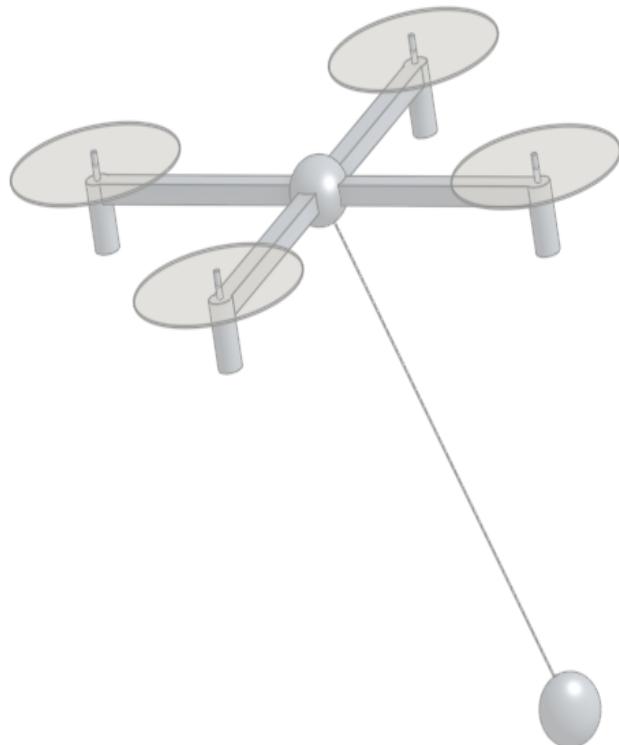


Nonlinear Geometric Control of a Quadrotor with a Cable-Suspended load

N.N. Vo



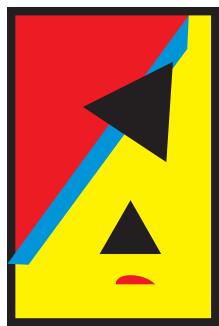
Nonlinear Geometric Control of a Quadrotor with a Cable-Suspended load

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Mechanical Engineering at Delft
University of Technology

N.N. Vo

August 15, 2017



AL TEN

The work in this Master of Science Thesis was supported by Alten Nederland.
Their cooperation is hereby gratefully acknowledged.



Copyright © Delft Center for Systems and control
All rights reserved.



DELFT UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF
DELFT CENTER FOR SYSTEMS AND CONTROL

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering for acceptance a thesis entitled

NONLINEAR GEOMETRIC CONTROL
OF A QUADROTOR WITH
A CABLE-SUSPENDED LOAD

by

N.N. Vo

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE MECHANICAL ENGINEERING

Dated: August 15, 2017

Supervisor(s):

dr.ir. T. Keviczky

ir. B. van Vliet

Reader(s):

Dr. J. Alonso-Mora

A. Sharifi Kolarijani

Abstract

A quadrotor is a type of Unmanned Aerial Vehicle that has received an increasing amount of attention recently with many applications being actively investigated. Possible applications include search and rescue, surveillance, supply of food and medicines in emergency situations and object manipulation in construction and transportation. An interesting subproblem of load transportation is the control of the position of a cable suspended load. The challenge is in the fact that the quadrotor-load system is highly nonlinear and under-actuated. The load cannot be controlled directly and has a natural swing at the end of each quadrotor movement.

The goal of this thesis is to present a nonlinear geometric control approach, investigate its possibilities and limitations to track the position of a cable suspended load. The focus lies on the quadrotor-load subsystem where the cable tension is non-zero, which is analogous to modeling a rigid link between the quadrotor and load.

After introducing the basic concepts, an introduction is given on geometric mechanics. This differential geometric based approach is used model and control the system, based on the geometric properties of the system dynamics. It is shown how the configuration of the quadrotor-load system can be described on smooth nonlinear geometric configuration spaces, and analyzed with the principles of differential geometry. This allows for modeling in an unambiguous coordinate-free dynamic fashion, while avoiding the problem of singularities that would occur on local charts.

Next, the geometric properties are utilized to define tracking error functions on these same spaces, making it possible to design almost-globally defined nonlinear geometric controllers. A backstepping approach is applied to generate a cascaded structure with multiple nonlinear geometric controllers, allowing control of several flight modes that are responsible for the control of 1) quadrotor attitude, 2) load attitude and 3) load position.

Finally, simulations demonstrate the stability and abilities of the nonlinear geometric controller. A Linear Quadratic Regulator is derived to compare control performance. The tracking performances of both controllers are discussed for three different experiments and final conclusions are presented. The thesis is concluded with a summary and recommendations for future work.

Acknowledgements

I would like to thank my supervisors dr.ir. T. Keviczky from *Delft Center of Systems and control*, and ir. B. van Vliet from *Alten Nederland B.V.* for their assistance during my research and the writing of this thesis. I would also like to thank all colleagues from Alten and TU Delft for their time and advice.

Delft, University of Technology
August 15, 2017

N.N. Vo

Table of Contents

Abstract	i
Acknowledgements	iii
1 Introduction	1
1-1 Aim and Motivation	2
1-2 Organization of the Report	3
2 Dynamic Model	5
2-1 Geometric Mechanics	6
2-2 Quadrotor Model	10
2-3 Quadrotor-Load Model	12
3 Control Design	17
3-1 Nonlinear Geometric Control	18
3-1-1 Error Functions	18
3-2 Backstepping Control	21
3-2-1 Quadrotor Attitude Tracking	22
3-2-2 Load Attitude Tracking	24
3-2-3 Load Position Tracking	25
4 Experiment	27
4-1 Procedure	28
4-2 Trajectories	29
4-3 Setup	32
4-4 Results	35
4-5 Summary	48

5 Conclusions and Future Work	49
5-1 Summary and Conclusions	49
5-2 Recommendations for Future Work	50
5-2-1 Investigate Implementation	50
5-2-2 Trajectory Generation	51
A Appendix	53
A-1 LQR controller	53
A-2 Additional Figures	55
Acronyms	65

List of Figures

2-1 Configuration Space of a 2-link arm	6
2-2 A manifold locally resembles a Euclidean space	7
2-3 Tangent Spaces on different manifolds	8
2-4 Quadrotor model representation	11
2-5 Quadrotor with load model representation	12
3-1 Transport map $\mathcal{T}(q, r)$	19
3-2 Nonlinear geometric control loop of the QR-load system [1]	21
3-3 QR Attitude controller	22
3-4 Construction of b_{1c} to define R_c	24
4-1 Desired load trajectory Case A, y-direction	29
4-2 Desired load trajectory Case B, y-direction	30
4-3 Desired load Position Case C	31
4-4 Desired load Position Case C	31
4-5 Command Filter	33
4-6 LQR control design	34
4-7 Load Position Tracking NGC Case A	36
4-8 Geometric Error functions NGC Case A	37
4-9 Geometric Tracking Error functions NGC Case A	37
4-10 Controller Comparison Case A. Solid: NGC, Dash-dot: LQR	38
4-11 Controller Comparison Case A. Solid: NGC, Dash-dot: LQR	38
4-12 Load Position Tracking nonlinear geometric control Case B	39
4-13 Geometric Error functions NGC Case B	40
4-14 Geometric Tracking Error functions NGC Case B	40
4-15 Controller Comparison Case B. Solid: NGC, Dash-dot: LQR	41

4-16 Controller Comparison Case B. Solid: NGC, Dash-dot: LQR	41
4-17 Desired load trajectory case B extended, y-direction	42
4-18 Controller Comparison Case B extended. Solid: NGC, Dash-dot: LQR	43
4-19 Load Position Tracking nonlinear geometric control Case C	45
4-20 Geometric Error functions NGC Case C	45
4-21 Geometric Tracking Error functions NGC Case C	46
4-22 Controller Comparison Case C. Solid: NGC, Dash-dot: LQR	46
4-23 Controller Comparison Case C. Solid: NGC, Dash-dot: LQR	47
A-1 Control InputsNGC Case A	55
A-2 Control InputsNGC Case B	55
A-3 Control InputsNGC Case B extended	56
A-4 Control InputsNGC Case C	56
A-5 QR Position Error and Load Angle Error LQR Case A. Dash-dot: LQR	57
A-6 QR Position Error and Load Angle Error LQR Case B. Dash-dot: LQR	57
A-7 QR Position Error and Load Angle Error LQR Case B extended. Dash-dot: LQR	58
A-8 QR Position Error and Load Angle Error LQR Case C. Dash-dot: LQR	58

List of Tables

2-1	Modeling assumptions quadrotor model	10
2-2	Modeling assumptions quadrotor-load model	12
4-1	Modeling Parameters	32
4-2	Hardware Parameters for Bebop Parrot Drone 1	32
4-3	Command Filter Parameters	33
4-4	Controller gains nonlinear geometric controller	35

Chapter 1

Introduction

A Quadrotor (QR) is a type of Unmanned Aerial Vehicle (UAV) that has received an increasing amount of attention recently with many applications being actively investigated. Possible applications include search and rescue, surveillance, reliable supply of food and medicines in emergency situations and object manipulation in construction and transportation. It has already proven itself useful for many tasks like multi-agent missions, mapping, explorations, transportation and entertainment such as acrobatic performances.

The inspiration for this research is build upon the idea of creating a system of multiple autonomous QRs for a cooperative towing task. The advantage of such systems for object manipulation is the increased reach and the possibility to reduce complexity of the individual robot, decrease cost over traditional robotic systems and high reliability. One can think of examples in nature, where individuals coordinate, cooperate and collaborate to perform tasks that they individually can not accomplish. Redundancy makes development of fail safe control methods possible and can extend the capabilities of a single robot.

Considering a multi-agent task, one can think of multiple QRs assisting in the transportation of a common load in many ways. One interesting method is the transportation of a load suspended via a cable. Prior to a multi-agent load transportation task, research must be done to this task involving a single QR. This research revolves around load position control of a single QR with a cable-suspended load in motion. The suspended object naturally continues to swing at the end of every movement. In case a residual motion can result in damage or in order to avoid obstacles and path following, an accurate positioning is required. Reducing the oscillation or controlling the position of the suspended load might be necessary, but is challenging in the fact that this cable-suspended system is under-actuated. Possible objectives are minimizing the oscillations of the load during or after motion, minimizing the time to position the load, trajectory tracking, trajectory generation and obstacle avoidance.

1-1 Aim and Motivation

The aim is to control the position of a suspended load using a QR. Before considering multiple QRs, it is important to investigate the possibilities of a single QR with load system. Hence, in this research a single QR is considered for the transportation of a cable suspended load, which will exert additional forces and torques on the QR. This is a challenging control problem in the fact that the QR system is under-actuated. Adding a suspended load will add extra Degree of Freedom (DOF)s and oscillations of the load occur at the end of every movement.

The system can be divided into two subsystems. The first subsystem is where the cable tension is non-zero and the distance between the QR and the load is defined by the cable length. Both QR and load are coupled as one system. The second subsystem is where the cable tension is zero, such that the QR and load in free fall are two separate decoupled systems. This research focuses on the first subsystem, such that the cable tension is non-zero. In order to control both subsystems, hybrid control must be applied, which is considered out of the scope of this research.

Former work on attitude control of QR and/or load often relies on linear control methods such as PID [2, 3], MPC [4] and LQR [5] control. The dynamics are linearized around an equilibrium point, describing the system dynamics by a set of linear differential equations. The control of a QR-load system is a very specific case and scarcely investigated. Former work includes MPC [6, 7] and LQR control approaches [8], where optimal control strategies are used to minimize the swing of the load.

The reason that linear control near an equilibrium state is commonly applied, is partly to avoid difficulties that come with modeling and controlling the non-linearities of the system. However, linear control limits the system to small angle movements, as the optimization will not allow large angles that deviate too much from the linearized point. This approach of modeling and control will not be sufficient for applications that require fast aggressive maneuvers. Nonlinear control systems are often governed by nonlinear differential equations and are able to represent the dynamics in a more realistic manner. Nonlinear control approaches to minimize the load swing includes a Model Based Algorithm controller applied by [9].

Nonlinear Geometric Control is a nonlinear model based control technique based on a modeling approach involving the concepts of differential geometry. This results in a globally defined coordinate-free dynamical model, while preventing issues regarding singularities, and enabling the design of controllers that offer almost-global convergence properties. For a study on rigid body dynamics and optimal control problems, where geometric features are incorporated, one can refer to [10, 11].

Former work includes a nonlinear geometric control of a QR [12, 13] and nonlinear geometric control of the load position, load attitude and QR attitude of a QR-load system [14, 15, 16]. Nonlinear Geometric Control for QR systems is rarely found in literature, despite the advantageous properties of differential geometry.

This motivates to investigate the potential and limitations of a rarely used nonlinear Geometric Control approach. The control performance of a nonlinear geometric controller for the task of a load transportation maneuver can be investigated, and a comparison can be made with linear control strategies.

Different aspects involving the modeling and control for the QR-load system must be investigated, for it can be expected that the non-linearity will have a great influence in the

representation of the dynamics and the stability, accuracy and type of the control design. It is possible to investigate which advantages or disadvantages this nonlinear approach has compared to a linear approach, in terms of stability and performance.

1-2 Organization of the Report

In this first chapter, a brief introduction of the subject is given and the problem is described. This is followed by discussing the aim, motivation and contributions of this thesis for this research. The organization of the report is as follows.

Chapter 2 introduces Geometric Mechanics to understand and derive the system's equations of motion in order to allow nonlinear geometric controller design and analysis. The system configuration space is described on a differentiable manifold using the tools of differential geometry instead of Euclidean geometry, where the system dynamics evolve in a three dimensional space. The dynamics of the QR-load system are then described by the laws of kinematics and the application of Newton's laws and Lagrangian mechanics. In contrast with classical modeling techniques, geometric modeling results in a compact, unambiguous and coordinate-free model.

Describing the system dynamics on nonlinear manifolds allows the design of nonlinear geometric controllers on these same manifolds. The control design is presented in Chapter 3. The controller has a cascaded structure, allowing the control of several flight modes that are accountable for the control of different degrees of freedom.

Chapter 4 describes the experiments that are done to investigate the abilities and performance of a nonlinear Geometric Control design. Different tracking objectives are defined in order to compare the performance between an LQR control design and a nonlinear Geometric Control design. The results are presented and findings are discussed.

In the final chapter a summary of the thesis is given, followed by the conclusions that were made based on the results of the research. Finally, recommendations are given which could serve as an starting point for future work.

Chapter 2

Dynamic Model

A mathematical model of the system needs to be derived in order to simulate and study the effects of nonlinear geometric control. In Section 2-1, an introduction is given on geometric mechanics, a modern description of classical mechanics from the perspective of differential Geometry, a discipline in mathematics that studies manifolds and their geometric properties, using the tools of calculus.

Section 2-2 starts by discussing assumptions that are applied to simplify the model. Next, a dynamical model of the QR is presented. In Section 2-3, the same is done to obtain a dynamical model of the QR-load system with geometric mechanics, resulting in a compact, coordinate-free, unambiguous representation of the dynamics, described on nonlinear manifolds.

2-1 Geometric Mechanics

For the derivation of the equations of motions, traditional modeling methods often parameterize the rotations in a local coordinate system. This can be done with Euler Angles, and despite this parametrization might result in singularities, this is a commonly used method to describe rotations. There are 24 possible sets of Euler angles and many different conventions are used, which introduces ambiguity. The definition of Euler angles is not unique and a sequence of rotations is not commutative. Therefore, Euler angles are never expressed in terms of the external frame, or in terms of the co-moving rotated body frame, but in a mixture.

An other disadvantage of Euler angles, is that the transformation from their time rates of change to the angular velocity vector is not globally defined. Furthermore, when angular errors are large, the difference in Euler angles is no longer a good metric to define the orientation error. Hence, the error is rather written as the required rotation to get from the current to a desired orientation, which can be achieved by considering geometric properties of the system.

In geometric mechanics the configuration space of systems is a *group manifold* instead of a Euclidean space. The kinetic and potential energies are expressed in terms of this configuration space and their tangent spaces. It explores the geometric structure of a Lagrangian- or Hamiltonian system through the concepts of vector calculus, linear algebra, differential geometry, and nonlinear control theory. geometric mechanics provides fundamental insights into the nonlinear system mechanics and yields useful tools for dynamics and control theory.

An example is given of a simple 2-link arm, to illustrate different representations of a configuration space, see Figure 2-1. Let the configuration of the arm be defined by two coordinates in a Cartesian coordinate system, which is a local representation. This can be seen in Figure 2-1b, where the colored edges illustrate singularities, because the definition of one point has multiple solutions.

Next, the configuration space is represented as a geometric shape called a *torus*, as shown in Figure 2-1c. It is a smooth manifold where each configuration is mapped uniquely, which allows the configuration to be defined globally.

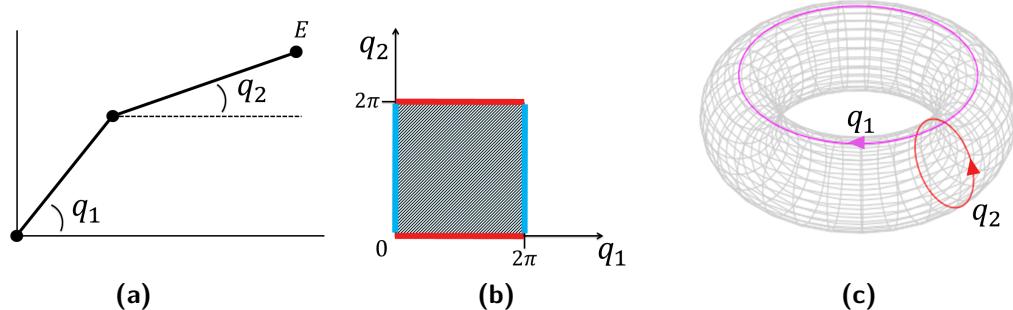


Figure 2-1: Configuration Space of a 2-link arm

Manifolds The fundamental object of differential geometry a manifold. A manifold is a mathematical space, a collection of points, that locally resembles Euclidean space near each

point. Examples are a plane, a ball, a torus and a sphere. Manifolds are important objects in mathematics and physics, because they allow more complicated structures to be expressed and understood in terms of the relatively well-understood properties of simpler spaces. Each point of an n-dimensional manifold has a neighborhood that is homeomorphic to the n-dimensional Euclidean space, meaning that there is a continuous function describing the relation between these spaces, illustrated in Figure 2-2. This allows coordinate-free modeling of a system, avoiding singularities that come with local representations.

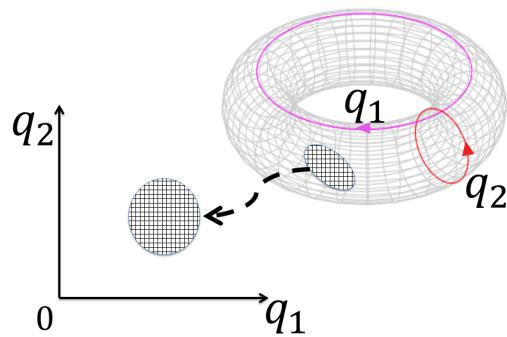


Figure 2-2: A manifold locally resembles a Euclidean space

A *differentiable manifold* is a smooth and continuous manifold and is locally similar enough to a linear space to allow to do calculus. One can define directions, tangent spaces, and differentiable functions on such a manifold [17].

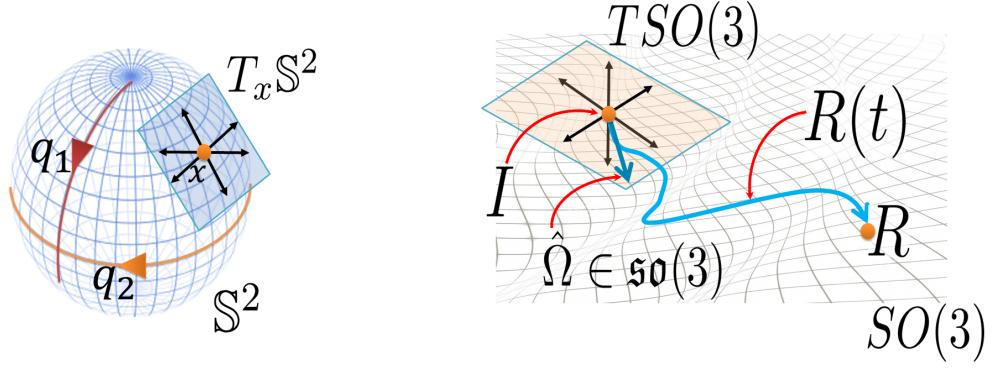
Taking the derivative at a point on a manifold is equivalent to a *tangent vector* at that point. Meaning that derivatives are conceptually equivalent to an infinitesimally short tangent vector. Each point of an n-dimensional differentiable manifold has a tangent space, which is an n-dimensional Euclidean space consisting of all the tangent vectors of *all* curves that pass through that point.

A tangent space describes a relationship between a position and a velocity at that position. This property is of importance for the determination of configuration error functions, which give a measure of the error between a desired state and an actual state. The configuration errors will be described in Section 3-1.

