University of Southampton

Faculty of Engineering and Physical Sciences

Electronics and Computer Science

Modelling and Adaptive Control of a Quadcopter

by

Xuankun Cai

5th September 2023

Supervisor: Matthew Turner
Second Examiner: Lie-Liang Yang

A dissertation submitted in partial fulfilment of the degree

of MSc Internet of Things

Abstract

A quadrotor model is commonly recognized as nonlinear, underactuated, and strongly coupled model, which is intrinsic challenge against its control system design. In this project, the advantage and challenge of quadrotor are both explained, and a serial simulation result based on Matlab/Simulink are presented. Various controllers implemented in quadrotor are also discussed with advanced solutions and corresponding theory for quadrotor challenges. Then, the process of dynamic model is presented with details and discussion. The design of three controller, based on linear model, is also detailed explained, and an initial conclusion is made to estimate effect from nonlinear terms. Ultimately, a comparison for stability, robustness and complexity is presented for evaluation of different control strategy.

***Key words*– Quadrotor, PID, Feedback Linearization, LQR, Matlab/Simulink, Nonlinear model.**

Statement of Originality

- I have read and understood the ECS Academic Integrity information and the University's Academic Integrity Guidance for Students.
- I am aware that failure to act in accordance with the Regulations Governing Academic Integrity may lead to the imposition of penalties which, for the most serious cases, may include termination of programme.
- I consent to the University copying and distributing any or all of my work in any form and using third parties (who may be based outside the EU/EEA) to verify whether my work contains plagiarised material, and for quality assurance purposes.

You must change the statements in the boxes if you do not agree with them.

We expect you to acknowledge all sources of information (e.g. ideas, algorithms, data) using citations. You must also put quotation marks around any sections of text that you have copied without paraphrasing. If any figures or tables have been taken or modified from another source, you must explain this in the caption and cite the original source.

I have acknowledged all sources, and identified any content taken from elsewhere.

If you have used any code (e.g. open-source code), reference designs, or similar resources that have been produced by anyone else, you must list them in the box below. In the report, you must explain what was used and how it relates to the work you have done.

I have not used any resources produced by anyone else.

You can consult with module teaching staff/demonstrators, but you should not show anyone else your work (this includes uploading your work to publicly-accessible repositories e.g. Github, unless expressly permitted by the module leader), or help them to do theirs. For individual assignments, we expect you to work on your own. For group assignments, we expect that you work only with your allocated group. You must get permission in writing from the module teaching staff before you seek outside assistance, e.g. a proofreading service, and declare it here.

I did all the work myself, or with my allocated group, and have not helped anyone else.

We expect that you have not fabricated, modified or distorted any data, evidence, references, experimental results, or other material used or presented in the report. You must clearly describe your experiments and how the results were obtained, and include all data, source code and/or designs (either in the report, or submitted as a separate file) so that your results could be reproduced.

The material in the report is genuine, and I have included all my data/code/designs.

We expect that you have not previously submitted any part of this work for another assessment. You must get permission in writing from the module teaching staff before re-using any of your previously submitted work for this assessment.

I have not submitted any part of this work for another assessment.

If your work involved research/studies (including surveys) on human participants, their cells or data, or on animals, you must have been granted ethical approval before the work was carried out, and any experiments must have followed these requirements. You must give details of this in the report, and list the ethical approval reference number(s) in the box below.

My work did not involve human participants, their cells or data, or animals.

ECS Statement of Originality Template, updated August 2018, Alex Weddell aiofficer@ecs.soton.ac.uk

Table of Contents

Section Ⅰ. Introduction

Over the past decades, unmanned aerial vehicle (UAV) has attracted much interest and attention, as it presents huge potential in various fields. UAVs is expected to play more important roles in future cyber life. It could be deployed to replace human from danger or repeatability in automatic logistics, transportation, disaster relief and industry detection. Beyond these sceneries, as shown in Fig. 1.1, it could also synchronously operate as mobile network server to enhance performance of entire system. Behind all of deployment sceneries and requirement of system, the foundation of that is to ensure the UAV could work with stability, accuracy, and robustness.

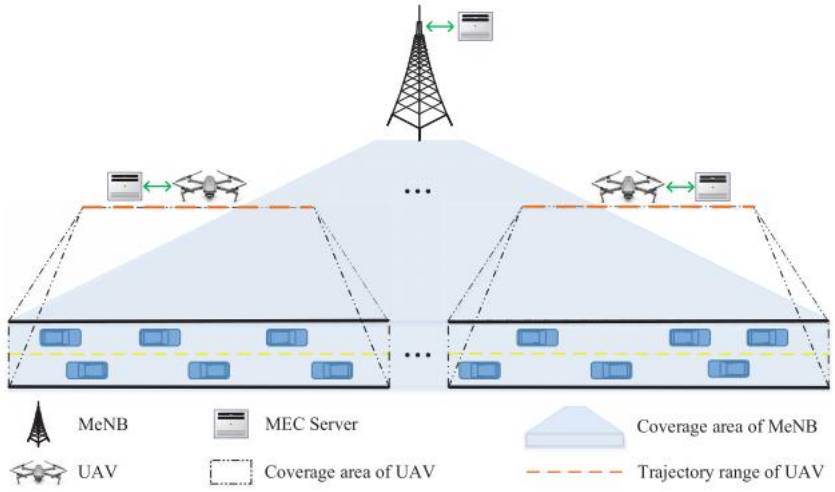

Fig. 1.1. A simplified Mobile Edge Computing- and UAV-assisted vehicular network scenario for multi-dimensional resource management [1].

Fig. 1.2. Illustration of quadrotor [2].

Another important information from UAV application sceneries is that there will be quiet limited maneuvering space and possible lack of device support in practical situation. Such feature makes quadcopter (also named quadrotor) distinct from various UAV categories. It is an on-coaxial multi-rotor aircraft as shown in Fig. 1.2. The advantages and suitability of quadrotor are discussed below.

- **Structure Simplicity and Compactness:**

  Quadrotors have a simple mechanical design with few moving parts compared to traditional helicopters or fixed-wing UAVs. This simplicity enhances reliability and maintainability. Its compactness and lightweight on size also making them easily transportable and deployable in complicated civic environments.

- **Vertical Takeoff and Landing (VTOL) & Hovering Capability:**

  Quadrotors can not only take off or land vertically with few requirements of runways or catapult systems, but also hover in place by a fixed position. These capabilities allow it to operate in narrow spaces to steadily complete tasks, such as observation, data collection, or interaction with the environment.

- **Ease of Control & Maneuverability:**

  Quadrotors are relatively easier to control compared to fixed-wing UAVs with high agility of changing direction quickly. This feature makes them suitable for tasks that require precise positioning, such as aerial photography, surveillance, and inspections.

- **Fault Tolerance & Payload Flexibility:**

  The four-rotor configuration provides inherent redundancy; even if one rotor fails, the quadrotor can still maintain stability and continue flying. Besides, it allows for flexible payload configurations, including various sensors, cameras, communication equipment, and others, so that it has adaption for a wide range of tasks, including mapping, surveillance, search and rescue, and environmental monitoring.

- **Electric Propulsion:**

  Different from traditional helicopters that use complicated mechanical "swashplate" with turboshaft engines, quadrotors typically use fixed electric-powered rotors with variables speeds. It offers benefits compared to those driven by internal combustion engines, such as lower noise levels, emission reduction, and better energy efficiency.

With these advantages, quadrotor could be deployed in both indoor and outdoor applications with satisfactory performance. However, there are some inherent challenges in quadrotor control system, which could be summarized as follow:

- The control of quadrotor is commonly recognized as nonlinear, underactuated and strongly coupled model with uncertainty parameters [3]. Nonlinear system is a typical and critical issue for control system. While there have been various applications about traditional linear controller in this field, some remaining problems make advanced nonlinear controller worth attempt to improve system performance.

- Underactuated: The six Degrees of Freedom (DOF) need to be controlled by four input parameters from four rotors, which requires control paradigm to provide constraints with accomplishment of satisfactory maneuverability, stability and other performance.

- Uncertainty parameters: It is an inherent difficulty to represent its dynamic model by precise mathematical equations with all possible variables. This phenomenon make

disturbance, from environment to sensors and actuator, unpredictable. The uncertainty exceeding limited range can cause stabilization problems, even make the quadcopter crash.

- Strongly coupled model: Quadrotor dynamic relations are a complicated systems that involve multiple interconnected components, such as the four rotors, sensors (including accelerometers, gyroscopes), and controller. As these components are strongly coupled in the dynamics, rapid changes in one component can directly affect or even be amplified in another component. This can make stability and robustness challenging.

These features lead to a requirement of efficient controller design, and current various control algorithms have respective advantages and disadvantages. Combination of different controller and simplification of dynamics are able to optimize the performance by complementation. To indicate an efficient method to attain expected performance, this article compares different schemes, consisting of distinct controller and dynamics model, with corresponding simulation results. The control schemes will concentrate on attitude control (Roll, Pitch, Yaw, and Altitude) for simplification. The simulation of environment and sensors is also included to estimate the robustness. The conclusion will generally consider stability, robustness, convergency, and complexity.

This article is arranged as follows. In Section II, a comprehensive literature review of main-stream quadrotor controller will be presented with discussion and selection. Section III will consist of dynamic model, its simplification, and parameter configuration, including selected simulated object and environment. Then, introduction of controller with initial simulation result is placed in Section IV. Section V will have a comprehensive evaluation with comparison. Ultimately, the conclusion and expectation, based on previous results, are arranged in Section VI.

Section II. Literature Review

Generally, categories of controller are linear, nonlinear, and intelligent controller. Traditional control methods were developed for mainly linear systems, but quadrotor dynamic model is intrinsically nonlinear model. This feature makes linear controller unable to satisfy all demand. With other challenges mentioned above, different controller expresses corresponding advantages and disadvantages, which is summarized by category in Table 2.1. The following subsections, aided by Table 2.1 and arranged by solution of above-mentioned challenges, will have detailed description and discussion of the various approaches to controller design with their advantages and disadvantages.

Table. 2.1. Summary of quadrotor controller [3-9].

Category	Approach	Advantage	Disadvantage
Linear	Proportional Integral Derivative (PID)	• Simple algorithm • Independent to model	• Incapability of uncertainty • Long response time
	Linear Quadratic Regulator (LQR)	• Simple algorithm • Efficiency	• Insufficient robustness with observer • Require precise model
	H_∞	• Robustness • Accurate error tracking	• Require precise model • Insufficient performance for nonlinear system
Nonlinear	Feedback linearization (FL) control	• Efficiency • Flexible design	• Incapability of uncertainty • Require full state information • Require precise model
	Backstepping control	• General Well • Handle uncertain disturbances well • Converge fast	• Require full states information • Require precise model • Large magnitude control signals

	Sliding mode control	• Robustness • Fault tolerance • Simple algorithm • Response fast	• Chattering phenomenon • Energy consumption
	Model predictive control	• Predictability • Ability for constraints from inputs • Good performance in error tracking	• High computational power • Weak stability • Require full states information • Require precise model
Intelligent	Fuzzy logic based control	• Simple design • Flexible adaption • Comprehensible result	• Low accuracy in steady state • Lack of guarantee against stability and performance
	Artificial neural network control	• Model free • Adaptability • Fast optimization • Continuous Learning • Function Approximation	• Computational resource • Potential cyber-attack • Difficulty to redesign structure and tune parameter • Lack of guarantee against stability and performance
	Reinforcement learning (RL) based control	• Model free • Fault tolerance • Adaptability • Continuous Learning	• Computational resource demand • Lack of environment perception • Difficulty to redesign structure and tune parameter • Lack of guarantee against stability and performance

2.1 Nonlinear

Although the quadcopter model is nonlinear, it is approximately linear in small attitude operation. This makes linear controller a potentially good choice for the quadcopter if the vehicle will not be maneuvered too aggressively. Meanwhile, since linear controllers are generally easier to design, they can serve as a baseline to which other controllers may be compared. For instance, PID controller, with its simplification and good performance, is still appropriate and popular choice of baseline strategy, when evaluate other controllers whether obtain better performance.

2.1.1 Linear Control

In [10], PID controller is implemented with simplified model for target tracking and attitude control. In some cases, dynamic model will not be fully simplified or linearized to remain partial nonlinear feature for more precise simulations [11, 12]. Besides PID, LQR controller is also implemented for quadrotor model. Elias and Sergio proposed a LQR controller by using Unit Quaternions [13]. Faraz et al. also accomplished the MATLAB simulation of developed LQR based quadrotor control [14]. A comparison of PID and LQR is presented by Shahida et al., which indicates that PID provide better stability while LQR provide better robustness and higher response speed [15].

2.1.2 Nonlinear Control

Synchronously, it is also in progress that attempts of nonlinear controller deployment into quadrotor. As nonlinear controller contains above-mentioned disadvantages, which make design of nonlinear controller generally complicated, the simplification or optimization of its design gradually attracted many specialists. Holger proposed a nonlinear control system, which is available for embedded microcontroller, based on FL and nested structure [16]. Backstepping control implemented its control law by each subsystem of state variables, so that its complexity is large in 6 DOF of quadrotor model. In simulation of such a complicated system [17], a good result is presented, but its complexity is recognized as large constraint.

Sliding mode control is also deployed to address problems but chattering phenomenon is main challenge that is a high frequency oscillation caused by pattern switch. Such oscillation makes stability of sliding mode control contains an inherent problem that requires designer process. In [18], a sliding mode controller is implemented by setting sliding surface as PD controller. A comparison between adaptive sliding mode control and FL control indicates that the previous one provide more robustness, but the chattering phenomenon around middle part of steady trajectory, shown in Fig. 2.1, is unavoidable [19].

