Comparison of PID and LQR controllers on a quadrotor helicopter

Demet Canpolat Tosun[1], Yasemin Işık[2], Hakan Korul[3]

***Abstract*—** This paper focuses on a quadrotor model, named as Qball-X4 developed by Quanser. The quadrotor simulation model includes both linear and nonlinear X, Y, and Z position, roll/pitch and yaw dynamics. The objective is to determine the control strategy that to delivers better performance with respect to quadrotor's desired attitudes. The Linear Quadratic Regulator (LQR) and PID control techniques are used to control the height, X and Y position, yaw and roll/pitch angle. The results of position control are obtained through simulations to reach desired attitudes. Various simulation parameters have been tested to demonstrate the validity of the proposed controllers and the effectiveness of the reconfigurable controller designs in LQR and PID control. Comparative simulation results are presented for the position controls along X, Y, and Z axis, roll/pitch and yaw angles of the Qball-X4.

***Keywords*—** Quadrotor, Qball-X4, LQR control, PID control, axis control, angle control, Matlab/Simulink

I. Introduction

Unmanned Aerial Vehicles (UAVs) has been the research subject of several recent applications. As an example of unmanned aerial vehicle systems, quadrotors are taken into account with the simple mechanical structure, being affordable and easy to fly.

Different from the classical helicopter, which uses a single main-rotor to lift the helicopter and one auxiliary tail-rotor to adjust the helicopter's attitude, a quadrotor is a special flying helicopter, which is composed of four rotors to lift the helicopter and adjust its attitude [1].

A quadrotor is controlled by manipulating thrust forces from invidual rotors as well as balancing drag torque. To hover the quadrotor, all rotors apply equal thrust forces as illustrated in Fig. 1(c). The yaw motion is produced by applying more thrust to rotors rotating in one direction as illustrated in Fig. 1(a), (b). The pitch and roll motions are the results of the application of more thrust to one rotor and less thrust to its diametrically opposite rotor as shown in Fig. 1(d).

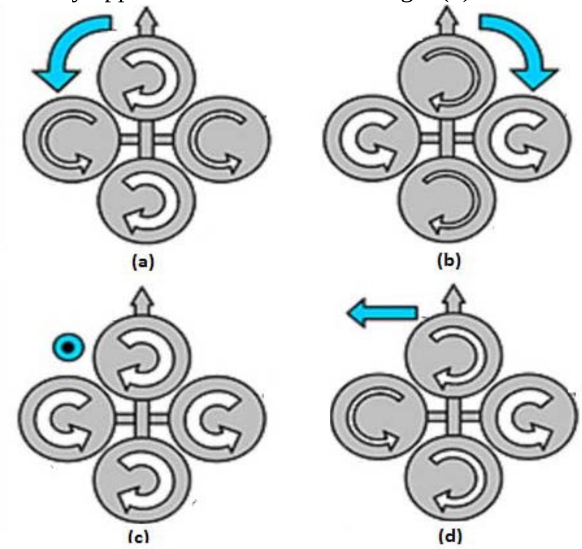

Fig. 1 Quadrotor dynamics, (a) and (b) difference in torque to manipulate the yaw angle(ψ); (c) hovering motion and vertical propulsion due to balanced torques; (d) difference in thrust to manipulate the roll angle and lateral motion [2].

In this study, the quadrotor named as Qball-X4 which is developed by Quanser is used.

In the literature, several research studies performed in both simulations and experiments with the Qball-X4.

Some of these are as follows:

Sadeghzadeh, Mehta, Chamseddine, and Zhang proposed a Gain-Scheduled PID controller for fault-tolerant control of the Qball-X4 system in the presence of actuator faults [3].

Abdolhosseini, Zhang and Rabbath developed an efficient Model Predictive Control (eMPC) strategy and tested it on the unmanned quadrotor helicopter testbed Qball-X4 to address the main drawback of standard MPC with high computational requirement [4].

Hafez, Iskandarani, Givigi, Yousefi and Beaulieu proposed a control strategy for tactic switching, going from line abreast formation to dynamic encirclement. Their results show that applying the MPC strategy solves the problem of tactic switching for a team of UAVs (Qball-X4) in simulation [5].