To illustrate a tangent space, a point x is chosen on a *2-sphere*, which is a manifold denoted by \mathbb{S}^2 and defined by a sphere of dimension 2. The tangent space at point x is the collection of all tangent vectors at point x and is denoted by $T_x\mathbb{S}^2$, see Figure 2-3a.

Geometric Configuration Spaces Several methods exist to describe rotations, such as *Euler Angles*, quaternions or rotation matrices. The main disadvantages of Euler angles are that some functions have singularities and they are a less accurate measure for the integration of incremental changes in attitude over time, compared to other methods. To avoid these problems, in geometric mechanics rotations are expressed as rotation matrices to provide a global representation of the attitude of a rigid body.

The QR attitude is expressed as a rotation matrix R in the Special Orthogonal Group $SO(3)$, which describes the rotation of a body frame relative to the spatial frame. The manifold



(a) Representation of a manifold with a tangent space

(b) Identity map of manifold $SO(3)$ with Lie Algebra $\mathfrak{so}(3)$

Figure 2-3: Tangent Spaces on different manifolds

$SO(3)$ is defined as

$$SO(3) \triangleq \left\{ R \in \mathbb{R}^{3 \times 3} \mid RR^T = I_{3 \times 3}, \det(R) = 1 \right\} \quad (2-1)$$

where $SO(3)$ is the group of all rotations about the origin of a 3-D Euclidean space, which preserves the origin, Euclidean distance and orientation. [18, 19]

Every rotation has a unique inverse rotation and the identity map satisfies the definition of a rotation. The elements of *Lie Algebra* $\mathfrak{so}(3)$, a property associated with $SO(3)$, are the elements of the *tangent space* of $SO(3)$ at the identity element, see Figure 2-3b. These elements define the relation between the rotation R and its derivative \dot{R} , such that

$$\dot{R} = R\hat{\Omega} \quad (2-2)$$

For $n \in \mathbb{N}$, $\mathfrak{so}(n)$ is the vector space of skew-symmetric matrices in $\mathbb{R}^{n \times n}$ and defined as

$$\mathfrak{so}(n) \triangleq \left\{ S \in \mathbb{R}^{n \times n} \mid S^T = -S \right\} \quad (2-3)$$

The hat map $\wedge : \mathbb{R}^3 \rightarrow \mathfrak{so}(3)$ is an isomorphism between \mathbb{R}^3 and the set of 3×3 skew symmetric matrices, such that $\hat{x}y = x \times y$ for any $x, y \in \mathbb{R}^3$. The vee map $\vee : \mathfrak{so}(3) \rightarrow \mathbb{R}^3$, and is the inverse isomorphism of the hat map. Several properties of the hat map are

$$\hat{x}y = x \times y = -y \times x = -\hat{y}x, \quad (2-4)$$

$$tr[A\hat{x}] = \frac{1}{2}tr[\hat{x}(A - A^T)] = -x^T(A - A^T)\vee, \quad (2-5)$$

$$\hat{x}A + A^T\hat{x} = (\{tr[A]I_{3 \times 3} - A\}x)^\wedge, \quad (2-6)$$

$$R\hat{x}R^T = (Rx)^\wedge, \quad (2-7)$$

for any $x, y \in \mathbb{R}^3$, $A \in \mathbb{R}^{3 \times 3}$, and $R \in SO(3)$. The mapping between the body angular velocity vector $\Omega \in \mathbb{R}^3$ and $\hat{\Omega} \in \mathfrak{so}(3)$ is written as

$$\hat{\Omega} = \begin{bmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{bmatrix}^\vee = \Omega \quad (2-8)$$

The load attitude is expressed as a unit vector q , which points from $\{\mathcal{B}\}$ to the load. The configuration space is a *two-sphere* \mathbb{S}^2 defined as

$$\mathbb{S}^2 \triangleq \left\{ q \in \mathbb{R}^3 \mid q \cdot q = 1 \right\} \quad (2-9)$$

The plane tangent to the sphere at q is the tangent space

$$T_q \mathbb{S}^2 \simeq \left\{ \omega \in \mathbb{R}^3 \mid q \cdot \omega = 0 \right\} \quad (2-10)$$

where ω is the angular velocity of the suspended load.

2-2 Quadrotor Model

Assumptions can be made to simplify the complexity of the mathematical model. Table 2-1 shows the assumptions that are used for modeling the QR system.

Modeling assumptions quadrotor model

- The structure of the QR is rigid and symmetric.
Elastic deformations and shock (sudden accelerations) of the QR are ignored.
- The mass distribution of the QR is symmetrical in the x-y plane.
- The inertia matrix is time-invariant.
- Aerodynamic effects acting on the QR are neglected.
Blade flapping, Turbulence, Ground Effects.
- The air density ρ around the QR is constant.
- The propellers are rigid \Rightarrow The thrust produced by rotor i is parallel to the axis of rotor i .
- Drag factor d and thrust factor b are approximated by a constant.
Thrust force F_i and moment M_i of each propeller is proportional to the square of the propeller speed.

Table 2-1: Modeling assumptions quadrotor model

The QR model representation is shown in Figure 2-4. Two Cartesian coordinate frames are defined:

- The body-fixed reference frame $\{\mathcal{B}\}$ (Body Frame)
 - with unit vectors $\{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$ along the axes
- The ground-fixed reference frame $\{\mathcal{I}\}$ (Inertial Frame)
 - with unit vectors $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ along the axes

such that $\{\mathcal{I}\}$ is fixed to earth and the body-frame axes \mathbf{b}_1 and \mathbf{b}_2 , coincide with the arms of the QR.

The QR is described as a rigid body with six degrees of freedom, driven by the system inputs: the total upward force f and the moments $M = [M_\phi \ M_\theta \ M_\psi]^T$ around the body axes.

The configuration of the QR can be described by 1) the location of the QR's Center of Mass (CM), $x_Q \in \mathbb{R}^3$, described in the Euclidean space w.r.t. $\{\mathcal{I}\}$, and 2) the *attitude* which is the orientation of $\{\mathcal{B}\}$ w.r.t. $\{\mathcal{I}\}$ evolving on a nonlinear space, described by a rotation matrix $R \in SO(3)$.

Rotations w.r.t. $\{\mathcal{B}\}$, about the axes b_1, b_2 and b_3 are parameterized by ϕ, θ and ψ , respectively, which is illustrated in Figure 2-4b.

The dynamics of a rigid body can be expressed on the manifold $SE(3)$, which is the group of *rigid displacements* in \mathbb{R}^3 . A rigid displacement describes both the rotation and the position of $\{\mathcal{B}\}$ relative to $\{\mathcal{I}\}$.

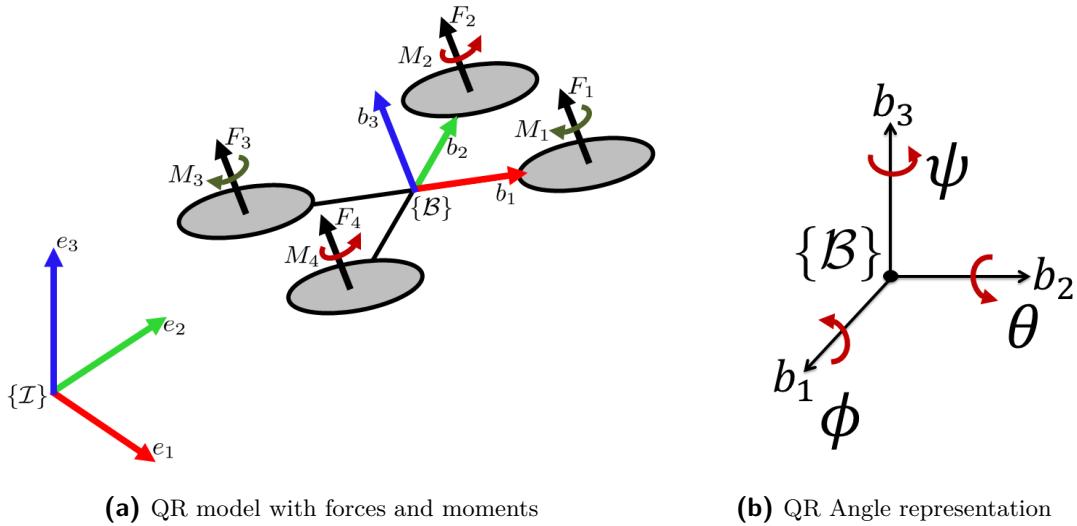


Figure 2-4: Quadrotor model representation

Rotor dynamics The complex dynamics of the rotors and their interactions with drag and thrust forces are represented by a simplified model. The angular speed ω_i of rotor i , for $i = 1, 2, 3, 4$, generates a force F_i parallel to the direction of the rotor axis of rotor i , given by

$$F_i = \left(\frac{K_v K_\tau \sqrt{2\rho A}}{K_t} \omega_i \right)^2 \simeq b \omega_i^2 \quad (2-11)$$

where K_v, K_t are constants related to the motor properties, ρ is the density of the surrounding air, A is the area swept out by the rotor, K_τ is a constant determined by the blade configuration and parameters, and b is the thrust factor.

The torque around the axis of rotor i , generated due to drag is given by

$$M_i = \frac{1}{2} R \rho C_D A (\omega_i R)^2 \simeq d \omega_i^2 \quad (2-12)$$

where R is the radius of the propeller, C_D is a dimensionless constant, and d is the drag constant.

The required rotor speeds ω_i can be calculated for a given desired total thrust f and total moment $M = [M_\phi \ M_\theta \ M_\psi]^T$, by solving the following equation

$$\begin{bmatrix} f \\ M_\phi \\ M_\theta \\ M_\psi \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -lb & 0 & lb \\ lb & 0 & -lb & 0 \\ -d & d & -d & d \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (2-13)$$

where l is the distance from the rotor to the QR's CM, d is the drag factor and b is the thrust factor.

2-3 Quadrotor-Load Model

Assumptions are also made to simplify the mathematical model for the QR-load system. Table 2-2 shows the assumptions that are used for modeling the QR-load system.

Modeling assumptions quadrotor-load model

- The cable is modeled as a rigid and massless cable.
- The cable is connected to a friction-less joint at the origin of the body-fixed.
- The tension in the cable is considered to be non-zero.
This implies that the QR-load subsystem that consists of a QR and a load in free fall, is disregarded.
- Aerodynamic effects acting on the load are neglected.

Table 2-2: Modeling assumptions quadrotor-load model

The total quadrotor-load model consists of two subsystems, 1) where the cable tension is zero, and 2) where the cable tension is non-zero. In this research, the focus is only on the subsystem where the cable tension is non-zero.

The QR-load model is shown in Figure 2-5. The unit vector $q \in \mathbb{S}^2$ gives the direction from the QR to the load expressed in $\{\mathcal{B}\}$. The position of the QR and load are related by

$$x_Q = x_L - Lq \quad (2-14)$$

where $x_Q \in \mathbb{R}^3$ is the position of the QR's CM expressed in $\{\mathcal{I}\}$, $x_L \in \mathbb{R}^3$ is the position of the load expressed in $\{\mathcal{I}\}$, and L is the length of the cable.

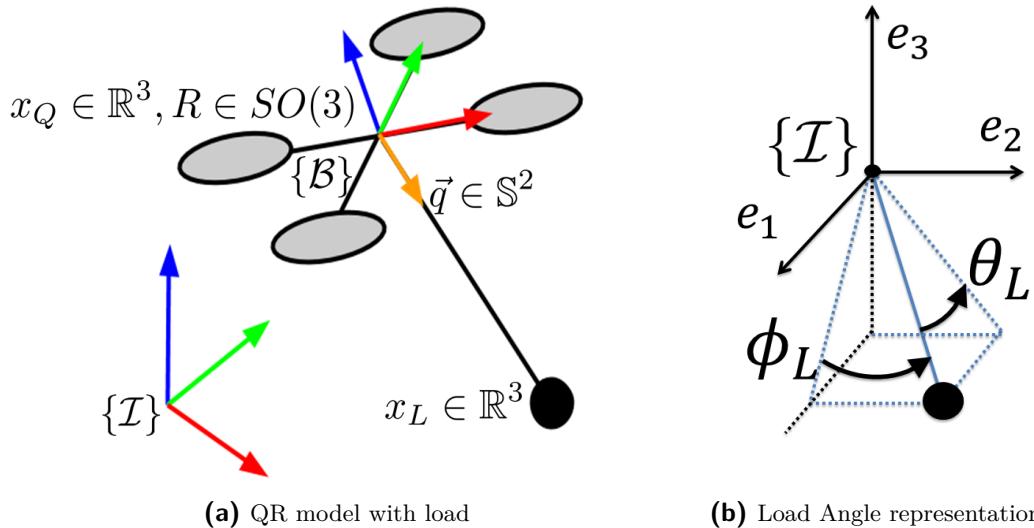


Figure 2-5: Quadrotor with load model representation

The configuration of the load can be described by its location $x_L \in \mathbb{R}^3$ w.r.t. $\{\mathcal{I}\}$, evolving in Euclidean space, and the load attitude evolving on a nonlinear space \mathbb{S}^2 , described by the

unit vector $q \in \mathbb{S}^2$. Rotations of the load w.r.t. $\{\mathcal{I}\}$, about the axes e_1 and e_2 are denoted by ϕ_L and θ_L , respectively, see Figure 2-5b.

Euler-Lagrange To develop the Euler-Lagrange equations for mechanical systems that evolve on manifolds, an approach developed by [10, 20, 21, 22] is applied. The basic idea is to express the variations of the curves evolving on \mathbb{S}^2 and $SO(3)$. This approach is based on Hamilton's principle, which states that the evolution of a physical system is a solution of the functional equation given by

$$\frac{\delta S}{\delta \mathbf{x}(t)} = 0 \quad (2-15)$$

where \mathbf{x} defines the configuration space. S is the action integral, defined as

$$S = \int_{t_1}^{t_2} \mathcal{L} dt \quad (2-16)$$

where $\mathcal{L} = \mathcal{T} - \mathcal{U}$ is the Lagrangian of the system, and \mathcal{T}, \mathcal{U} are the kinetic and potential energy, respectively.

Hamilton's principle of least action states that the path a conservative mechanical system takes between two states \mathbf{x}_1 and \mathbf{x}_2 at time t_1 and t_2 , is the one for which Equation 2-16 is a stationary point, resulting in

$$\delta S = \int_{t_1}^{t_2} \delta \mathcal{L} dt = 0 \quad (2-17)$$

where $\delta \mathcal{L}$ is the variation of the Lagrangian. For systems with non-conservative forces and moments, Equation 2-17 is extended to

$$\delta S = \int_{t_1}^{t_2} (\delta W + \delta \mathcal{L}) dt = 0 \quad (2-18)$$

where δW is the virtual work. Equation 2-18 is applied to the QR-load system, where the configuration manifold is $\mathbb{R}^3 \times \mathbb{S}^2 \times SO(3)$. With the following states

$$\mathbf{x} = [x_L \ \dot{x}_L \ q \ \omega \ R \ \Omega]^T \quad (2-19)$$

where ω is the angular velocity of the load and Ω denotes the angular velocity of the body-fixed frame.

The kinetic energy \mathcal{T} and the potential energy \mathcal{U} for the system are denoted as

$$\begin{aligned} \mathcal{T} &= \frac{1}{2} m_Q \dot{x}_Q \cdot \dot{x}_Q + \frac{1}{2} m_L \dot{x}_L \cdot \dot{x}_L + \frac{1}{2} \Omega \cdot J \cdot \Omega \\ \mathcal{U} &= m_Q g x_Q \cdot e_3 + m_L g x_L \cdot e_3 \end{aligned} \quad (2-20)$$

where $J \in \mathbb{R}^{3 \times 3}$ is the inertia tensor of the QR, g is the gravity constant, m_L is the mass of the load and m_Q is the mass of the QR.

The energy can be rewritten in terms of q and x_L , by substituting Equation 2-14, giving

$$\mathcal{T} = \frac{1}{2} (m_Q + m_L) \dot{x}_L \cdot \dot{x}_L - m_Q L \dot{x}_L \cdot \dot{q} + \frac{1}{2} m_Q L^2 \dot{q} \cdot \dot{q} + \frac{1}{2} \Omega \cdot J \cdot \Omega \quad (2-21)$$

$$\mathcal{U} = (m_Q + m_L) g x_L \cdot e_3 - m_Q g L q \cdot e_3 \quad (2-22)$$

Variations The variations of \mathcal{T} and \mathcal{U} are approximated by a first-order Taylor approximation, which results in

$$\begin{aligned}\delta\mathcal{T} &\approx \frac{\partial\mathcal{T}}{\partial\dot{x}_L}\delta\dot{x}_L + \frac{\partial\mathcal{T}}{\partial\dot{q}}\delta\dot{q} + \frac{\partial\mathcal{T}}{\partial\Omega}\delta\Omega \\ &= ((m_Q + m_L)\dot{x}_L - m_Q L\dot{q}) \cdot \delta\dot{x}_L + (-m_Q L\dot{x}_L + m_Q L^2\dot{q}) \cdot \delta\dot{q} + (J\Omega) \cdot \delta\Omega \\ \delta\mathcal{U} &\approx \frac{\partial\mathcal{U}}{\partial x_L}\delta x_L + \frac{\partial\mathcal{U}}{\partial q}\delta q \\ &= ((m_Q + m_L)g e_3) \cdot \delta x_L - (m_Q g L e_3) \cdot \delta q\end{aligned}\quad (2-23)$$

The first term of virtual work is obtained from f acting on the QR and is given by the following term,

$$\begin{aligned}\delta W_1 &= f R e_3 \cdot \sum_{j=1}^3 \frac{\partial x_Q}{\partial \mathbf{q}_j} \delta \mathbf{q}_j \\ &= f R e_3 \cdot (\delta x_L - L \delta q)\end{aligned}\quad (2-24)$$

where $\mathbf{q}_j = x_L, q, R$ and x_Q is substituted by Equation 2-14. The second term of virtual work is obtained from M acting on the QR. This gives the following term

$$\begin{aligned}\delta W_2 &= M \cdot \sum_{j=1}^3 \frac{\partial \Omega}{\partial \dot{\mathbf{q}}_j} \delta \dot{\mathbf{q}}_j \\ &= M \cdot (R^T \delta R)\end{aligned}\quad (2-25)$$

The variations in energy and the virtual work can be substituted into Equation 2-18, such that

$$\delta S = \int_{t_1}^{t_2} (\delta W_1 + \delta W_2 + \delta\mathcal{T} - \delta\mathcal{U}) dt \quad (2-26)$$

While x_L, \dot{x}_L vary on \mathbb{R}^3 , Equation 2-26 is also a function of variations on manifolds, where δR is a variation on $SO(3)$ and δq is a variation on \mathbb{S}^2 . These so called infinitesimal variations define how the curves on the manifold "vary", and are obtained as shown in [1, 23, 20, 22, 11].

$$\begin{aligned}\delta R &= R \hat{\eta} \in T_R SO(3), \text{ where } \eta \in \mathbb{R}^3, \hat{\eta} \in \mathfrak{so}(3) \\ \delta q &= \xi \times q \in T_q \mathbb{S}^2, \text{ where } \xi \in \mathbb{R}^3, \xi \cdot q = 0\end{aligned}\quad (2-27)$$

The following variations follow from differentiation,

$$\begin{aligned}\delta \dot{q} &= \dot{\xi} \times q + \xi \times \dot{q}, \\ \delta \dot{R} &= \dot{R} \hat{\eta} + R \hat{\eta}, \\ \delta \hat{\Omega} &= \delta(R^T \dot{R}) \\ &= \delta R^T \dot{R} + R^T \delta \dot{R} \\ &= (R \hat{\eta})^T \dot{R} + R^T (\dot{R} \hat{\eta} + R \hat{\eta}) \\ &= \hat{\eta}^T \hat{\Omega} + \hat{\Omega} \hat{\eta} + \hat{\eta} \\ &= (\hat{\Omega} \eta)^\wedge + \hat{\eta}, \\ \delta \Omega &= (\hat{\Omega} \eta) + \dot{\eta}\end{aligned}\quad (2-28)$$

These variations are substituted into Equation 2-26, allowing it to be a function of variations in each generalized coordinate.

$$\begin{aligned}\delta S &= \int_{t_1}^{t_2} (\delta W_1 + \delta W_2 + \delta \mathcal{T} - \delta \mathcal{U}) dt \\ &= \int_{t_1}^{t_2} (((m_Q + m_L)\dot{x}_L - m_Q L \dot{q}) \cdot \delta \dot{x}_L + (f R e_3 - (m_Q + m_L) g e_3) \cdot \delta x_L) dt \\ &\quad + \int_{t_1}^{t_2} ((m_Q L^2 \dot{q} - m_Q L \dot{x}_L) \cdot \delta \dot{q} + (m_Q g L e_3 - f L R e_3) \cdot \delta q) dt \\ &\quad + \int_{t_1}^{t_2} (\Omega^T J \cdot \delta \Omega + M \cdot (R^T \delta R)) dt\end{aligned}\tag{2-29}$$

After rearranging and setting each variation to 0, the following equations of motion for the QR-load system are found.

$$\frac{d}{dt} x_L = \dot{x}_L\tag{2-30}$$

$$(m_Q + m_L)(\ddot{x}_L + g e_3) = (q \cdot f R e_3 - m_Q L(\dot{q} \cdot \dot{q}))q\tag{2-31}$$

$$\dot{q} = \omega \times q\tag{2-32}$$

$$m_Q L \dot{\omega} = -q \times f R e_3\tag{2-33}$$

$$\dot{R} = R \hat{\Omega}\tag{2-34}$$

$$J \dot{\Omega} + \Omega \times J \Omega = M\tag{2-35}$$

where Equations 2-31 and 2-33 are the load position and attitude dynamics, and Equation 2-35 represents the QR attitude dynamics. Equation 2-33 can be rewritten, directly in terms of the load attitude q as follows

$$m_Q L \ddot{q} + m_Q L(\dot{q} \cdot \dot{q})q = q \times (q \times f R e_3)\tag{2-36}$$

The dynamics of the complete QR-load system can be globally expressed on the Special Orthogonal Group $SO(3)$, *two-sphere* \mathbb{S}^2 and Special Euclidean Group $SE(3)$, which are all smooth manifolds. This results in a compact notation of the equations of motion, making the large amount of trigonometric functions unnecessary, that normally are introduced by Euler angles.

