

# LQR controller design for quad-rotor helicopters

eISSN 2051-3305  
Received on 22nd June 2018  
Accepted on 27th July 2018  
E-First on 4th June 2019  
doi: 10.1049/joe.2018.8126  
www.ietdl.org

Emmanuel Okyere<sup>1</sup> ✉, Amar Bousbaine<sup>1</sup>, Gwangtim T. Poyi<sup>1</sup>, Ajay K. Joseph<sup>1</sup>, Jose M. Andrade<sup>1</sup>

<sup>1</sup>University of Derby, School of Engineering and Technology, Derby, UK

✉ E-mail: a.bousbaine@derby.ac.uk

**Abstract:** This paper presents an analysis and performance of a LQR control algorithm for quadrotor helicopters. For a successful analysis, first the dynamic model has been developed for the quadcopter and then the controller was designed, tuned, and tested. In tuning the LQR, much attention was given to the feedback gain matrix (K). The controller's performance was verified in terms of delay time, rise time, overshoot, settling time, and tolerance limits. The overall performance of the LQR controller was analysed.

## 1 Introduction

In recent years, unmanned aerial vehicles (UAVs) have attracted more attention due to their high manoeuvrability, simplicity of construction, low maintenance costs, and low noise. Quadcopters are rotary wing machines that make use of four propellers in achieving vertical take-off and landing and other flight manoeuvres. UAVs have become very useful not only for military purposes but in many areas such as; aerial photo and video shooting, farm irrigation and crop monitoring, border patrol and rescue missions, electric power line and gas pipeline monitoring and many others. Works are underway to see UAVs fully fledged in door-to-door delivery of goods bought over the Internet through Amazon, Google, and others.

Generally small in size, quadcopters use a variety of sensors to achieve a high level of stability and control, allowing them to navigate even in narrow spaces. Additionally, because each rotor is small, they require less power during flight, which makes quadcopters much safer both to human operators and to the flight environment. Finally, quadcopters are generally low cost and easy to construct. All these factors contribute to making them the rotorcraft of choice for most academic and research purposes.

Design of control system for quadrotor helicopters is an ongoing and growing research field. The controller is the main part in designing any autopilot system for aviation and quadrotor helicopter in particular. For a stable control, the throttle, roll, pitch, and yaw have to be controlled successfully so that the vehicle navigate its terrain, while maintaining steady, stable flight, and controlled movement.

The goal of this paper is to present a step-by-step design of the LQR controller for intelligent control of Quadrotor helicopter. The emphasis will be on the linearised state modelling of the quadrotor helicopter, system's performance index (state and control weighting matrices, (**Q** and **R**), the feedback gain matrix (**K**), and the tuning.

## 2 LQR controller

Linear quadratic regulator (LQR) is a modern control technique that uses state-space to analyse and design a system. The state-space representation of the quadcopter has been developed and used in designing an effective LQR controller. The state-space equations are usually of the form, (1) [1–3]:

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

where  $x$  is the state vector,  $y$  is the output vector,  $u$  is the input/control vector.  $A$ ,  $B$ ,  $C$ ,  $D$  are system matrix, input matrix, output

matrix, and feed forward matrix, respectively. The state vector  $x$  derived from (18), Appendix, is given by  $x = (\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}, x, \dot{x}, y, \dot{y}, z, \dot{z})^T$  and the input vector  $u$  is given by  $u = (u_1, u_2, u_3, u_4)^T$ .

## 3 Derivation of the quadcopter's state-space equation

Using aerodynamic laws, as in (18) and for simplicity in the mathematical calculations, only the linear and angular accelerations are considered for the state vector's derivative (&). This would result in the state-space equations, as in (2) [1].

