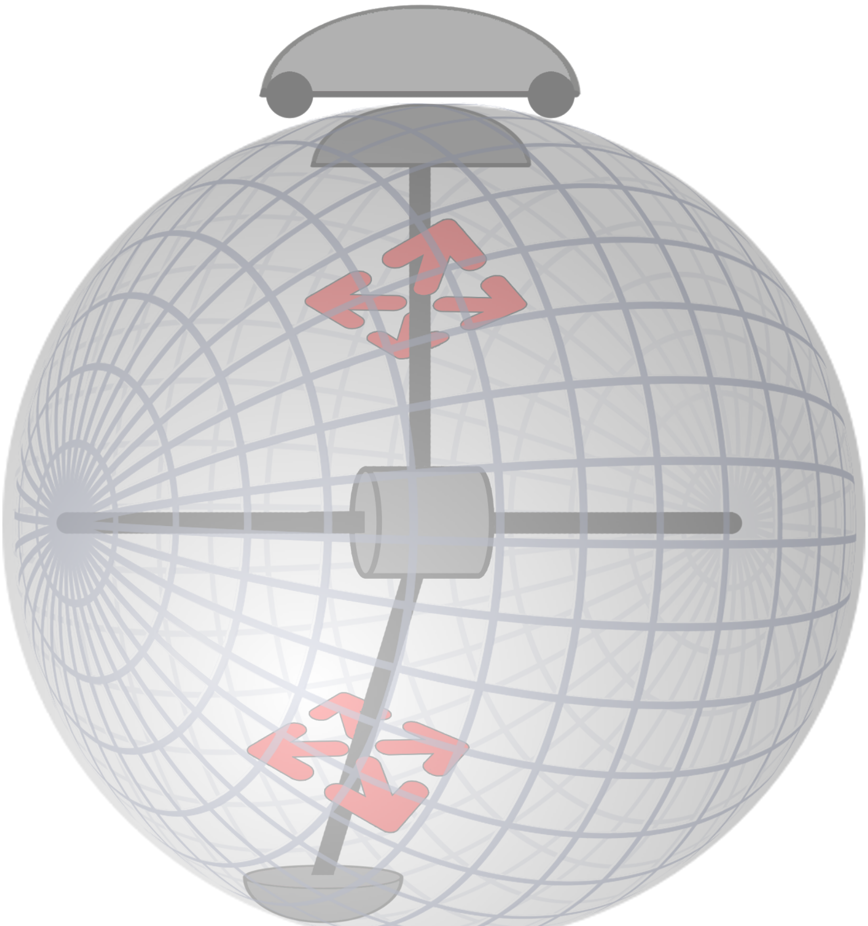
CHALMERS
UNIVERSITY OF TECHNOLOGY



Modeling and Controller Design for Spherical Robots

Robust and Nonlinear Approaches

Master's thesis in *MPSYS, Systems, Control and Mechatronics.*

CARL ANDERSSON

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019

Master's thesis 2019

Modeling and Controller Design for Spherical Robots

Robust and Nonlinear Approaches

CARL ANDERSSON

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Systems and Control
Chalmers University of Technology
Gothenburg, Sweden 2019

Modeling and Controller Design for Spherical Robots
Robust and nonlinear approaches
Carl Andersson

Supervisor: *Balázs Adam Kulcsár, Electrical engineering*
Examiner: *Balázs Adam Kulcsár, Electrical engineering*

Master's Thesis 2019
Department of Electrical Engineering
Division of Systems and Control
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2019

Modeling and Controller Design for Spherical Robots
Robust and Nonlinear Approaches
Carl Andersson
Department of Electrical Engineering
Chalmers University of Technology

Abstract

This master thesis explores and evaluates different linear, and nonlinear algorithms to control an unstable non-minimum phase nonlinear system, such as a spherical robot platform. The robot is described by means of rigid body modeling concepts using Lagrange's equations. The model is a double-pendulum driven spherical robot, with a regular and an inverted pendulum one. An algorithm is designed to control the robot's motion and balance an object at the same time. Simulations show how well each of the control algorithms (state and output feedback, dynamic output feedback $\mathcal{H}_2$, dynamic output feedback $\mathcal{H}_\infty$, real μ synthesis, and nonlinear feedback linearization) stabilizes the unstable nonlinear system. The report compares the algorithms in terms of potential strengths and weaknesses in a complex nonlinear simulation environment. Robustness, computational complexity and closed loop performance are compared.

Keywords: robust, linear, nonlinear, control, modeling, Lagrange, spherical, ball, robot, pendulum.

Acknowledgements

I would like to thank my supervisor, and examiner, *Balázs Adam Kulcsár, Chalmers University of Technology.* For the opportunity to make this master thesis, and for all the support, advise, and consultations throughout the making of this master thesis, which has greatly contributed towards the making of this master thesis.

I would also like to thank *Peter Folkow, Chalmers University of Technology*, for all the support, and advise of the making of the nonlinear model's for this master thesis.

I would also like to thank *Gabor Szederkenyi* for the support of implementing the nonlinear controller of *linear output selection for feedback linearization.*

I would also like to thank the opposition during the presentation of this master thesis, *Carl Hjerpe*, *Konstantinos Ektor Karyotakis*, *Christos Marinos*, and *Wendy Mo.* For all the feedback and questions, which helped in making the text more accurate, and improved the thesis.

Carl Andersson, Gothenburg, 2019

Contents

List of Figures

List of Tables

1
Introduction

This master thesis is to explore and evaluate different control algorithms, both linear and nonlinear, to control a nonlinear system. This project will explore the strengths and weaknesses of different control algorithms, and compare them to each other from different aspects. The nonlinear system model to be used is a spherical pendulum driven robot which is balancing a non-rigid mass on top of the sphere. The goal for the control algorithms is to make sure that when the spherical robot is moving, the mass on top stays, and tries to minimize the fluctuation. That is to make sure that the mass will be kept upright at all time, even when under changing inclination.

1.1 Background

In control theory both for linear and nonlinear cases, there exist several different control algorithms with each having there's strengths and weaknesses. To observe the different strengths and weaknesses it would be of interest to evaluate how the algorithms perform on a nonlinear system. This would give a deeper understanding on the advantages and disadvantages of using the different algorithm. Which would give a perspective of how they perform compared to each other, on a relative realistic system. Therefore, a comparative evaluation of different control algorithms, linear and nonlinear, on a system which is nonlinear. Would explore different aspects of control, performance, and robustness of the algorithms, that clarify different aspects on their respective strengths and weaknesses.

1.2 Objective

The objective of this master thesis is to explore and evaluate different control algorithms, strengths and weaknesses, of controlling a nonlinear system. Comparing the algorithms with each other to find advantages and disadvantages, and observe how well a low complexity control algorithm works compared against a highly complex control algorithm in different aspects. Some aspects to be considered is the closed loop performance, computational complexity, measurement noise effect, and robustness.

1.3 Limitations

The project will be limited by the fact that it focuses on theory. With one person to work on the project will be limitations with time. Therefore, this project will not consist of practical test the application in real life, but will consist of using simulation for testing and evaluation instead.

1.4 Research Questions

Some main research questions considered will be to explore and evaluate the different control algorithms, both for linear and nonlinear cases. The following research questions are considered.

- Do the linear algorithms provide stable controllers to control the nonlinear system, by that they use a linearized system of the nonlinear system?
- Are the linear algorithms enough to minimize wobbling and keeping the load mass balanced?
- Between the linear algorithms, what was the respective advantages, and disadvantages compared to each other?
- Between using linear and nonlinear algorithms, what was the respective advantages, and disadvantages compared to each other?
- Disadvantages and benefits of the linear and nonlinear algorithms? Such as numerical, dimensional, methodical etc.
- Is the simulation a reasonable representation of the system in a practical application?

2
Theory

In this chapter the theoretical parts are described. Theoretical background about rigid-body modeling of a nonlinear system, and linearization, both to a state-space form, are given. Some analysis tools and properties for linear and nonlinear systems will be described, and how to both solve and implement the different control algorithms that will be used. To start some other related works will be discussed, to widen the scope and observe what others have done.

2.1 Related Work

There have been many different projects, and inventions surrounding the use of ball- or spherical robots, and designs. Many are designed to be used in a widely spread area of applications, such as toys, transportation, security, planetary exploration, with more applications potentials being discussed. Some inventions that has been patented already as early 1900 and late 1800, such as a pendulum driven mechanical toys [1] [2], a spherical vehicle [3], and a marine vehicle [4]. [5] The ball or spherical body are of interest because of the properties that emerges of its use, such as its environmental capability, and also other properties that may benefit the design by having a spherical shaped construction. Even though there have been many articles, projects, patents, and constructions that use the pendulum drive principle with different ways of approach, designs, and methods. Not many exists (at this time) about the combination of a pendulum, and inverted pendulum, to observe how these two classical example systems is behaving, if merged together.

2.1.1 Spherical Design

Spherical designs have been used in different ways, and for different applications, such as told before toys, transportation, security, planetary exploration. Why the use of this ball or spherical shape is of that interest, may be because of its environment benefits. That potential benefit may be discussed, however, as mentioned before the use of this design choice had already schematics patented as early as the late 1800. The older patented designs are both designs of transportation, and a design that traverse water, these potential areas of use on land, obstacle handling in rough terrain, and in water could be proven to be benefiting.

Old Inventions

Some brief view of the early patents of spherical designs that has been mentioned, designs of toys [1], [2], a spherical vehicle [3], and a marine vehicle [4]. The two toys are using a pendulum to drive, the earliest of these where made 1893 by the inventor *J.L.Tate*, [1]. This invention is steered by a pendulum mechanic, and is interesting because that even though it is a toy from late 1800, the mechanics used in this invention is still a topic of research. The other spherical toy is a bit different from the toy by the inventor *J.L.Tate*, [1] in late 1800, is that this one has a figure on top of it, which is drives around with. This mechanical toy was made in 1909 by the inventor *E.E. Cecil*, [2]. This mechanical toy is based on similar mechanics, however, to accomplish this design some mechanical design have been made to accomplish that task. Some other notable spherical designs of old patents, was more focused on transportation. The spherical vehicle by the inventor *J.E. Reilley* in 1941,[3], and the marine vehicle by the inventor *H. William* in 1889, [4]. The spherical vehicle was designed for personal transportation, this design is interesting to observe, because if one considers the application of the spherical design offers. That the spherical form may be used as a vehicle design, potentially offers some beneficial properties for the use of the shape. One of these potentially beneficial properties may be observed from the marine vehicle by the inventor *H. William* in 1889, [4], where this mas made to traverse water.

Applications

The spherical design has potential of a wide range of applications, where one may argue that it can be used in different ways depending on what one wants to accomplish. Some inventions, and ideas that are using the spherical shape are constructed for different purposes, such as surveillance of different kinds, planetary exploration, robotics, and some potential ideas of newer transportation devises, and for underwater purposes. With the applications of use of the spherical shape, being investigated, and studied.

2.1.2 Other Works

Different studies have been made, such as articles, master thesis, and candidate works. Surrounding different aspects of pendulum driven spherical robots, and similar constructions. Everything from the spherical design itself, and different structures of the pendulum drive system, to controlling the pendulum driven spherical robot as trajectory, or just stabilize.

Summary of the Other Works

The different projects are focusing on different aspects of spherical design, some focuses in control, steering, and tracking. While others are more directed towards the design aspects of the model, or construction. Different ways, and methods to make a spherical robot seems to indicate that there is no direct standard way of approach. More directed to what is commonly used and is in different ways discussed to how one would design the mechanics to make a spherical robot. Even the internal mechanics are investigated and the interaction of the sphere to the external world.

Some of the design choices towards the pendulum drive system may involve multiple pendulums designs. Such as a four pendulum omnidirectional spherical robot [6], or "novel hybrid quadruped spherical mobile robot"[7], "Novel Spherical Robot with Hybrid Pendulum Driving Mechanism"[8], a spherical mobile robot driven by two perpendicular rotors [9], and other focuses on the displacement of the center of mass [10]. Some of the works focuses on control aspects of a sort of spherical robot, or model. They study the stability capability of their design, and some also include trajectory following as a case of study [10] [7]. It varies if one uses simulations for verification or others build a prototype and do tests with that one, or a combination.

A few designs focuses not just on land, but rather in water such as "A Novel Amphibious Spherical Robot Equipped with Flywheel, Pendulum, and Propeller"[11]. Facts about the benefits of the spherical design in water is explained, such as the resistance to water pressure. Some works are also specified to make a comparison on some controller designs on to a spherical robot [12], or uses some specific controller, such as a nonlinear model predictive control (NMPC) [9], feedback linearization loop with fuzzy controllers [13]. However, there are some works that are more focused on a theoretical basis, where almost all works include some description of a mathematical formulation of the model equation of motion, in various details. Such as to study the case if the sphere itself is not assumed to be totally rigid, but rather is flexible [14], and also how it behaves on a generic surface [15], such as inclined plane [16], with a variable slope [17]. A more mathematical formulation of geometric account of kinematic control [18] of spherical robots. Also, dynamics and motion planning strategy to make the control more suitable for real time applications [19], and optimal motion planning and control [20]. Because some have used Newton's laws of motion, others have used Lagrange's equation, and Hamiltonian mechanics, no obvious direct approach to formulate the model is found.

A notable paper collection that comes in two parts (released close to each other), that both is about "Controlled Motion of a Spherical Robot with Feedback" [21] [22]. Both papers are about the control of a spherical robot through feedback control, with the dependence on the phase variables (the current position and velocities).

A master thesis that is mentioned for the case of controlling a spherical robot was made in 2008 by *Nagai Masaki* [23]. That is about controlling a pendulum driven spherical robot, where he in his thesis did construct, and conducted tests with. It is shown a mathematical model of the drive system and steering system, where he later shows the progress to construct a PID controller for his system. That shows that even a relatively 'simple' controller has the capability to control a pendulum driven spherical robot. With an explanation of how the spherical robot was constructed, and a description of the part that made the real system. There a connection between the mathematical model, and the constructed spherical robot is shown.

The model to be made for this master thesis is a spherical robot which balances a mass on top. Can be considered to be an option to the internal construction of the fictional character BB-8 from the movie *Star Wars: The Force Awakens*. At *Chalmers University of Technology* there have been some candidate works regarding

to construct such robot type, by different methods of internal mechanics [24], [25], [26], [27]. Some of the different internal mechanics are based on omni-wheel design, pendulum drive methods, and combinations. All these may fall into the use of a pendulum construction way, by having a mass at the bottom and a sort of inverted pendulum to keeping the load on place at the top. Of course, it can be argued how much they resemble a pure pendulum with an inverted pendulum. However, that does not counteract the fact that they use a bottom mass and some inverted pendulum to balance the head. One specific candidate work has a similar model to the model which is to be used [25], that project was done by *Axel Andersson*, *Carl Andersson* ☺, *Gustav Andersson*, *Jacob Andrén*, *Jakob Laurell*, and *Fredrik Åvall*. They build up a mathematical model for both the drive system and steering, with a pendulum drive system that utilizes a central axis that is connected to the outer shell. With a sort of central connection box to be able to let the pendulum, and inverted pendulum move in a two-dimensional way.
One thing to notice is that these candidate works was to construct a spherical robot similar to the BB-8, therefore these projects are more focused to the construction of the robot. While this master thesis will be in more focused to the theoretical aspects of control, to compare the different control algorithms. Therefore, a new model will be constructed to fully embrace the non-linearity of the pendulum and inverted pendulum combined.