Summary

In this chapter, the dynamical model of the quadrotor-load system was derived. The motivation to use geometric mechanics and a basic understanding of its concepts are given in order to understand the difference between a nonlinear geometric model and a model obtained with classical modeling approaches.

With the tools of differential geometry, the system dynamics are expressed on nonlinear configuration manifolds, which results in a globally defined, compact, unambiguous representation of the model. This dynamical model is used for a nonlinear geometric control approach, which is discussed in the next chapter.

Chapter 3

Control Design

Section 3-1 introduces nonlinear geometric control and concepts of geometric properties that are used for analysis and control design. In the previous chapter, the configuration spaces of the system dynamics were expressed on nonlinear manifolds. Error functions and geometric mappings are defined on these same nonlinear manifolds in order to measure the error between current and desired states.

A backstepping control approach is applied to control the under-actuated nature of the system. This control design consists of multiple controllers operating in a cascaded structure. Control laws guarantee stabilization of the closed-loop system by using the states as virtual control inputs. This approach results in the possibility to track a load position, while stabilizing the system. Different flight modes and the corresponding controllers are discussed in Section 3-2.

3-1 Nonlinear Geometric Control

Many control systems are developed for the standard form of ordinary differential equations $\dot{x} = f(x, u)$, where x is the state and u the control input. It is assumed that the state and the control input lie in Euclidean spaces, and the system equations are defined in terms of smooth functions between Euclidean spaces. However, for many mechanical systems, the configuration space can only be expressed locally as a Euclidean space. A nonlinear space is required to express the configuration space globally, which is discussed in the previous chapter.

Geometric control theory is the study on how geometry of the state space influences control problems. In control systems engineering, the underlying geometric features of a dynamic system are often not considered carefully. Differential geometric control techniques utilize these geometric properties for control system design and analysis. The objective is to express both the system dynamics and control inputs on nonlinear manifolds instead of local charts. In contrast to locally defined linear control, nonlinear geometric control can be defined almost globally, avoiding singularities that would occur in the representation of large angles and complex maneuvering.

The design of the controllers for the QR attitude can be found in [12], and the controllers of load attitude- and position can be found in [1]. Thorough stability analyses are presented in these references. For a deeper understanding of Lyapunov stability analysis in geometric control, the reader can refer to [11]. Other control systems that are able to switch between control modes, such as the hybrid control described in [24], require complicated reachability set analysis to guarantee safe switching between different flight modes. nonlinear geometric control does not require such analysis, as the region of attraction for each flight mode covers the configuration space almost globally. A study on global nonlinear dynamics of various classes of closed loop attitude control systems can be found in [25].

3-1-1 Error Functions

The control of a trajectory tracking problem requires state feedback to define tracking errors, a measure of the difference between the current states and the desired states. Since the closed-loop system dynamics evolve on nonlinear manifolds, which describe the configuration space of the QR attitude $\in SO(3)$ and the load attitude $\in \mathbb{S}^2$, error functions are defined on these same manifolds [11]. These functions play a role in the definition of the potential function for the closed-loop system and form the basis for both stabilizing and tracking controllers of the QR-load system.

Quadrotor Attitude Error

Recall that R is the rotation matrix to describe the QR attitude, and R_d is the desired rotation matrix. To describe the relative rotation from the body frame to the desired frame, an *attitude error* is defined as $R_d^T R$. Note that $R_d^T R$ is again a rotation matrix itself. Based on this attitude error, the *tracking error function* Ψ_R on $SO(3)$ is chosen to be

$$\Psi_R(R, R_d) = \frac{1}{2} \text{tr} \left[I - R_d^T R \right] \quad (3-1)$$

such that Ψ_R is locally positive-definite about $R_d^T R = I$ within the region where the rotation angle between R and R_d is less than 180° . It can be shown that this region where $\Psi_R < 2$ almost covers $SO(3)$ [26].

Using Equation 2-5 and 2-27, the derivative of the tracking error function Ψ_R with respect to R along the direction of $\delta R = R\hat{\eta}$ for $\eta \in \mathbb{R}^3$ is given by

$$\begin{aligned}\mathbf{D}_R \Psi(R, R_d) \cdot R\hat{\eta} &= -\frac{1}{2} \text{tr}[R_d^T R\hat{\eta}] \\ &= \frac{1}{2}(R_d^T R - R^T R_d)^\vee \cdot \eta\end{aligned}\quad (3-2)$$

where the *vee map* $\vee : \mathfrak{so}(3) \rightarrow \mathbb{R}^3$ is the inverse of the *hat map* defined in Section 2-1. From this equation, the QR *attitude tracking error* $e_R \in \mathbb{R}^3$ is chosen as follows

$$e_R = \frac{1}{2}(R_d^T R - R^T R_d)^\vee \quad (3-3)$$

It is important to note that the velocities \dot{R} and \dot{R}_d cannot be compared directly, since they do not lie in the same space. At time $t = t_0$, assume that $R(t_0) = q$ and $R_d(t_0) = r$, then \dot{R} and \dot{R}_d lie in their own tangent spaces, denoted by $T_q SO(3)$ and $T_r SO(3)$, respectively. For this reason, \dot{R}_d must be transformed into a vector on $T_q SO(3)$ to allow a meaningful comparison with \dot{R} . Defining a velocity error can be achieved with a mathematical object called a *transport map*, which enables the comparison of tangent vectors living in different spaces.

In Figure 3-1, two curves $R(t)$ and $R_d(t)$ evolve on manifold $SO(3)$. Transport map $\mathcal{T}(q, r) : T_r SO(3) \mapsto T_q SO(3)$ allows comparison of the velocity curves \dot{R} and \dot{R}_d .

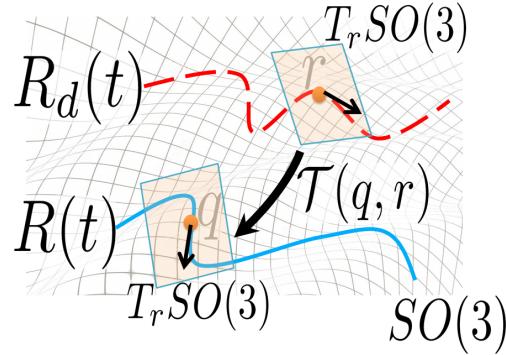


Figure 3-1: Transport map $\mathcal{T}(q, r)$

The *velocity error* \dot{e} is a vector field along R corresponding to the transport map. It defines the velocity error between the curves R and R_d , and is defined as

$$\dot{e} = \dot{R} - \dot{R}_d(R_d^T R) \quad (3-4)$$

This equation is rewritten to obtain the angular velocity tracking error, as follows

$$\begin{aligned}\dot{R} - \dot{R}_d(R_d^T R) &= R\hat{\Omega} - R_d\hat{\Omega}_d(R_d^T R) \\ &= R(\Omega)^\wedge - (RR^T)R_d\hat{\Omega}_dR_d^T R \\ &= R(\Omega)^\wedge - R(R^T R_d\Omega_d)^\wedge \\ &= R(\Omega - R^T R_d\Omega_d)^\wedge\end{aligned}\quad (3-5)$$

The angular velocity tracking error e_Ω expressed in $\{\mathcal{B}\}$ is defined as

$$e_\Omega = \Omega - R^T R_d \Omega_d \quad (3-6)$$

Similar to the form of Equation 2-34, e_Ω represents the angular velocity vector of the relative rotation matrix $R_d^T R$, represented in $\{\mathcal{B}\}$. Hence, it can be shown that the following equation holds

$$\frac{d}{dt}(R_d^T R) = (R_d^T R)\hat{e}_\Omega \quad (3-7)$$

The values of the QR attitude tracking error e_R and the QR angular velocity tracking error e_Ω are used later on to design control for the QR attitude.

Load Attitude Error

The load attitude dynamics evolve on \mathbb{S}^2 and its tangent space $T\mathbb{S}^2$, where the error of the load attitude is described in a similar approach. The error between the load attitude q and the desired load attitude q_d is defined by the error function $q_d^T q$. Based on the error function, the tracking error function Ψ_q on \mathbb{S}^2 is chosen to be

$$\Psi_q = 1 - q_d^T q \quad (3-8)$$

The derivative of the tracking error function Ψ_q is given by

$$d_1\Psi_q(q, q_d) = \dot{q}^2 q_d \quad (3-9)$$

From this equation, the load attitude error function e_q is defined as follows

$$e_q = \dot{q}^2 q_d \quad (3-10)$$

Again, a *transport map* is used for a comparison between the tangent vectors on different tangent spaces. Using the tracking error Ψ_q and the transport map $\mathcal{T}_{\mathbb{S}^2}$, a closed-loop energy function evolving on \mathbb{S}^2 is derived [11, 11.3.2]. From this energy function, the load angular velocity error function is defined as

$$e_{\dot{q}} = \dot{q} - (q_d \times \dot{q}_d) \times q \quad (3-11)$$

The values of the load attitude tracking error e_q and the load angular velocity tracking error $e_{\dot{q}}$ are used later on to design control for the load attitude.

Load Position Error

The tracking errors for the load position and load velocity are defined as

$$e_x = x - x_d \quad (3-12)$$

$$e_v = v - v_d \quad (3-13)$$

where $v_d = \dot{x}_d$. Furthermore, $x_d(t) \in \mathbb{R}^3$ must be a smooth twice-differentiable load trajectory, such that functions are well defined. The values of the position error e_x and the velocity error e_v are used later on to design control for the load position.

3-2 Backstepping Control

A backstepping approach will be used in this research for the control of the load trajectory tracking problem, and is commonly used for QR control[27]. Backstepping control is a Lyapunov based control technique for the stabilization of nonlinear dynamical systems, developed by [28]. The method relies on a *triangular structure* of the system in a certain set of coordinates. The system is split into subsystems in a cascaded structure and recursive techniques allow a systematically design of feedback control laws and corresponding Lyapunov functions.

The control structure is created by starting with a stable system as the most inner subsystem. By "stepping back" from this subsystem, a control loop can be added around it containing a control law that defines a change of coordinates. This creates a virtual control input, that must stabilize the inner structure. The control law is designed by using states as virtual control inputs, such that each loop computes a virtual command signal for the adjacent inner loop. This is repeated until the final external control is reached.

The backstepping approach is able to calculate the control inputs f and M that are required to stabilize the QR, while several controllers are able to track different states, see Figure 3-2. The inner controller determines what the required control inputs are, driven by R_c . The next controller calculates how to drive the computed rotation matrix R_c based on q_c , such that the QR is stabilized. And the last controller determines which load attitude q_c is required, such that the desired load position $x_{L,d}$ is tracked.

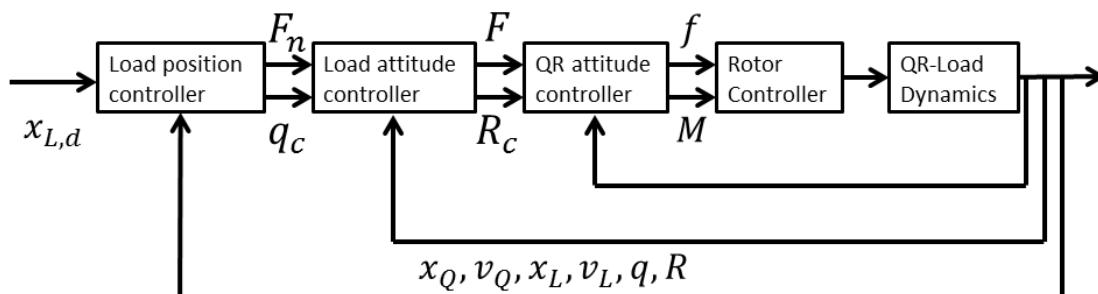


Figure 3-2: Nonlinear geometric control loop of the QR-load system [1]

Since the QR has only four actuators, it is not possible to control all DOFs of the QR-load system simultaneously. The backstepping approach allows control of different flight modes in which a combination of DOFs is controlled. The flight modes and their functions are defined below in order, from the most inner loop to the most outer loop.

- QR Attitude controlled Mode
 - Track a desired QR attitude $R_d(t)$ or commanded signal $R_c(t)$
 - Optional tracking of a desired heading direction $b_{1d}(t)$, the first column of $R_d(t)$
 - Calculate the control input M for the QR-load system
- load Attitude controlled Mode
 - Track a desired load attitude $q_d(t)$ or commanded signal $q_c(t)$

- Calculate a computed QR attitude R_c for the QR attitude controller
- Calculate the control input f for the QR-load system
- load Position controlled Mode
 - Track a desired load position $x_{L,d}(t)$
 - Calculate a computed load attitude q_c for the load attitude controller
 - Calculate the control input f for the QR-load system

where the subscript d denotes a desired tracking reference, which must be given whenever it is not calculated by a controller. For example in Figure 3-3, where the QR attitude controller works as a standalone controller, without taking load attitude and position into account. The subscript c denotes a computed tracking reference, calculated by the controllers.

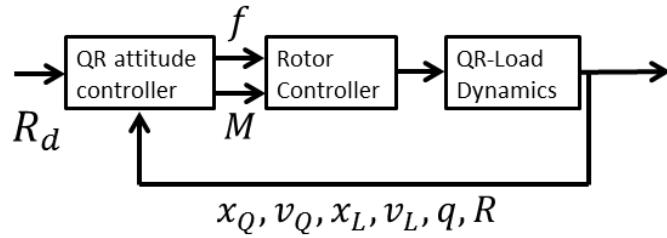


Figure 3-3: QR Attitude controller

3-2-1 Quadrotor Attitude Tracking

The QR Attitude controlled Mode is designed to control the QR attitude by tracking a smooth desired QR attitude $R_d(t)$. Analysis of the error dynamics e_R and e_Ω requires the calculation of their time derivatives.

Using Equations 3-3 and 3-6, the derivative of the attitude tracking error e_R can be written as

$$\dot{e}_R = \frac{1}{2}(R_d^T R \hat{e}_\Omega + \hat{e}_\Omega R^T R_d)^\vee \quad (3-14)$$

The derivative of the angular velocity tracking error e_Ω , follows from Equations 2-34, 3-6 and $\hat{\Omega}_d \Omega_d = 0$, such that

$$\dot{e}_\Omega = \dot{\Omega} + \hat{\Omega} R^T R_d \Omega_d - R^T R_d \dot{\Omega}_d \quad (3-15)$$

Recall from Equation 2-34, that the kinematics equation for the desired attitude can be written as

$$\dot{R}_d = R_d \hat{\Omega}_d \text{ and } \hat{\Omega}_d = R_d^T \dot{R}_d \quad (3-16)$$

The desired angular acceleration $\dot{\Omega}_d$ follows from the following equation

$$\begin{aligned}
 \dot{\hat{\Omega}}_d &= (\dot{R}_d^T \dot{R}_d) + (R_d^T \ddot{R}_d) \\
 &= (R_d \hat{\Omega}_d)^T (R_d \hat{\Omega}_d) + (R_d^T \ddot{R}_d) \\
 &= -\hat{\Omega}_d \hat{\Omega}_d + R_d^T \ddot{R}_d
 \end{aligned} \quad (3-17)$$

such that,

$$\dot{\Omega}_d = (-\hat{\Omega}_d \hat{\Omega}_d + R_d^T \ddot{R}_d)^\vee \quad (3-18)$$

By substituting Equation 2-35 into Equation 3-15, the following equation is obtained

$$\dot{e}_\Omega = J^{-1}(-\Omega \times J\Omega + M) + \hat{\Omega} R^T R_d \Omega_d - R^T R_d \dot{\Omega}_d \quad (3-19)$$

From this, the control input M is defined [26], and consists of a proportional term, a derivative term and a canceling term, as follows

$$M = -k_R e_R - k_\Omega e_\Omega + \Omega \times J\Omega - J(\hat{\Omega} R^T R_d \Omega_d - R^T R_d \dot{\Omega}_d) \quad (3-20)$$

Rapid exponential convergence of the attitude error function and angular velocity error function can be achieved by adding the parameter ϵ to Equation 3-20 as done in [1], where $0 < \epsilon < 1$, given by

$$M = -\frac{1}{\epsilon^2} k_R e_R - \frac{1}{\epsilon} k_\Omega e_\Omega + \Omega \times J\Omega - J(\hat{\Omega} R^T R_d \Omega_d - R^T R_d \dot{\Omega}_d) \quad (3-21)$$

Substituting Equation 3-21 into Equation 3-19 results in the time derivative of the angular velocity tracking error, given by

$$\dot{e}_\Omega = J^{-1}\left(-\frac{1}{\epsilon^2} k_R e_R - \frac{1}{\epsilon} k_\Omega e_\Omega\right) \quad (3-22)$$

for any positive constants k_R, k_Ω .

Equation 2-6 is used to rewrite the time derivative of e_R as follows

$$\begin{aligned} \dot{e}_R &= \frac{1}{2}(R_d^T R \hat{e}_\Omega + \hat{e}_\Omega R^T R_d)^\vee \\ &= \frac{1}{2}(tr[R^T R_d]I - R^T R_d)e_\Omega \equiv C(R_d^T R)e_\Omega \end{aligned} \quad (3-23)$$

where $\|C(R_d^T R)\|_2 \leq 1$, such that $\|\dot{e}_R\| \leq \|e_\Omega\|$ for all $R_d^T R \in SO(3)$, guaranteeing that \dot{e}_R will be bounded, whenever e_Ω is bounded. In [12] it is proven that the Lyapunov functions, which are functions of the error dynamics $e_R, e_\Omega, \dot{e}_R, \dot{e}_\Omega$ described above, are non-increasing and bounded. With this controller stability analysis, it is proven that the zero equilibrium of the closed loop tracking error $(e_R, e_\Omega) = (0, 0)$ is exponentially stable, if the initial conditions satisfy

$$\Psi_R(R(0), R_d(0)) < 2 \quad (3-24)$$

$$\|e_\Omega(0)\|^2 < \frac{2}{\lambda_M(J)} \frac{k_R}{\epsilon^2} (2 - \Psi_R(R(0), R_d(0))) \quad (3-25)$$

where $\lambda_M(\cdot)$ denotes the maximum eigenvalue.

Furthermore, there exist constants $\alpha_R, \beta_R > 0$ such that

$$\Psi_R(R(t), R_d(t)) \leq \min \left\{ 2, \alpha_R e^{-\beta_R t} \right\} \quad (3-26)$$

Equations 3-24 and 3-25 determine the domain of attraction, which is the region in which the trajectory of the system is able to converge to an asymptotically stable equilibrium point. The domain of attraction almost covers $SO(3)$, this is referred to as almost-global exponential attractiveness.

Note that the tracking of the QR attitude does not require any specification of the thrust magnitude f . During this flight mode, the translational motion can only be controlled partially, which makes this flight mode suitable for attitude maneuvers with short time periods. For a QR system without load, in [12] a tracking controller is described that calculates both a thrust magnitude f and total moment M in order to track a QR position trajectory.

3-2-2 Load Attitude Tracking

The load Attitude controlled Mode is designed to track a desired load attitude q_d . In order to influence the load dynamics, see Equation 2-33, the load attitude controller calculates a computed QR attitude R_c for the QR attitude controller, which replaces R_d .

The commanded directions of the body frame axes are defined by R_c , and is defined as follows

$$R_c = [b_{1c}; b_{3c} \times b_{1c}; b_{3c}] \quad (3-27)$$

Ω_d is replaced by Ω_c , where Ω_c is defined by

$$\hat{\Omega}_c = R_c^T \dot{R}_c \quad (3-28)$$

which will influence the QR attitude dynamics, see Equation 2-35.

The unit vector b_{1c} is the first column of R_c and is constructed by normalizing the projection of a desired heading angle $b_{1d} \in \mathbb{S}^2$ onto the plane normal to b_{3c} , see Figure 3-4. Defining $b_{1c} \in \mathbb{S}^2$ orthogonal to b_{3c} guarantees that $R_c \in SO(3)$ [26].