$$\dot{X} = \begin{cases} \ddot{\phi} = \dot{\theta}\psi\left(\frac{I_{xx} - I_{zz}}{I_{xx}}\right) + \dot{\theta}\left(\frac{J}{I_{xx}}\right)\Omega + \frac{1}{I_{xx}}U_2 \\ \ddot{\theta} = \dot{\phi}\psi\left(\frac{I_{zz} - I_{xx}}{I_{yy}}\right) - \dot{\phi}\left(\frac{J}{I_{yy}}\right)\Omega + \frac{1}{I_{yy}}U_3 \\ \ddot{\psi} = \dot{\theta}\dot{\phi}\left(\frac{I_{xx} - I_{yy}}{I_{zz}}\right) + \frac{1}{I_{zz}}U_4 \\ \ddot{Z} = g - (\cos\phi\cos\theta)\frac{1}{m}U_1 \\ \ddot{x} = (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)\frac{1}{m}U_1 \\ \ddot{y} = (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)\frac{1}{m}U_1 \end{cases}\quad (2)$$

where the input vector,  $u$  is defined, as in (3).

$$\begin{aligned}U_1 &= b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 &= b(-\Omega_2^2 + \Omega_4^2) \\ U_3 &= b(\Omega_1^2 + \Omega_3^2) \\ U_4 &= d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2)\end{aligned}\quad (3)$$

Effective LQR designs are only based on linear state-space models [2]. However, (3) is non-linear and thus, needs to be linearised.

## 4 Linearisation of the quadcopter

In linearising, the designer has to choose and work around the stable operating regions of the system which is to be controlled [3]. Here, the stable hovering position of the quadcopter was seen as the stable operating region suitable for the design and, thus chosen. Therefore, linearisation of (3) and other linearisation required in the LQR design would be done around this stable hovering

position. Equation (4) holds true for the quadcopter at its stable hovering position according to [3]. Where subscript  $h$  denotes hovering.

$$\left\{ \begin{array}{l} \theta = \phi = \psi = \dot{\theta} = \dot{\phi} = \dot{\psi} = \ddot{\theta} = \ddot{\phi} = \ddot{\psi} = 0 \\ \dot{x} = \ddot{x} = \dot{y} = \ddot{y} = \dot{z} = \ddot{z} = 0 \\ V_h = V_1 = V_2 = V_3 = V_4 \\ \Omega = \Omega_h \\ \dot{\Omega} = \ddot{\Omega} = 0 \end{array} \right\} \quad (4)$$

Applying linearised equations to this LQR design was a bit difficult since their state-space equations would not conform to the format as in (1). According to [4], Taylor series could be applied in linearising non-linear equations. However, in the LQR design, linearisation was simply achieved by approximating (2). Using the linearisation process removes the offsets from the dynamics of the system. These offsets are accounted for in the controller design after the gains are calculated and found to achieve satisfactory results [1]. The same technique was implemented in this application to design the LQR controller. Equation (5) shows the linear equations that were achieved after applying this technique [1].

$$\left\{ \begin{array}{l} \ddot{\phi} = \frac{1}{I_x} U_2 \\ \ddot{\theta} = \frac{1}{I_y} U_3 \\ \ddot{\psi} = \frac{1}{I_z} U_4 \\ \ddot{x} = u_x \frac{1}{m} U_1 \\ \ddot{y} = u_y \frac{1}{m} U_1 \\ \ddot{z} = u_z \end{array} \right\} \quad (5)$$

where

$$\left\{ \begin{array}{l} u_x = \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ u_y = \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\ u_z = g - (\cos \phi \cos \theta) \frac{1}{m} U_1 \end{array} \right\} \quad (6)$$

Another technique that was adopted in this LQR design was the reduction technique [1]. This was implemented in order to simplify the mathematical calculations here. With this technique, the state vector which is given as  $X = (\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}, x, \dot{x}, y, \dot{y}, z, \dot{z})^T$ , though consists of 12 members, will be reduced into the following; altitude, attitude, and position controls as in (7). The altitude control would be characterised by  $(z, \dot{z})^T$ , while the attitude control would be characterised by  $(\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi})^T$ . On the other hand, the position control would be characterised by  $(x, \dot{x}, y, \dot{y})^T$ .

$$X = \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \\ x \\ \dot{x} \\ y \\ \dot{y} \\ z \\ \dot{z} \end{bmatrix} = \begin{bmatrix} x1 \\ x1 \\ x3 \\ x4 \\ x5 \\ x6 \\ x7 \\ x8 \\ x9 \\ x10 \\ x11 \\ x12 \end{bmatrix} = (\text{Attitude, Position, Altitude})^T \quad (7)$$