2.2 Mathematical Modelling

Making a mathematical model of a description may become difficult, and time consuming. Therefore, it is important to grasp what type of approach one should take in order to easy up the burden, and effectively use one's time. A brief explanation of some potential approaches will be made, with some aspects that are to be consider for this case of making a mathematical model of a system.

Mathematical Model Development

To formulate a sketch model into a mathematical model, different approaches exists for this formulation. Together with an uncovering model sketch, that shows relations between angles, lengths, and masses. Some potential approaches to make a mathematical model are.

- *Newton's laws of motion*
- *D'Alembert's Principle*
- *Euler-Lagrange equation*
- *Hamiltonian mechanics*

All of these are valid to be used in the formulation of the mathematical model, using an uncovering model sketch. The four named potential approaches of dealing with the formulation of mathematical model equations, will be briefly explained.

Newton's laws of motion Using *Newton's laws of motion* to formulate the mathematical model, one utilizes Newton's three laws in order to get the formulation of the equations. This method focuses on forces (F), Newton's three laws of motion is as follows:

First law "Every body continues in its state of rest or of uniform rectilinear motion, except if it is compelled by forces acting on it to change that state."[28]

Second law "The change of motion is proportional to the applied force and takes place in the direction of the straight line along which that force acts."[28]

Third law "To every action there is always an equal and contrary reaction; or, the mutual actions of any two bodies are always equal and oppositely directed along the same straight line."[28]

The meaning, and full understanding of these three laws will not be discussed in more details. It is to note that these formulations of the three laws are not exactly how Newton formulated them, however, it is close to the original one. [28]

D'Alembert's Principle The *D'Alembert's Principle*, which is also known as the *Lagrange–d'Alembert principle*, is in short considering every force's, and constraint's individually in a system. With respect to each individual virtual displacement, the virtual displacement may be defined by the definition 2.1.

Definition 2.1. *Virtual Displacements*
"A virtual displacement $\{\delta r_i\}$ of the system is an arbitrary, infinitesimal change of the coordinates that is compatible with the constraints and the applied forces. It is performed at a fixed time and therefore has nothing to do with the actual, infinitesimal motion $\{dr_i\}$ of the system during the time change dt (i.e. the real displacement)." [28]

D'Alembert's Principle can be considered in two cases, namely the *static case*, and *dynamical case*. For each case what is considered is the total force applied to each particle $\mathbf{F}_i$. The force for the *static case* can then be described by the equation 2.1.

$$\mathbf{F}_i = \mathbf{K}_i + \mathbf{Z}_i \tag{2.1}$$

$\mathbf{K}_i$ is the real dynamic force, and $\mathbf{Z}_i$ is the constraint force, for the N particles considered, with some mass. The general formulation equation of the force and the virtual displacement in equilibrium is then formulated as equation 2.2.

$$\sum_{i=1}^{N} \mathbf{F}_i \cdot \delta r_i \tag{2.2}$$

"However, since the virtual displacements must be compatible with the constraints, the total work of the forces of constraints alone vanishes"[28]. Giving that the static case becomes as shown in the equation 2.3.

$$\sum_{i=1}^{N} (K_i) \cdot \delta r_i = 0 \tag{2.3}$$

For the *dynamical case* in which the system is in movement, the *D'Alembert's principle of virtual displacements* then by that $\mathbf{F}_i - \dot{p} = 0$. Gives that the dynamical case becomes as shown in equation 2.4 [28]

$$\sum_{i=1}^{N} (K_i - \dot{p}_i) \cdot \delta r_i = 0 \tag{2.4}$$

Lagrange's equation *Lagrange's equation*, also known as *Euler–Lagrange's equation*, utilizes the energy of the system. More specifically the *kinetic energy* (T), and *potential energy* (U), to make the Lagrangian ($\mathcal{L}$).
Remark. *It is to be noted that the Lagrange's equation may be formulated with the D'Alembert's Principle, how this is done is not mentioned here.*

The *Lagrangian* is the *kinetic energy* subtracted the *potential energy* ($\mathcal{L} = T - U$). That is used in the *Lagrange's equation*, shown in equation 2.5.

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right) - \frac{\partial \mathcal{L}}{\partial q_i} = Q_i \tag{2.5}$$

$$\mathcal{L} = T - U \tag{2.6}$$

$\mathcal{L}$ is the *Lagrangian* of the problem, q_i is the *generalized coordinate* (considered states of the system), and Q_i are the *generalized forces* that act onto the system. Sometimes the *generalized forces* are assumed to be zero, depending on the problem at hand. Otherwise one needs to take the virtual work into consideration that the system is experiences, by the external forces, and constraints [28] [29].

Hamiltonian mechanics *Hamiltonian mechanics* which has similarities to *Lagrange's equation*, where not to got to deep into this, the *Hamiltonian mechanics* is using the *Hamiltonian function* which is calculated with the *Legendre transformation* of $\mathcal{L}$. With that f is the degree of freedom of $q = \{q_1, q_2, \ldots, q_f\}$. Then the *Hamiltonian function* is defined if the *Lagrangian* has no explicit time dependence by the equation (2.7) [28].

$$\tilde{H}(q, \dot{q}) = \sum_{k=1}^{f} \dot{q}_k \frac{\partial \mathcal{L}}{\partial \dot{q}_k} - \mathcal{L}(q, \dot{q}) \tag{2.7}$$

This approach, or method is not described further here, there is more aspects to this, and the above is just a very short, and narrow description.

2.3 Model Dynamics

The approach to be used in this master thesis is the *Lagrange's equation* (*Euler-Lagrange's equation*), by solving the equation (2.5). With the Lagrangian shown in equation (2.6). A brief explanation on this approach are mentioned in the section 2.2, with the Lagrangian, and the generalized forces to the system in question. The reason for choosing this approach instead of the other ones, is because of some properties that follows from its use, and will later be explained. However, it is not to say that the other approaches are not suitable. Then to make a model using the *Lagrange's equation* approach a more wider explanation of the approach will be given.

Lagrange's Equation *Lagrange's equation*, or *Euler–Lagrange's equation*, is as one may say an approach to formulate mathematical equations of a physical systems. Instead of using force, it utilizes energies of the system, more specifically here the *kinetic energy*, and *potential energy*. What these energies mean may be shortly formulated. *kinetic energy*, is the energy of the system in question which is making the system move. While the *potential energy*, is an energy that describes the energy that is stored in the system, that can be given from an increase in height. The *Lagrange's equation* uses *Lagrangian* which is defined for *kinetic energy* (T), and *potential energy* (U), given in equation (2.6). Using the equation (2.6) one may observe how the *Lagrange's equation* works by its definition. With the *generalized forces* in its formulation, which is sometimes regarded as zero, depending on the problem. The *Lagrange's equation* is then formulated as given in the equation (2.5). q_i is the *generalized coordinate* (considered states of the system), with $\dot{q}$ the derivatives of the considered states, and Q_i the *generalized forces* that effect the system. The *generalized forces* may be explained as, the forces that effect the system, which is not described by the dynamics of the system. Such as external forces, and constraints, by the virtual displacement of that force onto the body in question. Some standard way to formulate the *kinetic energy*, and *potential energy*, may be advantage to formulate, and observe. Therefore a general formulation of these energies may be formulated, for N particles considered in the system. For a position vector r_k, and its derivative v_k. Together with angles θ_k, and the angular velocity ω_k. Shown in the equations (2.8), and (2.11). [28]

- *Kinetic energy*:

$$T = \sum_{k=1}^{N} (T_{trans} + T_{rot}) \tag{2.8}$$

$$T_{trans} = \frac{1}{2} m_k v_k^2 = \frac{1}{2} m_k \dot{r}_k^2, \text{ Translational kinetic energy} \tag{2.9}$$

$$T_{rot} = \frac{1}{2} \omega_k I_k \omega_k = \frac{1}{2} \dot{\theta}_k I_k \dot{\theta}_k, \text{ Rotational kinetic energy} \tag{2.10}$$

- *Potential energy*:

$$U = \sum_{k=1}^{N} m_k g h_k \tag{2.11}$$

m is the mass, I the inertia, g gravity, and h is the height compared to a certain height level.

Remark. *A noticeable point about the energies, is that they may have other aspects to them, and not just the general formulations* (2.8)-(2.11)*. If one express it an energy, in terms of the kinetic-, or potential- energies in some sense, then the general formulation may not be the necessary right formulation. However, it gives a grasp on how it works. Example that may contribute to a slightly different energy terms, are if one considers magnetic fields.*

Observed from *Lagrange's equation* (2.5), is that it is quite straightforward on how to solve the left part. However, for the *generalized forces* it is not so obvious. Therefore to find the *generalized forces* of the system, one needs to take into consideration all the external forces, and constraints that effects the system. The forces are effecting the system by some sort of virtual work, that arise from the *virtual displacement* of the acting forces. *Virtual displacement* may be described by the equation (2.12), for all the *generalized coordinates M* [29]. With its definition 2.1.[28]

$$\delta r_k = \sum_{i=1}^{M} \frac{\partial r_k}{\partial q_i} \delta q_i \tag{2.12}$$

This gives that when a displacement of a particle in the system occur, that inflict a *virtual work* from the particle. Then gives that the *virtual work* can be formulated, by equation 2.13 [29].

$$\delta W = \mathbf{F} \delta r = \sum_{i=1}^{M} \mathbf{F} \cdot \frac{\partial r}{\partial q_i} \delta q_i \tag{2.13}$$

The definition of a *generalized force* Q_i may be found. Which is defined as a force times a infinitesimal *virtual displacement* δq_i, which gives that the equation (2.13), may be reformulated into equation (2.14) [29].

$$\delta W = \sum_{i=1}^{M} Q_i \delta q_i \tag{2.14}$$

This may be expressed for a single particle, instead of the sum of all. Expressed easier to be used for *Lagrange's equation* (2.5), by the equation (2.15) [29].

$$Q_i = \mathbf{F} \cdot \frac{\partial r}{\partial q_i} \tag{2.15}$$

Depending on the formulations of one's assumptions. Then as per definition, the *generalized forces* is a *virtual work*. Then even though there is a force, if no displacement occur there is no *virtual work* occurring form that force. For example, as *Anders Boström*, says in his notes. "If the system is holonomic and the coordinates are free then the constraints forces do not perform any virtual work"[29]. There exist more cases when the virtual work is changed, or becomes zero, that however, will not be addressed furthermore here.

2.4 External Forces

The external forces that may affect a system, can be a wide range of different types, and potentially become confusing. Therefore, a brief explanation of some external forces, that may occur onto a system will be shown. It is to be noted, is that there may be other forces that affect one's system. However, these should give some grasp onto some basic external forces.

Frictional External Forces Various external forces exist that may affect one's system. For the case of frictional forces, that is friction opposes motion, where its magnitude is independent of velocity and contact area [30]. One may categorize the different types of frictions into two groups, in a standstill, and moving case.

- *Static friction*

- *Dynamic friction* (*Kinetic friction*, or *Sliding friction*)

What those means is that *static friction* occur when the object in question is not moving. Therefore, there is a counteracting force acting upon it, which oppose the moment of the object. While *dynamic friction* is about the counteracting force that is acting on an object which is in motion. Creating a *dynamic friction*, or as the other name nicely put it a *sliding friction*. However, these may not be directly obvious on how to add into one's mathematical equation. Instead one can consider the different components that they are made of. When thinking about the frictions that effect the moment, one can model it by a static classical model [30], one can then obtain the friction by the different components of frictional forces by the terms.

- *Coulomb friction* (Sometimes *Dry friction*)
- *Viscous friction*
- *Stiction*
- *Stribeck effect* (Sometimes *Stribeck friction*)

That may together be visualized by the frictional force diagram illustration, shown in the Figure 2.1.

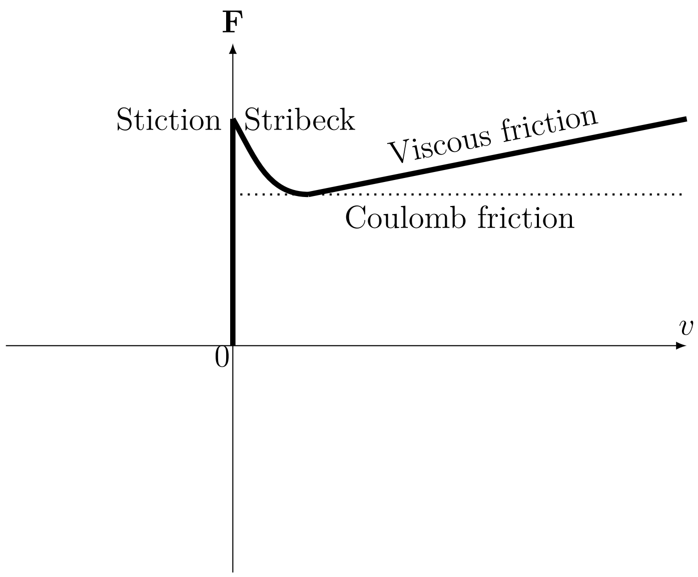


Figure 2.1: An illustration of different frictional forces (*Stribeck Friction Model*) that erupt from standing still to moving, where there names are represented in the figure.

If one assume that one's system is in a moving state, one may neglect the *Stiction*, and *Stribeck effect*. That's because it is more like a threshold from the *static friction* case, to the *dynamical fiction* case. While as can be observed from the Figure 2.1, is that the *Coulomb friction* is constant. That's because it is proportional to the normal force acting upon the place of interest. While the *viscous friction* is increasing with increased velocity.

Other External Forces Other types of external forces exist, than just frictional forces between two surfaces. Some may be interpreted as the same, where possible external force factors that may be argued to be considered may be.

- *Rolling resistance*
- *Braking friction*
- *Drag* (Also called *Air resistance*)

The interest reasons for these possible external forces is because. For *rolling resistance* that occur because of the normal force similarly to *dry friction*. Just that for the case of *rolling resistance* is more about the effect by deformation of surface. *Braking friction* may occur for a system, if there is something in its construction that may have a braking effect, or similar. *Drag* is airs effect onto the object, or wind disturbance.

External Forces Summary If one where to sum all these frictional forces, and other external forces that may occur in the system. One can express all these different types rather compressed. This may result in that all the external forces may be regarded as a summarized expression, as shown by equations (2.16), and (2.17).

$$F_s = \mu_s F_N \tag{2.16}$$

$$F_d = \mu_d F_N v \tag{2.17}$$

That is representing the different parts of a system, in moving state. F_s is the constant external force that act onto the system, and F_d is the velocity related external force. F_N is the normal force, v is the velocity and the coefficients μ_s, and μ_d needs to include all the considered coefficients. Such as *Coulomb friction*, *viscous friction*, *rolling resistance*, and *braking friction*, that effects the system. It may be needed to instead of just using the force directly, calculate the torque effecting the system. Also, express the velocity of an object into angular velocity. Torque (τ), and angular velocity (ω) may be found using the standard formulas, shown in the equations (2.18), and (2.19).

$$\tau = Fr \tag{2.18}$$

$$\omega = \frac{v}{r} \tag{2.19}$$

F is the force acting upon the object, and r is a length, and v a velocity.

