

Pitch Control of a Rocket with a Novel LQG/LTR Control Algorithm

Aliyu Bhar Kisabo, Aliyu Funmilayo Adebimpe and Sholiyi Olusegun Samuel

Centre for Space Transport and Propulsion (CSTP), Epe Lagos-State Nigeria

Article history

Received: 06-02-2019

Revised: 11-02-2019

Accepted: 11-03-2019

Corresponding Author:

Bhar Kisabo Aliyu

Centre for Space Transport
and Propulsion (CSTP) Epe
Lagos-State Nigeria

E-mail: aliyukisabo@yahoo.com

Abstract: This paper presents the design, simulation and analysis of a novel LQG and LQG/LTR control algorithm for the pitch angle of a sounding rocket. These improved LQG and LQG/LTR control algorithms stem from the fact that a Riccati Differential Equation (RDE) rather than the popular Algebraic Riccati Equation (ARE) is used to obtaining the Kalman gain in the observer of the traditional Linear Quadratic Gaussian (LQG) control algorithm. Thus, eight (8) different controllers were design, simulated and analysed, three (3) of such controllers are novel and two out of these novel controllers were able to recover completely the robustness lost in the traditional LQG controller. All controllers synthesized were analysed using time response characteristics of closed-loop system and compared with the LQR and LQG control system. Using the LQR controller as the benchmark for best performance and the LQG as the worst. This study shows an application option that demonstrates optimal control system design in MATLAB/Simulink® and the approach put forward here proves to be very effective.

Keywords: Rocket, LQG Control, LQG/LTR Control, Differential Riccati Equation

Introduction

Classical control system design is generally a trial-and error process in which various methods of analysis are used iteratively to determine the design parameters of an “acceptable” system. Acceptable performance is generally defined in terms of time and frequency domain criteria such as *rise time*, *settling time*, *peak overshoot*, *gain* and *phase margin*, and *band width*. To meet the demands of modern technology, different performance criteria must be satisfied, in a complex multiple-input multiple-out systems requirement. For example, the design of a spacecraft attitude control system that minimizes fuel expenditure is not amenable to solution by classical methods. A new and direct approach to the synthesis of these complex systems, called optimal control theory, has been made feasible by the development of digital computer.

The objective of the optimal control theory is to determine the control signals that will cause a process to satisfy the physical constraints and at the same time minimize (or maximize) some performance criterion. In certain cases, the problem statement may clearly indicate what to select for a performance measure (Kirk, 1998).

A lot of work has been done in the area of optimal controller design for aerospace vehicles (Jianqiao *et al.*, 2011; Das and Halder, 2014; Moshen Ahmed *et al.*, 2011; Liu, 2017; Zhang *et al.*, 2016; Nair and Harikumar, 2015) but non derives it *Kalman gain* from a *differential reccati equation* as we will demonstrate in this study. Also, in previous works, the synthesied LQG/LTR (Barzаноoni, 2015; Barbosa *et al.*, 2016; Ishihara and Zheng, 2017) did not restore completely the robustensens of the LQR or that of the Kalman filter.

The first stage of any control system theory is to obtain or formulate the system dynamics. There refers to *modeling* in terms of dynamical equations such as differential or difference equations. With such equations, the system is called the plant. This aspect has been addressed fully in Chapter One. In this Chapter, the realized plant will be used to design and analyse Optimal control algorithms (Brian and Moore, 1989) in MATLAB/Simulink®.

This chapter is divided into five sections. Section two presents the adotped mathematical model. In section three, stability analysis was done for the rocket mathematical model. Optimal Control theory was introduced in section four hence, the design of LQR, LQG and their two novel variants. Also, LQG/LTR was

introduced here with a design for a novel variant of it. Results were discussed in section five before the conclusion section.

Rocket Model

The modern control theory concerned with Multiple Inputs and Multiple Outputs (MIMO) is based on state variable representation in terms of a set of first order differential (or difference) equations. Here, the system (plant) is characterized by state variables, say, in linear, time invariant form as:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \quad (1)$$

where, dot denotes differentiation with respect to (w.r.t.) t , $x(t)$, $u(t)$ and $y(t)$ are n , r and m dimensional *state*, *control* and *output* vectors respectively and A is $n \times n$ state, B is $n \times r$ input, C is $m \times n$ output and D is $m \times r$ transfer matrices. For the rocket system in this study, it can be shown to be represented mathematically as:

$$\begin{aligned} \begin{bmatrix} \dot{\theta} \\ \dot{q} \\ \dot{w} \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 \\ 14.7805 & 0 & 0.01958 \\ -100.858 & 1 & -0.1256 \end{bmatrix} \begin{bmatrix} \theta \\ q \\ w \end{bmatrix} + \begin{bmatrix} 0 \\ 3.4858 \\ 20.42 \end{bmatrix} \\ \delta_{\eta} + \begin{bmatrix} 0 \\ 14.7805 \\ -94.8557 \end{bmatrix} \alpha_w & y = [1 \ 0 \ 0] \begin{bmatrix} \theta \\ q \\ w \end{bmatrix} \end{aligned} \quad (2)$$

where, θ is the pitch angle, q is the pitch rate and w is the velocity in z-axis.

Stability Analysis

After developing a model for the rocket as given in (2), it is imperative to investigate the system properties as it relates to certain characteristics of control system design. First, we want to view the trajectory of the system without a *controller* associated with it. This is called the open-loop response.

Open-loop Response of a Plant

Typically, this is done by stimulating the system with a *step*, *impulse* or *sine signal*.

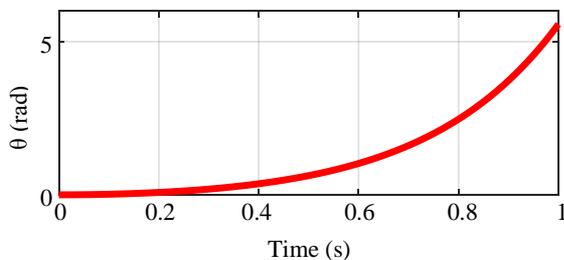


Fig. 1: Open-loop step response

The open-loop *step* response for the rocket system as described in (2) was simulated in MATLAB/Simulink. All the state of the system were view but only that of the pitch angle is presented. This was due to the fact that we are majorly interested in the control of the pitch angle of the rocket only.

The open-loop step response from Fig. 1 was obtained by the following code in MATLAB®:

```
sys=ss(A,B,C,D);
step(sys,1)
```

From Fig. 1, pitch angle keeps increasing with increase in time. This suggest instability, though we need to further investigates its eigenvalues.

Eigenvalues of a Plant

The eigenvalues of a matrix are its roots. To further validate that the rocket system is unstable, we computed its eigenvalues. This was done easily in MATLAB® with the command:

```
eig(A)
```

The above command gave $S_{1,2,3} = [3.0383, -3.8517, -0.1122]^T$, because only one real root is negative the rocket is unstable. Now that we are certain that the system as described in (2), it unstable we need to establish the fact that our mathematical model for the rocket is *controllable*-meaning, the formulated mathematical equation for the rocket is suitable for Optimal Control theory synthesis.