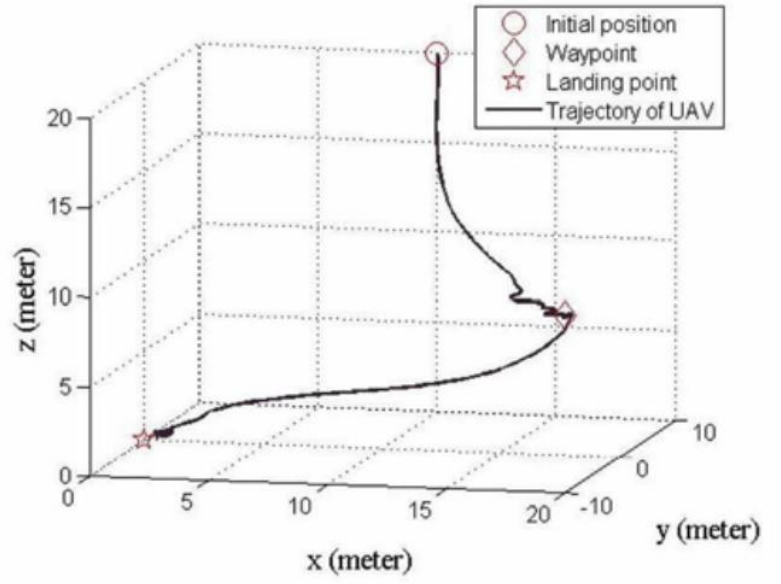

Fig. 2.1. Trajectory of UAV in 3-D axes with the adaptive sliding mode controller with uncertainty and sensor noise [19].

Besides these works, another method is to integrate controller for complement. For instance, Guilherme et al. proposed a structure that model predictive control is used for tracking the reference trajectory while H_∞ controller stabilizes the attitude [20]. The adaptive sliding mode control presented above is also a typical example [19]. Such combination is not limited in classical controller. Chuang et al. proposed adaptive fuzzy control with disturbance observer, whose stability and effectiveness are also proved with Lyapunov theory [21].

2.1.3 Intelligent Control

With dramatic development of intelligent algorithms, some of those are also expected to solve control challenges. The stability certification and initial demonstration is presented in [22]. With further development, Jemin et al. proposed a controller trained by RL to complete a throw-stabilizing task [23]. This tendency makes artificial intelligence more potential and attractive. However, comparing to specific controller, there are two critical issues dampening its competitiveness. On the one hand, controller designed by intelligent algorithms may have good performance in specific scenery, but it can't guarantee its performance, such as stability, will remain when operation scenery is expanded to more complicated environment. This phenomenon makes intelligent controller not appropriate in many situations. On the other hand, while training available controller demands lots of computational resource, the generated controller is not comprehensive for designer, which makes it hard to tune or adjust. If each situation will have corresponding specifically trained controller, abuse of computational resource is also not appropriate.

2.2 Underactuated

The common methods of this issue are two: (a) simplify state variables needed control to make inputs equal to outputs; (b) building mathematical relation between two redundant variables and inputs. The previous method will make a project concentrate on trajectory control (X, Y, Z, and

Yaw) or attitude control (Roll, Pitch, Yaw, and Z). The later one will propose additional requirement for controller. For instance, in a PID control system, an outer-loop will be needed for position control, while there is an inner-loop for attitude control [10].

2.3 Uncertainty Parameters

Although the quadrotor model has been very complicated, there are still many effect elements hard to describe by mathematical equations. Those variables out of model are recognized as uncertainty and leads to a discrepancy between the model outputs and the practical states. It leads to a challenge that control system is probably unable to remain its stability or other performance, which is summarized as 'robustness', when uncertain parameter rapidly changes. To address this problem, two common methods are widely used. One method is to enhance robustness of control system to decrease its effect. According to existing research, this method will require additional part to enhance robustness and adaption, such as adding adaptive component [19], directly building adaptive controller [24], or choosing a robust or optimal control strategy [13, 14, 18, 20, 25]. Another method is using intelligent algorithms to break away from model limits [22, 23, 26, 27], but black box is hard to provide a comprehensible result.

2.4 Strongly Coupled Model

This challenge is related to inherent nonlinear model, so that its solution is also related, which are nonlinear control strategy or simplification of dynamic model. In nonlinear strategy, these coupled terms will be considered, modeled and processed to attain expected performance. For instance, in FL control, the nonlinear term will be eliminated to linearize model. The simplification also eliminates these terms but based on assumptions. In quadrotor case, the assumption is that the speed and angle of rotation are sufficient low, so that could be regarded as zero. The detailed process will be presented in next section.

2.5 Selection & Summary

According to current research, while various proposals have been able to solve major challenges much or less. It is necessary to consider an optimal choice by comparison. Besides, majority of research concentrate on control strategy design rather than effect from specific component. Therefore, to quantify effectiveness of distinct solution, this article will provide a strategy comparison with model of potential influence from environment and sensors. The implementation will be based on simulation in MATLAB/Simulink, because it is convenient to track signals.

With general consideration of stability, robustness, convergency, and model complexity, which are all presented in other research, the control strategies discussed in this article will be PID control, PID-FL control, and LQR control.

Section III. Quadrotor Simulation Model

This section will describe a) the config of coordinate system and dynamic model of a quadrotor; b) simplification of model; c) config of quadrotor parameter, sensor and environment. The diagram of entire simulation is illustrated in Fig. 3.1. The process for underactuated and strongly coupled model will be contained in simplification, which will be used in initial controller design. The parameter config includes simulation of gyroscope and accelerometer, discussion of altitude measurer, noise and resistance from environment.

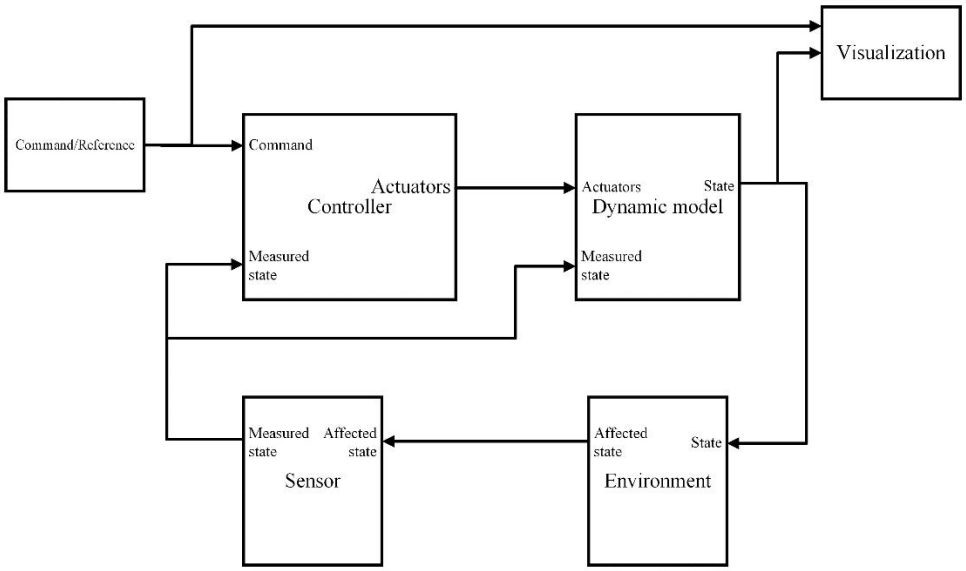

Fig. 3.1. Diagram of simulation model.

3.1 Coordinate System

For clear description of quadrotor motion by Newton's equations, it is necessary to build a relative reference frame fixed in vehicle body and absolute frame in working space, and then make them relevant by translation and rotation.

Assuming that there is an absolute inertial coordinate system F^i, fixed in particular location of space. The definition of unit vector is a) direction of $\hat{i}^i$ is North; b) direction of $\hat{j}^i$ is East; c) direction of $\hat{k}^i$ is towards to earth [10]. To indicate the position of vehicle, the initial frame translates to the mass center of vehicle. So that a vehicle frame F^v is attained, whose axis ($\hat{i}^v$, $\hat{j}^v$, $\hat{k}^v$) are parallel to the axis of initial absolute frame. So that position of quadrotor (*X*, *Y*, *Z*) could be described by translation between F^i and F^v.

In this case, to describe the rotated orientation of vehicle body by Euler angles, the reference frame should be respectively rotated 3 times around x, y, z axis. For instance, vehivle-1 frame, F^{v1}, could be derived by rotating F^v around z axis with yaw angle (ψ). Similarly, rotating F^{v1} by pitch angle (θ) attains vehivle-2 frame (F^{v2}), and rotating F^{v2} by roll angle (ϕ) attains body frame F^b. The entire rotation process could be summarized as Eq. 3.1.

$$R_v^b = R_{v2}^b(\phi) * R_{v1}^{v2}(\theta) * R_v^{v1}(\psi)$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos\phi & sin\phi \\ 0 & -sin\phi & cos\phi \end{pmatrix} \begin{pmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{pmatrix} \begin{pmatrix} cos\psi & sin\psi & 0 \\ -sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ s\phi s\theta c\psi - c\phi s\psi & s\phi s\theta s\psi + c\phi c\psi & s\phi c\theta \\ c\phi s\theta c\psi + s\phi s\psi & c\phi s\theta s\psi - s\phi c\psi & c\phi c\theta \end{pmatrix} \quad \text{Eq. 3.1.}$$

Note: $c\theta = cos\,\theta,\ s\theta = sin\,\theta,\ t\theta = tan\,\theta.$

After above translation and rotation, the relative coordinate system, which fixes in mass center of quadrotor and its direction is illustrated in Fig. 3.2, is attained to describe quadrotor rotation (ϕ, θ, ψ), velocity along axis (u, v, w) and angular velocity (p, q, r).

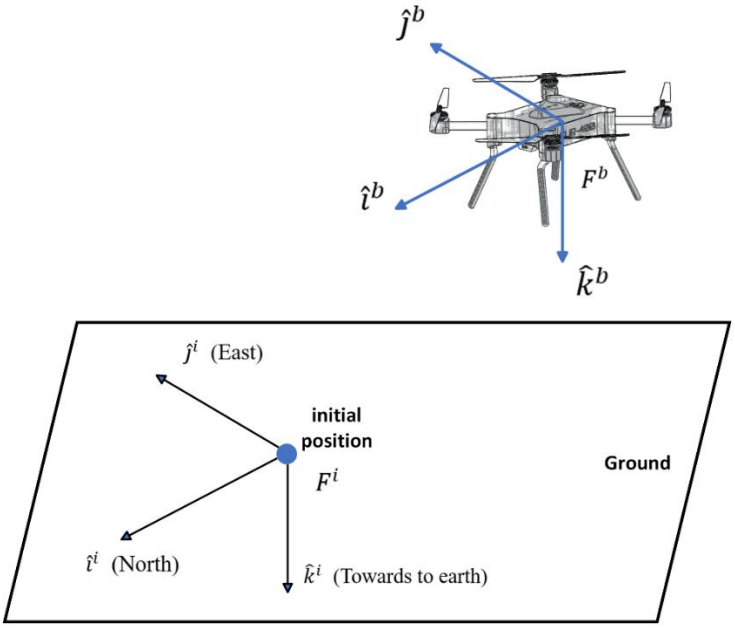

Fig. 3.2. Illustration of relative coordinate system and absolute coordinate system.

By consideration of a) the control system will require first or second-degree dynamic information, b) the coupling between state space and rotor inputs will require acceleration, equation of Coriolis, whose derivation refers to [28], is also involved for expression of acceleration and angular acceleration. The equation, shown in Eq. 3.2, indicates the relation of time derivation when there is a vector p moving with respect to F^b and F^b is also moving with respect to F^i. So that acceleration and angular acceleration could be expressed by derivation of state space function.

$$\frac{d}{dt_i}p = \frac{d}{dt_b}p + w_{b/i} * p \quad \text{Eq. 3.2. [10]}$$

Note:

$\frac{d}{dt_i}p$: the time derivation of vector p based on F^i;

$\frac{d}{dt_b}p$: the time derivation of vector p based on F^b;

$w_{b/i}$: the angular velocity of F^b referring to F^i.

3.2 Dynamic Model

Based on above coordinate relations, the variables in state space could be summarized, as shown in Table 3.1. The acceleration will be expressed by the time derivation of corresponding velocity variables. Note that, for intuitive presentation, Z will be expressed as *-h* in following equations, because positive direction of Z axis towards ground, but height is in opposite direction. The following paragraphs will present the relation of these variables, in which time derivation of variable will be expressed as $\dot{a}$.

Table 3.1. Variables in state space.

Variable Category	Variable	Direction	Reference Frame
Position	X	Along $\hat{\imath}^i$ (North)	F^i
	Y	Along $\hat{\jmath}^i$ (East)	F^i
	Z	Along $\hat{k}^i$ (Towards to earth)	F^i
Velocity	u	Along $\hat{\imath}^b$	F^b
	v	Along $\hat{\jmath}^b$	F^b
	w	Along $\hat{k}^b$	F^b
Angle	ϕ	Around $\hat{\imath}^{v2}$	F^{v2}
	θ	Around $\hat{\jmath}^{v1}$	F^{v1}
	ψ	Around $\hat{k}^v$	F^v
Angular Velocity	p	Along $\hat{\imath}^b$	F^b
	q	Along $\hat{\jmath}^b$	F^b
	r	Along $\hat{k}^b$	F^b

3.2.1 Position to Velocity

The position variables are defined in F^i, while velocity variables are defined in F^b. To build relation between $\dot{X}$, $\dot{Y}$, $\dot{Z}$ ($-\dot{h}$) and u, v, w, the transformation between reference frames is required. Based on previously involved theory, the relation could be summarized as Eq. 3.3.