As noted in (2), only linear and angular accelerations were considered in writing the state-space equation. However, with the above categorisation of the state vector ( $X$ ) into altitude, attitude, and position controls, new and much reduced state space-equations had to be considered. In the state-space equation, as in (8) (for attitude), the following were considered; (a) attitude of the quadcopter is characterised by  $(\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi})^T$ ; where  $(\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi})^T$  is equal to  $(x_1, x_2, x_3, x_4, x_5, x_6)^T$  from (7); (3) and, in (5), it was realised that;

$$\begin{aligned} \ddot{\phi} &= \frac{1}{I_x} U_2 \\ \ddot{\theta} &= \frac{1}{I_y} U_3 \\ \ddot{\psi} &= \frac{1}{I_z} U_4 \end{aligned}$$

In order to obtain the state-space equation for attitude to conform to the format  $\dot{x} = Ax + Bu$  as in (3),  $x$  had to be made equal to both  $(\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi})^T$  and  $(x_1, x_2, x_3, x_4, x_5, x_6)^T$ . In addition its derivative,  $\dot{x}$  be made equal to both  $(\dot{\phi}, \dot{\phi}, \dot{\theta}, \dot{\theta}, \dot{\psi}, \dot{\psi})^T$  and  $(x_1, x_2, x_3, x_4, x_5, x_6)^T$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_A \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{I_x} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{I_z} \end{bmatrix}}_B \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} \quad (8)$$

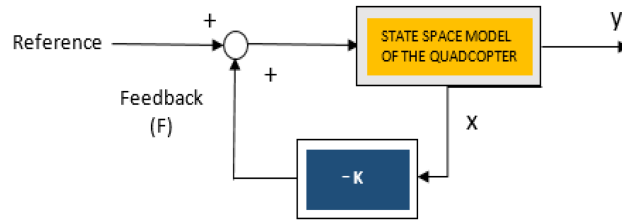
In deriving the state-space (9), the following were considered; (a) position of the quadcopter is characterised by  $(x, \dot{x}, y, \dot{y})^T$ ; (b) From (7),  $(x, \dot{x}, y, \dot{y})^T$  is equal to  $(x_7, x_8, x_9, x_{10})^T$ . From (5), it was realised that

$$\begin{aligned} \ddot{x} &= u_x \frac{1}{m} U_1 \\ \ddot{y} &= u_y \frac{1}{m} U_1 \end{aligned}$$

In order to make sure that the state-space equation conforms to the  $\dot{x} = Ax + Bu$  format,  $x$  had to be equal to both  $(x, \dot{x}, y, \dot{y})^T$  and  $(x_7, x_8, x_9, x_{10})^T$ . Its derivative  $\dot{x}$  be equal to both  $(\dot{x}, \ddot{x}, \dot{y}, \ddot{y})^T$  and  $(\dot{x}_7, \dot{x}_8, \dot{x}_9, \dot{x}_{10})^T$

$$\begin{bmatrix} \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \\ \dot{x}_{10} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_A \begin{bmatrix} x_7 \\ x_8 \\ x_9 \\ x_{10} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ a & 0 \\ 0 & 0 \\ 0 & a \end{bmatrix}}_B \begin{bmatrix} U_x \\ U_y \end{bmatrix} \quad (9)$$

where symbol  $a$  in matrix  $B$  is equal to  $(1/m)U_1$ . In writing the state-space equation for altitude, as in (10), the following were considered; (a) Altitude of the quadcopter is characterised by  $z$  and  $\dot{z}$ ; (b) from (7),  $z$  is equal to  $x_{11}$  and  $\dot{z}$  is equal to  $x_{12}$ ; (c) and in (5), it was shown that  $\ddot{z}$  is equal to  $U_z$ . To make sure the state-space equation conforms to the format  $\dot{x} = Ax + Bu$ ,  $x$  had to be equal to both  $(z, \dot{z})^T$  and  $(x_{11}, x_{12})^T$ . Whereas,  $\dot{x}$  (the derivative of  $x$ ) be equal to both  $(\dot{z}, \ddot{z})^T$  and  $(\dot{x}_{11}, \dot{x}_{12})^T$ .