The magnetic forces

The magnetic forces may be an external force. That's because of its properties to inflict forces onto the system. However, for clarity it is separated, because its effects are depending on some factors and therefore is suitable to be handled as its own feature. Different ways to define the magnetic forces exists, where one first consider the type of magnets in questions. One may have permanent magnets, electro-permanent magnet, or electromagnet.

Permanent Magnets As the name says they have a permanent magnetic field. Which is from the material which they are made of. This gives that no electricity is needed, and they will always be emitting a magnetic field. However, that gives no room for varying the intensity of the magnetic field.

Electro-permanent Magnet Has similarities to *permanent magnets* exempt to that it has a sort of switch that can turn on, and off the magnetic field. This switch is controlled by electric current.

Electromagnet Is creating its magnetic field by electric current. Which gives that it can control the intensity of the magnetic field.

To model the magnetic field there exist numerous ways to approach this, where different aspects to consider is the: *Coulomb's Law and the Electric Field*, *Gauss's Law and the Electric Potential*, *Faraday's Law and Induction*, and *Maxwell's Equations* [31]. Then gives information on how one may add into consideration the magnetic field to the modeling aspects of the system. However, it is here assumed that the magnetic field is constant and will not be considered more into. Which may give additional frictional forces onto the system, and potentially some additional disturbance.

2.5 Nonlinear State-Space Model Representation

After modeling one may want to structure it, both for visualization and calculation purposes. Therefore, for a nonlinear model system, one may form it in a state-space format. To structure the nonlinear equations, one may utilize that. A nonlinear system can be modeled by a finite number of coupled first-order ordinary differential equations (2.20) [32].

$$\begin{aligned} \dot{x}_1(t) &= f_1(t, x_1(t), ..., x_{n_x}(t), u_1(t), ..., u_{n_u}(t)) \\ \dot{x}_2(t) &= f_2(t, x_1(t), ..., x_{n_x}(t), u_1(t), ..., u_{n_u}(t)) \\ &\vdots \\ \dot{x}_{n_x}(t) &= f_{n_x}(t, x_1(t), ..., x_{n_x}(t), u_1(t), ..., u_{n_u}(t)) \end{aligned} \tag{2.20}$$

Internal states of the system $x_i(t)$, $i = 1, 2, 3, ..n_x$ and input states of the system $u_i(t)$, $i = 1, 2, 3, ..n_u$. That can be described with matrices and vectors, shown in (2.21) [32].

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix}, u(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_n(t) \end{bmatrix}, f(t, x, u) = \begin{bmatrix} f_1(t, x(t), u(t)) \\ f_2(t, x(t), u(t)) \\ \vdots \\ f_n(t, x(t), u(t)) \end{bmatrix} \tag{2.21}$$

That results in a (continuous) time varying nonlinear model, that can be represented by the state-space formulation in equations (2.22), and (2.23).

$$\dot{x}(t) = f(t, x(t), u(t)) \tag{2.22}$$

$$y(t) = h(t, x(t), u(t)) \tag{2.23}$$

With internal states $x(t)$ of the system and the input of the system $u(t)$ [32]. To represent the nonlinear system in a state-space representation. One could use a large matrix where all considered states and combinations of considered states are put into the states vector. Similarly, to a linear state-space representation. However, this may result in very large matrices. Another way is to consider the use of several matrices, each have a certain state vector to it. Then one could use a more concise nonlinear state-space representation. An example formulation of this is shown in equation (2.24).

$$A \begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \\ \vdots \\ \ddot{x}_n \end{bmatrix} + B \begin{bmatrix} \dot{x}_1^2 \\ \dot{x}_2^2 \\ \vdots \\ \dot{x}_n^2 \end{bmatrix} + C \begin{bmatrix} \dot{x}_1\dot{x}_2 \\ \dot{x}_1\dot{x}_3 \\ \vdots \\ \dot{x}_{n-1}\dot{x}_n \end{bmatrix} + D \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} + G = E \tag{2.24}$$

The matrices A, B, C, and D are shown what variables they depend upon in the equation (2.24). However, for G and E it may not be that apparent. The matrix G is for the nonlinear 'constant' terms, that may contain constants, but also trigonometrical functions. For the matrix E, that matrix contains the effects of the external forces that acts onto the system. That depending on what approach one chose to use, may wary somewhat to either be included in the equation directly or separately considered. The formulation may be expanded to include more variables if needed.

2.6 Linear State-space Model Representation

To linearize the nonlinear model equations given in the chapter 2.5, equations (2.22)-(2.23). One first need to have a reasonable stationary point, to linearize at (other used names for this point are, work point, operating point, reference point, linearization point, equilibrium point, steady state point). This stationary point should be around the working, or operating area, and beneficially should be chosen to be an equilibrium point, or steady state point. The reason for the importance of choice, is the nonlinear property that is lost in the linearization. Which may result in that the further away from that point, the larger the error becomes (How close one need to be is depended on the system of interest). A stationary point named x_0 for the internal states, and u_0 for the input states will be used. When formulating a general expression of the linearization, the stationary point for a system is a topic of discussion, that will be discussed later.
Remark. *For notation convenience and to follow some typical notations the linear and nonlinear states $x(t)$, and $u(t)$, have the same notions. However, these are not to be confused to be the same in the actual calculations.*

Small-angle Approximation

In the case that one's nonlinear mathematical expression contains trigonometrical functions, simplification and approximation of these terms may be advantageous. The trigonometrical functions may be approximated by using *Taylors series*, *Taylors series* expression of sinus, and cosinus trigonometrical equations is shown in equations (2.25), and (2.26).

$$\sin(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1} \qquad = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots \tag{2.25}$$

$$\cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n} \qquad = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots \tag{2.26}$$

For this case where the objective is to linearize the nonlinear model, it may be suitable to use first-order approximation or second-order approximation. This method is called *small-angle approximation* and as the name say it is an approximation of these trigonometric functions when they have small angels. Using the first-order *small-angle approximation* and second-order *small-angle approximation*. Then by *Taylor series* of the trigonometrical equations (2.25), and (2.26), first-, and second-order becomes as shown in equations (2.27)-(2.30).

First-order *small-angle approximation*

$$\sin(\theta) \approx \theta \tag{2.27}$$

$$\cos(\theta) \approx 1 \tag{2.28}$$

Second-order *small-angle approximation*

$$\sin(\theta) \approx \theta \tag{2.29}$$

$$\cos(\theta) \approx 1 - \frac{\theta^2}{2} \tag{2.30}$$

Linearization

Using the stationary point (x_0, u_0) to linearize the nonlinear model equations given in the chapter 2.5, equations (2.22), and (2.23). To get the (continuous) linear-time invariant (LTI) model equations which may be described by the general (continuous) LTI state-space notation of equations (2.31), and (2.32), with respective name. Notice that nonlinear and linear states x_0, and u_0 have the same notations, and the matrices A, B, C, and D. That is for notation convenience and to follow some typical notations, not to be confused to be the same.

$$\textit{State (dynamic) equation: } \dot{x}(t) = Ax(t) + Bu(t) \tag{2.31}$$

$$\textit{Measurement equation: } y(t) = Cx(t) + Du(t) \tag{2.32}$$

$x(t)$ are the states, and $u(t)$ the input signals. This gives that the formulation of the linearization, using the general (continuous) nonlinear functions (2.22), and (2.23), shown here for clarity.

$$\dot{x}(t) = f(t, x(t), u(t))$$
$$y(t) = h(t, x(t), u(t))$$

Using the general (continuous) nonlinear functions (2.22), and (2.23). One may formulate the Jacobean matrices, $\left[\frac{\partial f}{\partial x}\right]$, $\left[\frac{\partial f}{\partial u}\right]$, $\left[\frac{\partial h}{\partial x}\right]$, and $\left[\frac{\partial f}{\partial u}\right]$, to approximate the nonlinear model equations around a stationary point. That is defined as shown in the equations (2.33), and (2.34).

$$x(t) = \delta x(t) - x_0 \tag{2.33}$$
$$y(t) = \delta y(t) - y_0 \tag{2.34}$$

Then the Jacobian linearization [33] of the (continuous) nonlinear system (2.22), and (2.23), becomes with its higher order term as shown in the equations (2.35), and (2.36).

$$\frac{d}{dt}\left(x_0 + \delta x(t)\right) = f(x_0, u_0) + \left[\frac{\partial f}{\partial x}\right]_{x_0,u_0} \delta x(t) + \left[\frac{\partial f}{\partial u}\right]_{x_0,u_0} \delta u(t) + H.O.T. \tag{2.35}$$
$$\delta y(t) = \left[\frac{\partial h}{\partial x}\right]_{x_0,u_0} \delta x(t) + \left[\frac{\partial h}{\partial u}\right]_{x_0,u_0} \delta u(t) + H.O.T. \tag{2.36}$$

The Higher Order Terms (H.O.T) may be neglected and as one may observe from the state equation (2.35), is that it contain a constant term named $f(x_0, u_0)$. If this term is not equal, or sufficiently close to zero one would get a constant term in the expression. Resulting in that the linearization will not be equivalent to the linear state-space representation given in equations (2.31), and (2.32). With the matrices defined as shown in (2.37).

$$A = \left[\frac{\partial f}{\partial x}\right]_{x_0,u_0}, B = \left[\frac{\partial f}{\partial u}\right]_{x_0,u_0}, C = \left[\frac{\partial h}{\partial x}\right]_{x_0,u_0}, D = \left[\frac{\partial h}{\partial u}\right]_{x_0,u_0} \tag{2.37}$$

The Jacobian's are defined as matrices and are defined in general for a function $f \in \mathbf{R}^m$, with the states $x \in \mathbf{R}^n$. Shown in equation (2.38).

$$\left[\frac{\partial f}{\partial x}\right] = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \tag{2.38}$$

Therefore as one can observe form state dynamic equation (2.35), it is of importance to ensure that $f(x_0, u_0) = 0$. To linearize the nonlinear model equations into a linear model equation. That is the reason for the important choice of stationary point.

2.7 Model Analysis

From a linear and nonlinear model, one may want to observe and identify certain characteristics that the system has. This can be by analyzing the model, some tools and theoretical background of these properties will be described and explained.

2.7.1 Linear State-space Model Analysis

Using the linear state-space model from equations (2.31), and (2.32). Then one may analyze different aspects of the linear system, some of these are.

- Eigenvalues
- State Reachability
- State Controllability
- State Stabilizability
- State Observability
- State Detectability

These are important to consider in order to observe if the system can be controlled and that the states are observable. While if they are not, that at least they are stable. The properties will be explained in what they mean, and why they are important to consider. With how one may determine if they hold for a linear system.

Definition 2.2. *Eigenvalues Analysis*
For a linear time invariant (LTI) system: $\dot{x}(t) = Ax(t) + Bu(t)$*, where* $x(t)$ *is the states of the system, and* $u(t)$ *is the input states of the system. Then the eigenvalues, and eigenvectors of the LTI system are defined by*

$$Av = \lambda v \tag{2.39}$$

Which may be found by the determinant of $A - \lambda I_n$*, there* λ *is the eigenvalues, with its corresponding eigenvector* v*. The square matrix* A *is describing the internal dynamics of the system, where if the eigenvalue if on the right hand plane (RHP) then the system is instable. While if in the left hand plane (LHP) it is asymptotically stable.*

Definition 2.3. *Reachability*
A linear system is reachable if, for any $x_0, x_f \in \mathbf{R}^n$*, there exists a* $T > 0$ *and* $u : [0, T] \longrightarrow \mathbf{R}$ *such that the corresponding solution satisfies* $x(0) = x_0$ *and* $x(T) = x_f$ *[33].*

One possible test for *state reachability* of a linear system, (definition 2.3) is to use the *reachability matrix*. That is defined by equation (2.40).

$$\mathcal{R} = \begin{bmatrix} B & AB & A^2B & \dots & A^{n_x-1}B \end{bmatrix} \tag{2.40}$$

The system is reachable if the *reachability matrix* (equation (2.40)) is of full rank, if n is the number of states then ($rank(\mathcal{R}) = n$).

Definition 2.4. *State Controllability*
The dynamical system $\dot{x}(t) = Ax(t) + Bu(t)$ *or equivalently the pair* (A, B)*. Is said to be state controllable if, for any initial state* $x(0) = x_0$*, any time* $t_1 > 0$ *and any final state* x_1*, there exists an input* $u(t)$ *such that* $x(t_1) = x_1$*. Otherwise the system is said to be state uncontrollable [34].*

Two possible tests for *state controllability*, (definition 2.4) is to either use the *controllability matrix*, or the *controllability Gramian*. The *controllability matrix* is defined by equation (2.41).

$$\mathcal{C} = \begin{bmatrix} B & AB & A^2B & \dots & A^{n_x-1}B \end{bmatrix} \tag{2.41}$$

The system is controllable if the *controllability matrix* (equation (2.41)) is of full rank, if n is the number of states then ($rank(\mathcal{C}) = n$). Which is also the same for the use of *controllability Gramian*, if full rank then the system is controllable.

Remark. *It is to be noted in continuous systems the Reachability, and controllability properties of definition, is giving that if one holds then the other also holds. Which may be observed with that the reachability matrix, and controllability matrix is defined exactly the same. (Depending on literature theirs names may just be reachability matrix, or separated as shown, for clarity). However, this is not necessarily true for discrete systems.*

Definition 2.5. *State Stabilizability*
A system is stabilizable if all unstable modes are state controllable. While a system with unstabilizable modes is said to contain hidden unstable modes [34].

Definition 2.6. *State Observability*
The dynamical system $\dot{x}(t) = Ax(t) + Bu(t)$, $y(t) = Cx(t) + Du(t)$ *(or the pair* (A, C)*). Is said to be state observable if, for any time* $t_1 > 0$*, the initial state* $x(0) = x_0$ *can be determined from the time history of the input* $u(t)$ *and the output* $y(t)$ *in the interval* $[0, t_1]$*. Otherwise the system, or* (A, C)*, is said to be state unobservable [34].*

Two possible ways to test *state observability*, (definition 2.6) is to either use the *observability matrix*, or the *observability Gramian*. The *observability matrix* is defined by equation (2.42).

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ \dots \\ CA^{n_x-1} \end{bmatrix} \tag{2.42}$$

The system is observable if the *observability matrix* (equation (2.42)) is of full rank, if n is the number of states then ($rank(\mathcal{O}) = n$). Which is also the same for the use of *observability Gramian*, if full rank then the system is observable.

Definition 2.7. *State Detectability*
A system is detectable if all unstable modes are observable. While a system with undetectable modes is said to contain hidden unstable modes [34].

An important part of an analysis of a system, is if the system is stable or not. This importance can be interpreted in different ways, such as input-output stable, or internally stable. Both tells if a system is stable or not, but for different ways of viewing stability of the system.