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \\ -\dot{h} \end{pmatrix} = \begin{pmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \qquad \text{Eq. 3.3.}$$

3.2.2 Angle to Angular Velocity

Although angle variables are defined in three different reference frames, the relation between angle and angular velocity could be derived by similar method. From instantaneous view, $\dot{\phi}$ could be regarded as small value of ϕ and this conclusion is also suitable for other angle variables, so that Eq. 3.4. could be derived by assuming it is a sufficient small value. Therefore, the relation between angle derivation and angular velocity could be summarized as Eq. 3.5.

$$R_{v2}^{b}(\dot{\phi}) = R_{v1}^{v2}(\dot{\theta}) = R_{v}^{v1}(\dot{\psi}) = \begin{pmatrix} 1 & & \\ & 1 & \\ & & 1 \end{pmatrix} \qquad \text{Eq. 3.4.}$$

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \qquad \text{Eq. 3.5.}$$

3.2.3 Acceleration

To express linear acceleration along axis, Newton's law is applied here combining equation of Coriolis.

$$\boldsymbol{F} = \frac{d\boldsymbol{p}}{dt_i} = m\frac{d\boldsymbol{v}}{dt_i} = m\left(\frac{d}{dt_b}\boldsymbol{v} + w_{b/i} * \boldsymbol{v}\right) \qquad \text{Eq. 3.6}$$

where, in this case, m is total mass of body, $\boldsymbol{p}$ is quantity of motion, $\boldsymbol{v} = (u, v, w)^T$, $w_{b/i} = (p, q, r)^T$, $\boldsymbol{F} = (f_x, f_y, f_z)^T$. So that the above equation could be rewritten as Eq. 3.7.

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} rv - qw \\ pw - ru \\ qu - pv \end{pmatrix} + \frac{1}{m}\begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} \qquad \text{Eq. 3.7.}$$

It worth a recall that quadrotor is an underactuated model that 6 DOF should be controlled by four inputs from four rotors. Therefore, the relation from force and torque to inputs ($U = (U_1, U_2, U_3, U_4)$) is required. Begin from forces and torques provided by each rotor, the definition of those is illustrated in Fig. 3.3.

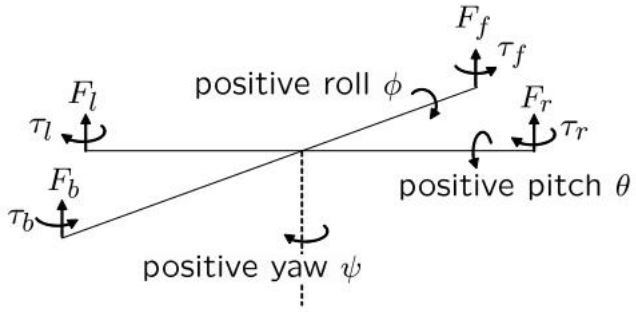

Fig. 3.3. Definition of forces and torques provided by each rotors [10].

Developed from definition, the relation between provided forces and torques and inputs could be summarized as Eq. 3.8-3.11. Note that the model from rotor speed to provided force and torque is neglected for simplification.

$$U_1 = F = F_l + F_f + F_r + F_b \quad \text{Eq. 3.8.}$$

$$U_2 = \tau_\phi = l(F_l - F_r) \quad \text{Eq. 3.9.}$$

$$U_3 = \tau_\theta = l(F_f - F_b) \quad \text{Eq. 3.10.}$$

$$U_4 = \tau_\phi = \tau_r + \tau_l - \tau_f - \tau_b \quad \text{Eq. 3.11.}$$

For intuitive expression, the control input is set as $U = (F, \tau_\phi, \tau_\theta, \tau_\psi)^T$. To express gravity effect in state space function, which is intrinsically presented in F^i, should be also transformed into F^b. The transformation is similar to abovementioned method and the result is presented by Eq. 3.12.

$$G^b = R_v^b \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} = \begin{pmatrix} -mgs\theta \\ mgs\phi c\theta \\ mgc\phi c\theta \end{pmatrix} \quad \text{Eq. 3.12.}$$

With above process, Eq. 3.7 could be derived as Eq. 3.13.

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} rv - qw \\ pw - ru \\ qu - pv \end{pmatrix} + \begin{pmatrix} -gs\theta \\ gs\phi c\theta \\ gc\phi c\theta \end{pmatrix} + \frac{1}{m}\begin{pmatrix} 0 \\ 0 \\ -F \end{pmatrix} \quad \text{Eq. 3.13.}$$

3.2.4 Angular Acceleration

For angular acceleration around axis, Newton's law combining equation of Coriolis is also available. But $\boldsymbol{p}$ should be replaced by angular momentum $\boldsymbol{p_\theta}$, and $\boldsymbol{F}$ should be replaced by torque $\boldsymbol{T}$. This replacement could be written as Eq. 3.14.

$$\boldsymbol{T} = \frac{d\boldsymbol{p_\theta}}{dt_i} = \frac{d}{dt_b}\boldsymbol{p_\theta} + w_{b/i} * \boldsymbol{p_\theta} \quad \text{Eq. 3.14.}$$

Where $\boldsymbol{T} = (\tau_\phi, \tau_\theta, \tau_\psi)^T$, $\boldsymbol{p_\theta} = \boldsymbol{J} * w_{b/i}$, $\boldsymbol{J}$ is the constant inertia matrix of quadrotor, whose configuration is listed in next subsection. Because of quadrotor symmetric structure, there will only have J_x, J_y, and J_z.

Therefore, torque variables could be expressed in F^b to calculate angular acceleration, as shown in Eq. 3.15.

$$\begin{pmatrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{pmatrix} = \begin{pmatrix}\frac{J_y - J_z}{J_x} qr\\ \frac{J_z - J_x}{J_y} pr\\ \frac{J_x - J_y}{J_z} pq\end{pmatrix} + \begin{pmatrix}\frac{1}{J_x}\tau_\phi\\ \frac{1}{J_y}\tau_\theta\\ \frac{1}{J_z}\tau_\psi\end{pmatrix} \qquad \text{Eq. 3.15.}$$

3.2.5 Quadrotor Nonlinear Model

In summary, the nonlinear model consists of Eq. 3.3, 3.5, 3.13 and 3.15. To clearly indicate their relations, these equations are listed below as state space function.

$$\begin{pmatrix}\dot{X}\\ \dot{Y}\\ -\dot{h}\end{pmatrix} = \begin{pmatrix}c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi\\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi\\ -s\theta & s\phi c\theta & c\phi c\theta\end{pmatrix}\begin{pmatrix}u\\ v\\ w\end{pmatrix} \qquad \text{Eq. 3.16.}$$

$$\begin{pmatrix}\dot{u}\\ \dot{v}\\ \dot{w}\end{pmatrix} = \begin{pmatrix}rv - qw\\ pw - ru\\ qu - pv\end{pmatrix} + \begin{pmatrix}-gs\theta\\ gs\phi c\theta\\ gc\phi c\theta\end{pmatrix} + \frac{1}{m}\begin{pmatrix}0\\ 0\\ -F\end{pmatrix} \qquad \text{Eq. 3.17.}$$

$$\begin{pmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{pmatrix} = \begin{pmatrix}1 & s\phi t\theta & c\phi t\theta\\ 0 & c\phi & -s\phi\\ 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta}\end{pmatrix}\begin{pmatrix}p\\ q\\ r\end{pmatrix} \qquad \text{Eq. 3.18.}$$

$$\begin{pmatrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{pmatrix} = \begin{pmatrix}\frac{J_y - J_z}{J_x} qr\\ \frac{J_z - J_x}{J_y} pr\\ \frac{J_x - J_y}{J_z} pq\end{pmatrix} + \begin{pmatrix}\frac{1}{J_x}\tau_\phi\\ \frac{1}{J_y}\tau_\theta\\ \frac{1}{J_z}\tau_\psi\end{pmatrix} \qquad \text{Eq. 3.19.}$$

3.3 Simplification of Model

The above nonlinear model is not appropriate for controller design, because it is a complicated and strongly coupled model. Therefore, it is sensible to simplify the nonlinear model into a linear model that will be used to design, verify, correct, and optimize initial controller, which will be introduced in next section.

The simplification is proposed based on two assumptions:

a) Roll and pitch are small in operation.

b) p, q, r are sufficient small to make Coriolis terms pq, qr, and pr are close to 0.

Developing from these two assumptions, Eq. 3.16 and 3.17 could also be simplified to Eq. 3.20 and 3.21 respectively.

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \\ -\dot{h} \end{pmatrix} = \begin{pmatrix} c\psi & 0 & 0 \\ 0 & c\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad \text{Eq. 3.20.}$$

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} + \frac{1}{m} \begin{pmatrix} 0 \\ 0 \\ -F \end{pmatrix} \quad \text{Eq. 3.21.}$$

Because this project concentrates on attitude and altitude control, the simplified model could neglect problem in X and Y. Therefore, the simplified model could be summarized as Eq. 3.22-3.25.

$$-\ddot{h} = g - \frac{F}{m} \quad \text{Eq. 3.22.}$$

$$\ddot{\phi} = \frac{1}{J_x}\tau_\phi \quad \text{Eq. 3.23.}$$

$$\ddot{\theta} = \frac{1}{J_y}\tau_\theta \quad \text{Eq. 3.24.}$$

$$\ddot{\psi} = \frac{1}{J_z}\tau_\psi \quad \text{Eq. 3.25.}$$

In addition, to enhance performance of controller, in ultimate evaluation, the nonlinear model (Eq. 3.16-3.19) will not be directly deployed. Because its effect is possible to disable controller in unexpected situation, which will be discussed later. For instance, when rising or holding height, changing roll angle will influence its ability. To accomplish proper result for comparison, a partial nonlinear model is involved. The process refers to [29] and the model is place below, whose implementation is based on expressing variable on F^{v1}. Such process could not only simplify nonlinear model, while it concentrates on relative position to specific target, but also remain nonlinear feature to evaluate its effect.

$$\ddot{X} = \frac{c\phi s\theta c\psi + s\phi s\psi}{m} F \quad \text{Eq. 3.26.}$$

$$\ddot{Y} = \frac{c\phi s\theta c\psi - s\phi s\psi}{m} F \quad \text{Eq. 3.27.}$$

$$-\ddot{h} = g - \frac{c\phi c\theta}{m} F \quad \text{Eq. 3.28.}$$

$$\ddot{\phi} = \frac{J_y - J_z}{J_x} qr + \frac{1}{J_x}\tau_\phi \quad \text{Eq. 3.29.}$$

$$\ddot{\theta} = \frac{J_z - J_x}{J_y} pr + \frac{1}{J_y}\tau_\theta \quad \text{Eq. 3.30.}$$

$$\ddot{\psi} = \frac{J_x - J_y}{J_z} pq + \frac{1}{J_z}\tau_\psi \qquad \text{Eq. 3.31.}$$

3.4 Parameter Configuration

3.4.1 Simulation Object

KUR 400 is chosen as simulation object, which is the most advanced and compact drone in 4kg range [30]. The illustration of it is placed in Fig. 3.4. The reason of choosing advanced product is that it could avoid some intrinsic challenges from body or components. The advanced product has more optimization of problems and challenges, so that this project could concentrate on comparison of controller without influence from unexpected aspects. The parameter configuration of Kur-400 in this project is listed in Table. 3.2.

Fig. 3.4. Illustration of KUR-400 [30].

Table 3.2. parameter configuration of Kur-400 [2].

Property	Quantity	Unit	Description
Central Mass	2.2	kg	= Battery Mass (1.2) + partial Frame Mass (1)
Rotor Mass	0.2	kg	partial Frame Mass (0.2 *4) Frame Mass = 1.8 kg
Total Mass	3	kg	= Central Mass + 4* Rotor Mass = Frame Mass + Battery Mass
Radius of Mass Center	0.1	m	Assumption of spherical solid center

Arm Length	0.3	m	Calculating by Maximum dimensions of body
J_x	0.0448	$kg * m^2$	Inertial moment in X axis
J_y	0.0448	$kg * m^2$	Inertial moment in Y axis
J_z	0.0808	$kg * m^2$	Inertial moment in Z axis

3.4.2 Gyroscope and Accelerometer

To involve effect from sensors, the simulation models of gyroscope and accelerometer are built up. The effect from sensors could be summarized as Eq. 3.32-3.37.

$$y_{gyro,x} = k_{gyro,x}p + \beta_{gyro,x} + \eta_{gyro,x} \quad \text{Eq. 3.32.}$$

$$y_{gyro,y} = k_{gyro,y}q + \beta_{gyro,y} + \eta_{gyro,y} \quad \text{Eq. 3.33.}$$

$$y_{gyro,z} = k_{gyro,z}r + \beta_{gyro,z} + \eta_{gyro,z} \quad \text{Eq. 3.34.}$$

$$y_{acc,x} = k_{acc,x}\dot{u} + \beta_{acc,x} + \eta_{acc,x} \quad \text{Eq. 3.35.}$$

$$y_{acc,y} = k_{acc,y}\dot{v} + \beta_{acc,y} + \eta_{acc,y} \quad \text{Eq. 3.36.}$$

$$y_{acc,z} = k_{acc,z}\dot{w} + \beta_{acc,z} + \eta_{acc,z} \quad \text{Eq. 3.37.}$$

Where k_{gyro} or k_{acc} is a gain that refers to sensitivity in datasheet. β_{gyro} and β_{acc} refer to bias term that caused by temperature and age. η_{gyro} and η_{acc} refer to white noise existing in environment.

According to [2], the Flight Control Unit (FCU) is set as Pixhawk 2.1 The Cube Orange. Based on information provided in [31], ICM-20948 is suitable for both Orange and Orange+ type, which presents that ICM-2094 probably have universality and flexibility. Therefore, the simulation of sensors is built based on data of ICM-20948. The parameter config of it is summarized in Table. 3.3.

Table 3.3. parameter configuration of ICM-20948 [32].

