Transportation of Cable Suspended Load using Unmanned Aerial Vehicles

A Real-time Model Predictive Control Approach

R. Praveen Kumar Jain

Master of Science Thesis

TU Delft
Delft University of Technology

Delft Center for Systems and Control

Transportation of Cable Suspended Load using Unmanned Aerial Vehicles

A Real-time Model Predictive Control Approach

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

R. Praveen Kumar Jain

August 21, 2015

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

The work in this thesis was supported and conducted at the Micro Air Vehicle Laboratory, Faculty of Aerospace Engineering, Delft University of Technology (TU Delft).

Abstract

Unmanned Aerial Vehicles (UAV) have received an increasing amount of attention recently with many applications being actively investigated across the globe, and several related open research questions being actively pursued. Possible applications include search and rescue, disaster relief, environmental monitoring and surveillance, transportation, and construction. Transportation of cable suspended payloads using Unmanned Aerial Vehicles is one such application which is the topic of this research. Autonomous transportation of objects using UAV can contribute to the safe and reliable supply of food and medicine in remote or disaster-affected areas and even in commercial delivery of goods.

The state-of-the-art approaches towards the slung load transportation either develop nonlinear feedback control laws to stabilize the system to a predefined trajectory or employ open loop off-line trajectory planning schemes to generate optimal control inputs to the system. Most of these techniques often rely on availability of an accurate model of the system backed up with simulation results. Very few results exist which target experimental validation of the proposed method. Based on the findings of the previously conducted literature survey, it appears that the application of closed loop on-line trajectory generation and control schemes to transport a slung payload in swing free manner remains unanswered. The work in this thesis sets off to answer the research questions in this direction and address the issues that come along with experimental validation. Model Predictive Control (MPC) is a promising framework, which provides the means to tackle both the trajectory generation problem and the feedback control problem in an unified manner. As a result, it forms the most important component of this thesis.

Specific research problem that is addressed in this thesis is to transport a cable suspended load using quadrotor from one point to another, while minimizing the swing through the use of Linear Time Invariant MPC techniques. A nonlinear dynamic model for the quadrotor-slung load system is obtained and the structure within the system dynamics is exploited to decide the control strategy. Two different MPC formulations viz. MPC with integral action and MPC with Δu formulation are simulated and compared to Linear Quadratic control with integral action which acts as a benchmark controller. Backed with simulation results, it is shown through experimental validation that it is possible to control the swing of cable suspended load using linear control techniques. MPC being an computationally expensive task, state-of-the-art fast optimization solvers such as FORCES PRO is used to achieve online implementation of MPC for the quadrotor-slung load system. To this end, a new software framework for implementation of MPC is developed which establishes a wireless link with the quadrotor resulting in a real-time networked control loop.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank the two most important people without whom this thesis work would not be a success. Firstly, I would like to express my sincere gratitude to my MSc thesis supervisor dr.ir. Tamás Keviczky, Associate Professor, Delft Center for Systems and Control for his continuous support and valuable guidance during the execution of the thesis. My sincere thanks also goes to Mr. Erik van der Horst from the Micro Air Vehicle Lab, Faculty of Aerospace Engineering, TU Delft for his crucial inputs and insightful discussions during the experimental phase of this project work. Finally, I thank all those who made this journey a pleasant experience.

Delft, University of Technology
August 21, 2015

R. Praveen Kumar Jain

Dedicated to Mom and Dad

"Dreams is not what you see in sleep, it is the thing which doesn't let you sleep."
"Man needs difficulties in life because they are necessary to enjoy the success."
— *Dr. A.P.J. Abdul Kalam*

Chapter 1

Introduction

Unmanned Aerial Vehicles (UAV) have received an increasing amount of attention recently with many applications being actively investigated across the globe, and several related open research questions being pursued. Possible applications include search and rescue [1], disaster relief [2], environmental monitoring and surveillance, transportation, and construction. These are the applications which demand increased autonomous operation with minimal human/operator interference. The demands for autonomy coupled with advancements in the field of computer technology, miniaturized actuators and sensors have made it possible for development of smaller next generation UAV called Micro Aerial Vehicles (MAV). The advantages that come along with these vehicles is that they can be deployed in urban cluttered environments with low altitude flights, hence opening up the possibilities of using these flying robots for many civilian applications such as commercial delivery services. An overview of opportunities and challenges in development of autonomous MAV can be found in [3].

One of the most widely used MAV, is the Quadrotor, a Vertical takeoff and landing (VTOL) vehicle which can also take off and land in rough or inaccessible areas. These vehicles are actuated by four fixed propellers and have the ability to make agile and aggressive maneuvers. Their small size and agile maneuverability allows them to be used in both indoor and outdoor scenarios. However, this comes at a cost of increased power consumption. Most of the autonomous UAV developments in the past have been focused on the non-cooperative/non-interactive tasks where the main focus was on achieving autonomous flight, landing, takeoff and hovering capabilities. In recent years, the attention has however shifted towards interactive tasks which require the flying robots to interact with each other and with the environment. A flying inverted pendulum [4], grasping and manipulation [5, 6, 7], construction [8, 9], mobile manipulation [10], dance by flying robots [11, 12], transportation of a slung load [13], cooperative manipulation and transport [14, 15] are some of the reported applications.

1-1 Aim and Motivation

Among multitude of possible application areas, this master's thesis work targets one challenging problem namely, the transportation of cable suspended payload using an UAV. Such a scenario is very common in military applications where objects are transported from one place to another using helicopters. The slung payload exerts certain undesired forces and torques on the aircraft as it swings during the flight. These forces can result in instability of the flight leading to disastrous situations and must be controlled. It requires very skilled pilots to man the aircraft who can anticipate the undesired forces and pilot the aircraft safely. Clearly, this task is not trivial and considerable amount of research has been conducted to understand the dynamics of the system with cable suspended payloads [16, 17]. Consequently, the attention was directed towards use of automatic control techniques to reduce the effects of slung payload on the flight of aircraft.

With advent of Unmanned/Micro Air Vehicle technology, the slung load transportation problem finds applications in civilian domain such as commercial delivery services, transportation of food and medicine supplies in disaster affected areas, etc. The research work in this thesis aims to tackle the control design problem of slung load transportation using quadrotors. Quadrotor is a highly nonlinear, under-actuated system with fast unstable dynamics. Addition of the slung load to the system, results in increase in degree of under-actuation and hence making the control problem a challenging task.

1-2 State-of-the-art Methods

Transportation of cable suspended loads using UAV has been widely investigated in the literature. However, several open problems exist pertaining to control design and implementation. In this section, the state-of-the-art techniques in the literature are reviewed outlining the control design approach. Based on the findings of the literature survey, a problem statement for this thesis is formulated in next section. The approaches proposed in the literature can be roughly classified into nonlinear control, optimal control and learning based control.

Nonlinear Control

Reference [18] proposes a nonlinear control method which uses a hybrid model of quadrotor-slung load system specialized to a two-dimensional plane. The hybrid model captures two different behaviors of the system. First, when the cable has zero tension and the payload is a free falling object. Second, when the cable has a non-zero tension and hence influences the quadrotor. They show that the quadrotor-slung load system is a differentially flat system with the position of the load acting as a differentially flat output. Differential flatness is the property of the system, where all the internal variables of the system can be represented as a function of certain special variables and its derivatives, called the flat outputs. The problem of trajectory generation for the full nonlinear system is then reduced to generating trajectories for the flat outputs. The proposed controller developed for a planar 2D case, is able to track one of the following quantities, (a) quadrotor attitude, (b) load attitude, or (c) load position. The extension of the proposed approach for 3D case is given in [19]. A geometric nonlinear controller is presented in [20], which is an extension to the above mentioned approach to asymptotically stabilize the position of the quadrotor while aligning the flexible cable (series connection of n-rigid links) suspended load in the vertical position.

Optimal Control

The control of quadrotor-slung load system has been tackled as an open loop trajectory planning problem using Dynamic Programming or locally optimal Linear Quadratic feedback controllers are derived using iterative LQG method based on Linear Time Varying (LTV) models obtained by linearizing the nonlinear model around the operating point. Reference [21, 22] uses Dynamic Programming algorithm to generate swing free trajectories in open loop fashion which are then applied to the quadrotor. The quadrotor utilizes an adaptive controller, which deals with changes in center of gravity of the quadrotor due to swinging load [23]. They show that the controller is able to track the desired position and orientation references accurately and hence approximated as a double integrator model. The model used here assumes a massless rigid cable with non zero tension acting on the cable. The nonlinear suspended load model (basically a spherical pendulum) is linearized around the trajectory to obtain a LTV model. The combined LTV model of the quadrotor (approximated as double integrator) and suspended load model is used in the Dynamic Programming iterations to generate swing free trajectories. The proposed approach is inspired from [24, 25]. The Dynamic Programming algorithm is executed off-line to obtain a solution (swing free optimal trajectory) which is then executed on the real setup. Since the proposed approach uses linearized system models and assumes that the suspended cable is always taut during the motion, it is applicable only to non-aggressive maneuvers of the quadrotor. A limitation of this method is that it is an open loop method and hence is not robust to unseen uncertainties and disturbances.

Reference [26] presents an iterative LQG (iLQG) method for initial control trajectory design for a quadrotor with a cable suspended load. The initial control trajectory is designed in simulation and forms as policy initialization to the reinforcement learning algorithm during online learning trials. An iLQG method is an iterative method which returns locally optimal linear feedback controller for arbitrary nonlinear cost functions. In each iteration step, the system is linearized around the nominal trajectory and a quadratic approximation of the nonlinear cost function is minimized to obtain a new control input. Reader can refer to [27] for more details on iLQG method. The proposed approach utilizes the hybrid model of the quadrotor-slung load system presented in [18] and is able to not only reduce the swinging of the load during and after transport, but also generate aggressive motions to pass through a small opening like a window. Simulation results are provided to show the efficacy of the proposed approach. The drawback of the proposed approach is the inability to handle constraints on state and input variables during the control design process.

Artificial Intelligence based control methods

All the control methodologies described above heavily rely on the model of the system and its performance depends a lot on the accuracy of the model. Artificial Intelligence based methods like fuzzy control, neural network based control, learning control have been applied when the model of the system is either unavailable or not accurately known. Reference [28] presents a fuzzy based anti swing controller for a helicopter slung load system in hover flight. This anti swing controller acts in addition to the helicopter position tracking controller. The output from the fuzzy based anti swing controller are the displacements that are added to the helicopter trajectories in lateral and longitudinal directions. The nonlinear dynamic equations of the helicopter load system is linearized about the hovering conditions. The resulting linear model is used to design an LQR controller for tracking helicopter position. The fuzzy rule base is constructed using a clustering technique (exact clustering technique used is not mentioned). The load swing angles and swing rates are the fuzzy inputs and the traveling distance of the helicopter center of gravity (deviation in the helicopter center of gravity due to load swing) in the lateral and longitudinal direction are the fuzzy outputs. Yet another documented method is the use of Reinforcement learning framework to generate swing free motions [29].

Fast Model Predictive Control

Since the core component of this thesis is real-time[1] implementation of MPC, examples of real-time MPC documented in the literature are reviewed. There are no MPC approaches reported in the literature to control the quadrotor-slung load system to best of my knowledge. However, MPC has been applied to control the quadrotor. Reference [30] applies a real-time MPC technique for trajectory tracking control of a quadrotor. An on-board trajectory tracking scheme for control of quadrotor is proposed which consists of a three-tiered hierarchical control strategy. At low level resides the motor controller, which takes the desired thrust value of the individual rotor as input from the mid level attitude dynamics controller. Mid level controller consists of a high gain feedback controller applied to the attitude dynamics. Under this control, it is assumed that the vehicle angular velocity tracks the desired angular velocity accurately. This process is termed as dynamic reduction. Feedback linearization is then applied to the dynamically reduced system and the resulting Linear Time Invariant (LTI) system is used to design the high level MPC controller. Unconstrained linear MPC method is designed to control the LTI system with a quadratic cost function. The resulting Optimal Control Problem (OCP) is then an unconstrained Quadratic Programming (QP) problem, for which solution can be obtained analytically [31]. The mathematical details of the feedback linearization and MPC can be found in [30].

Another successful application of MPC is given in [32] which applies on-line, a MPC scheme for miniature helicopters. The nonlinear dynamic model of the helicopter is approximated by a Linear Time Varying (LTV) model. The proposed approach is validated on the experimental platform. The differential flatness property of the system is exploited to generate feasible trajectories in the flat output space for the nonlinear system. The method described in [33] is used for trajectory generation. The nonlinear dynamic equations are linearized along the generated trajectories to obtain LTV model of the system. The MPC problem is then posed as a convex optimization problem with a quadratic performance index with linear constraints. Automatic code generation method for convex optimization problem, CVXGEN [34] is then used to generate code for the real system implementation. The proposed concept of linearizing the nonlinear system equations along the trajectories to obtain a LTV model could be extended to address the control problem of quadrotor-slung load system. For more details related to the state-of-the-art methods, reader can refer to a previously conducted literature survey [35].

[1] Real-time in this case implies implementation of MPC fast enough for a system such as quadrotor-slung load system which has fast dynamics

1-3 Research and Engineering goals

Considering the state-of-the-art methods presented in the previous section, the following inference can be made regarding the control problem of the quadrotor-slung load system.

1. The nonlinear control techniques reported in the literature aim at stabilizing the quadrotor - slung load system to a reference trajectory. They do not address the trajectory generation problem. In the instances where the trajectory generation problem is addressed, they do so in an open loop fashion.
2. Optimal control methods aim at linearizing the system along the initial nonlinear trajectories to obtain a LTV model. The LTV model is then used to plan a swing free motion in an open loop manner.
3. Efficacy of most of the control methods are demonstrated through simulation results with very few examples of experimental validation.

The advantages of the nonlinear methods are that they can be used to generate aggressive motions for the quadrotor. They can however be sensitive to model mismatch. Optimal methods on the other hand, as the name suggests generate optimal control actions. An optimal control problem can be formulated to generate system trajectories in open loop fashion. Both the methods can generate unrealistic control inputs which may saturate the actuators. It would clearly be useful, if the advantages of both the methods could be combined in a single framework, while mitigating their limitations. Based on these findings, it appears that the application of closed loop on-line trajectory generation and control schemes to transport a slung payload in swing free manner remains an unexplored region. MPC is a promising framework, which provides the means to tackle both the trajectory generation problem and the feedback control problem in an unified manner. Another added advantage is the constraint handling capability of MPC.

The possibility of using LTI, LTV and nonlinear models within the MPC framework opens up a whole new set of research problems that is yet to be answered. Application of fast MPC methods for helicopters and the techniques used to control such systems using MPC could easily be extended to the quadrotor-slung load system. Reference [32] demonstrate that there exists state-of-the-art tools to solve computationally intensive operations like MPC for fast systems. The work in this thesis sets off to explore the relevant research problems in this direction. As a result, the goal of this thesis can be classified as research goals and engineering goals. Research goals of this thesis are

1. Investigate application of LTI Model Predictive Control methods to transport a cable suspended load from one point to another in swing free manner using a quadrotor.
2. Design MPC controllers for slung load transportation task, compare different formulations and evaluate the performance through simulations. Different control formulations are explored which address issues of model mismatch when such controllers are to be implemented on a real system.
3. Explore the use of state-of-the-art fast optimization solvers which would enable implementation of MPC for a fast system such as quadrotor-slung load system.

4. Experimentally validate the designed control methods and also address the problem of state estimation.

5. From a futuristic point of view, the work in this thesis would pave the way to investigate challenging problems such as LTV MPC and Nonlinear MPC.

The engineering goals of this thesis is to develop a new software framework which would allow implementation of MPC for quadrotors and could be easily extended to implement advanced methods based on LTV and Nonlinear models. The work in this thesis would lay a foundation for future work from both control theoretical point of view and experimental validation point of view.

1-4 Organization of the Report

In this chapter the problem of slung load transportation using quadrotor was introduced and motivated. State-of-the-art methods were briefly reviewed and problem statement was formulated for the thesis. The remaining report is organized as follows.

The modeling of the quadrotor-slung load system is discussed in Chapter 2. The nonlinear model of the system obtained from the first principles are discussed with sufficient depth. The assumptions and drawbacks of the model are also explained. Since the experimental validation is one of the important part of this thesis, the need for closed loop identification of attitude dynamics is motivated and validation results are presented. A LTI model is obtained which would be used in the subsequent chapter for control design.

Chapter 3 then dives into the state estimation and control problem. The chapter starts with a brief explanation of the Kalman filter and presents the results of state estimation. The need for a state estimator is motivated. The rest of the chapter presents the control methods such as Linear Quadratic (LQ) control with integral action which forms the benchmarking controller. A generic MPC problem is briefly explained and two different MPC formulations are discussed. The fast optimization solvers used in this work are explained followed by remarks on tuning and stability of the controllers. The chapter ends with extensive analysis of simulation results and comparison of different formulations.

Having explained the theoretical aspects of control design, Chapter 4 focuses on experimental aspects of this thesis. The hardware and software setup is explained. A new software framework developed for implementation of MPC is briefly explained. The controllers developed in Chapter 3 are experimentally validated.

The report finally concludes with Chapter 5 where the main research findings of the thesis are discussed. Future recommendations are provided which would further enhance the results and findings obtained in this work.

Chapter 2

System Modeling

The first and foremost requirement in the design, analysis, simulation and implementation of a model based control strategy, is to have a reliable and acceptable description or a model of the system. Quadrotor-slung load system can be seen as a system consisting of two rigid bodies physically connected to each other. The complete dynamics of such a system is complicated and it would be difficult to model them accurately considering all the external forces and disturbances such as the aerodynamic forces acting on the cable and the quadrotor, elasticity of the cable, etc. Even if an accurate model could be obtained, it would be too complicated and unmanageable in terms of number of states and inputs for the controller design. Therefore, it is desirable to consider a simplified model which retains the most important dynamics of the system resulting in a model with minimum number of states and input.

Section 2-1 explains the equations of motion of the quadrotor obtained using the Newton-Euler method. The quadrotor kinematics, external inputs acting on the quadrotor and modeling assumptions are explained. The modeling of the quadrotor-slung load system is discussed in Section 2-2. The model structure is then exploited to decide on the control strategy and the need for identification of closed loop attitude dynamics is motivated in Section 2-3. The results of identification are also discussed. Since LTI models are needed for the control design, linearization results are presented in Section 2-4. The model of the quadrotor-slung load system is validated in Section 2-5 and the sources of model mismatch are explained.

The notations used throughout the thesis are as follows. A scalar variable is denoted by a lower case letter x whereas a vector is denoted by boldface lower case letter $\mathbf{x}$. Matrices are denoted by upper case letter such as A. Superscripts and subscripts are used extensively. Hence their meaning and the quantities they represent are explained as and when they are used.

2-1 Quadrotor Model

A quadrotor is modeled as a rigid body with 6 Degrees of Freedom (DOF) - 3 DOF in position and 3 DOF in attitude. In order to do so, define two right handed Cartesian coordinate

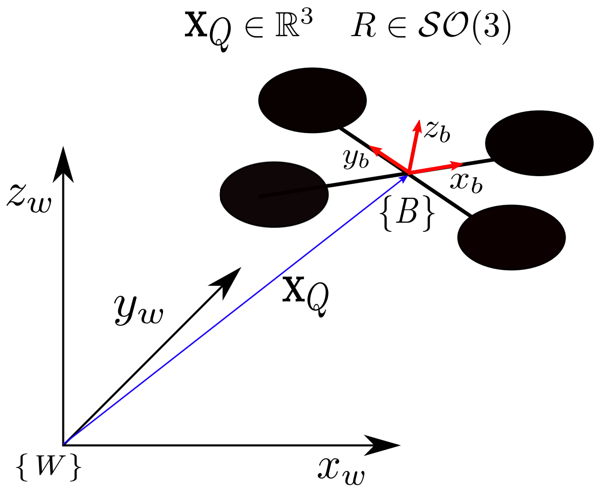


Figure 2-1: Description of coordinate frames associated with the quadrotor

frames $\{W\}$ and $\{B\}$ which implies a world or inertial coordinate frame and a coordinate frame attached to the body of the quadrotor respectively. Technically, the inertial frame is a reference frame which remains unchanged and all the measurements are made with respect to this particular coordinate frame. In aerospace applications, Earth Centered Earth Fixed (ECEF) frame is typically used as an inertial frame. However, since the experiments presented in this report are conducted in a small flight area, it would be safe to consider a local right handed coordinate frame attached to the flight area as an inertial frame. The chosen coordinate frames are East-North-Up (ENU) coordinate frames in which the x-axis points East, y-axis points North and z-axis points Upwards. Figure 2-1 shows the coordinate frames associated with the quadrotor.

2-1-1 Kinematic relations

Kinematics is a branch of mechanics which studies the motion of a body or a system of bodies without consideration of the forces and torques acting on it. While deriving equations of motion for system of bodies, it is always important to have equations that convert variables of interest from one reference coordinate frame to another. In this section, the kinematic relations pertaining to the quadrotor are presented which will be useful to obtain the equations of motion for a quadrotor.

Lets first define the generalized coordinates, namely, the position and the attitude which corresponds to the 6 DOF motion of the quadrotor. The position of the quadrotor $\mathbf{x}_Q \in \mathbb{R}^3$ and $\mathbf{x}_Q = \begin{bmatrix} x_Q & y_Q & z_Q \end{bmatrix}^T$, is measured with respect to the $\{W\}$ frame. The attitude of the quadrotor is denoted by the rotation matrix $R \in \mathcal{SO}(3)$. The quadrotor is not expected to make aggressive maneuvers and hence the rotation matrix is parameterized as given in Eq. (2-1) using ZYX Euler angles $\boldsymbol{\Theta} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$ which corresponds to the roll, pitch and yaw rotations respectively[1].

$$R(\phi,\theta,\psi) = R_Z(\psi)R_Y(\theta)R_X(\phi) = \begin{bmatrix} C_\psi C_\theta & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi C_\theta & S_\psi S_\theta S_\phi + C_\psi C_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix} \tag{2-1}$$

where $C_x = \cos x$ and $S_x = \sin x$

The linear velocity of the quadrotor, $\mathbf{v^B} = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^T$ expressed in the body fixed frame can be transformed into linear velocities expressed in the world frame $\dot{\mathbf{x}}_Q$ as

$$\dot{\mathbf{x}}_Q = R(\phi,\theta,\psi)\mathbf{v^B} \tag{2-2}$$

Similar to the linear velocities, the angular velocities of the quadrotor can be expressed both in body fixed frame, $\boldsymbol{\omega^B} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$ and the world frame $\dot{\boldsymbol{\Theta}}$. The transformation of angular velocities from the body frame to the world frame is given by Eq. (2-3) where $T_x = \tan x$.

$$\begin{aligned} \dot{\boldsymbol{\Theta}} &= \begin{bmatrix} 1 & S_\phi T_\theta & C_\phi T_\theta \\ 0 & C_\phi & -S_\phi \\ 0 & S_\phi/C_\theta & C_\phi/C_\theta \end{bmatrix} \boldsymbol{\omega^B} \\ &= W(\phi,\theta)\boldsymbol{\omega^B} \end{aligned} \tag{2-3}$$

The inverse relation which transforms the rate of change of Euler angles to representations in the body fixed frame is given in Eq. (2-4). This equation is important to derive the dynamics of the quadrotor-slung load system using Euler-Lagrange approach in the next section.

$$\begin{aligned} \boldsymbol{\omega^B} &= \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \cos\theta\sin\phi \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \dot{\boldsymbol{\Theta}} \\ &= W_{inv}(\phi,\theta)\dot{\boldsymbol{\Theta}} \end{aligned} \tag{2-4}$$

Eq. (2-2) and Eq. (2-3) can be collectively written as Eq. (2-5) to describe the velocity kinematics, also called the "Jacobian" matrix.

$$\begin{bmatrix} \dot{\mathbf{x}}_Q \\ \dot{\boldsymbol{\Theta}} \end{bmatrix} = \begin{bmatrix} R(\phi,\theta,\psi) & 0 \\ 0 & W(\phi,\theta)) \end{bmatrix} \begin{bmatrix} \mathbf{v^B} \\ \boldsymbol{\omega^B} \end{bmatrix} \tag{2-5}$$

2-1-2 Equations of motion: Newton-Euler method

The equations of motion of mechanical systems can be obtained using two widely known methods, namely, Newton-Euler method and Euler-Lagrange method. Both these methods result

[1] Note that the Euler angle representations are prone to gimbal lock or singularities for a particular orientation. For ZYX Euler angles, gimbal lock occurs with a 90 degree rotation about y-axis causing the z-axis and the x-axis to align.

in equivalent set of equations. For simpler systems, Newton-Euler method is the preferred choice as it is easy and intuitive. However, as the complexity of the system increases, it is difficult to apply the Newton-Euler method and this is where the Euler-Lagrange method has its advantages. Consequently, Newton-Euler method has been used to obtain the equations of motion of the quadrotor.

The equations of motion can be derived with respect to the body fixed frame or with respect to the world frame. The Newton-Euler equations of motion for a rigid body in 3D space, expressed in the body frame are given by Eq. (2-6). For more details regarding the Newton-Euler equation, reader is referred to the consult [36] and [37].

$$\begin{bmatrix} mI & 0 \\ 0 & \mathcal{I} \end{bmatrix} \begin{bmatrix} \dot{v}^B \\ \dot{\omega}^B \end{bmatrix} + \begin{bmatrix} \omega^B \times mv^B \\ \omega^B \times \mathcal{I}\omega^B \end{bmatrix} = f^B \tag{2-6}$$

The matrix $\mathcal{I} = \text{diag}(I_{xx}, I_{yy}, I_{zz})$ is the rotational inertia matrix and m is the mass of the rigid body. f^B denotes the external forces applied to the system expressed in the body fixed coordinate frame. The first part of the equation describes the translational dynamics of the rigid body and the second part describes the rotational dynamics. The quadrotor aircraft can be considered as a rigid body in 3D space and hence, Eq. (2-6) clearly captures the dominant dynamics of the quadrotor. Eq. (2-6) is re-written for quadrotor as Eq. (2-7)

$$\begin{bmatrix} m_Q I & 0 \\ 0 & \mathcal{I} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}}^{\mathbf{B}} \\ \dot{\boldsymbol{\omega}}^{\boldsymbol{B}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\omega}^{\boldsymbol{B}} \times m_Q \mathbf{v}^{\mathbf{B}} \\ \boldsymbol{\omega}^{\boldsymbol{B}} \times \mathcal{I}\boldsymbol{\omega}^{\boldsymbol{B}} \end{bmatrix} = \mathbf{f}^{\mathbf{B}} \tag{2-7}$$

Since, the translational position and velocity measurements are obtained in world frame, it would be ideal to write the translational dynamics expressed in world frame. It turns out that the resulting equations are much simpler and useful from control point of view as can be seen in Eq. (2-8) where g is the gravity acting on the quadrotor, $\mathbf{f}^{\mathbf{W}}$ are external forces acting on the quadrotor expressed in world frame and $\mathbf{z}_w = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$ is unit vector along the z-axis of world frame.

$$m_Q \ddot{\mathbf{x}}_Q = \mathbf{f}^{\mathbf{W}} + m_Q g \mathbf{z}_w \tag{2-8}$$

It is however advantageous to express the rotational dynamics in body fixed frame, as the equation is compact and the sensors measure angular velocities in body frame. The resulting dynamics of the quadrotor is given in Eq. (2-9). $\boldsymbol{\tau}^{\boldsymbol{B}}$ are the torques applied to the quadrotor by the motors expressed in the body frame.

$$\begin{aligned} m_Q \ddot{\mathbf{x}}_Q - m_Q g \mathbf{z}_w &= \mathbf{f}^{\mathbf{W}} \\ \mathcal{I}\dot{\boldsymbol{\omega}}^{\boldsymbol{B}} + \boldsymbol{\omega}^{\boldsymbol{B}} \times \mathcal{I}\boldsymbol{\omega}^{\boldsymbol{B}} &= \boldsymbol{\tau}^{\boldsymbol{B}} \end{aligned} \tag{2-9}$$

2-1-3 External inputs

In the previous subsection, the dynamics of the quadrotor was obtained subject to external forces and torques. In this subsection, the relation between these external forces and torques with the rotational velocities of the four quadrotor motors are established. The four motors of the quadrotor rotate with angular velocity w_i producing a vertical lift/thrust T_i for $i = \{1, \cdots, 4\}$. The individual thrust of each rotor is approximately related to the angular velocity as $T_i = bw_i^2$, where b is the thrust factor. Combination of these four individual thrusts results in a collective thrust acting along z-axis of body fixed frame, denoted by f and torques about the three body axes $\boldsymbol{\tau^B} = \begin{bmatrix} \tau_x & \tau_y & \tau_z \end{bmatrix}^T$. The external forces and torques acting on the quadrotor expressed in the body fixed frame is then given as

$$\mathbf{f^B} = \begin{bmatrix} 0 \\ 0 \\ f \end{bmatrix} \tag{2-10}$$
$$\boldsymbol{\tau^B} = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}$$

The total thrust f and the torques $\boldsymbol{\tau^B}$ are related to angular velocities of each rotor as

$$\begin{bmatrix} f \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ Lb & -Lb & -Lb & Lb \\ Lb & Lb & -Lb & -Lb \\ -d & d & -d & d \end{bmatrix} \begin{bmatrix} w_1^2 \\ w_2^2 \\ w_3^2 \\ w_4^2 \end{bmatrix} \tag{2-11}$$

where d is the drag factor and L is the distance between the motor and center of mass of the quadrotor. Since we have the translational dynamics defined in the world frame, the external linear forces acting on the quadrotor expressed in world frame can be written as

$$\begin{aligned} \mathbf{f^W} &= fR(\phi, \theta, \psi)\mathbf{z}_w \\ &= R(\phi, \theta, \psi)\mathbf{f^B} \end{aligned} \tag{2-12}$$

Substituting Eq. (2-10) and Eq. (2-12) in Eq. (2-9), we have the complete dynamic model of the quadrotor given in Eq. (2-13)

$$\begin{aligned} m_Q\ddot{\mathbf{x}}_Q - m_Q g\mathbf{z}_w &= fR(\phi, \theta, \psi)\mathbf{z}_w \\ \mathcal{I}\dot{\boldsymbol{\omega}}^{\boldsymbol{B}} + \boldsymbol{\omega}^{\boldsymbol{B}} \times \mathcal{I}\boldsymbol{\omega}^{\boldsymbol{B}} &= \boldsymbol{\tau^B} \end{aligned} \tag{2-13}$$

The above equations can be written in an elaborated way as follows

$$\begin{aligned} \ddot{x}_Q &= \frac{f}{m_Q}\left(C_\psi S_\theta C_\phi + S_\psi S_\phi\right) \\ \ddot{y}_Q &= \frac{f}{m_Q}\left(S_\psi S_\theta C_\phi - C_\psi S_\phi\right) \\ \ddot{z}_Q &= \frac{f}{m_Q}\left(C_\theta C_\phi\right) - g \\ \dot{\omega}_x &= \frac{I_{yy} - I_{zz}}{I_{xx}}\omega_y\omega_z + \tau_x \\ \dot{\omega}_y &= \frac{I_{zz} - I_{xx}}{I_{yy}}\omega_z\omega_x + \tau_y \\ \dot{\omega}_z &= \frac{I_{xx} - I_{yy}}{I_{zz}}\omega_x\omega_y + \tau_z \end{aligned} \tag{2-14}$$

Note that the translational dynamics and the attitude dynamics of the quadrotor are decoupled. The structure within the dynamics of the quadrotor allows design of two cascaded, decoupled controllers to control the position and attitude respectively. This important strategy forms the basis for controller design in this work, both for the quadrotor and the quadrotor-slung load system.

2-1-4 Modeling assumptions

The quadrotor is a highly nonlinear, under actuated, unstable system. It is not possible to consider all the nonlinearities/disturbances acting on the system and explicitly model them for control design. The reason being that some of the nonlinearities cannot be accurately modeled or some effects can safely be ignored. In this subsection, assumptions made in order to obtain a simple yet useful model are explained.

- The origin of the body fixed frame $\{B\}$ coincides with the center of mass of the quadrotor.
- The structure of the quadrotor is rigid and symmetric.
- The aerodynamic disturbances acting on the quadrotor are neglected. Since the flight tests are conducted in the indoor environment, there are no distrubances such as wind gusts acting on the quadrotor.
- The propellers are rigid and hence the thrust produced by individual rotor is parallel to the axis of rotor.