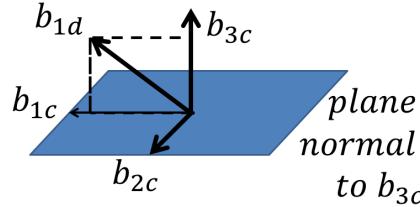


Figure 3-4: Construction of b_{1c} to define R_c

b_{1c} is defined as follows

$$b_{1c} = -\frac{1}{\|b_{3c} \times b_{1d}\|} (b_{3c} \times (b_{3c} \times b_{1d})) \quad (3-29)$$

Such that $b_{2c} = b_{3c} \times b_{1c}$, and b_{1d} is chosen, not parallel to $b_{3c} \in \mathbb{S}^2$, the third column of R_c . b_{3c} is defined by a normalization of F ,

$$b_{3c} = \frac{F}{\|F\|} \quad (3-30)$$

where F is defined by a normal component F_n , a proportional-derivative component F_{pd} and feedforward control force F_{ff}

$$F = F_n - F_{pd} - F_{ff} \quad (3-31)$$

The inclusion of F_n ensures that b_{3c} is always well defined. F_n is defined as

$$F_n = -(q_d \cdot q)q \quad (3-32)$$

The control forces F_{pd} and F_{ff} are defined for trajectory tracking in [11, 11.2.5] as follows

$$\begin{aligned} F_{pd} &= -k_P \hat{q}^2 q_d - k_D (\dot{q} - (q_d \times \dot{q}_d) \times q) \\ &= -k_q e_q - k_\omega e_{\dot{q}} \end{aligned} \quad (3-33)$$

and

$$F_{ff} = m_Q L \langle\langle q, q_d \times \dot{q}_d \rangle\rangle_{\mathbb{R}^3} (q \times \dot{q}) + m_Q L (q_d \times \ddot{q}_d) \times q \quad (3-34)$$

for positive constants k_q, k_ω .

This controller affects the control input f , which is defined as

$$f = F \cdot Re_3 \quad (3-35)$$

and the control input M will be defined as

$$M = -\frac{1}{\epsilon^2}k_R e_R - \frac{1}{\epsilon}k_\Omega e_\Omega + \Omega \times J\Omega - J(\hat{\Omega}R^T R_c \Omega_c - R^T R_c \dot{\Omega}_c) \quad (3-36)$$

It is proven in [1] and [11, Lemma 11.23] that the zero equilibrium of the closed loop tracking error $(e_q, e_{\dot{q}}, e_R, e_\Omega) = (0, 0, 0, 0)$ is exponentially stable, if the initial conditions satisfy

$$\Psi_q(q(0), q_d(0)) < 2 \quad (3-37)$$

$$\|e_{\dot{q}}(0)\|^2 < \frac{2}{m_Q L} k_R (2 - \Psi_q(q(0), q_d(0))) \quad (3-38)$$

The domain of attraction is defined by Equations 3-24, 3-25, 3-37 and 3-38. Equation 3-37 states that the initial load attitude error should be less than 180° , which means that the controller achieves almost-global exponential convergence for load attitude q . Furthermore, there exist constants $\alpha_q, \beta_q > 0$ such that

$$\Psi_q(q(t), q_d(t)) \leq \min \left\{ 2, \alpha_q e^{-\beta_q t} \right\} \quad (3-39)$$

3-2-3 Load Position Tracking

The load Position controlled Mode is designed to track a desired load position $x_{L,d}$. Analysis of the error dynamics e_x and e_v requires the calculation of their time derivatives.

The derivative of the load position error e_x is given by

$$\dot{e}_x = e_v \quad (3-40)$$

and from Equation 2-31 and given that $\dot{e}_v = \ddot{x}_L - \ddot{x}_{L,d}$ follows

$$(m_Q + m_L)\dot{e}_v = -(m_Q + m_L)(ge_3 + \ddot{x}_{L,d}) - m_Q L(\dot{q} \cdot \dot{q})q + (q \cdot fRe_3)q \quad (3-41)$$

Equations 3-40 and 3-41 are used in a stability analysis of the controller. The load position controller calculates a computed load attitude q_c for the load attitude controller. R_d and q_d are replaced by R_c and q_c , respectively. In order to stabilize the error dynamics, it is proven in [1] that the required computed load attitude is defined as

$$q_c = -\frac{A}{||A||} \quad (3-42)$$

where

$$A = -k_x e_x - k_v e_v + (m_Q + m_L)(\ddot{x}_{L,d} + ge_3) + m_Q L(\dot{q} \cdot \dot{q})q \quad (3-43)$$

Furthermore, Equation 3-32 is redefined as

$$F_n = (A \cdot q)q \quad (3-44)$$

which is substituted in Equation 3-31, resulting in a new control input f .

This controller ensures that the zero equilibrium of the closed loop tracking error $(e_x, e_v, e_q, e_{\dot{q}}, e_R, e_{\Omega}) = (0, 0, 0, 0, 0, 0)$ is exponentially attractive, if the initial conditions satisfy

$$\Psi_q(q(0), q_c(0)) < \psi_1 < 1 \quad (3-45)$$

$$\| e_x(0) \|^2 < e_{x_{max}} \quad (3-46)$$

where $e_{x_{max}}$ and ψ_1 are fixed design depended constants.

The domain of attraction is defined by Equations 3-24, 3-25, 3-45 and the following equation

$$\| e_{\dot{q}}(0) \|^2 < \frac{2}{m_Q L} k_q (\psi_1 - \Psi_q(q(0), q_d(0))) \quad (3-47)$$

Summary

In this chapter, control design based on nonlinear geometric control is discussed. What is particular in this control technique, is the fact that error functions are defined on non-Euclidean manifolds, similar to the manifolds that describe the configuration space of the system. Since these manifolds are locally Euclidean, local stability properties of a closed-loop equilibrium solution can be determined by using standard Lyapunov methods. Based on these error functions, controllers are designed in a backstepping approach, enabling both load position tracking and stabilization of the system. Using the geometric properties of the system allows the design of globally defined controllers that ensure almost-global convergence of the QR attitude and load attitude. In order to test the control performance of a load position tracking objective, experiments are defined in the next chapter.

Chapter 4

Experiment

The experimental procedure is explained in Section 4-1. It is discussed what experiments can be done in order to investigate closed-loop stability and the tracking performance of nonlinear geometric control. In addition, a comparison is made between the performances of the nonlinear geometric controller and a linear LQR controller.

The controllers are tested on their ability to track a desired load trajectory. Section 4-2 presents several desired load trajectories that create different challenges for load position tracking, and it is discussed what could be expected from these experiments.

In Section 4-3 the experimental setup is discussed. The model parameters for the QR-load system are presented, as well as the controller parameters for both nonlinear geometric controller and LQR controller. The notion of a backstepping command filter is made to explain a mathematical simplification in the experiments.

Finally, in Section 4-4 the results that are obtained from the load trajectory tracking experiments are presented and discussed. The stability of the closed-loop system is demonstrated for the nonlinear geometric controller and the differences in linear- and nonlinear controller performance are discussed.

4-1 Procedure

The goal of the experiments is to analyze the controller performance and closed-loop stability in a load position tracking task. The load positions are described by smooth trajectories $x_{L,d}(t)$ in order to get well-defined control functions. In this work the desired load paths are generated by hand. The corresponding required velocity and acceleration are calculated by a *command filter*, which is explained in more detail in Section 4-3. For the purpose of load transportation, both controllers can be used. The difference is in the definition of the problem.

As described in Chapter 3, the stability of the nonlinear geometric controller is evaluated by analyzing the error functions to check whether 1) the zero equilibrium of the closed loop tracking error $(e_x, e_v, e_q, e_{\dot{q}}, e_R, e_{\Omega}) = (0, 0, 0, 0, 0, 0)$ is exponentially stable and 2) the tracking errors Ψ_R and Ψ_q are bounded by an exponential decay function and the maximum error. Performance of both nonlinear geometric control and LQR control are evaluated by comparing their ability to track a load trajectory with minimal error. The differences based on response time, load tracking accuracy and peculiarities are analyzed and discussed.

A linearized model is obtained by assuming small angles of both load and QR around an equilibrium point. The model is obtained by assuming an equilibrium point such that the QR is in a hover position with the load hanging directly underneath it. The LQR cost function allows control of the states that define the QR position, QR attitude and load attitude by calculating the control inputs f and M , in such a way that the system is stable. As a result, the linearized model does not allow direct reference tracking of the load position. This limitation illustrates an important difference between the use of a linear and a nonlinear model.

The LQR controller in this thesis is designed to track a QR position and to minimize the load swing. The tuning of the LQR controller involves a trade-off between accurate QR movements and minimal load swing. The QR position is based on the same desired load trajectories that are used for the nonlinear geometric controller. When assuming small angles and minimal load swing, the QR position should be approximately a cable length above the predefined desired load position. Note that this will not allow a direct comparison of the load trajectory tracking performance, nevertheless this will illustrate important differences between the controllers.

4-2 Trajectories

This section discusses a number of cases that describe different load trajectories for the QR-load system. The trajectories are generated to obtain closed-loop responses and test stability of the system. A description of the desired trajectory is given in each case and the challenges that are involved are discussed.

Case A

In the first case, a smooth step-like trajectory is generated to investigate the step response of the system. The step response is a commonly used analysis tool to obtain information about the stability of a dynamical system. A step function is used to investigate the effects of a sudden input to the system. Typical response properties that can be investigated are: rise time, overshoot, settling time and steady-state error. The goal is to transport the load from a starting position to a final position along the y-axis.

The geometric control design requires the desired load trajectory to be twice-differentiable. For this reason a smooth step-like function is generated. Figure 4-1 shows the desired load position, velocity and acceleration in the y-direction. The desired position, velocity and acceleration in x- and z-direction are zero. It can be expected that the step response is only able to track a trajectory up to a limited steepness. The commanded acceleration in Equation 3-43 is required to be bounded, such that

$$\| (m_Q + m_L)(\ddot{x}_{L,d} + g e_3) + m_Q L(\dot{q} \cdot \dot{q})q \| < B \quad (4-1)$$

where B is a positive constant [12]. This might result in large errors, and it must be investigated whether the system is able to maintain stable.

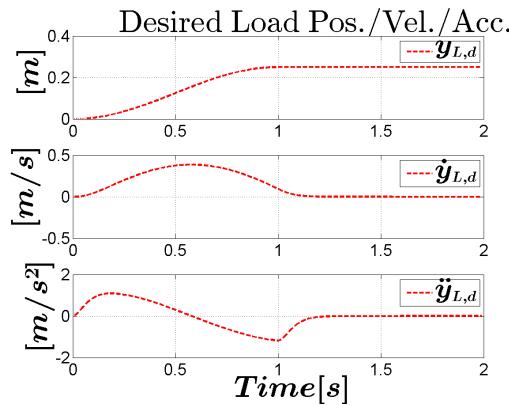


Figure 4-1: Desired load trajectory Case A, y-direction

Case B

The nonlinear geometric controllers are expected to be able to deal with large angles in the QR attitude, allowing aggressive maneuvering. This can be tested by describing the next trajectory as a sine wave, growing in amplitude over time, in the direction of one axis in

$\{\mathcal{I}\}$. This will result in an increasing distance between the load positions at each end of the movement, requiring increasing velocities on both the QR and the load. To achieve this, it can be expected that the QR requires large rotations. It is investigated whether the system is able to perform load position tracking while dealing with increasing QR rotations.

The trajectory is generated by the product of signals A and B, shown in Figure 4-2a, of which the first signal is a sine and the second consists of a smooth step up and down. This product results in signal C, shown in the same figure, which is chosen to be the trajectory in the direction of the y-axis of $\{\mathcal{I}\}$. Figure 4-2 shows the desired load position, velocity and acceleration in the y-direction. The desired position, velocity and acceleration in x- and z-direction are zero.

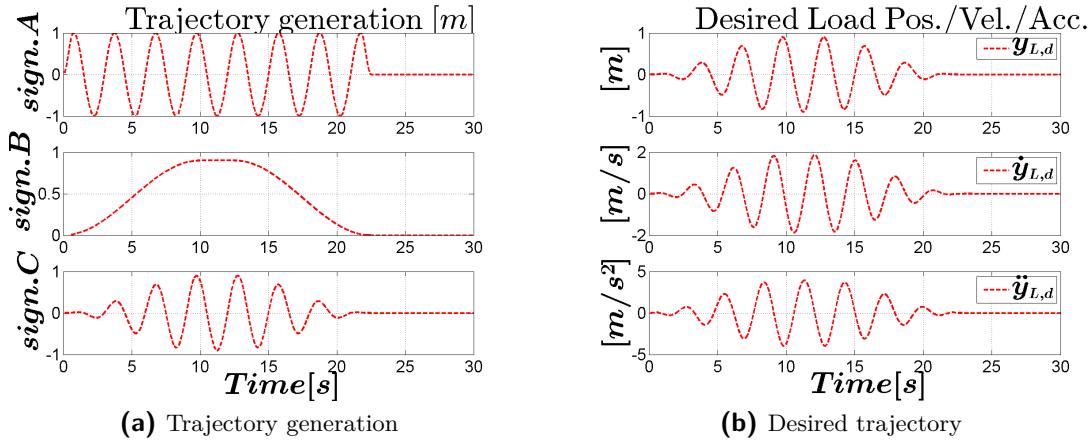


Figure 4-2: Desired load trajectory Case B, y-direction

Case C

This case is generated in order to investigate the response on tracking multiple conditions at the same time. The trajectory along the y-axis in $\{\mathcal{I}\}$ is described as a sine wave with an increasing and decreasing amplitude over time, as in case B. While following this wave, the load is commanded to move forward along the x-axis, while tracking an increasing and decreasing height. The changing amplitude of the trajectory that moves from side to side, requires varying velocities to 'keep up' with the trajectory.

In this case it can also be expected that large QR rotations are required to track the changing amplitude of the sine wave and the varying velocities. While doing so, the QR is expected to lose height due to the rotations. It can be investigated how the close-loop system responds to the forward movement, while tracking a swinging motion and whether the controller can correct for the expected height loss, while tracking an up and down movement.

Figure 4-3 shows the desired load position and a three dimensional representation, which can be seen as a *figure eight* increasing and decreasing in size.

Figure 4-4 shows the desired velocity and accelerations in all directions.

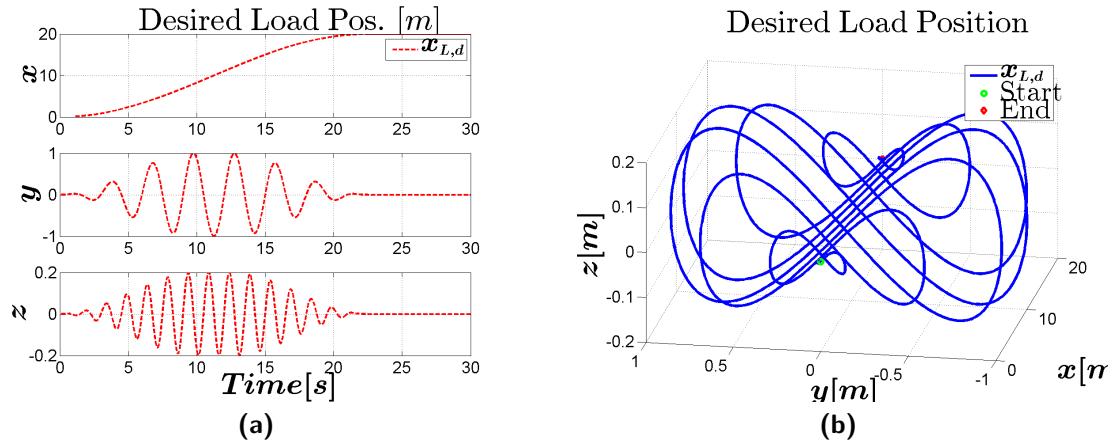


Figure 4-3: Desired load Position Case C

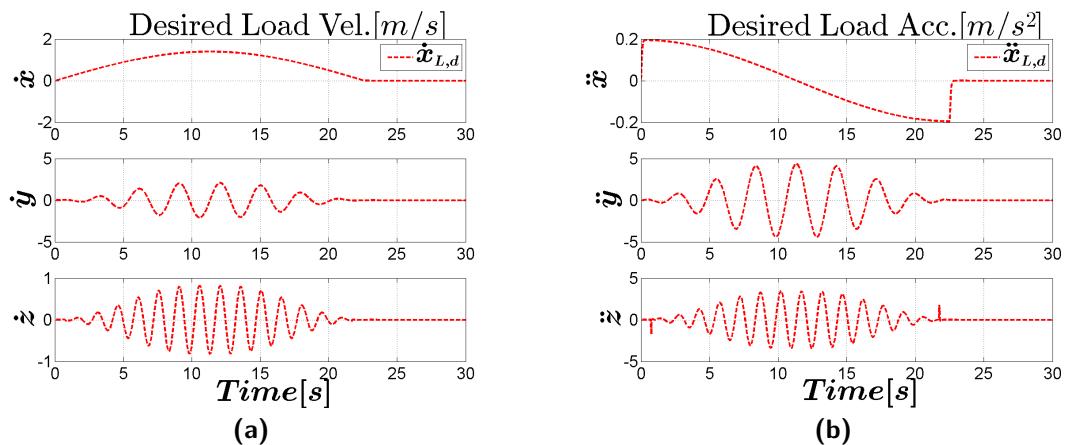


Figure 4-4: Desired load Position Case C

4-3 Setup

Model parameters The simulations are developed using Matlab[®] and Simulink[®] R2013b. The model parameters to define the system are based on a Parrot[®] Bebop Drone 1, also used in a research by [29], see Table 4-1. The maximum thrust per rotor defines the maximum total thrust and moment. m_L and L are chosen arbitrarily.

Parameter	Value	Description
m_Q	0.4 kg	Quadrotor Mass
l	0.126 m	Arm length from QR CM to rotor
I_{xx}	2.23×10^{-3} kgm^2	Quadrotor Inertia about x-axis
I_{yy}	2.99×10^{-3} kgm^2	Quadrotor Inertia about y-axis
I_{zz}	4.8×10^{-3} kgm^2	Quadrotor Inertia about z-axis
$f_{i,max}$	1.96 N	Maximum thrust per rotor
m_L	0.1 kg	Load Mass
L	0.7 m	Cable Length

Table 4-1: Modeling Parameters

In order to get a feel for realistic physical limits, these can be based on hardware data, which could be found on the web ¹. These values are shown in Table 4-2, and can be compared with the simulation results, in order to check whether the outcome is in the same order of magnitude of a real physical system.

Value	Description
13 m/s	Maximum top speed
2.5 m/s	Maximum vertical speed
30 deg	Maximum inclination

Table 4-2: Hardware Parameters for Bebop Parrot Drone 1

Command Filtering A consequence of a backstepping control approach, is that it increases the order of the commanded states. The controllers generate control laws that are a function of the commanded signals and their derivatives. As can be seen in Chapter 3, the control laws require the values R_c , \dot{R}_c , \ddot{R}_c , q_c , \dot{q}_c and \ddot{q}_c . Time derivatives of the virtual control variables may be quite complex and the calculation could result in high computational costs. Instead of analytic differentiation of these terms, which can be tedious, these values can be obtained with the use of a *Command Filter* [30, 31].

The state space implementation of such a filter is

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -2\zeta\omega_n - \omega_n^2(x_1 - x_c^o)\end{aligned}\tag{4-2}$$

¹<http://blog.parrot.com/2016/01/12/comparison-bebop-2-vs-bebop-drone/>

where $x_c = x_1$ and $\dot{x}_c = x_2$. If x_c^o is bounded, the x_c and \dot{x}_c are bounded and continuous. The transfer function from x_c^o to x_c is

$$\frac{X_c(s)}{X_c^o(s)} = H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (4-3)$$

which has a unity gain at low frequencies, ζ is the damping ratio and ω_n the undamped natural frequency.

The result of a command filter is that the command signal is being pre-filtered by low pass filters and generates an estimation of the derivatives of the commanded signal. Backstepping command filters are implemented to compute \dot{R}_c , \ddot{R}_c , \dot{q}_c , \ddot{q}_c , $\dot{x}_{L,d}$ and $\ddot{x}_{L,d}$.

The implemented command filter is shown in Figure 4-5.

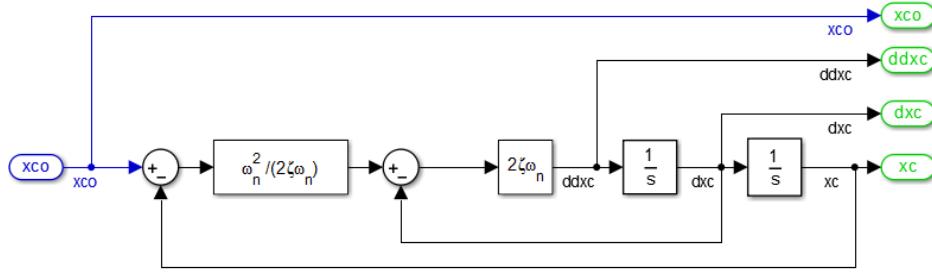


Figure 4-5: Command Filter

The chosen values for ω_n and ζ for the command filters are shown in Table 4-3. The parameters were determined by trial and error. The higher the value for ω_n , the higher the frequencies that are passed through the filter. Whenever the frequency was chosen too high, noisy derivatives are calculated, resulting in a destabilization of the system. Choosing the values too low resulted in slow responses, and a bad estimation of the derivatives. The damping ratio ζ was chosen high, in order to give a sufficient damping. Limits for the magnitude and rates can be implemented, which is further explained in [32].