Controllability of a Plant

The concept of controllability was introduced by Kalman (1960) and plays an important role in the control of multivariable systems. Considering a system as described by (1), the system is said to be controllable if a control vector $u(t)$ exist that will transfer the system from any initial state $x(t_0)$ to some final state $x(t)$ in a finite time interval. Sufficient condition for complete state controllability is defined by (3) such that M is of the rank n :

$$M = [B : AB : \dots : A^{n-1}B] \quad (3)$$

The following MATLAB code was used to determine wether (2) is controllable:

```
Co = ctrb(A,B)
r=rank(Co)
l=length(A)
unco = length(A) - rank(Co)
```

The result of the above code gave, the rank of (2) to be 3 and the number of uncontrollable states as 0. Hence, (2) is controllable and the controllability matrix is:

$$Co = \begin{bmatrix} 0 & 0 & 3.5 & 14.8 & 0 & -0.2 \\ 3.5 & 14.8 & 0 & 0.2 & 51.5 & 218.5 \\ 20.4 & -94.9 & 0.9 & 26.7 & -351.6 & -1494.3 \end{bmatrix}. \quad (4)$$

Optimal Control Theory

In modern control theory, the optimal control problem is to find a control which causes the dynamical system to reach a target or follow a state variable (or trajectory) and at the same time extremize a performance index (Naidu, 2003). Thus, the design is usually with respect to a performance index-this is Optimization in its general form.

The control $u(t)$ and state $x(t)$ vectors are either unconstrained or constrained depending upon the physical situation. The unconstrained problem is less involved and gives rise to some elegant results. From the physical considerations, often we have the controls and states, such as currents and voltages in an electrical circuit, speed of a motor, thrust of a rocket, constrained as:

$$u_+ \leq u(t) \leq u_- \text{ and } x_- \leq x(t) \leq x_+, \quad (5)$$

where, +, and - indicate the maximum and minimum values the variables can attain.

Hence we can state that the optimal control problem is about finding the optimal control $u^*(t)$ which causes the linear time-invariant plant to give the trajectory $x^*(t)$ that optimizes or extremizes (minimizes or maximizes) a performance index with some constraints on the control variables $u(t)$ and/or the state variables $x(t)$. This is summarized in Fig. 2. Thus, the Optimal control problem is basically interested in finding the control $u^*(t)$ which when applied to the plant described, gives an optimal performance index J^* . A lot of control algorithms fall in this class, starting with the Linear Quadratic Regulator.

Linear Quadratic Control (LQR)

Linear Quadratic Regulator seeks to minimize a cost function, in general terms, the optimal control problem is to find u which causes the system:

$$\dot{x} = g(x(t), u(t), t) \quad (6)$$

To follow an optimal trajectory that minimizes the performance criterion (Burns, 2001), or cost function:

$$J = \int_{t_0}^{t_1} h(x(t), u(t), t) dt \quad (7)$$

The Hamilton-Jacobi equation is usually solved for the case of linear time-invariant plant with quadratic performance criterion (called the performance), which takes the form of the matrix Riccati equation. Thus,

producing an optimal control law as a linear function of the state vector components which is stable provided the system is controllable.

Define a cost function of the form:

$$f(x, t) = \min_u \int_{t_0}^{t_1} h(x, u) dt \quad (8)$$

where, over the time interval t_0 to t_1 :

For a linear, time invariant plant, (7) becomes:

$$\begin{aligned} f(x, t_0) &= f(x(0)) \\ f(x, t_1) &= 0 \end{aligned} \quad (9)$$

And if (7) is a quadratic performance index, then:

$$J = \int_{t_0}^{t_1} (x^T Q x + u^T R u) dt \quad (10)$$

where, Q is an $l \times l$ symmetric positive-definite matrix and R an $m \times m$ symmetric positive-definite matrix. A first selection for the matrices Q and R is given by Bryson's rule as:

$$\begin{aligned} Q_{ii} &= \frac{1}{\text{maximum acceptable value of } x_i^2} \quad i \in \{1, 2, \dots, l\} \\ R_{jj} &= \frac{1}{\text{maximum acceptable value of } u_j^2} \quad j \in \{1, 2, \dots, m\} \end{aligned} \quad (11)$$

Bryson's rule scales the variables that appear in (10) so that the *maximum acceptable value for each term is one*. Bryson's rule sometimes gives good results, often it is just the starting point to a trail-and-error iterative design procedure aimed at obtaining desirable properties for the closed-loop system (Franklin *et al.*, 2006; Williams and Lawrence, 2007; Anderson and Moore, 1989). For this study, a combination of Bryson's rule with simulation was used to obtain the following values area used:

$$Q = \begin{bmatrix} 0.0007 & 0 & 0 \\ 0 & 0.0007 & 0 \\ 0 & 0 & 0.0000105 \end{bmatrix}, \quad (12)$$

$$R = 0.35.$$

The optimal control law can be shown to be given as:

$$u_{opt} = -R^{-1} B^T P x, \quad (13)$$

Or:

$$u_{opt} = -K_{LQR} x, \quad (14)$$

Where:

$$K_{LQR} = R^{-1} B^T P_{LQR} \quad (15)$$

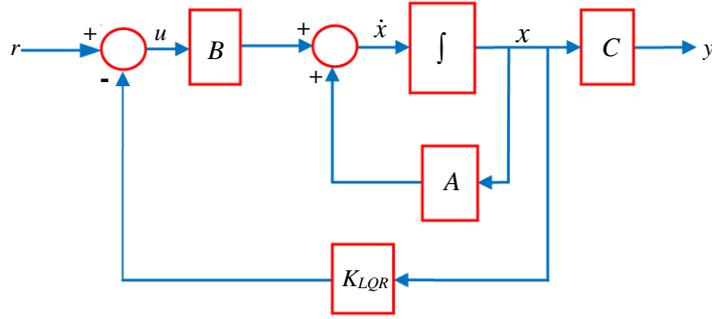


Fig. 2: LQR control schematic

Note that P is the stabilizing solution of Algebraic Riccati Equation (ARE) in (16):

$$P_{LQR}A + A^T P_{LQR} + Q - P_{LQR}B R^{-1} B^T P_{LQR} = 0 \quad (16)$$

The MATLAB in-built function $[K_{LQR}, P_{LQR}, E_{LQR}] = lqr(sys, Q, R)$ was used to compute P_{LQR} , of the associated algebraic Riccati equation as stated in (16), the state feedback gain matrix K_{LQR} , as given in (15) and $E = eig(A - BK_{LQR})$, the closed-loop eigenvalues (User's Guide, 2014) for the associated rocket. These computed values are:

$$E_{LQR} = \begin{bmatrix} -3.9623 \\ -3.7222 \\ -0.2428 \end{bmatrix} \quad (17)$$

The LQR controller as depicted in Fig. 2 was implemented in Simulink[®] was designed for a tracking scenario of a reference signal of $r = [0.053 \ 0 \ 0]^T$ and a system initial condition of $x_0 = [0 \ 0 \ 0]^T$:

$$K_{LQR} = [8.4765 \ 2.2055 \ 0.0056] \quad (18)$$

$$P_{LQR} = \begin{bmatrix} 3.3257 & 0.8509 & 0.0000 \\ 0.8509 & 0.2197 & 0.0003 \\ 0.0000 & 0.0003 & 0.0000 \end{bmatrix} \quad (19)$$

The state feedback controller with (18) as the feedback gain, was design and simulated in Simulink and the simulation result is given in Fig. 3.

It is clearly seen from Fig. 3 that the LQR has appreciable step-response characteristics and very good tracking of the pitch angle. LQR formulation has the drawback that all states must be available for feedback. This assumption is not realistic, hence the need for an observer to measure all or some of the state is inevitable.

It is impractical to expect in real-world applications that all state variables are measurable hence, the ability to implement a state feedback control law is in jeopardy.