2-2 Quadrotor-Slung Load Model

In the previous section, a dynamic model for the quadrotor aircraft was derived. However, the objective of this thesis work is to design a model based controller for swing free transport of a slung mass using quadrotor. Hence, the effects of the slung mass on the dynamics of the quadrotor needs to be modeled. Clearly, it is easy to see that the slung mass attached to the quadrotor contributes to additional forces acting along the translational axes. It also increases the degree of under-actuation making the control design problem a challenging task. In this section, equations of motion for the quadrotor-slung load system are derived.

2-2-1 Modeling assumptions

Before we proceed towards the dynamic modeling of the quadrotor-slung load system, certain assumptions and approximations made are explained.

- The cable used to connect the payload with the quadrotor is massless and rigid. The rigid cable assumption means that there is always a non-zero tension acting on the cable. As a result, the current model does not take into account the flexibility of the cable and instances when the tension acting on the cable is zero.

- The payload connected to the cable, is approximated as a point mass.
- Aerodynamic effects acting on the payload and the cable as the quadrotor flies are neglected.
- The suspension point of the cable on the quadrotor is same as the center of mass of the quadrotor and the origin of the body fixed frame.
- The suspension point is frictionless. Hence the slung payload can be considered as a spherical pendulum connected to a spherical frictionless joint.

2-2-2 Kinematic relation

Similar to the dynamic modeling of the quadrotor, the kinematics of the quadrotor-slung load system is derived. The kinematic equations are important to obtain the Jacobian matrix which would later be used to derive the equations of motion using Euler-Lagrange method. Figure 2-2 shows the schematic of a quadrotor connected to the payload of mass m_L using a rigid cable of length l. As before, the position of the quadrotor with respect to the world frame is denoted by $\mathbf{x}_Q \in \mathbb{R}^3$, the position of the payload is denoted by $\mathbf{x}_L \in \mathbb{R}^3, \mathbf{x}_L = \begin{bmatrix} x_L & y_L & z_L \end{bmatrix}^T$ with respect to the world frame and the attitude of the quadrotor is denoted by rotation matrix $R \in SO(3)$ parameterized by ZYX euler angles (ϕ, θ, ψ) as given in Eq. (2-1). The position of the slung payload can also be described in terms of two swing angles, ϕ_L and θ_L. In order to do so, a new coordinate frame $\{S\}$ is attached to the suspension point with its origin coincident with the origin of the body fixed frame $\{B\}$. The load swing angles ϕ_L and θ_L are described as rotations about the y-axis $(R_Y(\phi_L))$ and x-axis $(R_X(\theta_L))$ of the $\{S\}$ frame respectively[2]. The generalized coordinates $\mathbf{q}$ and the generalized velocities $\dot{\mathbf{q}}$ for the quadrotor-slung load system are

$$\mathbf{q} = \begin{bmatrix} x_Q & y_Q & z_Q & \theta_L & \phi_L & \phi & \theta & \psi \end{bmatrix}^T$$

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{x}_Q & \dot{y}_Q & \dot{z}_Q & \dot{\theta}_L & \dot{\phi}_L & \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T$$

The kinematic map $g : \mathbf{q} \to \mathbb{R}^6$ which relates the generalized coordinates to the Cartesian coordinates of the quadrotor and slung load is given in Eq. (2-15)

$$\begin{aligned} \mathbf{x}_Q &= \mathbf{x}_Q \\ \mathbf{x}_L &= \mathbf{x}_Q + R_S^B R_Y(\phi_L) R_X(\theta_L) \begin{bmatrix} 0 & 0 & l \end{bmatrix}^T \end{aligned} \tag{2-15}$$

where

$$R_Y(\phi_L) = \begin{bmatrix} \cos\phi_L & 0 & \sin\phi_L \\ 0 & 1 & 0 \\ -\sin\phi_L & 0 & \cos\phi_L \end{bmatrix}$$

[2] The particular choice of rotation, $R_X(\theta_L)$ and $R_Y(\phi_L)$ ensures that there is no singularity at the vertically downward stable equilibrium position of the slung load. Use of azimuth and elevation angle to describe the swing angle of slung load would lead to singularity at equilibrium position.

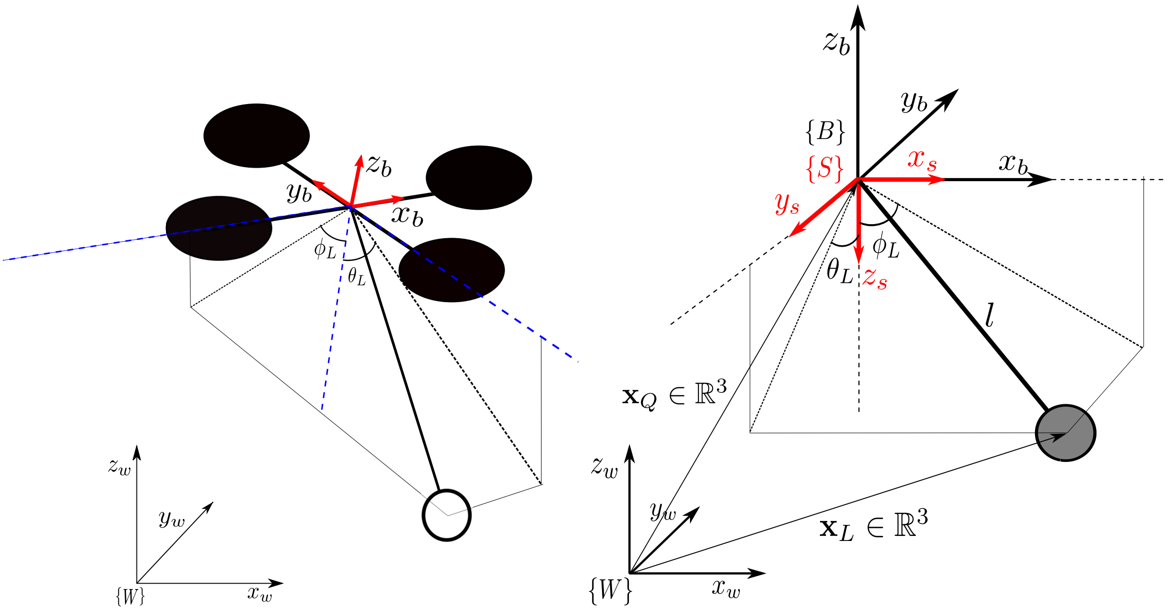


Figure 2-2: Schematic of quadrotor-slung load system

$$R_X(\theta_L) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_L & -\sin\theta_L \\ 0 & \sin\theta_L & \cos\theta_L \end{bmatrix}$$

$$R_S^B = R_X(\pi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

Inverse kinematic equations for slung mass

Though the inverse kinematic equations are not used in the modeling of the quadrotor-slung load system, it is presented here as they are used for experimental validation of the controllers. The position of the slung mass is measured using external tracking cameras, which provide the 3D position of the slung mass $\mathbf{x}_L$. The inverse equations presented here are used to convert the 3D position to the swing angles. The position of the slung mass is related to the swing angles through the following equation

$$\begin{aligned} x_L &= x_Q + l\cos\theta_L\sin\phi_L \\ y_L &= y_Q + l\sin\theta_L \\ z_L &= z_Q - l\cos\theta_L\cos\phi_L \end{aligned} \tag{2-16}$$

Solving these three equations for two unknowns result in the inverse equations for ϕ_L (Eq. (2-17)) and θ_L (Eq. (2-18)).

$$\phi_L = \tan^{-1}\left(\frac{-(x_L - x_Q)}{z_L - z_Q}\right) \tag{2-17}$$

$$\theta_L = \sin^{-1}\left(\frac{y_L - y_Q}{l}\right) \tag{2-18}$$
$$\theta_L = \tan^{-1}\left(\frac{(y_L - y_Q)\sin\phi_L}{x_L - x_Q}\right)$$

Velocity Kinematics

Having obtained the kinematic map $g(\mathbf{q})$, the velocity kinematics is obtained by partially differentiating $g(\mathbf{q})$ of Eq. (2-15) with respect to $\mathbf{q}$.

$$\begin{bmatrix} \dot{x}_Q \\ \dot{y}_Q \\ \dot{z}_Q \\ \dot{x}_L \\ \dot{y}_L \\ \dot{z}_L \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & -l\sin\phi_L\sin\theta_L & l\cos\phi_L\cos\theta_L \\ 0 & 1 & 0 & l\cos\theta_L & 0 \\ 0 & 0 & 1 & l\cos\phi_L\sin\theta_L & l\cos\theta_L\sin\phi_L \end{bmatrix}}_{J_1(\mathbf{q})} \begin{bmatrix} \dot{x}_Q \\ \dot{y}_Q \\ \dot{z}_Q \\ \dot{\theta}_L \\ \dot{\phi}_L \end{bmatrix} \tag{2-19}$$

As mentioned in the previous section, the angular velocities are always measured in the body fixed frame and hence, the angular velocity in body fixed frame written in the terms rate of change of Euler angles is given by Eq. (2-4) reproduced below

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \cos\theta\sin\phi \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix}}_{J_2(\mathbf{q})} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

The jacobian matrix, which will be used in the next section for obtaining the dynamic model can then be written as

$$J(\mathbf{q}) = \begin{bmatrix} J_1(\mathbf{q}) & 0 \\ 0 & J_2(\mathbf{q}) \end{bmatrix}$$

2-2-3 Equations of motion: Euler-Lagrange method

The principle of Euler-Lagrange method relies on the energy property of the mechanical system to obtain the equations of motion. While Newton-Euler method which relies on summation of forces and torques acting on a rigid body was used to model the quadrotor, it becomes difficult to model the interaction forces and torques between a system of multiple rigid bodies. The Euler-Lagrange method then becomes a preferred choice for modeling a system such as the quadrotor-slung load system. The preferred choice also off-course depends on the proficiency of the user with a particular method. In this subsection, the quadrotor-slung load system is modeled using the Euler-Lagrange method. The quadrotor-slung load system can also be modeled using Newton-Euler method, where the effects of slung mass on the dynamics of the quadrotor are modeled as input disturbances as shown in [21, 22].

Firstly, the kinetic energy and potential energy equation of the quadrotor-slung load system is obtained. The kinetic energy of the system T' can be divided into two components, namely, kinetic energy for translational motion T'_{trans} and kinetic energy for rotational motion T'_{rot} [38].

$$T'_{\text{trans}} = \underbrace{\frac{1}{2}m_Q\dot{x}_Q^2 + \frac{1}{2}m_Q\dot{y}_Q^2 + \frac{1}{2}m_Q\dot{z}_Q^2}_{\text{Quadrotor}} + \underbrace{\frac{1}{2}m_L\dot{x}_L^2 + \frac{1}{2}m_L\dot{y}_L^2 + \frac{1}{2}m_L\dot{z}_L^2}_{\text{Slung load}}$$

$$T'_{\text{rot}} = \frac{1}{2}I_{xx}\omega_x^2 + \frac{1}{2}I_{yy}\omega_y^2 + \frac{1}{2}I_{zz}\omega_z^2$$

Written in a compact manner

$$T' = \frac{1}{2}\dot{\mathbf{x}}_Q^T M_Q \dot{\mathbf{x}}_Q + \frac{1}{2}\dot{\mathbf{x}}_L^T M_L \dot{\mathbf{x}}_L + \frac{1}{2}\boldsymbol{\omega}^T \mathcal{I}\boldsymbol{\omega}$$

where $M_Q = \text{diag}(m_Q, m_Q, m_Q)$, $M_L = \text{diag}(m_L, m_L, m_L)$ and $\boldsymbol{\omega} = \boldsymbol{\omega}^{\boldsymbol{B}}$. Using the Jacobian matrix obtained in the previous section, the kinetic energy $T(\dot{\mathbf{q}}, \mathbf{q})$ can be written in terms of generalized coordinates as

$$T(\dot{\mathbf{q}}, \mathbf{q}) = \frac{1}{2}\dot{\mathbf{q}}^T J(\mathbf{q})^T M J(\mathbf{q})\dot{\mathbf{q}} \tag{2-20}$$

with

$$M = \begin{bmatrix} M_Q & 0 & 0 \\ 0 & M_L & 0 \\ 0 & 0 & \mathcal{I} \end{bmatrix}$$

The potential energy $V(\mathbf{q})$ of the system is then given as

$$V(\mathbf{q}) = m_Q g z_Q + m_L g \left(z_Q - l\cos\phi_L \cos\theta_L\right) \tag{2-21}$$

Having obtained the expressions for kinetic and potential energy, the Lagrangian $\mathcal{L}(\dot{\mathbf{q}}, \mathbf{q}) = T(\dot{\mathbf{q}}, \mathbf{q}) - V(\mathbf{q})$ is formulated. Solving the well known result of Euler-Lagrange equation given in Eq. (2-22), dynamics of the quadrotor-slung load system written in generic form in (2-23) is obtained

$$\frac{d}{dt}\left[\frac{\partial \mathcal{L}}{\partial \dot{q}}\right] - \frac{\partial \mathcal{L}}{\partial q} = \mathbf{f}_{\text{ext}} \tag{2-22}$$

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + G(\mathbf{q}) = \mathbf{f}_{\text{ext}} \tag{2-23}$$

where $M(\mathbf{q})$ is the mass matrix, $C(\mathbf{q}, \dot{\mathbf{q}})$ is the coriolis matrix and $G(\mathbf{q})$ captures the effects of gravity. The nonlinear equations of motion of the quadrotor-slung load system are given as

$$\begin{aligned} \ddot{\mathbf{q}} &= M^{-1}(\mathbf{q})\left(\mathbf{f}_{\text{ext}} - C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - G(\mathbf{q})\right) \\ &= \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{f}_{\text{ext}}) \end{aligned} \tag{2-24}$$

The control inputs $\mathbf{f}_{\text{ext}} = \begin{bmatrix} \mathbf{f}^{\mathbf{W}^T} & 0 & 0 & \boldsymbol{\tau}^{\boldsymbol{W}^T} \end{bmatrix}^T$, where $\mathbf{f}^{\mathbf{W}}$ is the control input given to the quadrotor as given in Eq. (2-12) and $\boldsymbol{\tau}^{\boldsymbol{W}}$ is the control torque provided to the attitude dynamics of the quadrotor expressed in the world frame.

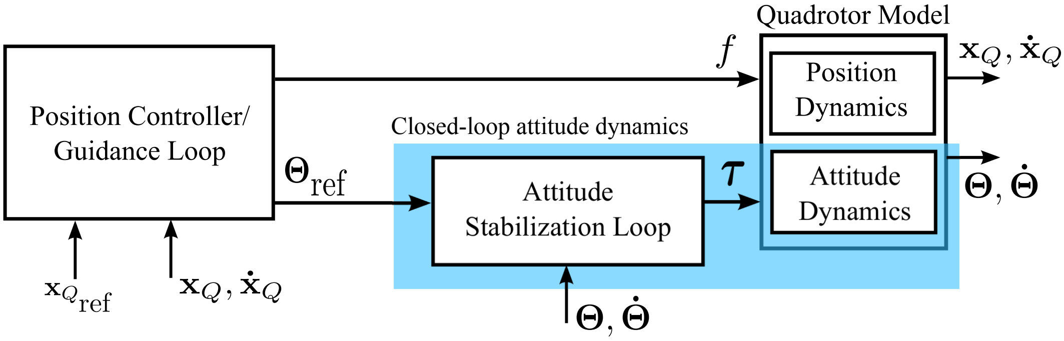

Quadrotor Model
Position Controller/ Guidance Loop
f
Position Dynamics
$\mathbf{x}_Q, \dot{\mathbf{x}}_Q$
Closed-loop attitude dynamics
$\boldsymbol{\Theta}_{\text{ref}}$
Attitude Stabilization Loop
$\boldsymbol{\tau}$
Attitude Dynamics
$\boldsymbol{\Theta}, \dot{\boldsymbol{\Theta}}$
$\mathbf{x}_{Q_{\text{ref}}}$
$\mathbf{x}_Q, \dot{\mathbf{x}}_Q$
$\boldsymbol{\Theta}, \dot{\boldsymbol{\Theta}}$

Figure 2-3: Cascaded controller structure for quadrotor

One important observation that must be pointed out here is that the attitude dynamics derived in this section, is expressed in the world frame unlike the attitude dynamics of Section 2-1. Similar to the dynamics of the quadrotor, the dynamics of the quadrotor-slung load system is decoupled from the attitude dynamics. This decoupled structure motivates the control design strategy in this work. So basically, under the assumption that the attitude of the quadrotor is perfectly controlled, the quadrotor-slung load system can be seen as a 3D crane or a Cartesian manipulator with a slung mass (spherical pendulum).

2-3 Closed loop attitude dynamics

In the previous section, it was noted that the translational and rotational dynamics of the quadrotor is decoupled and that this structure would be exploited to choose an appropriate control strategy. The decoupled dynamics of the quadrotor allows, the design of the cascaded, decoupled controllers for position and attitude control as shown in Figure 2-3. The discussion in this section is centered around the attitude dynamics and the inner loop controller, more aptly called as the attitude stabilization loop. Quadrotor being a highly nonlinear and unstable system, the main objective of the inner loop controller is to stabilize the system and the outer loop controller then controls this stable system. Hence the performance of the outer loop controller, which is also the subject of investigation in this thesis work mainly depends on the performance of the inner loop controller.

The problem of attitude stabilization is equivalent to the problem of controlling the roll, pitch and yaw angles of the quadrotor. The main objectives of the attitude stabilization loop are the following

- Accurately track the desired attitude references provided by the outer loop controller with a fast response time (high bandwidth).
- Reject the disturbances acting on the quadrotor attitude dynamics.

Ideally, it is preferable if the inner loop controller can track the attitude references perfectly i.e., $\boldsymbol{\Theta}_{\text{ref}} = \boldsymbol{\Theta}$. Consequently, the control design task is greatly simplified as only the translational dynamics needs to be considered. However, this assumption is never true in practice and it is important to model the dynamics of the inner loop controller along with the attitude dynamics of the quadrotor. The resulting model can then be used with the position dynamics of the quadrotor for better predictions.

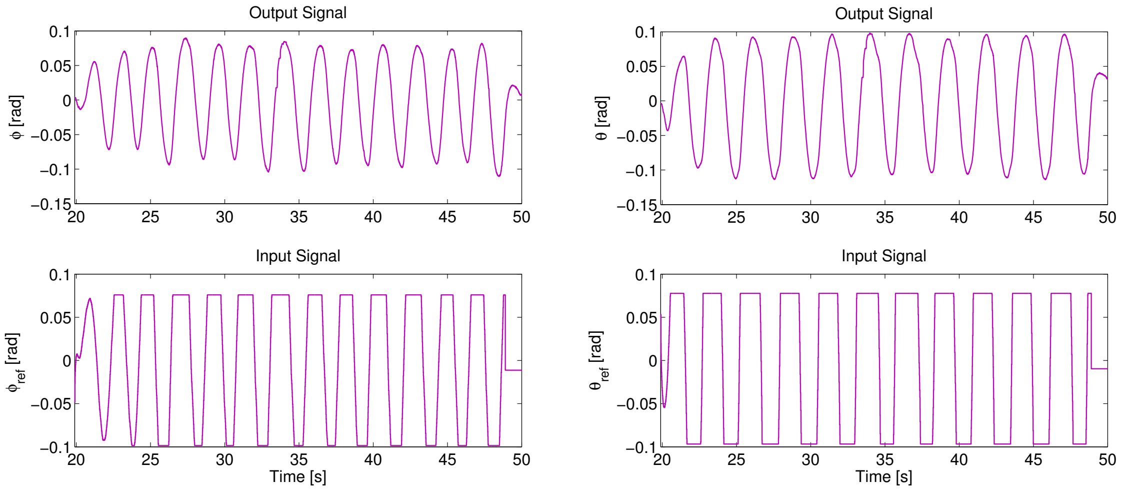
Since the focus point of this thesis is to design the outer loop controller, the design and implementation of the attitude controller is out of scope of this work. Hence, the Proportional-Integral-Derivative with Feed Forward (PID-FF) controller available within the Paparazzi open source autopilot software has been used as the inner loop attitude stabilization controller. The attitude stabilization problem of the quadrotor is a solved problem and many control methods exist in the literature. See for example [39] for a treatment of PID controller for attitude stabilization and [40] for an attitude stabilization controller in $\mathcal{SO}(3)$. The PID controller for stabilization of attitude of the quadrotor, basically consists of three decoupled Single Input Single Output (SISO) PID control loops controlling the roll, pitch and the yaw angles respectively. Hence there exist two ways to model the closed loop attitude dynamics of the quadrotor (controller + attitude dynamics).

First, given that the attitude dynamics is accurately modeled and that the equations of the PID controllers are available at our disposal, a simplified equation could be obtained for the closed loop system. However, this is not practically possible as we do not have accurate knowledge of the attitude dynamics, actuator dynamics and the disturbances acting on them. Second alternative is to resort to the system identification of closed loop attitude dynamics. The method is practically viable and can capture the dominant dynamics to a sufficient accuracy. As mentioned previously, the roll, pitch and yaw angles of the quadrotor are controlled separately using three PID controllers. Hence, owing to this structure, continuous time state space models are identified for the controlled roll and pitch subsystems. The dynamics of the yaw subsystem was not identified as a constant yaw angle is maintained throughout the experiments.

2-3-1 Identification Experiment

Prior to setting up an experiment to identify the roll and pitch dynamics, it is a wise idea to assess the performance of the currently implemented PID-FF controller. Step response of the controlled system could give crucial insights regarding its bandwidth and whether the "perfect tracking" assumptions are valid. It also aids in deciding the sampling frequency and the nature of the input signal required for performing system identification. Step response experiment was conducted for the quadrotor used in this thesis with PID-FF controller as the attitude stabilization loop in [41]. The PID-FF loop has a rise time of 0.32 seconds which gives a bandwidth of approximately 1.1 Hz accompanied with a steady state error. This gives two important insights regarding the attitude stabilization loop. Firstly, the bandwidth is low, which means that the "perfect tracking" assumptions are not valid and hence the dynamics needs to be modeled. Secondly, as a rule of thumb the sampling frequency must be selected 10 times the bandwidth.

An identification experiment was setup to identify the closed loop roll and pitch dynamics of the quadrotor. Input reference command (roll reference command ϕ_{ref} [rad] and pitch reference command θ_{ref} [rad]) was communicated to the drone over Wi-Fi at 100 Hz ($>$ 10 times the bandwidth). The resulting roll ϕ and pitch θ angles were obtained from the telemetry data of the drone. Using the recorded data, state space models of the following



(a) Roll subsystem

(b) Pitch subsystem

Figure 2-4: Input Output identification data

form were identified using the System Identification toolbox of Matlab.

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) + Ke(t) \\ y(t) &= Cx(t) + e(t)\end{aligned} \tag{2-25}$$

where $x(t) \in \mathbb{R}^n$ are the states, n is the order of the model, $u(t) \in \mathbb{R}^m$ are the inputs, $y(t) \in \mathbb{R}^r$ are the outputs. The signal $e(t) \in \mathbb{R}^r$ is the zero mean white noise also called innovation signal with statistical property $e(t) \sim \mathcal{N}(0, \Sigma_e^2)$.

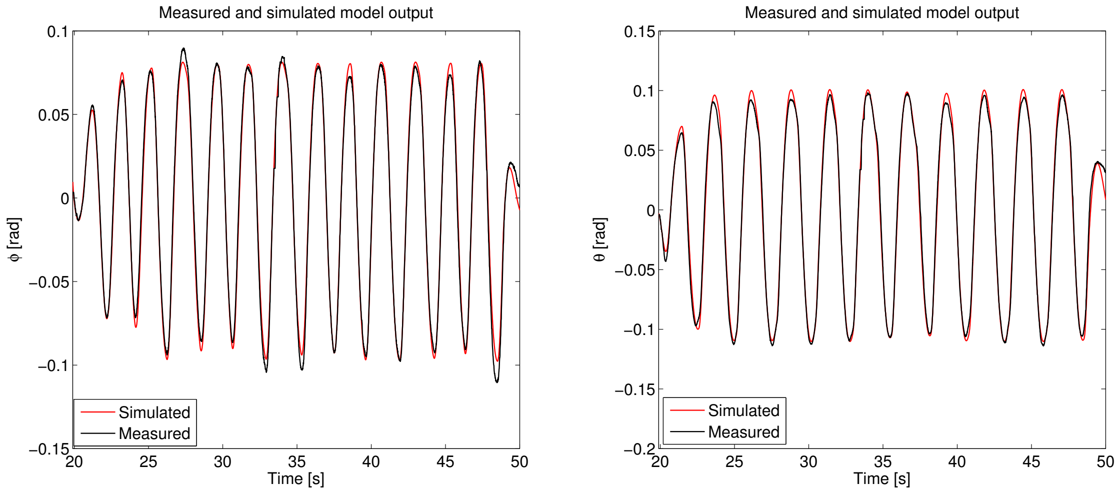
2-3-2 Identification of Roll and Pitch dynamics

Figure 2-4 shows the plot of input and output signal used for identification of roll and pitch dynamics. The input step like sequence is rich in frequency content and was considered suitable for identification. As with any standard identification process, means and trends were removed from the input signal as the Linear Time Invariant (LTI) model cannot capture these effects within the model.

A second order state space model was then obtained with a fit of 91.44 % and 89.9 % for the roll and pitch identification dataset of Figure 2-4 respectively. The dynamics of the identified roll subsystem is given in Eq. (2-26) followed by the dynamics of pitch subsystem in Eq. (2-27).

$$\begin{bmatrix} \dot{x}_{\phi_1}(t) \\ \dot{x}_{\phi_2}(t) \end{bmatrix} = \begin{bmatrix} -1.909 & 1.309 \\ -6.528 & -1.484 \end{bmatrix} \begin{bmatrix} x_{\phi_1}(t) \\ x_{\phi_2}(t) \end{bmatrix} + \begin{bmatrix} -0.5616 \\ -2.729 \end{bmatrix} \phi_{\text{ref}}(t) + \begin{bmatrix} -39.36 \\ -179.7 \end{bmatrix} e_{\phi}(t) \tag{2-26}$$

$$\phi(t) = \begin{bmatrix} -1.874 & -0.007115 \end{bmatrix} \begin{bmatrix} x_{\phi_1}(t) \\ x_{\phi_2}(t) \end{bmatrix} + e_{\phi}(t)$$



(a) Roll subsystem

(b) Pitch subsystem

Figure 2-5: Plot of measured and simulated model ouput for roll and pitch subsystem with identification dataset

$$\begin{bmatrix} \dot{x}_{\theta_1}(t) \\ \dot{x}_{\theta_2}(t) \end{bmatrix} = \begin{bmatrix} -2.188 & 1.394 \\ -5.690 & 0.254 \end{bmatrix} \begin{bmatrix} x_{\theta_1}(t) \\ x_{\theta_2}(t) \end{bmatrix} + \begin{bmatrix} -0.4533 \\ -1.2230 \end{bmatrix} \theta_{\text{ref}}(t) + \begin{bmatrix} -38.64 \\ -120.60 \end{bmatrix} e_\theta(t) \tag{2-27}$$

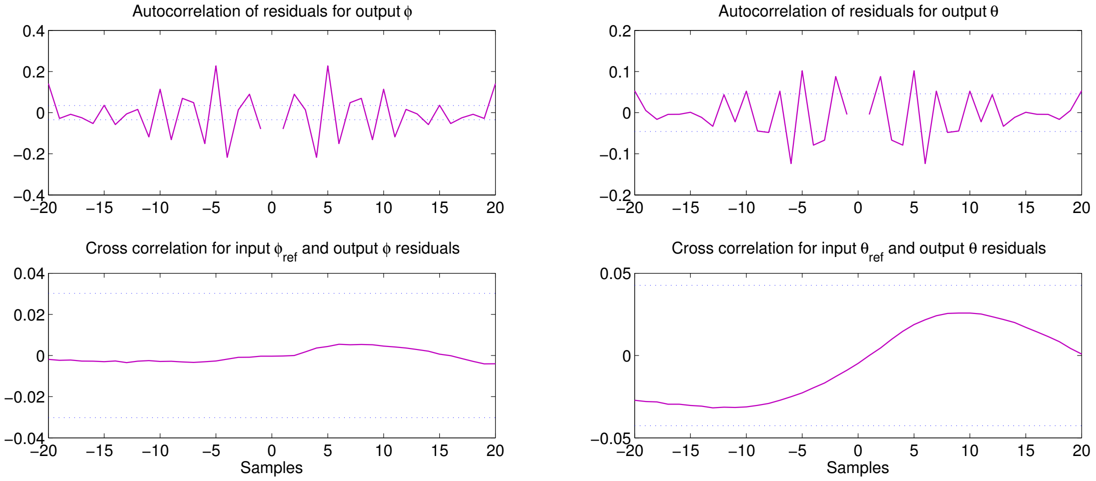
$$\theta(t) = \begin{bmatrix} -2.564 & -0.01674 \end{bmatrix} \begin{bmatrix} x_{\theta_1}(t) \\ x_{\theta_2}(t) \end{bmatrix} + e_\theta(t)$$

The above equations written in compact form are given below

$$\begin{aligned} \dot{\mathbf{x}}_\phi(t) &= A_\phi \mathbf{x}_\phi(t) + B_\phi \phi_{\text{ref}}(t) + K_\phi e_\phi(t) \\ \phi(t) &= C_\phi \mathbf{x}_\phi(t) + e_\phi(t) \\ \dot{\mathbf{x}}_\theta(t) &= A_\theta \mathbf{x}_\theta(t) + B_\theta \theta_{\text{ref}}(t) + K_\theta e_\theta(t) \\ \theta(t) &= C_\theta \mathbf{x}_\theta(t) + e_\theta(t) \end{aligned}$$

Figure 2-5 shows the plot of measured and simulated model. As can be seen from the figure, the model is able to capture the dynamics of the roll and pitch angle very accurately. The results of autocorrelation and cross correlation test for the model residuals is shown in Figure 2-6. Ideally the correlation values must be within the bounds (dotted blue lines) which is not the case. However, having obtained a good fit, the identified models were used for control design.

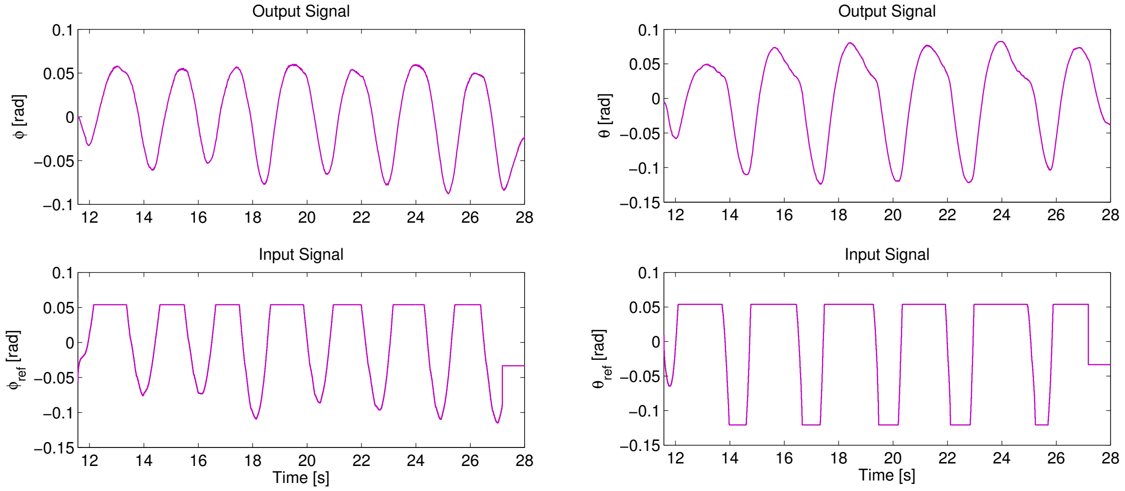
The fitness of the model was further validated with a validation dataset obtained from a different experiment. Figure 2-7 shows the plot of input - output data used as validation dataset. Figure 2-8 shows the plot of measured and simulated output for validation dataset. From the figure it can be seen that the model provides a good fit for validation data.