	Filter		
	R	q	$x_{L,d}$
ω_n	30	25	25
ζ	0.98	0.98	0.98

Table 4-3: Command Filter Parameters

LQR control Linear Quadratic Regulator (LQR) control uses an algorithm to obtain a state-feedback controller, minimizing a cost function depending on the states and weight factors.

LQR control is based on a small angle assumption. Therefore, following a traditional modeling method, the rotation matrix is represented with a local coordinate system, for example with an Euler Angle parameterization. The implemented LQR design is shown in Figure 4-6.

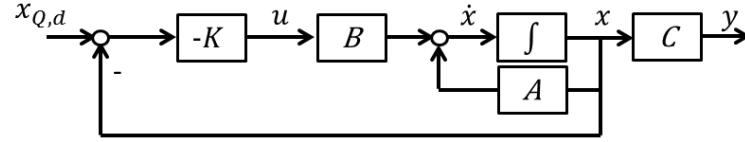


Figure 4-6: LQR control design

A continuous time linearized model of the system used in this controller is represented in the following form

$$\dot{\mathbf{x}} = A\mathbf{x} + Bu \quad (4-4)$$

$$y = C\mathbf{x} + Du \quad (4-5)$$

where \mathbf{x} is the state vector and u is the input vector, defined as follows

$$\begin{aligned} \mathbf{x} &= [x \ y \ z \ \phi \ \theta \ \psi \ \phi_L \ \theta_L \ \dot{x} \ \dot{y} \ \dot{z} \ \dot{\phi} \ \dot{\theta} \ \dot{\psi} \ \dot{\phi}_L \ \dot{\theta}_L]^T \\ u &= [f \ M_\phi \ M_\theta \ M_\psi]^T \end{aligned} \quad (4-6)$$

Using Matlab command `lqr(A,B,Q,R)`, an optimal gain matrix K is calculated, such that the state-feedback law $u = -K(\mathbf{x} - \mathbf{x}_{\text{ref}})$ [5] minimizes the quadratic cost function. Where the cost function is defined as

$$J(u) = \int_0^\infty (\mathbf{x}^T Q \mathbf{x} + u^T R u) dt \quad (4-7)$$

The weight matrices Q and R define the effects of the states and inputs in the cost function, and the gain matrix K can be calculated. The derivation of the state space matrices A, B, C, D , the weight matrices Q, R and the calculated gain matrix K can be found in Section A-1.

4-4 Results

In this section the results of the experiments are discussed, for each case separately. The load tracking performance for the nonlinear geometric controller is discussed by analyzing the desired and actual load trajectories, together with the corresponding load position and velocity errors.

The stability of the closed-loop system is investigated by observing whether the controller is able to bring the system to a stationary final state. The error dynamics are analyzed through the error functions, as described in Chapter 3. The configuration errors Ψ_R, Ψ_q and the corresponding tracking errors of the QR attitude e_R, e_Ω , load attitude $e_q, e_{\dot{q}}$ are presented. Finally, the load position tracking results of the nonlinear geometric controller and a LQR controller are compared.

The chosen controller gains in Equations 3-20, 3-33 and 3-43 can be found in Table 4-4.

Gain	Case A	Case B	Case C
k_R	0.9	0.9	0.9
k_Ω	0.1	0.1	0.1
k_q	3.9	3.9	3.9
k_ω	2	2	1.8
k_x	6	6	3.6
k_v	2.75	2.75	0.55

Table 4-4: Controller gains nonlinear geometric controller

Case A

In this case, the desired trajectory was shaped like a smooth step-like function to investigate the response of the system to a sudden input. The desired and actual load position, velocity and acceleration are shown in Figure 4-7a and 4-7b. The corresponding load position and velocity error are shown in Figure 4-7c.

Since this function is different from a normal step function, it might be less meaningful to say something about the rise time. Normally one checks the time it takes to reach 90% of the step height. In this case, the step is a smooth function, meaning that the system is not forced to the step height instantly, but in a smooth manner. However, it can be seen in Figure 4-10c that the required time such that 90% of the desired value is reached for the first time is 0.61s. It can be observed that the system responds with approximately 15% overshoot in the y-direction, and loses height during this maneuver. This was to be expected due to large QR rotations. The error remains within a 5% error bound after 1.26s, to eventually reach a steady-state at the step size of 0.25m.

Figure 4-8a and 4-8b show the tracking errors of the QR attitude and load attitude, respectively. The error in QR attitude and angular velocity are large at the moments the slope starts and ends, indicating the required sudden QR rotation and high angular velocity are larger than the system can manage. Nevertheless, the figures indicate that the errors in both QR and load attitude are driven to zero relatively fast. Furthermore, Figure 4-9a and 4-9b

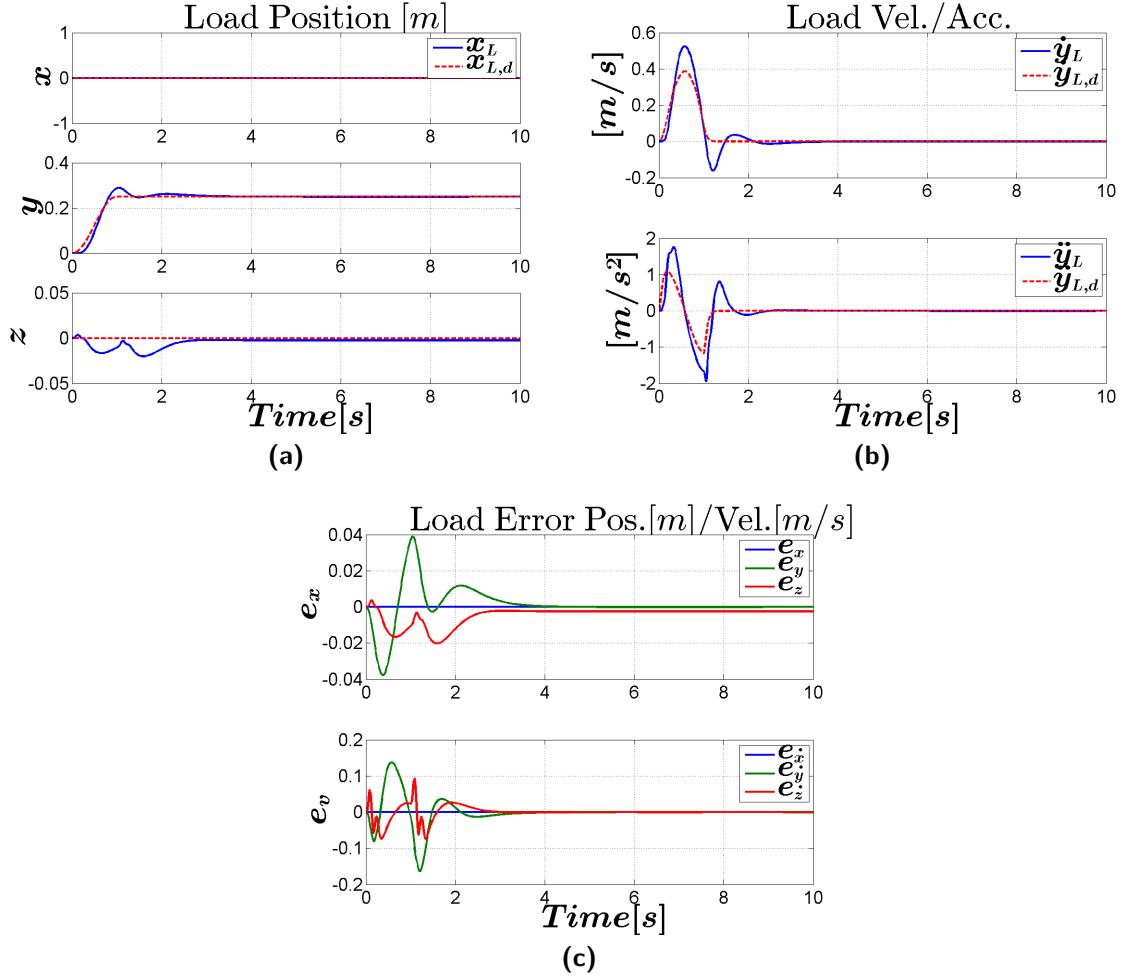


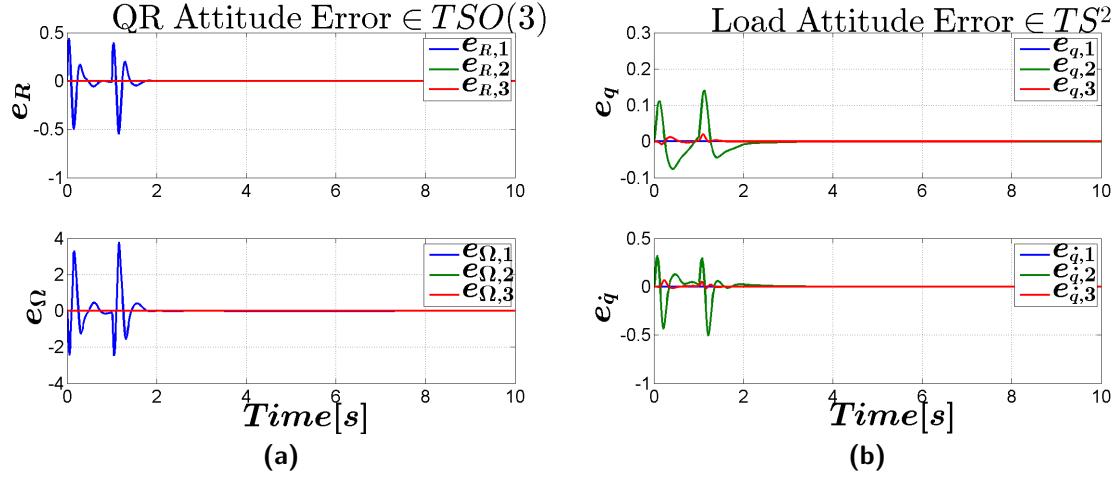
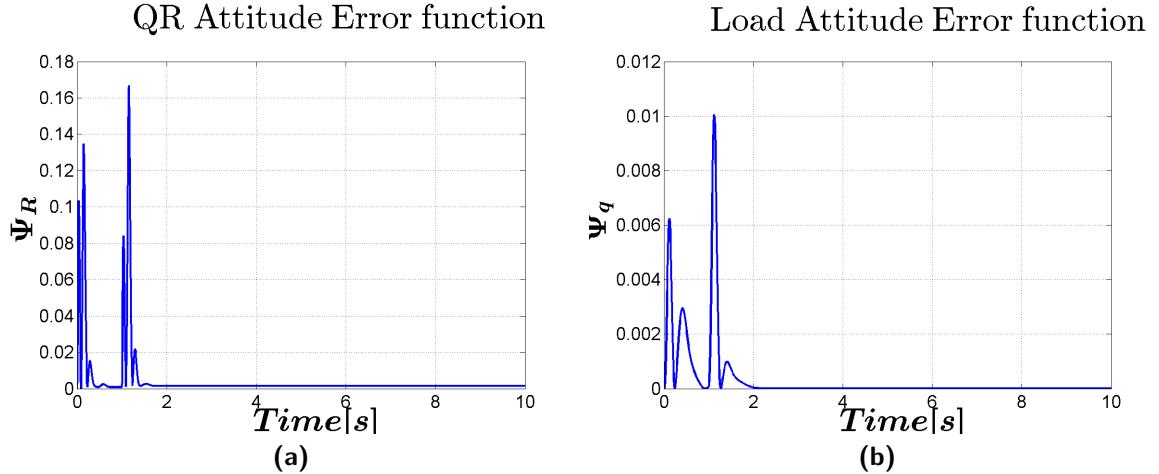
Figure 4-7: Load Position Tracking NGC Case A

show that the tracking error functions of the QR and load, respectively, are also driven to zero. From this can be concluded that $(e_x, e_v, e_q, e_{\dot{q}}, e_R, e_\Omega) = (0, 0, 0, 0, 0, 0)$ is exponentially stable.

For both control approaches, Figure 4-10a shows the load position and Figure 4-10b shows the corresponding load position error. From this can be observed that the LQR controller has a slower response, larger overshoot and higher settling time. Due to the aggressive maneuver by the NGC control, the system has a steady-state offset in height. It was found that this can be resolved by tuning the controller with smaller gains or decreasing the cut-off frequency of the low pass filters, which both result in smoother, but slower responses.

Figure 4-10c shows the absolute position error over the desired load position in percentages. It can be observed that the LQR controller requires more time to stabilize the load angle, whereas the NGC controller is capable of damping the oscillation in a smooth manner.

Figure 4-11a shows the QR attitude with respect to $\{\mathcal{I}\}$, and a huge difference in the QR rotations during the maneuver can be observed, note the difference in scale. It is obvious that the NGC controller is capable of handling more aggressive maneuvers. Figure 4-11b shows the

**Figure 4-8:** Geometric Error functions NGC Case A**Figure 4-9:** Geometric Tracking Error functions NGC Case A

load angle with respect to $\{\mathcal{I}\}$. The NGC controller allows larger angles of the load attitude, while stabilization is reached relatively fast. The LQR controller tries to minimize the swing along the entire trajectory, not allowing large angles of the load attitude.

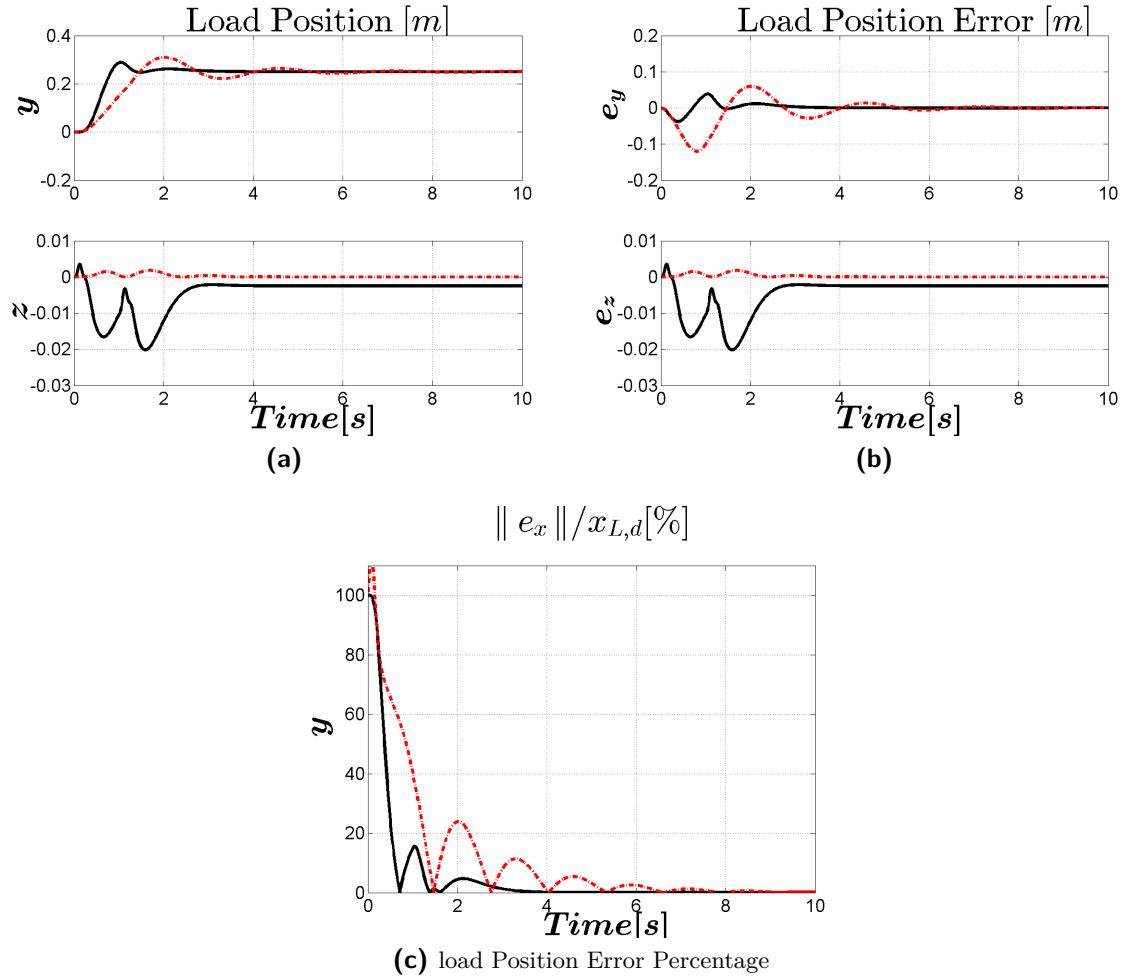


Figure 4-10: Controller Comparison Case A. Solid: NGC, Dash-dot: LQR

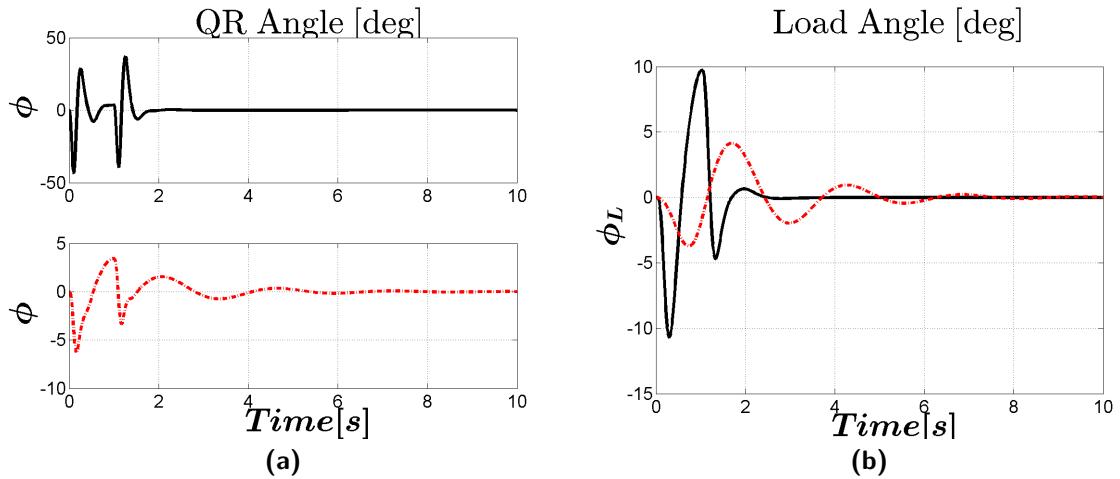


Figure 4-11: Controller Comparison Case A. Solid: NGC, Dash-dot: LQR

Case B

In this case, the desired trajectory was designed to investigate the response of the system to increasing distance and velocities between the ends of a swinging motion. The desired and actual load position, velocity and acceleration are shown in Figure 4-12a and 4-12b. The corresponding load position and velocity error are shown in Figure 4-12c. Despite the fact that case B requires higher velocities than in case A, the trajectory gradually increases the distance of the load position, which requires less sudden aggressive commanded rotations. It can be seen that acceleration in case B is a lot smoother than in case A, which explains why also both the velocity and position are smooth enough for the system to track.