Abdolhosseini, Zhang, and Rabbath have tried to design an autopilot control system for the purpose of three-dimensional

[1]MSc. Student, Faculty of Aeronautics and Astronautics, Anadolu University 26470 Eskisehir / Turkey (e-mail: demetcanpolat@anadolu.edu.tr)

[2]Asst. Prof., Faculty of Aeronautics and Astronautics, Anadolu University 26470 Eskisehir/Turkey (e-mail: yaisik@anadolu.edu.tr)

[3]Asst. Prof., Faculty of Aeronautics and Astronautics, Anadolu University, Turkey (e-mail: hkorul@anadolu.edu.tr)

trajectory tracking of the Qball-X4. Besides, they successfully implemented a constrained MPC framework on the Qball-X4 to demonstrate effectiveness and performance of the designed autopilot in addition to the simulation results [6].

Chamseddine, Zhang, Rabbath, Fulford and Apkarian worked on actuator fault-tolerant control (FTC) for Qball-X4. Their strategy is based on Model Reference Adaptive Control (MRAC). Three different MRAC techniques which are the MIT rule MRAC, the Conventional MRAC (C-MRAC) and the Modified MRAC (M-MRAC) have been implemented and compared with a Linear Quadratic Regulator (LQR) controller [7].

In this study, the PID and LQR control techniques have been used to control the three-dimensional motion of the Qball-X4.

II. The Qball-X4 Quadrotor Model

The Qball-X4 is a test platform suitable for a wide variety of UAV research applications. The Qball-X4 is propelled by four motors fitted with 10-inch propellers. The quadrotor is covered within a protective carbon fiber cage. The Qball-X4 ensures safe operation as well as opens the possibilities for a variety of novel applications with this proprietary design.

The Qball-X4 has onboard avionics data acquisition card (DAQ), named HiQ, and the embedded Gumstix computer to measure onboard sensors and drive the motors. Many research applications are enabled through the HiQ which has a high-resolution inertial measurement unit (IMU) and avionics input/output (I/O) card. Besides, the Qball-X4 comes with real-time control software, QuaRC. By means of the QuaRC, developers and researchers can rapidly develop and test controllers through a Matlab/Simulink interface.

QuaRC is a rapid-prototyping and production system for real-time control that is so tightly integrated with Simulink that it is virtually transparent. QuaRC consists of a number of components that make this seamless integration possible [8]:

- **QuaRC Code Generation**: QuaRC extends the code generation capabilities of Simulink Coder by adding a new set of targets, such as a Windows target and QNX x86 target. These targets appear in the system target file browser of Simulink Coder. These targets change the source code generated by Simulink Coder to suit the particular target platform. QuaRC automatically compiles the C source code generated from the model, links with the appropriate libraries for the target platform and downloads the code to the target.
- **QuaRC External Mode Communications:** QuaRC provides an "external mode" communications module that allows the Simulink diagram to communicate with real-time code generated from the model.
- **QuaRC Target Management:** Generated code is managed on the target by an application called the QuaRC Target Manager. It is the QuaRC Target Manager that allows generated code to be seamlessly downloaded and run on the target from Simulink.

QuaRC's open-architecture structure allows user to develop powerful controls. QuaRC can target the Gumstix embedded computer. The Gumstix computer automatically generates codes and executes controllers on-board the vehicle. With this structure, users can observe sensor measurements and tune parameters in realtime from a host computer while the controller is performing on the Gumstix [9].

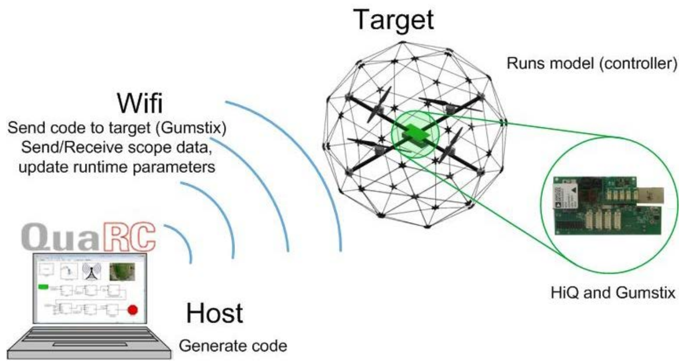

Fig. 2 Communication hierarchy [9]

The interface between the Qball-X4 and Matlab/Simulink is the QuaRC. The developed controller models in Simulink are downloaded and compiled into executables on the Gumstix by the QuaRC. The configuration of the system is as shown in Fig. 2.

The required hardware and software for Qball-X4 are as follows [9]:

- Qball-X4: Qball-X4 as shown in Fig. 3,
- HiQ: QuaRC aerial vehicle data acquisition card (DAQ),
- Gumstix: The QuaRC target computer. An embedded, Linux-based system with QuaRC runtime software installed,
- Batteries: Two 3-cell, 2500 mAh Lithium-Polymer batteries,
- Real-Time Control Software: The QuaRC-Simulink configuration.