(a) Roll subsystem

(b) Pitch subsystem

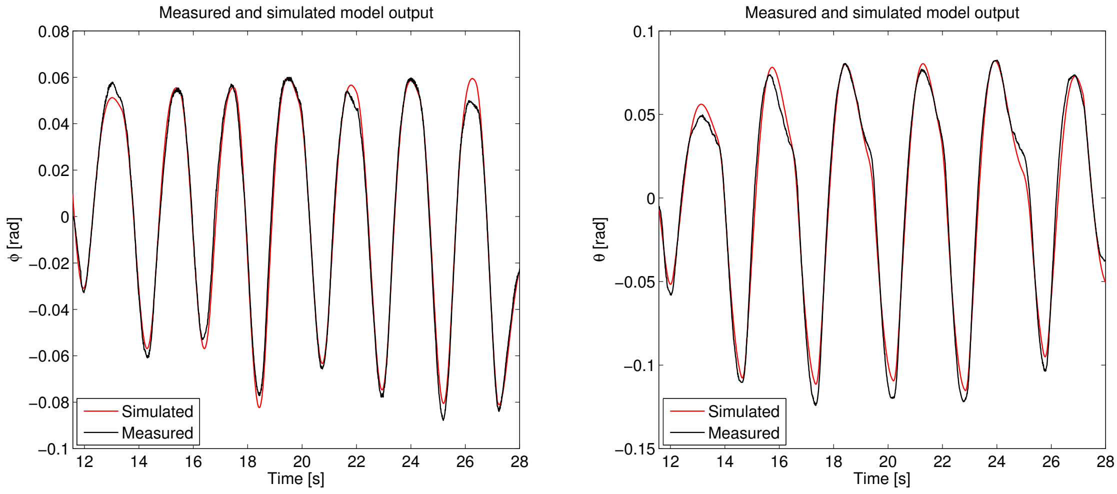
Figure 2-6: Auto correlation and cross correlation test results for roll and pitch subsystem



(a) Roll subsystem

(b) Pitch subsystem

Figure 2-7: Input output validation data



(a) Roll subsystem

(b) Pitch subsystem

Figure 2-8: Plot of measured and simulated model ouput for roll and pitch subsystem with validation dataset

2-3-3 Equations of motion with Identified closed-loop dynamics

Now that we have a model for the closed-loop attitude dynamics, the equations of motion for quadrotor and quadrotor-slung load system derived in Section 2-1 and Section 2-2 respectively are updated. The resulting model is partly identified and partly obtained from the first principles. Specifically, the translational dynamics of the quadrotor are described by the first principles model where as the controlled attitude dynamics is governed by the identified model. The quadrotor model of Eq. (2-13) updated with the attitude dynamics can now be written as given in Eq. (2-28). The time dependence of the variables are not mentioned for the sake of brevity.

$$\begin{aligned}
\ddot{x}_Q &= \frac{f}{m_Q}\left(C_\psi S_\theta C_\phi + S_\psi S_\phi\right) \\
\ddot{y}_Q &= \frac{f}{m_Q}\left(S_\psi S_\theta C_\phi - C_\psi S_\phi\right) \\
\ddot{z}_Q &= \frac{f}{m_Q}\left(C_\theta C_\phi\right) - g \\
\dot{\mathbf{x}}_\phi &= A_\phi \mathbf{x}_\phi + B_\phi \phi_{\text{ref}} \\
\dot{\mathbf{x}}_\theta &= A_\theta \mathbf{x}_\theta + B_\theta \theta_{\text{ref}} \\
\phi &= C_\phi \mathbf{x}_\phi \\
\theta &= C_\theta \mathbf{x}_\theta
\end{aligned} \tag{2-28}$$

Similarly, the attitude dynamics of the quadrotor-slung load system is replaced by the identified closed-loop dynamics.

$$\begin{aligned}
\ddot{x}_Q &= h_1(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{f}_{\text{ext}}) \\
\ddot{y}_Q &= h_2(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{f}_{\text{ext}}) \\
\ddot{z}_Q &= h_3(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{f}_{\text{ext}}) \\
\ddot{\theta}_L &= h_4(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{f}_{\text{ext}}) \\
\ddot{\phi}_L &= h_5(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{f}_{\text{ext}}) \\
\dot{\mathbf{x}}_\phi &= A_\phi \mathbf{x}_\phi + B_\phi \phi_{\text{ref}} \\
\dot{\mathbf{x}}_\theta &= A_\theta \mathbf{x}_\theta + B_\theta \theta_{\text{ref}} \\
\phi &= C_\phi \mathbf{x}_\phi \\
\theta &= C_\theta \mathbf{x}_\theta
\end{aligned} \tag{2-29}$$

where $h_i \quad \forall i = 1, \cdots, 5$ are the first five rows of Eq. (2-24). The equations are very complex and hence are not reproduced here.

2-4 Linearization and discretization of the model

The nonlinear models obtained must be linearized around the equilibrium/hover conditions in order to be able to design linear controllers. Clearly, the control inputs $\mathbf{u}$ to the quadrotor and quadrotor-slung load model of Eq. (2-28) and Eq. (2-29) are the thrust f [N], the roll angle reference ϕ_{ref} [rad] and the pitch angle reference θ_{ref} [rad]. The heading angle is kept constant $\psi = \psi_0 = 0$, as it does not hamper the motion of quadrotor in 3D Euclidean space. The state space $\boldsymbol{\xi} \in \mathbb{R}^{14}$ and the input $\mathbf{u} \in \mathbb{R}^3$ of the quadrotor-slung load system is given as

$$\boldsymbol{\xi} = \begin{bmatrix} x_Q & y_Q & z_Q & \theta_L & \phi_L & \dot{x}_Q & \dot{y}_Q & \dot{z}_Q & \dot{\theta}_L & \dot{\phi}_L & x_{\phi_1} & x_{\phi_2} & x_{\theta_1} & x_{\theta_2} \end{bmatrix}^T$$

$$\mathbf{u} = \begin{bmatrix} f & \phi_{\text{ref}} & \theta_{\text{ref}} \end{bmatrix}^T$$

The equilibrium conditions for input at which the quadrotor-slung load system can hover at a particular position (x, y, z) is $f = (m + m_L)g$, $\phi_{\text{ref}} = \theta_{\text{ref}} = 0$. Clearly, at equilibirum condition the states of the quadrotor-slung load system are

$$\boldsymbol{\xi}_{\text{eq}} = \begin{bmatrix} x_Q & y_Q & z_Q & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

$$\mathbf{u}_{\text{eq}} = \begin{bmatrix} (m + m_L)g & 0 & 0 \end{bmatrix}^T$$

A continuous time Linear Time Invariant (LTI) model is obtained by approximating the nonlinear dynamics of Eq. (2-29) using first order taylor approximation around the equilibrium point $(\boldsymbol{\xi}_{\text{eq}}, \mathbf{u}_{\text{eq}})$. The resulting continuous time LTI model is given in Eq. (2-30). The subscript c denotes that the system matrices are in continuous time domain.

$$\begin{aligned}
\dot{\boldsymbol{\xi}}(t) &= A_{\xi,c}\boldsymbol{\xi}(t) + B_{\xi,c}\mathbf{u}(t) \\
\mathbf{y}(t) &= C_\xi \boldsymbol{\xi}(t)
\end{aligned} \tag{2-30}$$

In the above equation $y \in \mathbb{R}^7$ is the vector of measured outputs given as

$$\mathbf{y} = \begin{bmatrix} x_Q & y_Q & z_Q & \theta_L & \phi_L & \phi & \theta \end{bmatrix}^T$$

The position of the quadrotor and the slung load is measured using external tracking cameras, whereas the measured roll and pitch angles are obtained from telemetry data from the drone. Hence the matrix $C_\xi \in \mathbb{R}^{7\times 14}$ is

$$C_\xi = \begin{bmatrix} I_{5\times 5} & 0_{5\times 5} & 0_{5\times 2} & 0_{5\times 2} \\ 0_{1\times 5} & 0_{1\times 5} & C_\phi & 0_{1\times 2} \\ 0_{1\times 5} & 0_{1\times 5} & 0_{1\times 2} & C_\theta \end{bmatrix}$$

The system matrices $A_{\xi,c} \in \mathbb{R}^{14\times 14}$ and $B_{\xi,c} \in \mathbb{R}^{14\times 3}$ are not produced here due to the large size of the matrices. However, one important observation is the eigenvalues of the matrix $A_{\xi,c}$. The eigenvalues are $\lambda = \{0, 0 \pm \mathrm{i}3.1024, -1.6965 \pm \mathrm{i}2.9155, -0.9668 \pm \mathrm{i}2.5378\}$. The eigenvalue 0 has a geometric multiplicity of 3 and algebraic multiplicity of 6, these eigenvalues correspond to the position dynamics of quadrotor. The complex conjugate eigenvalue pair corresponding the slung load is $0 \pm \mathrm{i}3.1024$ with geometric and algebraic multiplicity of 2. The complex conjugate eigenvalue pair $-1.6965 \pm \mathrm{i}2.9155$ and $-0.9668 \pm \mathrm{i}2.5378$ correspond to eigenvalues of roll and pitch subsystem with geometric and algebraic multiplicity of 1. Most of the eigenvalues lie on the imaginary axis and hence the system is marginally stable. The controllers and estimators designed are implemented on a digital computer, hence the system matrices were discretized using zero order hold. Given the sampling period of T_s seconds, the discrete time system matrices are computed as

$$\begin{aligned} A_\xi &= I + T_s A_{\xi,c} \\ B_\xi &= T_s B_{\xi,c} \end{aligned}$$

The resulting discrete time LTI model of quadrotor-slung load system is given in Eq. (2-31) where subscript k denotes the time step and time is obtained as kT_s.

$$\begin{aligned} \boldsymbol{\xi}_{k+1} &= A_\xi \boldsymbol{\xi}_k + B_\xi \mathbf{u}_k \\ \mathbf{y}_k &= C_\xi \boldsymbol{\xi}_k \end{aligned} \tag{2-31}$$

The above model which will be used in Chapter 3 for design of controllers and state estimator is controllable and observable which indicates that it is possible to influence the swing angle of slung load using the available control inputs and the states can be reconstructed with the available measurements.

2-5 Model Validation

The dynamic model obtained for the quadrotor and the quadrotor-slung load system has to be validated with experimental data, in order to assess its usability for controller and state estimator design. However, due to the highly nonlinear and unstable nature of quadrotor dynamics, the experimental validation by gathering input-output data through open-loop experiments is a difficult or even impossible task. Same challenges apply to the validation of the quadrotor-slung load model.

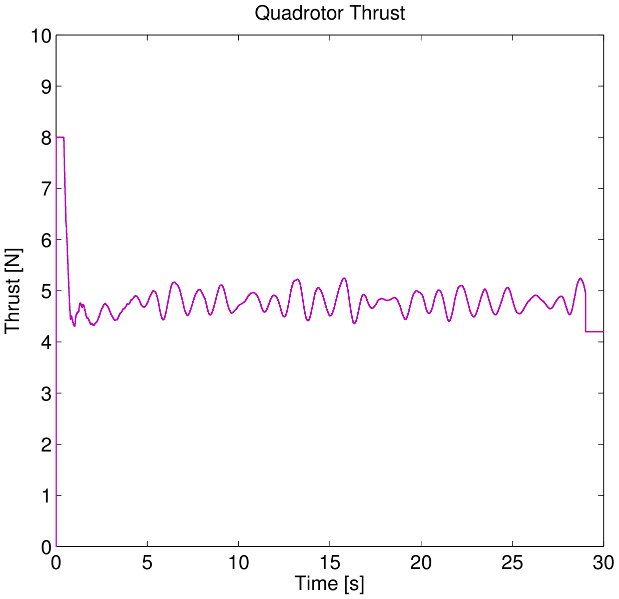


Figure 2-9: Thrust input applied to the quadrotor for model validation

Hence an experiment was conducted with a LQ controller with integral action in the loop which allowed the quadrotor to hover at a constant altitude. The roll and pitch commands of Figure 2-4 was then applied to the quadrotor. Resulting position and velocity of the quadrotor was measured using the external tracking camera system. In order to maintain the quadrotor at a constant altitude, LQ controller provided the thrust command given in Figure 2-9. The model used to obtain a LQ controller for position control was the translational dynamics of the quadrotor of Eq. (2-13) linearized around the equilibrium point with f, ϕ, and θ as the control inputs.

Given the attitude reference inputs of Figure 2-4 and thrust commands of Figure 2-9, Figure 2-10 shows the plot of measured quadrotor position and the position obtained by simulating the quadrotor model of Eq. (2-28). Similarly, Figure 2-11 shows the plot of measured and simulated quadrotor velocities. Clearly there are both the similarities and discrepancies between the measured and simulated quadrotor model. In order to clearly explain the plots, let us first understand the sources of uncertainty present in the system.

From the quadrotor model of Eq. (2-28), it can be seen that the quadrotor will hover at a constant altitude if applied thrust $f = m_Q g$ [N]. Similarly, assuming that the attitude stabilization loop accurately stabilizes the quadrotor attitude to the desired references, zero roll and pitch references would mean that the quadrotor would hover at a constant (x,y) position. This is an ideal case and never true in practice. Maintaining the quadrotor attitude at zero roll and pitch angles would cause the quadrotor to drift away. Hence a non zero (unknown) roll and pitch attitude references need to be provided (by the controller) so that the quadrotor does not drift away. Also an exact thrust of $m_Q g$ is not sufficient to hover at constant altitude. This uncertainty in the inputs are due to the following reasons.

- The quadrotor structure is not exactly symmetric and hence the assumption that the

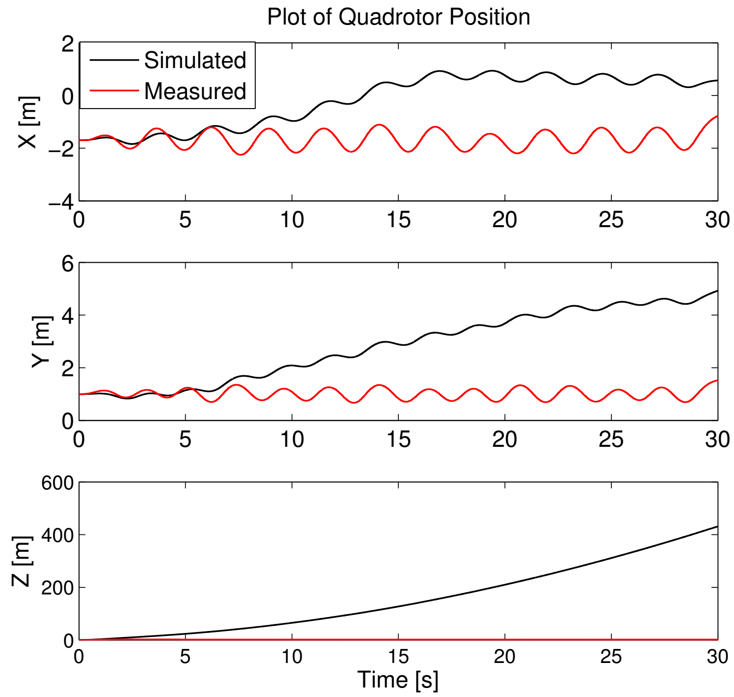


Figure 2-10: Plot of measured and simulated quadrotor position

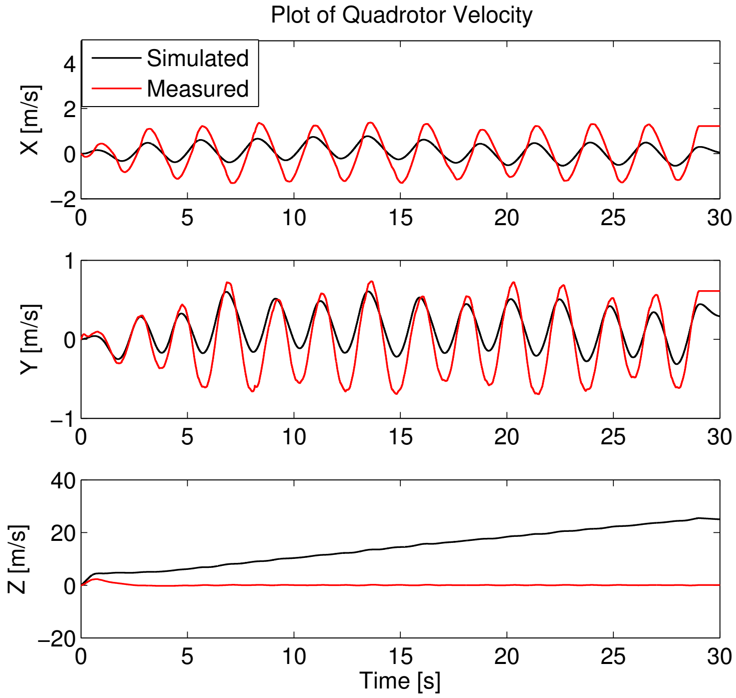


Figure 2-11: Plot of measured and simulated quadrotor velocity

center of mass of the quadrotor coincides with the geometric center of the quadrotor and hence the origin of the body fixed frame is no more valid.

- All four rotors of the quadrotor are not identical and the uncertainties in the lower level motor controllers influence the performance of higher level controllers.
- Attitude estimation errors also play an important role. When the drone is placed flat on the ground, the AHRS estimates some non zero attitude. Clearly 'flat' itself is a relative term. Also note that the inertial sensors are not perfectly aligned with the body of the quadrotor.
- As explained in Section 4-2-2 of Chapter 4, the thrust command applied to the quadrotor is a unit less value. Whereas the controllers compute thrust inputs in Newton and hence a static map is used to convert the computed thrust into the drone compatible value which adds uncertainty of its own. Such effects and uncertainties are mitigated by the use of integral action within the control loop.

All these uncertainties combined together warrant a need to use a controller to make a quadrotor hover at a constant position in 3D space. This also explains why it is difficult to identify the dynamics of the quadrotor in open-loop. The above points can be proved through experimental plots provided in Chapter 4, where one can notice that the attitude reference values are non-zero when quadrotor is hovering at a particular position.

The above practical issues have the following consequences when applied to the quadrotor model of Eq. (2-28).

- From the model it can be seen that the quadrotor inputs act on the accelerations of the quadrotor. The attitude references provided by the controller to make the quadrotor hover at a constant position, has a non zero mean value. When such an input is applied to the model of the quadrotor, the effect is that the simulated velocity drifts linearly from the measured value. Hence the mean values were deducted from the attitude references and applied to the quadrotor model of Eq. (2-28). The resulting plot of quadrotor velocity is shown in Figure 2-11. Clearly the trend in the measured velocity matches with the simulated velocity in X and Y axes. The discrepancies in the simulated and measured quadrotor velocity causes a drift in simulated position as can be seen in Figure 2-10.
- The thrust inputs produced by the LQ controller with integral action is able to handle the uncertainty in mass of the quadrotor and applied thrust inputs. As a result the quadrotor hovers at a constant altitude. These uncertainties are not captured by the model of the quadrotor and hence when the applied thrust inputs are used to simulate the model, it can be noticed that the velocity in z-axis drifts linearly. Consequently, the simulated position diverges quadratically as can be seen in Figure 2-10.

For the reasons mentioned above, the identification and validation of the quadrotor model is difficult and efforts could be directed towards obtaining a better model as part of future work. Owing to the same reasons, the validation of the quadrotor-slung load system could not be done. The approach adopted in this thesis was to use the obtained models for control design and revisit the modeling phase if the control performance is not satisfactory.

2-6 Conclusion

In this chapter the modeling aspects of the quadrotor-slung load system were discussed. The dynamic models were obtained using the first principles and assumptions made to obtain those models were presented. The most crucial assumption made to obtain a simpler model is that the cable is considered as rigid massless link and that the suspension point of the slung payload is coincident with the center of mass of the quadrotor (and hence the origin of body fixed frame). What remains to be answered in the following chapters is, "How valid are these assumptions? What are its consequences on control design and performance? What are the limitations of making such an assumption?"

It was shown that the structure within the model could be exploited to design two cascaded controllers for attitude and position control respectively. Since the already existing PID-FF controller in the Paparazzi autopilot software will be used for attitude control, its performance such as rise time and hence the bandwidth was assessed motivating the need to identify the closed loop attitude dynamics. A second order state space model was identified for the roll and pitch subsystem and validation results were presented. The final model which is partly identified and partly obtained from the first principles was linearized to obtain a LTI model which will be used for controller and estimator design in the next chapter. It was found that the resulting linear model is controllable and observable and hence it is possible to influence the slung load using the control inputs. Finally, the nonlinear model was validated against the measured data and discrepancies in the model were explained. It was found that the uncertainties within the real system violates the modeling assumptions which causes the differences between the simulated and measured model. The specific sources for these uncertainties were identified. It was shown that system identification and validation of a unstable system such as quadrotor-slung load system is difficult to perform in open-loop and a controller must be used to gather some useful data. Clearly, a better model can be obtained and considerable efforts must be directed towards it as a part of future work.

Chapter 3

Model based Control Design

In the previous chapter, a dynamic model was developed for the quadrotor-slung load system. The modeling assumptions and limitations of using such a model was outlined. In this chapter, the focus shifts to the control design problem. In particular, the linearized model obtained in Section 2-4 is used for model based controller design. Since the designed controllers are to be implemented on an experimental setup, particular attention is given to the techniques used to overcome these limitations or model mismatch and achieve real-time implementation. The controllers designed in this chapter are supported with simulation results. The nonlinear dynamic model obtained in Section 2-2 is used as a plant for simulation purposes and hence does not capture the model mismatch and uncertainties that would be present in the real system. The simulation results act mainly as a proof of concept, so that the appropriate control strategies could be validated in the experimental setup.

The control scheme is presented in Section 3-1 which describes the methodology of the control design and steps that need to be followed in order to have a working controller on a real system. The choice of sampling frequency selected for implementation is motivated. Section 3-2 describes briefly the well known Kalman filter along with the practical considerations and results. With a state estimator being designed, the focus shifts to the control design problem considering that the state information is available. Section 3-3 presents the Linear Quadratic (LQ) control design problem which forms the benchmarking controller followed by various Linear Time Invariant MPC formulations used to control the quadrotor-slung load system in Section 3-4. The advantages and drawbacks of each formulation are explained. Section 3-5 gives an overview of the state-of-the-art Quadratic Programming (QP) solvers used for real-time implementation of the MPC. Tuning and stability aspects of the MPC are discussed in Section 3-6 and Section 3-7 respectively. The chapter concludes with discussion of simulation results in Section 3-8.

3-1 Control Scheme

As mentioned in Section 2-4, the only measurements that are available are the position of the quadrotor and the slung load obtained from the external tracking cameras. The attitude

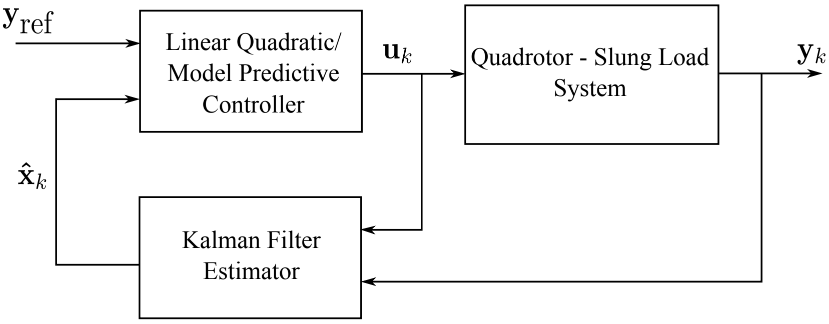

$\mathbf{y}_{\text{ref}}$
Linear Quadratic/ Model Predictive Controller
$\mathbf{u}_k$
Quadrotor - Slung Load System
$\mathbf{y}_k$
$\hat{\mathbf{x}}_k$
Kalman Filter Estimator

Figure 3-1: Block diagram of the control scheme

information is obtained from the drone's telemetry data. Consequently, all the states are not measured and need to be estimated. As a result, the control design problem breaks down into two subproblems. First, design of the controller assuming complete knowledge of the states. Second, state estimation problem given the applied control inputs, measured outputs and the plant model. Collectively, the aim is to design an output feedback controller. Figure 3-1 shows the block diagram of the control problem. The control design problem and the state estimation problem are dual problems and by invoking the separation principle, both the problems can be solved separately.

Selection of Sampling Frequency

The first and foremost decision that needs to be made is to select the sampling frequency. For Single Input Single Output (SISO) systems, the rule of thumb is to sample at least 8-10 times within the rise time for step response of the system. However, no such rule of thumb exists for MIMO systems. Ideally, it would be preferred to execute the controller as fast as possible although it is not always possible in practice. All the operations such as computation of control input, acquiring measurements, state estimation, data logging etc need to be accomplished within a single sample step. In Section 2-3, an identification experiment was performed with a sampling frequency of 100 Hz. The same frequency is used as the sampling frequency for controller design and implementation. The experimental system itself can provide intuitive insights for selection of appropriate sampling frequency. Additional insights obtained from the system, due to which a sampling frequency of 100 Hz is selected are as follows

1. The attitude stabilization controller which is executed on the drone runs at 500 Hz. Hence it is preferable to select a sampling frequency which is an integer multiple of 500 Hz.

2. 500 Hz sampling frequency is one suitable sampling frequency. However, computation of a MPC input is an expensive operation since it involves solving an optimization problem online. With the state-of-the-art tools and the problem dimension, it is not possible to compute the control inputs at this frequency. Hence, a sampling frequency of 100 Hz is chosen as a starting point. It could however be changed depending on the MPC problem formulation and the time it takes to solve the resulting optimization problem.

The following sections address the state estimation and control problem for the quadrotor-slung load system using the discrete time LTI model of Eq. (2-31) reproduced below

$$\begin{aligned}\boldsymbol{\xi}_{k+1} &= A_\xi \boldsymbol{\xi}_k + B_\xi \mathbf{u}_k \\ \mathbf{y}_k &= C_\xi \boldsymbol{\xi}_k\end{aligned} \tag{3-1}$$

Control Objectives

The control objectives for the work in this thesis are as follows

1. Design different MPC formulations based on LTI model of the quadrotor-slung load system.
2. The primary objective is transport the slung load from one point to another in swing free manner using the quadrotor. While this must be possible in simulation, it cannot be expected in practical setup. Hence in case of experimental validation, the swing angle of the slung load can be relaxed to lie within ± 10 degrees.
3. Since the model used is LTI, it is expected that the system will be associated with slow motion. Hence rise time would not be a major concern, however it is desirable to reach the set point as fast as possible. This would be one of the research questions that would be investigated through different MPC formulations.
4. The quadrotor positions must be tracked with zero steady state error and hence controllers must deal with model mismatch.
5. Finally it is desired to have a maximum overshoot of 10% in quadrotor position during transport. It must be noted that the rise time, settling time and overshoot requirements presented here are secondary objectives.

3-2 State Estimation - The Kalman Filter

As remarked before, the position of the quadrotor and the slung load along with the quadrotor attitude are the measured outputs. Hence, it is necessary to employ a method to reconstruct the velocity measurements from the known outputs, control inputs and knowledge of the plant in the form of a model. The position measurements obtained from the external tracking cameras are very accurate with position tracking error in the order of millimeters. The velocity estimates could be obtained by numerically differentiating the position signal. This must however be avoided as numerically differentiating a position signal also amplifies the noise. The result is therefore a noisy velocity signal. An alternative is to employ a low pass filter to filter the high frequency components. There is a downside to this solution, the low pass filter also introduces lag in the signal and it is advantageous to use a state estimator for the purpose.

The quadrotor-slung load system is a nonlinear system and the controller is developed using the linearized model of Eq. (2-31) assuming complete knowledge of the states of the system. State estimation being a dual problem, the very same model can be used to reconstruct the states which in turn would be used by the controller. Kalman filter is used for the state estimation purpose in this thesis work. A Kalman filter is an optimal estimator which 'filters' noisy/uncertain measurements obtained from a system governed by noisy LTI dynamics. The underlying assumption is that the process noise and the measurement noise have a normal distribution with zero mean.

Consider a discrete time LTI system described by the following equation

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k \\ y_k &= Cx_k + v_k \end{aligned} \tag{3-2}$$

where $w_k \sim \mathcal{N}(0, \Sigma_w)$, $v_k \sim \mathcal{N}(0, \Sigma_v)$ are the process and measurement noise respectively and $x_0 \sim \mathcal{N}(x_0, P_0)$ is the information about the initial state with mean x_0 and covariance P_0.

The Kalman filter problem can be divided into two phases namely, the time update step and the measurement update step

Time Update

In the time update step, at time k, given the initial state estimate $\hat{x}_{k-1} \sim \mathcal{N}(\hat{x}_{k-1}, P_{k-1})$, control input u_k along with the process covariance matrix Q and measurement covariance matrix R, the mean and covariance estimates of the states are projected ahead using the model of the system as given by Eq. (3-3). Note that the hat sign indicates that the quantity is an estimate of the true value. For example $\hat{x}$ is an estimate of the true value x.

$$\begin{aligned} \hat{x}_k^- &= A\hat{x}_{k-1} + Bu_k \\ P_k^- &= AP_{k-1}A^T + Q \end{aligned} \tag{3-3}$$

In this step, the prior knowledge of the system is used to predict the next state estimate given the control input. This quantity is not filtered as the measurement information is not incorporated into the estimate and hence the superscript '-' sign.

Measurement Update

In the measurement update step, the predicted state estimates are fused with the measurements to obtain filtered state estimates. The measurement update equations are given in Eq. (3-4).

$$\begin{aligned} K_k &= P_k^- C^T \left(CP_k^- C^T + R \right)^{-1} \\ \hat{x}_k &= \hat{x}_k^- + K_k \left(y_k - C\hat{x}_k^- \right) \\ P_k &= (I - K_k C) P_k^- \end{aligned} \tag{3-4}$$

The first equation in Eq. (3-4) denotes computation of Kalman gain which weighs the extent to which the innovation signal $(y_k - C\hat{x}_k^-)$ is accounted into the predicted state estimate $\hat{x}_k^-$ to obtain a filtered state estimate $\hat{x}_k$ (second equation in Eq. (3-4)). The last equation updates the covariance of the state estimates.

Kalman filter is a recursive estimator and updates the state estimates as and when the new measurements are available. It is the best possible linear estimator and easy to use resulting in wide usage. The only inputs to the filter are the initial state estimates $\hat{x}_0 \sim \mathcal{N}(\hat{x}_0, P_0)$, process and measurement noise covariance matrices Q and R

Dual rate Kalman Filter

It is possible that the rate at which the measurements are obtained is slower than the rate at which we would want the Kalman filter to produce the state estimates. The external tracking cameras provide a new measurement at the rate of 50 Hz, whereas the controller and the estimator are executed at 100 Hz. Hence from a practical view point a small modification was made to the measurement update equations as given below

$$\hat{x}_k = \hat{x}_k^- + aK_k\left(y_k - C\hat{x}_k^-\right) \tag{3-5}$$

where $a = \{0, 1\}$. $a = 0$ when there are no new measurements to update state estimate $\hat{x}_k$ and $a = 1$ when information from the measurements needs to be fused with the predicted estimate.

Tuning and Results

The tuning parameters in a Kalman filter are the process and measurement covariance matrices Q and R. They quantify the uncertainty in the process and the measurements. The inputs to the Kalman filter are the initial state estimate $\hat{x}_0$ and its covariance P_0. These parameters are however are not the tuning parameters and it is usual to set the covariance matrix P_0 to identity.

The process covariance matrix Q indicates noise acting on the system states. The noise acting on the system states are assumed to be independent resulting in Q being a diagonal matrix. The number of parameters to be tuned are hence reduced. Similarly, the matrix R indicates the measurement noise and is easy to tune as the covariance information can be extracted from the time series measurements. The position measurements are obtained from the external tracking cameras and the attitude of the quadrotor is obtained from the drone telemetry both of which are accurate with very low noise. Hence, R matrix can be set to very low values. This however will result in the filter assigning more weight to the measurements and discarding any predictions made by the model. Also the measurement noise will be transferred into the state estimates which is not desirable. Hence the covariance values were set higher than is actually the case.

$$R = \left[\begin{array}{ccccc|cc} 0.1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.01 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0.0001 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0001 \end{array}\right]$$

The upper block matrix corresponds to the measurement noise covariance for the quadrotor-slung load system whereas the lower block diagonal matrix corresponds to measurement noise covariance for the roll and pitch measurements. The roll and pitch attitude data which is obtained from the Attitude Heading and Reference System (AHRS) of the drone is already a filtered reading and hence the lower values of their covariances.

The process noise covariance matrix Q is generally tuned such that the diagonal entries have a larger value. This allows the filter to rely more on the measurements. However, larger value of diagonal entries also results in noisy state estimates. Hence, the values were increased as long as we have smooth state estimates. In the Section 2-5, it was shown that the input uncertainty causes a linear trend in the velocity. This uncertainty is captured by the larger covariance values for the velocity entries as can be seen in the upper block diagonal matrix of Q.