Definition 2.8. *Internal Stability*
A system is (internally) stable if none of its components contain hidden unstable modes, and the injection of bounded external signals at any place in the system results in bounded output signals measured anywhere in the system [34].

What this *Internal Stability* more or less means, is that there does not exist some hidden dynamics of the system which is unstable. Which is guaranteed if the system is *stabilizable*, definition 2.5. This may be observed by the eigenvalues of the system dynamics, more specifically from the linear state-space, matrix A. As explained by the *Eigenvalue Analysis*, definition 2.2.

Definition 2.9. *Input-Output Stability*
For a linear system Input-Output stability can be interpreted to be that given a bounded input, then the system should give a bounded output.

Bounded Input Bounded Output (BIBO) is that if the input is not going to infinity, the output will not go to infinity.

Theorem 2.1. *Minimality*
Given the triplet (A, B, C), the LTI state space system is called minimal if it is both observable and controllable at the same time.

If one has made a controller of some sort, one may want to observe how well the controller is doing in different cases. That however may sometimes not be an option for a real time application, or too costly to simulate for all (or many) cases. Therefore, one may utilize some properties that consider the system with the controller. It can consider cases of stability, performance, robustness, uncertainties in the model, and such. The properties, or conditions are.

- Nominal Stability (NS)
- Nominal Performance (NP)
- Robust Stability (RS)
- Robust Performance (RP)

The properties are useful tools to observe how well the controller is acting towards the system in different cases of not only stability, but more aspects such as the performance in cases with perturbations to the system.

Definition 2.10. *Nominal Stability (NS)*
The system is stable with no model uncertainty [34].
This property may be observed to hold if the interconnected structure N, by the lower LFT, definition 2.18, is internally stable, by the definition 2.8.

Definition 2.11. *Nominal Performance (NP)*
The system satisfies the performance specification with no model uncertainty [34].
This property may be observed to hold if the interconnected structure N is nominal stable, by the definition 2.10, and that the infinity norm of the transfer from w to z is below one ($\|N_{22}\|_\infty < 1$), as shown in the Figure 2.2a.

Definition 2.12. *Robust Stability (RS)*
The system is stable for all perturbed plants about the nominal model up to the worst-case model uncertainty [34].
This property may be observed to hold if the interconnected structure N is nominal stable, by the definition 2.10, and that the infinity norm of the transfer from u_Δ to y_Δ is below one ($\|N_{11}\|_\infty < 1$), as shown in the Figure 2.2b. When the perturbation coming from Δ is limited such that $\|\Delta\|_\infty \leq 1$.

Definition 2.13. *Robust Performance (RP)*
The system satisfies the performance specifications for all perturbed plants about the nominal model up to the worst-case model uncertainty [34].
This property may be observed to hold if the robust stability holds, by the definition 2.12, and that the effect of the disturbance w will never be amplified over z, when the

perturbation coming from Δ is limited such that $\|\Delta\|_\infty \leq 1$, followed the structure shown in Figure 2.2b. That may be calculated by observing if the infinity norm of the interconnected structure N is less than 1, ($\|N\|_\infty < 1$).

One more analysis tool that are of interest is the μ-analysis. This will be described later on how it works, in the chapter 2.9 under the title of μ-synthesis. That are based on the same unit called structured singular value, or μ.

2.7.2 Nonlinear State-space Model Analysis

To analyze a nonlinear system model, is more difficult than for the linear system model. That is because nonlinear systems have some properties that differs from a linear system. Which may provide with greater challenges when trying to analyze the nonlinear system. Some phenomenons that occurs with nonlinear systems, and not in linear systems may be [32]

- *Finite escape time*
- *Multiple isolated equilibria*
- *Limit cycles*
- *Subharmonic, harmonic, or almost-periodic oscillations*
- *Chaos*
- *Multiple modes of behavior*

These may prove to make calculations more difficult, and potentially result in a system which is not controllable, and stabilizable. Therefore, because of the increased difficulties of analyzing a nonlinear system. This topic will be brief, mainly focusing on a certain method. It is to be noted that all the points above is only to show that there exist several things to be considered when dealing with nonlinear systems. One method of determine the stability of the nonlinear system is to use the *Lyapunov stability* criteria which is a sufficient (but not necessary) condition [34]. That may be explained by the *Lyapunov's theorem* 2.2.

Theorem 2.2. *Lyapunov's Stability theorem*
Given a positive definite function $V(x) > 0 \quad \forall x \neq 0$ and a autonomous system $\dot{x} = f(x)$, then the system $\dot{x} = f(x)$ is stable if [34]

$$\dot{V}(x) = \frac{\partial V}{\partial x} f(x) < 0 \quad \forall x \neq 0 \tag{2.43}$$

Remark. *It is to be noted that using the Lyapunov's theorem may be very hard for some nonlinear systems. Since one need to find a positive definite function $V(x)$, that both suits ones needs and is a valid Lyapunov function. In such case one may turn to other methods, or when the Lyapunov stability fails, one can turn to other method to verify. That is because it is not a necessary condition.*

2.7.3 Definitions of Norms

A norm can be described as to be a "a single number which gives an overall measure of the size of a vector, a matrix, a signal, or a system." [34]. The general definition of a vector norm is defined as given in the definition 2.14, with the equation (2.44).

Definition 2.14. *Vector Norms*
The general definition of vector norms are expressed as equation (2.44).

$$\|q\|_p = \left(\sum_i |q_i|^p\right)^{1/p} \tag{2.44}$$

For the definitions of the system norms, $\mathcal{H}_2$ and $\mathcal{H}_\infty$, one should note that they are based on matrix norms. If $G(s)$ and $G(t)$, is the plants or process, transfer function of the system. Then the different system norms are defined in the definitions 2.16, and 2.17, following the Parseval equality, definition 2.15.

Definition 2.15. *Parseval theorem (equality)*
Let $g(t)$ be the function g in its time domain, where $G(s)$ is the Laplace transformation of $g(t)$, which is defined in the frequency domain. Then by Parseval theorem, the equality of the norms is given as

$$\|g(t)\|_2 = \|G(s)\|_2 \tag{2.45}$$

Definition 2.16. *$\mathcal{H}_2$ system norm*
Let G be a proper linear stable system, and H stand for the complex conjugate transpose. Then the $\mathcal{H}_2$ system norm is defined [34]

$$\|G(s)\|_2 \triangleq \sqrt{\frac{1}{2\pi}\int_{-\infty}^{\infty} tr\left(G(j\omega)^H G(j\omega)\right) d\omega} = \|g(t)\|_2 \triangleq \sqrt{\int_0^{\infty} tr\left(g^\intercal(\tau) g(\tau)\right) d\tau} \tag{2.46}$$

Definition 2.17. *$\mathcal{H}_\infty$ system norm*
Let G be a proper linear stable system, and $\bar{\sigma}$ is the maximum singular value. Then the $\mathcal{H}_\infty$ system norm is defined [34]

$$\|G(s)\|_\infty \triangleq \max_\omega \bar{\sigma}\left(G(j\omega)\right) = \max_\omega \max_{w(\omega)\neq 0} \frac{\|z(\omega)\|_2}{\|w(\omega)\|_2} = \max_{\|w(\omega)\|_2 = 1} \|z(\omega)\|_2 \tag{2.47}$$

Remark. *$\mathcal{H}_\infty$ system norm, definition 2.17 is both the $\mathcal{H}_\infty$ norm, and the induced (worst-case) 2-norm. This also holds for time domain also, by Parseval's theorem, definition 2.15.*

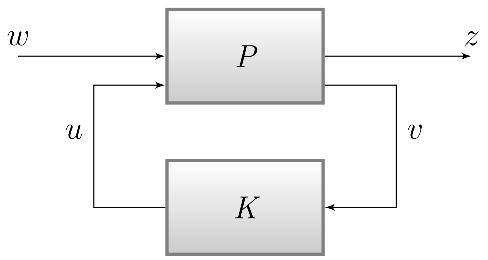
One may wonder how to compute the $\mathcal{H}_\infty$ system norm, one way is to use Hamiltonian matrix H. That is calculated numerically from the linear state-space realization. It is a iterative process to find the smallest γ value such that all the Hamiltonian matrix, equation (2.48), eigenvalues have imaginary values (there is no eigenvalues on the imaginary axis). The Hamiltonian matrix H is defined as in equation (2.48) [34].

$$H = \begin{bmatrix} A + BR^{-1}D^\intercal C & BR^{-1}B^\intercal \\ -C^\intercal \left(I + DR^{-1}D^\intercal\right) C & -\left(A + BR^{-1}D^\intercal C\right)^\intercal \end{bmatrix} \tag{2.48}$$

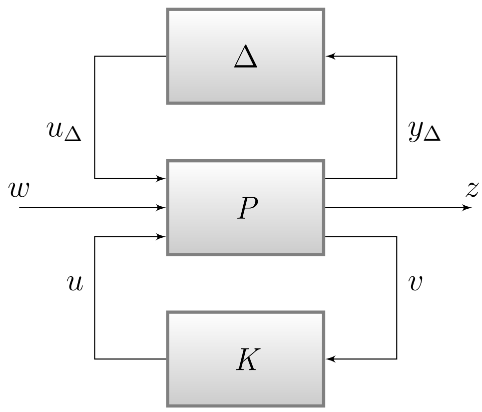
With that $R = \gamma^2 I - D^\intercal D$, as mentioned before this is an iterative procedure That one begin with a suitable large value for γ, and decrease it in steps [34].

2.8 Control Structure

Many ways exist in which feedback design problems could be formulated. It is therefore very useful to have a standard problem formulation of the control problem, a general control problem formulation. That any of the particular problems could be manipulated [34]. Two forms of standard problem formulation, namely one with, and without uncertainty, would be useful for the different control algorithms to be used. Both structures are shown in Figures 2.2.



(a) Standard problem formulation of feedback design, without uncertainty



(b) Standard problem formulation of feedback design, with uncertainty

Figure 2.2: Standard problem formulations of feedback designs

K is the controller, P the plant, and Δ the uncertainty. w potentially contains several different signals, these signal may be $w = \begin{bmatrix} n & r & d \end{bmatrix}^{\intercal}$. n is the measurement noise, r is the reference signal, and d is the disturbance signal. It is worth to notice that depending on one's system, they do not need to be a part of the system. However, for generality they are included here. The output of the system, z, may contain different signals depending on what one wants to take into consideration. A common output may be to observe the feedback signal v. The two standard problem formulations of feedback design Figures 2.2a, and 2.2b. Can be formulated together as one box, this is done with *linear fractional transformation* (LFT). That have two types of, depending on which blocks one box together, namely *lower* LFT, and *upper* LFT.

Definition 2.18. *lower Linear Fractional Transformation*
Following the standard structure of Figure 2.2a, and consider the combined block to be the block N, then N can be calculated by equation (2.49)*.*

$$N = F_l\left(P, K\right) = P_{11} + P_{12}K\left(I - P_{22}K\right)^{-1}P_{21} \tag{2.49}$$

Definition 2.19. *upper Linear Fractional Transformation*
Following the standard structure of Figure 2.2b, and that the Figure 2.2a tells what the block N is consisting of. Then with the uncertainty block Δ and N one can

calculate the combination by equation (2.50).

$$F_u(N, \Delta) = N_{22} + N_{21}\Delta (I - N_{11}\Delta)^{-1} N_{12} \tag{2.50}$$

The Figures 2.2a, and 2.2b, shows a general structure for a feedback system, with and without uncertainty. One may add some sort of weights to each signal of w and z, to reduce and determine the effects of these signals or for stability purposes. The weights can be filters or constants, where one can have diagonal matrices as weights. The added weights have different purposes, there one is to determine the output for the system, namely z. This is a design choice to be made, that may affect the stability and performance of the system. Which may for example be a measurement of the error of the system to some reference signal. Another important part of the system, as it may be designed with or without an uncertainty block Δ. If one does have that one considers uncertainties in the system, which may be separated in two different terms of uncertainties, namely structured and unstructured terms.

- *Structured uncertainty (real or complex)* (or, *Parametric uncertainty*)
- *Unstructured uncertainty* (or, *Dynamic (frequency-dependent) uncertainty*

Parametric uncertainty it considers the uncertainty of ones parameters, such as masses, lengths, and other parameters uncertainties. Which can be modeled in different ways, such as in percentage of a nominal value or in units around its nominal value. As for *unstructured uncertainty* it considers the error made from missing dynamics in the model, typically from the higher frequencies. This may be from different reasons, such as deliberate neglecting some dynamics. Potentially because that it is either hard to model, or ones lack of understanding of the physical system. This *unstructured uncertainty* is always present for a real system. [34]

2.9 Control Algorithms

Here the different linear and nonlinear control algorithms that will be evaluated, will be defined and explained to some extent. The control algorithms will be generally explained and include a brief explanation of PID - controller which will not necessary be used for evaluation. However, it gives a certain perception on control, and therefore is also included. An important point to make out about all the different control algorithms is that to be able to control a system, the system needs to be stabilizable as defined in 2.5. If it is controllable as defined in 2.4, then it is stabilizable, which is assumed the system is.

2.9.1 Linear Control Algorithms

The evaluated linear control algorithms will be formulated here using the linear mathematical model. That either is defined as linear state-space model, or have been linearized at a certain stationary point, as shown in chapter 2.6.

2.9.1.1 PID - Controller

A PID Controller, is a relative simple controller that one may say utilize three different mathematical ways, or terms to control. Namely *Proportional*, *Integral*, and

Derivative, summarized as PID for short. That is the reason that is was originally called *three-term controllers* [33]. There exist several ways one may define a PID controller. What I mean is that because the PID controller is such a common controller to use in different applications, such as in more than 95% of all industrial control problems are solved PID control. Of these mostly PI controllers are used, (*Proportional-Integral* controller) [33], and "is the most widely used control algorithm in industry" [34]. For each way to control there are one adjustable parameter. The case of a general expression of how the PID controller works, may be described and shown by observing its input control signal $u(t)$ directly, which is defined for a PID by the equation (2.51) [33].

$$u(t) = k_P e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{de(t)}{dt} = k_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (2.51)$$

How the PID controller works is that it takes into consideration the output as feedback to the controller. That depending on if one uses a one, or two degree of freedom controller, it can be defined differently. As mentioned before there exist different ways to define the PID controller, even to that if one just wants a PI, PD, or P controller. Also, one may argue on how one should tune the weights of the PID controller, and so on. However, a major thing with PID controllers is that they handle single input single output systems, or SISO for short. This will give some restriction to the control purpose, that may result in that one need to have more than one PID controller, to handle system with more than one inputs and outputs. Therefore, if one wish to use a PID controller, there are three parameters to tune, which may become time consuming. However, there exist some different tuning approaches to use. One is the **Ziegler-Nichols Tuning**, which is a method developed by *Ziegler*, and *Nichols*, for controller tuning. That is based on simple characterization of process dynamics in the time and frequency domains [33]. Also, an newer approach is the **SIMC** (Skogestad/Simple IMC).