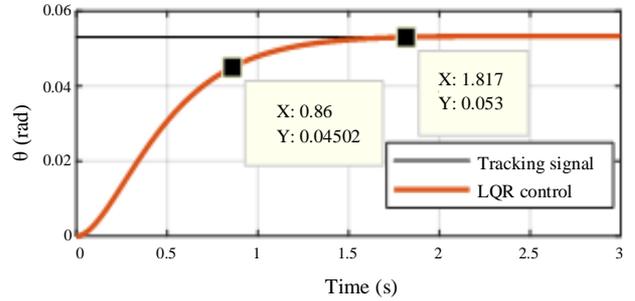


Fig. 3: Set-point tracking LQR control of the rocket

This brings in the need for an estimate of the state vector derived from measurements of the input and output. Before an estimator is design the plant must be checked for *observeverbility*.

Observability of a Plant

An observer in its simplest term is an algorithm that eatimate and predicts the state of a dynamic system. One of the most popular observer with wide range of application is Kalman Filter. A necessary condition for the Kalman Filter to work correctly is that the system for which the states are to be estimated, is *observable*. Therefore, you should check for *observability* before applying the Kalman Filter. (There may still be other problems that prevent the Kalman Filter from producing accurate state estimates, as a faulty or inaccurate mathematical model).

The system described by (2) is completely observable if the matrix N as given in (20) is of the rank n :

$$N = [C^T : A^T C^T : \dots : (A^T)^{n-1} C^T] \quad (20)$$

Also, we use MATLAB[®] to compute (20) and compred its rank with the system $n \times n$ matrix:

$$\begin{aligned} Ob &= \text{obsv}(A, C) \\ r &= \text{rank}(Ob) \\ l &= \text{length}(A) \\ unob &= \text{length}(A) - \text{rank}(Ob) \end{aligned}$$

$$ob = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 14.7805 & 0 & 0.002 \\ 14.7805 & 0 & 0.002 \\ -0.1975 & 14.7825 & -0.0002 \end{bmatrix} \quad (21)$$

If the determinant of N is zero, the system is non-observable. Non-observability has several consequences:

- The transfer function from the input variable y to the output variable has an order that is less than the number of state variables (n)
- There are state variables or linear combinations of state variables that do not show any response
- The steady-state value of the Kalman Filter gain cannot be computed. This gain is used to update the state estimates from measurements of the (real) system

Linear Kaman Filter

In real world system there would be noise (disturbance) on these measurements, and in order to analyse this problem realistically noise need to be added to the model. In general the output y is affected by measurement noise and the process dynamics are also affected by disturbance, specifically turbulence of atmosphere.

In light of this, a more reasonable model for the plant as given in (1) is:

$$\begin{aligned} \dot{x} &= Ax + Bu + Gw(t) \\ y &= Cx + v(t) \end{aligned} \quad (22)$$

where, $w(t)$ and $v(t)$ are zero-mean Gaussian noise processes (uncorrelated from each other) with power spectrum:

$$S_w(\omega) = Q_N, S_v(\omega) = R_N, \forall \omega$$

Q_N is the process noise covarian and R_N is the measurement noise covariance matrix. The following process (Q_N) and measurement (R_N) covariance matrices hold namely:

$$\left. \begin{aligned} Ev(t)v^T(s) &= R_N \delta(t-s) \\ Ev(t)w^T(s) &= N \\ Ew(t)w^T(s) &= Q_N \delta(t-s) \end{aligned} \right\} t, s \in \mathfrak{R}. \quad (23)$$

A linear state observer is an n -dimensional linear state equation that accepts u and y as input and whose state represents the estimate of x . Suppose we construct the estimate \hat{x} by replacing the process dynamics in (23) to be:

$$\dot{\hat{x}} = A\hat{x} + Bu \quad (24)$$

To see if this would generate a good estimate for x , we can define the state estimation error in (24) and study its dynamics by:

$$e = x - \hat{x} \quad (25)$$

From (22) and (24), we can conclude that:

$$\dot{e} = Ax - A\hat{x} = Ae \quad (26)$$

This shows that when the matrix A is asymptotically stable the error e converges to zero for any input u , which is good news because it means that \hat{x} eventually converges to x as $t \rightarrow \infty$. However, when A is not stable e is unbounded and \hat{x} grow further and further apart from x as $t \rightarrow \infty$. To avoid this, one includes a correction term in (24) which yields:

$$\dot{\hat{x}} = A\hat{x} + Bu + K_{LQE}(y - \hat{y}), \quad \hat{y} = C\hat{x}. \quad (27)$$

where, \hat{y} should be viewed as an estimate of y and K_{LQE} a given $n \times k$ matrix. When \hat{x} is equal (or very close) to x , then \hat{y} will be equal (or very close) to y and the correction term $K_{LQE}(y - \hat{y})$ plays no role. However, when \hat{x} grows away from x , this term will (hopefully) correct the error. To see how this can be done, we re-write the estimation error dynamics for (22) and (24) as:

$$\dot{e} = Ax - A\hat{x} - K_{LQE}(Cx - C\hat{x}) = (A - K_{LQE}C)e \quad (28)$$

Now e converges to zero as long as $A - K_{LQE}C$ is asymptotically stable. It turns out that, even when A is unstable, in general we will be able to select K_{LQE} so that $A - K_{LQE}C$ is asymptotically stable. The system in (27) can be re-written as:

$$\dot{\hat{x}} = (A - K_{LQE}C)\hat{x} + Bu + yK_{LQE} \quad (29)$$

Now (29) is called a *full-order observer* for the process. Full-order observers have two inputs- the process' control input u and its measured output y (via sensors) - and a single output-the state estimate \hat{x} . Schematically, the full-order observe is represented as shown in Fig. 4:

Any choice of K_{LQE} in (29) for which $A - K_{LQE}C$ is asymptotically stable will make the estimated state vector to converge to the actual state vector.

The Kalman gain, K_{LQE} , for the optimal observer can be shown to be as given in (30), where P_{LQE} is the stabilizing solution P_{LQE} of Algebraic Riccati Equation (ARE) given as (31):

$$K_{LQE} = P_{LQE}C^TQ_N^{-1} \quad (30)$$

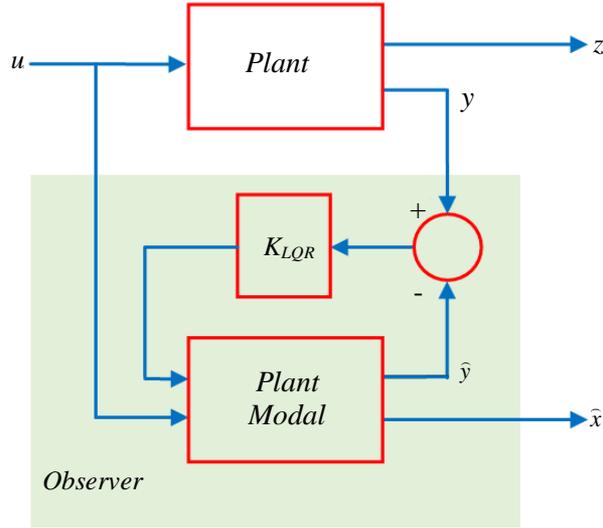


Fig. 4: Full-order observer diagram

$$P_{LQE}A + A^T P_{LQE} + G^T Q_N G - P_{LQE} C^T R_N^{-1} C P_{LQE} = 0 \quad (31)$$

For this study, the following values were used for the LQG design:

$$R_N = \text{diag}([0.002 \quad 0.002 \quad 0.002]) \quad (32)$$

$$Q_N = \text{diag}([0.0326 \quad 0.0326 \quad 0.0326]) \quad (33)$$

$$P_0 = \text{diag}([0.0000625 \quad 0.0000625 \quad 0.0000625]) \quad (34)$$

$$G_N = GVG^T = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 2.1846 & -14.0201 \\ 0 & -14.0201 & 89.9760 \end{bmatrix} \quad (35)$$