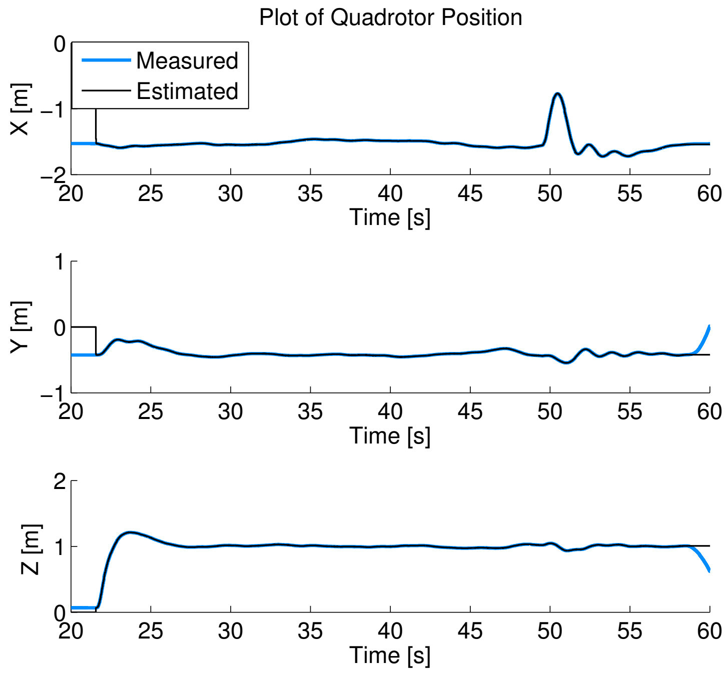
$$Q = \left[\begin{array}{cccccccccc|cccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 100 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 100 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 100 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.00001 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.00001 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.00001 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.00001
\end{array}\right]$$

Figure 3-2a shows the plot of quadrotor position indicating the performance of the Kalman filter for position estimation of the quadrotor. From the plot, it can be noticed that the measurement is very accurate as expected and the Kalman filter estimates perfectly follows the measurement. However, it is the Figure 3-2b which justifies the use of Kalman filter for velocity estimation. As can be seen, the velocity measurements are very jittery. They are basically obtained by numerically differentiating the position signal. Such a signal is not usable from control point of view. From the figure it can be seen that the Kalman filter is able to reconstruct the velocity signal with sufficient accuracy. The filter also estimates the states associated with the attitude dynamics (plots not shown to keep the treatment concise).

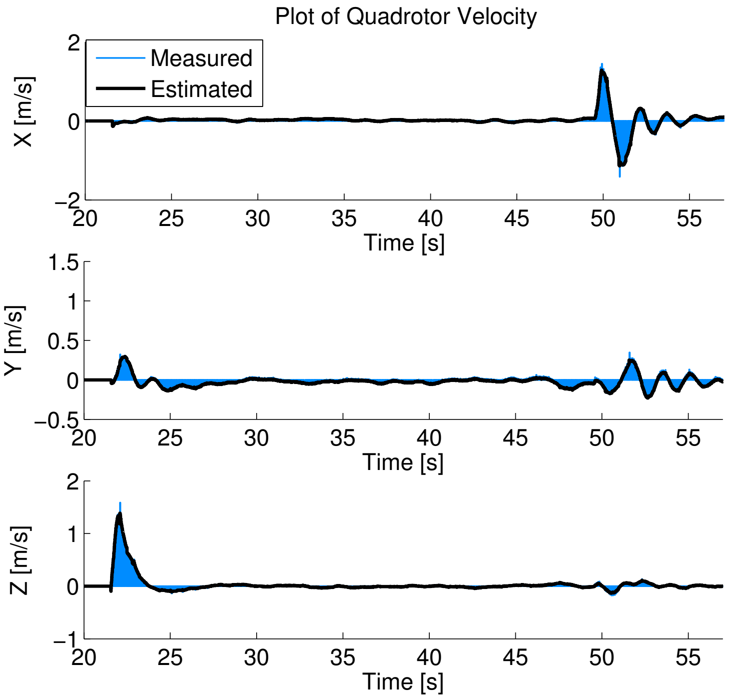
Having designed a Kalman filter for state estimation, the focus now shifts towards the control design problem for the rest of the chapter. Henceforth it is considered that the state estimates are available for design of feedback control laws.

3-3 Linear Quadratic Control with Integral Action

It is always desirable to have a benchmarking controller while investigating application of an advanced control technique for complicated nonlinear systems such as the quadrotor-slung



(a) Measured and estimated quadrotor position



(b) Measured and estimated quadrotor velocity

Figure 3-2: Results of Kalman filter state estimation

load system. Linear Quadratic (LQ) control is a very well known control method and forms the benchmarking controller for comparing the MPC formulations discussed in the later sections. The main advantage of using the LQ controller is that the computational requirements are negligible and the ease with which the controller can be tuned. Moreover, LQ controller can be considered as infinite horizon unconstrained MPC [42, 43] and hence it is logical to compare different finite horizon constrained MPC formulations with this unconstrained infinite horizon case.

In practice, there always exists mismatch between the model and the real system which results in steady state error at the output. The computation of the control input is dependent on the mass of the quadrotor and the slung load. While it is easy to measure the mass, even a slight mismatch would result in steady state error at the altitude. Moreover, the controller computes thrust command in Newton while the actual thrust input is a unit less number (referred to as Paparazzi units) which is eventually mapped to the individual motors PWM input. The mapping from the computed thrust input in Newton to applied thrust input in Paparazzi units is governed by a static map obtained from the measurements (See Section 4-2-2 for more details) and is prone to uncertainty since it is an approximation. Hence the result is steady state error in altitude. Similarly, uncertainties in the lower level motor controllers results in steady state error in X, Y position of the quadrotor.

In order to eliminate the steady state error, a standard technique used in the control literature is to introduce integral action into the system. Since the objective is to eliminate steady state error in the position of the quadrotor, integral action is added to the output position of the quadrotor by introducing integral states $\mathbf{v}_k \in \mathbb{R}^3$ with the following dynamics.

$$\mathbf{v}_{k+1} = \mathbf{v}_k + \begin{bmatrix} x_Q \\ y_Q \\ z_Q \end{bmatrix}$$

The state vector is then augmented with these integral states to obtain a augmented discrete time state space model as given in Eq. (3-6). The model is used with the Matlab command `dlqr` to obtain the static controller gain K.

$$\begin{bmatrix} \boldsymbol{\xi}_{k+1} \\ \mathbf{v}_{k+1} \end{bmatrix} = \begin{bmatrix} A_\xi & 0 \\ C_I C_\xi & I \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi}_k \\ \mathbf{v}_k \end{bmatrix} + \begin{bmatrix} B_\xi \\ 0 \end{bmatrix} \mathbf{u}_k \tag{3-6}$$

where matrix $C_I = \begin{bmatrix} I_{3\times 3} & 0_{3\times 4} \end{bmatrix}$ allows only the quadrotor positions to be integrated. The control input for regulation problem is then computed as

$$\mathbf{u}_k = -K \begin{bmatrix} \boldsymbol{\xi}_k \\ \mathbf{v}_k \end{bmatrix} \tag{3-7}$$

where $\mathbf{v}_k$ is computed as

$$\mathbf{v}_{k+1} = \mathbf{v}_k + \begin{bmatrix} x_Q \\ y_Q \\ z_Q \end{bmatrix} \tag{3-8}$$

The regulation problem is transformed into output tracking problem through computation of control input as follows

$$\mathbf{u}_k = -K \begin{bmatrix} \boldsymbol{\xi}_k - \boldsymbol{\xi}_{\text{ref}} \\ \mathbf{v}_k \end{bmatrix} \tag{3-9}$$

where $\boldsymbol{\xi}_{\text{ref}} = \left[\begin{array}{cccccccccccccc} x_{Q_{\text{ref}}} & y_{Q_{\text{ref}}} & z_{Q_{\text{ref}}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right]^T$ and $\mathbf{v}_k$ is computed as

$$\mathbf{v}_{k+1} = \mathbf{v}_k + \begin{bmatrix} x_Q - x_{Q_{\text{ref}}} \\ y_Q - y_{Q_{\text{ref}}} \\ z_Q - z_{Q_{\text{ref}}} \end{bmatrix} \tag{3-10}$$

The applied control input $\mathbf{u}_{\text{applied}}$ is the sum of computed control input $\mathbf{u}_k$ and the control input at equilibirum position $\mathbf{u}_{\text{eq}}$.

$$\mathbf{u}_{\text{applied}} = \mathbf{u}_k + \mathbf{u}_{\text{eq}}$$

The simulation results of LQ control with integral action is discussed in subsection 3-8-1.

3-4 Linear Time Invariant Model Predictive Control

In the previous section use of LQ control with integral action as a benchmarking controller was briefly discussed. One of the main drawbacks of the LQ control is that the control inputs can easily exceed the physical limitations. Hence, the control inputs must be bounded before being applied to the system. Such a way of restricting the control input externally could compromise with the stability of the system rendered by the LQ control. The control designer must ensure that the control inputs are within the physical limits through extensive tuning. This drawback can be overcome by using Model Predictive Control (MPC) which can handle constraints on the inputs and states. Constraint handling is one of the main features which distinguishes MPC from LQ control.

Another advantage that MPC offers in addition to constraint handling is that it is a time varying control law as opposed to the LQ controller which is a static control law. It is shown in [43] that if the prediction horizon is “long enough” then the unconstrained MPC solution is same as the solution of LQ control. However, when the constraints on the states or inputs are active, the control solution of MPC deviates from the LQ solution in order to satisfy the constraints. This is achieved through repeated on-line optimization of a convex cost function at every time step also known as the Receding Horizon Control. This also happens to be the drawback of MPC which restricts the application of MPC to fast dynamic systems such as quadrotor-slung load system in real-time. The optimization problem which needs to be solved at every time step is a computationally expensive task limiting the application of MPC to slower systems. However, tremendous effort has been put to overcome this limitation and tools/techniques are available which solves the optimization problem fast enough to be applicable for systems with fast dynamics. In this thesis work, state-of-the-art tools are employed to solve the MPC problem for the quadrotor-slung load system in real-time [1]. The control formulations discussed through out this section make use of the discrete time LTI model of Eq. (2-31).

[1] Real-time in this case would mean that the time taken to solve the optimization problem is well within the sampling period.

3-4-1 Generic Model Predictive Control problem

Before proceeding towards control methods for the specific case of the quadrotor-slung load system, it is worth describing a generic Model Predictive Control problem and highlight its advantages. Model Predictive Control or Receding Horizon control is a form of control method in which the control action is computed by solving at each sampling instant k, an open loop finite horizon optimal control problem using the current states of the system as the initial state. The optimization problem yields an optimal control sequence out of which the first control input is applied to the plant. The process is repeated again at next sampling instant and hence incorporating feedback. Since the main focus of this thesis is to design linear controllers, the further discussion is applicable only to the linear time invariant form of MPC.

Consider a discrete time LTI model of the plant

$$x_{k+1} = Ax_k + Bu_k$$

where $x_k \in \mathbb{R}^n$ and $u_k \in \mathbb{R}^m$ are the state and control inputs subject to constraints

$$x_k \in \mathcal{X}, \quad u_k \in \mathcal{U}, \quad \forall k \geq 0$$

where $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{U} \subseteq \mathbb{R}^m$ denote set of feasible states and inputs. The states of the above given system can be driven to origin (regulation problem) by solving the finite horizon optimal control problem of Eq. (3-11) at every time instant k.

$$\begin{aligned} \min_{u_{k+i|k}} \quad & p(x_{k+N|k}) + \sum_{i=0}^{N-1} l(x_{k+i|k}, u_{k+i|k}) \qquad (3\text{-}11)\\ \text{subject to} \quad & x_{k+i+1|k} = Ax_{k+i|k} + Bu_{k+i|k}, \quad i = 0, \cdots, N-1 \\ & x_{k+i|k} \in \mathcal{X}, \quad u_{k+i|k} \in \mathcal{U}, \quad i = 0, \cdots, N-1 \\ & x_{k+N|k} \in \mathcal{X}_f \\ & x_{k|k} = x_k \end{aligned}$$

In the above MPC formulation, $p(x_{k+N|k})$ denotes the terminal cost and $l(x_{k+i|k}, u_{k+i|k})$ denotes the stage cost. The constraint $x_{k+N|k} \in \mathcal{X}_f$ is a terminal constraint on the value of the states at the end of prediction horizon N. This constraint can be a set or a particular point in state space. The terminal cost and terminal constraints are important aspects for stability of MPC. Stability issues are discussed in a later section. The constraint $x_{k|k} = x_k$ is the initial condition which is input parameter for the MPC problem. The notation $x_{k+i|k}$ denotes the value of state x at time $k+i$ predicted at time k. As explained before, the solution of Eq. (3-11) yields an optimal control input sequence $u_{k+i|k} = \{u_{k|k}, u_{k+1|k}, \cdots, u_{k+N-1|k}\}$. The control input $u_{k|k}$ is applied to the plant at time k and the optimization problem of Eq. (3-11) is solved again at time $k+1$ with initial state $x_{k+1|k+1} = x_{k+1}$.

LTI plant model has two important consequences on the optimization problem of Eq. (3-11). Firstly, the optimization problem is a convex optimization problem with convex cost function and constraints. The sets $\mathcal{X}$ and $\mathcal{U}$ are polyhedral sets. See [44] for more details regarding polytopes and polyhedra. The most commonly used cost functions are the quadratic cost function also called 2-norm, max cost function also called ∞-norm and min cost function also called 1-norm. Eq. (3-11) can be recast as a Quadratic Programming problem with 2-norm cost function whereas 1-norm and ∞-norm cost functions result in Linear Programming problem. In this thesis, 2-norm or quadratic cost functions are used for different MPC formulations.

Second consequence is that of time invariance. The plant dynamics, cost function and the constraints are time invariant and hence the problem solved at time k is same as the one solved at time $k = 0$. Hence the notation '$|k$' can be dropped and the optimal control problem can be re-written with quadratic cost function as given in Eq. (3-12)

$$\begin{aligned}
\min_{u_i} \quad & x_N^T P x_N + \sum_{i=0}^{N-1} x_i^T Q x_i + u_i^T R u_i \qquad (3\text{-}12)\\
\text{subject to} \quad & x_{i+1} = Ax_i + Bu_i, \quad i = 0, \cdots, N-1\\
& x_i \in \mathcal{X}, \quad u_i \in \mathcal{U}, \quad i = 0, \cdots, N-1\\
& x_N \in \mathcal{X}_f\\
& x_0 = x_k
\end{aligned}$$

In the above MPC problem, Q and R are weighing matrices which place penalties on the states and the control inputs respectively. Hence they form the tuning parameters along with the prediction horizon N. Tuning of these parameters are described in Section 3-6. Now that a brief explanation of MPC technique is provided, the constraints and MPC formulations for quadrotor-slung load system is described in the following sections.

3-4-2 System Constraints

In order to formulate a MPC problem, appropriate constraints needs to be placed on the system variables. In this section, the constraints placed on the system variables are explained.

Input Constraints

The inputs to the system are the thrust f [N], the reference roll angle ϕ_{ref} [rad] and the reference pitch angle θ_{ref}. The maximum possible bank angle of the Parrot AR.Drone 2.0 is 30 degrees. However, since the LTI model of Eq. (2-31) is linearized around the hover conditions of quadrotor and stable equilibrium position of the slung load, the model is only locally valid around the linearization point. As a result the bank angle is restricted to ±5 degrees.

$$-5\pi/180 \leq \phi_{\text{ref}} \leq 5\pi/180$$

$$-5\pi/180 \leq \theta_{\text{ref}} \leq 5\pi/180$$

Similarly, based on the measured data the minimum thrust f_{min} provided by the quadrotor is 0.4 N and the maximum thrust f_{max} is 8 N. Therefore the thurst command is constrained as follows where $f_{\text{eq}} = (m + m_L)g$

$$f_{\text{min}} - f_{\text{eq}} \leq f \leq f_{\text{max}} - f_{\text{eq}}$$

The above constraints are written in compact way as given in Eq. (3-13)

$$\mathbf{u}_{\text{min}} \leq \mathbf{u}_k \leq \mathbf{u}_{\text{max}} \tag{3-13}$$

These box constraint equations denote the polyhedral set $\mathcal{U}$.

State Constraints

In the current formulation of the MPC no constraints on the states have been added due to the following reasons.

1. Any position (x, y, z) in the 3D Euclidean space is an equilibirum position of the quadrotor. Hence no constraints need to be added to the position of the quadrotor.
2. The control inputs act on the accelerations of the quadrotor and they are constrained around the linearization point. Adding constraints on inputs imposes constraints on accelerations. Since the quadrotor is not expected to make aggressive or fast motion, velocity states of the quadrotor cannot attain unrealistic values eliminating need to add constraints on them explicitly.

Therefore $\mathcal{X} = \mathbb{R}^n$ with $n = 14$ for the quadrotor-slung load system.

3-4-3 Model Predictive Control with Integral Action

Similar to the LQ control with integral action, the first logical problem formulation within MPC is to design a Model Predictive Control with integral action (MPC-I). The model used here is same as the one used for LQ control with integral action. Let the Eq. (3-6) be written in a compact manner as follows

$$\begin{aligned} \begin{bmatrix} \boldsymbol{\xi}_{k+1} \\ \mathbf{v}_{k+1} \end{bmatrix} &= \begin{bmatrix} A_\xi & 0 \\ C_I C_\xi & I \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi}_k \\ \mathbf{v}_k \end{bmatrix} + \begin{bmatrix} B_\xi \\ 0 \end{bmatrix} \mathbf{u}_k \\ \mathbf{x}_{k+1} &= A\mathbf{x}_k + B\mathbf{u}_k \end{aligned} \tag{3-14}$$

MPC-I problem formulation is then given in Eq. (3-15)

$$\begin{aligned} \min_{\mathbf{u}_i} \quad & \mathbf{x}_N^T P \mathbf{x}_N + \sum_{i=1}^{N-1} \mathbf{x}_i^T Q \mathbf{x}_i + \mathbf{u}_i^T R \mathbf{u}_i \\ \text{subject to} \quad & \mathbf{x}_{i+1} = A\mathbf{x}_i + B\mathbf{u}_i, \quad i = 0, \cdots, N-1 \\ & \mathbf{u}_{\text{min}} \leq \mathbf{u}_i \leq \mathbf{u}_{\text{max}}, \quad i = 0, \cdots, N-1 \\ & \mathbf{x}_0 = \hat{\mathbf{x}}_k \end{aligned} \tag{3-15}$$

The optimization problem of Eq. (3-15) is a regulation problem and is solved at every time instant k with initial condition $\mathbf{x}_0$ initialized with the estimates of states obtained from the

Kalman filter $\hat{\boldsymbol{\xi}}_k$ and integral states $\mathbf{v}_k$ [2].

$$\hat{\mathbf{x}}_k = \begin{bmatrix} \hat{\boldsymbol{\xi}}_k \\ \mathbf{v}_k \end{bmatrix}$$

$$\mathbf{v}_k = \mathbf{v}_{k-1} + \begin{bmatrix} x_Q \\ y_Q \\ z_Q \end{bmatrix}$$

The problem is converted to output tracking problem in the same way as described in Section 3-3, i.e.

$$\hat{\mathbf{x}}_k = \begin{bmatrix} \hat{\boldsymbol{\xi}}_k - \boldsymbol{\xi}_{\text{ref}} \\ \mathbf{v}_k \end{bmatrix}$$

$$\mathbf{v}_k = \mathbf{v}_{k-1} + \begin{bmatrix} x_Q - x_{Q_{\text{ref}}} \\ y_Q - y_{Q_{\text{ref}}} \\ z_Q - z_{Q_{\text{ref}}} \end{bmatrix}$$

The simulation results of the MPC-I formulation is discussed in subsection 3-8-2. The main advantage that MPC-I formulation has over LQ control with integral action is that it inherits all the desired properties of LQ control in addition to constraint handling. Similar to the LQ control with integral action, MPC-I formulation is able to track the desired quadrotor outputs without any steady state error. The disadvantage however is that there is large overshoot in the step response due to the introduction of integral states. More elaborate discussion is presented in Section 3-8.

In order to circumvent this problem of large overshoot in output response due to the integral action, Δu formulation is proposed which is the topic of discussion in the next subsection. The question is whether the Δu formulation can eliminate the steady state error without large overshoot in the output which was the noticed in the MPC-I formulation.

3-4-4 $\Delta \mathrm{u}$ formulation

The basic idea of the Δu formulation is to plan the deviation in control inputs $\Delta \mathbf{u}_k$, instead of the control inputs $\mathbf{u}_k$ themselves. This is equivalent to adding integral action at the input in contrast to the previous subsection where the integral action was added at the output/output error. The state vector of the quadrotor-slung load system is augmented with the previously (at time $k-1$) computed control inputs as additional states with deviation in the control inputs as the new input to the system. The augmented state space model for Δu formulation is given in Eq. (3-16).

$$\begin{bmatrix} \boldsymbol{\xi}_{k+1} \\ \mathbf{u}_k \end{bmatrix} = \begin{bmatrix} A_\xi & B_\xi \\ 0 & I \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi}_k \\ \mathbf{u}_{k-1} \end{bmatrix} + \begin{bmatrix} B_\xi \\ I \end{bmatrix} \Delta \mathbf{u}_k \tag{3-16}$$

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\Delta\mathbf{u}_k$$

[2] Setting the value of integral states $\mathbf{v}_k$ to zero at every time instant, results in a nominal controller performance.

The MPC problem with Δu formulation is then given in Eq. (3-17).

$$\begin{aligned} \min_{\Delta \mathbf{u}_i} \quad & \mathbf{x}_N^T P \mathbf{x}_N + \sum_{i=1}^{N-1} \mathbf{x}_i^T Q \mathbf{x}_i + \Delta \mathbf{u}_i^T R \Delta \mathbf{u}_i \\ \text{subject to} \quad & \mathbf{x}_{i+1} = A\mathbf{x}_i + B\Delta \mathbf{u}_i, \quad i = 0, \cdots, N-1 \\ & \mathbf{u}_{\text{min}} \leq \mathbf{u}_i \leq \mathbf{u}_{\text{max}}, \quad i = 1, \cdots, N-1 \\ & \mathbf{x}_0 = \hat{\mathbf{x}}_k \end{aligned} \tag{3-17}$$

Note that it is possible to place constraints on the changes in the control input $\Delta \mathbf{u}_k$, also called slew rate constraints. However, in this case the computed control inputs are actually the reference values for lower level controllers and hence only the bounds on the control inputs are sufficient. Placing constraints on the slew rate do not serve any purpose. The initial condition $\hat{\mathbf{x}}_k$ is given as

$$\hat{\mathbf{x}}_k = \begin{bmatrix} \hat{\boldsymbol{\xi}}_k \\ \hat{\mathbf{u}}_k \end{bmatrix}$$

where $\hat{\boldsymbol{\xi}}_k$ are the estimated states of the quadrotor-slung load system obtained from the Kalman filter. The variable $\hat{\mathbf{u}}_k$ can be initialized with two possibilities. First possibility is to initialize it with the previously computed control input $\mathbf{u}_{k-1}$. Utilizing the previously computed control input in this way would defeat the purpose of using this formulation in the first place. Since the control inputs are computed using the model, it would be unable to compensate for the model mismatch. As a result, there would still be a non zero steady state error at the output.

Another approach is to initialize $\hat{\mathbf{u}}_k$ as given in [44]. The idea is to allow the state estimator (Kalman filter in this case) to estimate this value. The estimated value need not necessarily be equal to the previously computed control input to the system $\mathbf{u}_{k-1}$. An advantage of this method is that the model uncertainties are lumped into this estimate and can be loosely considered as additional states pertaining to the input disturbances.

The problem of Eq. (3-17) is a regulation problem and is converted to tracking problem as described before in Section 3-3. The simulation results of the Δu formulation is discussed in subsection 3-8-3. The simulation results show that the controller is able to completely eliminate the steady state offset introduced due to model mismatch. Another added advantage is that the integral windup effect observed in previous formulation is eliminated. However there is one issue with this formulation that needs to be addressed.

When there is change in step reference command, the MPC problem is infeasible during transients. This infeasibility is due to the fact that the constraint $\mathbf{u}_{\text{min}} \leq \hat{\mathbf{u}}_k + \Delta \mathbf{u}_k \leq \mathbf{u}_{\text{max}}$ is not feasible for the first step in the N-step prediction horizon. As a result solver returns a value which is not useful for control. One reason for such infeasiblity is that the Kalman filter estimate $\hat{\mathbf{u}}_k$ could already be out of bounds or the input constraints could be too stringent during the transients. In order to make the problem feasible, the constraints on the inputs are relaxed for first time step in prediction horizon. Hence the input constraints are applied from $i = 1$ in Eq. (3-17). More details are discussed in subsection 3-8-3.

The control input applied to the system at time k, denoted by $\mathbf{u}_{\text{app}|k}$ is computed externally as

$$\mathbf{u}_{\text{app}|k} = \mathbf{u}_{\text{app}|k-1} + \Delta \mathbf{u}_k \tag{3-18}$$

Note that $\mathbf{u}_{\text{app}|k}$ need not necessarily be equal to $\hat{\mathbf{u}}_k$. Another advantage is that the target inputs need not be computed for this formulation unlike the previous formulations where $\mathbf{u}_{\text{eq}}$ was added to the computed control inputs.

3-5 Fast Quadratic Program solvers

At the core of a MPC problem lies an optimization problem which needs to be solved. The computed control input is the result of solution of an optimization problem. Solving an optimization problem is computationally expensive and hence the use of Model Predictive Controllers have been limited to systems with slow dynamics (sampling time of the order of minutes or even hours) which are typically found in process industries. The properties such as guaranteed convergence and convergence within finite time (often quick convergence for faster systems) are important for application in real systems. The ability to achieve these properties depend on the type of the problem and the method used to solve the optimization problem.

Since the system model and the constraints are linear, the optimization problem becomes a convex problem. Convex problems are known to possess nice properties such as guaranteed convergence within finite time. With the use of quadratic cost functions, the MPC problem formulations of Eq. (3-15) and Eq. (3-17) can be recast as a Quadratic Programming (QP) problem. Many well known methods such as active set method, interior point method exist to solve such problems [45, 44]. Even with such nice properties, application of such methods to fast systems are still a problematic task. However new tools and techniques are available which allow implementation of MPC for fast systems such as quadrotor-slung load system. These methods exploit structural properties of the problem to achieve fast convergence and solution times. The discussion related to such methods are out of scope of this thesis.

Practical considerations

From the practical view point, in order to solve the MPC problems within the sample time of 100 Hz state-of-the-art solvers like FORCES Pro [46], FiOrdOs[3], CVXGEN [34] were considered. These solvers basically generate an efficient C object file given the required data (cost function, equality and inequality constraints, bounds on variables, initial conditions etc.) pertaining to the Quadratic programming problem. The generated C object file can then be used in the application software and executed at every sample time. In [47] it is shown that FORCES Pro outperforms CVXGEN in terms of computation time. Hence the only two alternatives that were available were FORCES Pro and FiOrdOs. FORCES Pro was selected over FiOrdOs due to the following reasons.

1. Supports generation of code for multiple platforms such as ARM, x86 etc in their commercial version. In this thesis, the free version of FORCES Pro (which is no longer available for future users) which generates a C object file for genetic host computer is used. Since the drone used in this thesis has an ARM Cortex A8 processor, a C code could be generated for the specific processor using the commercial version thereby enabling on-board execution of controllers developed in this thesis.

[3] http://fiordos.ethz.ch/dokuwiki/doku.php

2. Availability of multiple methods in the commercial version to solve the optimization problem resulting in better performance (solution time) than obtained within this thesis.

Having listed the reasons to use FORCES Pro, use of FiOrdOs cannot be ruled out and it would be a nice experiment to employ FiOrdOs and compare its performance with FORCES Pro.

MPC problem in FORCES format

FORCES Pro solvers are capable of solving convex multistage Quadratically Constrained Quadratic Programs of the form given in Eq. (3-19).

$$\begin{aligned} \min_{z_i} \quad & \sum_{i=1}^{N} \frac{1}{2} z_i^T H_i z_i + f_i^T z_i \\ \text{subject to} \quad & C_1 z_1 = c_1 \\ & C_i z_i + D_{i+1} z_{i+1} = c_i \\ & \underline{z}_i \leq z_i \leq \bar{z}_i \\ & A_i z_i \leq b_i \end{aligned} \tag{3-19}$$

where $i = 1, \cdots, N$. Notice that the starting index is $i = 1$ in this case instead of $i = 0$ as defined in MPC problem formulations. This is because the FORCES Pro uses a Matlab client to generate a custom solver for the optimization problem of type given above and the starting index in Matlab is $i = 1$. The cost function shown is separable objective function, and is defined over N stages and hence the use of word 'multistage'. The variables z_i are called stage variables. The first constraint defines the initial condition constraint which is an equality constraint. The second constraint is an inter-stage equality constraint and are used to enforce system dynamics constraint in the optimization problem. The third constraint defines bounds on the stage variables. Final constraint defines polytopic inequality constraints which can used to place constraints on linear combinations of stage variables.

The matrices H_i, A_i, C_i, D_i and vectors $f_i, \underline{z}_i, \bar{z}_i, b_i, c_i$ are the data needed to define the QP problem and must be specified by the user. The problem data can be different in every stage and can change with every call to the solver, allowing implementation of Linear Time Varying Model Predictive Controllers. The MPC-I formulation reproduced below can be converted into multistage QP optimization problem as follows

$$\begin{aligned} \min_{\mathbf{u}_i} \quad & \mathbf{x}_N^T P \mathbf{x}_N + \sum_{i=1}^{N-1} \mathbf{x}_i^T Q \mathbf{x}_i + \mathbf{u}_i^T R \mathbf{u}_i \\ \text{subject to} \quad & \mathbf{x}_{i+1} = A\mathbf{x}_i + B\mathbf{u}_i, \quad i = 0, \cdots, N-1 \\ & \mathbf{u}_{\text{min}} \leq \mathbf{u}_i \leq \mathbf{u}_{\text{max}}, \quad i = 0, \cdots, N-1 \\ & \mathbf{x}_0 = \hat{\mathbf{x}}_k \end{aligned} \tag{3-20}$$

- The stage variables z_i are given as

$$z_i = \begin{bmatrix} \mathbf{x}_i \\ \mathbf{u}_i \end{bmatrix} \quad \forall i = 1, \cdots, N-1$$

For the N^{th} stage, the stage variable $z_N = \mathbf{x}_N$. Note that the state vector $\mathbf{x}_i$ is a decision variable for the optimization problem. This is due to the 'multistage' nature of the problem.

- The cost function is then defined as

$$H_i = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \quad \forall i = 1, \cdots, N-1$$

Since the stage variable for N^{th} stage is $\mathbf{x}_N$, the cost function for N^{th} stage is $H_N = P$. $f_i = 0 \quad \forall i = 1, \cdots, N$ as we do not have any precomputed reference state and input trajectories that need to be followed.

- The inter-stage equality constraints basically enforce the system dynamics to the evolution of state trajectories. Each stage can be seen as a solution to the two point boundary value problem. The final value of a stage i must be equal to initial value of stage $i+1$. This enforces continuity in the state trajectory. Due to initial conditions the matrices C_1 and D_2 are initialized differently

$$\begin{bmatrix} I & 0 \\ A & B \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{u}_1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -I & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_2 \\ \mathbf{u}_2 \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}_k \\ 0 \end{bmatrix}$$
$$C_1 z_1 + D_2 z_2 = c_1$$

The initial condition c_1 is the input parameter to the optimization problem. Similarly,

$$\begin{aligned} C_i &= \begin{bmatrix} A & B \end{bmatrix} \quad \forall i = 2, \cdots, N-1 \\ D_i &= \begin{bmatrix} -I & 0 \end{bmatrix} \quad \forall i = 3, \cdots, N-1 \\ c_i &= 0 \quad \forall i = 2, \cdots, N \end{aligned}$$

Since $\mathbf{x}_N$ is the only stage variable for N^{th} stage, $D_N = -I$

- Lastly, the bounds on the stage variables are straight forward.

$$\begin{aligned} \underline{z}_i &= \begin{bmatrix} - \\ \mathbf{u}_{\min} \end{bmatrix} \quad \forall i = 1, \cdots, N-1 \\ \bar{z}_i &= \begin{bmatrix} - \\ \mathbf{u}_{\max} \end{bmatrix} \quad \forall i = 1, \cdots, N-1 \end{aligned}$$

The '$-$' sign indicates that no bounds are placed on the system states. Having defined the optimization problem data in the form as just described, one can notice that the structure of the optimization problem is sparse. That is, the QP problem is written as

$$\begin{aligned} \min_{z} \quad & \frac{1}{2} z^T H z \\ \text{subject to} \quad & A_e z = b_e \\ & \underline{z} \leq z \leq \bar{z} \end{aligned}$$

where the matrices H, A_e and vectors $z, \underline{z}, \bar{z}$ are given as

$$H = \begin{bmatrix} H_1 & & \\ & \ddots & \\ & & H_N \end{bmatrix} \quad z = \begin{bmatrix} z_1 \\ \vdots \\ z_N \end{bmatrix} \quad \underline{z} = \begin{bmatrix} \underline{z}_1 \\ \vdots \\ \underline{z}_N \end{bmatrix} \quad \bar{z} = \begin{bmatrix} \bar{z}_1 \\ \vdots \\ \bar{z}_N \end{bmatrix}$$

$$A_e = \begin{bmatrix} C_1 & D_2 & 0 & \cdots & 0 \\ 0 & C_2 & D_3 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & C_{N-1} & D_N \end{bmatrix} \quad b_e = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

It is this sparsity that is exploited to generate fast solvers by FORCES Pro. Reader is referred to [47] for more information regarding the optimization method and numerical methods employed for its efficient solution.