Property	Quantity	Description
k_{gyro}	65.5 LSB/dps (Degree per Second)	GYRO_FS_SEL=1
β_{gyro}	(Random number) *10-5 + temperature difference *0.05	Initial bias = ± 5 dps Bias variation = ± 0.05 dps/℃
η_{gyro}	0.015 dps/√Hz	Noise Spectral Density based on 10 Hz bandwidth
k_{acc}	8192 LSB/g	ACCEL_FS=1
β_{acc}	(Random number) *0.05-0.025 + temperature difference *0.8	Initial bias = ± 50 mg Bias variation = ± 0.8 mg/℃
η_{acc}	230 μg/√Hz	Noise Spectral Density based on 10 Hz bandwidth

3.4.3 Altitude Measurement

While the ground measurement is not available for now [2], it is appropriate to consider a vision-based altitude measurement, so that a specific Lidar or Ultrasonic sensor could be dropped by chance. In this case, those complicated algorithms for height measurement will not be considered for simplification, but an easy implementation will be discussed.

The implementation of that requires relation between number of pixels and height, and this process refers to [10]. As shown in Fig. 3.5, the size of measured target will be transformed into number of pixels in output picture, so that the ratio of that is equal to ratio between height and focal length.

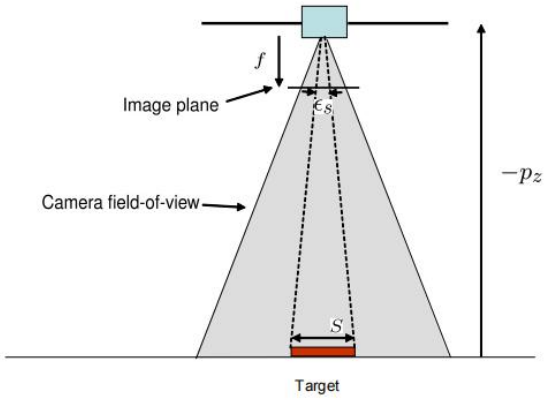

Fig. 3.5. Illustration of vision-based altitude measurement [10].

This relation could be written as Eq. 3.38. Then, to implement its control, the derivation of s is required, which could be attained by Eq. 3.39. Assuming a PID controller is implemented in this case, so that second-degree derivation is also needed, which is derived in Eq. 3.40.

$$\frac{-Z}{S} = \frac{f}{\epsilon_s} \quad \text{Eq. 3.38.}$$

$$\dot{Z} = \frac{fS\dot{\epsilon_s}}{\epsilon_s^2} \quad \text{Eq. 3.39.}$$

$$\ddot{Z} = \frac{fS\ddot{\epsilon_s}}{\epsilon_s^2} - 2fs\frac{\dot{\epsilon}_s^2}{\epsilon_s^3} \quad \text{Eq. 3.40.}$$

By substituting $u_z = \ddot{Z}$ in Eq. 3.40, Eq. 3.41 is attained.

$$\ddot{\epsilon_s} = \frac{\epsilon_s^2}{fS}u_z + 2\frac{\dot{\epsilon}_s^2}{\epsilon_s} \quad \text{Eq. 3.41.}$$

To implement PID controller, defining $\ddot{\epsilon_s} = u_s$, then the controller could drive ϵ_s to ϵ_s^d that could be calculated by command height and Eq. 3.38. Based on previous definition and derivation, the controller is derived as Eq. 3. 42.

$$u_z = \frac{fS}{\epsilon_s^2}\, k_{ps}(\epsilon_s^d - \epsilon_s) - \frac{fS}{\epsilon_s^2}\, k_{ds}\dot{\epsilon_s} - 2fs\frac{\dot{\epsilon}_s^2}{\epsilon_s^3} + \frac{fS}{\epsilon_s^2}\, k_{is}\int_0^t (\epsilon_s^d - \epsilon_s)d\tau \quad \text{Eq. 3.42.}$$

The problem of is that the inverse relation between pixel number and height is possible to disenable the controller. When the height is quite small, which is also the initial state, the proportion term (P-term) can commonly work, the integration term (I-term) hasn't began working, and the derivation term (D-term) will provide an absolute high value that close to infinite large. In this case, D-term overwhelms P and I term, which causes controller failure.

Considering that the height measurement and its controller will start working when it passes a threshold. There will be a working available range. However, when the distance between target and quadrotor body, which means the height, is sufficient large, the controller failure will happen again. Because the error between command pixel number and current pixel number will be too small to disenable P-term, and D-term will be close to 0 according to inverse relation. In this case, only I-term keeps working, which will take too long to response.

The above process is summarized in Table. 3.4. Therefore, as other complicated algorithms are out of consideration that could provide available measurement, the height measurement is assumed as Lidar sensor based. As the specific sensor type is not available for now, it is set as an ideal sensor that could deliver signal directly.

Table 3.4. Situation of PID controller in vision-based height measurement.

Height/Component	P-term	I-term	D-term	Controller
Low	Work	→0	→∞	Disable
Median	Work	Work	Work	Able
High	→0	Work	→0	Disable

3.4.4 Environment Configuration

Typically, in working environment config of quadrotor, it will contain gravity, temperature, sound speed, air pressure and density, magnetic model, and other component depends on designer, such as resistance. In this case, the used components are temperature, gravity, and resistance caused by wind. Wind resistance is added to evaluate robustness of controller. The environmental parameter is listed in Table. 3.5.

Table 3.5. Environmental parameter Configuration.

Property	Quantity	Description
Gravity (g)	9.8 m/s^2	Although precise gravity depends on location, such effect is neglected here.
Temperature	30 ℃	Set as a common working temperature to activate bias term in sensor models.
Wind_x Wind_y Wind_z Wind_ϕ Wind_θ Wind_ψ	5 m/s	Set as a step function to simulate uncertain disturbance. Begin at t=10 s.

Section IV. Controller Design

The controller design will be implemented based on simplified linear model (Eq. 3.22-3.25). This section will introduce implementation of three controllers that are PID, PID with FL, and LQR. The optimized parameter configuration of controller will be also introduced with the manual optimization process.

4.1 PID Control

The process of PID control could be summarized as below.

- P-term:

  It produces a control output that is directly proportional to the error between the desired value and the current system state. The larger the error, the stronger the correction applied to the control output.

- I-term:

  It adjusts the control output based on the accumulated past errors to eliminate any residual steady-state error. It helps to address issues such as system bias or disturbances that cause a persistent error over time.

- D-term:

  It anticipates future error by calculating the slope of the error signal. It helps to reduce overshoot and oscillation by dampening the system's response.

In quadrotor control system, setting roll control as an instance, the control output is τ_ϕ.The roll controller could be derived as Eq. 4.1, in which ϕ^d is implemented by a step function from 0 to 0.68 rad (38.98 degree) with high changing speed. The entire control loop, consists of controller and linear model, is illustrated in Fig. 4.1.

$$\tau_\phi = k_{p\phi}(\phi^d - \phi) - k_{d\phi}p + k_{i\phi}\int_0^t (\phi^d - \phi)d\tau \quad \text{Eq. 4.1.}$$

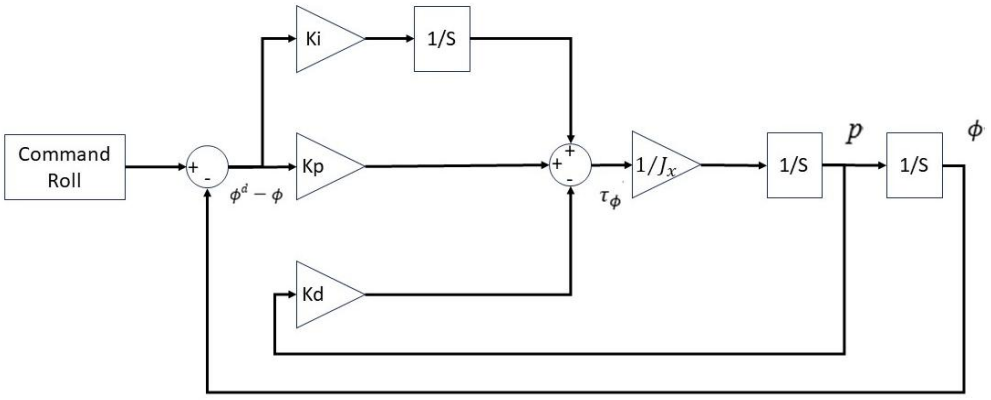

Fig. 4.1. Block diagram of roll control loop.

By similar method, pitch, yaw, and altitude controller could be attained, which is placed in Eq. 4.2, 4.3, and 4.4, respectively. Note that a) P and D term in yaw control have a minus because of $E_\psi = \psi - \psi^d$; b) $E_z = Z - Z^d$ for opposite direction between Z and height.

$$\tau_\theta = k_{p\theta}(\theta^d - \theta) - k_{d\theta}q + k_{i\theta}\int_0^t (\theta^d - \theta)d\tau \quad \text{Eq. 4.2.}$$

$$\tau_\psi = -k_{p\psi}(\psi - \psi^d) - k_{d\psi}r - k_{i\psi}\int_0^t (\psi - \psi^d)d\tau \quad \text{Eq. 4.3.}$$

$$F = k_{pz}(Z - Z^d) - k_{dz}w + k_{iz}\int_0^t (Z - Z^d)d\tau \quad \text{Eq. 4.4.}$$

To begin with (Kp_roll, Ki_roll, Kd_roll) = (1,1,1), the simulation output based on linear dynamic model is illustrated in Fig. 4.2.

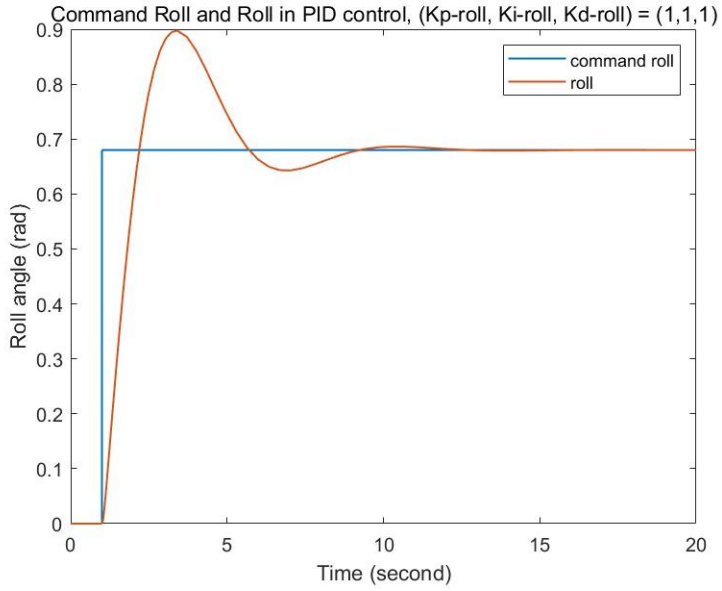

Fig. 4.2. Command Roll and Roll in PID control when (Kp_roll, Ki_roll, Kd_roll) = (1,1,1).

Typically, the first optimizing object is rising time, which refers to P-term. After testing, the rising time is optimized to 0.43s with Kp_roll = 80. The simulation output is placed in Fig. 4.3, and it is apparent that there is huge oscillation before state becomes stable, which require tuning D-term.

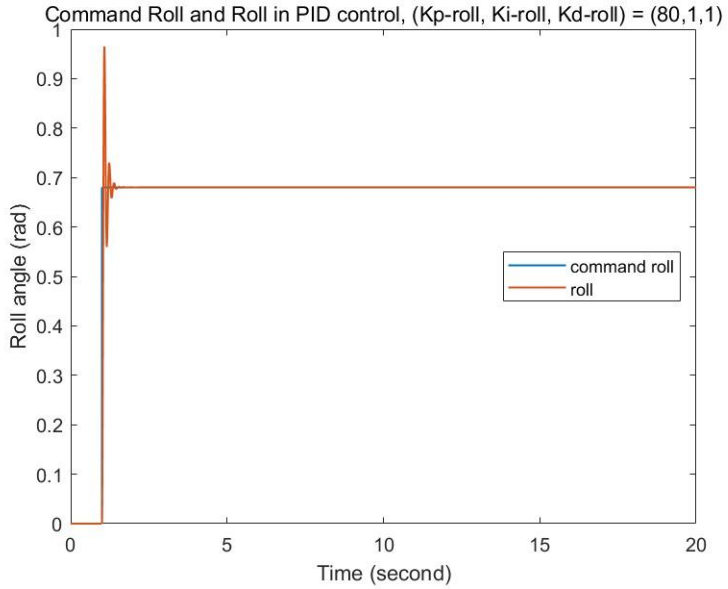

Fig. 4.3. Command Roll and Roll in PID control when (Kp_roll, Ki_roll, Kd_roll) = (80,1,1).

By setting Kd_roll = 15, an intuitively ideal result is attained, but notice that the roll angle keeps larger than 0.681, which means a steady error exists. The Fig. 4.4 illustrates optimized result, in which parameter config is set as (Kp_roll, Ki_roll, Kd_roll) = (80,1,15).

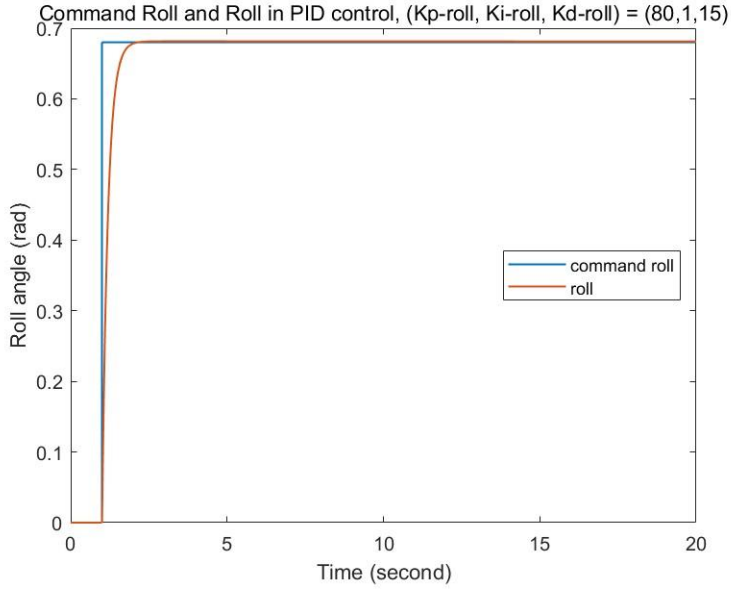

Fig. 4.4. Command Roll and Roll in PID control when (Kp_roll, Ki_roll, Kd_roll) = (80,1,15).