Skogestad/Simple IMC The PID control feedback gain can be given by the 'ideal' (or parallel) form, in Laplace domain, equation (2.52) [34].

$$K_{PID}(s) = K_c \left(1 + \frac{1}{\tau_I s} + \tau_D s \right) \quad (2.52)$$

K_c is the proportional gain, τ_I is the integral time, and the τ_D is the derivative time. It is described a typical implementation of a PID controller [34], this controller is a two degrees of freedom controller. With the control signal becoming, in Laplace domain, as equation (2.53).

$$u(s) = K_c \left[\left(1 + \frac{1}{\tau_I s} \right) (r(s) - y_m(s)) - \frac{\tau_D s}{\epsilon \tau_D s + 1} y_m \right] \quad (2.53)$$

SIMC (Skogestad/Simple IMC) which is a "PID design for first- or second-order plus delay process" [34], that won't be described in more details here. It exists more ways to determine the parameters, to get a working PID controller, which can be used.

2.9.1.2 Linear Quadratic Controllers (LQR, LQI, LQG)

Linear Quadratic Controllers, or LQ for short, exist in three general chases. Such as *Regulator* (LQR), *Integrator* (LQI), and *Gaussian* (LQG) (exists also LQGI). All these are similar in their cost function structure, with is as the name says quadratic. The three are different in some structure, such as, an added integrator function, or with a Kalman filter that estimates the states from the system output. In general, for the LQ controllers is that they are utilizing a quadratic cost function, which can be described by the equation (2.54).

$$J(x,u) = \frac{1}{2}\int_0^{\infty} \left(x^{\intercal}(\tau)Q_x x(\tau) + u^{\intercal}(\tau)Q_u u(\tau) + 2x^{\intercal}(\tau)Nu(\tau)\right) d\tau \tag{2.54}$$

However, there exist also a type of LQ controller that utilizes a sort of filter, which estimates the states from the system output, more specifically a Kalman filter. Named LQG, for *linear quadratic Gaussian*, that has a slightly variant of the cost function in equation (2.54), which is defined in the average sense as equation (2.55).

$$J(x,u) = \lim_{T\to\infty} E\left\{\frac{1}{2}\int_0^{T} \frac{1}{T}\left(x^{\intercal}(\tau)Q_x x(\tau) + u^{\intercal}(\tau)Q_u u(\tau) + 2x^{\intercal}(\tau)Nu(\tau)\right) d\tau\right\} \tag{2.55}$$

To yield the optimal solution, i.e. min $J(x,u)$. One may utilize *Lagrange's equation* shown in chapter (2.5), to find the controller. With the considered states of $q(t) = \{x(t), \lambda(t), u(t)\}$, with the *Lagrangian* defined by.

$$\mathcal{L}(t,x,u,\lambda) = x^{\intercal}(t)Q_x x(t) + u^{\intercal}(t)Q_u u(t) + 2x^{\intercal}(t)Nu(t) + \lambda^{\intercal}(t)\left(Ax(t) + Bu(t) - \dot{x}(t)\right)$$

With that the system is a linear system, the *generalized forces* becomes zero, $Q_i = 0$. This may be more time consuming than utilizing the *Riccati equation*. Which is the result from that calculation and therefore will not be addressed any further here. This gives for different continuous cases of *Riccati equation*, what type of formulation one may need to use. Some of the different ones are.

- (Continuous) Differential Riccati equation (DRE)
- Algebraic Riccati equation (ARE)
- Continuous Algebraic Riccati equation (CARE)
- Filter Algebraic Riccati equation (FARE)

These are given for the continuous case, discrete case for each type exists. Then instead they then utilize *Difference Riccati equation*, and evaluate it to *Discrete-time Algebraic Riccati equation*, or DARE instead of CARE. However, this will not be discussed further, just that there exist for discrete time to. It is to be noted that they do not necessarily have the same formulation.

Definition 2.20. *(Continuous) Differential Riccati equation (DRE)*
Let (A,B) be the linear state space matrices ($\dot{x}(t) = Ax(t) + Bu(t)$), and $P \in \mathrm{R}^{n\times n}$ is a positive definite, symmetric matrix that satisfy the equation (2.56)*, based on the cost function equation* (2.54)*.*

$$\dot{P}(t) + A^{\intercal}P(t) + P(t)A - \left(P(t)B + N\right)Q_u^{-1}\left(B^{\intercal}P(t) + N^{\intercal}\right) + Q_x = 0 \tag{2.56}$$

If the matrix N is equal to a zero matrix, then $P \in \mathrm{R}^{n\times n}$ *is a positive definite, symmetric matrix that satisfy the equation* (2.57) *[33].*

$$\dot{P}(t) + A^{\intercal}P(t) + P(t)A - P(t)BQ_u^{-1}B^{\intercal}P(t) + Q_x = 0 \tag{2.57}$$

Remark. *There exists also a algebraic Riccati equation, however to not confuse with notations, that one is not directly shown. Since the algebraic Riccati equation, or ARE, is similar to the (continuous) differential Riccati equation shown. With the exception that* $\dot{P}(t) = 0$*. The ARE are similar to the continuous algebraic Riccati equation, or CARE, and therefore won't have its own definition.*

Definition 2.21. *Continuous Algebraic Riccati equation (CARE)*
Let (A,B) be the linear state space matrices ($\dot{x}(t) = Ax(t) + Bu(t)$)*, and* $P \in \mathrm{R}^{n\times n}$ *is a positive definite, symmetric matrix that satisfy the equation* (2.58)*, based on the cost function equation* (2.54)*.*

$$A^{\intercal}P + PA - (PB + N)\,Q_u^{-1}\,(B^{\intercal}P + N^{\intercal}) + Q_x = 0 \tag{2.58}$$

If the matrix N is equal to a zero matrix, then $P \in \mathrm{R}^{n\times n}$ *is a positive definite, symmetric matrix that satisfy the equation* (2.59) *[33].*

$$PA + A^{\intercal}P - PBQ_u^{-1}B^{\intercal}P + Q_x = 0 \tag{2.59}$$

The state-feedback gain K is defined as equation (2.60)*.*

$$K = -Q_u^{-1}\,(B^{\intercal}P + N^{\intercal}) \tag{2.60}$$

For the case of the *Filter Algebraic Riccati equation (FARE)*, that is based on the theorem, (Kalman-Bucy, 1961 [35]), which says [33].

Theorem 2.3. *(Kalman-Bucy, 1961 [35])*
The optimal estimator has the form of a linear observer

$$\frac{d\hat{x}(t)}{dt} = A\hat{x}(t) + Bu(t) + L\,(y(t) - C\hat{x}(t))$$

Where $L(t) = P(t)C^{\intercal}R_w^{-1}$ *and* $P(t) = E\{(x(t) - \hat{x}(t))(x(t) - \hat{x}(t))^{\intercal}\}$ *and satisfies*

$$\frac{dP(t)}{dt} = AP(t) + P(t)A^{\intercal} - P(t)C^{\intercal}R_w^{-1}CP(t) + NR_vN^{\intercal}, \quad P[0] = E\{x[0]x^{\intercal}[0]\}$$

When R_v*, and* R_w *has constant intensity,* $0 \le R_v$*,* $0 < R_w$*.*

With the theorem 2.3 in mind, one can define the FARE, which is defined by the definition 2.22.

Definition 2.22. *Filter Algebraic Riccati equation (FARE)*
Let (A, B) *be the linear state-space matrices* ($\dot{x}(t) = Ax(t) + Bu(t) + Nv(t)$)*, and (C,D) be the linear state space matrices* ($y(t) = Cx(t) + Du(t) + w(t)$)*. Where* $w(t)$*, and* $v(t)$ *are assumed to be Gaussian noise. Then* $P \in \mathrm{R}^{n\times n}$ *is a positive definite, symmetric matrix that satisfy the equation, based on the cost function equation*

(2.55). *Which by the theorem 2.3 and that the system becomes stationary,* $P(t)$ *converges, gives*

$$AP + PA^{\intercal} - (PC^{\intercal})\, R_w^{-1}\, (PC^{\intercal})^{\intercal} + NR_vN^{\intercal} = 0 \tag{2.61}$$

Then the observer gain becomes

$$L = PC^{\intercal}R_w^{-1} \tag{2.62}$$

Remark. *An important thing to notice, is that the LQ controllers are controlling in the local frame. What that means is from the fact that the linear model may be based upon the true (nonlinear) system. This gives that one have linearized around a certain point (described in the chapter 2.6). Therefore, the LQ is considering working around that point, and not necessary around the origin. LQR is therefore a local, not a global controller.*

Linear Quadratic Regulator (LQR) *Linear quadratic regulator*, LQR for short, is a state feedback controller which takes the states, and a reference as inputs, with that formulate a control signal $u(t)$. One degree of freedom, or two degrees of freedom, can be used depending if one want to consider both the feedback, and the reference individual (two d.o.f.), or together as an error (one d.o.f). To get this controller gain K, one utilizes the cost function mentioned in equation (2.54), this is more easier directly found by using one of *Riccarti equations*. Specifically the *continuous algebraic riccati equation*, CARE for short, found in definition 2.21. There Q_x, and Q_u are design parameters for the controller, costs for the states, and the input respectively. The weights are needed to have positive elements, including the zero value, $Q_x \geq 0$, and $Q_u \geq 0$. How to apply the LQR is 'quiet' strait forward, which is that it makes the input signal $u(t)$ to the system. However, there is a small difference with one, or two d.o.f, which can be illustrated with the linear state-space representation (2.31)-(2.32), (shown for clarification).

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

The input signal $u(t)$ can be formulated as either a one, or two d.o.f controller. The input signals for these cases can be observed by first that $e(t)$ is the error between the feedback of the internal states $x(t)$ and the reference $r(t)$. For two d.o.f the feedback, and reference (can be considered zero) are separately considered, which is shown by the equation (2.64).

$$\text{One d.o.f:}\ u(t) = -Ke(t) = \tag{2.63}$$

$$\text{Two d.o.f:}\ u(t) = -Kx(t) + K_r r(t) \tag{2.64}$$

For the case of one d.o.f controller the input signal is equation (2.63). Resulting in that the linear state-space representation (2.31), and (2.32), becomes the equations (2.65), and (2.66).

$$\dot{x}(t) = Ax(t) + B\left(-Kx(t)\right) = (A - BK)\,x(t) \tag{2.65}$$

$$y(t) = Cx(t) + D\left(-Kx(t)\right) = (C - DK)\,x(t) \tag{2.66}$$

Combining the LQR two d.o.f. control signal, equation (2.64), with the linear state-space system equations (2.31), and (2.32). Gives that one may write the new linear state-space system with reference as the input to the controlled system, as shown by equations (2.67), and (2.68).

$$\dot{x}(t) = (A - BK)\,x(t) + BK_r r(t) \tag{2.67}$$
$$y(t) = (C - DK)\,x(t) + DK_r r(t) \tag{2.68}$$

This gives more freedom to tune and may contribute to more time consumption in order to get a working controller. However, as one may observe is that if one where to want to formulate K_r, to just counteract the dynamic to make that if the states and the reference is equal, then $\dot{x}(t) = 0$. This gives that the gain for the reference may be calculated as equation (2.69).

$$K_r = \Big(D - (C - DK)\,(A - BK)^{-1}\,B\Big)^{-1} \tag{2.69}$$

The feedback gain is the matrix K, and if one uses a two d.o.f control. The two individual matrix multiplications are only added together to form the input signal to the system.

Linear Quadratic Integrator (LQI) *Linear quadratic integrator*, LQI for short, has a likewise properties as the LQR with an added integral. This integral action considers the error between the desired, or reference, values and the current state values. It integrates the error over time, which may improve control, such as the steady state error which may occur. To add the integral action onto the linear system, one may then augment the system with an integrated error $z(t)$, shown in equation (2.70).

$$z(t) = \int_0^t (r(\tau) - y(\tau))\,d\tau \Rightarrow \dot{z}(t) = (r(t) - y(t)) = r(t) - Cx(t) - Du(t) \tag{2.70}$$

That gives the augmented system in matrix form, with consideration to the $z(t)$ in equation (2.70). Which gives that the augmented system in equations (2.71), and (2.72).

$$\begin{bmatrix}\dot{x}(t)\\ \dot{z}(t)\end{bmatrix} = \underbrace{\begin{bmatrix}A & 0_{n\times q}\\ -C & 0_q\end{bmatrix}}_{A_{augmented}}\begin{bmatrix}x(t)\\ z(t)\end{bmatrix} + \underbrace{\begin{bmatrix}B & 0_{n\times q}\\ -D & I_q\end{bmatrix}}_{B_{augmented}}\begin{bmatrix}u(t)\\ r(t)\end{bmatrix} \tag{2.71}$$

$$y(t) = \underbrace{\begin{bmatrix}C & 0_q\end{bmatrix}}_{C_{augmented}}\begin{bmatrix}x(t)\\ z(t)\end{bmatrix} + Du(t) \tag{2.72}$$

This gives an augmented linear state-space model, a illustrative figure to show how a typical LQI system may be formulated can be seen in Figure 2.3.

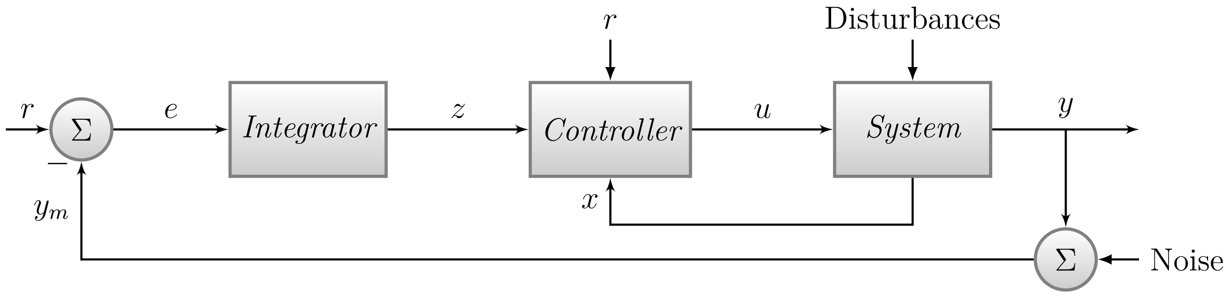


Figure 2.3: Linear Quadratic Integrator, a general negative feedback LQI system block diagram

Remark. *Observe that the Figure 2.3 is a negative feedback, it can also be defined for a positive feedback. However, then the augmented system given in the equation* (2.71)-(2.72)*, needs to be altered to have a positive feedback instead.*

The input signal to the plant, or process, seen in the Figure 2.3. Is similar to the two d.o.f LQR, equation (2.64), with an added integral action. The two d.o.f control law for the LQI may then be formulated by the equation (2.73) [33].

$$u(t) = -Kx(t) - K_z z(t) + K_r r(t) \tag{2.73}$$

This added integral action expands the states that the controller is using. Therefore one may say that it has a similar structure to an LQR, in that the input controller may be formulated by equation (2.74).

$$u(t) = -\begin{bmatrix} K & K_z \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} + K_r r(t) \tag{2.74}$$

To find this controller, is like how one finds the controller for the LQR case. With the only difference that the states are augmented with $z(t)$, the integrated error. This gives that one can use the cost function 2.54, with the augmented states instead of $x(t)$. That one needs to solve the *continuous algebraic Riccati equation* shown in equation (2.58), with the augmented states instead of $x(t)$, using $\begin{bmatrix} x(t) \\ z(t) \end{bmatrix}$. This results in that the design parameter Q_x is expanded to also embrace the new integral action. Which has been added in the augmented linear state-space system, in equations (2.71), and (2.72).