3-6 Tuning of Model Predictive Controllers

In every controller, there are always some parameters which need to be tuned in order to achieve the desired performance. Stability of the controllers rely on the tuning parameters. Wrong tuning of controllers can even destabilize an inherently stable system. In this section, the tuning parameters for MPC are discussed. It was earlier remarked that the LQ controllers are equivalent to infinite horizon, unconstrained Model Predictive Controllers. Hence the MPC inherits the tuning parameters of LQ controllers. However, there are additional parameters within the MPC setting which have important implications on the performance and stability aspects of the controller.

State and control input weight matrices

First set of tuning parameters which are common to both the LQ controllers and MPC are the weights on the states Q and control inputs R. Generally, both the matrices are chosen to be diagonal thereby reducing the number of tunable entries within the matrix. It also allows to place weights on the states and control inputs individually. Higher weights on states results in faster response times (high bandwidth) accompanied with aggressive control actions and sometimes even large overshoots. High weights on the control inputs makes the system response slower (low bandwidth) with low magnitude of control inputs. Hence there is a trade-off that needs to be made by the designer depending on the objectives. It is possible with appropriate tuning, to achieve both the fast response with aggressive control inputs and slow response performance for a given system in simulation. However, this might not be the case in practice. It is usual that these tuning parameters are always changed during practical implementation. The system can sometimes give useful insight to choose the desired control performance and hence the tuning.

For the quadrotor-slung load system, it is important to remember that the controllers are designed based on the linear approximation of the original nonlinear system. Hence the used model is only valid locally and it is necessary to operate the system around the linearization point. As a result, the possibility of performing aggressive maneuvers with the quadrotor-slung load system is ruled out restricting the system to only slower motions. This forms the first guideline in tuning the controllers. Aggressive control inputs are not desired as it would destabilize the system. Hence, high penalty was placed on the attitude reference control inputs ϕ_{ref} and θ_{ref} such that they are restricted to lie between the bounds[4]. For simulations and experiments, the control inputs weight matrix was set to

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 10^4 & 0 \\ 0 & 0 & 10^4 \end{bmatrix}$$

The objective is also to minimize the swing angle of the slung load. Ideally it would be desirable to have a zero swing angle, though not possible in practice. Hence, relatively higher weights were placed on swing angle states compared to the other states. Finally we also want the quadrotor to reach a desired set point and the weights were set through extensive testing in experiments. Weights were also placed on the velocity states so that there is some damping effect introduced in the system. The Q matrix used in the simulation and experiments is given below. Note that the lower block diagonal matrix weight corresponds to the states of the attitude dynamics. Since weights were placed on the inputs to the attitude dynamics, weights on these states were not necessary and hence non-zero weights were assigned.

$$Q = \left[\begin{array}{cccccccccc|cccc}
50 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 50 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 100 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 100 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^{-3} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^{-3} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^{-3} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^{-3}
\end{array}\right]$$

For the LQ control with integral action and MPC with integral action, low penalty was placed on integral states as high penalty would destabilize the system. The weights on the integral states were placed as

$$Q_i = \begin{bmatrix} 0.001 & 0 & 0 \\ 0 & 0.001 & 0 \\ 0 & 0 & 0.001 \end{bmatrix}$$

It can be noted that with increase in number of states/control inputs tuning can be a difficult task and hence one must resort to intuitive reasoning and educated guess in order to tune these parameters. Finally, these parameters will always need to be tuned through repeated experimental testing in order to achieve desired performance.

[4] Maximum bank angle of 5 degrees was set as bounds

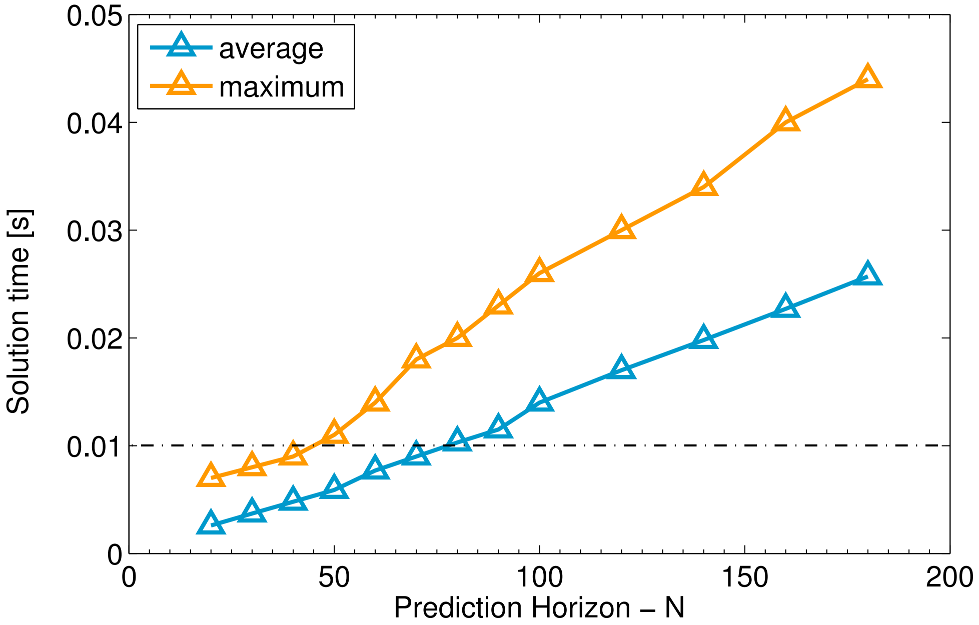


Figure 3-3: Solution time versus prediction horizon

Prediction Horizon

Prediction horizon N forms a tuning parameter in a MPC. One of the ways to guarantee stability in MPC is to have infinite horizon. It is shown in [43] that as N approaches infinity, the solution of the unconstrained MPC approaches that of LQ control. Hence, from stability point of view it is desirable to have infinite prediction horizon. However this is not computationally feasible and hence N needs to be large enough to achieve stability. More discussions regarding stability in Section 3-7. A large prediction horizon also leads to higher computational burden which is a crucial aspect in implementation of controller on real system. As a result, a prediction horizon had to be chosen keeping in mind the computation requirements. Figure 3-3 shows the plot of average and worst case computation time required to solve MPC-I problem of Eq. (3-15) versus prediction horizon. With a sampling frequency of 100 Hz, the maximum possible prediction horizon which could be used was $N = 40$ samples (0.4 seconds). In order to accommodate for larger prediction horizon, either the sampling frequency needs to be reduced or move blocking schemes [48] can be employed. In this thesis work, a prediction horizon of $N = 20$ samples (0.2 seconds) was chosen.

3-7 Stability

It is well known that the finite horizon constrained MPC does not guarantee stability and hence several techniques are used to guarantee stability of MPC. The different techniques used to guarantee stability of MPC are

1. Infinite Horizon

2. End point constraints or terminal equality constraint
3. Terminal cost function
4. Terminal constraint set
5. Terminal cost and constraint set.

Clearly given the generic MPC problem of Eq. (3-12), the factors that determine stability properties are the prediction horizon N, terminal cost function P and the terminal constraint set $\mathcal{X}_f$. In unconstrained linear systems, it has been shown that the infinite horizon MPC solution approaches that of the LQ control and hence inherits the stability properties of the LQ control. However, having an infinite horizon with constraint handling is not practically possible. Hence generally N is chosen large enough and terminal cost function P is chosen to guarantee stability. Due to computational limitations, N was chosen to be 20 for the experiments in this thesis. Reference [44, 42] show that the terminal cost P is chosen as solution to the Ricatti equation related to the LQ control problem. Hence the matrix P was computed using the `dlqr` Matlab function in all the MPC formulations presented in this chapter as a means to guarantee stability.

Other methods to guarantee stability are to enforce terminal set constraints $x_N \in \mathcal{X}_f$. When $\mathcal{X}_f = 0$, we have the terminal equality constraint which is a hard constraint and may lead to an infeasible MPC problem. The reason being it may not be possible to achieve the end-point constraint in N steps. This is usually not a problem for longer N. Since the chosen prediction horizon is short, no end point constraints are added. To deal with the problem of in-feasibility, terminal set constraints are enforced. Calculation of terminal constraint set $\mathcal{X}_f$ is given in [44]. For linear constrained unstable systems it is necessary to employ both the terminal cost function and terminal set constraint to guarantee stability. However, in the case of quadrotor-slung load system, there are no constraints on the states and the system is marginally stable system hence the choice of $\mathcal{X}_f$ is less critical. As a result only the terminal cost function was added to the system to ensure stability.

3-8 Simulation Results

In this section, the simulation results for different control methods are discussed. The advantages and limitations of these control methods are presented and compared. The following criteria are used for evaluation and comparison of the controllers performance

1. Rise time
2. Setting time
3. Steady state error
4. Percentage Overshoot
5. Computation time

In order to be able to compare different control formulations, the simulations are performed with same tuning parameters. State and control weight matrices are common tuning parameters for both the LQ control and MPC. Hence the tuning parameters presented in Section 3-6 are used. The prediction horizon of $N = 20$ was used for the reasons described earlier. The quadrotor is required to track step changes in reference positions in $x_{Q_{\text{ref}}}$ and $y_{Q_{\text{ref}}}$ while maintaining a constant altitude $z_{Q_{\text{ref}}}$. The slung load swing angle is initialized at $\theta_L = \pi/10$ radians (18 degrees) and $\phi_L = -\pi/10$ radians (-18 degrees). The controller must be able to stablize the slung load and keep the swing angles minimum (close to zero) while tracking the step reference positions.

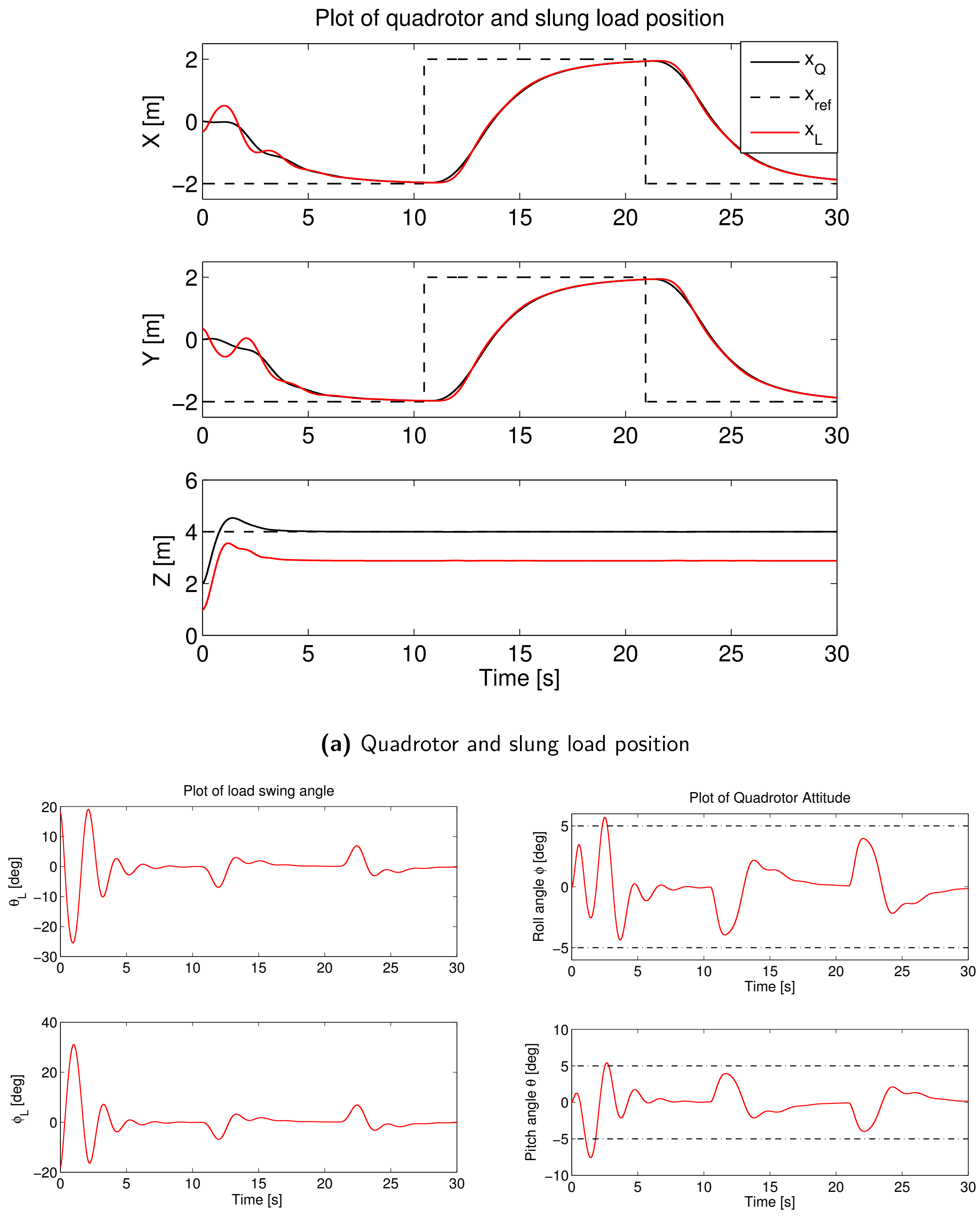
In order to keep the treatment concise and convey the core message, some simulation plots are not presented in this section. Reader can refer to these plots in the Appendix A.

3-8-1 Linear Quadratic Control with Integral Action

In this section the simulation results of LQ control with integral action is discussed. This controller forms the benchmark controller for comparing the MPC formulations. As discussed in Section 3-3, the tracking of step reference position is achieved by using the control law of Eq. (3-9) and Eq. (3-10). However use of such a controller results in a very large overshoot of 74% in (X,Y) position which is undesirable. Consequently the rise time is approximately 3 seconds with settling time of 8 seconds. The controller is able to stabilize the slung load through very high control inputs during transients which is practically not realizable. The control inputs have to be bounded external to the controller before being applied to the system. Even though the controller is able to stabilize the slung load, due to control inputs of higher magnitude, the slung load is subjected to larger swing angles during changes in step reference. In order to reduce the large overshoot, a small modification was made to the controller. The modification is to provide the state of the system $\hat{\boldsymbol{\xi}}_k$ as feedback (henceforth referred as LQ control with `position` feedback) instead of the error term $\hat{\boldsymbol{\xi}}_k - \boldsymbol{\xi}_{\text{ref}}$ (referred as LQ control with `position error` feedback). Tracking is achieved through integral states $\mathbf{v}_k$ which integrates the error in states using Eq. (3-10). Figure 3-4 shows the simulation results using the controller with modifications mentioned above. See Figure A-1 in Appendix A for simulation results of LQ control with integral action with position error of quadrotor as feedback.

Figure 3-4a shows the plot of the quadrotor and the slung load. Clearly, the controller is able to stabilize the slung load and also track the position reference. The modification suggested above is able to eliminate the large overshoot obtained using the controller of Eq. (3-9). This improvement in performance is obtained at the cost of slower response i.e., a rise time and settling time of approximately 9 seconds and 10 seconds respectively. The controller also maintains a constant altitude without any steady state error. Note that the overshoot in altitude is due to fact that $z_Q - z_{Q_{\text{ref}}}$ is provided as feedback to the controller. Overshoot should not be a problem since we require the quadrotor to maintain a constant altitude during the horizontal motion. Also note that the vertical motion of the quadrotor does not influence the swing angles of the slung load.

Figure 3-4b shows the plot of swing angles of the slung load and it can be noticed that the controller is able to stabilize the slung load with relatively lower swing angles during step reference changes. Figure 3-4c and 3-4d shows the plot of quadrotor attitude and reference

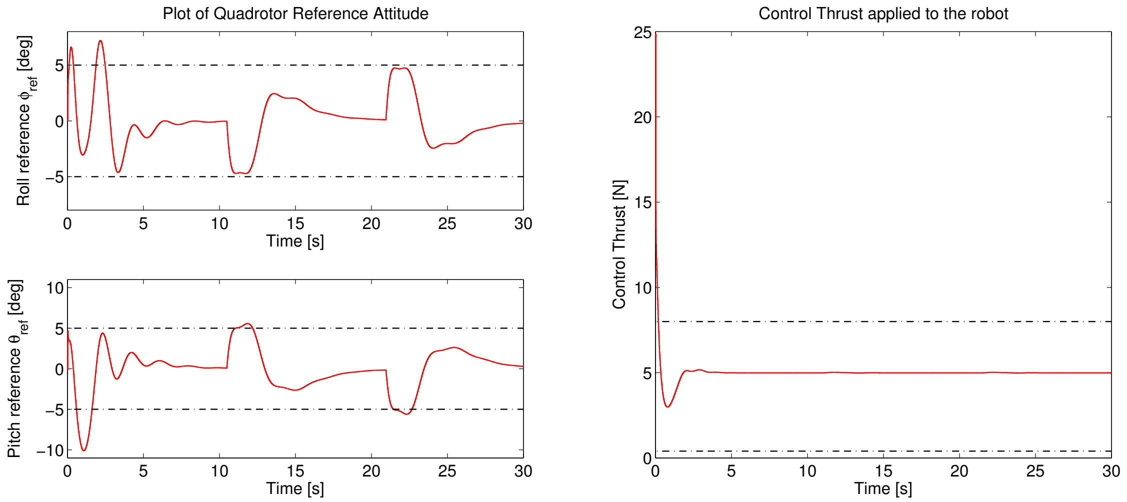


(a) Quadrotor and slung load position

(b) Swing angles of the slung load

(c) Quadrotor attitude

Figure 3-4: Linear Quadratic control with integral action with position of the quadrotor $\mathbf{x}_Q$ as feedback



(d) Attitude reference

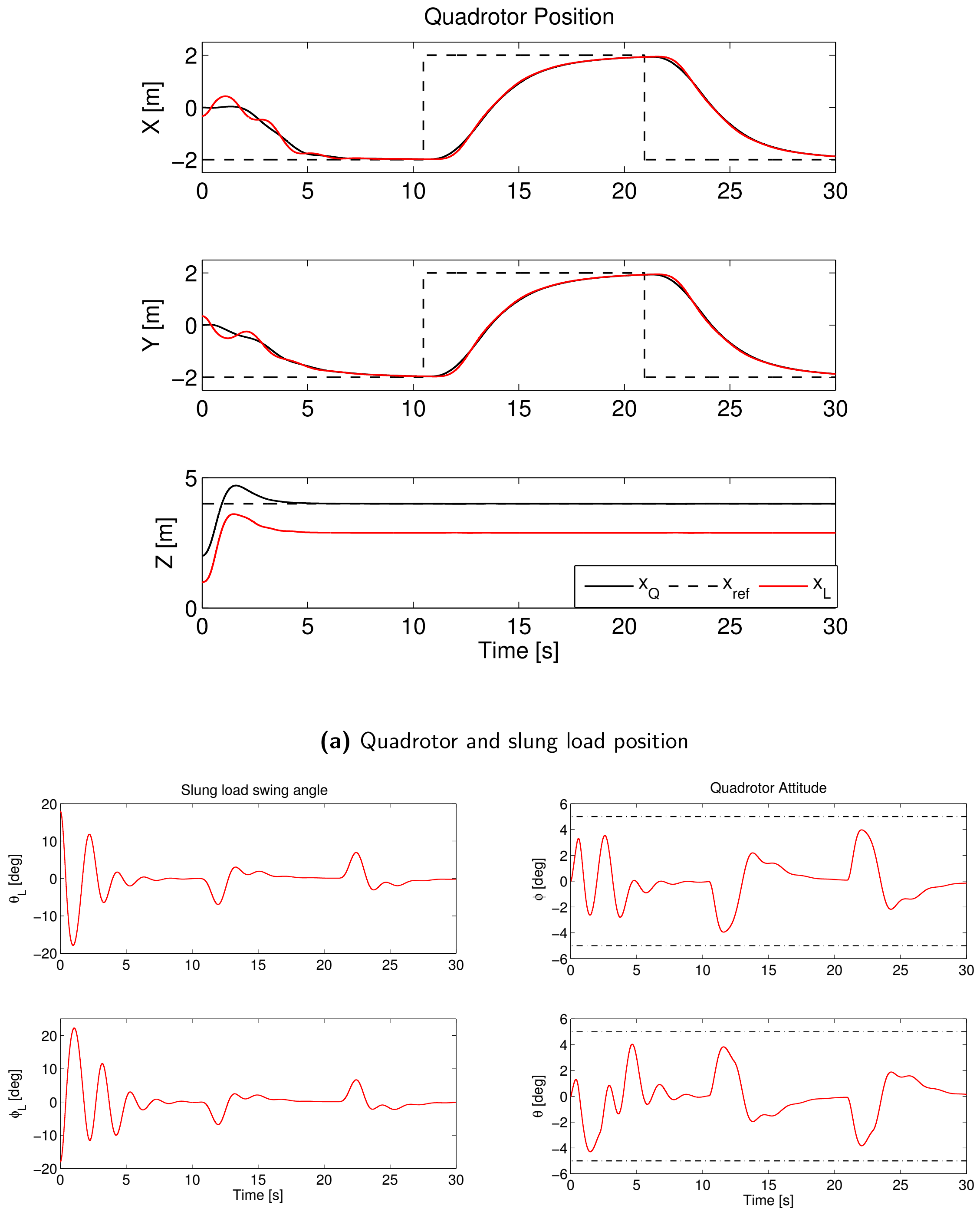
(e) Control thrust

Figure 3-4: Linear Quadratic control with integral action with position of the quadrotor $\mathbf{x}_Q$ as feedback

command issued to the lower level attitude controller respectively. Due to high penalty on attitude references, the values are less aggressive and lie within the bounds (dashed line) most of the time. Figure 3-4e shows the plot of applied thrust command.

The simulations results in this section leads to following conclusions. Firstly, a controller based on linear time invariant model can be used to transport slung load from one point to another in a swing free manner using a quadrotor. The drawback of this control method is the static control law and inability to handle constraints. Additionally, the controller also introduces a large overshoot in the output response due to the inclusion of integral states. A small modification to the controller was made to avoid a large overshoot while compromising the response time. Hence, advanced control methods like MPC can be used to achieve better performance. MPC clearly can address the issues of constraint handling and provides many flexibilities to formulate the control problem in order to alleviate the issues with the LQ control.

3-8-2 Model Predictive Control with Integral Action

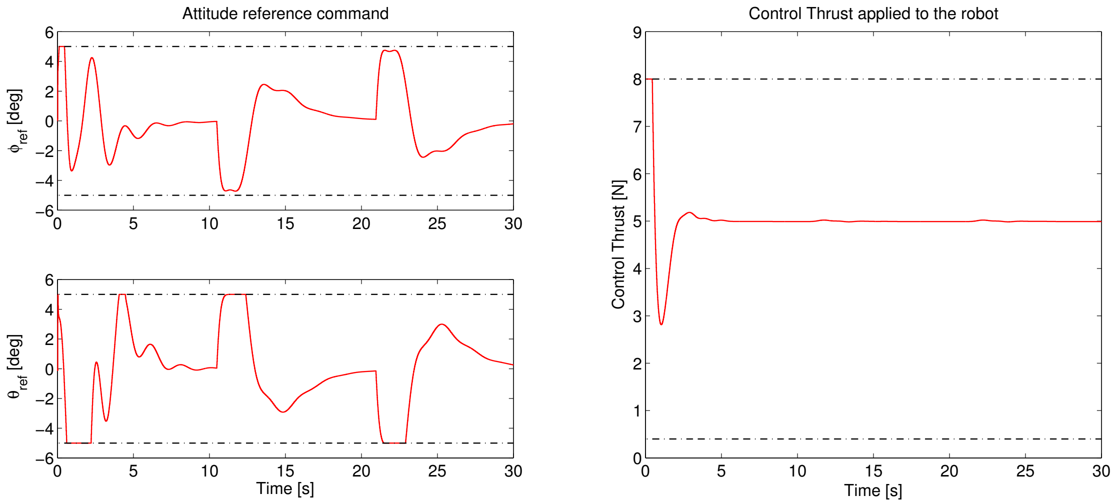


(a) Quadrotor and slung load position

(b) Swing angles of the slung load

(c) Attitude of the quadrotor

Figure 3-5: Model Predictive Control with integral action with position of the quadrotor $\mathbf{x}_Q$ as feedback



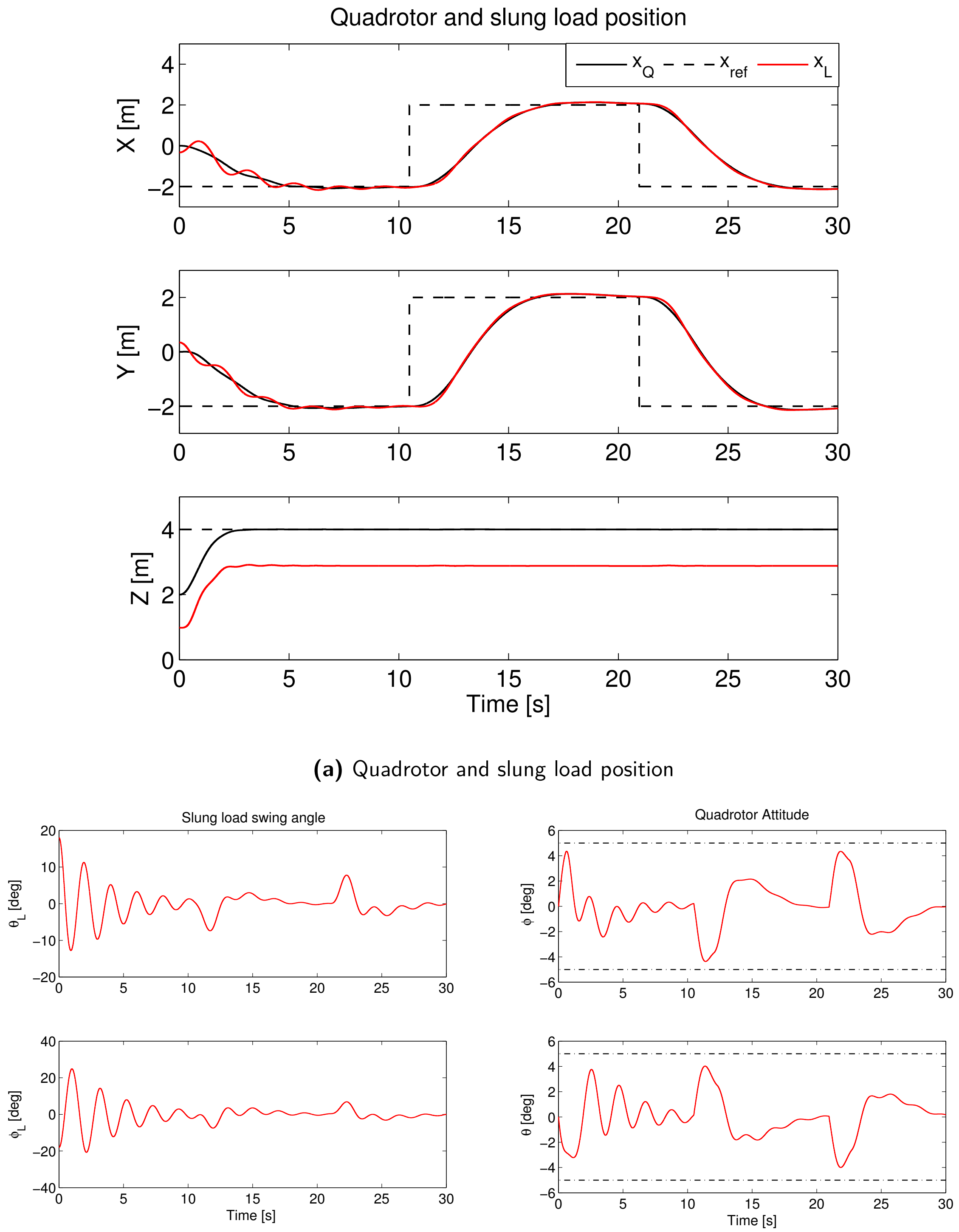
(d) Attitude reference

(e) Control thrust

Figure 3-5: Model Predictive Control with integral action with position of the quadrotor $\mathbf{x}_Q$ as feedback

Figure 3-5 shows the simulation results with MPC-I controller discussed in subsection 3-4-3. The MPC-I controller suffers from the same drawback as the LQ control with integral action. The overshoot (30%) is now lesser than the LQ controller. See Figure A-2 in Appendix A for results of MPC-I with position error feedback. Hence, similar modification to the controller were made for the results presented in this subsection. Figure 3-5a shows the plot of quadrotor and slung load position. As expected the controller is able to stabilize the slung load and follow the step reference position while maintaining a constant altitude. One advantage that MPC-I offers over LQ control is constraint handling as can be seen from the attitude reference plot of Figure 3-5d and applied thrust plot of Figure 3-5e. Also note that most of the simulation time the constraints are not active. Hence the output response of both the MPC-I and the LQ controller are similar demonstrating the fact that the unconstrained infinite horizon (achieved through terminal cost as described in Section 3-7) MPC controller is equivalent to the LQ control. The current control formulation suffers from the same drawback of slow response with rise time and settling time of 9 seconds and 10 seconds respectively.

