs ($\dot{\phi}, \dot{\theta}\ and\ \dot{\psi}$)) without controller %

```
A = [ 0 0 0 1 0 0;0 0 0 0 1 0;0 0 0 0 0 1;

1.807*10^-8 0 0 0 0 1.07*10^-3;0 3.171*10^-8 0 0 0 0;

0 0 1.989*10^-8 -1.892 0 0 ];

B = [ 0 0 0;0 0 0;0 0 0; 0.102 0 0; 0 0.103 0; 0 0 0.103];

C = [ 0 0 0 0 0 0;0 0 0 0 0 0;0 0 0 0 0 0;

   0 0 0 1 0 0;0 0 0 0 1 0;0 0 0 0 0 1 ];

D = [ 0 0 0;0 0 0;0 0 0; 0 0 0;0 0 0;0 0 0];
```

%% Construct inputs and outputs corresponding to steps in *xi* and *Cxi* positions *(i=1, 2, 3)* %%

%% The vectors *x1d, x2d and x3d* correspond to the states that are the desired equilibrium states for the system. The matrices *Cx1, Cx2 and Cx3* are the corresponding outputs%%%The way these vectors are used is to compute the closed-loop system dynamics as; %%

% x-dot = Ax + B u =>x-dot = (A-BK) x + K xid %

% u = -K(x - xid); y = Cxi %

%% The closed-loop dynamics can be simulated using the "step" command, with *K*xid* as the input vector (assumes that the "input" is unit size, so that *xid* corresponds to the desired steady-state) %%

```
x1d = [1; 0; 0; 0; 0; 0]; Cx1 = [0 0 0 1 0 0];

x2d = [0; 1; 0; 0; 0; 0]; Cx2 = [0 0 0 0 1 0];

x3d = [0; 0; 1; 0; 0; 0]; Cx3 = [0 0 0 0 0 1];

% Start with a diagonal weighting%

Q1 = diag([1, 1, 1, 1, 1, 1]);

R1a = 0.1 * diag([1, 1, 1]);

K1a = lqr(A, B, Q1, R1a);
```

% Close the loop: *x-dot = Ax + B K (x-xid)* %

```
Roll = ss(A-B*K1a,B(:,1)*K1a(1,:)*x1d,Cx1,0);

Pitch = ss(A-B*K1a,B(:,2)*K1a(2,:)*x2d,Cx2,0);

Yaw = ss(A-B*K1a,B(:,3)*K1a(3,:)*x3d,Cx3,0);

figure (4.1a); step(Roll, Pitch, Yaw,50);grid;
```

% Look at different input (R1b) Weightings keeping Q1 constant%

```
Q1 = diag([1, 1, 1, 1, 1, 1]);

R1c = diag([5, 5, 5]);

K1b = lqr(A, B, Q1, R1c);
```

% Close the loop: xdot = Ax + B K (x-xd)%

```
Roll = ss(A-B*K1b,B(:,1)*K1b(1,:)*x1d,Cx1,0);

Pitch = ss(A-B*K1b,B(:,2)*K1b(2,:)*x2d,Cx2,0);

Yaw = ss(A-B*K1b,B(:,3)*K1b(3,:)*x3d,Cx3,0);

figure(4.1b); step(Roll, Pitch, Yaw,50);grid;
```

% Look at different input (Q4), Weightings keeping R1 constant%

```
Q4 = diag([6, 6, 6, 6, 6, 6]);

R1 = 0.1*diag([1, 1, 1]);

K4 = lqr(A, B, Q4, R1);

% Close the loop: xdot = Ax + B K (x-xd)

Roll = ss(A-B*K4,B(:,1)*K4(1,:)*x1d,Cx1,0);

Pitch = ss(A-B*K4,B(:,2)*K4(2,:)*x2d,Cx2,0);

Yaw = ss(A-B*K4,B(:,3)*K4(3,:)*x3d,Cx3,0);

figure(4.2b); step(Roll, Pitch, Yaw,50);grid
```

% Optimum Weighting Matrices, (Qd, Rd) %

```
Q5 = diag([8, 8, 8, 8, 8, 8]);

R1 =0.1* diag([1, 1, 1]);

Kd = lqr(A, B, Q5, R1);

% Close the loop: xdot = Ax + B K (x-xd)%

Roll = ss(A-B*Kd,B(:,1)*Kd(1,:)*x1d,Cx1,0);

Pitch = ss(A-B*Kd,B(:,2)*Kd(2,:)*x2d,Cx2,0);

Yaw = ss(A-B*Kd,B(:,3)*Kd(3,:)*x3d,Cx3,0);

figure(4.2c); step(Roll, Pitch, Yaw,50);grid;
```

%Compute Torque on the satellite body%

```
Torque1=(K5*(Roll^2));
```

```
Torque2=(K5*(Pitch^2));

Torque3=(K5*(Yaw^2));

step(Torque1,Torque2,Torque3,50);grid

xlabel('Time(Sec)'),ylabel('Torque(mNm)')

title('Torque on the Satellite Body');grid
```

Authors:

Dr Mbaocha, C. C: Is a PhD holder and a senior lecturer at the Federal University of Technology, Owerri (FUTO). His areas of interest are in Control Systems Engineering, Aircraft and Satellite Control Designs.

Email: christian_ij@yahoo.com

Dr Onojo, J. O.: Is a PhD holder and a senior lecturer in the department of Electrical and Electronic Engineering, Federal University of technology, Owerri.

Email: jamesonojo@yahoo.com

Eze, Charles U.: obtained his B.Eng degree in Electrical Engineering, from the University of Nigeria, Nsukka, 2012. He is currently an M.Eng student (Control Systems) of Electrical and Electronic Engineering, FUTO, Nigeria. His areas of interest include; Efficient Energy Management Systems, Renewable Energy Resources, Satellite Control System Design, System Identification and Control.

Email: ezecharlesugo1@gmail.com