With these values declared in a MATLAB m-file, the comamd $[K_{LQE}, P_{LQE}, E_{LQE}] = \text{lqe}(A, G_N, C, Q_N, R_N, P_0)$ was used to compute the kalman gain K_{LQE} for our observer and was used to estimate the pitch angle of the rocket which gave the result in Fig. 5. The Kalman filter design has properties as given in (36), (37) and (38):

$$K_{LQE} = \begin{bmatrix} 0.3155 & -0.0026 & -0.2413 \\ -0.0026 & 9.0765 & -56.5430 \\ -0.2413 & -56.5430 & 363.2142 \end{bmatrix} \quad (36)$$

$$P_{LQE} = \begin{bmatrix} 0.0006 & -5.18 \times 10^{-6} & -0.0005 \\ -5.18 \times 10^{-6} & 0.0180 & -0.1122 \\ -0.0005 & -0.1122 & 0.7208 \end{bmatrix} \quad (37)$$

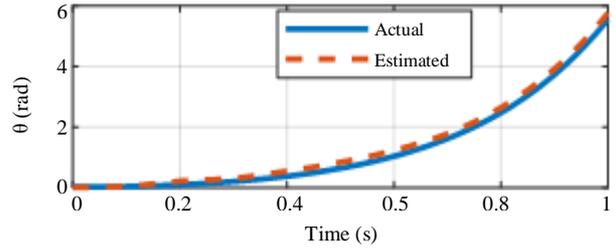


Fig. 5: Kalman filter estimate of the pitch angle of the rocket

$$E_{LQE} = \begin{bmatrix} -372.13 + 0.0000i \\ -0.30316 + 0.95107i \\ -0.30316 - 0.95107i \end{bmatrix} \quad (38)$$

The closer to zero, the elements of the diagonal matrix of P_{LQE} (covariance matrix) is, suggests that the error is minimal. Hence, the value of 0.0006 in (38) is associated with the kalman filter estimate of the pitch angle. Thus, it validates the result displayed in Fig. 6 (kalman filter estimate of the pitch angle is very close to the actual trajectory). The eigenvalues as displayed in (38) are all negative, this suggests a stable observer.

LQG Control Design

Basically, the LQG approach addresses the problem where we consider a system dynamic model perturbed by a dynamical noise w and a state observation corrupted by measurement noise v affecting the sensors data acquisition. It is essentially a combination of LQR and a *Kalman* based estimator (Aliyu, 2011).

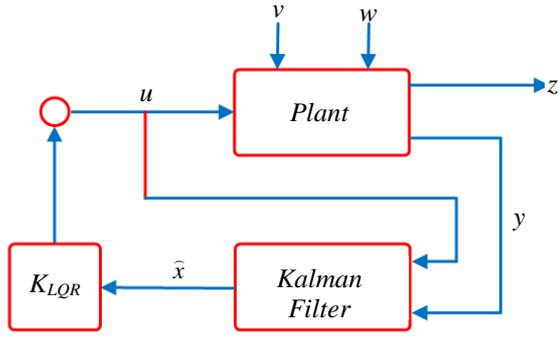


Fig. 6: LQG control schematic

The estimator or observer addresses the short fall of the LQR's assumption that all states are available for feedback (via sensors). Here, all the required states will be estimated by the renowned kalman filter.

From Fig. 6, z is the controlled system output and K_{LQR} is the controller feedback gain in our case is the LQR gain. Consequently, the LQG control law is:

$$u = -k\hat{x}. \quad (39)$$

where, \hat{x} the estimated state by Kalman filter.

The designed LQG controller in MATLAB/simulink was simulated to track a pitch angle of 3 degrees (0.053 radians) and the result is depicted in Fig. 7.

To check the stability of the LQG system, it is more convenient to consider the dynamics of the estimation error. In matrix form, the related equation can be shown as:

$$\begin{bmatrix} \dot{\hat{x}} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A - BK_{LQR} & BK_{LQR} \\ 0 & A - K_{LQE}C \end{bmatrix} \begin{bmatrix} \hat{x} \\ e \end{bmatrix}, \quad (40)$$

For the LQG designed in this study, the eigenvalues of (40) are as given in (41). Notice that (34) is a combination of the roots of the LQR controller and those of the Kalman as earlier designed in (17) and (38):

$$LQG_{roots} = \begin{bmatrix} -3.9476 \\ -3.7392 \\ -0.24116 \\ -372.13 \\ -0.30312 + 0.95106i \\ -0.30312 - 0.95106i \end{bmatrix} \quad (41)$$

Improved LQG Control Design ($LQG_{i,1}$)

Base on the dual principle, (16) is the algebraic riccati equation for the LQR controller and (31) is its counterpart for the kalman filter. Hence, we can write (42) as the non-steady form of (31). This is a *riccati differential equation* (Aliyu *et al.*, 2012):

$$\dot{P}_{LQE} = -P_{LQE}A - A^T P_{LQE} - G^T Q_N G + P_{LQE} C R_N^{-1} C^T P_{LQE} \quad (42)$$

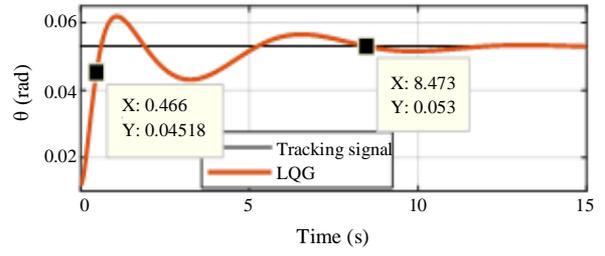


Fig. 7: Set-point tracking LQG control of the rocket

We propose an improved LQG controll design base on the solution of (42). This Riccati Differential Equation (RDE) is an initial value differential problem with initial conditon $P_0 = [6.25 \times 10^{-5} \ 6.25 \times 10^{-5} \ 6.25 \times 10^{-5}]^T$. Thus, we put forward an observer gain as given in (43). This kalman gain values change all through the regime of flight:

$$K_{LQE}(t) = P_{LQE} C^T R_N^{-1} \quad (43)$$

State estimation was done using (43) as our Kalman gain and the result for pitch angle estimation is shown in Fig. 8a (zoom of Fig. 8b). Notice that the actual and estimated state is exactly the same, this is why both plots are super-imposed on each other as seen in Fig. 8b.

In MATLAB/Simulink®, (43) as observer gain was combined with the LQR control designed earlier for the rocket to form the first improved LQG controller ($LQG_{i,1}$). Simulation result of such controller is depicted in Fig. 9.

Kalman gain for simulation in Fig. 9 gave is 916 values of Kalman gain which were updated throughout the simulation time of 5 sec.