3-8-3 Model Predictive Control with Δu formulation

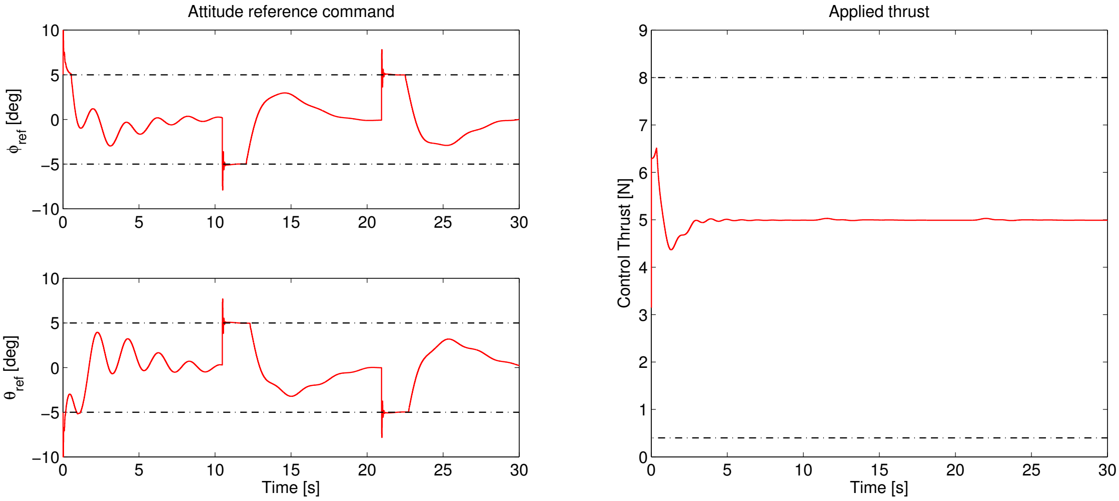


(a) Quadrotor and slung load position

(b) Swing angles of the slung load

(c) Attitude of the quadrotor

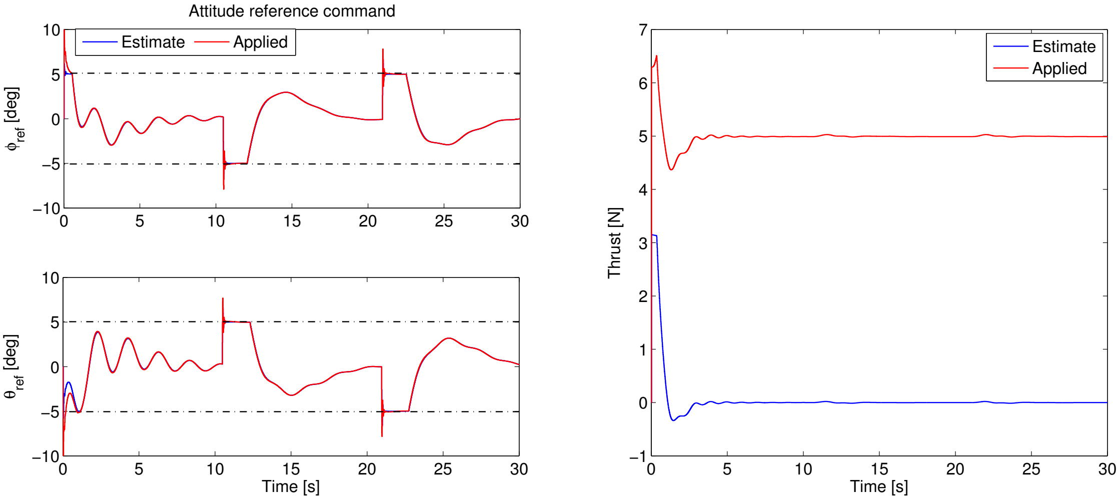
Figure 3-6: Model Predictive Control with Δu formulation



(d) Attitude reference

(e) Control thrust

Figure 3-6: Model Predictive Control with Δu formulation



(a) Attitude reference

(b) Control thrust

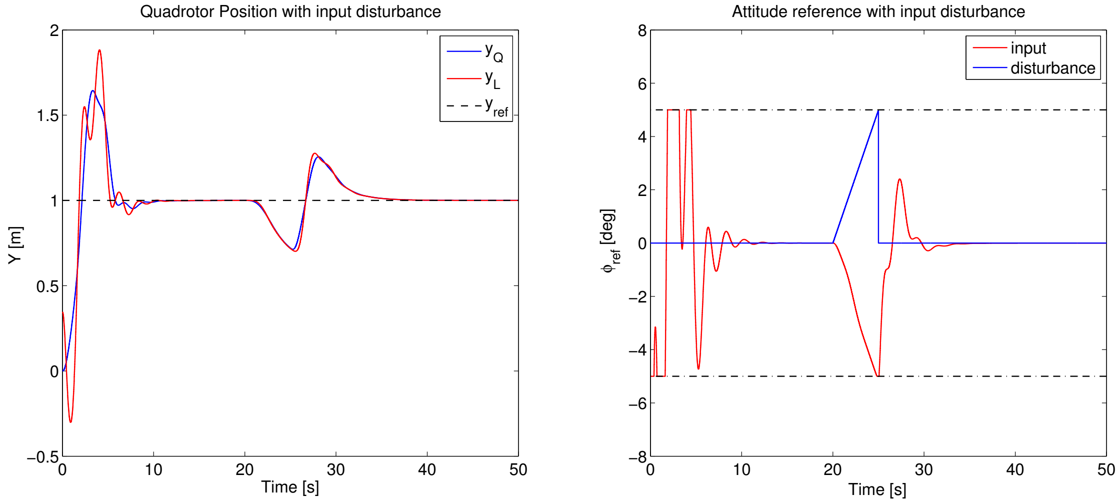
Figure 3-7: Estimated and applied control inputs

Figure 3-6 shows the simulation results of MPC with Δu formulation. The motivation and advantages of using the controller with Δu formulation was explained in subsection 3-4-4. Figure 3-6a shows the plot of quadrotor and slung load position. The first thing which can be noticed is that the response of the system during transients is considerably improved over both the LQ control with integral action and MPC-I. The rise time and settling time is 5 seconds and 6 seconds respectively with overshoot of just 7%. Clearly the controller exhibits performance improvement in both the overshoot and settling time aspects. Figure 3-6b shows the plot of load swing angle and it can be noticed that the controller is able to stabilize the slung load with decaying oscillations. This is due to the heavy penalty placed on attitude references. Figure 3-6d and 3-6e shows the plot of applied attitude references and thrust respectively. While the thrust input is within the bounds, the attitude reference constraints are violated during the step changes in reference quadrotor position. It is due to the fact that the constraints are relaxed for the first step in N-step prediction horizon as explained in subsection 3-4-4. Since these constraints are not hard constraints, violation of these constraints do not pose any problem to this system. The attitude of the quadrotor are however within bounds as can be seen in Figure 3-6c. This is mainly due to the attitude dynamics specific to the quadrotor used in this thesis and it is not guaranteed that the attitude will remain within bounds. Violation of constraints also happens to be the drawback within this formulation and better constraint handling techniques can be used to achieve feasibility like the use of soft constraints. It is reiterated that these violations are not crucial for this system.

One of the important consequence of using the Δu formulation is that it can be seen as a disturbance estimator where the states pertaining to the previous input $\hat{\mathbf{u}}_k$ are estimated using a Kalman filter. The estimated value need not necessarily be equal to the previously applied value. Hence, in the presence of input disturbances or model mismatch, these unknown effects are lumped into the estimates and hence giving an effective method to tackle the model mismatch in the system. Figure 3-7 shows the plot of applied control input and estimated control input. As can be seen from Figure 3-7a, the estimated values follow the applied control input as no disturbances were acting on the system in X and Y axes. However, the effect of disturbance estimation can be seen in Figure 3-7b where the applied and the estimated thrust values are different. The only disturbance acting along Z direction is the deliberately introduced mismatch in the mass of the quadrotor which resulted in steady state error in altitude. The estimate drops to zero when the quadrotor achieves the desired altitude and the computed control signal is zero. The change in the control thrust Δf computed by the controller is however integrated externally and applied to the quadrotor. Hence the applied the thrust is different than the estimated thrust.

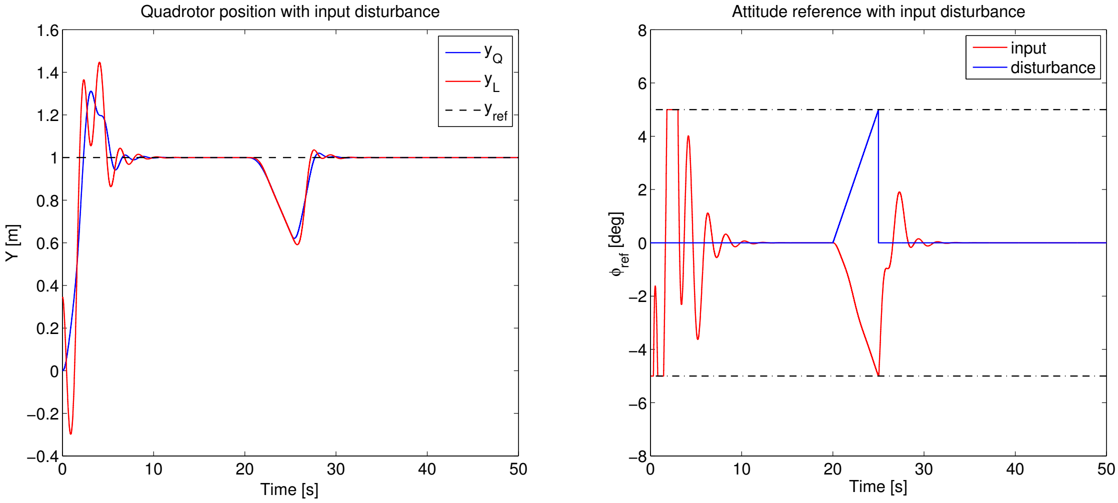
The MPC with Δu formulation has improved performance compared to previous formulations and also indirectly incorporates the disturbance estimation capability. Hence it is also expected to perform better in presence of input disturbances which is discussed next.

3-8-4 Input disturbance rejection



(a) Quadrotor position **(b)** Control input and applied disturbance

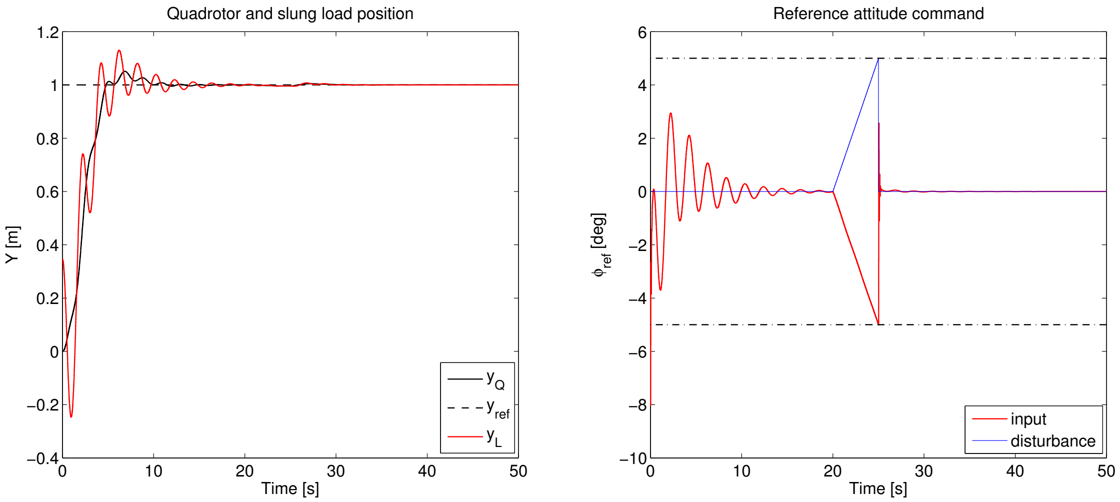
Figure 3-8: Effect of input disturbance on MPC with integral action



(a) Quadrotor position **(b)** Control input and applied disturbance

Figure 3-9: Effect of input disturbance on MPC without integral action

In the previous subsection, the performance of the controller for tracking a step reference was discussed. Another property which is equally important in assessing the controller performance is its disturbance rejection capabilities. Small ticks to the quadrotor or the slung load can be considered as output disturbances and previous section results are sufficient to prove the controllers ability to reject disturbances at the output. Basically, the initial non-zero swing angles can be seen as a disturbance acting on the output (in form of a small tick provided to the slung load). Hence, the focus is on unknown input disturbances acting on



(a) Quadrotor position

(b) Control input and applied disturbance

Figure 3-10: Effect of input disturbance on MPC with Δu formulation

the quadrotor. A ramp like input disturbance was applied on roll angle reference input ϕ_{ref}. Figure 3-8 shows the plot of applied input disturbance and its effect on the position of the quadrotor for MPC-I. As can be seen in Figure 3-8b, the controller is able to generate counter action to the input disturbance, while causing the position to drift away from the reference position as shown in Figure 3-8a. The main result that need to be explained is the effect of integral windup. When the disturbance on input goes to zero, the quadrotor returns to the reference position only after a large overshoot. This happens due to integral action. Figure 3-9 shows the plot of same scenario in absence of integral action. Clearly, from Figure 3-9a the overshoot in position is eliminated. This is one of the drawbacks of the control formulation with integral action.

Figure 3-10 shows the plot of input disturbance applied to the system with Δu formulation. Due to the disturbance estimation property of this controller, it can be seen in Figure 3-10b that the controller is able to produce exactly the counter inputs so that the influence of disturbance on the input is minimal. From Figure 3-10a it can be seen that the input disturbance causes a very small perturbation on the position of the quadrotor. Clearly, MPC with Δu formulation outperforms both the LQ control with integral action and the MPC-I in terms of disturbance rejection and set point tracking.

Table 3-1: Performance comparison of different control formulations

Control method	Rise time [s]	Settling time [s]	Percentage Overshoot	Computation time
LQI - error feedback	2	8	74%	Negligible
LQI - position feedback	9	10	0	Negligible
MPC-I - error feedback	4	7	30%	2 - 6 ms
MPC-I - position feedback	9	10	0	2 - 6 ms
MPC - Δu formulation	5	6	7%	2 - 6 ms

3-8-5 Performance comparison

In the previous subsections, the simulation results of LQ controller with integral action and different MPC formulations were discussed. In this subsection, the results are analysed and compared. Table 3-1 shows the performance comparison of different control methods. The performance is assessed in terms of rise time, settling time, percentage overshoot and computation time. It is worth noting that all the considered control methods are able to eliminate the steady state error. Considering the computation time as a assessment criteria, LQ controller is the clearly a better controller as it is easy to implement. However, the controller is also associated with huge overshoot of 74 % with error feedback which is undesirable. While the overshoot could be eliminated through the position feedback, it could achieve this at the expense of rise time and settling time. Clearly, better methods could be explored to achieve the control objectives. From the simulation results, it can be seen that MPC-I controller performs similar to LQ control with integral action. The only advantage that can be gained is the constraint handling capability. The introduction of integral action into the controllers completely eliminates the steady state error however they also suffer from undesirable effects of integrator wind up as was pointed out in previous subsection related to input disturbances. Given the simulation plots and performance metrics it can be concluded that use of integral action at the output is not a very desirable control method as it is accompanied with large settling time and overshoots. The only advantage being elimination of steady state error at the quadrotor position outputs. However, it does serve as a proof of concept and that the quadrotor-slung load system can be controlled using the LTI MPC control techniques.

In order to mitigate the drawbacks of the LQ/MPC with integral action, Δu formulation was utilized which can be considered as control method with integral action at the input. It can also be considered as input disturbance estimator when the previously applied control inputs are estimated using Kalman filter. From Table 3-1 it can be clearly seen that the overshoot has drastically reduced to 7% compared to other control methods. The performance improvement can also be observed in rise time and settling time. Clearly this control method outperforms the previous control formulations in all aspects. This finding is further cemented by the fact that this controller is able to eliminate both the steady state error and the integral windup effect seen with control methods having integral action at the output. In Section 2-5, the sources of uncertainty were explained and the major source of uncertainty was the input uncertainty. Considering the input disturbance rejection properties exhibited by this control formulation, it seems to be an apt choice for practical implementation.

3-9 Conclusion

In this chapter, the state estimation and control design problem for the quadrotor-slung load system was addressed. It was shown that the Kalman filter was able to reconstruct the states using the LTI model obtained in previous chapter. The control design problem was then addressed with three control formulations, namely, LQ control with integral action, MPC with integral action and MPC with Δu formulation being discussed. The state-of-the-art fast optimization solvers were introduced. The methodology adopted to tune the MPC controllers were presented followed by steps taken to ensure stability of MPC. The simulation results of the control formulations were discussed and their performances were compared. It could be shown that the Δu formulation outperforms the other control methods in all aspects. Furthermore, it is advantageous to use this formulation as it also estimates the disturbances acting on the model and hence giving a better method to tackle model mismatch.

Chapter 4

Experimental Setup and Results

In this chapter, one of the major contributions of this thesis namely the experimental validation is discussed. Previous chapters dealt with modeling and control aspects backed with simulation results. The treatment focused mainly on theoretical aspects of control design. The controllers developed in the previous chapters were validated on a practical setup. This chapter highlights the practical issues encountered in implementation of the control methods.

The chapter is organized as follows. Section 4-1 gives a brief introduction to the practical setup used in this thesis. Section 4-2 describes the control software used. The new software framework developed to implement MPC is also explained briefly. The controllers developed in previous chapter are implemented on practical setup and their results are presented in Section 4-3.

4-1 Hardware

In this section, the hardware components used in the experiments are briefly introduced. Specifically, the quadrotor used for the experiments and external tracking camera's used for position measurement is introduced.

4-1-1 Parrot AR Drone 2.0

The quadrotor used in experiments is the commercially available Parrot AR Drone 2.0. The advantage of this device is that it is portable, light weight and comes with a very impressive set of on-board electronics. The technical specifications of the quadrotor are as follows

1. 1GHz 32 bit ARM Cortex A8 processor with 800MHz video DSP TMS320DMC64x and NEON SIMD instruction set extension

2. Linux 2.6.32 operating system

3. 1GB DDR2 RAM at 200MHz
4. USB 2.0 high speed for extensions
5. Wi-Fi b/g/n
6. 3 axis gyroscope 2000°/second precision
7. 3 axis accelerometer ±50mg precision
8. 3 axis magnetometer 6° precision
9. Pressure sensor ± 10 Pa precision
10. Ultrasound sensors for ground altitude measurement
11. 60 FPS vertical QVGA camera for ground speed measurement

Among all these technical specifications, the most interesting specification is a decent processor with Linux operating system and available on-board memory. The drone with a default firmware on-board can be controlled through an external device like a laptop or a smartphone. The communication with the drone is established through Wi-Fi communication link. The on-board sensors like accelerometer and gyroscopes are used by Attitude Heading and Reference System (AHRS) for attitude estimation.

Here two terminologies are introduced which would be used frequently. One is the on-board computer which refers the processor on the drone. The laptop computer which communicates with the drone is called ground station computer. Ground station computer has a Intel(R) Core(TM) i5-3210M @ 2.50GHz processor running Ubuntu 14.04 operating system.

Quadrotor and slung load model parameters

The only quadrotor parameter of interest is the mass of the quadrotor m_Q which was measured to be 0.451 [kg]. Additional parameters such as Inertia, motor and battery dynamics etc. for Parrot AR Drone quadrotors can be found in [41] and is not used in the model. Other than the parameters of the quadrotor, the parameters of slung load which are necessary are the length of the cable l and mass of the suspended payload m_L which were 1.12 [m] and 0.045 [kg] respectively.

4-1-2 Optitrack system

The output feedback controller designed in previous chapter relies on position of quadrotor and the slung load. Set of external tracking cameras are installed in the Cyberzoo arena which accurately track the position of the quadrotor and the slung load with millimeter accuracy. Reflective markers are attached to the quadrotor and the slung load so that they can be tracked by the cameras. Figure 4-1 shows a picture of the quadrotor-slung load system in the Cyberzoo flight area.

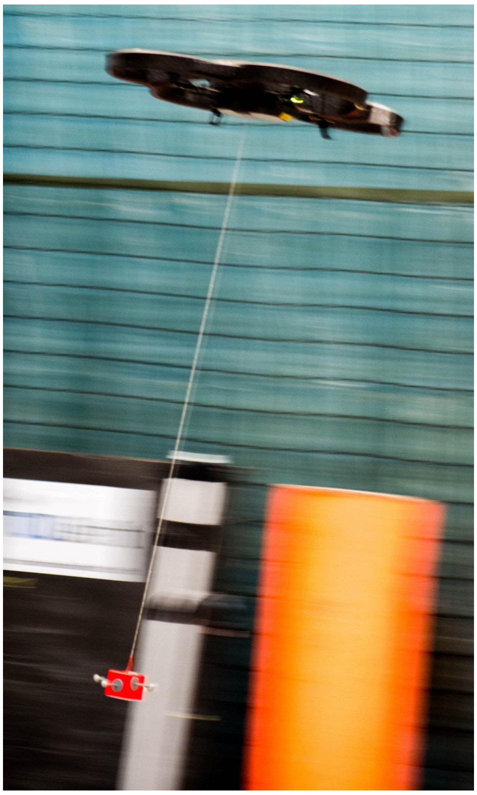


Figure 4-1: Quadrotor-slung load system in action

4-2 Software

Model Predictive Controllers are computationally expensive and state-of-the-art optimization solvers were presented in previous chapter to achieve fast solution times and convergence. A powerful processor is required to execute the efficient code generated by the solvers. Even though the drone has a decent processor, it is still not sufficient to execute the code generated by the solver[1]. A much powerful processor such as the ones found in laptops or desktop computers are required. Consequently, the drone had to be remotely controlled with the controllers being implemented on the ground station computer. This lead to modification of currently available control software and development of a new software framework which allows implementation of MPC for quadrotors. Apart from the research contributions explained in previous chapter, the development of a new software framework is an engineering contribution of this thesis work.

In this section the software framework used to implement MPC is explained. The control software can be divided into two main modules. First module, Paparazzi autopilot software is the one that runs on the on-board computer, sometimes referred to as on-board controller. Second module is the MPC software that runs on the ground station computer, also referred to as off-board controller. The two software modules are discussed next. Figure 4-2 shows the block diagram representation of the experimental setup, with different software modules that

[1] Commercial version of the solver can generate an efficient code specific to the on-board processor i.e., ARM processors which would allow the on-board implementation of controllers

execute on the on-board computer and ground station computer. Blue modules are the one that already existed within Paparazzi and orange modules are the one that were developed in this project.

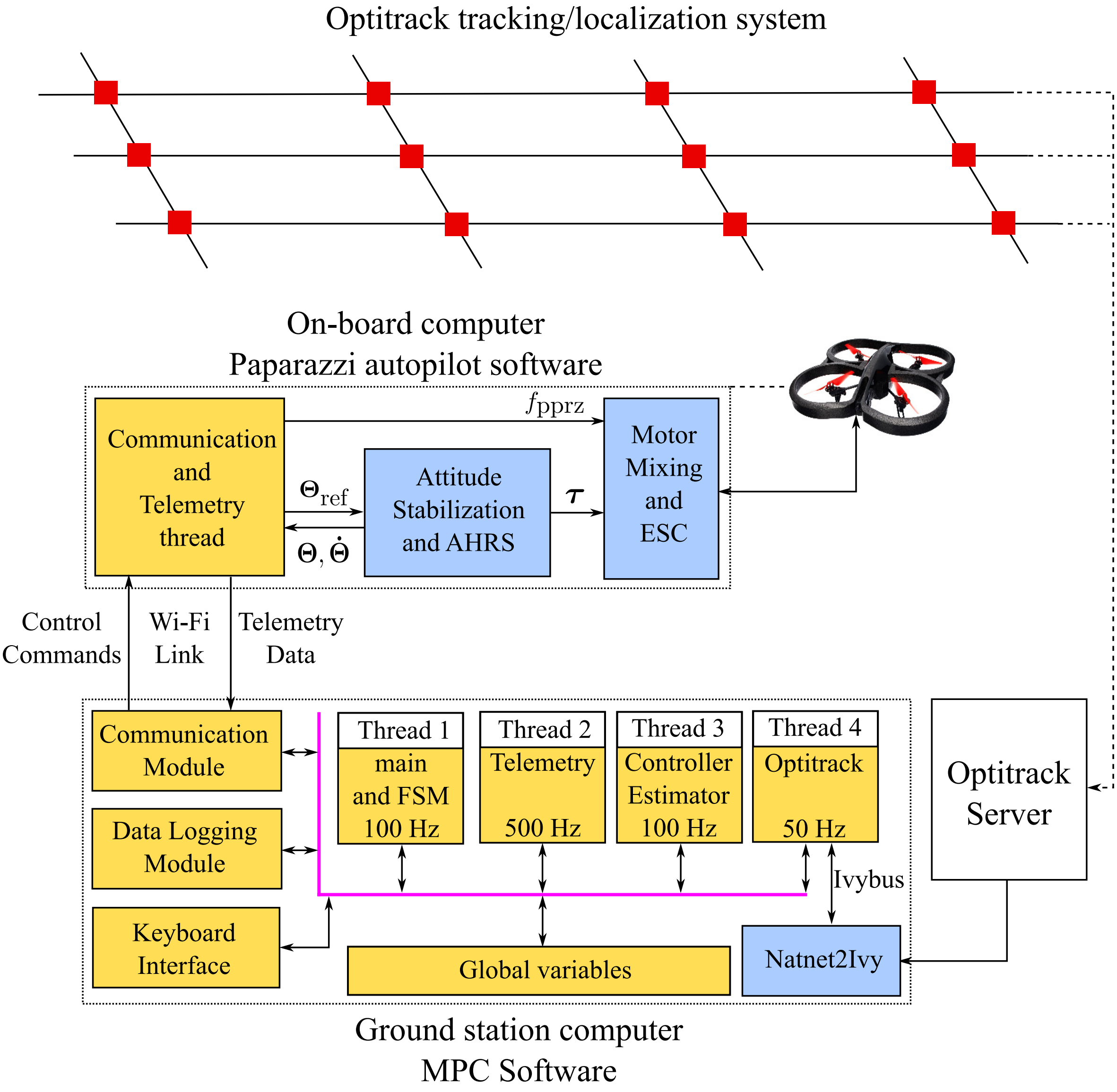


Figure 4-2: MPC software framework

4-2-1 Paparazzi

Paparazzi[2] is an open source autopilot software being actively developed at the Micro Air Vehicle Laboratory of Faculty of Aerospace Engineering, Delft University of Technology. It is an extremely versatile autopilot software supporting various fixed wing aircrafts and rotorcrafts

[2] http://wiki.paparazziuav.org/wiki/Main_Page

such as the Parrot AR Drone 2.0 used in this thesis. It is a collection of algorithms for guidance, control, navigation and estimation written in C. These algorithms include variants of PID-FF controllers for attitude stabilization, PID controllers for position control (also called guidance loop) and AHRS algorithms for attitude estimation. It also establishes communication with the ground station computer and handles the communication channel in order to receive higher level navigation commands, send telemetry data. The control and estimation algorithms are executed on-board the quadrotor. The software also provides a GUI called as the Paparazzi Center, which runs on a ground station computer. The Paparazzi Center allows the user to customize flight plans, plot the telemetry data in real time, log the navigation data of the UAV during the flight, provide instructions to the UAV (such as takeoff, land etc.), build and upload the firmware to UAV and many more.

Clearly, the Paparazzi software is intended to implement the controllers on-board the drone and provide higher level interfaces from ground station computer. This leads to a conflicting scenario where it is not possible to implement MPC on-board and Paparazzi does not support or provide an interface to communicate with the off-board controller. Hence modifications had to be made to the existing Paparazzi implementation to allow for a remote/off-board controller to communicate with the Paparazzi autopilot software forming a real-time networked control loop. As explained in Chapter 2, the decoupled rotational and translational dynamics of the quadrotor lead to a design strategy of two cascaded controllers, one for attitude stabilization and the other for position control (See Figure 2-3). Hence the idea is to retain the attitude stabilization loop of the paparazzi software and communicate the reference commands from a remotely computed position controller.

A new thread was launched which establishes an User Datagram Protocol (UDP) socket connection though Wi-Fi and receives control commands from the MPC software at 100 Hz. It also sends the telemetry data at 500 Hz which contains the attitude and angular rate information of the quadrotor. The off-board controller (MPC software) sends the control thrust in paparazzi units f_{pprz}, roll and pitch angle references ϕ_{ref} and θ_{ref} in radians as can be seen in Figure 4-2. Additional information such as heading angle measured from the optitrack system is received from the MPC software.

4-2-2 MPC software framework

Now that the need for implementing the MPC remotely has been motivated, the newly developed MPC software is briefly discussed in this section. Figure 4-2 gives a pictorial representation of the developed framework. The position measurement from the optitrack system is parsed and sent over a software bus called the ivybus[3] by the `natnet2ivy` process. The MPC software is another process executed in parallel with the `natnet2ivy` application. The MPC software or the off-board controller is a multi-threaded application written in C which achieves the following

- Creates a dedicated UDP socket on the ground station computer and establishes communication with the drone over Wi-Fi. The control commands computed by the controller are sent to the drone and telemetry data from the drone is received through Linux socket interface [49, 50].

[3] www.eei.cena.fr/products/ivy/

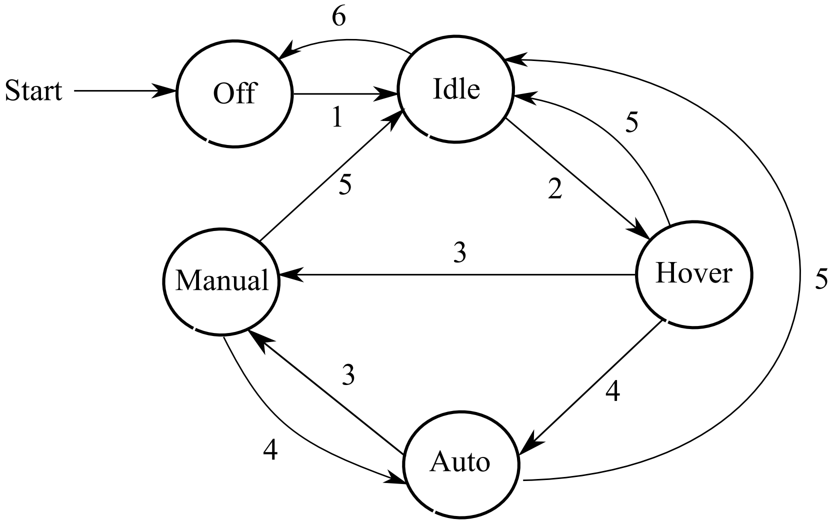


Figure 4-3: Finite State Machine used in the MPC software

- Receive and parse data obtained from the `natnet2ivy` application. A software Ivybus is setup and configured to listen to a particular message string. The string is then parsed in 'thread safe' manner.
- Parse the telemetry data obtained from the drone.
- Data logging and keyboard interface for manual control.
- Execute the controller and Kalman filter for state estimation. The task of control computation, data logging, parsing the telemetry data are achieved through multiple POSIX threads.
- Implements a finite state machine for safe operation of the drone. More details regarding finite state machine is given in the following subsection.

Finite state machine

When trying to implement a controller for complicated system like the quadrotor-slung load system, it is usually not possible to just execute the controller at start of the experiment. It is important to initialize the system/software and get the system to a predefined safe configuration before executing/testing the controller. In the case of quadrotor-slung load system it is important to get the quadrotor to hover at a sufficient altitude to have a completely suspended payload before executing the controller to transport the load from one point to another. This requires a sequence of actions, like allowing the quadrotor AHRS to align (else the attitude stabilization loop will not work as there is no attitude feedback), initialize the motors, take off and hover at constant altitude, land when the experiment is finished etc. It is possible during testing, that the controller drives the quadrotor unstable. In such scenarios, there must be a way to switch to a safe behavior to avoid any damage to the equipment. In order to achieve the capabilities mentioned above, the behavior of the control software needs to change at various stages during the experiment. One of the ways to achieve this ability in

Table 4-1: Events of the state machine

Event Number	Event
1	Event_Init
2	Event_Takeoff
3	Event_Manual
4	Event_Automatic
5	Event_Land
6	Event_Kill

a reliable manner is to use a finite state machine. Figure 4-3 depicts the finite state machine used in this thesis. The numbers in the figure corresponds to the occurrence of a 'event'. There are six possible events shown in Table 4-1 that can occur and are provided by the user through the keyboard interface in the current implementation.

The state machine has five different states and the switching between them through occurrence of events as shown in Figure 4-3 is self explanatory. Hence only the behavior of the software at different states are explained.

- **State Off**: This is the default state of the quadrotor after the paparazzi firmware has been uploaded to the drone. In this state, the user basically waits for the AHRS to align and receive telemetry data.
- **State Idle**: Once the AHRS is aligned, the system is ready to fly. With occurrence of `Event_Init`, the quadrotor motors are turned on and the state transitions to the `Idle` state. In this state the whole cycle of sending control commands at 100 Hz and receiving telemetry commands at 500 Hz is being executed in parallel. The data from the optitrack system is also parsed. The system is now ready for takeoff.
- **State Hover**: Once `Event_Takeoff` is issued, the state of the system transitions to `Hover` mode, where a PI controller is used to bring the quadrotor to hover at a constant altitude. Once the quadrotor is hovering at a constant altitude with the slung load completely suspended, the transition can be made to either `Automatic` mode or `Manual` mode depending on the event that has been issued.
- **State Auto**: This mode is the `Automatic` mode which can be used to integrate a flight plan or allow the controller to follow a predefined trajectory. In the current implementation, the system hovers at a constant set-point while stabilizing the swinging mass (if the slung mass is perturbed). In this mode and the manual mode the entire cycle of estimation and control is executed.
- **State Manual**: In this mode the estimation and control cycle is still executed while the position set points can be changed by the user through keyboard input. Basically the user can navigate the quadrotor in 3D space inside the flight area.
- In order to recover from instability or any other issues such as low battery, the system can be switched to `Idle` state through issue of `Event_Land`. The quadrotor slowly descends to ground and the once the quadrotor has landed, the motors can be killed using the `Event_Kill`.