III. Qball-X4 Dynamics

In this section, the dynamic model of the Qball-X4 is decribed. Both nonlinear and linearized models are decsribed to develop controllers.

The axes of the Qball-X4 are denoted (x, y, z) as shown in Fig. 3.The angles of the rotation about x, y, and z are roll/pitch, and yaw, respectively. The global workspace axes are denoted (X, Y, Z) and are defined with the same orientation as the Qball-X4 sitting upright on the ground.

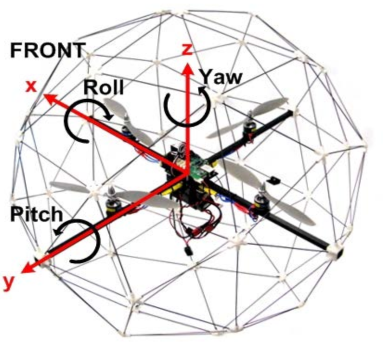

Fig. 3 Qball-X4 axes and sign convention [9]

The Qball-X4 uses brushless motors. They are mounted to the frame along the X and Y axes and to the four speed controllers which are also mounted to the frame. The motors and propellers are configured so that the front and back motors spin clockwise and the left and right motors spin counter-clockwise [9].

A. Actuator Dynamics

The relationship between the thrust (F_i) generated by i th motor and the i th PWM input (u_i) is [9]:

$$F_i = K\frac{w}{s+w}u_i \tag{1}$$

where w is the actuator bandwidth and K is a positive gain.

The calculated and verified parameters through experimental studies by Quanser are stated in Table I.

A state variable, v, is defined to represent the actuator dynamics as follows:

$$v = \frac{w}{s+w}u \tag{2}$$

B. Height Model

The vertical motion of the Qball-X4 results from all thrusts generated by the four propellers. Therefore, the height dynamics can be written as [9]:

$$M\ddot{Z} = 4F\cos(r)\cos(p) - Mg \tag{3}$$

where F is the thrust generated by each propeller M is the mass of the quadrotor, Z is the height and r and p are the roll and pitch angles, respectively. With the assumption that the roll and pitch angles are close to zero, Eq. (3) is linearized and written in the following state space form as follows [9]:

$$\begin{bmatrix}\dot{Z}\\ \ddot{Z}\\ \dot{v}\\ \dot{s}\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & \frac{4K}{M} & 0\\ 0 & 0 & -w & 0\\ 1 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}Z\\ \dot{Z}\\ v\\ s\end{bmatrix} + \begin{bmatrix}0\\ 0\\ w\\ 0\end{bmatrix}u + \begin{bmatrix}0\\ -g\\ 0\\ 0\end{bmatrix} \tag{4}$$

C. X-Y Position Model

The motion along the X and Y axes are coupled to roll and pitch motions, respectively. The motions are caused by changing roll/pitch angles. With the assumption that the yaw angle is zero, the dynamics of motion along the X and Y axes can be written as [9]:

$$M\ddot{X} = 4F\sin(p) \tag{5}$$

$$M\ddot{Y} = -4F\sin(r) \tag{6}$$

By assuming the roll and pitch angles are close to zero, linearized equations gives the following state-space models [9]:

$$\begin{bmatrix}\dot{X}\\ \ddot{X}\\ \dot{v}\\ \dot{s}\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & \frac{4K}{M}p & 0\\ 0 & 0 & -w & 0\\ 1 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}X\\ \dot{X}\\ v\\ s\end{bmatrix} + \begin{bmatrix}0\\ 0\\ w\\ 0\end{bmatrix}u \tag{7}$$

$$\begin{bmatrix}\dot{Y}\\ \ddot{Y}\\ \dot{v}\\ \dot{s}\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & -\frac{4K}{M}r & 0\\ 0 & 0 & -w & 0\\ 1 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}Y\\ \dot{Y}\\ v\\ s\end{bmatrix} + \begin{bmatrix}0\\ 0\\ w\\ 0\end{bmatrix}u \tag{8}$$

D. Roll/Pitch Model

The roll/pitch motion is modelled as shown in Fig. 4 with the assumption that the rotations about the x and y axes are decoupled.