Improved LQG Control Design ($LQG_{i,2}$)

To establish a fare comparison with the traditional LQG design, we decided to take the 916th value of the Kalman gain from the solution of (43) as given in (44) was used to designed an estimator which was able to estimate the pitch angle of the rocket as shown in Fig. 10. The estimator has properties as given in (45) and (46):

$$K_{LQE_{916}} = \begin{bmatrix} 6.3336 & -0.11602 & -3.1046 \\ -0.11602 & 5.8228 & -32.458 \\ -3.1046 & -32.458 & 210.93 \end{bmatrix} \quad (44)$$

$$P_{RDE_{139}} = \begin{bmatrix} 0.01267 & -0.0002322 & -0.00621 \\ -0.0002322 & 0.01165 & -0.06492 \\ -0.00621 & -0.06492 & 0.4219 \end{bmatrix} \quad (45)$$

$$E_{RDE_{138}} = \begin{bmatrix} -214.67 \\ -7.6623 \\ -0.85508 \end{bmatrix} \quad (46)$$

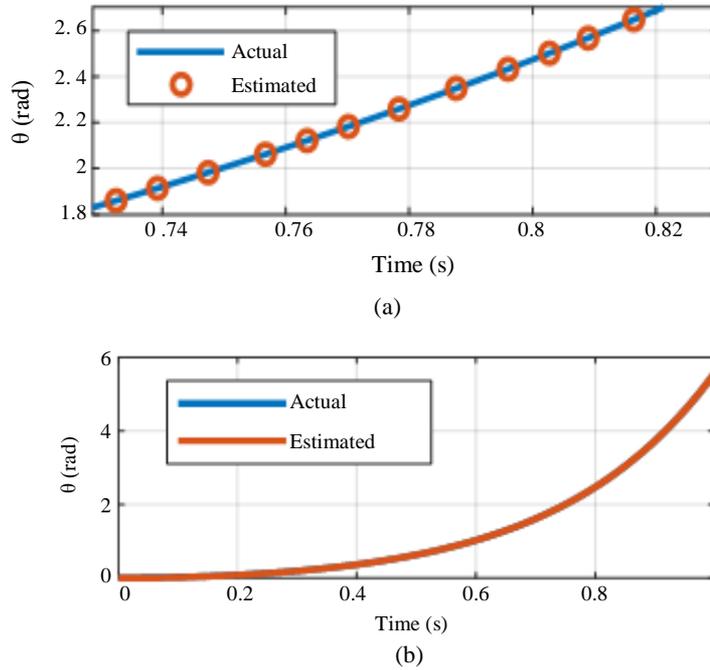


Fig. 8: Pitch angle estimation using RDE to obtain kalman gain

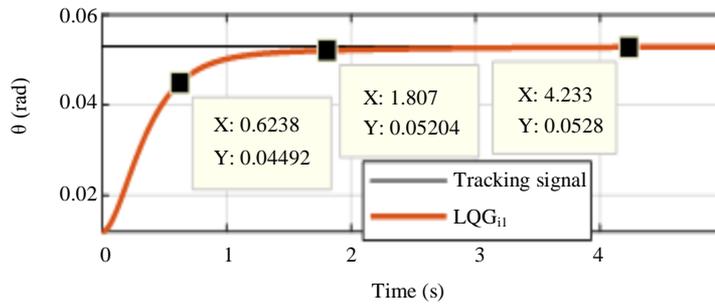


Fig. 9: Set-point tracking $LQG_{i,1}$ control of the rocket with Kalamn gains from differential ricatti equation

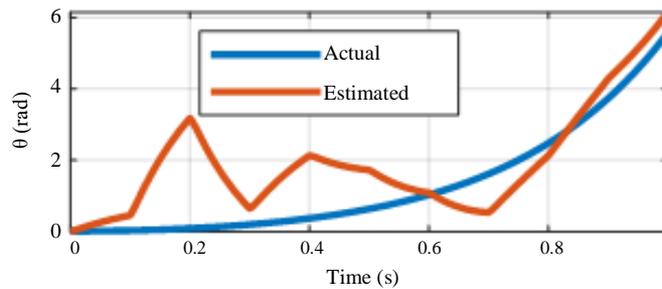


Fig. 10: Pitch angle estimation using the 916th value of kalman gain from the solution of a RDE

From the eigenvalues in (46), the estimator is stable (negative eigenvalues). Comparing the P_{11} element in (45) with that of (37), one could clearly say that the latter will give a better estimate of the pitch angle compared to the former. This fact is validated with Fig. 10, it can be seen that the

algorithm poorly estimated the actual state for about 0.8 sec. When this variant of the Kalman gain was used as an observer to realise our second novel LQG algorithm ($LQG_{i,2}$), simulation result (Fig. 11) gave us a controller with better time response characteristics than the traditional LQG.

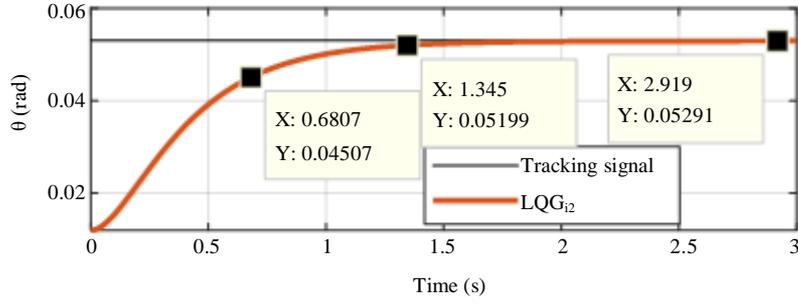


Fig. 11: Set-point tracking control of pitch angle with LQG_{i,2}

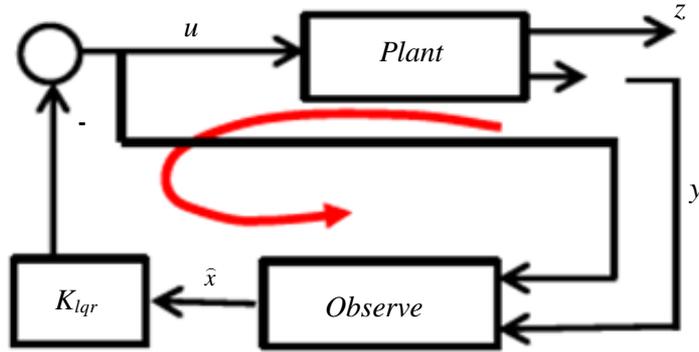


Fig. 12: Breaking the LQG loop at plant output

Our quest for a better controller will lead us to designing LQG/LTR hoping we can get one with better properties than that of LQG_{i,2}. Hence, we proceed with the traditional LQG/LTR design.

LQG/LTR Control Design

Based on the LQR solution of the above nominal optimal control, we can design a Linear Quadratic Gaussian (LQG) with Loop Transfer Recovery (LTR) controller. The LQG/LTR method is rooted in optimal control theory and in spite of systematic design procedure, shows some useful properties of robustness and good performance. In this method, the desired time response characteristics of the closed-loop system (plant and controller) must be designed in an LQG problem and then these time response characteristics are recovered at the input or output of the real plant by successive tuning of a gain in an LQR problem or reduction of the measurement noise as the case may be. Hence, there are two main methods for LQG/LTR design (Kulcsar, 2000), these are:

- LQG/LTR at input of the plant by tuning the kalman gain matrix KLQE
- LQG/LTR at output of the plant by tuning gain matrix KLQR

LQG/LTR Design at Plant Output

Now, breaking the LQG loop at the plant output as shown in Fig. 12, the return ratio at the output of the plant is:

$$L_p(s) = -G(s)L_{LQG}(s), \tag{47}$$

The associated sensitivity function of plant is defined as:

$$S_p(s) = \{I + G(s)L_{LQG}(s)\}^{-1}. \tag{48}$$

Investigation the effect of decreasing the intensity Q of the LQR problem. Suppose:

$$Q_{LQR} = Q_0 + \rho M, \tag{49}$$

where, M is a fixed symmetric positive-definite weighting matrix and ρ a positive number. Investigating the asymptotic behaviour of the closed-loop system as $\rho \downarrow 0$. Also we assume that the plant in this case is minimum-phase. The feedback gain for the LQ problem is given by:

$$K_{LQR} = R^{-1}B^T P_{LQR} \tag{50}$$

where, $P_{LQR} \geq 0$ is the maximal solution of the control Algebraic Riccati Equation (CARE):

$$A^T P_{LQR} + P_{LQR} A + Q - P_{LQR} B R^{-1} B^T P_{LQR} = 0. \quad (51)$$

Examining the situation while $\rho \rightarrow \infty$:

$$\lim_{\rho \rightarrow \infty} \left\{ \frac{A^T P_{LQR}}{\rho} + \frac{P_{LQR} A}{\rho} + \frac{Q_0}{\rho} + M - \frac{P_{LQR} B R^{-1} B^T P_{LQR}}{\rho} \right\} = 0, \quad (52)$$