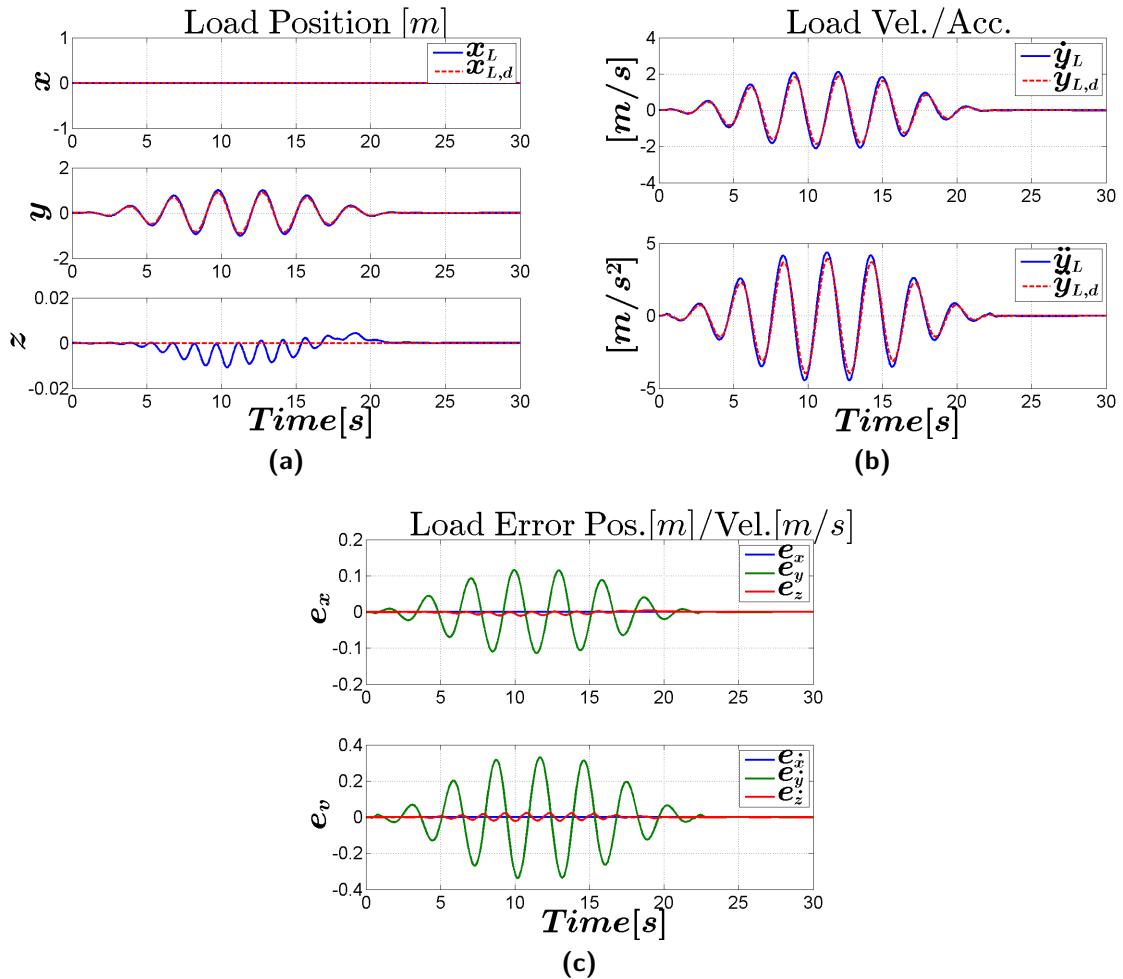


Figure 4-12: Load Position Tracking nonlinear geometric control Case B

Figure 4-13a and 4-13b show the tracking errors of the QR attitude and load attitude, respectively. The error in QR attitude and angular velocity is much smaller compared to the results in case A, which indicates that the system is fast enough to respond to the required QR rotations and velocities.

Both QR and load attitude errors stay bounded during the trajectory and are driven to zero.

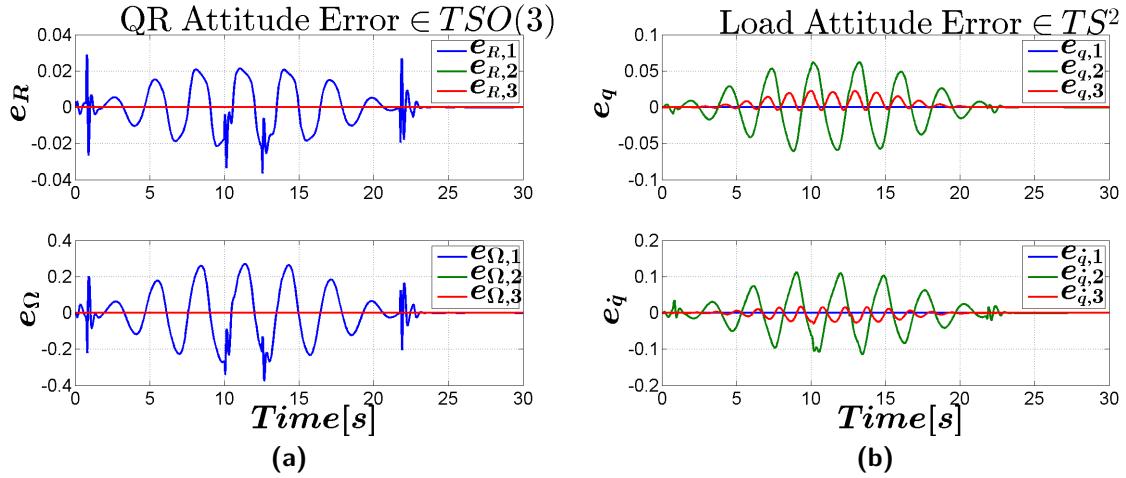


Figure 4-13: Geometric Error functions NGC Case B

Furthermore, Figure 4-14a and 4-14b show a gradual increase and decrease in error, similar to the profile of the desired velocity. All error functions are driven to zero, and it can also be concluded for this case that $(e_x, e_v, e_q, e_{\dot{q}}, e_R, e_{\Omega}) = (0, 0, 0, 0, 0, 0)$ is exponentially stable.

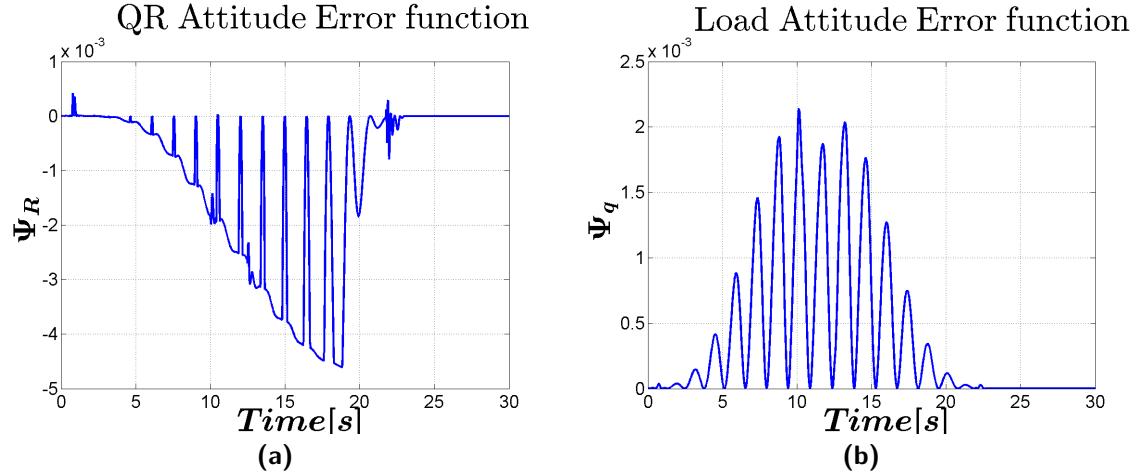


Figure 4-14: Geometric Tracking Error functions NGC Case B

Figure 4-15a shows the load position for both control approaches. The shape of the actual load position appears to be quite similar. However, taking a look at Figure 4-15b shows the load position error for both control approaches. It is clear that the LQR controller fails to track the load position, since errors can be seen to reach values up to $1.1m$.

The profiles of the QR rotation angles look notably alike, as can be seen in Figure 4-16a. It shows that the LQR controller is certainly capable of controlling the QR to large rotations. The gradually increasing velocity allows the LQR controller to reach larger angles. However, it can be seen in Figure 4-16b that the load swings significantly more when controlled by the LQR. The load continues its natural swing, which increases the error even more. This shows

that the LQR controller struggles with both keeping up with the position and minimizing the load swing.

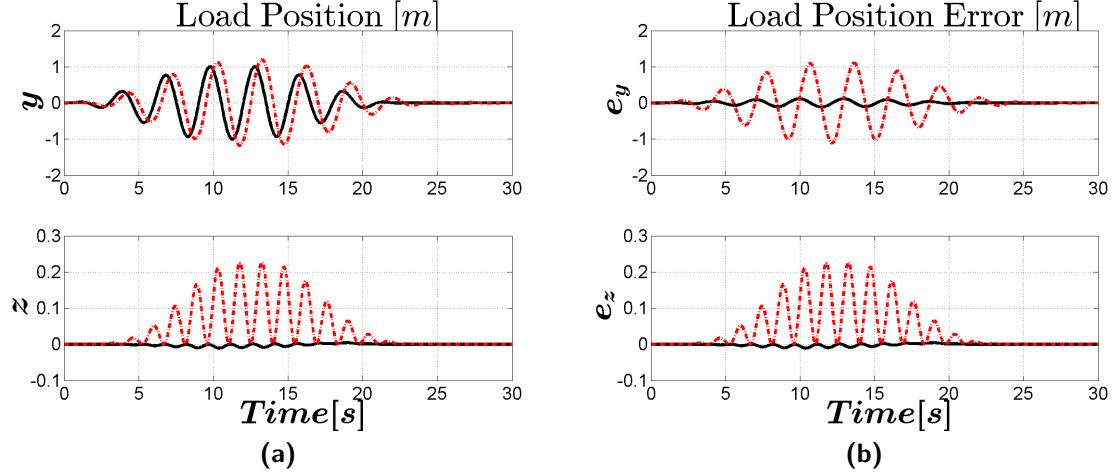


Figure 4-15: Controller Comparison Case B. Solid: NGC, Dash-dot: LQR

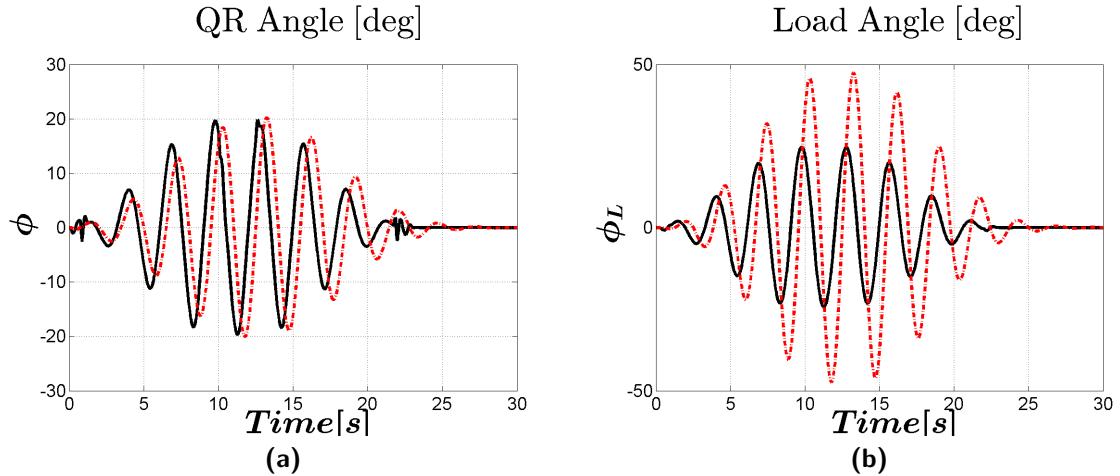


Figure 4-16: Controller Comparison Case B. Solid: NGC, Dash-dot: LQR

It is also notable that the shape of the load trajectory is nearly correct, this can be explained due to the regular shape of the trajectory and the natural swing of the load. It is easily illustrated that the controller has large load position errors due to the swing of the load, when the desired load trajectory becomes more complex.

An irregular shape for the desired load trajectory can be created by multiplying the desired load position from this case with a sine wave of an other frequency. The signal C in Figure 4-2a gets multiplied with signal D to obtain signal E, see Figure 4-17a. The corresponding desired position, velocity and acceleration is shown in Figure 4-17b.

Figure 4-18a shows the load position and it can be seen that the swinging motion of the actual trajectory is no longer similar to the desired trajectory. Figure 4-18b shows the corresponding

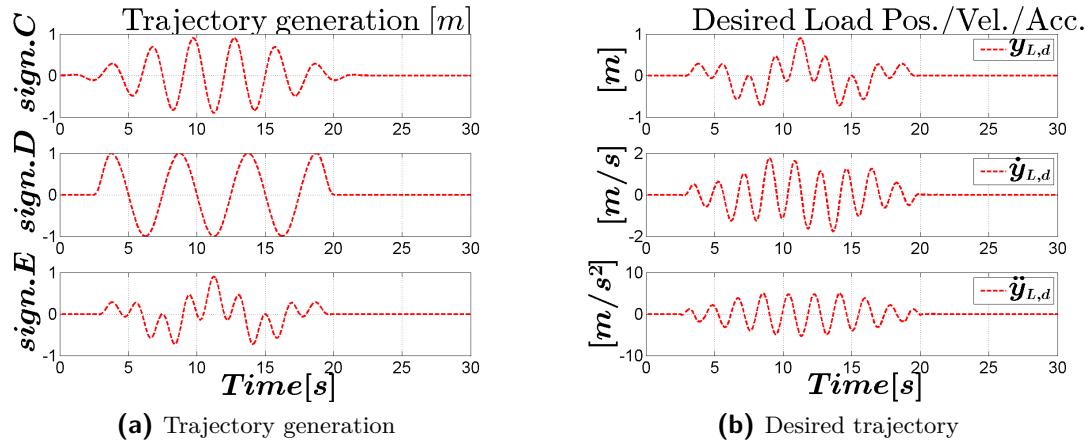


Figure 4-17: Desired load trajectory case B extended, y-direction

load position error. In Figure 4-18c can be seen that the LQR controller does not react to the large load angles, seen in Figure 4-18d. The NGC shows fast changes in the QR angles to correct for the load position errors.

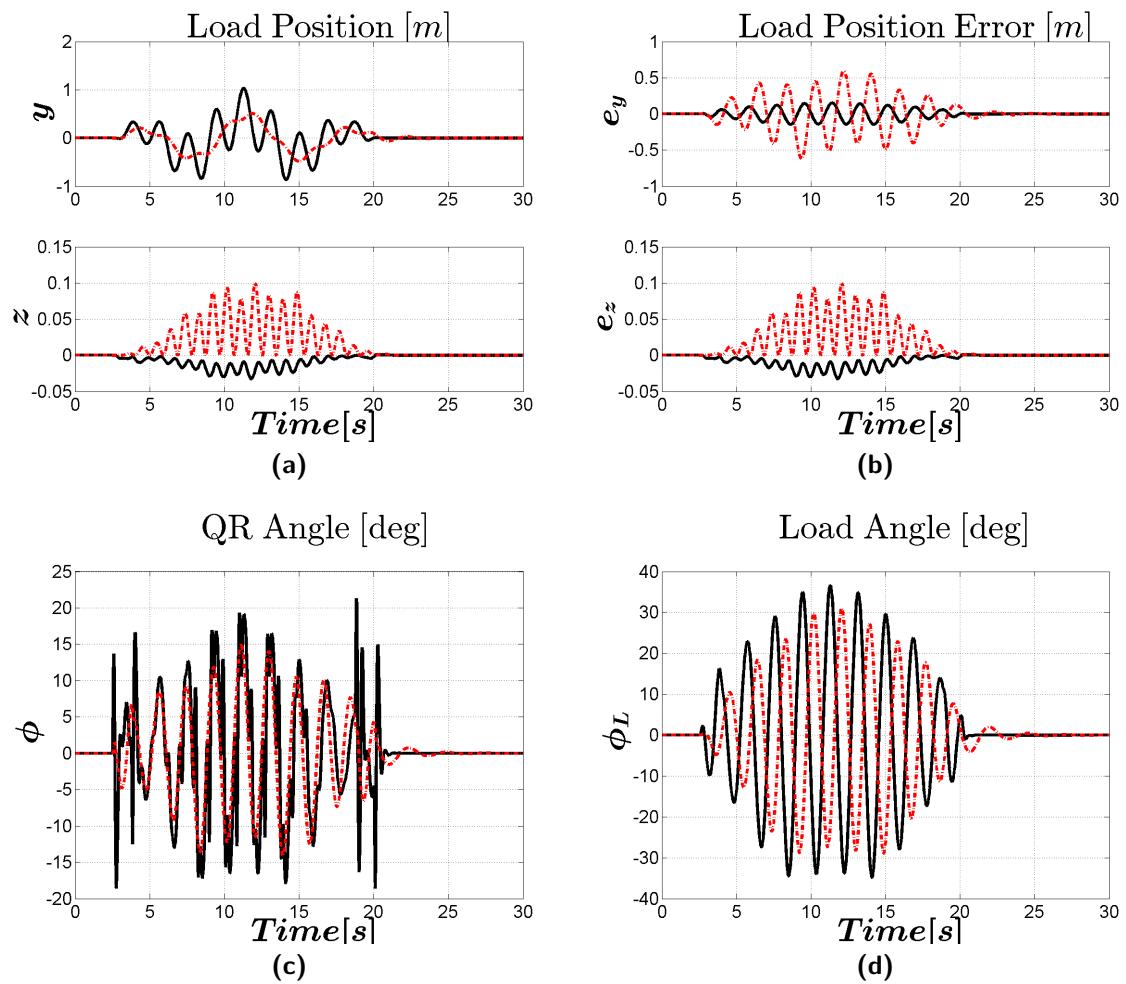


Figure 4-18: Controller Comparison Case B extended. Solid: NGC, Dash-dot: LQR

Case C

The last case is designed to investigate the response to a more complex desired load trajectory in three directions, each direction with a different shape. The desired and actual load position, velocity and acceleration are shown in Figure 4-19a and 4-19b.

As can be seen in Figure 4-19c, the error in height is quite large. This can be explained by the combination of height loss due to rotation and the changing desired height trajectory. At some point in time the load is required to fly a distance of $2m$ straight, side to side in the y -direction with approximately $2m/s$. At the same time when the QR is inclined, the trajectory in the z -direction requires the load to go up and down.

As expected, the QR loses height due to a decreasing total upward force whenever the QR is inclined. The error in the y -direction is most likely the result of the increasing required speed in both the x - and y -direction. The QR is capable of creating a moment around a body axis. However, when the QR is rotated around one body axis, a rotation around the other body axis will not result in a straight translation. This makes it challenging to move fast in one direction while swinging perpendicular to that direction.

Despite the errors, the QR is able to stabilize to an equilibrium, this can be seen in Figures 4-19c, 4-20a, 4-20b, 4-21a and 4-21b. It can be observed that $(e_x, e_v, e_q, e_{\dot{q}}, e_R, e_{\Omega}) = (0, 0, 0, 0, 0, 0)$ is exponentially stable.

Figure 4-22a shows the load position and Figure 4-22b shows the load position error for both control approaches. It is evident that the LQR controller is not suitable for load position tracking of such complex trajectories.

Figure 4-23a and 4-23b show the QR and load angle w.r.t. $\{\mathcal{I}\}$. The main difference that can be observed are the spikes in the NGC approach. The NGC QR attitude controller is able to calculate the required rotation along the trajectory, based on the error information. This indicates fast maneuvering, something what the LQR controller is not capable of. The load angle by NGC has a peculiar shape, for it is not a smooth sine-wave like shape. The pointy almost triangular shapes indicate that load angle is controlled actively, rather than letting it swing freely. This can be related to the peaks in the QR angle, see Figure 4-23a.