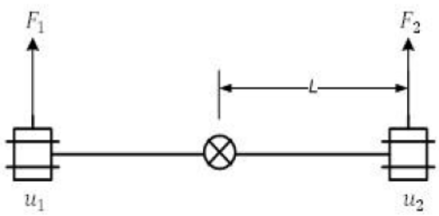

Fig. 4 The roll/pitch axis model [9]

As shown in Fig. 4, two propellers causes the motion in each axis. The difference in the generated thrusts produces the rotation around the center of gravity. The roll/pitch angle, θ, can be formulated using the following dynamics [9]:

$$J\ddot{\theta} = \Delta F L \tag{9}$$

where L is the distance between the propeller and the center of gravity, and

$$J = J_{roll} = J_{pitch} \tag{10}$$

are the rotational inertia of the device in roll and pitch axes.

The difference between the forces generated by the motors are represented as follows [9]:

$$\Delta F = F_1 - F_2 \tag{11}$$

The following state space representation can be derived from the dynamics of the motion and the actuator dynamics [9]:

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \frac{KL}{J} \\ 0 & 0 & -w \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ w \end{bmatrix} \Delta F \tag{12}$$

A fourth state denoted as $\dot{s} = \theta$ can be defined to facilitate the use of integrator in the feedback structure and the augmented system dynamics can be rewritten as follows [9]:

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{v} \\ \dot{s} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{KL}{J} & 0 \\ 0 & 0 & -w & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ v \\ s \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ w \\ 0 \end{bmatrix} \Delta F \tag{13}$$

E. Yaw Model

Yaw motion is caused by the difference between torques exerted by the two clockwise and the two counter-clockwise rotating propellers.

The relation between the torque, τ, generated by each propeller and the PWM input (u) is [9]:

$$\tau = K_y u \tag{14}$$

where K_y is a positive gain. Yaw motion is modeled by the following equation [9]:

$$J_y \ddot{\psi} = \Delta\tau \tag{15}$$

In this equation, J_y is the rotational inertia about the z axis, and the ψ is the yaw angle.

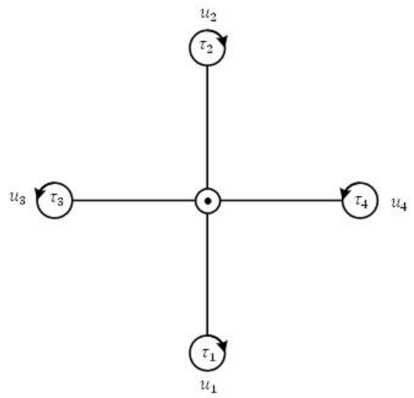

Fig. 5 The yaw axis model with propeller direction of rotation [9]

The resultant torque of the motors, $\Delta\tau$, can be calculated from

$$\Delta\tau = \tau_1 + \tau_2 - \tau_3 - \tau_4 \tag{16}$$

The yaw dynamics can be written in state-space form as follows [9]:

$$\begin{bmatrix} \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{K_y}{J_y} \end{bmatrix} \Delta\tau \tag{17}$$

Table I System parameters [9]

Parameter	Value
K	120 N
w	15 rad/s
J_{roll}	0.03 kg.m^2
J_{pitch}	0.03 kg.m^2
M	1.4 kg
K_y	4 N.m
J_y	0.03 kg.m^2
L	0.2m

IV. PID Control

The block diagram of a closed loop PID controller is shown in Fig. 6.

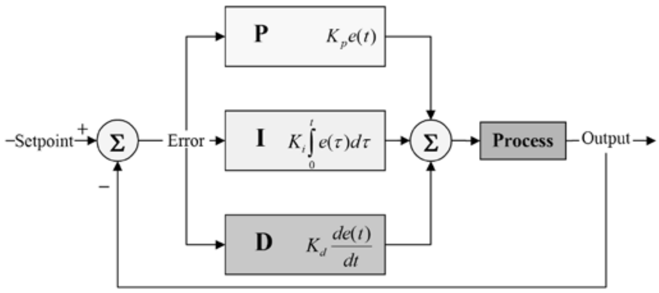

Fig. 6

An ideal interacting PID controller can be represented as [10]

$$G_c(s) = K_c\left(1 + \frac{1}{t_i s} + t_d s\right) \tag{18}$$

K_c is proportional gain, t_i is integral time and t_d is the derivative time.

There are different tuning methods of PID controller. Some methods are empirical methods, some methods are based on frequency response analysis of the system and other methods are based on minimization of performance measures. Despite advances in PID tuning methods the ground reality is that in most of the cases, PID controller is tuned using trial and error method [10].

The chosen controller gains to control angles and positions of the Qball-X4 are given in Table II.