Remark. *As discussed before for the case of LQR, if one wish to just have a one d.o.f. Then one can just ignore* $K_r r(t)$ *term, and set* $x(t) = e(t)$ *to take instead account of the error between the feedback, and the reference.*

Linear Quadratic Gaussian (LQG) *Linear quadratic gaussian*, LQG for short, is a combination of a LQR, and a state observer. The state observer is typically a Kalman filter. This may be used if one cannot observe all the states, which leads to that one may not utilize a state feedback controller. That however may not be the case if one uses a state observer to estimate the states using the output of the system, with an appropriate model for estimation of the internal states. One may at first glance think, this will take time to make a combined LQR, and kalman filter, that maybe true in some respects. However, by the *Principle of separation* that is defined as, definition 2.23.

Definition 2.23. *Principle of separation*
The feedback gain of the state observer and there of the feedback controller can separately be designed.

By the *Principle of separation* definition 2.23, one can design the LQR, and the state observer separately. Giving that one can utilize the mentioned way to make a LQR controller and separately design a state observer. An general illustration block diagram of an linear quadratic Gaussian may be seen in the Figure 2.4.

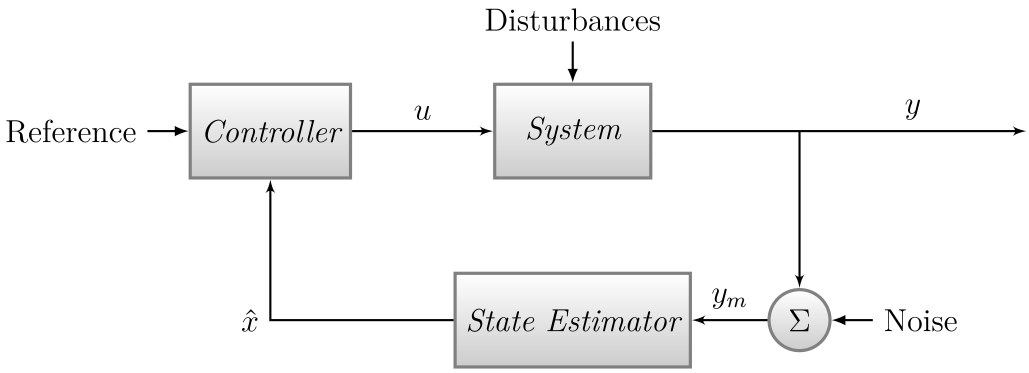


Figure 2.4: Linear Quadratic Gaussian, a general LQG system block diagram

To make the state observer one may first consider its state-space, which is used to estimate the states. Using the real systems output $y(t)$, then the state-space is described as equations (2.75), and (2.76).

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L\left(y(t) - \hat{y}(t)\right) \tag{2.75}$$
$$\hat{y}(t) = C\hat{x}(t) \tag{2.76}$$

$\hat{x}$ is the estimated states, and $\hat{y}$ the estimated output, with L as the observer gain. This state estimation may be augmented together with the linear state-space, equations (2.31), and (2.32). The error between the estimated and true states, is expressed as $\tilde{x}(t) = x(t) - \hat{x}(t)$. This gives that the augmented system becomes as shown by the equations (2.77), and (2.78).

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\tilde{x}}(t) \end{bmatrix} = \begin{bmatrix} A - BK & BK \\ 0_{n\times q} & A - LC \end{bmatrix} \begin{bmatrix} x(t) \\ \tilde{x}(t) \end{bmatrix} + \begin{bmatrix} BK_r \\ 0_{n\times q} \end{bmatrix} r(t) \tag{2.77}$$
$$y(t) = \begin{bmatrix} C & 0_n \end{bmatrix} \begin{bmatrix} x(t) \\ \tilde{x}(t) \end{bmatrix} \tag{2.78}$$

An advantage of using a Kalman filter is that one may account for noise that may occur. This gives a sort of extended linear state-space, with noise.

$$\dot{x}(t) = Ax(t) + Bu(t) + Nv(t) \tag{2.79}$$
$$y(t) = Cx(t) + Du(t) + w(t) \tag{2.80}$$

$v(t)$, and $w(t)$ are assumed to be white Gaussian noise. Then one may use the FARE, definition 2.22, to make the state estimator. That defines a observer gain L using the state observer state-space (2.77)-(2.78) and the linear system which it is trying to control is based on the linear state-space (2.79)-(2.80). Then one have the estimated states $\hat{x}(t)$ which can be used with a LQR, or LQI, to make the control signal $u(t)$. That may be defined as one- or two d.o.f. controller, equations (2.81), and (2.82).

$$u(t) = -K\hat{x}(t) + K_r r(t) \tag{2.81}$$

$$u(t) = -K\hat{x}(t) - K_z z(t) + K_r r(t) \tag{2.82}$$

The controller gain K is calculated as mentioned before, for the LQR, or LQI cases. It is to be noted that the names may become different, there if one uses an LQR and a state estimator, this gives an LQG. While an LQI and a state estimator gives a LQGI.

Remark. *A point to add is that LQG is often compared to be a LQR with a Kalman filter, which is in itself true. However, it is to be noted that may not always be the case, because one may also as the LQI add an integral action to the LQR. Which is then an LQG that is made from a LQI and a Kalman filter, and may be named as Linear Quadratic Gaussian Integral (LQGI).*

2.9.1.3 $\mathcal{H}_2$ and $\mathcal{H}_\infty$ Control

$\mathcal{H}_2$ and $\mathcal{H}_\infty$ control utilizes norms function for matrices. The matrix norms are called $\mathcal{H}_2$ *system norm*, and $\mathcal{H}_\infty$ *system norm*. An observation is that the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control have similarities to the linear quadratic controllers. Which one may observe by the definition of the general vector norms in definition 2.14, shown in the equation (2.44).

Remark. *A point to make out is that even though there are similarities with the linear quadratic controllers, and these controllers. It should be notice that LQR $\iff$ $\mathcal{H}_2$ state feedback, that does not mean that LQG is equivalent to $\mathcal{H}_2$ output feedback, (LQG $\not\iff$ $\mathcal{H}_2$ output feedback). LQG, and $\mathcal{H}_2$ control is only equivalent for a specific special case of the generalized output feedback $\mathcal{H}_2$.*

Similarities with LQ - Control It is of some interest to observe the similarities as explained in some degree in the remark above. This may be observed through how one find the controllers of LQ, specifically CARE and FARE shown in definitions 2.21, and 2.22, respectively. That gives a slightly modified definition, by that the cost function becomes modified. Which can be seen in the definition of MCARE definition 2.24, and MFARE definition 2.25.

Definition 2.24. *Modified Continuous Algebraic Riccati equation (MCARE)*
Let (A, B_1, B) be the linear state space matrices $\dot{x}(t) = Ax(t) + B_1 d(t) + Bu(t)$, and $P \in \mathrm{R}^{n\times n}$ is a positive definite, symmetric matrix that satisfy the equation, based on the cost function equation

$$J(x,u,d) = \frac{1}{2}\int_0^\infty \left(x^\intercal(\tau)Q_x x(\tau) + u^\intercal(\tau)Q_u u(\tau) + 2x^\intercal(\tau)Nu(t) - \gamma d^\intercal(\tau)d(\tau)\right)d\tau$$

Alternatively, if the matrix N is an zero matrix, then the cost function would become as shown by equation (2.83)*.*

$$J(x,u,d) = \frac{1}{2}\int_0^\infty \left(x^\intercal(\tau)Q_x x(\tau) + u^\intercal(\tau)Q_u u(\tau) - \gamma d^\intercal(\tau)d(\tau)\right) d\tau \tag{2.83}$$

Then using the cost function, equation (2.83) *and $P \in \mathrm{R}^{n\times n}$ is a positive definite, symmetric matrix that satisfy the equation* (2.84)*.*

$$PA + A^\intercal P + Q_x - P\left(BQ_u^{-1}B^\intercal - \frac{1}{\gamma^2}B_1B_1^\intercal\right)P = 0 \tag{2.84}$$

In which the design parameters that needs to be chosen, or found is Q_x, Q_u, and γ. There the best control input signal $u(t)$, and the worst disturbance $d(t)$ is then given by the equations (2.85)*, and* (2.86)*.*

$$u(t) = -\,Q_u^{-1}B^\intercal Px(t) = -Kx(t) \tag{2.85}$$

$$d(t) = \frac{1}{\gamma^2}B_1^\intercal Px(t) = K_d x(t) \tag{2.86}$$

Definition 2.25. *Modified Filter Algebraic Riccati equation (MFARE)*
Let (A, B) be the linear state space matrices ($\dot{x}(t) = Ax(t) + B_1 d(t)$), and (C,D) be the linear state space matrices ($y(t) = C_2 x(t) + D_{2,1}d(t)$). There $d(t)$ is a deterministic disturbance, and supposed that the (A, C_2) pair is detectable and (A, B_1) is stabilizable. Then $P_F \in \mathrm{R}^{n\times n}$ is a positive definite, symmetric matrix that satisfy the equation, based on the cost function equation (2.55)*. Which then by the theorem 2.3, also that the system becomes stationary, and $P(t)$ converges, giving equation* (2.87)

$$.P_F A^\intercal + AP_F + P_F\left(\gamma_F^{-2}I_n - C_2^\intercal C_2\right)P_F + B_1B_1^\intercal = 0 \tag{2.87}$$

Then the observer gain becomes

$$L = -P_F C_2^\intercal \tag{2.88}$$

The similarities with the LQ case for the modified versions of CARE and FARE, can be listed to be

- MCARE: $\mathcal{H}_\infty$ state-feedback
- MFARE: $\mathcal{H}_\infty$ state filter

That may be observed if comparing the definitions of the MCARE (definition 2.24), and MFARE (definition 2.25). With how one may numerically compute the $\mathcal{H}_\infty$, by the Hamiltonian matrix shown in equation (2.48).

$\mathcal{H}_2$ - Controller The control objective that $\mathcal{H}_2$ optimal control problem is trying to do, is to find a controller K which minimizes the $\mathcal{H}_2$ system norm of the *lower LFT* (definition 2.18), which may be written as $\min \|N\|_2$.

$\mathcal{H}_\infty$ - Controller The control objective that $\mathcal{H}_\infty$ optimal control problem is trying to do, is to find a controller K which minimizes the $\mathcal{H}_\infty$ system norm of the *lower LFT* (definition 2.18), which may be written as $\min \|N\|_\infty$.
Remark. *If one desire to achieve γ_{min} (with some tolerance), then one may use γ-iteration, where one iteratively improves the γ value. It should also be noted that $\mathcal{H}_\infty$ can be structured, like μ that will be explained next.*

2.9.1.4 μ-synthesis (*DK*-iteration)

The μ-synthesis is a control synthesis which utilizes the *structured singular value μ* (other short notation names are SSV, mu, and Mu). The structured singular value μ is as *Skogetad and Postlethwaite* wrote: "a very powerful tool for the analysis of RP with a given controller"[34]. RP, or robust performance is a criterion with may be hard to achieve deepening on the system, and controller. Therefore μ-synthesis is of interest because it considers RP. If one can get a controller that consider to get a good robust performance, then other criteria will follow, such as RS, robust stability. To fully embrace the μ-synthesis, first one consider the μ-analysis, more specific what the meaning of *structured singular value μ* is. Then proceed to utilize μ to make a controller for the system. However, a point to note is at the current time there is no direct method to synthesize a μ-optimal controller. There is however, a method named DK-iteration, for complex perturbations, which can be used. [34] This is the method for the μ-synthesis that will be mentioned here.

Structured Singular Value (μ) The structured singular value, μ, may be formulated with that it "is a function which provides a generalization of the singular value, $\bar{\sigma}$, and the spectral radius, ρ."[34]. It may also be formulated, "Conceptually, the structured singular value is nothing but a straightforward generalization of the singular values for constant matrices."[36]. What this implies is that that the μ, is in its core a singular value, with a structure to it. This structure for this case is the uncertainties, structured uncertainties, which was explained earlier in chapter 2.9.1.3 and share properties with the spectral radius. The singular value, and the spectral radius, are defined as definitions 2.26, and 2.27.

Definition 2.26. *Singular Value*
Let G be a complex matrix, and G^H is the complex conjugate transpose of G. Then the singular value are the positive square roots of the eigenvalues of $G^H G$. [34]

$$\sigma_i(G) = \sqrt{\lambda_i(G^H G)} \tag{2.89}$$

Remark. *The maximum, and minimum singular values are indicated by a bar on its top or bottom. The maximum singular value, $\bar{\sigma}$, the minimum singular value, $\underline{\sigma}$.*

Definition 2.27. *Spectral Radius*
The eigenvalues are sometimes called characteristic gains. The set of eigenvalues of a matrix A is called the spectrum of A. The largest of the absolute values of the eigenvalues of A is the spectral radius of A. [34]

$$\rho(A) \triangleq \max_i |\lambda_i(A)| \tag{2.90}$$

Now one may wonder how this μ is defined, then a nicely and short statement. "*Find the smallest structured Δ (measured in terms of $\bar{\sigma}(\Delta)$) which makes the matrix $I - M\Delta$ singular; then $\mu(M) = 1/\bar{\sigma}(\Delta)$*"[34]. The concept that the structural singular value is based upon, and how it works, can be described by the *small gain theorem*, which it is shown in theorem 2.4.

Theorem 2.4. *Small Gain Theorem*
Consider a system with a stable loop transfer function $L(s)$. Then the closed-loop system, the Figure 2.5, is stable from $\{d_1, d_2\}$ to $\{y_1, y_2\}$ if

$$\|L(j\omega)\| < 1, \forall\omega \tag{2.91}$$

$\|L\|$ denotes any matrix norm satisfying $\|AB\| \leq \|A\| \cdot \|B\|$ [34]. That is $\{d_1, d_2\}$ have to be bounded.

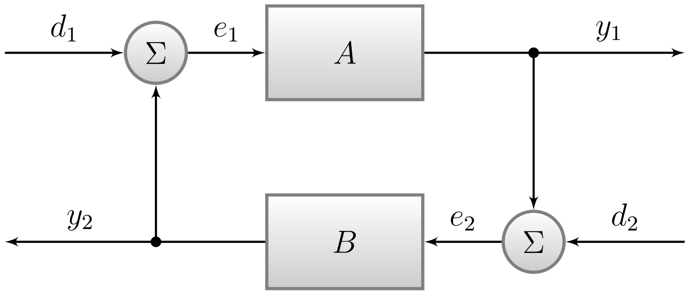


Figure 2.5: Block diagram illustration of the *small gain theorem*.