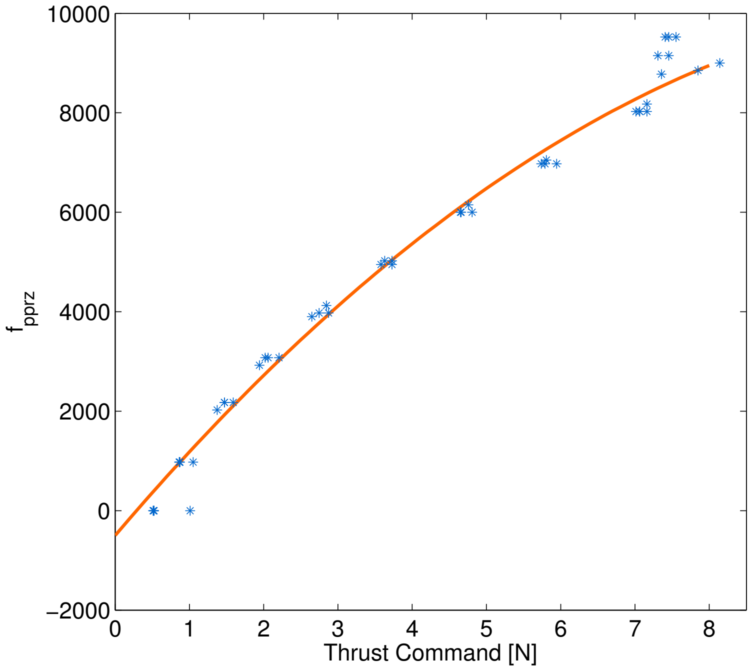


Figure 4-4: Static map relating the computed thrust command f [N] with the applied thrust command f_{pprz}. The blue dots are the measured values and the orange curve is the plot of static map

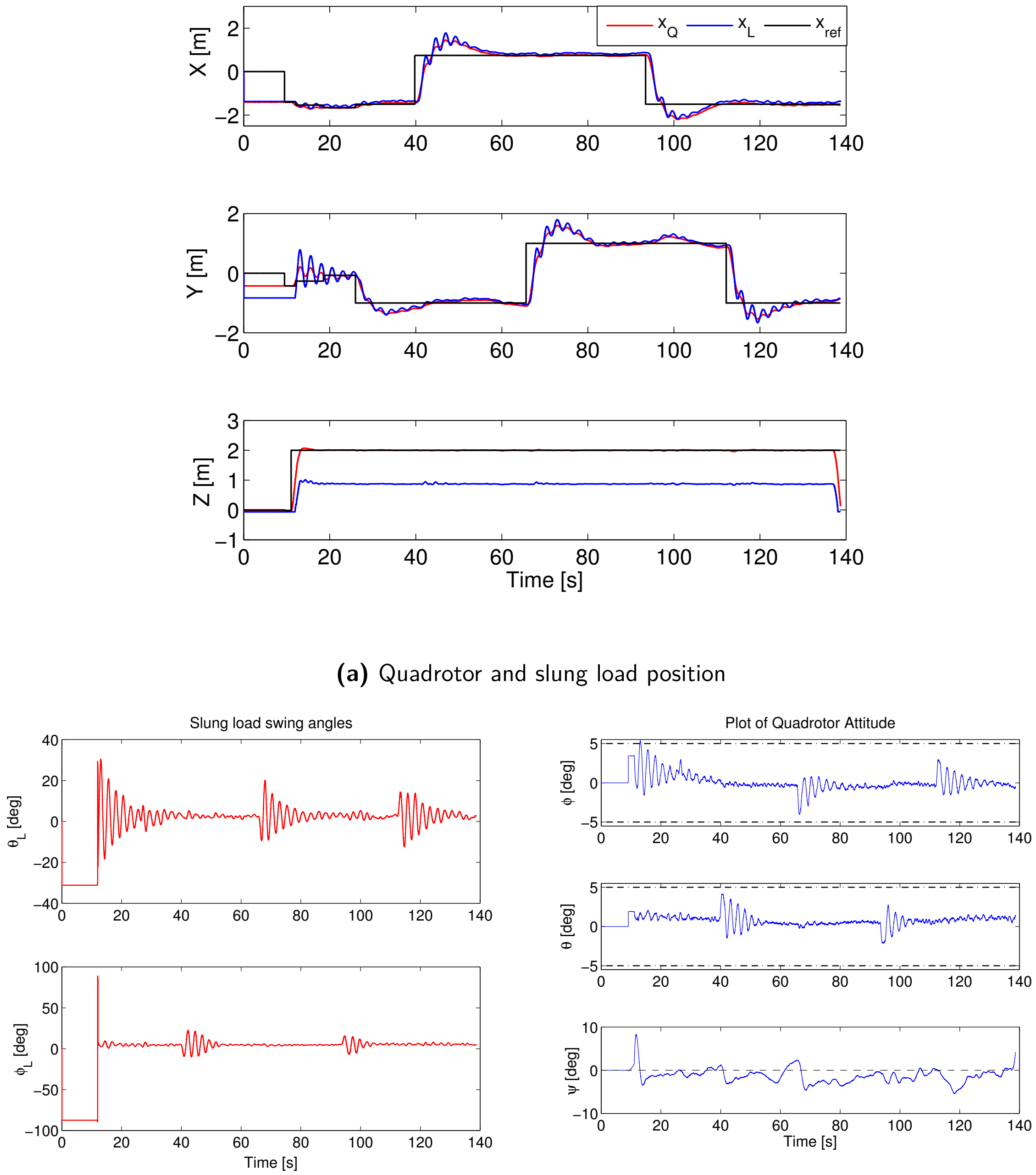
Thrust to Paparazzi Unit conversion

The controllers designed in the previous chapter compute thrust input commands f in Newtons. However, within the Paparazzi software, the thrust command is issued in terms of a number ranging from 0 to 9600 corresponding to minimum and maximum thrust respectively. This thrust command is denoted by f_{pprz}. An experiment was conducted previously by another master student [41] where the quadrotor was placed upside down on a weighing scale and thrust input was applied. The applied thrust command f_{pprz}, the measured battery voltage and measured weight on a weighing scale were recorded. The readings are given in Appendix B for future reference. A second degree polynomial was fit to the measured data and a static map was obtained between the thrust f in Newton and f_{pprz}. The static map is given in Eq. (4-1) and plot of the measured data and simulated data is shown in Figure 4-4. The static map has a good fit in the operating range of 4-5 [N] during hover.

$$\begin{aligned} f_{\text{pprz}} &= a_3 f^2 + a_2 f + a_1 \\ &= -71.2 f^2 + 1751 f - 497.5 \end{aligned} \tag{4-1}$$

4-3 Experimental results

4-3-1 Linear Quadratic Control with Integral action

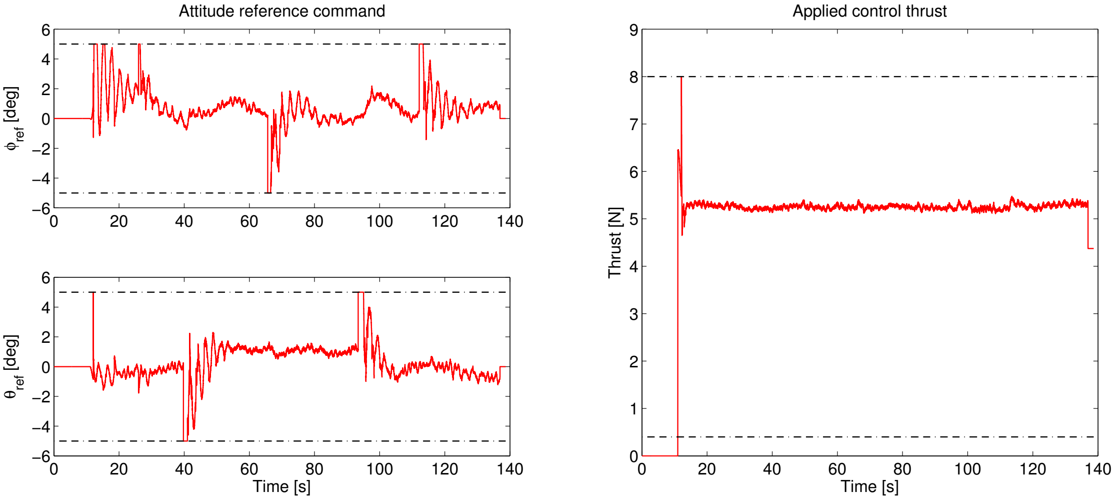


(a) Quadrotor and slung load position

(b) Swing angles of the slung load

(c) Attitude of the quadrotor

Figure 4-5: Linear Quadratic control with integral action with position error of the quadrotor $\mathbf{x}_Q - \mathbf{x}_{Q\text{ref}}$ as feedback



(d) Attitude reference

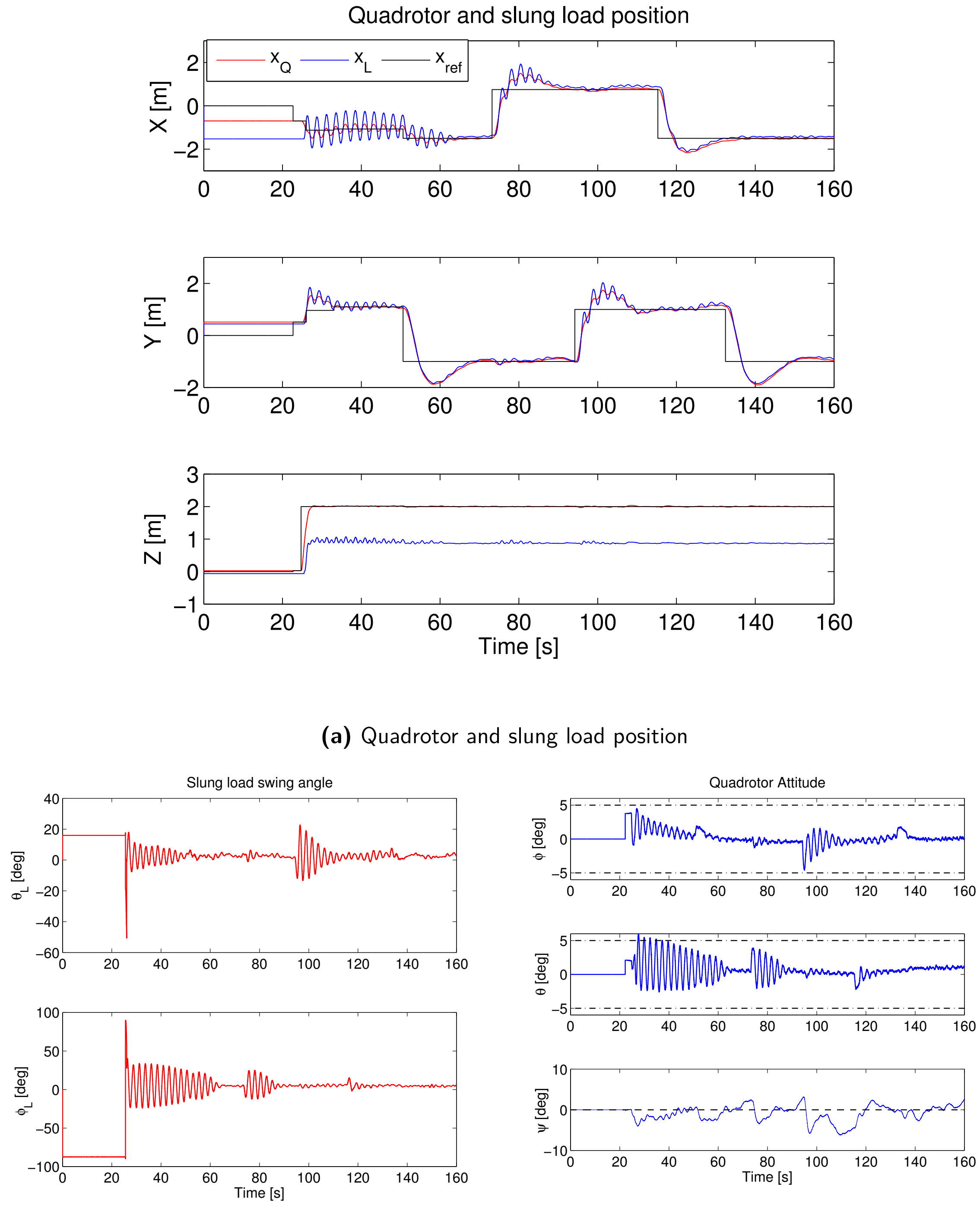
(e) Control thrust

Figure 4-5: Linear Quadratic control with integral action with position error of the quadrotor $\mathbf{x}_Q - \mathbf{x}_{Q_{\text{ref}}}$ as feedback

Figure 4-5 shows the experimental results of LQ control with integral action with quadrotor position error provided as feedback. Figure 4-5a shows the plot of the quadrotor and slung load position. Clearly from the Figure 4-5a and Figure 4-5b, it can be seen that the controller is able to transport the load from one point to another without swing. However, when the quadrotor reaches a desired set point, oscillations can be seen in the slung load which is eventually damped by the controller. The controller is also able to track the quadrotor position with zero steady state error. Small deviations in position (Y position at around $t = 100$ seconds) are mainly due to the unknown disturbances and small oscillations in the slung load. One of the characteristics noted in simulation is the large overshoot in the quadrotor position due to integral action. This effect can also be seen in the experimental results. However the effect is more dominant in experimental results with a overshoot of 100%, before settling to the desired value with settling time of 18 seconds. The swing angles of the slung load is bound within 10 degrees throughout the transport and reaches a maximum swing of approximately 20 degrees during transients. Figure 4-5c shows the plot of quadrotor attitudes and one notable difference is the non zero yaw angle. The yaw dynamics is not captured within the model and it is assumed that the yaw angle is always zero. This discrepancy between the assumptions and real data is also a source of uncertainty.

High weights were placed on the control inputs in order to have the control values of Figure 4-5d within the bounds before being applied to the system. At some instances the control values are externally bounded and these constraints are not handled by the controller itself.

4-3-2 Model Predictive Control with Integral Action

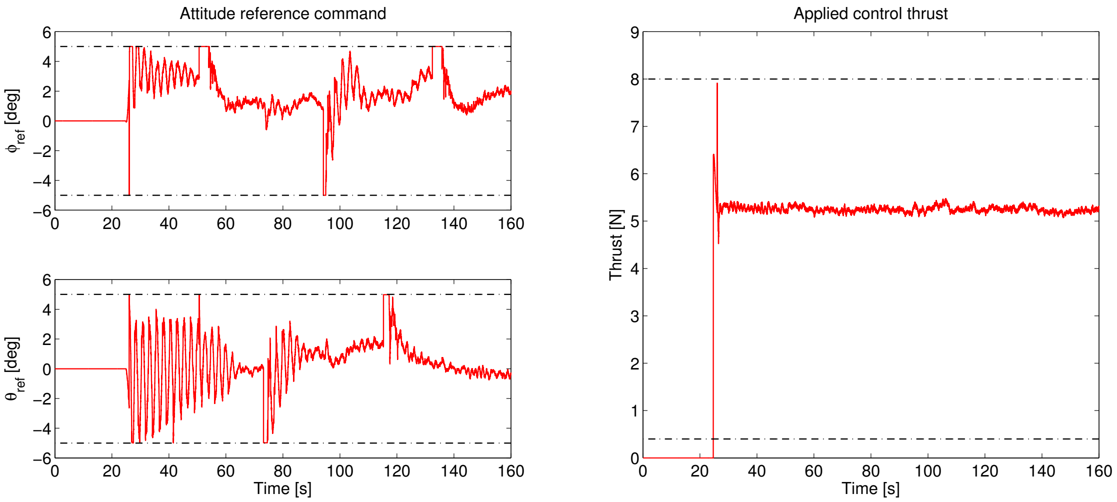


(a) Quadrotor and slung load position

(b) Swing angles of the slung load

(c) Attitude of the quadrotor

Figure 4-6: Model Predictive Control with integral action with position error of the quadrotor $\mathbf{x}_Q - \mathbf{x}_{Q_{\text{ref}}}$ as feedback



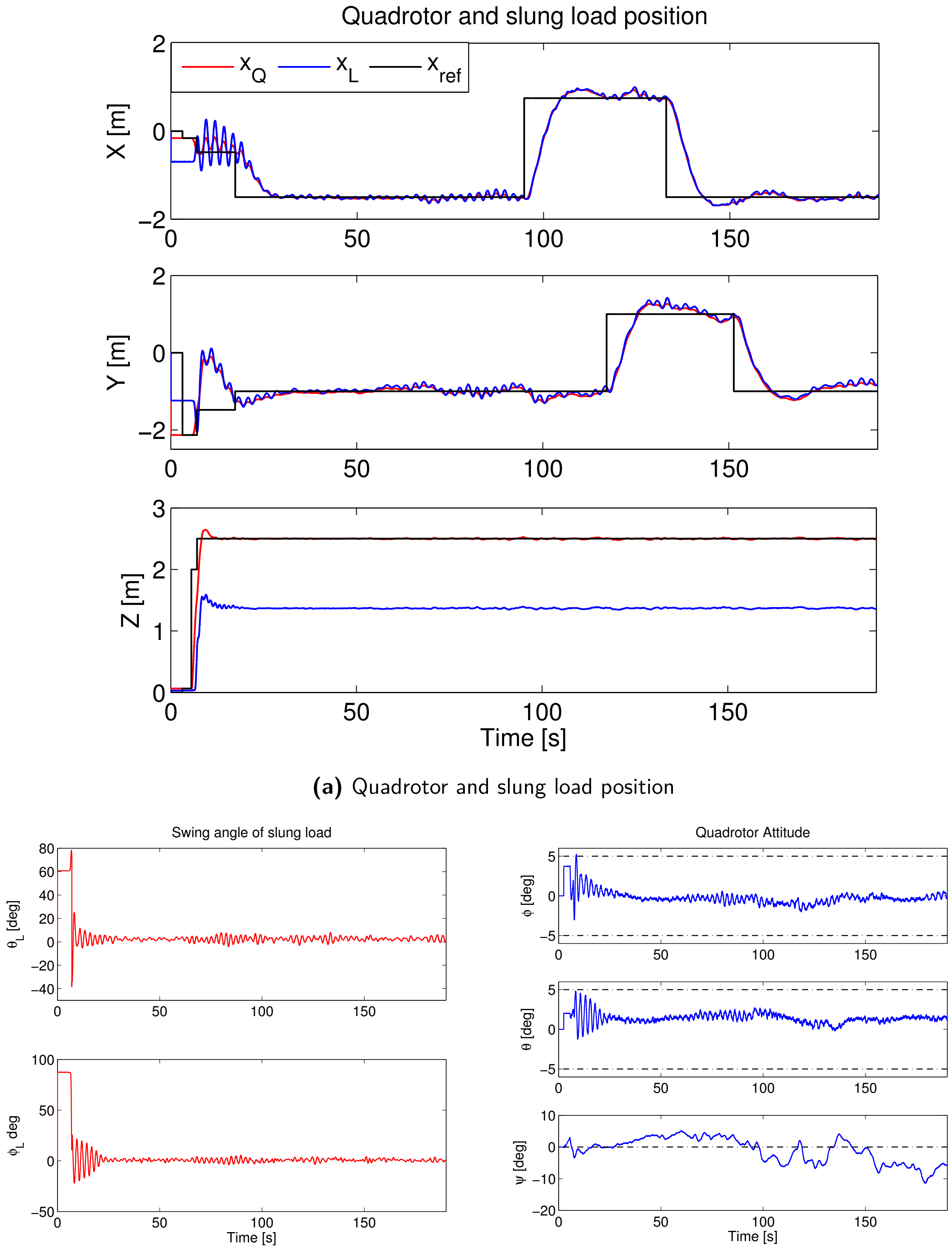
(d) Attitude reference

(e) Control thrust

Figure 4-6: Model Predictive Control with integral action with position error of the quadrotor $\mathbf{x}_Q - \mathbf{x}_{Q_{\text{ref}}}$ as feedback

Figure 4-6 shows the experimental results of MPC with integral action with position error of the quadrotor as feedback. From the simulation results it can be expected that the results of MPC with integral action would be identical to the results of LQ control except the constraint handling. Figure 4-6a shows the plot of quadrotor and slung load position. As expected the controller is able to stabilize the slung load similar to the LQ control with integral action. The rise time and settling time are 4 seconds and 16 seconds respectively which is similar to that of LQ controller. The overshoot of 73 % is however lesser than that of the previous controller. The swing angles too are within the bounds of ± 10 degrees and reaches maximum of approximately 20 degrees during transients as can be seen in Figure 4-6b.

An oscillatory behavior in the slung load can be noticed during the time 20-50 seconds in Figure 4-6a (X axis) and Figure 4-6b (ϕ_L). This is one of the drawback/limitation of this method and the controller is not always to able to damp out the oscillation. Instead it seems that the controller is inducing the oscillations into the system. However when the $x_{Q_{\text{ref}}}$ reference set point is changed at around 50 seconds, the controller damps these oscillations while reaching the desired set point. Another effect that can be observed in the quadrotor and slung load position plot in Y axis is that the slung load swings during transients at around 30 seconds and 100 seconds, however there are no oscillations in slung load at 60 seconds and 140 seconds during transients. This behavior could not be completely explained and could be attributed to the fact that there is a lot of uncertainty with the system as explained in Section 2-5 of Chapter 2. One major source of mismatch or uncertainty not explained before is violation of assumption that the suspension point of the slung load coincides with the center of mass of gravity and hence the origin of body fixed frame. In practice we do not have accurate knowledge of the center of mass of the quadrotor and hence the origin of body fixed frame nor the position of the suspension point with respect to the body fixed frame (assumed zero in the model).

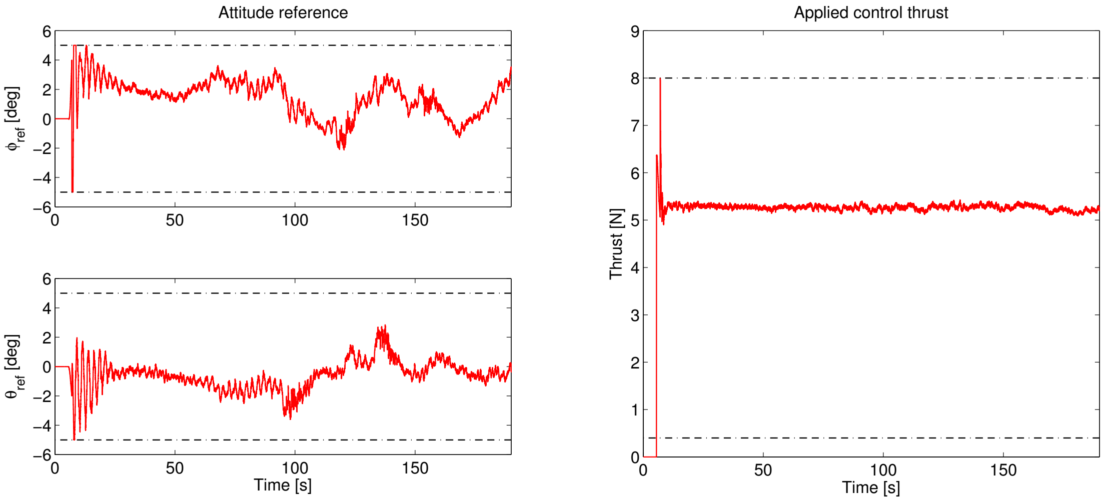


(a) Quadrotor and slung load position

(b) Swing angles of the slung load

(c) Attitude of the quadrotor

Figure 4-7: Model Predictive Control with integral action with position of the quadrotor $\mathbf{x}_Q$ as feedback



(d) Attitude reference

(e) Control thrust

Figure 4-7: Model Predictive Control with integral action with position of the quadrotor $\mathbf{x}_Q$ as feedback

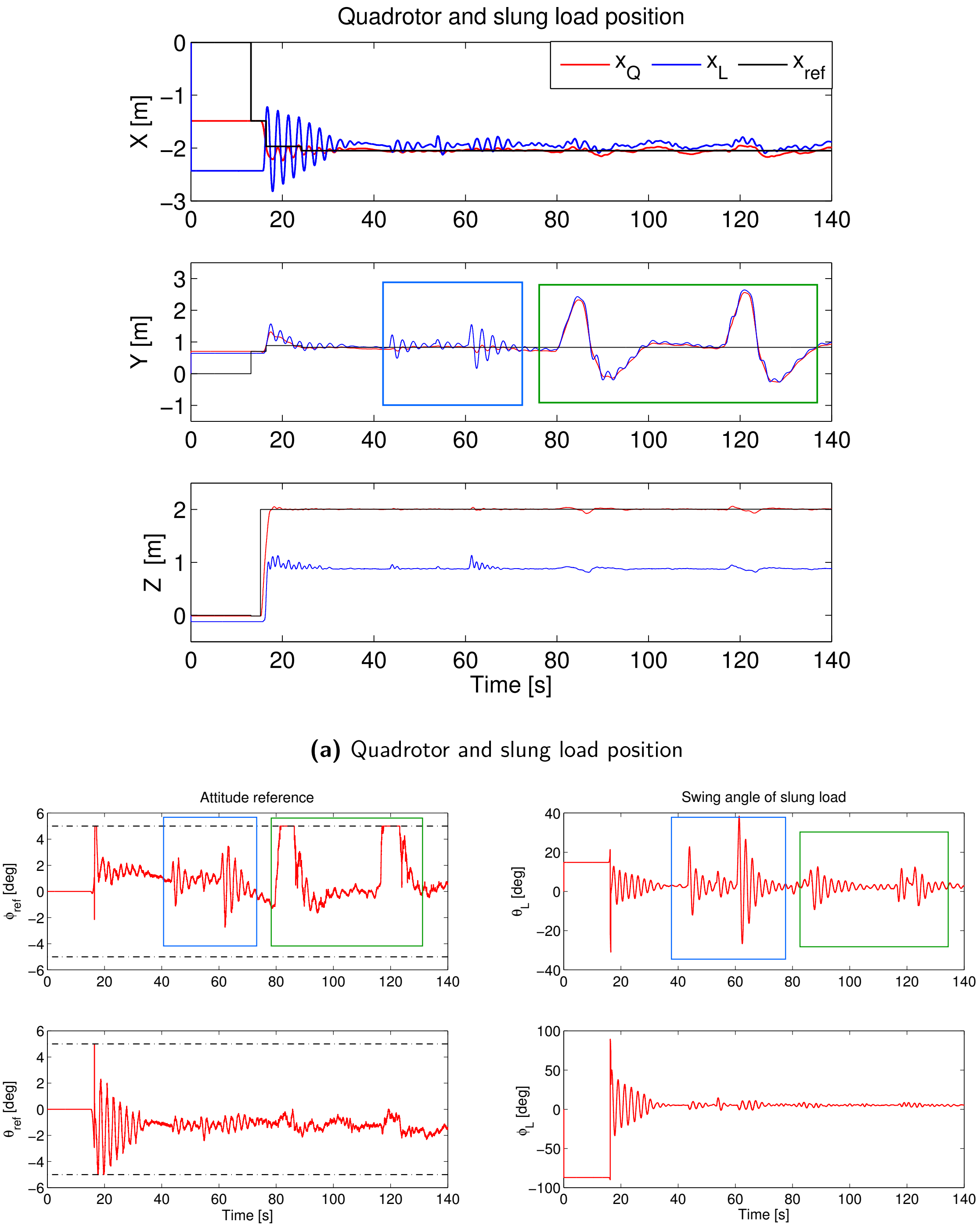
Consequently, there are additional forces and torques acting on the quadrotor not captured by the model. If the center of mass of the quadrotor is not exactly at the geometric center of the quadrotor, then the swinging mass can also alter the center of mass during flight which is a source of uncertainty. The oscillations in the slung load can be attributed to these effects. This is certainly a limitation of the work in this thesis and needs further analysis. It must also be noted that these effects can only be mitigated by control design. It was explained in previous chapter that the overshoot due to integral action can be eliminated by using position of the quadrotor as feedback.

The experimental results of MPC with integral action with position of quadrotor as feedback is shown in Figure 4-7. The performance of both the MPC with integral action and LQ control with integral action given the position of the quadrotor $\mathbf{x}_Q$ as feedback are similar. As expected, the only advantage is the reduction in overshoot to 24 %. For a step reference position of 0.75 m, overshoot of 24 % would correspond to the quadrotor achieving a maximum value of 0.93 m i.e, 18 cm deviation from the set point, which does not seem much of a deviation considering the dimensions of the drone and the primary objective being to reduce the swing of the slung load. The rise time and settling time are 10 seconds and 20 seconds respectively which is slower than the previous two control results. It is expected as the drone relies only on the integral states to achieve tracking.

Given the simulation results, the disadvantages of the control formulation with integral action can be mitigated through the use of Δu formulation. However, the controller could not be implemented on the real system due to practical issues and lack of time.

4-3-3 Input and output disturbance

The performance of the controller with input and output disturbances are discussed in this subsection. The plots presented in this section are obtained with MPC with integral action.



(a) Quadrotor and slung load position

(b) Attitude of the quadrotor

(c) Swing angles of the slung load

Figure 4-8: Effect of input disturbances (green box) and output disturbance (blue box) on quadrotor-slung load system with MPC with integral action

Large perturbations to the slung load was provided as output disturbance as can be seen within the blue box in Figure 4-8c and Figure 4-8a. Clearly the controller is able to reject these disturbances. These disturbances are rejected while maintaining the same quadrotor position as can be seen in Figure 4-8a. In fact the influence of the swinging load can be seen as disturbances acting on the quadrotor which are very well rejected.

A ramp like input disturbance was provided at input to the quadrotor and simulation results were analyzed. It was found that the control methods suffers from integral windup effect due to the integral states. The same experiment was reproduced with practical setup by persistently pushing the quadrotor away from the desired position for some duration. Result can be seen within the green box of Figure 4-8a. The integral windup effect can be clearly noticed. Considering the similarities between the simulation and the practical results, it is expected that the Δu formulation would perform better in practice in presence of input disturbances.

4-3-4 Practical results versus Simulation results

Now that both the practical results and the simulation results have been discussed, it is logical to compare them to gain additional insights. In order to have common grounds for comparison, both the practical and simulation results were obtained with same tuning values. All the major traits such as large overshoot, integral windup, slower response could be found in both the practical and simulation results. However, the rise time, settling time and overshoot were different with the practical results producing larger overshoot values and slower transient response compared to the simulation results. In simulation, once the slung load is stabilized, the controller is able to transport the load from one point to another without any oscillations during transients when there is change in reference quadrotor position. This is not true however in practical experiments, where oscillations were induced in the transients. All these discrepancies can be explained considering the uncertainties present in the system. The slung load behavior is more oscillatory in experiments when compared to simulations and it is due to the fact that the model used in simulation does not capture effects such as propeller wash acting on the slung load. The mass of the payload used in this work is 0.045 kg and it is understandable that it is sensitive to external disturbances. The effects could be different with heavier load, where its influence on the quadrotor would also be prominent. Use of a better (in terms of accuracy) model will certainly help in design of better control laws. At the same time, it must be noted that modeling the quadrotor-slung load system can be a very challenging task and from practical point of view it is impossible to model all possible uncertainties.

4-4 Conclusion

In this chapter the practical aspects of the thesis were discussed. Section 4-1 gave a brief introduction to the equipment used in this thesis. The control software was described in Section 4-2 and the newly developed software framework was explained. The practical results were discussed in Section 4-3 and it was shown that the controllers developed in this thesis are able to successfully transport the slung load from one point to another. It was observed that the major traits of the control performance obtained in simulation results could also be observed in the practical experiments. However, there were some discrepancies which were due to the uncertainty in the model.

The takeaway message is that the model of the quadrotor-slung load system developed in Chapter 2 is suitable for controlling the quadrotor-slung load system using linear control methods. Off course there is scope to improve the accuracy of the model which in turn would influence the control performance. Also it was experimentally validated that the rigid cable approximation made in the model holds true as long as the motion of the quadrotor is not aggressive. The assumption that the suspension point and the quadrotor center of mass is coincident is not true and there is always a non zero, unknown vector directed from the quadrotor center of mass to the suspension point of the slung load. This results in additional forces acting on the quadrotor not captured by the model. Consequently, the MPC is not able to foresee these effects. However, MPC shows some robustness to these uncertainties and is able to stabilize the slung load. There is plenty of scope to improve the control performance and one worthy candidate is to implement the Δu formulation presented in the simulations.

Chapter 5

Conclusions and Future Work

The main goal of this thesis was to investigate the application of LTI MPC techniques to transport a cable suspended load using quadrotor from one point to another while minimizing the swing. The aim and motivation to pursue research in this direction was explained in Chapter 1 supported by the survey of state-of-the-art methods. Special emphasis was given to practical validation of the research findings. Consequently, the work in this thesis makes contributions on two fronts, one being the contributions in applied research and the other being the engineering contribution. The results are summarized next followed by the contributions of this thesis work.