Table II PID controller parameters

Parameter	Proportional	Integral	Derivative
X position	8.6	2.5	4.5
Y position	8.6	2.5	4.5
Height	20	5	10
Roll/pitch angle	14.4	0.82	56.9
Yaw angle	0.2	0	0.1

V. LQR CONTROL

Linear quadratic regulator (LQR) is one of the most commonly used optimal control tecniques for linear systems.This control method takes into account a cost function which depends on the states of the dynamical system and control input to make the optimal control decisions.

A system can be expressed in state space form as

$$\dot{x} = Ax + Bu \tag{18}$$

$$y = Cx \tag{19}$$

and suppose we want to design state feedback control

$$u = -Kx \tag{20}$$

to stabilize the system.

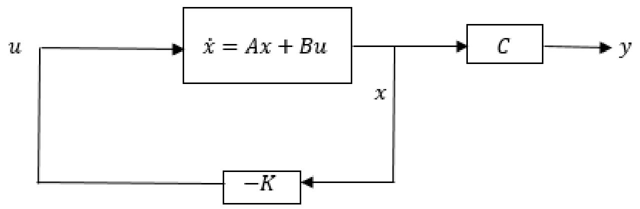

Fig. 7 LQR controller diagram

The closed-loop system using this control becomes

$$\dot{x} = (A - BK)x \tag{21}$$

The design of K is a tradeoff between the transient response and the control effort. The optimal control approach to this tradeoff is to define a cost function and search for the control, $u = -Kx$, that minimizes this cost function.

$$J = \int_0^\infty (x^T Q x + u^T R u)dt \tag{22}$$

where Q is an $n \times n$ positive definite matrix and R is an $n \times n$ positive definite matrix, both are symmetric.

The LQR gain vector K is given by

$$K = R^{-1} B^T P \tag{23}$$

where, P is a positive definite symmetric constant matrix obtained from the solution of matrix algebraic reccatti equation

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \tag{24}$$

The objective in optimal design is to select the K that minimizes the cost function as stated above. The cost function also known as performance index J can be interpreted as an energy function, so that making it small keeps small the total energy of the closed-loop system [11].

As seen from cost function, both the state x(t) and control input u(t) have weights on the total energy of the system.Therefore, if J is small, x(t) and u(t) can not be too large and as a control objective, if we minimize the cost function, the cost function will be an infinite integral x(t).This means that x(t) goes zero as t goes to infinity and guarantees the stability of the closed-loop system.

A. Height Control

For the height control model of the Qball-X4, the state matrices, A and B, obtained from the state-space form of the height model and the gain matrix K is calculated from the Q and R matrices which are chosen suitable for the system. Eventually, the height control model of the Qball-X4 is constructed through Matlab/Simulink as shown in Fig. 8.

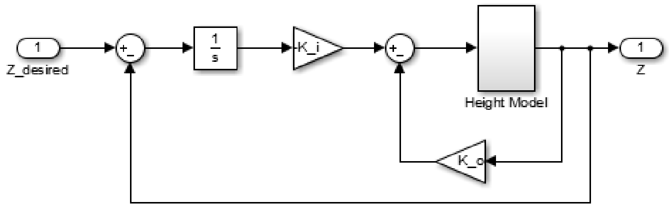

Fig. 8 Simulink model for the height control

B. X-Y Position Control

The Simulink model for X and Y position control is constructed by obtaining state matrices and the suitable weight matrices. The X and Y position control models are as shown in Fig. 9 and Fig. 10, relatively.

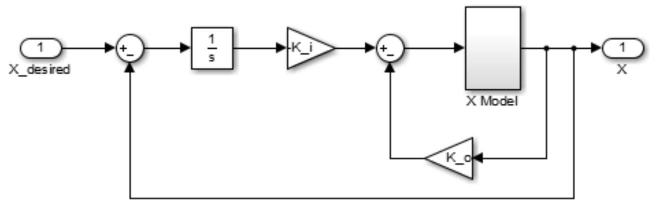

Fig. 9 Simulink model for the X position control

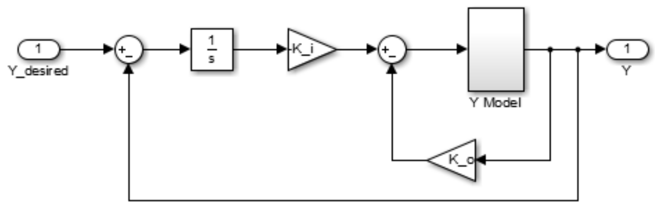

Fig. 10 Simulink model for the Y position control

C. Roll/Pitch Angle Control

In like manner, the state matrices are obtained from the state space form of the roll/pitch model and the weight matrices (Q and R) are assigned and the gain matrix K is calculated to construct the control model.