And considering the fact that:

$$\lim_{\rho \rightarrow \infty} \frac{P_{LQR}}{\rho} = 0. \quad (53)$$

Then (52) reduces to:

$$M - \frac{P_{LQR} B R^{-1} B^T P_{LQR}}{\rho} = 0, \quad (54)$$

Thus:

$$\rho M = P_{LQR} B R^{-1} B^T P_{LQR}, \quad (55)$$

Multiplying both sides of (55) with the power of $1/2$ gives, we can write:

$$(\rho M)^{1/2} = P_{LQR}^{1/2} P_{LQR}^{1/2} B^{1/2} B^{1/2} R^{-1/2} = P_{LQR} B^{1/2} B^{1/2} R^{-1/2}, \quad (56)$$

Multiplying both sides of (56) with $B^T/2B-1/2R-1/2$ gives:

$$R^{-1/2} \rho^{1/2} M^{1/2} B^{-1/2} B^{1/2} = P_{LQR} B^T R^{-1}, \quad (57)$$

$$\lim_{\rho \rightarrow \infty} F = \lim_{\rho \rightarrow \infty} \left\{ \rho^{1/2} R^{1/2} M^{1/2} D^{1/2} D^{1/2} B^{-1/2} B^{1/2} \right\}, \quad (58)$$

Choosing $M = I$, the open loop transfer function becomes:

$$\lim_{\rho \rightarrow \infty} L_{LQG}(s) = \lim_{\rho \rightarrow \infty} \left\{ \begin{array}{l} -K_{LQR} (\psi^{-1}(s) + K_{LQR} B)^{-1} \\ K_{LQE} C \Phi(s) B \end{array} \right\} \quad (59)$$

$$\lim_{\rho \rightarrow \infty} L_{LQG}(s) = K_{LQR} \Phi(s) C.$$

From (59) it is inferred that with a q very high we can approach the return ratio of the Kalman filter. Also, during Loop Transfer Recovery, the poles of:

$$L_{LQG}(s)_{\rho \rightarrow \infty} \rightarrow \text{The zeros of the plant } C \Phi(s) K, \quad (60)$$

and the zeros of:

$$L_{LQG}(s)_{\rho \rightarrow \infty} \rightarrow \text{The zeros of the plant } C \Phi(s) B, \quad (61)$$

Thus, the design procedure is:

- First, we design a Kalman filter whose transfer function $C \Phi(s) K_{LQE}$ is desirable. By choosing the power spectral density matrices w and v so that the minimum singular value of $C \Phi(s) K_{LQE}$ is large enough at low frequencies for good performance and its maximum singular value is small enough at high frequencies for robust stability;
- When the singular values of $C \Phi(s) K_{LQE}$ are thought to be satisfactory, loop transfer recovery is achieved by designing K_{LQR} in an LQR problem with $B=C$ in (59), $Q = Q_0 + \rho M$, where ρ is a scalar.

LQG/LTR Design at Plant Input

Loop Transfer Recovery at input of the plant (as shown in Fig. 13) is achieved by primarily tuning the kalman gain matrix K_{LQE} , its return ratio is given in (62). Notice that (62) is just a form of (47) being casted for LQG/LTR design at Plant input. Thus, extending the *dual* principle of the LQG problem to the LTR definition:

$$L_{\rho}(s) = -G(s) L_{LQR}(s), \quad (62)$$

Since the K_{LQR} controller is ready, we only need to design the Kalman filter which can help recover the robustness of LQR controller. Notice, we are simply taking the converse steps of LQG/LTR at the plant output design as earlier outlined. The Loop Transfer Recovery design is done by increasing the spectral density S_w of the plant noise $w(t)$. This method is popular in the research community (Ishihara *et al.*, 2005):

$$\lim_{\rho \rightarrow \infty} K_{LQR} \approx K_{LQE} (sI - A)^{-1} B, \quad (63)$$

We choose different values of S_w to design the Kalman filter and compare the results to the LQR results. We choose different values of S_w by doing:

$$Q_{NN} = Q_N + \rho I, \quad (64)$$

where, I is an identity matrix and we want to find the K_{LQE} feedback while $\rho \rightarrow \infty$.

The approach suggested in (64) is similar to that expressed in (49). Here, we were able to prove with simulation that the same approach can be implemented for LTR at plant input.

The Kalman filter gain for the LQG/LTR at the plant input for this study is obtained with MATLAB® command $[K_{LQE}, P_{LQE}, E_{LQE}] = \text{lqe}(A, G, Q, C, Q_{NN}, R_N, P_0)$. Next we show the system response of LQG/LTR control set-point tracking response for different values of ρ in Fig. 14.