5-1 Summary

Quadrotor-slung load system is a highly nonlinear, unstable and under-actuated system making the control design task a challenging problem. In order to be able to design model based controllers, the dynamics of the system had to be modeled. The model would not only be used for simulation purposes but also for control design which would be experimentally validated. Hence, a simple model was required which captures the dominant dynamic behavior of the system and still is useful for controller design. A model of the quadrotor-slung load system was obtained using the Euler-Lagrange method based on certain simplifying assumptions. The most important assumption which led to simplified dynamics was the approximation of the suspended payload as a spherical pendulum. Similar to the dynamics of the quadrotor, it was found that the translational dynamics of the quadrotor-slung load system is decoupled with the rotational dynamics of the quadrotor. Consequently, two decoupled controllers can be designed for attitude control and position control of quadrotor respectively. The PID-FF controller within the paparazzi software was used for the attitude control of the quadrotor and hence the controlled attitude dynamics were identified. The quadrotor-slung load dynamic model which is partly identified and partly obtained from the first principles is thus obtained. The nonlinear model was linearized around the hover conditions and discretized to obtain a LTI model which would be used for control design.

In order to assess the quality of the obtained model, the model was validated using test data recorded with an LQ control with integral action in the loop. The challenges in identification of a system such as the quadrotor-slung load system were explained. The similarities and discrepancies in the model were explained and the sources of uncertainty were outlined. From practical viewpoint, the major source of uncertainty is the unknown input uncertainty which causes the error in simulated and measured model. It was found that the modeling assumptions do not hold true in practice and it is often difficult or impossible to model the uncertainties. Hence, controllers must be designed to handle such unknown uncertainties present in the system.

Considering the uncertainties in the system, the output feedback control design problem was addressed with three control formulations, namely, the LQ control with integral action, MPC with integral action and the MPC with Δu formulation. Kalman filter was designed to estimate the states of the system given the applied control inputs, measured outputs and the LTI model of the plant. Integral action was introduced in the controller in order to account for the model mismatch and eliminate the steady state error at the quadrotor position output. It was found that the LQ/MPC with integral action was able stabilize the slung load and eliminate the steady state error. However, the controller also produced a large overshoot and demonstrated the effect of integral windup with unknown disturbances acting on the input. To alleviate these limitations, Δu formulation was proposed. It was found that the MPC with Δu formulation outperforms the LQ control with integral action and MPC with integral action. Advantages of the Δu formulation was evident when its input disturbance rejection capabilities were compared with the other formulations. Simulation results served as proof of concept and provided an important insight that the quadrotor-slung load system can be controlled using LTI MPC techniques. All that remained was to experimentally validate the developed controllers.

In order to be able to implement MPC in real-time for quadrotor-slung load system, a new software framework was developed in C which implements the MPC and state estimator on a ground station computer and communicates the control commands to the drone over Wi-Fi resulting in a real-time networked control loop. The need for development of such a software framework was motivated and is the engineering contribution of this thesis. The software framework would allow implementation of not just LTI MPC, but also the LTV and nonlinear variants of MPC for a single quadrotor. It also provides a way to communicate with the Paparazzi autopilot software running on-board the drone. Experiments were conducted which prove the working of the software framework.

The experimental results provide additional useful insights into the quadrotor-slung load system. It was experimentally validated that the LQ control with integral action and the MPC with integral action formulations are able to stabilize the slung load and transport them from one point to another while minimizing the swing. This shows that the spherical pendulum assumption of the slung load is valid as long as the motions are not aggressive. The practical results and simulation results were compared. Similarities and discrepancies were explained. All the major performance traits of the controllers found in simulation results were observed in experiments. However, the practical results indicate additional effects (not observed in simulations) such as oscillation of slung load during transients, larger overshoots and slower response times. These effects could be attributed to the assumption that the suspension point of the slung load and the quadrotor center of mass is coincident is not true and there is always a non zero, unknown vector directed from the quadrotor center of mass to the suspension point of the slung load. This results in additional forces acting on the quadrotor and possible perturbations in the center of mass of the quadrotor not captured by the model.

The major disadvantages of the control formulation with integral action were large overshoots and integral windup effects. These effects were demonstrated on the practical setup and hence it is expected that the Δu formulation will result in improved performance. Due to lack of time and practical issues, the Δu formulation could not be validated on the practical setup.

5-2 Thesis contributions

Having summarized the results, the major contributions of this thesis are as follows.

1. Model Predictive Controllers were developed based on LTI model of the quadrotor-slung load system and it was shown through simulations that it is possible to transport the slung load in a swing free manner using quadrotor. Furthermore, the control methods were experimentally validated in presence of uncertainties in the system. Considering the state-of-the-art methods surveyed in the beginning of the report, this is the first practical implementation of MPC for transportation of slung load using quadrotor to the best of my knowledge.

2. A new software framework was developed which allows implementation of MPC for quadrotors. LTI, LTV and nonlinear variants can be easily implemented as long as the optimization solvers compute control inputs within the sampling frequency. Note that the sampling frequency is not fixed and can be easily changed within the framework.

5-3 Future recommendations

In order to provide future recommendations and extension to the work in this thesis, let us point out the limitations of the control methods presented in this thesis.

1. The control methods are based on LTI models and hence associated with only slow motions. Also from the transient response of the controllers it can be noticed that the bandwidth of the system is low. As a result these control methods will certainly not work for aggressive motion.

2. The modeling assumptions are usually not true and it would be beneficial to have these assumptions relaxed. For example, if the position of the suspension point is known with respect to the origin of body fixed frame, it can be incorporated into the model. However, in this case it is not known and hence, it would be better to handle these uncertainties in the control design itself.

3. The nonlinear model of the quadrotor-slung load system could not be validated and needs further analysis. A better model would definitely improve the control performance.

Given the limitations, the future recommendations are now provided and can be classified into practical and theoretical extensions.

Theoretical extensions

The possible theoretical extensions are

1. **LTV or Nonlinear MPC**: The limitation of the implemented controllers are that they are limited to slow motions. The response times can be improved and aggressive motions with the slung load can be demonstrated through the use of LTV MPC or Nonlinear MPC.
2. **Trajectory generation**: Trajectory generation problem can be posed as an Optimal Control problem and state and control reference trajectories can be generated for aggressive motions. The generated reference trajectories can be used with LTV MPC for tracking. See [35] for survey of trajectory generation methods
3. **Hybrid system**: The quadrotor-slung load system can be modeled as a differentially flat hybrid system as shown in [18]. Hence it may be possible to model the system as a Mixed Logical Dynamical (MLD) system and the MLD-MPC methods given in [44, 51] could be explored. Using MLD-MPC, it could be possible to execute aggressive maneuvers like passing through a window by swinging the load.

Practical extensions

The possible practical extensions are as follows

1. **Δu formulation**: The first obvious extension to the work presented in this thesis is to implement, validate and analyze the performance of the Δu formulation. Since the formulation can be considered as a disturbance estimator at the input, it is expected to result in improved performance.
2. **On-board implementation**: The controllers developed in this thesis can easily be ported on-board by using the commercial version of FORCES Pro.
3. **Outdoor trials**: The controllers developed in this thesis only act on the position measurements and the states are estimated by the Kalman filter. Dependence on the external tracking cameras can be eliminated through the use of differential GPS sensors on the drone and the slung load. Other sensing methods such as a downward facing camera could be used along with GPS to obtain position of the slung load.
4. **LTV or Nonlinear MPC**: Demonstrating aggressive motions with the slung load through the use of LTV MPC or Nonlinear MPC is certainly a challenging research problem that can be addressed and experimentally validated. The software framework developed in this thesis can be easily used to implement the advanced controllers.

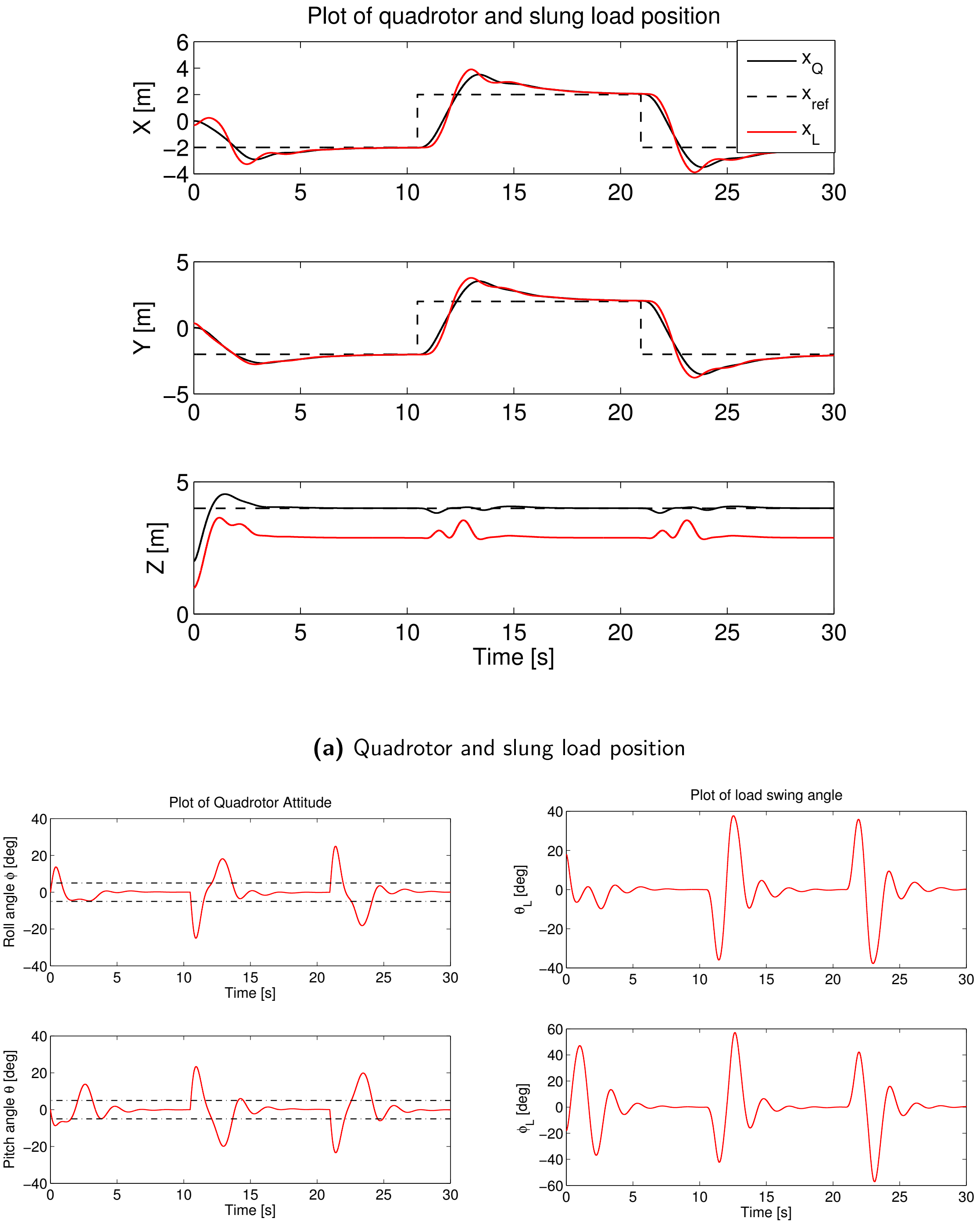
Appendix A

Additional simulation plots

In this appendix, the additional simulation plots are presented.

A-1 LQ Control with integral action with position error feedback

Figure A-1 shows the simulation results of the LQ controller with position error of quadrotor as feedback. Figure A-1a shows the plot of quadrotor and the slung load position. As explained in Section 3-8, the position error feedback controller results in large overshoot of 74%. However, it is able to stabilize the slung load and transport it in a swing free manner. The rise time and settling time are approximately 2 seconds and 8 seconds respectively. The disadvantage of such a control can be seen in Figure A-1d where the computed attitude references are unrealistic and cannot be applied to the system. These reference values must be bounded externally. Figure A-1b shows the plot of quadrotor attitude. Clearly the quadrotor achieves attitude values as high as 20 degrees. While this may work in simulations, it may not work in practice as the effects of model mismatch would be more prominent.

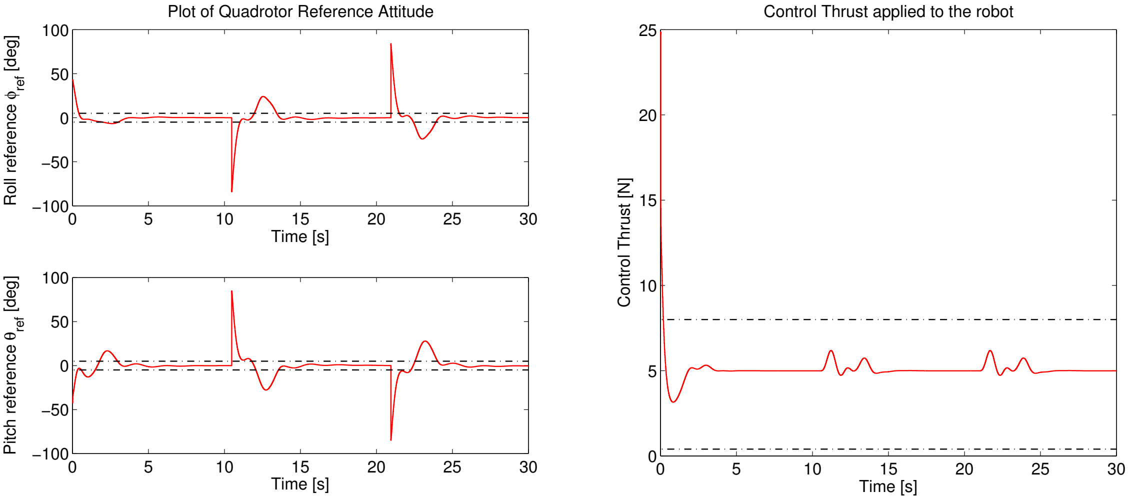


(a) Quadrotor and slung load position

(b) Attitude of the quadrotor

(c) Swing angles of the slung load

Figure A-1: Linear Quadratic control with integral action with position error $\mathbf{x}_Q - \mathbf{x}_{Q_{\text{ref}}}$ as feedback



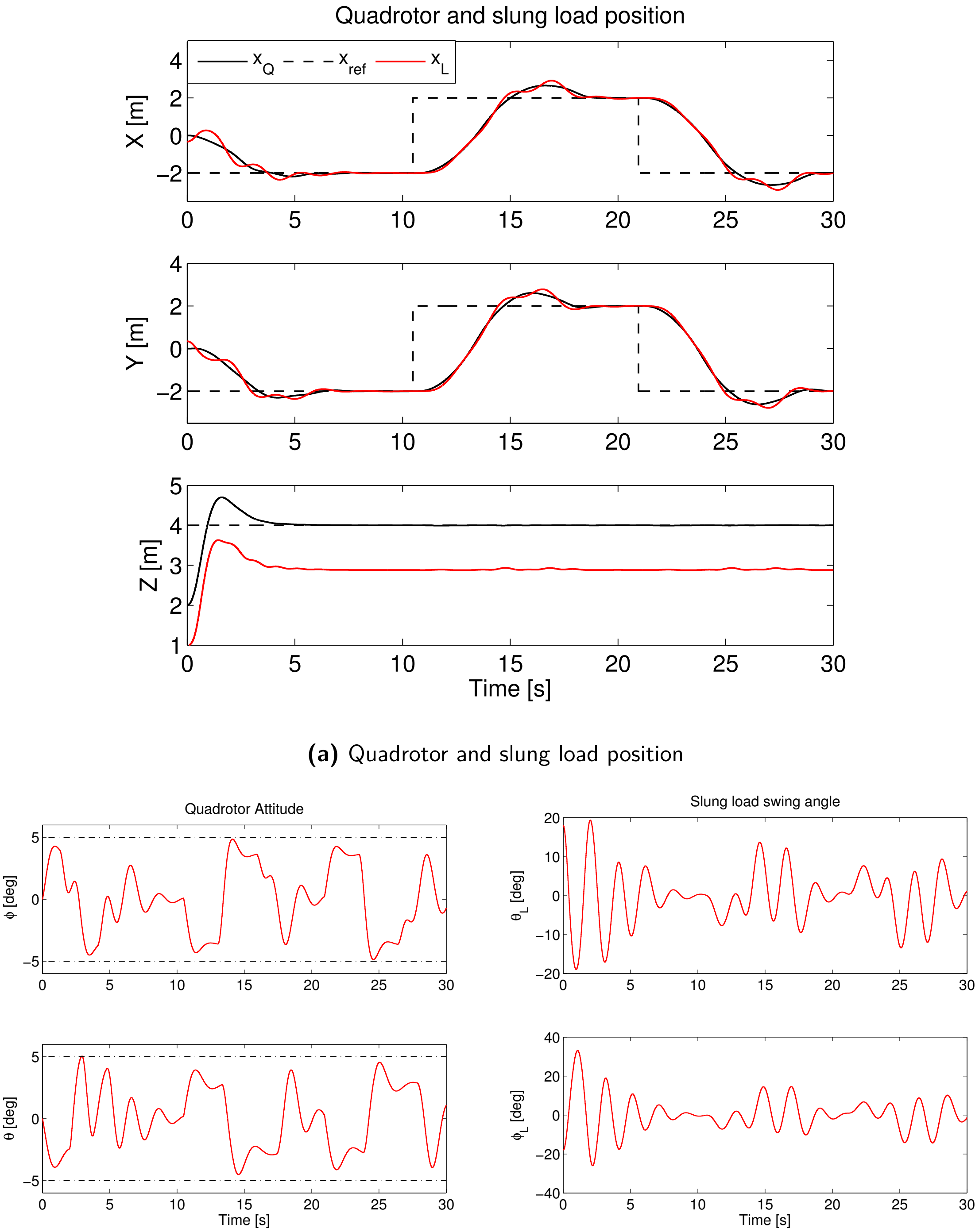
(d) Attitude reference

(e) Control thrust

Figure A-1: Linear Quadratic control with integral action with position error $\mathbf{x}_Q - \mathbf{x}_{Q_{\text{ref}}}$ as feedback

A-2 MPC with integral action with position error feedback

Figure A-2 shows the simulation result of MPC with integral action with position error of the quadrotor as feedback. From the Figure A-2a, it can be seen that the overshoot in the quadrotor position is reduced compared to the LQ control with integral action. It is also able to stabilize the slung load with oscillations during transients as can be seen in Figure A-2c. The advantage of this formulation over LQ control is constraint handling as can be seen in Figure A-2d and Figure A-2e. The rise time and settling time are approximately 4 seconds and 7 seconds respectively.

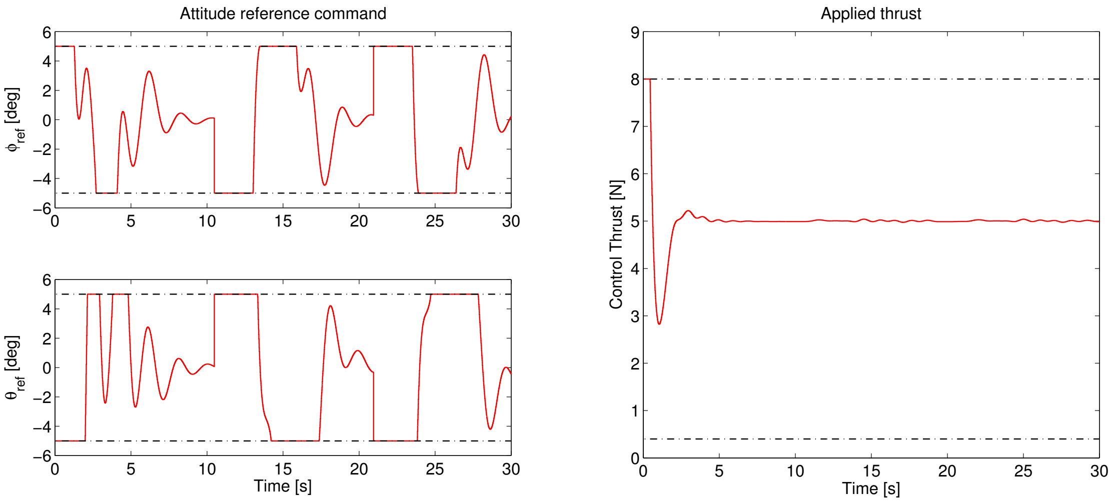


(a) Quadrotor and slung load position

(b) Attitude of the quadrotor

(c) Swing angles of the slung load

Figure A-2: Model Predictive Control with integral action with position error of the quadrotor $\mathbf{x}_Q - \mathbf{x}_{Q\text{ref}}$ as feedback



(d) Attitude reference

(e) Control thrust

Figure A-2: Model Predictive Control with integral action with position error of the quadrotor $\mathbf{x}_Q - \mathbf{x}_{Q_{\text{ref}}}$ as feedback

Appendix B

Thrust Measurements

The thrust measurements presented in this appendix was performed by a master student [41] at Micro Air Vehicle Laboratory. It is used to obtain a static map relating the applied thrust command f_{pprz} to the computed thrust command f [N]. It is reproduced here for future reference.

Table B-1: Thrust measurements

Measured Voltage [V]	Thrust Level	Weight [grams]
12.35	0	52
12.30	975	88.20
12.20	2175	154
12.10	3075	211
12	3975	280
11.90	5100	380
11.70	6225	500
11.60	7050	600
11.50	8175	745
11.20	9000	845
10.85	9525	820
11.80	0	53
11.80	975	88.30
11.70	2175	150
11.60	3075	206
11.50	4125	290
11.40	5025	370
11.30	6000	474
11.20	7050	592
11	8175	730
10.80	8850	800

Continued on next page

Table B-1 – Continued from previous page

Measured Voltage [V]	**Thrust Level**	**Weight [grams]**
11.30	0	53.40
11.20	975	88
11.10	2175	150
11	3075	210
10.90	3975	280
10.70	4950	365
10.55	6150	485
10.50	6975	590
10.45	8025	715
10.40	9150	760
10.35	9525	760
10.60	0	53
10.50	975	88
12.60	0	103
12.50	975	107
12.40	2175	162
12.30	3075	225
12.20	3975	293
12.10	4950	380
12	6000	490
11.75	6975	606
11.55	8025	730
11.50	9150	745
11.40	9525	755
11.15	9000	830
11.30	8025	720
11.40	6975	585
11.50	6000	475
11.60	5025	380
11.60	3900	270
11.70	2925	198
11.75	2025	140
11.80	975	89
11.85	0	52
10.40	9525	770
10.45	8775	750
10.70	8025	720

Bibliography

[1] M. Bernard, K. Kondak, I. Maza, and A. Ollero, "Autonomous Transportation and Deployment with Aerial Robots for Search and Rescue Missions," *Journal of Field Robotics*, vol. 28, no. 6, pp. 914–931, 2011.

[2] L. Merino, F. Caballero, J. Martínez-de Dios, I. Maza, and A. Ollero, "An Unmanned Aircraft System for Automatic Forest Fire Monitoring and Measurement," *Journal of Intelligent and Robotic Systems*, vol. 65, no. 1-4, pp. 533–548, 2012.

[3] V. Kumar and N. Michael, "Opportunities and Challenges with Autonomous Micro Aerial Vehicles," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1279–1291, 2012.

[4] M. Hehn and R. D'Andrea, "A Flying Inverted Pendulum," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 763–770, May 2011.

[5] P. E. I. Pounds, D. Bersak, and A. Dollar, "Grasping from the air: Hovering capture and load stability," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2491–2498, May 2011.

[6] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, "Design, Modeling, Estimation and Control for Aerial Grasping and Manipulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2668–2673, IEEE, 2011.

[7] F. Augugliaro, A. Mirjan, F. Gramazio, M. Kohler, and R. D'Andrea, "Building Tensile Structures with Flying machines," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3487–3492, Nov 2013.

[8] Q. Lindsey, D. Mellinger, and V. Kumar, "Construction with quadrotor teams," *Autonomous Robots*, vol. 33, no. 3, pp. 323–336, 2012.

[9] J. Willmann, F. Augugliaro, T. Cadalbert, R. D'Andrea, F. Gramazio, and M. Kohler, "Aerial robotic construction towards a new field of architectural research," *International journal of architectural computing*, vol. 10, no. 3, pp. 439–460, 2012.

[10] C. Korpela, T. Danko, and P. Oh, "Designing a system for mobile manipulation from an unmanned aerial vehicle," in *IEEE Conference on Technologies for Practical Robot Applications (TePRA)*, pp. 109–114, April 2011.

[11] A. Schoellig, H. Siegel, F. Augugliaro, and R. D'Andrea, "So you think you can dance? rhythmic flight performances with quadrocopters," in *Controls and Art* (A. LaViers and M. Egerstedt, eds.), pp. 73–105, Springer International Publishing, 2014.

[12] F. Augugliaro, A. Schoellig, and R. D'Andrea, "Dance of the flying machines: Methods for designing and executing an aerial dance choreography," *Robotics Automation Magazine, IEEE*, vol. 20, pp. 96–104, Dec 2013.

[13] M. Bernard and K. Kondak, "Generic slung load transportation system using small size helicopters," in *IEEE International Conference on Robotics and Automation*, pp. 3258–3264, May 2009.

[14] N. Michael, J. Fink, and V. Kumar, "Cooperative manipulation and transportation with aerial robots," *Autonomous Robots*, vol. 30, no. 1, pp. 73–86, 2011.

[15] R. Ritz and R. D'Andrea, "Carrying a flexible payload with multiple flying vehicles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3465–3471, Nov 2013.

[16] L. S. Cicolani and G. Kanning, *Equations of motion of slung-load systems, including multilift systems*, vol. 3280. National Aeronautics and Space Administration, Office of Management, Scientific and Technical Information Program, 1992.

[17] L. S. Cicolani, G. Kanning, and R. Synnestvedt, "Simulation of the dynamics of helicopter slung load systems," *Journal of the American Helicopter Society*, vol. 40, no. 4, pp. 44–61, 1995.

[18] K. Sreenath, N. Michael, and V. Kumar, "Trajectory Generation and Control of a Quadrotor with a Cable-suspended Load - a Differentially-flat Hybrid System," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4888–4895, May 2013.

[19] K. Sreenath, T. Lee, and V. Kumar, "Geometric Control and Differential Flatness of a Quadrotor UAV with a Cable-suspended Load," in *IEEE 52nd Annual Conference on Decision and Control (CDC)*, pp. 2269–2274, IEEE, 2013.

[20] F. A. Goodarzi, D. Lee, and T. Lee, "Geometric Stabilization of Quadrotor UAV with a Payload Connected by Flexible Cable," *arXiv preprint arXiv:1309.6717*, 2013.

[21] I. Palunko, P. Cruz, and R. Fierro, "Agile Load Transportation : Safe and Efficient Load Manipulation with Aerial Robots," *Robotics Automation Magazine, IEEE*, vol. 19, pp. 69–79, Sept 2012.

[22] I. Palunko, R. Fierro, and P. Cruz, "Trajectory Generation for Swing-free Maneuvers of a Quadrotor with Suspended Payload: A Dynamic Programming Approach," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2691–2697, May 2012.

[23] I. Palunko and R. Fierro, "Adaptive Control of a Quadrotor with Dynamic Changes in the Center of Gravity," in *Proceedings of the 18th IFAC World Congress*, pp. 2626 – 2631, August 2011.

[24] G. Starr, J. Wood, and R. Lumia, "Rapid Transport of Suspended Payloads," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 1394–1399, IEEE, 2005.

[25] D. Zameroski, G. Starr, J. Wood, and R. Lumia, "Rapid Swing-free Transport of Nonlinear Payloads using Dynamic Programming," *Journal of Dynamic Systems, Measurement, and Control*, vol. 130, no. 4, 2008.

[26] C. de Crousaz, F. Farshidian, and J. Buchli, "Aggressive optimal control for agile flight with a slung load,"

[27] E. Todorov and W. Li, "A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems," in *American Control Conference, 2005. Proceedings of the 2005*, pp. 300–306, IEEE, 2005.

[28] H. Omar, "New Fuzzy-based anti-swing Controller for Helicopter Slung-load system near Hover," in *IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, pp. 474–479, Dec 2009.

[29] A. Faust, I. Palunko, P. Cruz, R. Fierro, and L. Tapia, "Learning Swing-free Trajectories for UAVs with a Suspended Load," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4902–4909, May 2013.

[30] M. Bangura, R. Mahony, *et al.*, "Real-time Model Predictive Control for Quadrotors,"

[31] T. van den Boom and B. de Schutter, *Lecture Notes: Optimization in Systems and Control.* 2012.

[32] K. Kunz, S. M. Huck, and T. H. Summers, "Fast Model Predictive Control of Miniature Helicopters," in *European Control Conference (ECC)*, pp. 1377–1382, IEEE, 2013.

[33] R. Murray, "Lecture notes: Optimization based Control, Chapter Trajectory Generation and Tracking," 2010.

[34] J. Mattingley and S. Boyd, "CVXGEN: A code generator for Embedded Convex Optimization," *Optimization and Engineering*, vol. 13, no. 1, pp. 1–27, 2012.

[35] R. Praveen Jain, "Cooperative Load Manipulation using Unmanned Aerial Vehicles," October 2014. Literature survey report.

[36] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation.* CRC press, 1994.

[37] D. Greenwood, *Principles of Dynamics, the 2nd Edition.* Englewood Cliffs, NJ Prentice-Hall, 1988.

[38] L. R. García Carrillo, A. E. Dzul López, R. Lozano, and C. Pégard, "Modeling the quad-rotor mini-rotorcraft," in *Quad Rotorcraft Control*, Advances in Industrial Control, pp. 23–34, Springer London, 2013.

[39] R. Mahony, V. Kumar, and P. Corke, “Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor,” *Robotics Automation Magazine, IEEE*, vol. 19, pp. 20–32, Sept 2012.

[40] T. Lee, M. Leoky, and N. H. McClamroch, “Geometric Tracking Control of a Quadrotor UAV on SE(3),” in *49th IEEE Conference on Decision and Control (CDC)*, pp. 5420–5425, IEEE, 2010.

[41] C. Poppe, E. van Kampen, C. de Wagter, C. de Visser, and Q. Chu, “Aggressive Quadrotor Maneuvering by using Nonlinear Dynamic Inversion with Pseudo-Control Hedging and Flight Mode Transition.”.

[42] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, “Constrained Model Predictive Control: Stability and Optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[43] J. Rawlings and D. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill Pub., 2009.

[44] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. 2014.

[45] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2009.

[46] A. Domahidi and J. Jerez, “FORCES Professional.” embotech GmbH (http://embotech.com/FORCES-Pro), July 2014.

[47] A. Domahidi, A. Zgraggen, M. Zeilinger, M. Morari, and C. Jones, “Efficient Interior Point Methods for Multistage Problems arising in Receding Horizon Control,” in *IEEE 51st Annual Conference on Decision and Control (CDC)*, pp. 668–674, Dec 2012.

[48] R. Cagienard, P. Grieder, E. C. Kerrigan, and M. Morari, “Move Blocking Strategies in Receding Horizon Control,” *Journal of Process Control*, vol. 17, no. 6, pp. 563–570, 2007.

[49] N. Matthew and R. Stones, *Beginning Linux Programming*. Birmingham, UK, UK: Wrox Press Ltd., 4th ed., 2007.

[50] R. Love, *Linux System Programming: Talking Directly to the Kernel and C Library*. O’Reilly Media, Inc., 2007.

[51] B. de Schutter and M. Heemels, *Lecture Notes: Modeling and Control of Hybrid Systems*. 2014.

Glossary

List of Acronyms

TU Delft	Delft University of Technology
UAV	Unmanned Aerial Vehicles
MAV	Micro Aerial Vehicles
VTOL	Vertical takeoff and landing
DOF	Degrees of Freedom
OCP	Optimal Control Problem
MPC	Model Predictive Control
LTI	Linear Time Invariant
LTV	Linear Time Varying
PID-FF	Proportional-Integral-Derivative Control with Feed Forward term
SISO	Single Input Single Output
MIMO	Multiple Input Multiple Output
AHRS	Attitude Heading and Reference System
MPC-I	Model Predictive Control with integral action
LQ	Linear Quadratic
QP	Quadratic Programming
UDP	User Datagram Protocol