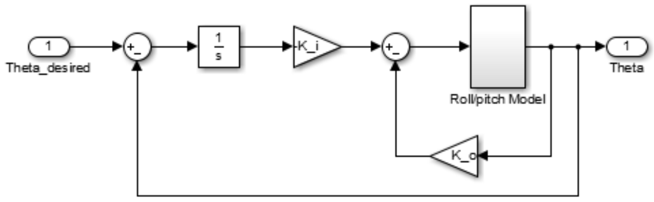

Fig. 11 Simulink model for the roll/pitch angle control

D. Yaw Angle Control

The yaw angle control of the Qball-X4 is contructed as shown in Fig. 12 by means of Simulink.

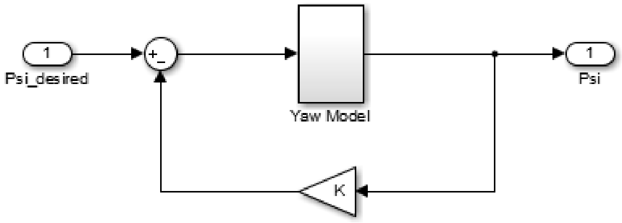

Fig. 12 Simulink model for the yaw angle control

VI. Simulation Results

The Qball-X4 system and the proposed controllers are modeled and simulated in the MATLAB/Simulink environment. The responses of two controllers in terms of reaching desired positions and angles are compared.

If we examine the X and Y positions control to reach a desired value (2m), the results are shown in Fig. 13 and Fig. 14.

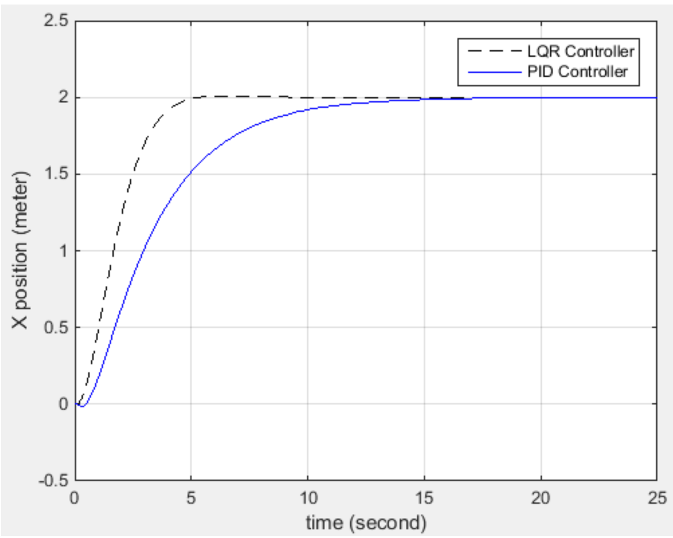

Fig. 13 The X position response

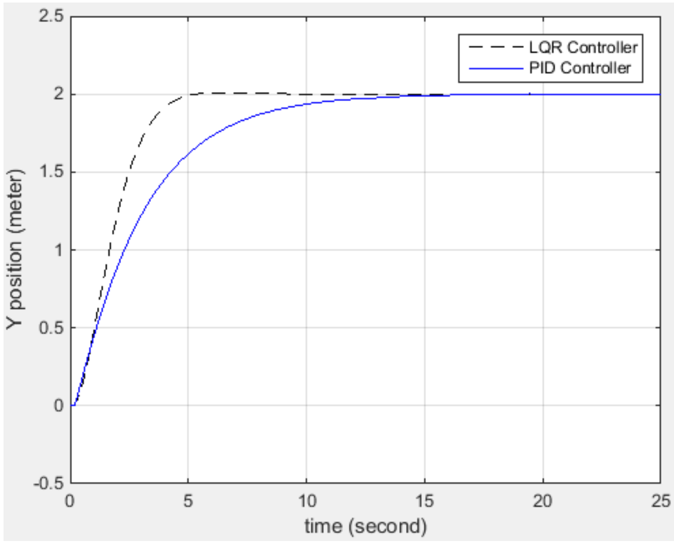

Fig. 14 The Y position response

The vertical motion control of the device is performed via the Simulink model in Fig. 8 and the simulation results are shown in the Fig. 15.