Then the formulation of the structural singular value can be made through if one consider the *small gain theorem*, theorem 2.4. That is by considering the Figure 2.5, and say that $A = \Delta$, and $B = M$. This gives that the main question can be formulated to how large Δ can be in the sense of $\|\Delta\|_\infty$ (infinity norm). That does not turn the feedback system $(I - M\Delta)$ singular, in other words, destabilizes the feedback system. The feedback system can be described by $(I - M\Delta)$. Then one may find that the system becomes singular if the determinant of the system becomes equal to zero ($\det(I - M\Delta) = 0$), that is because it describes the closed-loop poles. Now for the main point to observe is that the *small gain theorem*, theorem 2.4, says that the two interconnected systems, needs to have the term $\|A\|_\infty \|B\|_\infty < 1$ fulfilled. This givens that for this system the condition becomes $\|\Delta\|_\infty \|M\|_\infty < 1$, if one assumes that for the Δ block, $\|\Delta\|_\infty < \alpha$, with that $\alpha > 0$, that is sufficiently small, so that the closed-loop system is stable. One may find a maximum for the α, where the closed-loop system becomes singular (unstable), namely α_{max}. Which gives by the *small gain theorem*, theorem 2.4.

$$\frac{1}{\alpha_{max}} = \|M\|_\infty := \sup_{s \in \bar{C}_+} \bar{\sigma}(M(s)) = \sup_{\omega} \bar{\sigma}(M(j\omega)) \tag{2.92}$$

Remark. *Clarification, the 'sup' in the equation* (2.92)*, is called the supremum. Which may be explained by saying that for a certain subset S of real numbers, the largest element of all elements in the subset S is the supremum. There it is also referred to as the least upper bound.*

A notable thing is that the α_{max} is the robust stability margin [36]. It is to be noted that Δ is not specified, that is because it can be unstructured or structured. However, even though the definitions are similar in expression, there is difference. The structured way is the representation for the μ, as the name also implies, which may be formulated for any fixed $s \in \bar{C}_+$, when Δ is structured.

$$\bar{\sigma}\left(M\left(s\right)\right) = \frac{1}{\min\{\bar{\sigma}\left(\Delta\right) : \det\left(I - M\Delta\right) = 0, \Delta \text{ is structured}\}} = \mu\left(M\left(s\right)\right) \quad (2.93)$$

This shows that μ is interpreted as the largest structured singular value of $M(s)$, when Δ is structured, more specifically, structured uncertainty. Which follows the definitions of both singular values, and spectral radius, definitions 2.26, and 2.27 accordingly. This results in that one may define the structured singular value μ as told in the definition 2.28.

Definition 2.28. *Structured Singular Value, μ*
For $M \in \mathcal{C}^{n\times n}$, $\mu_{\mathbf{\Delta}}\left(M\right)$ is defined as

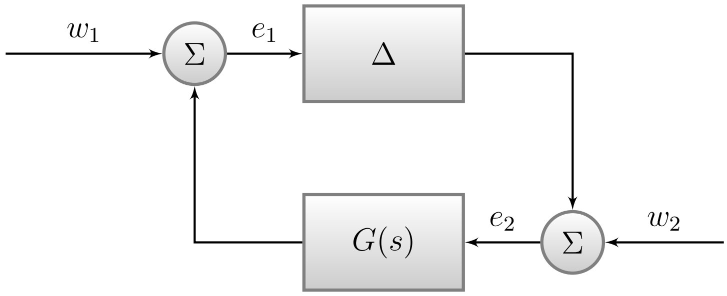
$$\mu_{\mathbf{\Delta}}\left(M\right) := \frac{1}{\min\{\bar{\sigma}\left(\Delta\right) : \Delta \in \mathbf{\Delta}, \det\left(I - M\Delta\right) = 0\}} \quad (2.94)$$

Unless no $\Delta \in \mathbf{\Delta}$ makes $I - M\Delta$ singular, in which case $\mu_{\mathbf{\Delta}} := 0$.[36]
Remark. *To not be confused with the terminology of the definition of μ. It is to notice that in comparison with the deviation to $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control algorithms, with is based on a norm, μ is not a norm.*

There one may use μ to get necessary and sufficient conditions for both RS (robust stability), and RP (robust performance). Which can be described by the theorems 2.5, and 2.6.

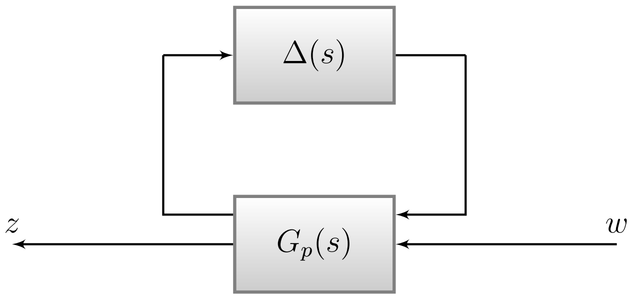
Theorem 2.5. *Structured Robust Stability*
Let $\beta > 0$. The loop shown below is well-posed and internally stable for all $\Delta\left(\cdot\right) \in \mathcal{M}\left(\mathbf{\Delta}\right)$ with $\left\|\Delta\right\|_\infty < \frac{1}{\beta}$ if and only if [36]

$$\sup_{\omega\in\mathbf{R}} \mu_{\mathbf{\Delta}}\left(G\left(j\omega\right)\right) \leq \beta$$



Theorem 2.6. *Structured Robust Performance*
Let $\beta > 0$. For all $\Delta\left(s\right) \in \mathcal{M}\left(\mathbf{\Delta}\right)$ with $\left\|\Delta\right\|_\infty < \frac{1}{\beta}$, the loop shown below is well-posed, internally stable, and $\left\|F_u\left(G_p, \Delta\right)\right\|_\infty \leq \beta$ if and only if [36]

$$\sup_{\omega\in\mathbf{R}} \mu_{\mathbf{\Delta}_P}\left(G_p\left(j\omega\right)\right) \leq \beta.$$



Using the two theorems 2.5, and 2.6, one may observe that μ-synthesis, and μ-analysis may express itself to get RS (robust stability), and RP (robust performance). Depending if the value μ is sufficient for these criteria, which is the idea behind the μ-synthesis and will be described how to do.

Remark. *It has been mentioned two different names with μ, namely μ-synthesis, and μ-analysis. To clarify μ-analysis is when one analyzes a system, and an existing controller with the help of structured singular value. While μ-synthesis is when one uses structured singular value to make a controller.*

It is not totally obvious what values that μ needs to have in order to be 'good', or 'bad'. If $\mu = 1$ then there exists a perturbation with $\bar{\sigma}(\Delta) = 1$, that makes the matrix $I - M\Delta$ singular. Based on this value, one can then get a feeling that larger value of μ gives that the matrix $I - M\Delta$ becomes singular with smaller perturbations. While smaller values of μ gives that $I - M\Delta$ becomes singular with larger perturbations [34].

***DK*-iteration** As mentioned before because there are no presents of a method that is directly synthesizes a μ-optimal controller. Since it is difficult to explore all of Δ, and that it is a non-convex problem. One may use the DK-iteration method that combines the $\mathcal{H}_\infty$ synthesis, and μ-analysis to utilize the μ in its synthesis of a controller. The DK-iteration is an iterative synthesis, it more or less repeats three steps, until satisfactory performance is achieved. Such as $\|DN(K)D^{-1}\|_\infty < 1$, when the $\mathcal{H}_\infty$ norm no longer decreases or within some tolerance. To formulate the DK-iteration two building blocks needs to be studies, which is used in the DK-iteration. One of these is how the upper bound on μ in terms of the scaled singular value is defined.

Definition 2.29. *Improved Upper Bound*
Define $\mathcal{D}$ to be the set of matrices D which commute with Δ (i.e. satisfy $D\Delta = \Delta D$). Then it follows from the properties of μ for complex perturbation, ($\mu(DM) = \mu(MD)$ and $\mu(DMD^{-1}) = \mu(M)$),($\rho(M) \leq \mu(M) \leq \bar{\sigma}(M)$) that [34]

$$\mu(M) \leq \min_{D \in \mathcal{D}} \bar{\sigma}\left(DMD^{-1}\right) \tag{2.95}$$

The last building block that is needed to define the DK-iteration, is the definition in which to find the controller. This controller minimizes the peak value over frequency of this upper bound, definition 2.29, that then is defined by the equation (2.96).

$$\min_K \left(\min_{D \in \mathcal{D}} \left\| DN(K)D^{-1} \right\|_\infty \right) \tag{2.96}$$

This leads to the formulation of the DK-iteration, that first starts by selecting an initial stable rational transfer matrix $D(s)$. It is appropriate to have a good initial guess of the transfer matrix $D(s)$. However, it is not always necessary because a good initial choice given that the system is reasonably scaled for performance, is to have the initial choice of D as the identity matrix [34]. Then the DK-iteration procedure can be explained by these three steps [34]

1. *K-step:* Synthesize a $\mathcal{H}_\infty$ controller for the scaled problem of the equation (2.96), with the transfer matrix $D(s)$ fixed to a certain value. This should give the combined controller, and system, named N.
   Before continuing one needs first to consider if the stop criteria is reached, this criteria may wary (depending on tolerance, and considered criteria), however, if the criteria is reached, stop, otherwise continue.
2. *D-step:* Find an upper bound that is defined in definition 2.29, with the equation (2.95). One want to find $D_{new}(j\omega)$ that minimizes at each frequency, with a fixed N (Note, fit only in magnitude).
3. For each of the elements in $D_{new}(j\omega)$: Fit the element's magnitude to a stable and minimum-phase transfer function $D(s)$.

Remark. *A point to make about the DK-iteration, is that it is a fundamental problem with this approach. Which is that although each of the minimization steps (K-step, and D-step) are convex, the joint convexity can not be guaranteed. However, the iterations may converge to a local optimum [34], which is occurring because of the increase in order of* $D(j\omega)$, $(\deg(K_{dk_\infty}(j\omega)) << \deg(K_{dk}(i\omega)))$.

2.9.2 Nonlinear Control Algorithms

The evaluated nonlinear control algorithms will be described and formulated here. Compared to the linear controllers, these controllers directly consider the nonlinear system, instead of the linearized one. It is to be noted that these nonlinear controllers are more complex, and because of that, only the main parts of these controllers will be gone through. For the feedback linearization types of nonlinear controllers, there is a certain condition that needs to be hold, which is defined in the definition 2.30.

Definition 2.30. *From the nonlinear systems in equations* (2.22)*, and* (2.23)*. The input affine nonlinear system becomes*

$$\dot{x}(t) = f(x) + g(x)u(t) \tag{2.97}$$

$$y(t) = h(x) \tag{2.98}$$

There $f : D \to R^n$ *and* $g : D \to R^{n\times p}$ *are sufficiently smooth (all the partial derivatives, that will appear later on, are defined and continuous) on a domain* $D \subset R^n$*. Is said to be feedback linearizable (or input-state linearizable) if there exists a diffeomorphism* $T : D \to R^n$ *such that* $D_z = T(D)$ *contains the origin and the change of variables* $z = T(x)$ *transforms the system* **??** *into the form*

$$\dot{z}(t) = A_c z(t) + B_c\gamma(x)\left[u(t) - \alpha(x)\right] \tag{2.99}$$

With (A_c, B_c) *controllable and* $\gamma(x)$ *non-singular for all* $x \in D$ *[32]. In our case the* A_c*, and* B_c *are in controller canonical form. However, the matrices can just be controllable.*

An illustrative figure of how a feedback linearization may looks like with the non-linear system defined as in equations (2.97), and (2.98), can be seen in the Figure 2.6.

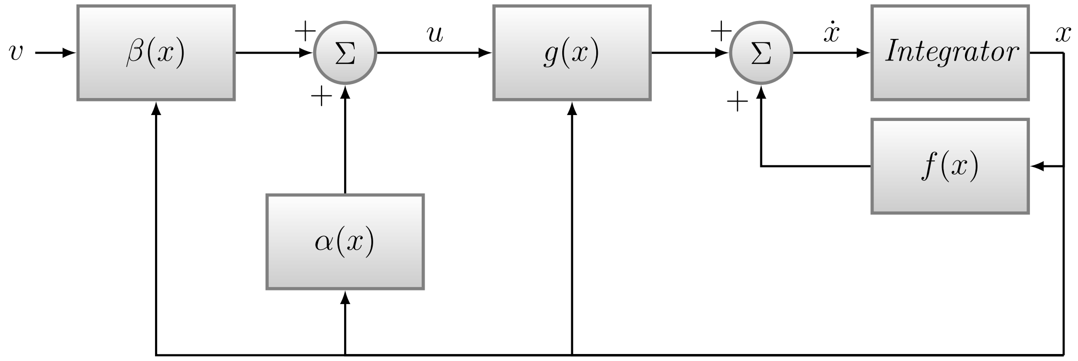


Figure 2.6: Illustrative figure of a feedback linearization of a nonlinear system.

v is the new input to the system, $f(x)$ and $g(x)$ is the nonlinear system. With the feedback linearization part of β, and α, would aim to counteract the non-linearity of the system, to make the system look linear. This gives that one may design a controller of the feedback linearized system instead, with looks linear rather than the nonlinear system. However, it is to be noted that the Figure 2.6, is missing the output $h(x)$, which will be explained with some methods of feedback linearization.

2.9.2.1 Feedback Linearization

The two considered *feedback linearization* forms that will be considered, and described is the *Input to output linearization*, and *Input to state linearization. Input to output linearization* is a sort of feedback linearization that takes the output into consideration. Its similar controller *input to state Linearization* which takes the state into consideration instead. The *feedback linearization* can be formulated as a single input single output, (SISO) system, and multiple inputs multiple outputs (MIMO). However, it is to be noted that to solve for MIMO systems there is much larger difficulties and does not directly have a general method to solve for MIMO systems. While for SISO systems there exist a general approach for the *input to output linearization*, and *input to state linearization.* For SISO systems can be explained for a nonlinear system in the form of the equations (2.97), and (2.98), shown for convenience.

$$\begin{aligned}\dot{x}(t) =& f(x) + g(x)u(t)\\ y(t) =& h(x)\end{aligned}$$

$f(x)$, $g(x)$, and $h(x)$ are sufficiently smooth functions as described earlier by the definition 2.30, then the time derivative of $y(t)$, named $\dot{y}(t)$ becomes as shown in equation (2.100).

$$\begin{aligned}\dot{y}(t) =& \frac{\partial h(t,x(t))}{\partial t} = \frac{\partial h(t,x(t))}{\partial x}\dot{x}(t) = \frac{\partial h}{\partial x}\left[f(t,x(t)) + g(t,x(t))u(t)\right]\\ =& \frac{\partial h}{\partial x}f(t,x(t)) + \frac{\partial h}{\partial x}g(t,x(t))u(t) = \mathcal{L}_f h(t,x(t)) + \mathcal{L}_g h(t,x(t))u(t)\end{aligned} \tag{2.100}$$

$$\implies \mathcal{L}_f h(t,x(t)) = \frac{\partial h}{\partial x} f(t,x(t)), \quad \mathcal{L}_g h(t,x(t)) = \frac{\partial h}{\partial x} g(t,u(t))u(t)$$

$\mathcal{L}_f$ and $\mathcal{L}_g$ are called the *lie derivatives*, with respect to f or g respectively. It can be said that the *Lie derivatives* takes the derivatives of what is in front of it, with respects of the states. For better clarity an example of the *Lie derivatives*, with definition, can be seen in the definition 2.31. Which *Lie algebra* makes it transparent.