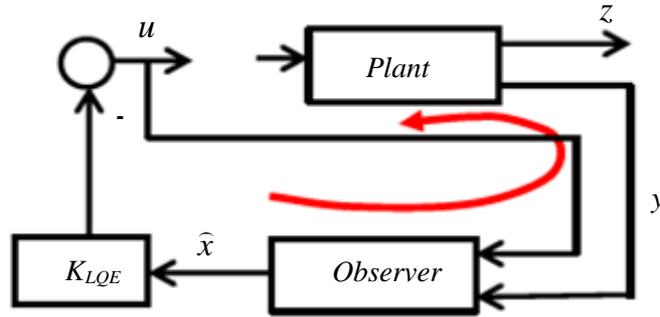


Fig. 13: Breaking the LQG loop at plant output

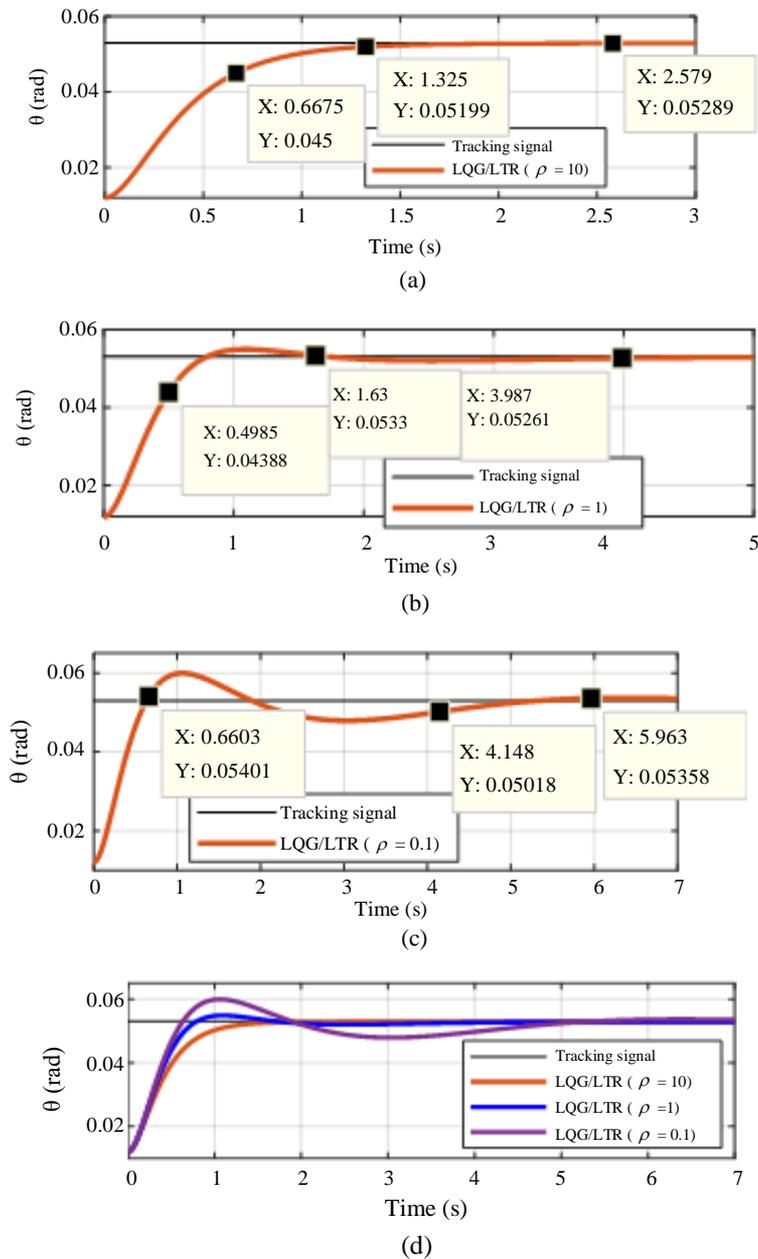


Fig. 14: LQG/LTR set-point tracking control of pitch angle

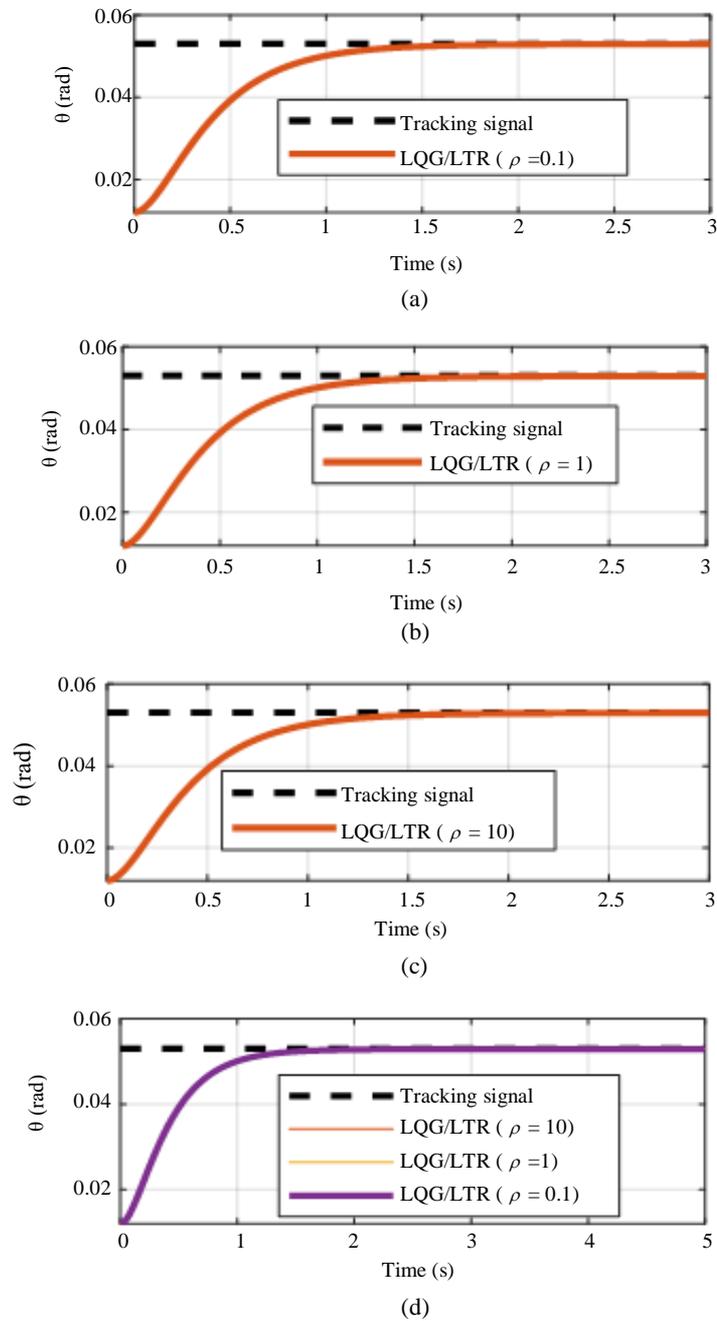


Fig. 15: Novel LQG/LTRi set-point tracking control of pitch angle

Table 1: Time response characteristics of controllers (rise time, t_r , settling time t_s , percentage overshoot P_o and steady-state-error, SSE)

S/N	Controllers	t_s (s)	t_r (s)	P_o (%)	SSE (rad)
1	LQR	1.82	0.86	0	0.000
2	LQG	8.47	0.47	0.9	0.0001
3	LQG _{i,1}	1.81	0.62	0	0.0002
4	LQG _{i,2}	1.35	0.68	0	0.0001
5	LQG/LTR ($\rho = 10$)	1.33	0.67	0	0.0002
6	LQG/LTR ($\rho = 1$)	1.63	0.49	0.2	0.0004
7	LQG/LTR ($\rho = 0.1$)	4.15	0.66	0.7	0.0006
8	LQG/LTR _i ($\rho = 0.1$)	1.2	0.68	0	0.00

Next we will design LQG/LTR using (44), as the Kalman gain for the observed. Remember this Kalman gain was harvested from a DRE. Simulation result for such design are presented in Fig. 15. We call this controller LQG/LTR_i.

Discussion of Results

It can be clearly seen from Table 1, that the LQR controller has appreciable time response characteristics with zero steady-state-error (see Fig. 4). This is viewed as an ideal situation and its position is well supported in literature (Aschepkov, 2016; ViorelBadescu, 2017; Zhang *et al.*, 2016). The combination of LQR controller with Kalman filter as an observer caused some of the fine properties of both the LQR controller and Kalman filter to be lost. The result is a closed-loop system with an overshoot (0.9%) as shown in Fig. 8. This result is also in line with popular literature position (Wang Jian, 2016; Zhu *et al.*, 2017). Our first novel controller, LQG_{i,1}, used updated values of Kalman gain all through the regime of simulation to arrive at the result depicted in Fig. 9 and Table 1. Notice that this controller was capable of restoring almost all the robustness of the LQR. In practical terms, LQG_{i,1} might be challenging to implement owing to the fact that if the rocket will fly for a long time, Kalman gain values might increase beyond the 916 steps that we have here for a simulation time of 5 second. This means large memory allocation on a hardware and at the same time prolonged computation throughout the regime of flight. Hence, we introduced the second novel controller LQG_{i,2} as a practical substitute to LQG_{i,1}. Here we used the last integrated value of Kalman gain as our preferred values of estimator gain to design the novel LQG_{i,2} controller. The simulation result of LQG_{i,2} are as depicted in Table 1 and Fig. 11. This controller is also very promising in the sense that it was also capable of restoring the robustness of the LQR controller with even a better settling time than that of the LQR. The traditional LQG/LTR at plant input was designed, and the results for three scenarios are depicted in Fig. 14. The tuned Kalman gain with the largest value of $\rho = 10$ (Fig. 14a) was able to restore all the robustness and is comparable with the LQR, LQG_{i,1} and LQG_{i,2} controllers. For LQG/LTR with $\rho = 0.1$, in Fig. 15c, this controller is comparable with the LQG and LQG/LTR with $\rho = 1$ in Fig. 14b has better time response characteristics than the LQG but less, compared to LQR, LQG_{i,1} and LQG_{i,2}. For close perusal of the three LQG/LTR controllers synthesized, Fig. 14d is perspective. The novel LQG/LTR at plant input was designed using the Kalman gain from the solution of RDE and simulation results are as depicted in Fig. 15. Notice, that LQG/LTR_i with $\rho = 0.1$ in Fig. 15a was able to restore all the robustness needed and has better time response characteristics compared to LQG_{i,1} and LQG_{i,2}

as shown in Table 1. We tried to improve on this controller by making $\rho = 1$ and $\rho = 10$ but got the same result (Fig. 15b and 15c). Notice, for this 3 LQG/LTR_i controllers to be shown to have the same time response characteristics, Fig. 15d was plotted. Hence, we consider all the controllers in Fig. 15d to be just one- LQG/LTR_i.