As the model for roll, pitch and yaw is similar, this config is used repeatedly. However, when it is used in altitude control, a steady error exists as shown in Fig. 4.5, which means its I-term should be tuned.

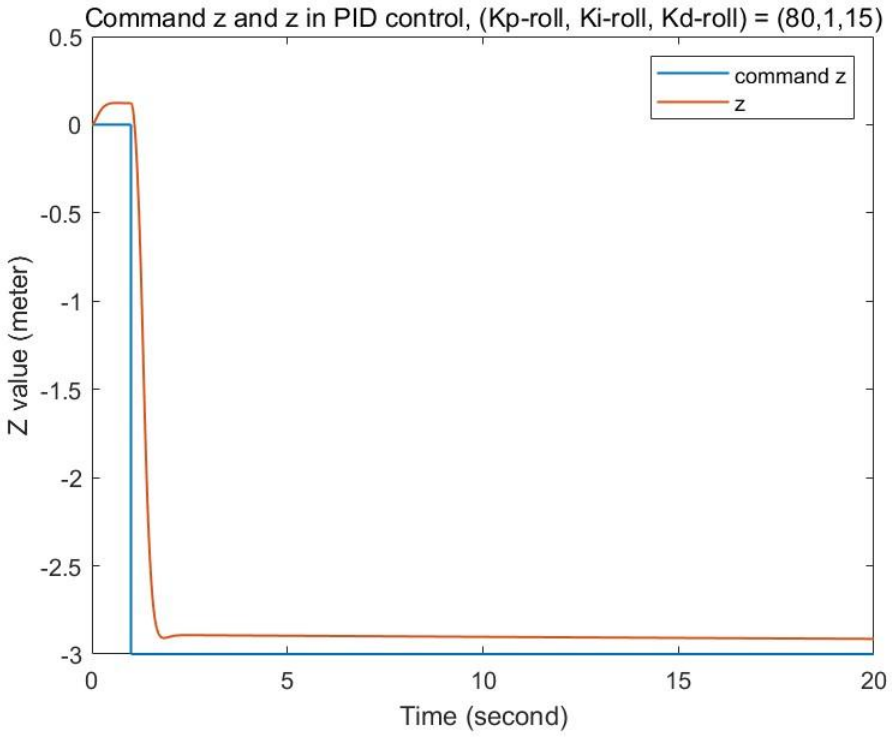

Fig. 4.5. Command Z and Z in PID control when (Kp_roll, Ki_roll, Kd_roll) = (80,1,15).

After optimization, with (Kp_roll, Ki_roll, Kd_roll) = (80,10,15), expected output is attained as shown in Fig. 4.6. The ultimately optimized parameter configurations are summarized as Table. 4.1.

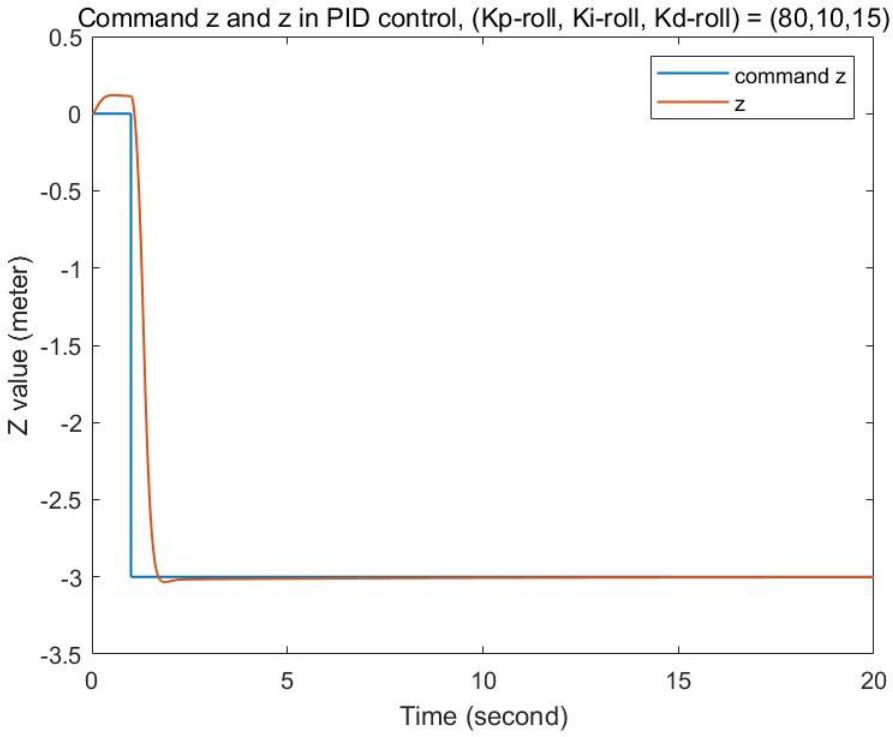

Fig. 4.6. Command Z and Z in PID control when (Kp_roll, Ki_roll, Kd_roll) = (80,10,15).

Table 4.1. PID parameter configuration.

Parameter	Value
Kp_roll	80
Ki_roll	1
Kd_roll	15
Kp_pitch	80
Ki_pitch	1
Kd_pitch	15
Kp_yaw	80
Ki_yaw	1
Kd_yaw	15
Kp_z	80
Ki_z	10
Kd_z	15

4.2 PID with FL

For better solution of nonlinear problem, FL is implemented. It accomplishes a linear model by eliminate nonlinear terms. Assuming that the output of PID controller will not be directly used but recognized as virtual signal. For instance, in attitude control loop, setting $(\tau_\phi, \tau_\theta, \tau_\psi, F)^T$ as virtual signals $(v_1, v_2, v_3, v_4)^T$ and processed outputs as $(u_1, u_2, u_3, u_4)^T$. The diagram of this relation is illustrated in Fig. 4.7.

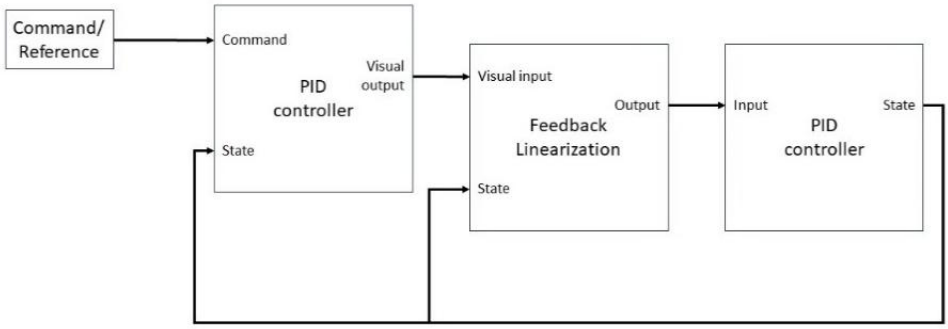

Fig. 4.7. Block diagram of PID control with FL.

Recall that the original nonlinear model is shown as Eq. 3.18 and 3.19. To attain similar performance when deploy linear model (Eq. 3.22-3.25), the virtual signals could be processed as shown in Eq. 4.5-4.8, which subtract Coriolis terms and effect from attitude to altitude. Note that the process for altitude is based on partial nonlinear model (Eq. 3.26-3.31).

$$u_1 = J_x\left(-\frac{J_y - J_z}{J_x} qr + v_1\right) \quad \text{Eq. 4.5.}$$

$$u_2 = J_y\left(-\frac{J_z - J_x}{J_y} pr + v_2\right) \quad \text{Eq. 4.6.}$$

$$u_3 = J_z\left(-\frac{J_x - J_y}{J_z} pq + v_3\right) \quad \text{Eq. 4.7.}$$

$$u_4 = \frac{m}{c\phi c\theta}(g + v_4) \quad \text{Eq. 4.8.}$$

Then, the inputs of dynamic model will be $(u_1, u_2, u_3, u_4)^T$, and nonlinear effect will be reduced. The config of PID controller is same to previous one. The simulation result is illustrated in Fig. 4.8 and 4.9.

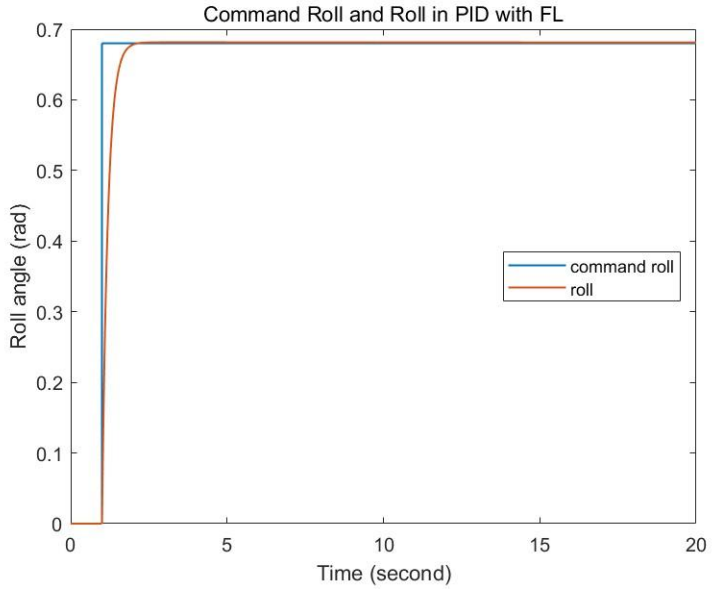

Fig. 4.8. Command Roll and Roll in PID with FL.

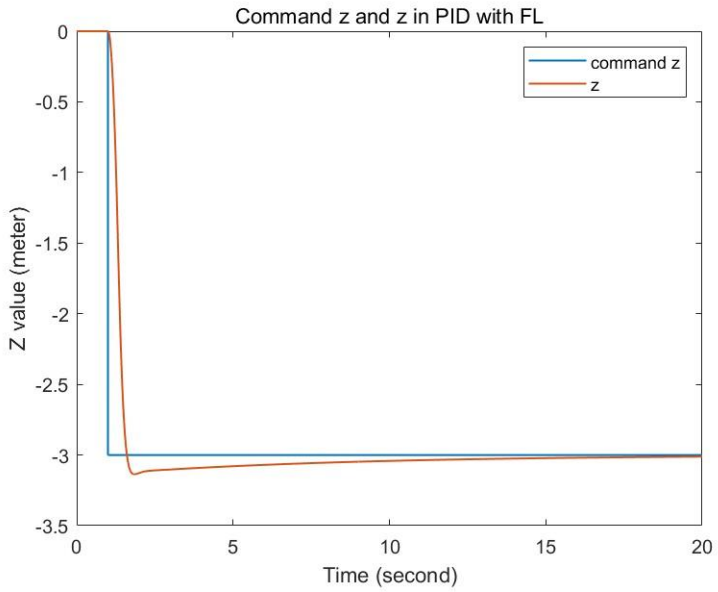

Fig. 4.9. Command Z and Z in PID with FL.

4.3 LQR Control

As an optimal control method, LQR control could attain command value and expected stability by minimizing the cost function that is presented in Eq. 4.9. The implementation of LQR involves '[K,S,P] = lqr(A,B,Q,R)' function provided by Matlab [33]. The solution will be formed within vector K.

$$J = \int_0^\infty (x_v^T Q x_v + u^T R u)\, dt \qquad \text{Eq. 4.9.}$$

Where Q (size n*n, n is number of variables in state space) and R (size m*m, m is number of inputs) are configured by user that depend on demand.

A and B here refer to two matrixes in state space function. Generally, the state space could be summarized as Eq. 4.10 and 4.11.

$$\dot{x_v} = Ax_v + Bu \qquad \text{Eq. 4.10.}$$

$$y_v = Cx_v + Du \qquad \text{Eq. 4.11.}$$

Where x_v, u, and y_v are state variables, control inputs, and outputs respectively. A, B, C, and D are matrixes, and they depend on the system design. By config that $u = -Kx_v + r$, in which r is command signal, the control loop is accomplished.

Based on above description, the diagram of LQR implementation could be attained, as shown in Fig. 4.10. For further development of system performance, the error between command and state is also included into state variables. Implementation of this improvement requires integral feedback, as shown in Fig. 4.11. In this system, the state space functions are modified from Eq. 4.12 and 4.13 to 4.13 and 4.14.

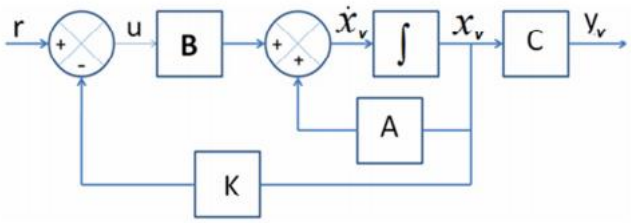

Fig. 4.10. Block diagram of LQR control [14].

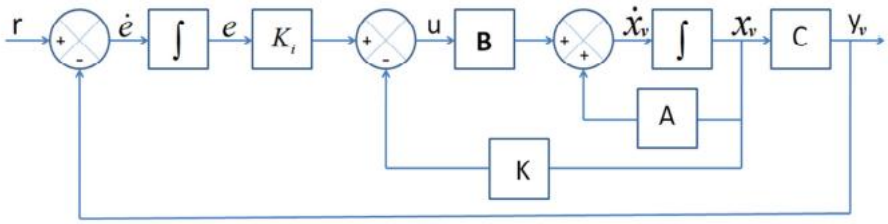

Fig. 4.11. Block diagram of LQR control with integral feedback [14].

$$\begin{bmatrix} \dot{x_v} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x_v \\ e \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ I \end{bmatrix} r \qquad \text{Eq. 4.12 [14].}$$

$$y_v = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x_v \\ e \end{bmatrix}$$ Eq. 4.13 [14].