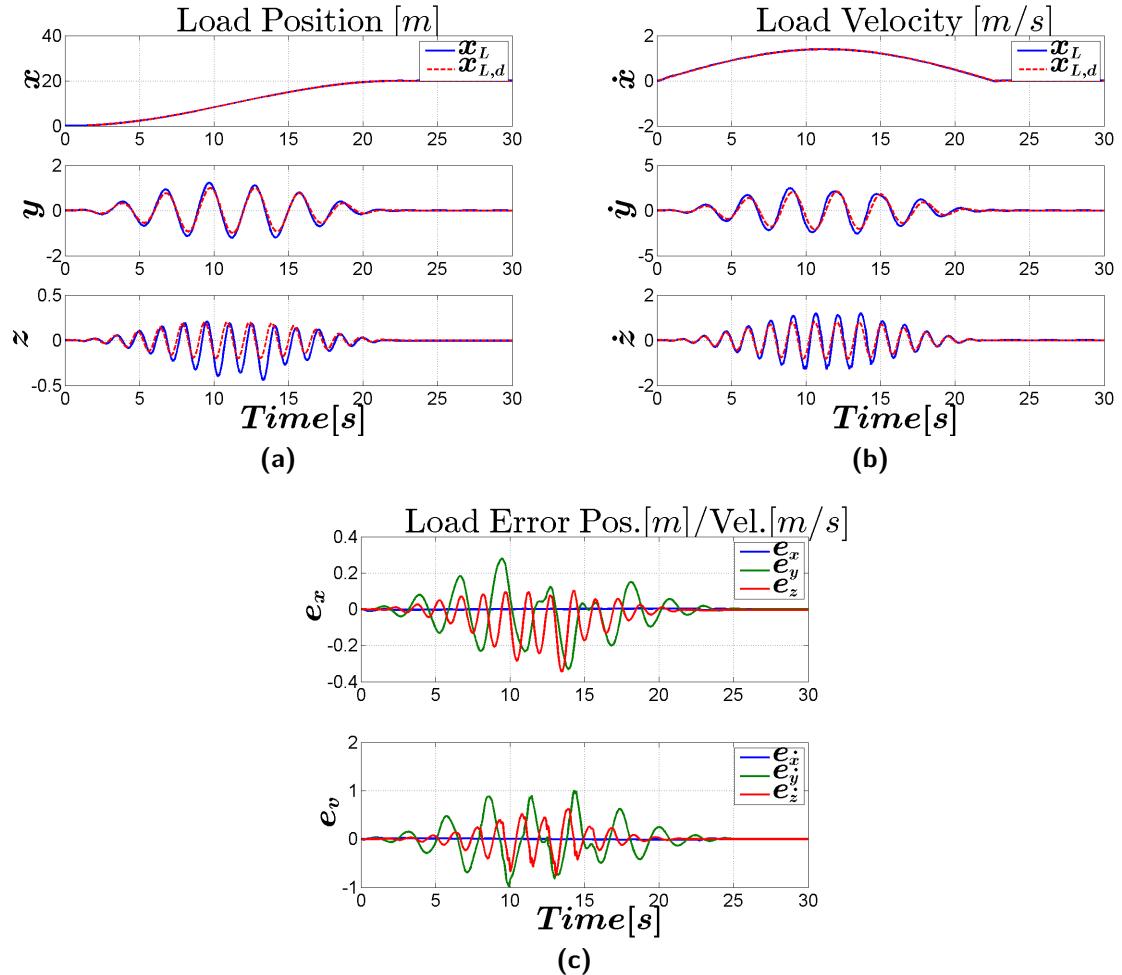


Figure 4-19: Load Position Tracking nonlinear geometric control Case C

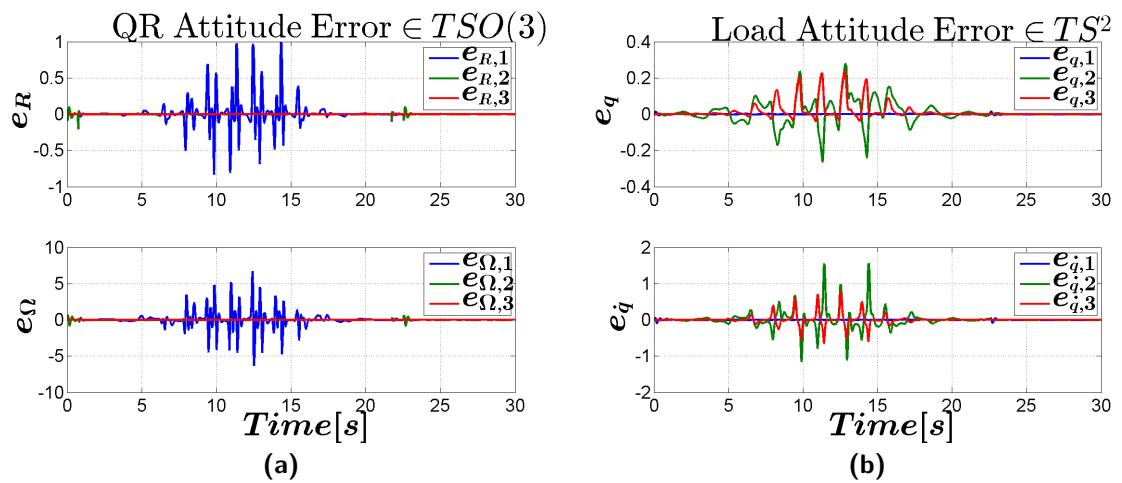


Figure 4-20: Geometric Error functions NGC Case C

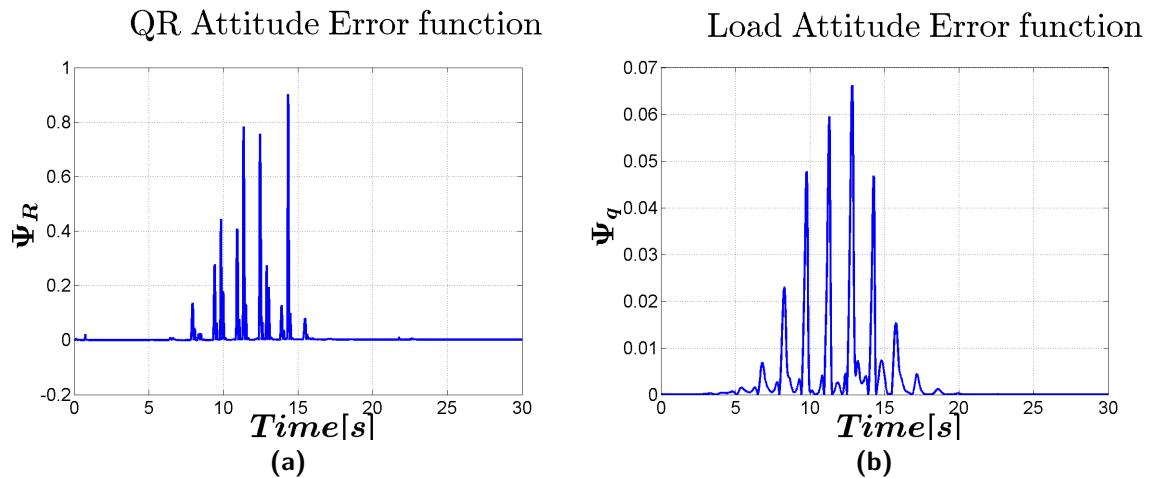


Figure 4-21: Geometric Tracking Error functions NGC Case C

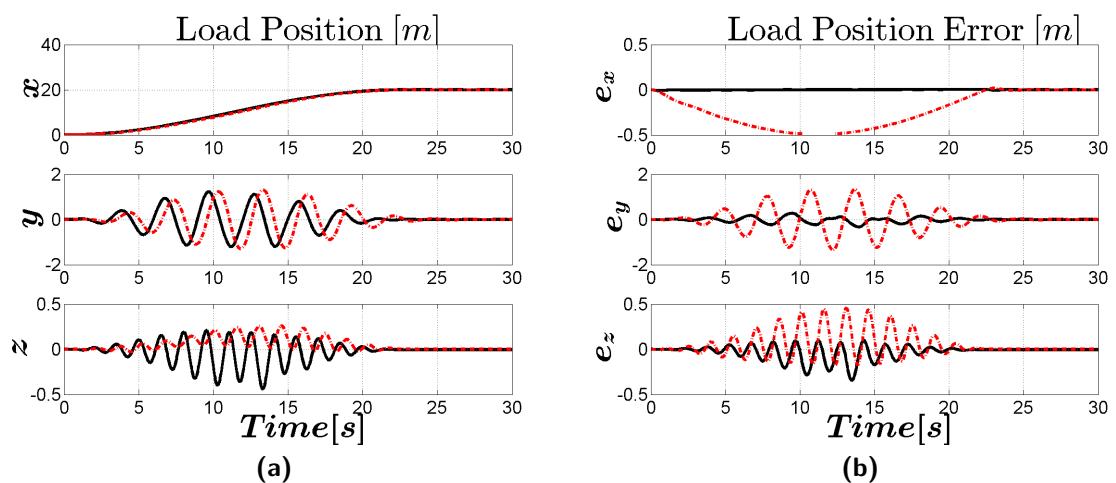


Figure 4-22: Controller Comparison Case C. Solid: NGC, Dash-dot: LQR

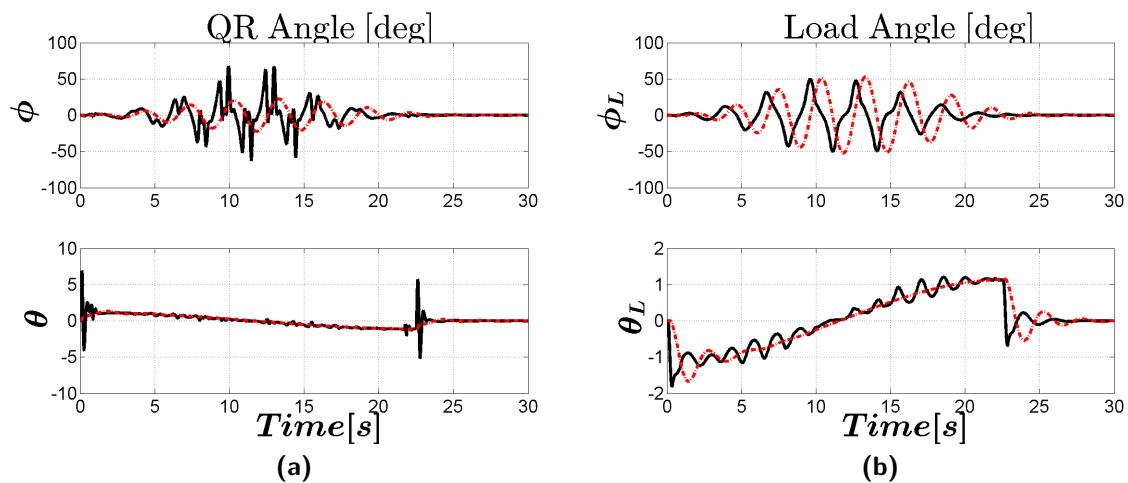


Figure 4-23: Controller Comparison Case C. Solid: NGC, Dash-dot: LQR

4-5 Summary

In this chapter, the experiments to test the nonlinear geometric controller on its ability to track load trajectories are described. It is explained how the stability of the system and the load position tracking performance can be analyzed.

The first trajectory is a smooth step-like function, which allows properties similar in the analysis of a step response to be observed. Next, the second trajectory describes an increasing load position trajectory to investigate the systems response on tracking of fast load velocities. Finally, the last trajectory describes a three-dimensional load position tracking, by generating a combination of trajectories to test the limits of the controller.

Matlab and Simulink were used to create simulations of the experiments, and data was generated to analyze the responses of the system. Along with the chosen parameters, the results of the experiments are presented. The experiments were done with both the nonlinear geometric controller and an LQR controller. Differences in the control performances were discussed and the concept of the nonlinear geometric controller is proven.

Chapter 5

Conclusions and Future Work

5-1 Summary and Conclusions

This thesis aims to give insight in a nonlinear geometric approach for the control of a quadrotor with a cable suspended load. In order to test the concept of the nonlinear geometric approach, a mathematical model is composed by applying techniques from differential geometry. Differential geometry is the mathematical discipline that is used to study these geometric problems. This method allows the model to be defined globally on nonlinear manifolds, which is not possible on Euclidean spaces. The configuration space of the system is described on nonlinear manifolds, instead of applying Euclidean spaces that are defined by Cartesian coordinates. With the tools of differential geometry, differential calculus and integral calculus, a globally defined model is obtained.

Nonlinear geometric control is used for the design of a control system for this dynamic system that evolves on nonlinear manifolds. The model allows nonlinear geometric controllers to define error functions to calculate error values that evolve on the same nonlinear manifolds, through matrix operations that arise from linear algebra. A backstepping approach allows different DOFs of the under-actuated system to be controlled. The controllers consists of control laws that guarantee stabilization of the closed-loop system in a cascaded structure by using the states as virtual control inputs.

The stability and performance of the nonlinear geometric controller is tested in a set of experiments, each describing different load trajectories. Experiments are simulated and the nonlinear control performance is compared to the proven concept of an LQR controller. The nonlinear tracking controller proves almost-global exponential attractiveness to the zero equilibrium of the tracking errors.

The nonlinear geometric approach allows the control of multiple states with the objective of controlling the load position. From the results of the experiments can be concluded that the nonlinear control design has proven effective in tracking complex trajectories and fast maneuvers, while maintaining closed-loop stability of the system.

Furthermore, the advantage that the controllers are almost-globally defined, makes them

suitable for complex load position tracking. The controller shows overall satisfying results for the load position tracking experiments.

Tuning of the controller can be done by checking the system responses and the error functions. Tuning the controllers for the QR and load attitude controlled mode can be done in a similar fashion as tuning a simple PD controller. Adding proportional gain results in a faster response, where adding derivative gain for a better damping. However, tuning the load position controlled mode is no longer intuitive due to the nonlinearities and coordinate changes by the control laws.

Where the nonlinear geometric controller employs a nonlinear model that is defined on nonlinear manifolds, the LQR controller employs a model that is defined and linearized in a Euclidean space. The nonlinear geometric approach allows the control of the dynamics that are described by the nonlinear model. The linear controller has an incomplete model due to unmodeled dynamics, and the controller does not allow direct tracking of a load trajectory. For that reason, the LQR controller is tuned by making a trade-off must be made between sufficient QR position tracking and minimization of the load swing.

From the comparison between the controllers it can be concluded that, near the equilibrium configuration, the LQR controller is able to reduce the swing relatively fast and accurate. Furthermore, where the nonlinear geometric controller fails to track a non-smooth trajectory, the LQR is capable of responding to a regular step function, keeping the system stable and reaching the final state. However, the shortcomings of the LQR controller become evident when it is given the task to track complex trajectories. The controller is not able to cope with the fast changing dynamics, resulting in overshoots and slow responses, causing the load to lag behind the desired trajectory. Keep in mind that the comparison with an LQR controller is meant as a proof of concept of NGC, and the goal is not to optimize the LQR controller.

5-2 Recommendations for Future Work

5-2-1 Investigate Implementation

Digital control The concept of geometric control is shown under the assumption of continuous-time control. However, an analysis must be done in the discrete-time domain for the implementation of a real-time application. The feasibility to run the controller on an on-board processor on a QR must be investigated. The control performance could be limited by the bandwidth of either the discretized control system or the wireless communication. It must be investigated whether the control system is still able to deal with the fast dynamics that are required for aggressive maneuvering. Continuous-time Euler-Lagrange equations were found by minimizing the action integral, which is a function of the Lagrangian. In a similar procedure the discrete-time Euler-Lagrange can be obtained, by minimizing the summation of a discrete Lagrangian, which is demonstrated in [10].

Furthermore, command filters were applied to approximate the derivatives of the command inputs. This was also done under the assumption of continuous-time control. The calculations can be achieved by implementation of simple low-pass filters, potentially saving computational power and interesting for an on-board implementation. However, it must be investigated whether this approach preserves the geometric properties of a mechanical system and controller stability in the discrete-time domain.

Model identification, validation and robustness In this thesis the model parameters are either obtained from examples in literature or arbitrarily chosen. In practice, identification and validation of the QR model and rotor dynamics is required.

The control in this thesis assumes perfect state feedback. In practice the controller depends on visual feedback or data obtained from an on-board inertial measurement unit. Unlike in simulations, this data will contain noise, uncertainties and possibly drift.

Based on a nonlinear geometric approach for a QR without load, [13] includes uncertainties in the translational dynamics and rotational dynamics to prove robustness. As a theoretical extension, this approach could be extended to a QR-load system to test the influence of model mismatches and robustness of the controller.

Hybrid System Control This thesis is focused on the subsystem where the tension in the cable is non-zero. The dynamics of the QR-load system will be better covered by adding the subsystem where the cable tension is zero. More specifically, the second subsystem simply considers a QR with a disconnected load in free fall. Both subsystems can be modeled via the same nonlinear geometric approach.

A possible extension is to investigate the possibilities to apply hybrid control, such that the controller is able to switch between control for each of the two subsystem depending on whether the cable tension is zero or non-zero. In [1, 16, 33] both subsystems are expressed in the form of one hybrid nonlinear geometric model and a trajectory generation method that accounts for the switching dynamics of the hybrid system is presented.

5-2-2 Trajectory Generation

Minimum Snap Trajectory Generation The trajectories described in Section 4-2 were arbitrarily generated by hand to test the performance of the controller in different situations. Whenever more complex or optimal trajectories are required, this approach is no longer efficient and can become too complex to solve by hand. A recommended extension to this thesis is the automatic generation of a trajectory. Trajectory generation algorithms exist that are able to generate a smooth desired position, velocity and acceleration by solving a QP optimization problem. This approach is presented by [34] and applied in [16, 33]. The QP even allows inclusion of constraints on inputs or trajectory in the optimization. Furthermore, it is proven that the system is *differential flat*, meaning that all states and inputs can be expressed in terms of only four states and their derivatives. This property is used to transform the high-dimensional optimization problem into a four-dimensional problem.

Appendix A

Appendix

A-1 LQR controller

The linearized model is written into a first order ODE of the form

$$\dot{\mathbf{x}} = A\mathbf{x} + Bu \quad (\text{A-1})$$

$$y = C\mathbf{x} + Du \quad (\text{A-2})$$

with the state- and input vectors as defined in Equation 4-6.

The model is linearized about the hovering flight mode. All translational and rotational velocities are zero during hover. The positional states and the yaw angle do not affect the dynamics, and are set equal to zero. A thrust input $u_1 = g(mQ + mL)$ is required to maintain hover, and all other control inputs are set equal to zero. The states and inputs in the equations of motion are substituted by an initial condition and a perturbation

$$\dot{\mathbf{x}} \rightarrow \dot{\mathbf{x}}_0 + \delta\dot{\mathbf{x}}, \quad \mathbf{x} \rightarrow \mathbf{x}_0 + \delta\mathbf{x}, \quad u \rightarrow u_0 + \delta u \quad (\text{A-3})$$

$$\begin{aligned} \mathbf{x}(0) &= \mathbf{0} \\ u(0) &= [g(m_Q + m_L) \quad 0 \quad 0 \quad 0]^T \end{aligned} \quad (\text{A-4})$$

The linearized equations of motion are rearranged in Equation A-5, as done in [8], and substituted back into Equation A-1.

$$M_1\delta\ddot{\mathbf{x}} + M_2\delta\mathbf{x} = M_3\delta u \quad (\text{A-5})$$

With $I_L = mL^2$ and $m_T = m_Q + mL$, the matrices $M1, M2, M3$ are defined as follows

$$M_1 = \begin{bmatrix} m_T & 0 & 0 & 0 & 0 & 0 & 0 & mL \\ 0 & m_T & 0 & 0 & 0 & 0 & Lm_L & 0 \\ 0 & 0 & m_T & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{xx} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{yy} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{zz} & 0 & 0 \\ 0 & mL & 0 & 0 & 0 & 0 & I_L + L^2m_L & 0 \\ mL & 0 & 0 & 0 & 0 & 0 & 0 & I_L + L^2m_L \end{bmatrix} \quad (\text{A-6})$$

$$M_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & -gm_T & 0 & 0 & 0 \\ 0 & 0 & 0 & gm_T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & gLm_L & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & gLm_L \end{bmatrix} \quad (\text{A-7})$$

$$M_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (\text{A-8})$$

The tuning parameters of the LQR controller are chosen as follows

```
1 Qdiag = [50000 50000 50000 0.001 0.001 1 25000 25000 0.001*ones(1,8)];
```

```
1 Rdiag = [1000 100*ones(1,3)];
```

Matlab command `lqr(LQRA,LQRB,Q,R)` generates the following gain matrix K

```
1 K =
2
3 Columns 1 through 7
4
5 -0.0000    0.0000    7.0711   -0.0000   -0.0000   -0.0000    0.0000
6 -0.0000  -22.3607    0.0000    2.3683   -0.0000   -0.0000    8.3778
7 22.3607    0.0000   -0.0000   -0.0000    4.0067    0.0000   -0.0000
8 0.0000   -0.0000   -0.0000    0.0000    0.0000    0.1000   -0.0000
9
10 Columns 8 through 14
11
12 0.0000   -0.0000   -0.0000    2.6591    0.0000    0.0000   -0.0000
13 0.0000   -0.0000   -6.9992   -0.0000    0.2112    0.0000    0.0000
14 -9.6733    7.8971    0.0000   -0.0000   -0.0000    0.0272    0.0000
15 -0.0000    0.0000   -0.0000   -0.0000    0.0000   -0.0000    0.0311
16
17 Columns 15 through 16
18
19 -0.0000   -0.0000
20 -5.0244   -0.0000
21 0.0000    4.6145
22 -0.0000    0.0000
```

A-2 Additional Figures

Figures A-1, A-2, A-3 and A-4 show the control inputs corresponding to the experiments in cases A, B, B extended and C, respectively. Figures A-5, A-6, A-7 and A-8 show the QR

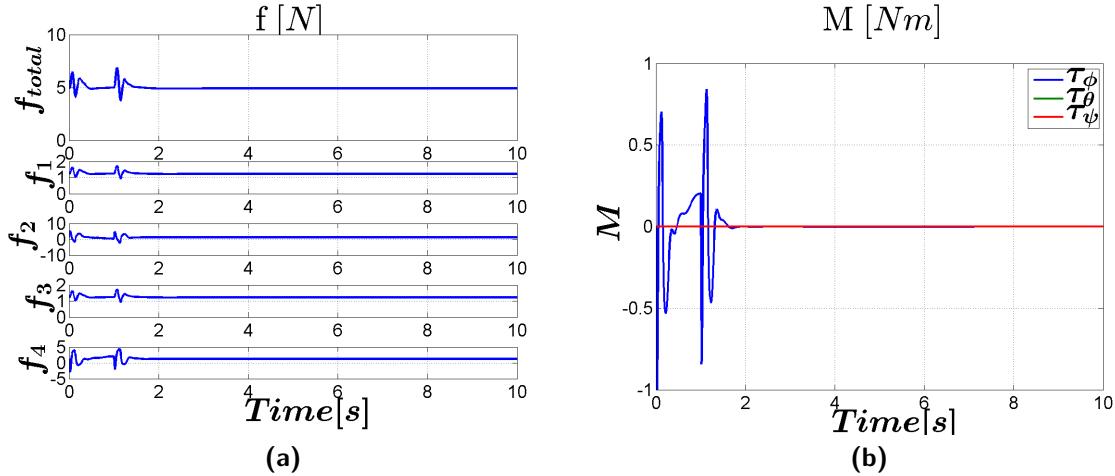


Figure A-1: Control InputsNGC Case A

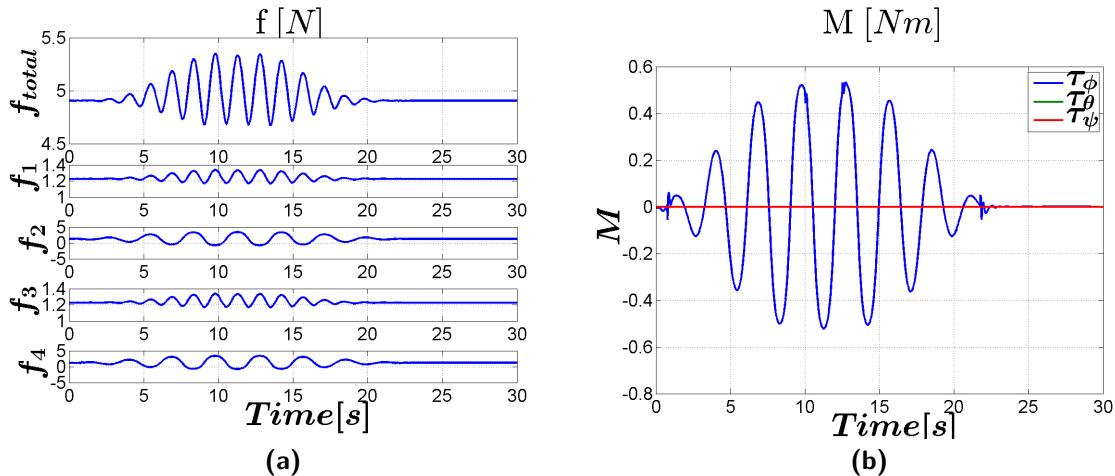


Figure A-2: Control InputsNGC Case B

position and load angle tracking error corresponding to the experiments done with the LQR controller in cases A, B, B extended and C, respectively.