Conclusion

To design an optimal controller for a rocket, a mathematical model was developed and simulated in MATLAB/Simulink®. The cascade combination of a mathematical model and a control algorithm (with feedback) gives a closed-loop control system (controller). Thus, eight controllers were designed in this study: LQR, LQG, LQG_{i,1}, LQG_{i,2}, LQG/LTR ($\rho = 10$), LQG/LTR ($\rho = 1$), LQG/LTR ($\rho = 0.1$) and LQG/LTR_i. Simulation for all controllers were carried out in MATLAB/Simulink®. Results of all controllers were compared based on time response characteristics and the LQR controller was used as the benchmark, LQG_{i,1}, LQG_{i,2} and LQG/LTR_i are novel in nature based on the origin of the Kalman gain used the observer design. LQG/LTR_i outperformed the other two controllers in terms of time response characteristics (better than the LQR).

Future Work

It is imperative to formulate an analytic mathematical proof for the simulation concept and result of the LQG/LTR control presented in this study.

Acknowledgement

Authors wish to acknowledge the relentless effort of staff members at CSTP working with the instrumentation and control unit.

Ethics

Authors declare that there are no ethical issues that may arise after the publication of this manuscript. This article is original and contains unpublished material.

References

- Aliyu, B.K., 2011. Expendable Launch Vehicle Flight Control; Design and Simulation with MATLAB/Simulink, 2011. LAP LAMBERT Academic Publishing Germany, ISBN-13: 978-3844327290.
- Aliyu, B.K., C.A. Osheku, L.M.A. Adetoro and A.A. Funmilayo, 2012. Optimal solution to matrix riccati equation-for Kalman filter implementation. MATLAB - A Fundamental Tool for Scientific Computing and Engineering Applications - Volume 3, INTECH.

- Anderso, B.O. and J.B. Moore, 1989. Optimal Control. Linear Quadratic Methods. Prentice-Hall International, Inc., USA. ISBN-10: 0-13-638651-2
- Aschepkov, 2016. Optimal Control. Springer International Publishing. ISBN-10: 978-3-319-49781-5
- Barbosa, F.S., G.P. das Neves and B.A. Angélico, 2016. Discrete LQG/LTR control augmented by integrators applied to a 2-DOF helicopter. Proceedings of the IEEE Conference on Control Applications (CCA), Sept. 19-22, IEEE Xplore press, Argentina. DOI: 10.1109/CCA.2016.7587976
- Barzanoi, E., 2015. Attitude flight control system design of UAV using LQG/LTR multivariable control with noise and disturbance. Proceedings of the 3rd RSI International Conference on Robotics and Mechatronics (ICROM).
- Brian, D.O. and J.B. Moore, 1989. Optimal Control: Linear Quadratic Methods. Prentice-Hall International, New Jersey. ISBN-10: 0-13-638651-2
- Burns, R.S., 2001. Advanced Control Engineering. Butterworth Heinemann Boston. ISBN-10: 0750651008
- Das, S. and K. Halder, 2014. Missile attitude control via a hybrid LQG-LTR/LQI control scheme with optimum weight selection. Proceedings of the 1st International Conference on Automation, Control, Energy and Systems (ACES). DOI: 10.1109/ACES.2014.6807996
- Franklin, G.F., J.D. Powell and A. Emami-Naeini, 2006. Feedback Control of Dynamic Systems. 5th Edn., NJ, Prentice Hall.
- Ishihara, T. and L.A. Zheng, 2017. LQG/LTR procedure using reduced-order Kalman filters. Int. J. Control. DOI: 10.1080/00207179.2017.1359673
- Ishihara, T., H.J. Guo and H. Takeda, 2005. Integral controller design based on disturbance cancellation: Partial LTR approach for non-minimum phase plants. Automatica, 41: 2083-2089.
- Jianqiao, Y.U., L.U.O. Guanchen and M.E.I. Yuesong, 2011. Surface-to-air missile autopilot design using LQG/LTR gain scheduling method. ELSEVIER, Chinese J. Aeronautics, 24: 279-286.
- Kirk, D.E., 1998. Optimal Control Theory: An Introduction. Prentice-Hall, Inc., New Jersey, ISBN-10: 0-486-43484-2
- Kulcsar, B., 2000. LQG/LTR controller design for an aircraft model. Periodica Polytechnica SER. Transp. Eng., 28: 131-142.
- Liu, X., 2017. Fuel-optimal rocket landing with aerodynamic controls. Proceedings of the AIAA Guidance, Navigation and Control Conference, AIAA SciTech Forum, (AIAA 2017-1732), Beijing Institute of Technology, China, pp: 1-15. DOI: 10.2514/6.2017-1732
- Moshen Ahmed, W.A.E.L. and Q.U.A.N. Quan, 2011. Robust hybrid control for ballistic missile longitudinal autopilot. ELSEVIER, Chinese J. Aeronautics, 24: 777-788.
- Naidu, D.S., 2003. Optimal Control. CRC Press LLC, New York. ISBN-10: 0-8493-0892-5
- Nair, M.P. and R. Harikumar, 2015. Longitudinal dynamics control of UAV. Proceedings of the International Conference on Control Communication and Computing India (ICCC). Nov. 19-21, IEEE Xplore press, India. DOI: 10.1109/ICCC.2015.7432865
- User's Guide, 2014. Control System Toolbox™. Revised for Version 9.7 (Release 2014a). The MathWorks, Inc.
- ViorelBadescu, 2017. Optimal Control in Thermal Engineering. Springer, ISBN-10: 9783319529684
- Wang Jian, 2016. The optimal guidance of rocket fixed-attitude and fixed-point orbit injection. Proceedings of the Chinese Control and Decision Conference, May 28-30, IEEE Xplore press, China. DOI: 10.1109/CCDC.2016.7531741
- Williams, R.L. and D.A. Lawrence, 2007. Linear State-space Control Systems.
- Zhang, Y.J., A.A. Malikopoulos and C.G. Cassandras, 2016. Optimal control and coordination of connected and Automated vehicles at urban traffic intersections. Proceedings of the American Control Conference (ACC) Boston Marriott Copley Place, Jul. 6-8, Boston, MA, USA, pp: 6227-6232.
- Zhang, Y.J., A.A. Malikopoulos and C.G. Cassandras, 2016. Optimal control and coordination of connected and automated vehicles at urban traffic intersections. Proceedings of the American Control Conference, Jul. 6-8, IEEE Xplore press, USA. DOI: 10.1109/ACC.2016.7526648
- Zhu, J., E. Trélat and M. Cerf, 2017. Geometric optimal control and applications to aerospace. Pacific J. Mathematics Industry. DOI: 10.1186/s40736-017-0033-4