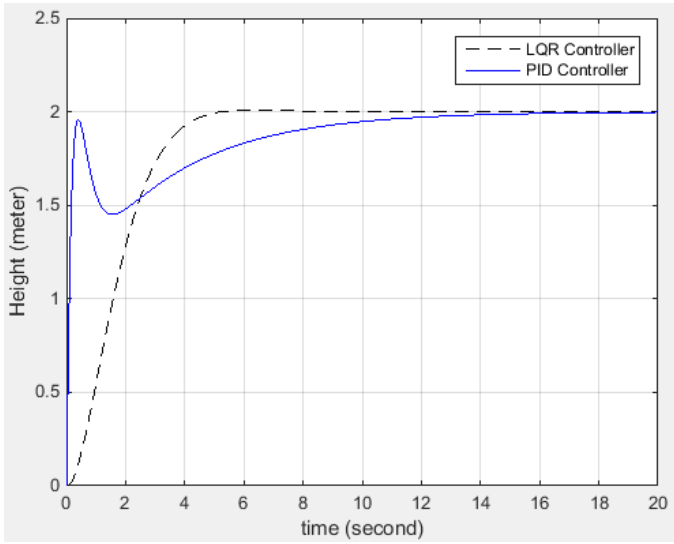

Fig. 15 The height response

The simulation results for the roll/pitch control models are shown in Fig. 16.

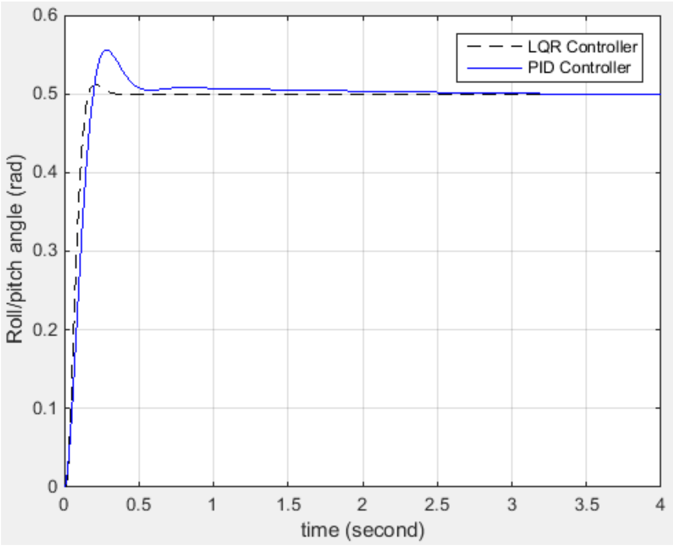

Fig. 16 The roll/pitch angle response

The simulation results which belong to the yaw angle control of the device are shown in Fig. 17. The control objective is to keep the yaw angle 0.8 radian.

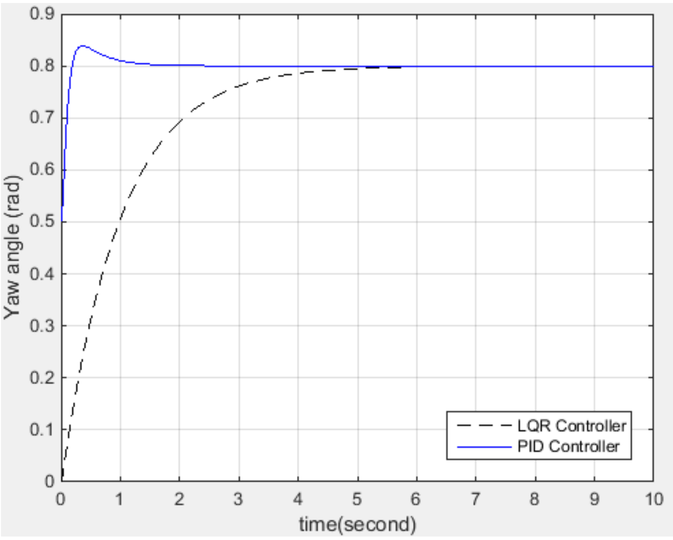

Fig. 17 The yaw angle response

The time response specification for the Qball-X4 system equipped with the proposed controllers are given in Table III, IV, V, VI, and VII.