**Fig. 1** Closed loop control system for the Quadcopter

$$\begin{bmatrix} \dot{x}_{11} \\ \dot{x}_{12} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_A \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_B [U_z] \quad (10)$$

After obtaining these state-space equations, they were verified using them on the quadcopter plant, the system would be both controllable and observable.

## 5 Determining the weighting matrices of the cost function

In LQR designs, the system's performance index is characterised by a cost function ( $J$ ) for which the controller seeks to minimise [4]. This cost function is given by the formula;

$$J = \int_0^{\infty} [x^T Q x + u^T R u] dt \quad (11)$$

where  $Q$  is the state weighting matrix with real symmetry and positive semi-definite in nature.  $R$  is the control weighting matrix of real symmetry but positive definite in nature [5]. These weighting matrices help determine the relative importance of the existing error as well as the energy expenditure of the system [4]. It is, therefore, important that, for a successful LQR design, these parameters be chosen accurately.

Here, a hybrid form of the classical approach based on the Bryson's method [6] and the trial-and-error methods are combined. This method was chosen for its ability to offset the disadvantages of just using the trial-and-error or Bryson's method. The Bryson's method was first used in determining the initial  $Q$  and  $R$  weighting matrices. The trial-and-error, then, was relied on to fine-tune these two parameters to achieve a better performance of the controller [7].

The Bryson's Rule: According to this rule,  $Q$  and  $R$  are diagonal matrices whose diagonal elements are, respectively, expressed as the reciprocals of the squares of the maximum acceptable values of the state variable ( $X$ ) and the input control variable ( $u$ ). The diagonal elements  $Q_{ii}$  of matrix  $Q$ , thus, can be written as [7];

$$Q_{ii} = \frac{1}{\text{maximum acceptable value of } X_i^2} \quad (12)$$

where  $i \in (1, 2, 3, \dots, l)$

The diagonal elements  $R_{jj}$  of matrix  $R$ , also, can be written as [5];

$$R_{jj} = \frac{1}{\text{maximum acceptable value of } u_j^2} \quad (13)$$

where  $j \in (1, 2, 3, \dots, k)$

Applying Bryson's rule to the state-space equation for attitude, as in (9), the following initial  $Q$  and  $R$  values were obtained;

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad R = 0.3$$

Also applying Bryson's rule to the state-space equation for position, as in (10), the following initial  $Q$  and  $R$  values were obtained;

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}; \quad R = 0.5$$

Finally applying Bryson's rule to (10) (altitude), the following initial  $Q$  and  $R$  values were achieved;

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad R = 1$$

## 6 Obtaining the feedback gain matrix through Riccati equation

After obtaining the  $Q$  and  $R$  matrices above, they had to be substituted into the algebraic Riccati equation, as in (14) to solve for  $P$  [8].

$$A \times P + P A - P B R^{-1} B^T P + Q = 0 \quad (14)$$

With  $P$  solved, the feedback gain matrix ( $K$ ) would then be calculated using (15) [8].

$$K = R^{-1} B^T P \quad (15)$$

MATLAB was used as it provides a convenient way of solving for  $K$  by just using the following command

$$K = \text{lqr}(A, B, Q, R) \quad (16)$$

The MATLAB command, as in (16), was also used to derive the  $K$  values for attitude, position, and altitude controllers. In the altitude controller, for instance, where

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \quad Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$R = 1$ , the values obtained for  $K = [1 \ 1.7321]$ .