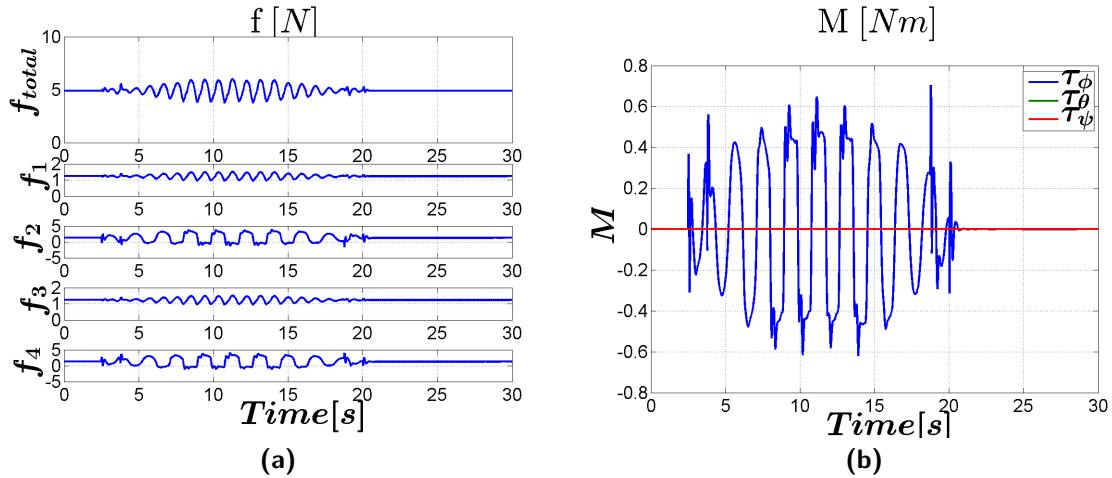


Figure A-3: Control InputsNGC Case B extended

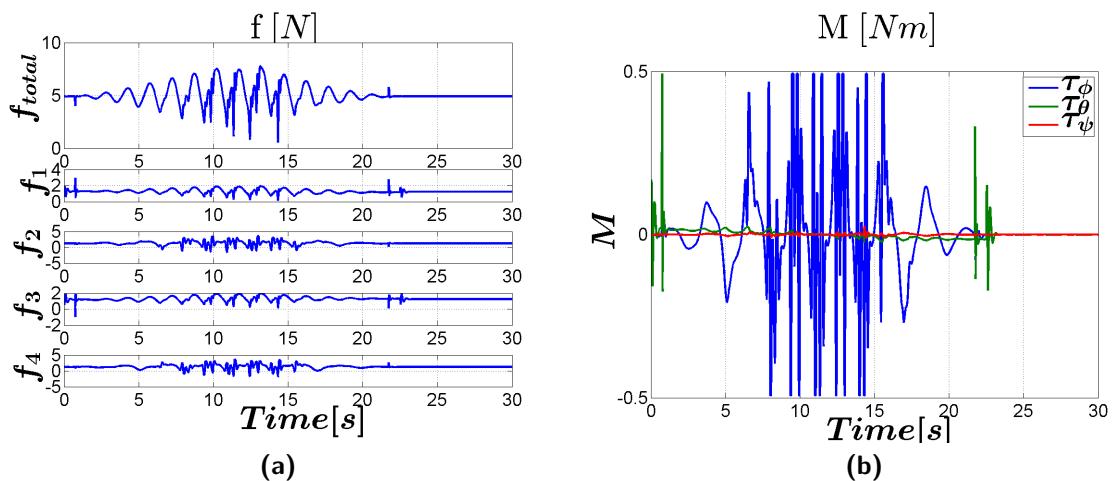


Figure A-4: Control InputsNGC Case C

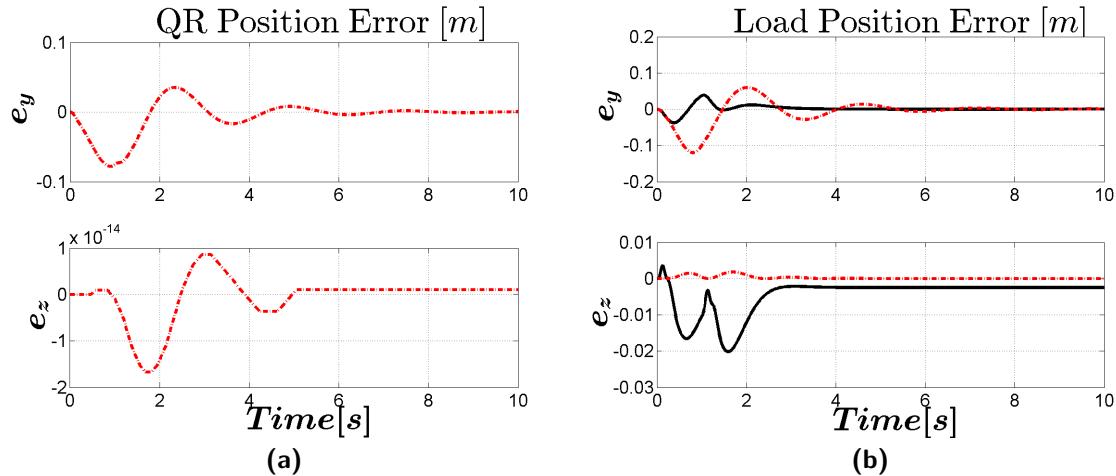


Figure A-5: QR Position Error and Load Angle Error LQR Case A. Dash-dot: LQR

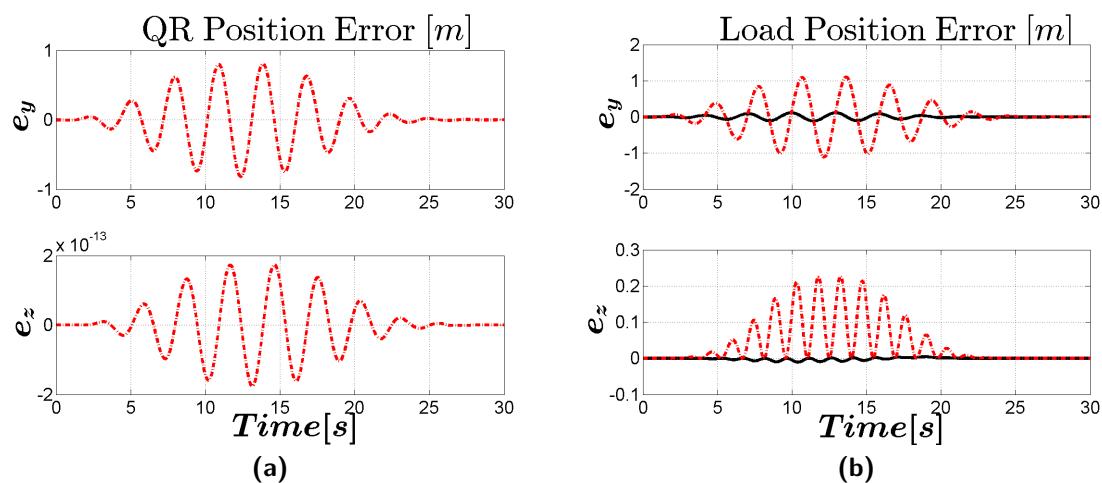


Figure A-6: QR Position Error and Load Angle Error LQR Case B. Dash-dot: LQR

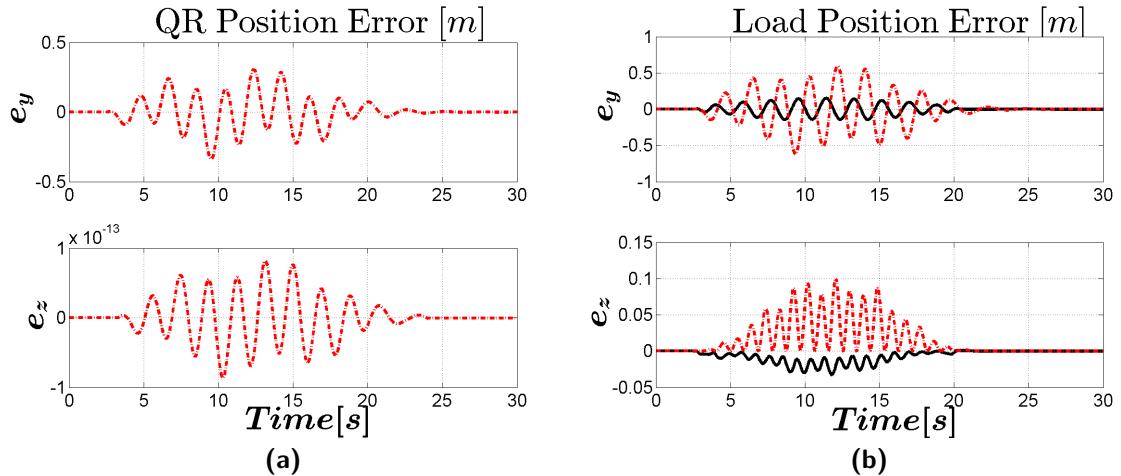


Figure A-7: QR Position Error and Load Angle Error LQR Case B extended. Dash-dot: LQR

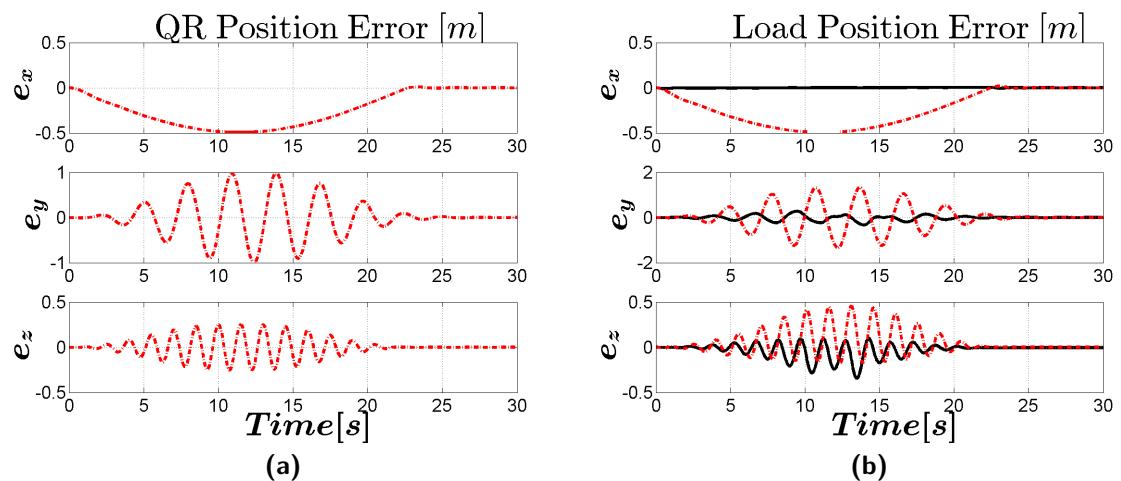


Figure A-8: QR Position Error and Load Angle Error LQR Case C. Dash-dot: LQR

Bibliography

- [1] K. Sreenath, Taeyoung Lee, and V. Kumar, “Geometric control and differential flatness of a quadrotor UAV with a cable-suspended load,” in *52nd IEEE Conference on Decision and Control*, pp. 2269–2274, IEEE, dec 2013.
- [2] S. Bouabdallah, A. Noth, R. Siegwart, and R. Siegwart, “PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor,” *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, pp. 2451–2456, 2004.
- [3] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, “The GRASP multiple micro-UAV testbed,” *IEEE Robotics and Automation Magazine*, vol. 17, no. 3, pp. 56–65, 2010.
- [4] M. Bangura and R. Mahony, “Real-time model predictive control for quadrotors,” in *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 19, pp. 11773–11780, 2014.
- [5] E. Reyes-Valeria, R. Enriquez-Caldera, S. Camacho-Lara, and J. Guichard, “LQR control for a quadrotor using unit quaternions: Modeling and simulation,” *International Conference on Electronics, Communications and Computing. CONIELECOMP 2013*, pp. 172–178, 2013.
- [6] R. P. K. Jain and T. Keviczky, “MSc Thesis: Transportation of Cable Suspended Load using Unmanned Aerial Vehicles: A Real-time Model Predictive Control approach,” 2015.
- [7] F. Luis, M. Zürn, and F. Gonzalez, “MPC controlled Multirotor with suspended slung Load : System Architecture and Visual Load Detection,” 2016.
- [8] J. D. A. Becker, *Modeling and Control of a Quadrotor with Dynamic Inertia*. PhD thesis, 2013.
- [9] S. Sadr, S. Ali, A. Moosavian, and P. Zarafshan, “Dynamics Modeling and Control of a Quadrotor with Swing Load,” *Journal of Robotics, Hindawi Publishing Corporation*, vol. 2014.
- [10] T. Lee, *Computational Geometric Mechanics and Control of Rigid Bodies*. PhD thesis, The University of Michigan, 2008.

- [11] F. Bullo and A. D. Lewis, *Geometric control of mechanical systems: modeling, analysis, and design for simple mechanical control systems*. Springer, 2005.
- [12] T. Lee, M. Leok, and N. H. McClamroch, “Geometric Tracking Control of a Quadrotor UAV on $\text{SE}(3)$,” *49th IEEE Conference on Decision and Control*, no. 3, pp. 5420–5425, 2010.
- [13] F. Goodarzi, D. Lee, and T. Lee, “Geometric nonlinear PID control of a quadrotor UAV on $\text{SE}(3)$,” *Control Conference (ECC), 2013 European*, pp. 3845–3850, 2013.
- [14] 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 *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4888–4895, 2013.
- [15] K. Sreenath, T. Lee, and V. Kumar, “Geometric control and differential flatness of a quadrotor UAV with a cable-suspended load,” in *Proceedings of the IEEE Conference on Decision and Control*, pp. 2269–2274, 2013.
- [16] S. Tang, “Aggressive Maneuvering of a Quadrotor with a Cable-Suspended Payload,” tech. rep., University of Pennsylvania Philadelphia, Pennsylvania, 2014.
- [17] W. M. Boothby, *An Introduction to Differentiable Manifolds and Riemannian Geometry*. 2003.
- [18] N. Chaturvedi, “Rigid-Body Attitude Control,” *IEEE Control Systems*, vol. 31, no. 3, pp. 30–51, 2011.
- [19] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, vol. 29. 1994.
- [20] T. L. T. Lee, N. McClamroch, and M. Leok, “A lie group variational integrator for the attitude dynamics of a rigid body with applications to the 3D pendulum,” *Proceedings of 2005 IEEE Conference on Control Applications, 2005. CCA 2005.*, pp. 962–967, 2005.
- [21] T. Lee, M. Leok, and N. H. McClamroch, “Lagrangian mechanics and variational integrators on two-spheres,” *International Journal for Numerical Methods in Engineering*, vol. 79, no. 9, pp. 1147–1174, 2009.
- [22] T. Lee, M. Leok, and N. H. McClamroch, “Stable Manifolds of Saddle Points for Pendulum Dynamics on S^2 and $\text{SO}(3)$,” p. 9, 2011.
- [23] T. Lee, M. Leok, and N. H. McClamroch, “Lie group variational integrators for the full body problem in orbital mechanics,” *Celestial Mechanics and Dynamical Astronomy*, vol. 98, no. 2, pp. 121–144, 2007.
- [24] J. H. Gillula, H. Huang, M. P. Vitus, and C. J. Tomlin, “Design of guaranteed safe maneuvers using reachable sets: Autonomous quadrotor aerobatics in theory and practice,” in *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1649–1654, 2010.
- [25] N. A. Chaturvedi, T. Lee, M. Leok, and N. H. McClamroch, “Nonlinear dynamics of the 3D pendulum,” *Journal of Nonlinear Science*, vol. 21, no. 1, pp. 3–32, 2011.

-
- [26] T. Lee, M. Leok, and N. McClamroch, “Control of complex maneuvers for a quadrotor UAV using geometric methods on SE (3),” *arXiv*, 2010.
 - [27] R. Mahony, V. Kumar, and P. Corke, “Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor,” *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 20–32, 2012.
 - [28] I. Kanellakopoulos, P. V. Kokotovic, and A. S. Morse, “Systematic Design of Adaptive Controllers for Feedback Linearizable Systems,” *IEEE Transactions on Automatic Control*, vol. 36, no. 11, pp. 1241–1253, 1991.
 - [29] U. A. V. Cornelis and M. Vlaar, *Incremental Nonlinear Dynamic Inversion flight control*. PhD thesis, Tecnico Lisboa, 2014.
 - [30] J. A. Farrell, M. Polycarpou, M. Sharma, and W. Dong, “Command Filtered Backstepping,” pp. 1923–1928, 2008.
 - [31] J. Farrell, M. Polycarpou, and M. Sharma, “Backstepping-Based Flight Control with Adaptive Function Approximation,” *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 6, pp. 1089–1102, 2005.
 - [32] B. M. Borra, “Nonlinear UAV Flight Control Using Command Filtered Backstepping,” *Simulation*, no. March, pp. 1–161, 2012.
 - [33] S. Tang and V. Kumar, “Mixed Integer Quadratic Program Trajectory Generation for a Quadrotor with a Cable-Suspended Payload,” *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2216–2222, 2015.
 - [34] D. Mellinger and V. Kumar, “Minimum snap trajectory generation and control for quadrotors,” in *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2520–2525, 2011.

List of Symbols

ϵ	Tuning parameter to enable rapid exponential convergence of e_R, e_Ω
$\lambda_M(\cdot)$	Maximum eigenvalue
ω	Angular velocity of the load
$\Omega \in \mathbb{R}^3$	Body angular velocity
ω_i	Angular speed of rotor i
ϕ_L	Angle of rotation about e_1 w.r.t. $\{\mathcal{I}\}$
$\{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$	Unit vectors along the axes of $\{\mathcal{B}\}$
$\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$	Unit vectors along the axes of $\{\mathcal{I}\}$
$\{\mathcal{B}\}$	Body Frame
$\{\mathcal{I}\}$	Inertial World Frame
b	Thrust factor
d	Drag factor
f	Total thrust in direction of \mathbf{b}_3 , expressed in $\{\mathcal{B}\}$. $f = \sum_{i=1}^4 F_i$
F_i	Force generated by rotor i
g	Gravitation constant $9.81 m/s^2$
$J \in \mathbb{R}^{3 \times 3}$	Inertia tensor of QR
L	Length of the cable
l	Distance from the rotor to the QR CM
M	Total moment around axes of $\{\mathcal{B}\}$, expressed in $\{\mathcal{B}\}$. $M = [M_\phi \ M_\theta \ M_\psi]^T$
m_L	Mass of the load
m_Q	Mass of the QR
M_i	Drag moment generated by each rotor
$q \in \mathbb{S}^2$	Unit vector from QR to load
$x_L \in \mathbb{R}^3$	Position of the load
$x_Q \in \mathbb{R}^3$	Position of the QR CM
x_{CM}	Position CM of QR-load system
θ_L	Angle of rotation about e_2 w.r.t. $\{\mathcal{I}\}$

Acronyms

QR	Quadrotor
CM	Center of Mass
DOF	Degree of Freedom
PD	Proportional-Dervative (controller)
PID	Proportional-Integral-Derivative (controller)
LQR	Linear Quadratic Regulator
MPC	Model Predictive control
NGC	nonlinear geometric control
QP	Quadratic Programming