Where $u = -Kx_v + K_i e$, $x_v = (\phi, p, e_\phi, \theta, q, e_\theta, \psi, r, e_\psi, Z, w, e_Z)^T$, $y_v = (\phi, \theta, \psi, Z)^T$, $\widehat{A} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix}$, $\widehat{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}$. Configurations of matrixes ($\widehat{A}$, $\widehat{B}$, C, D, Q, R) are presented below.

$$\widehat{A} = \begin{bmatrix} a & & & \\ & a & & \\ & & a & \\ & & & a \end{bmatrix}$$ Eq. 4.14.

$$\widehat{B} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$ Eq. 4.15.

$$C = \begin{bmatrix} c & & & \\ & c & & \\ & & c & \\ & & & c \end{bmatrix}$$ Eq. 4.16.

$$D = \begin{bmatrix} 0 \end{bmatrix}$$ Eq. 4.17.

$$Q = \begin{bmatrix} w & & & \\ & w & & \\ & & w & \\ & & & w \end{bmatrix}$$ Eq. 4.18.

$$R = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$ Eq. 4.19.

Where $a = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}$, $b_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1/J_x & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$, $b_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1/J_y & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$, $b_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1/J_z & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$, $b_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/M \\ 0 & 0 & 0 & 0 \end{bmatrix}$, $c = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$, $w = \begin{bmatrix} 5 & & \\ & 30 & \\ & & 90 \end{bmatrix}$.

The corresponding simulation results of LQR control with integral feedback are illustrated in Fig. 4.12 and 4.13.

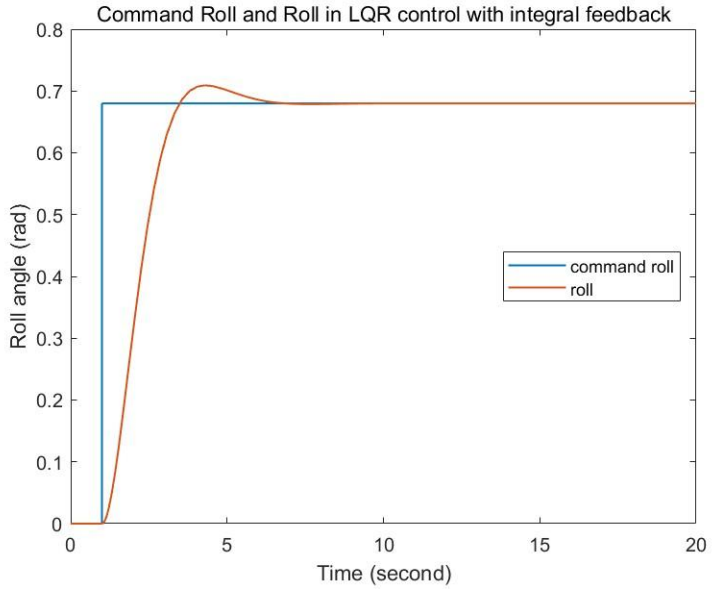

Fig. 4.12. Command Roll and Roll in LQR control with integral feedback.

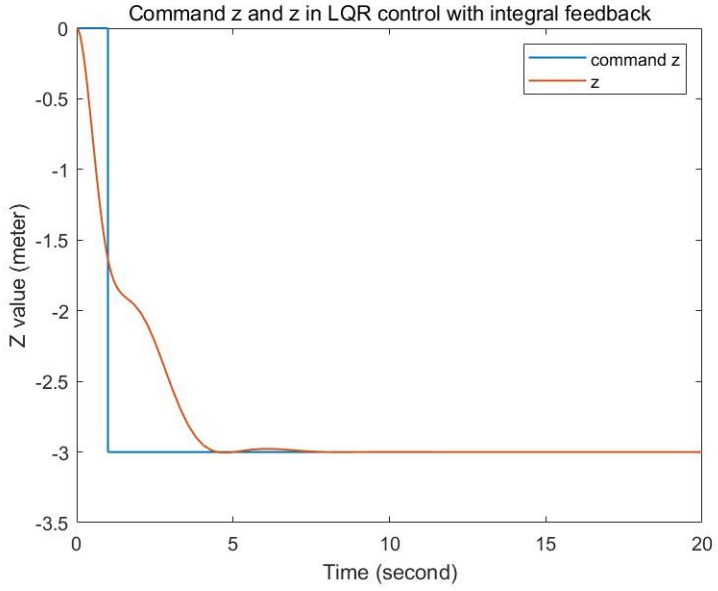

Fig. 4.13. Command Z and Z in LQR control with integral feedback.

4.4 Initial Comparison

Because there are lots of work implemented by linear model, it is worth to make initial comparison and conclusion. It could be also a good reference for difference between linear and nonlinear model, which will emphasize nonlinear effect.

The extracted data is summarized in Table. 4.2, in which 'convergence time' refers to the time of obvious oscillation between the time points when signal reaches target value for the first time and when signal is stable around target value within specific error range ($\pm$ 0.002 rad and $\pm$ 0.02 meter).

Table 4.2. Comparison of three controller in linear model.

	Roll			Z		
Controller	Rising Time (s)	Convergence Time (s)	Peak Value (rad)	Rising Time (s)	Convergence Time (s)	Peak Value (m)
PID	1.1342	0	0.6816	0.7030	0.5042	-3.0344
PID with FL	1.1581	0	0.6816	0.6393	14.1653	-3.1371
LQR	1.5624	3.05	0.7092	3.5444	0	-3.004

According to statistics above, in roll control loop, PID control and PID with FL control provide a similar performance, but convergence time of PID overwhelms that of PID with FL in Z control loop. Such result seems distinct but makes sense, because FL is possible not a positive component for linear model.

Statistics of LQR control indicates that its process require more time than others, which possibly caused by improper parameter configuration. Its performance in Z control could also prove that. This phenomenon leads to LQR tuning problem that could be optimized in further works.

Section V. Evaluation

By consideration of challenges in quadrotor model, the evaluation concentrates on stability, robustness, complexity, and convergence. It worthies a recall that simplification of nonlinear model is implemented based on assuming angular velocity is sufficient low. Therefore, in some subsection, results of both high and low angular velocity, deployed by different command pattern, will be presented. At the end of this section, a conclusion will be made with general comparison.

Before the content expands, it is necessary to explain the reason why nonlinear model (Eq. 3.16-3.19) is not the best choice in this case. The Fig. 5.1 and 5.2 illustrates the simulation output of PID control based on nonlinear model. An obvious phenomenon is that the nonlinear effect, comparing to its performance in linear model, can rapidly dampen even disable the controller in complicated maneuver. It makes statistics from simulation difficult to define, normalize and compare, which is not appropriate for evaluation. Therefore, a partial nonlinear model is deployed to enhance controller performance for distinct data.

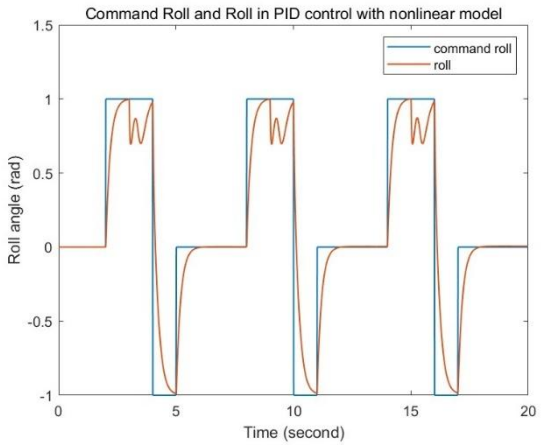

Fig. 5.1. Command roll and roll of PID controller in nonlinear model.

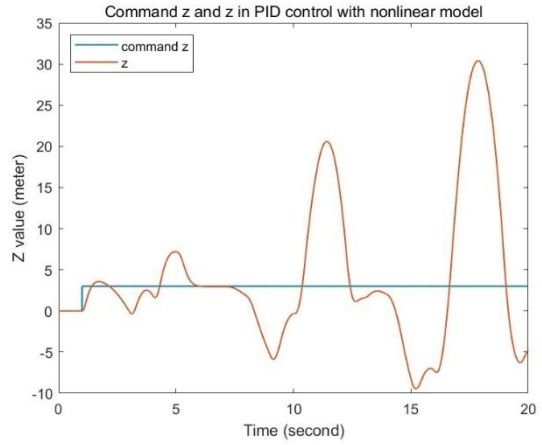

Fig. 5.2. Command Z and Z of PID controller in nonlinear model.

5.1 Stability

5.1.1 Low Angular Velocity

The low angular rate pattern is implemented by setting command angle through sinusoidal function. Derivation of trigonometric function will not larger than its amplitude, which means angular velocity is limited in a sufficient small range. The amplitude is set as 0.68 rad, which is close to maximum lean angle of KUR-400 [2]. Besides, reference of pitch, yaw and height are also activated to apply nonlinear effect. The simulation result of roll angle is illustrated by Fig. 5.3.

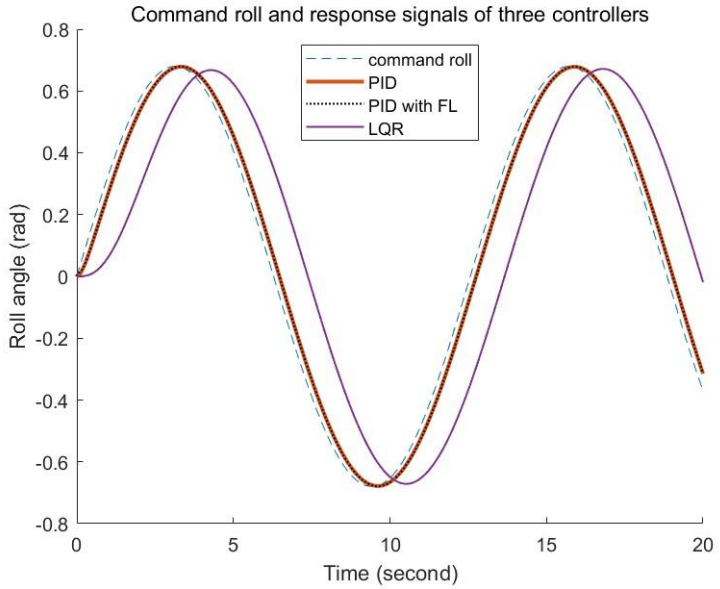

Fig. 5.3. Command roll and response signal of three controllers in low angular velocity.

Intuitively, the error tracking ability of LQR is worse than other two controllers. Based on simulation data, the time points, when four curves reach the first peak value (+0.68), are listed in Table.5.1. It indicates that performance between PID and PID with FL are similar, while LQR provide a longer delay.

Table 5.1. Time points of first peak value.

Signal	Time (s)	Delay(s)
Command Roll	3.1210	0
PID	3.3192	0.1982
PID with FL	3.2908	0.1698
LQR	4.2847	1.1637

Meanwhile, the result of altitude control emphasizes this distinction. As shown in Fig. 5.4, the LQR controller provides an unexpected oscillation. It also presents that PID with FL has excellent

stability, while simple PID control remains small oscillation and a long convergence time. Another information is that FL seems reduce overshoot phenomenon of simple PID, but causes a bit longer rising time. However, the total stabilizing time is much shorter than simple PID.

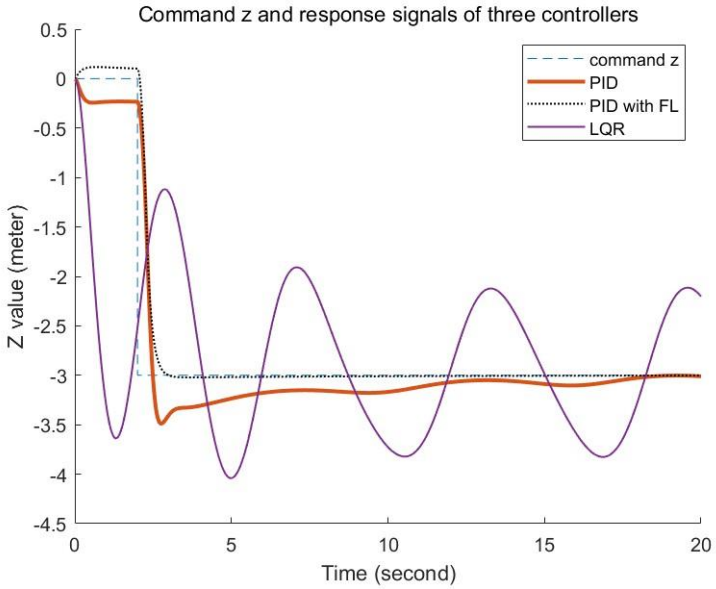

Fig. 5.4. Command Z and response signal of three controllers in low angular velocity.

5.1.2 High Angular Velocity

The implementation of high angular velocity pattern is similar to low angular velocity pattern, but sinusoidal function is replaced. It is implemented by repeat sequence generator that contains several step functions. Through rapid command value, the instantaneous derivation will be close to infinite, so that the system will respond with high angular velocity. The simulation result is shown in Fig. 5.5. Note that oscillation of PID with FL is very possibly caused by yaw changes.