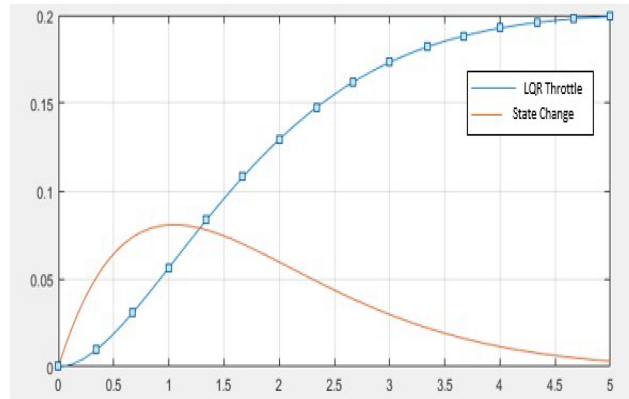
## 7 Closed-loop system with LQR controller

With the feedback gain matrix ( $K$ ) and state-space equations obtained, the closed-loop system of the quadcopter can now be developed, Fig. 1.

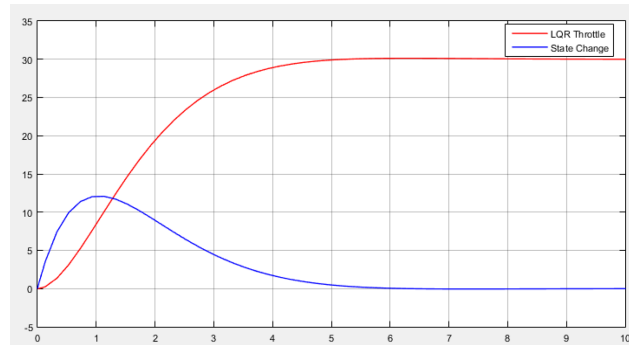
Where the state-space model is of the format, as in (1) and the state feedback ( $F$ ) obeys the optimal control law, which is expressed by (17)

$$F = -Kx \quad (17)$$

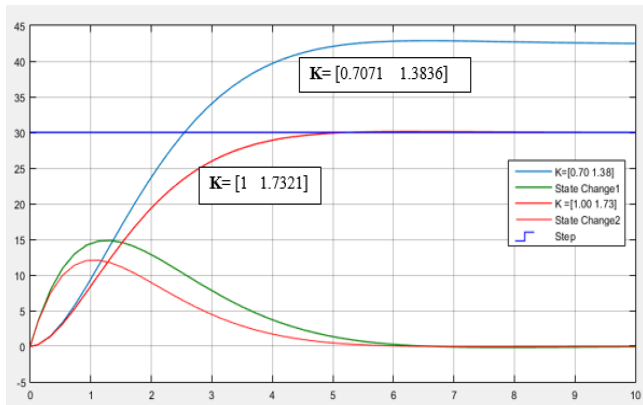
After realising the closed-loop system with the quadcopter using LQR controller, simulations have been conducted to ascertain its performance in terms of meeting the design specifications.



**Fig. 2** Simulating the LQR controller with  $K = [1 \ 1.7321]$  at a throttle position of 0.2 m



**Fig. 3** Simulating the LQR controller with  $K = [1 \ 1.7321]$  at a throttle position of 30 m



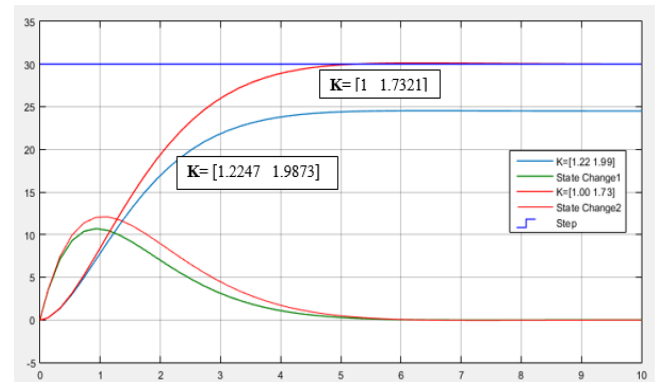
**Fig. 4** Simulating with  $K = [1 \ 1.7321]$  and  $K = [0.7071 \ 1.3836]$  at a throttle position of 30 m

## 8 Tuning the LQR controller

In tuning the LQR controller, much attention was given to the feedback gain matrix ( $K$ ). This is because, until the right  $K$  values have been reached, the performance of the LQR controller would not be satisfactory. With the calculated feedback gain,  $K = [1 \ 1.7321]$  (for altitude), the closed-loop system of the quadcopter was simulated at different throttle positions as seen in Figs. 2 and 3.