Table III Summary of the performance characteristics for X position

Time Response Specification	LQR Controller	PID Controller
Settling time (T_s)	4.4539s	24.8540s
Rise time (T_r)	2.7531s	4.4070s
Overshoot	0.4395%	0.0919%

Table IV Summary of the performance characteristics for Y position

Time Response Specification	LQR Controller	PID Controller
Settling time (T_s)	4.4539s	24.6808s
Rise time (T_r)	2.7531s	4.0620s
Overshoot	0.4395%	0.1145%

Table V Summary of the performance characteristics for height

Time Response Specification	LQR Controller	PID Controller
Settling time (T_s)	4.3658s	10.7045s
Rise time (T_r)	2.7344s	0.2235s
Overshoot	0.4333%	0.0041%

Table VI Summary of the performance characteristics for roll/pitch angle

Time Response Specification	LQR Controller	PID Controller
Settling time (T_s)	0.2369s	0.4978s
Rise time (T_r)	0.0929s	0.1331s
Overshoot	2.4832%	10.3963%

Table VII Summary of the performance characteristics for yaw angle

Time Response Specification	LQR Controller	PID Controller
Settling time (T_s)	3.8811s	1.2136s
Rise time (T_r)	2.1747s	0.1274s
Overshoot	0%	4.8199%

From both controller LQR and PID controller's result, it is clear that both are successfully designed but LQR controller exhibits better response and performance.

The linear stability of the system is assured in simulation environment with the control gains which are designed with the weighting matrices, Q and R in LQR control and with chosen suitable P, I, D controller parameters in PID control.

VII. Results and Future Works

In this study, the position controls along X, Y and Z axis, roll/pitch and yaw angle controls are performed and compared with LQR and PID control techniques in the Matlab/Simulink for the Qball-X4 quadrotor model.The LQR controllers are designed for each model. The suggested controllers are tested in simulation environment. The simulation results show that the performance specifications are met through choosing suitable weight matrices for each controller. Because of the LQR technique deals with balance between low control effort and faster response, the matrices are chosen to meet this two performance criteria.

As a conclusion, to meet the control objective, the following directions should be assured:

- Getting system dynamics as closely as possible the real system
- Calculating the control gains with choosing appropriate

weighting matrices.

The future work is to test the proposed controllers experimentally on the Qball-X4 testbed with the positional data obtained from the external camera system (Optitrack camera system).Thus, the real-time performance of the proposed controllers would be examined. Then, performing the research applications suitable for the Qball-X4, including:

- Path planning,
- Obstacle avoidance,
- Sensor fusion,
- Fault-tolerant control, and more

will be the key subjects of the next study.

REFERENCES

[1] Y. Zhang, W. Zhao, T. Lu, and J. Li, "The attitude control of the four-rotor unmanned helicopter based on feedback linearization control," WSEAS Trans. Syst., vol. 12, no. 4, pp. 229–239, 2013.

[2] T. Tsay, "Guidance and control laws for quadrotor UAV," WSEAS Trans. on Systems and Control, vol. 9, pp. 606-613, 2014.

[3] I. Sadeghzadeh, A. Mehta, A. Chamseddine, and Y. Zhang, "Active fault tolerant control of a quadrotor UAV based on gainscheduled pid control," Electrical & Computer Engineering (CCECE), vol. ., pp. 1–4, May 2012.

[4] M. Abdolhosseini , Y. M. Zhang , C. A. Rabbath, "An efficient model predictive control scheme for an unmanned quadrotor helicopter," Journal of Intelligent and Robotic Systems, vol. 70, no.1-4, pp. 27-38, 2013.

[5] A. T. Hafez, M. Iskandarani, S. N. Givigi, S. Yousefi, A. Beaulieu, "UAVs in formation and dynamic encirclement via model predictive control," 19th IFAC World Congress, Cape Town, South Africa, August 24-29, 2014

[6] M. Abdolhosseini, Y. M. Zhang, C. A. Rabbath, "Trajectory tracking with model predictive control for an unmanned quad-rotor helicopter: theory and flight test results," ICIRA'12, Proceedings of the 5th international conference on Intelligent Robotics and Applications, Part I, LNAI 7506, pp. 411–420, 2012.

[7] A. Chamseddine, Y.M. Zhang, C.A. Rabbath, C. Fulford, J. Apkarian, "Model reference adaptive fault tolerant control of a quadrotor UAV," in: AIAA Infotech@Aerospace, St. Louis, Missouri, USA, 29–31 March 2011.

[8] Quanser Inc., "QUARC user's guide". Avaliable: http://www.quarcservice.com/ReleaseNotes/files/quarc_user_guide.html

[9] Quanser Inc., "Qball-X4 user manual," Canada, 2010.

[10] S. Padhee, "Controller Design for Temperature Control of Heat Exchanger System: Simulation Studies," WSEAS Trans. on Systems and Control, Vol. 9, no. 2, pp. 485-491, 2014.

[11] F.L. Lewis, "Linear quadratic regulator (LQR) state feedback design," [PDF document]. Retrieved from Lecture Notes Online Web site: http://www.uta.edu/utari/acs/Lectures/lqr.pdf.