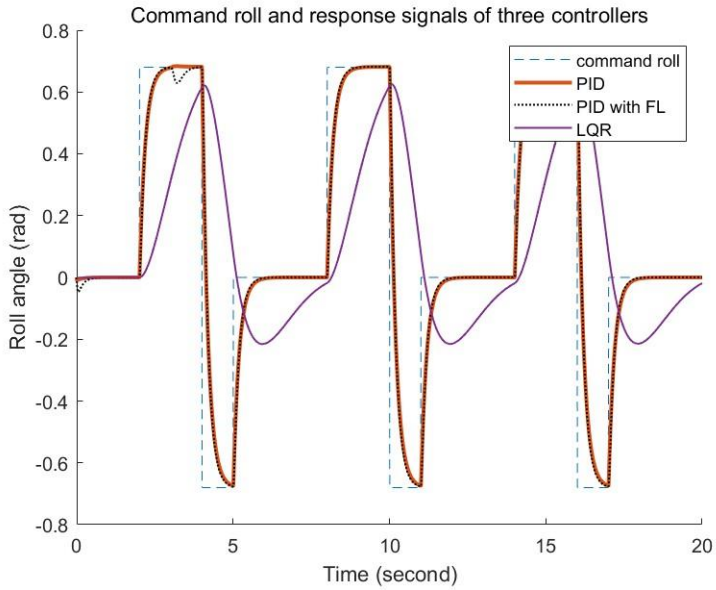

Fig. 5.5. Command roll and response signal of three controllers in high angular velocity.

Apparently, LQR control provide a poor performance, because its state hasn't reached target value

when the new command arrives. This phenomenon supports the conclusion in last section that LQR requires more rising time than other controllers. Another interesting phenomenon is that PID control and PID with FL still have similar performance. The rising time of both PID and PID with FL control for three target value in first loop is summarized in Table. 5.2. It could be calculated that the average rising time of PID is 0.9492 s, while that of PID with FL is 0.9213 s.

Table 5.2. Rising time for three target value.

Controller	Target value	Switch Time (s)	Signal Arriving Time (s)	Rising Time (s)
PID	0.68	2	3.0320	1.0320
	-0.68	4	5.0030	1.0030
	0	5	5.8126	0.8126
PID with FL	0.68	2	3.0068	1.0068
	-0.68	4	5.0008	1.0008
	0	5	5.7562	0.7562

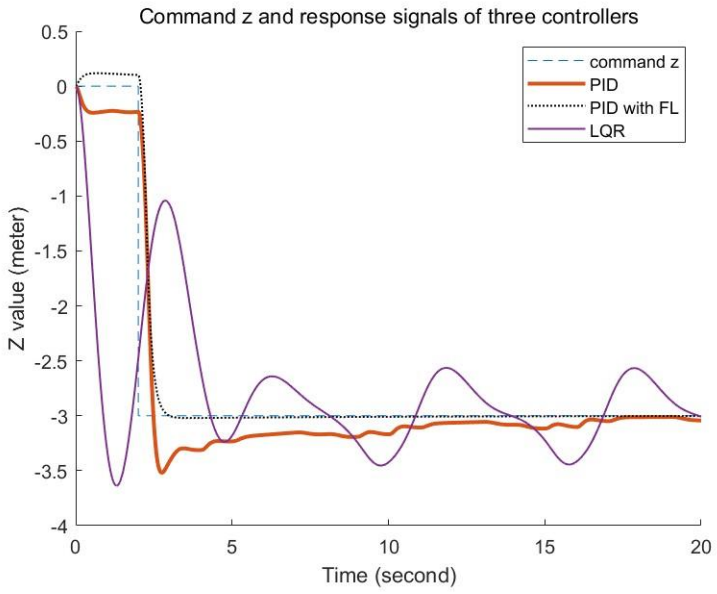

Fig. 5.6. Command Z and response signal of three controllers in high angular velocity.

About the altitude control, the simulation results, placed in Fig. 5.6, express a similar phenomenon to low angular velocity pattern. A distinction is that oscillation amplitude of LQR output is reduced, but its frequency remains steady. The small oscillation of PID control has a few changes. By consideration of two simulation case, the changes here are probably caused by attitude changes, which leads to nonlinear effect.

5.1.3 Steady State

In summary of previous result, there are two questions need further verification. One thing is that LQR could work in high angular velocity if it has sufficient rising time. Another question is whether these controllers could remain long term steady when target value has been arrived, especially whether altitude could be held when roll and pitch are not 0.

These two questions lead to another simulation whose command signal is provided by step function. The simulation result is illustrated as Fig. 5.7 and 5.8. The result illustrates that all three controllers could remain steady state in this case. Fig. 5.7 presents that LQR control requires more rising time than others two, which proves the previous assumption. Its performance, shown in Fig. 5.8, also indicates that LQR is sensitive and require a long time for stabilization.

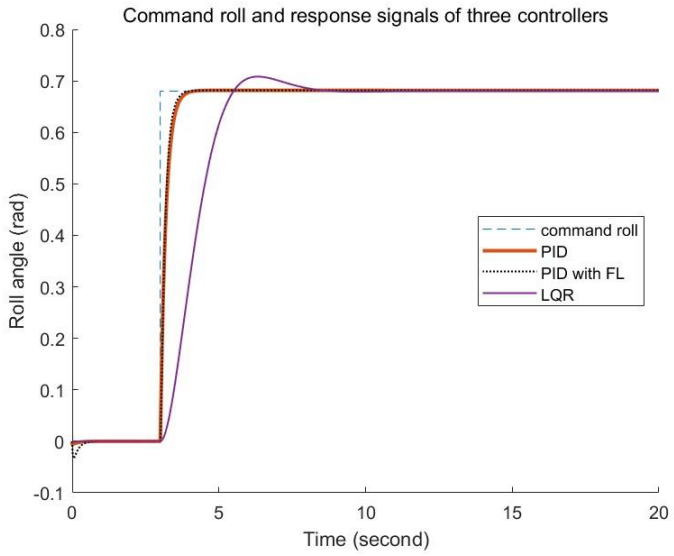

Fig. 5.7. Command roll and response signal of three controllers in steady state.

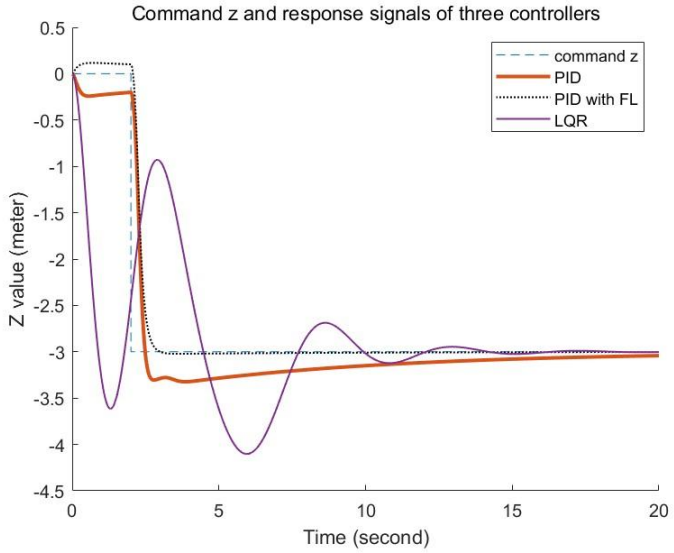

Fig. 5.8. Command Z and response signal of three controllers in steady state.

In summary from above simulation outputs, the PID control with FL has better general stability than others. Another assumption could be proved that the small oscillation of PID altitude control

is caused by attitude changes, because it disappears when attitude remaining steady.

5.2 Robustness

Since uncertainty is one of main challenges in quadrotor model. It is important to discuss robustness of controller. The disturbance to quadrotor originates from two parts: component and environment. The effect from component could be sensors, actuators, body and others.

In this case, only sensors are simulated to evaluate robustness of controller. It is also too difficult and inconvenient to build a complicated and detailed environment model. Therefore, the evaluation will be made with some simple assumption and explanation for specific phenomenon.

5.2.1 Initial State Adjustment

In above illustrations (Fig. 5.1-5.6), there is a phenomenon hasn't been discussed that a slope always appears near point t=0. It will be distinct when initial state will remain for a while. The reason of it is that the system is adjusting to disturbance caused by sensors and environment. In previous simulation, the environmental effect only has noise while sensors have random bias in initial state. So, that process could be used to evaluate robustness.

Considering process time or peak value could not completely represent robustness, the area of this period, integration value, will be used, as it contains both process time and peak value. The evaluation is arranged as follow. For roll control loop, it will be the maximum of absolute value of error integration within first 3 seconds, when quadrotor will remain initial state. For altitude control loop, as there is a steady error exists, the integration value will refer to value in t = 3 s for convenience. The result is summarized in Table. 5.3.

Table 5.3. Error integration of three controllers in initial state.

Controller	Maximum of Integration absolute value	
	Roll	Altitude
PID	0.0012	0.2107
PID with FL	0.0074	1.1736
LQR	0.0004	7.8787

This result indicates that a) controller is more robust against attitude disturbance rather than altitude disturbance; b) in altitude control, LQR is sensitive for disturbance; c) in this case, PID control has better robustness in both aspects. Note that conclusion b) could be proved by previous illustrations (Fig. 5.4, 5.6, 5.8), in which LQR provide an unexpected rapid rising.

5.2.2 Environmental Resistance

After discussion of sensor effect, another challenge need attraction is uncertainty from environment. In this case, the resistance is assumed as wind. The simulation patterns are the

steady state and high angular velocity to evaluate general robustness in static and dynamic. The simulation output of steady state pattern is illustrated in Fig. 5.9 and 5.10, while that of high angular velocity pattern is placed in Fig. 5.11 and 5.12.

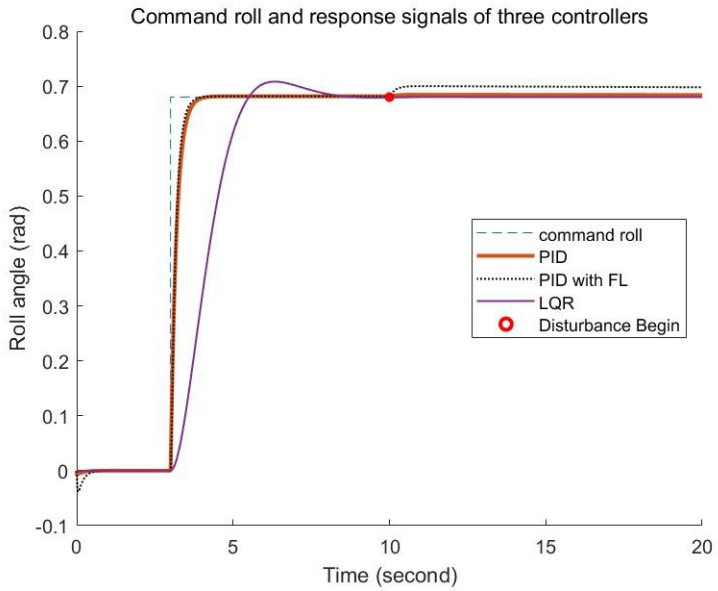

Fig. 5.9. Command roll and response signal of three controllers in steady state with disturbance.

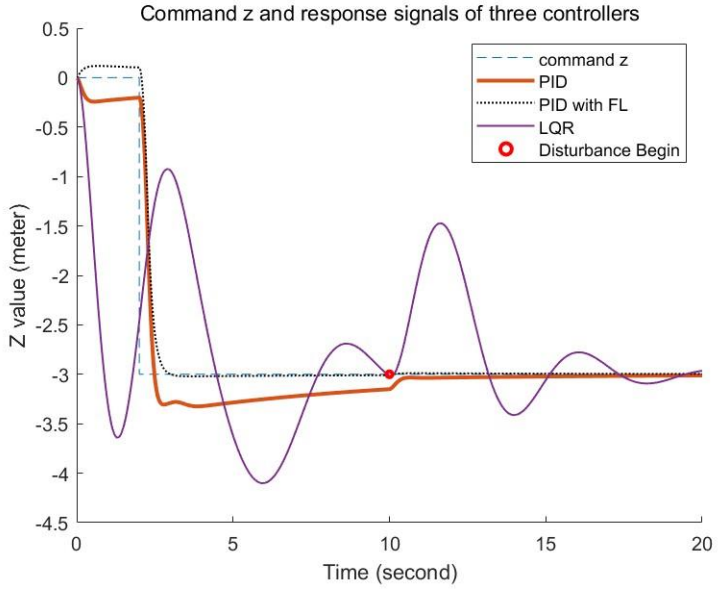

Fig. 5.10. Command Z and response signal of three controllers in steady state with disturbance.

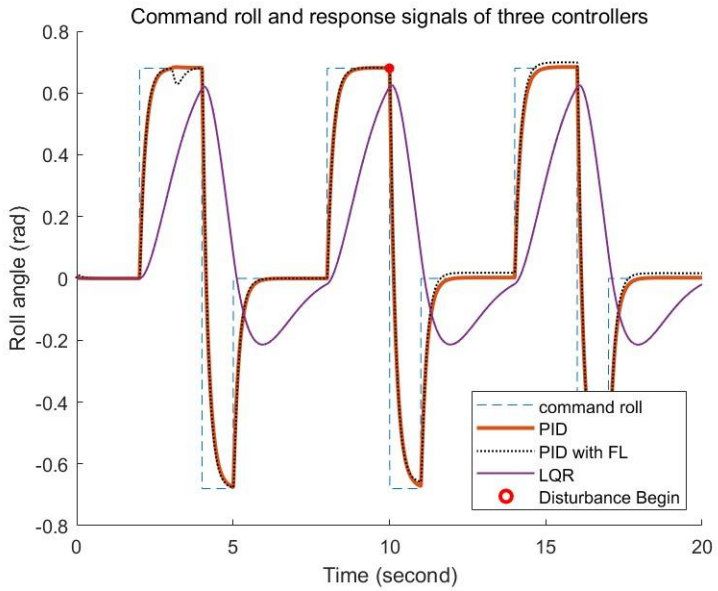

Fig. 5.11. Command roll and response signal of three controllers in high angular velocity with disturbance.