It was realised from the above simulations, that the LQR controller could control the dynamic response of the plant with no overshoot. It was, also, able to optimally change the plant's state to zero (in  $\sim 5$  s). These met the design specifications required. However, the rise time and settling time which were expected to be around 2 s remained same in both simulations (at 5 s). These did not meet the expected specifications, thus, the need to re-tune the feedback gain,  $K$ .

Since  $Q$  and  $R$  weighting matrices have corresponding effects on the values of  $K$ , some guidelines were followed to obtain the right values for  $K$  [9];



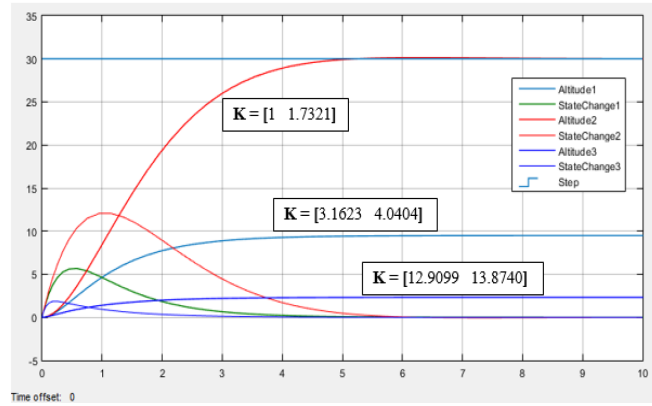
**Fig. 5** Simulating with  $K = [1 \ 1.7321]$  and  $K = [1.2247 \ 1.9873]$  at a throttle position of 30 m

- The larger the values of  $R$ , the lower  $K$  becomes and the slower the state variables approach zero.
- The lower the values of  $R$ , the higher  $K$  becomes and the faster the state variables approach zero.
- The larger the values of  $Q$ , the higher  $K$  becomes and the faster the state variables approach zero.
- The lower the values of  $Q$ , the lower  $K$  becomes and the slower the state variables approach zero.

Using the above guidelines for tuning, the LQR controller was simulated with different values of  $K$  (which were calculated using MatLab command by varying  $Q$  and  $R$ ).

Figs. 5 and 6 show the response curves of these simulations. In both Figs. 4 and 5 for example, the value of  $R$  was maintained at 1 while  $Q$  was, respectively, varied from  $[1 \ 0; 0 \ 1]$  to  $[0.5 \ 0; 0 \ 0.5]$  and  $[1.5 \ 0; 0 \ 1.5]$ . These corresponded to  $K$  values being varied from  $[1 \ 1.7321]$  to  $[0.7071 \ 1.3836]$  and  $[1.2247 \ 1.9873]$ , respectively.

It was realised from the response curves that, the lower  $K$  became, not only did the state variables slowly change to zero but the controller's response became faster (as rise time became



**Fig. 6** Simulating the LQR controller with different  $K$  matrices at a throttle position of 30 m

shorter). There were overshoots and the response never settled at the commanded value as can be seen in Fig. 4.

Also the higher  $K$  became, not only did the state variables change to zero faster but the controller's response became slower (as its rise time became longer). It also never reached the commanded value as seen in Fig. 5.

In Fig. 6, the value of  $Q$  was maintained at  $[1 \ 0; \ 0 \ 1]$ , while  $R$  was varied from 1 to 0.1 and, then, 0.006. These corresponded to  $K$  values of  $[1 \ 1.7321]$  to  $[3.1623 \ 4.0404]$  and, then, to  $[12.9099 \ 13.8740]$ , respectively. Since in both cases,  $K$  was being increased, the state variables changed to zero faster and the controller's response became slower (as its rise time became longer) and never reached the commanded value. It should be noted that, despite the number of attempts to tune the rise time and settling time of the LQR controller to 2 s, they proved difficult. For this reason, the tuning finally settled on  $K = [1 \ 1.7321]$ , since it had no overshoot, no steady-state error and was robust.

## 9 Conclusion

A linearised LQR dynamics Simulink model controller for quadrotor helicopter has been developed. The model verification was successful, and it has been seen that the model dynamics faithfully responds to the commanded inputs. The designed LQR controller was able to meet the performance parameters in terms of overshoot, setting time, and response time.

The developed controller was simulated with under various conditions and the effect the  $Q$  and  $R$  weighting matrices on the feedback gain matrix  $K$ . It was found that for lower values of  $K$ , the controller's response was faster with some issues in the overshoots and settling time. It was, also, found that controller's stability hinges on the right tuning parameter of the gain matrix  $K$ .

## 10 References

- [1] Okyere, E.: 'Comparative analysis and performance studies of PID, LQR and Fuzzy logic control algorithms for quadcopters'. MSc, University of Derby, 2016
- [2] Kumar, A., Sharma, S., Mitra, R.: 'Design of type-2 fuzzy controller based on LQR mapped fusion function', Electronics & Computer Engineering Department, Indian Institute of Technology Roorkee, Uttarakhand, India, 2012
- [3] Poyi, G.T.: 'A Novel Approach To The Control Of Quad-Rotor Helicopters Using Fuzzy- Neural Networks', University of Derby, UK, 2014
- [4] Ogata, K.: 'Modern control engineering' (Prentice-Hall, Boston, 2010, 5th edn.)
- [5] Hassani, K., Lee, W.: 'Optimal tuning of linear quadratic regulators using quantum particle swarm optimization', University of Ottawa, School of Electrical Engineering and Computer Science 161 Louis Pasteur, Ottawa, Ontario, Canada, 2014
- [6] Bryson, A.E.: 'Control of spacecraft and aircraft' (Princeton University Press, Princeton, NJ, 2015)
- [7] Hespanha, J.P.: 'Linear systems theory' (Princeton University Press, Princeton, 2009)
- [8] Dharan, A., Storhaug, S.H.O., Karimi, H.R.: 'LQG control of a semi-active suspension system equipped with MR rotary brake', Department of Engineering, Faculty of Engineering and Science, University of Agder, N-4898 Grimstad, Norway, 2012
- [9] Rahman, A., Ali, S.M.: 'Design And analysis Of A quadratic optimal control system For A type One plant model', Department of Electronics and Communication Engineering, National University of Singapore, Singapore; School of Mechanical & Aerospace Engineering, Nanyang Technological University, Singapore, 2013

## 11 Appendix

Analytical model for the quadrotor helicopter [3] (see (18))

$$\begin{aligned}
 V &= \frac{JR\dot{\Omega}}{k_q} + k_e\Omega + \frac{RD\Omega^2}{k_q} \\
 \begin{bmatrix} V \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} &= \begin{bmatrix} \frac{2\rho A}{m} \left[ \frac{f\eta k_t}{k_q} \right]^2 (V_1^2 + V_2^2 + V_3^2 + V_4^2) (\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi) \\ \frac{2\rho A}{m} \left[ \frac{f\eta k_t}{k_q} \right]^2 (V_1^2 + V_2^2 + V_3^2 + V_4^2) (\sin\psi \sin\theta \cos\phi - \cos\psi \sin\phi) \\ \frac{2\rho A}{m} \left[ \frac{f\eta k_t}{k_q} \right]^2 (V_1^2 + V_2^2 + V_3^2 + V_4^2) (\cos\theta \cos\phi) - g \\ \frac{2\rho A}{m} \left[ \frac{f\eta k_t}{k_q} \right]^2 (V_2^2 - V_4^2) \\ \frac{2\rho A}{m} \left[ \frac{f\eta k_t}{k_q} \right]^2 (V_3^2 - V_1^2) \\ \frac{J}{I_{zz}} (\dot{\Omega}_1 + \dot{\Omega}_3 - \dot{\Omega}_2 - \dot{\Omega}_4) + \frac{D}{I_{zz}} (\Omega_1^2 + \Omega_3^2 - \Omega_2^2 - \Omega_4^2) \end{bmatrix} \quad (18)
 \end{aligned}$$