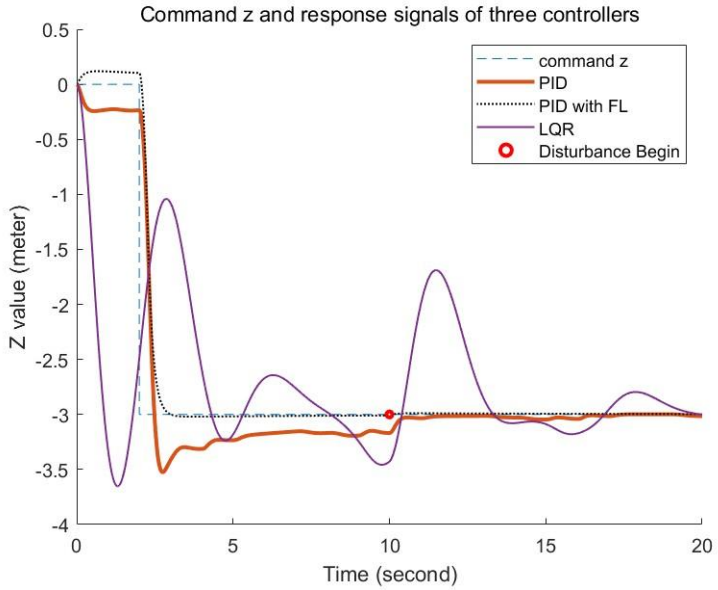

Fig. 5.12. Command Z and response signal of three controllers in high angular velocity with disturbance.

In steady state, Fig 5.9 illustrates that attitude control has satisfactory robustness against uncertain disturbance or resistance from environment. The simulated disturbance causes a few bias for PID with FL control in steady state and brings few influence for others. In high angular velocity, shown in Fig. 5.11, even disturbance appears on a switch point, the response of system remains expected performance, but disturbance still causes steady bias for PID with FL control.

The simulation of altitude control in both situations emphasize that LQR altitude control could not provide sufficient robustness against disturbance in this case, while the rest controllers could keep stable. This phenomenon coincided with result in last subsection. Another attracting phenomenon is that PID with FL provide better robustness than PID control in these instances. It makes these

two strategies competitive parity to each other in robustness.

In summary, LQR control is sensitive in altitude control, but could provide good robustness for attitude control. The possible reason of that is the controller parameter is not at optimal configuration, which means it is expected for further development. By comparison of these result, in robustness, PID with FL control and PID control have a similar superiority.

5.3 Complexity

It is hard to quantize complexity of controller. However, as Matlab/Simulink is used for simulation, the complexity of system could be transformed as number of used modules. The statistics of three controller implementations is listed in Table. 5.4.

Table 5.4. Used Simulink module for controller.

Controller	Number of used modules
PID	60
PID with FL	89
LQR	64

5.4 Summary

The evaluation result in above subsections could be summarized as Table. 5.5. It indicates that two PID related controllers are both appropriate choices. Generally, PID with FL control is more appropriate for its better performance in many aspects. Although it is more complicated than simple PID control, its proper cost performance makes it still available.

Table 5.5. Rank of controller in different aspects.

Controller	Property			
	Stability	Convergence	Robustness	Complexity
PID	2	2	1	1
PID with FL	1	1	1	3
LQR	3	3	3	2

The LQR performance of may have two reasons that needs further verification. The most apparent question is that deployed LQR parameter config, which mainly depends on matrix Q and R, is optimized based on linear model rather than nonlinear model. Tuning of LQR parameter is intrinsically difficult. So that such difference possibly makes it unable to provide expected result.

Besides, the variables within state space are not all included into LQR design. It may cause different solution and avoid optimal configuration.

Section VI. The Conclusion and Expectation

6.1 Conclusion

In conclusion, this article proposes an evaluation structure that includes stability, robustness, and complexity. Within that, challenges of quadrotor control and those important solutions with corresponding assumption and condition are explore comprehensively. The stability consists of error tracking ability, rising time and convergence time, and the operation pattern is set as low angular velocity, high angular velocity, and steady state switch to cover various situation. The evaluation of robustness contains two parts: a) initial state adjustment to test robustness of random noise; b) operation disturbance to test robustness of resistance. The complexity is included to consider cost performance of controller. At the end, the evaluation with data extraction, operation configuration, and comparison is presented. The reasons of various simulation result are also discussed, and sensible explanations are attained.

By comparison of simulation outputs, each controller has both satisfactory performance and remained development space. In stability evaluation, PID control could basically satisfy demand, but it is hard to develop a more precise and accurate result. Because it couldn't handle strongly coupled nonlinear effect well in altitude control, while its robustness of uncertain resistance is insufficient. It could be concluded from simulation result that its threshold of resolvable disturbance is smaller than PID with FL. From view of correlation, PID with FL control is developed edition. It remains those satisfactory performance and improves unexpected performance by eliminating nonlinear effect. In this project, although LQR control hasn't provide expected performance because of not optimal parameter config, it still presents that LQR could provide excellent robustness but long response time. In summary, in this project, PID with FL is prior choice with better general performance.

In addition, the dynamic model and different controller design has been mathematically presented with reliable and sufficient details. By comparison of linear, partial nonlinear and nonlinear simulation result, it could be concluded that nonlinear effect is proportional to nonlinear terms in state space. It directly affects stability of specific controller performance. The uncertain disturbance is also a challenge for controller robustness, especially when controller is sensitive.

6.2 Expectation

For further development, there are two main problems need further development: a) process of dynamic model; b) optimization of existing algorithms. The process of nonlinear and strongly coupled model relies on many assumptions. These assumptions brought convenience and good performance. However, the simulation result illustrates that controller performance will be dampened even disabled when out of limitations. To improve system performance, there are some further works could be developed from this project.

1. Tuning of LQR

Although LQR control does not completely work in this project, in some aspects it works, the

satisfactory performance expresses its potential. It makes attempts of tuning or further optimization of its structure worth rather than redesign or explore new control strategy.

2. Development of FL

Since PID with FL attains satisfactory result in this project, it is appropriate to consider a method that enable controller to work in completely nonlinear model. For future work, it has two directions: more complex FL component and integration with other control strategies, for example, LQR with FL. The more complex FL component will combine other terms to eliminate remained nonlinear terms. Design of such a component or system will be very difficult, because its complexity rapidly increases.

3. Exploration of innovative control strategy

To explore an optimal control strategy to attain satisfactory performance in completely nonlinear model, it is worth to implement another advanced control strategy, such as model predictive control, or even design a specific control strategy. However, there is a potential risk that it is possible to attain an innovative controller with low cost performance.

Reference

[1] H. Peng and X. Shen, "Multi-agent reinforcement learning based resource management in MEC-and UAV-assisted vehicular networks," *IEEE Journal on Selected Areas in Communications,* vol. 39, no. 1, pp. 131-141, 2020.

[2] U. c. via. *KUR-400 technical data*. [Online]. Available: https://www.up-caelivia.it/_files/ugd/8379a5_e1ec6c784dee46e58fc8fab817f88279.pdf

[3] H. Mo and G. Farid, "Nonlinear and Adaptive Intelligent Control Techniques for Quadrotor UAV – A Survey," *Asian Journal of Control,* vol. 21, no. 2, pp. 989-1008, 2018, doi: 10.1002/asjc.1758.

[4] L. Li, L. Sun, and J. Jin, "Survey of advances in control algorithms of quadrotor unmanned aerial vehicle," in *2015 IEEE 16th international conference on communication technology (ICCT)*, 2015: IEEE, pp. 107-111.

[5] S. I. Abdelmaksoud, M. Mailah, and A. M. Abdallah, "Control strategies and novel techniques for autonomous rotorcraft unmanned aerial vehicles: A review," *IEEE Access,* vol. 8, pp. 195142-195169, 2020.

[6] J. Kim, S. A. Gadsden, and S. A. Wilkerson, "A comprehensive survey of control strategies for autonomous quadrotors," *Canadian Journal of Electrical and Computer Engineering,* vol. 43, no. 1, pp. 3-16, 2019.

[7] R. Roy, M. Islam, N. Sadman, M. P. Mahmud, K. D. Gupta, and M. M. Ahsan, "A review on comparative remarks, performance evaluation and improvement strategies of quadrotor controllers," *Technologies,* vol. 9, no. 2, p. 37, 2021.

[8] A. Carrio, C. Sampedro, A. Rodriguez-Ramos, and P. Campoy, "A review of deep learning methods and applications for unmanned aerial vehicles," *Journal of Sensors,* vol. 2017, 2017.

[9] B. Han, Y. Zhou, K. K. Deveerasetty, and C. Hu, "A review of control algorithms for quadrotor," in *2018 IEEE international conference on information and automation (ICIA)*, 2018: IEEE, pp. 951-956.

[10] R. W. Beard, "Quadrotor dynamics and control," *Brigham Young University,* vol. 19, no. 3, pp. 46-56, 2008.

[11] J. Li and Y. Li, "Dynamic analysis and PID control for a quadrotor," in *2011 IEEE International Conference on Mechatronics and Automation*, 2011: IEEE, pp. 573-578.

[12] A. L. Salih, M. Moghavvemi, H. A. Mohamed, and K. S. Gaeid, "Flight PID controller design for a UAV quadrotor," *Scientific research and essays,* vol. 5, no. 23, pp. 3660-3667, 2010.

[13] E. Reyes-Valeria, R. Enriquez-Caldera, S. Camacho-Lara, and J. Guichard, "LQR control for a quadrotor using unit quaternions: Modeling and simulation," in *CONIELECOMP 2013, 23rd International Conference on Electronics, Communications and Computing*, 2013: IEEE, pp. 172-178.

[14] F. Ahmad, P. Kumar, A. Bhandari, and P. P. Patil, "Simulation of the Quadcopter Dynamics with LQR based Control," *Materials Today: Proceedings,* vol. 24, pp. 326-332, 2020.

[15] S. Khatoon, D. Gupta, and L. Das, "PID & LQR control for a quadrotor: Modeling and simulation," in *2014 international conference on advances in computing, communications and informatics (ICACCI)*, 2014: IEEE, pp. 796-802.

[16] H. Voos, "Nonlinear control of a quadrotor micro-UAV using feedback-linearization," in *2009 IEEE International Conference on Mechatronics*, 2009: IEEE, pp. 1-6.

[17] A. Saibi, R. Boushaki, and H. Belaidi, "Backstepping control of drone," *Engineering Proceedings,* vol. 14, no. 1, p. 4, 2022.

[18] M. Herrera, W. Chamorro, A. P. Gómez, and O. Camacho, "Sliding mode control: An approach to control a quadrotor," in *2015 Asia-Pacific Conference on Computer Aided System Engineering*, 2015: IEEE, pp. 314-319.

[19] D. Lee, H. Jin Kim, and S. Sastry, "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter," *International Journal of control, Automation and systems,* vol. 7, pp. 419-428, 2009.

[20] G. V. Raffo, M. G. Ortega, and F. R. Rubio, "An integral predictive/nonlinear H∞ control structure for a quadrotor helicopter," *Automatica,* vol. 46, no. 1, pp. 29-39, 2010.

[21] C. Li, Y. Wang, and X. Yang, "Adaptive fuzzy control of a quadrotor using disturbance observer," *Aerospace Science and Technology,* vol. 128, p. 107784, 2022.

[22] H. Bou-Ammar, H. Voos, and W. Ertel, "Controller design for quadrotor uavs using reinforcement learning," in *2010 IEEE International Conference on Control Applications*, 2010: IEEE, pp. 2130-2135.

[23] J. Hwangbo, I. Sa, R. Siegwart, and M. Hutter, "Control of a quadrotor with reinforcement learning," *IEEE Robotics and Automation Letters,* vol. 2, no. 4, pp. 2096-2103, 2017.

[24] K. M. Thu and A. Gavrilov, "Designing and modeling of quadcopter control system using L1 adaptive control," *Procedia Computer Science,* vol. 103, pp. 528-535, 2017.

[25] E. Okyere, A. Bousbaine, G. T. Poyi, A. K. Joseph, and J. M. Andrade, "LQR controller design for quad-rotor helicopters," *The Journal of Engineering,* vol. 2019, no. 17, pp. 4003-4007, 2019.

[26] U. Maqbool, T. Nomani, and H. Talat, "Neural network controller for attitude control of quadrotor," in *2019 Second International Conference on Latest trends in Electrical Engineering and Computing Technologies (INTELLECT)*, 2019: IEEE, pp. 1-8.

[27] T. Dierks and S. Jagannathan, "Neural network control of quadrotor UAV formations," in *2009 American Control Conference*, 2009: IEEE, pp. 2990-2996.

[28] L. S. Brian and L. L. Frank, "Aircraft control and simulation," *John Wiley & Sons, Inc., Hoboken, New Jersey,* 2003.

[29] M. F. Rahmat, A. Eltayeb, and M. A. M. Basri, "Adaptive feedback linearization controller for stabilization of quadrotor UAV," *International Journal of Integrated Engineering,* vol. 12, no. 4, pp. 1-17, 2020.

[30] U. C. Via. "Kur-400." https://www.up-caelivia.it/kur-400 (accessed 27 August, 2023).

[31] "The cube module overview." https://docs.cubepilot.org/user-guides/autopilot/the-cube-module-overview#the-cube-fixed-board (accessed 27 August, 2023).

[32] T. InvenSense, "World's Lowest Power 9-Axis MEMS MotionTracking™ Device," *URI: https://invensense. tdk. com/wp-content/uploads/2016/06/DS-000189-ICM-20948-v1,* vol. 3, 2017.

[33] MathWorks. "Linear-Quadratic Regulator (LQR) design - MATLAB lqr - MathWorks United Kingdom." https://uk.mathworks.com/help/control/ref/lti.lqr.html (accessed 27 August, 2023).