Definition 2.31. *Lie Derivative Example's*
Lie Derivative's are defined as such [32]

$$\mathcal{L}_f h(t,x(t)) = \frac{\partial h}{\partial x} f(t,x(t)), \quad \mathcal{L}_g h(t,x(t)) = \frac{\partial h}{\partial x} g(t,u(t))u(t) \tag{2.101}$$

Which is shown by the results from the equation (2.100)*. This definition may be explained by these example's*

$$\begin{aligned}
\mathcal{L}_g \mathcal{L}_f h(t,x(t)) =& \frac{\partial \mathcal{L}_f h(t,x(t))}{\partial x(t)} g(t,x(t)\\
\mathcal{L}_f^2 h(t,x(t)) =& \mathcal{L}_f \mathcal{L}_f h(t,x(t)) = \frac{\partial \mathcal{L}_f h(t,x(t))}{\partial x(t)} f(t,x(t))\\
\mathcal{L}_f^k h(t,x(t)) =& \mathcal{L}_f \mathcal{L}_f^{k-1} h(t,x(t)) = \frac{\partial \mathcal{L}_f^{k-1} h(t,x(t))}{\partial x(t)} f(t,x(t))\\
\mathcal{L}_f^0 h(t,x(t)) =& h(t,x(t))
\end{aligned}$$

The *Lie derivatives* are convenient to use when one have repeated calculations of the derivatives with respect to the same vector field or a new one [32]. Using the *Lie derivatives* one may obtain the *relative degree* of the system, what the relative degree is may be defined by the definition 2.32.

Definition 2.32. *Relative Degree*
The nonlinear system (2.97)-(2.98) *is said to have relative degree ρ, $1 \leq \rho \leq n$, in a region $D_0 \subset D$ if*

$$\mathcal{L}_g \mathcal{L}_f^{i-1} h(t,x(t)) = 0, i = 1,2,\ldots,\rho-1; \quad \mathcal{L}_g \mathcal{L}_f^{\rho-1} h(t,x(t)) \neq 0 \tag{2.102}$$

For all $x \in D_0$ [32].

To calculate the *relative degree* one does not necessary need to use the *Lie derivatives*. As shown in the definition 2.32, one may also use the derivatives of the output y. That is because as one may notice from the equations (2.100), and (2.101). Is that if the output of the SISO system is shown for a certain derivative of $y^{(i)}$, for $0 \leq i \leq n$. Then that number n is the *relative degree*, as can be observed from the calculation shown in equation (2.103).

$$\begin{aligned} y =& h(t, x(t)) = \mathcal{L}_f^{(0)} h(t, x(t)) \\ y^{(1)} =& \frac{\partial h(t, x(t))}{\partial t} = \mathcal{L}_f^{(1)} h(t, x(t)) \\ \vdots& \\ y^{(i-1)} =& \mathcal{L}_f^{(i-1)} h(t, x(t)) \\ y^{(i)} =& \mathcal{L}_f^{(i)} h(t, x(t)) + \mathcal{L}_g \mathcal{L}_f^{(i-1)} h(t, x(t)) u(t) \end{aligned} \tag{2.103}$$

Which shows that one may get the relative degree by just derive $y = h(t, x(t))$. Then the *relative degree* becomes equal to the number of times one could derive until a input is showing in the derivative, following the definition 2.32, $\rho = n$. If the *relative degree* of the nonlinear system is known, one may start consider how to formulate the *input to output linearization*. As shown by the condition definition 2.30, a change of variables is to be formulated. This transformation will have to be a *diffeomorphism* transformation, by the definition 2.30. Also, that the transformation does not inflict difficulties to the calculation. One may use *Lie derivatives* to formulate this transformation, there one may obtain the transformation by the theorem 2.7.

Theorem 2.7. *Consider the system* (2.97)-(2.98)*, and suppose it has relative degree* $\rho \leq n$ *in* D*. If* $\rho = n$*, then for every* $x_0 \in D$*, a neighborhood* N *of* x_0 *exists such that the map*

$$T(x) = \begin{bmatrix} \mathcal{L}_f^{(0)} h(t, x(t)) \\ \mathcal{L}_f^{(1)} h(t, x(t)) \\ \vdots \\ \mathcal{L}_f^{(n-1)} h(t, x(t)) \end{bmatrix} \tag{2.104}$$

Restricted to N*, is a diffeomorphism on* N*. If* $\rho < n$*, then for every* $x_0 \in D$*, a neighborhood* N *of* x_0 *and smooth functions* $\phi_1(x), \ldots, \phi_{n-\rho}(x)$ *exist such that equation* (2.107) *is satisfied for all* $x \in N$ *and the map* $T(x)$ *of equation* (2.106)*, restricted to* N*, is a diffeomorphism on* N *[32].*

Another way to formulate the map $T(x)$ is to use the derivatives of $y = h(t, x(t))$. As one may observe that they have similar structure to how the transformation $T(x)$ is formulated, if one compares the two equations (2.103), and (2.104). Which then gives that the transformation $T(x)$ can be formulated as equation (2.105).

$$T(x) = \begin{bmatrix} \mathcal{L}_f^{(0)} h(t, x(t)) \\ \mathcal{L}_f h(t, x(t)) \\ \vdots \\ \mathcal{L}_f^{n-1} h(t, x(t)) \end{bmatrix} = \begin{bmatrix} y \\ y^{(1)} \\ \vdots \\ y^{(i-1)} \end{bmatrix} \tag{2.105}$$

Which is the transformation used for *input to state linearization*, that is when the *relative degree* is equal to the number of states. However, if the relative degree is not equal to the number of states, *input to output linearization* may be used. Then a more general transformation may be expressed as equation (2.106).

$$z = T(x) = \begin{bmatrix} \phi_1(x) \\ \vdots \\ \phi_{n-\rho}(x) \\ \mathcal{L}_f^{(0)} h(t, x(t)) \\ \mathcal{L}_f h(t, x(t)) \\ \vdots \\ \mathcal{L}_f^{n-1} h(t, x(t)) \end{bmatrix} \triangleq \begin{bmatrix} \phi(x) \\ \psi(x) \end{bmatrix} = \begin{bmatrix} \eta \\ \xi \end{bmatrix} \tag{2.106}$$

$\phi_1(x)$ to $\phi_{n-\rho}(x)$ are chosen such that $T(x)$ is *diffeomorphism* on a domain $D_0 \subset D$ and that the condition, equation (2.107), is fulfilled.

$$\frac{\partial \phi_i}{\partial x} g(t, x(t)) = 0, \text{ for } 1 \leq i \leq n - \rho, \forall x \in D_0 \tag{2.107}$$

The new state model following from the nonlinear system (2.97)-(2.98) with the variable transformation given in equation (2.106), and following the definition 2.30, gives that the new state model becomes

$$\dot{\eta} = f_0 (\eta, \xi) \tag{2.108}$$

$$\dot{\xi} = A_c \xi + B_c \underbrace{\gamma(x) [u - \alpha(x)]}_{v} \tag{2.109}$$

$$y = C_c \xi \tag{2.110}$$

There $\xi \in \mathbf{R}^{\rho}, \eta \in \mathbf{R}^{n-\rho}$, $u = \alpha(x) + \beta(x)v$, $\gamma(x) = \beta^{-1}(x)$, and (A_c, B_c, C_c) is matrices that have canonical form, representation of chain of ρ integrals. The terms in the state-space model (2.108)-(2.110), is as given in the equations (2.111), and (2.112).

$$f_0 (\eta, \xi) = \frac{\partial \phi}{\partial x} f(x) \bigg|_{x = T^{-1}(z)} \tag{2.111}$$

$$\gamma(x) = \beta^{-1}(x) = \mathcal{L}_g \mathcal{L}_f^{\rho-1} h(x), \text{ and } \alpha(x) = -\frac{\mathcal{L}_f^{\rho} h(x)}{\mathcal{L}_g \mathcal{L}_f^{\rho-1} h(x)} \tag{2.112}$$

A thing to take into consideration is the unobservable states that one's controller may inflict to the internal dynamics. These internal dynamics are described in the equation (2.108), in which by setting that $\xi = 0$, one then gets the so called *zero dynamics*,equation (2.113).

$$\dot{\eta} = f_0 (\eta, 0) \tag{2.113}$$

If this *zero dynamics*, equation (2.113), is asymptotically stable then it is said that the system is said to be of *minimum phase*. Then if no *zero dynamics* exists, then the system is said to be *minimum phase* [32]. However, if unstable *input to output linearization* may not work [37].

2.9.2.2 Feedback Linearization Control

From the different ways to define a feedback linearization, given in chapter 2.9.2.1. Gives a way to make the system as described in the beginning of the chapter 2.9.2, look linear like, which one then may use linear control algorithms to control the feedback linearized system. However, some problems may occur, such as that the *zero dynamics* might become difficult to investigate. This will result in that the control of the system is not guaranteed, because the *zero dynamics*, or the unobserved states, may inflict unknown effects upon the system being controlled. Also, because controlling a nonlinear system that has been feedback linearized, one may also think about what type of solution one will get, namely a global, or local one. There some pole cancellation of non-minimum phase may have occurred in the process, and other possibilities also exists. Therefore it may be difficult to make a working feedback linearization, and controller. One way to solve this problem with that the *zero dynamics* may be to hard to solve, is to use "linear output selection for feedback linearization" [37]. That is going to be used for the nonlinear control algorithm, and may solve the difficult task of ensuring that the *zero dynamics* is at least locally asymptotically stable.

Feedback Linearization - Linear Output Selection As described in the article "linear output selection for feedback linearization" [37], it can be hard to ensure that the *zero dynamics* is stable. Therefore, they are saying that if one uses a linear output selection, an LQR as the output $y(t) = h(x) = Kx(t)$. Then one can obtain a *relative degree* of the open loop to one, and then at the desired operating, or stationary point. The *zero dynamics* at that point and in a neighborhood is locally (or globally) asymptotically stable. They use a feedback linearized system, with the output $y(t)$ as an LQR, that results in that one gets that $y(t) = h(x) = Kx(t)$ as can be seen in the nonlinear state-space realization (2.114)-(2.115).

$$\dot{x}(t) = f(x) + g(x)u(t) \tag{2.114}$$
$$y(t) = h(x) = Kx(t) \tag{2.115}$$

Following feedback linearization procedure one gets that the input $u(t)$ to the system dynamics, can be written as the equation (2.116).

$$u(t) = \alpha(x) + \beta(x)v(t) \tag{2.116}$$

They use a linear feedback gain for parameterizing the controller, however one can also use another linear controller. As shown in the Figure 2.6, with that the output is determined by the LQR output selection, K. With a controller that controls the feedback linearized system with output selection. The controller is giving $v(t)$ as an output to be used as the input of the system, as is shown in the Figure 2.7.

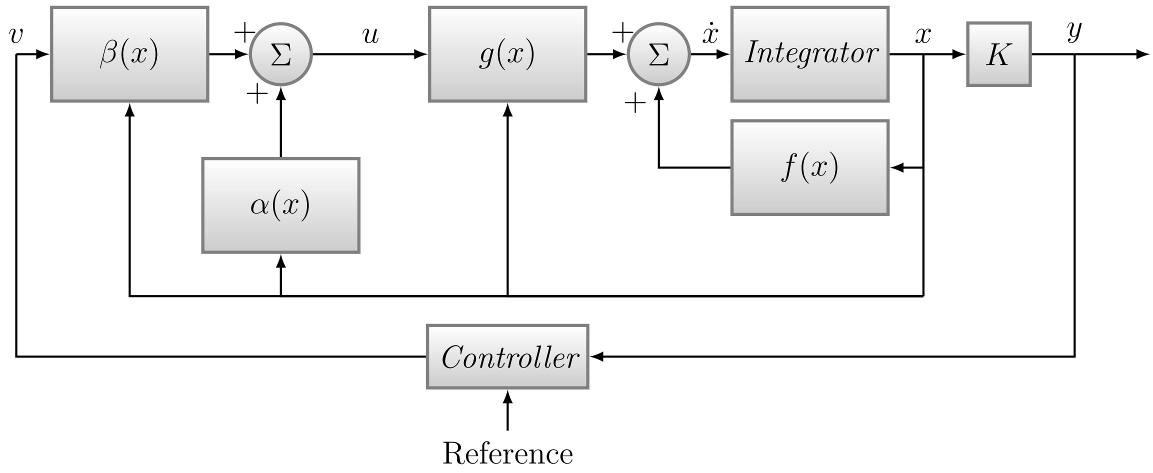


Figure 2.7: Illustrative figure of the feedback linearized system, with linear output selection of K, and a controller.

In Figure 2.7 the controller can be either a one d.o.f controller or a two d.o.f controller. While how to set the $\alpha(x)$, and $\beta(x)$ can be found in the description of the two feedback linearization ways, shown in equation (2.112). Using those equations with that the output is the output selection K, $y(t) = h(x) = Kx(t)$. Gives that the $\alpha(x)$, and $\beta(x)$ can be formulated by the equation (2.117), and (2.118).

$$\alpha(x) = -\frac{\mathcal{L}_f h(x)}{\mathcal{L}_g h(x)} = -\frac{K_1 f(x)}{K_1 g(x)} \tag{2.117}$$

$$\beta(x) = \gamma(x)^{-1} = \frac{1}{\mathcal{L}_g h(x)} = \frac{1}{K_1 g(x)} \tag{2.118}$$

What to set the design parameters of the LQR, they [37] use cheap control policy for the output selection. It is to observe that the controller doesn't become too fast, as it may then potentially drift away from the local controller.
Remark. *Cheap control policy is when for the controller it is less expensive to increase the control signal, rather than the states. This then penalize the states more than the control signal, leading to that one may have high control signal values, rather than high state values.*

3

Approach

This chapter will describe how the approach of the different stages was done. From making the models, to how the different controllers were implemented. This include the workflow, mathematical formulation of the problem, the design of the simulations, analysis of the models, and how the control algorithms were implemented.

3.1 Mathematical Model

The mathematical modeling of the described problem will be shown how it was made. Together with the assumptions that was taken into making a general model for this problem. This results in nonlinear and linear models. The parameters that was used and the structure of the problem will be described.

3.1.1 Description of the System

The system to be investigated and controlled is a pendulum driven spherical robot with an inverted pendulum to balance a mass on top of the sphere. This system combines two common examples of systems used in describing a linear system such as a pendulum, and a nonlinear system such as an inverted pendulum. In this system they are connected in the center of the sphere, which also gives rise to some connections questions between the two sub-systems. The center of the sphere is connected via an axis to the edges of the sphere, a central axis of sort, with the connection box in the center of the sphere. Each of these sub-systems having their specific work to accomplish, for the pendulum it is to steer the spherical robot, and the inverted pendulum to balance the mass. For the inverted pendulum to hold on to a load, magnets that attract the load to the inverted pendulum inside the sphere. This gives that the load will be able to move on the surface of the sphere, if some appropriate bearings is used, which is assumed. A illustrative figure of the system is shown in the Figure 3.1.

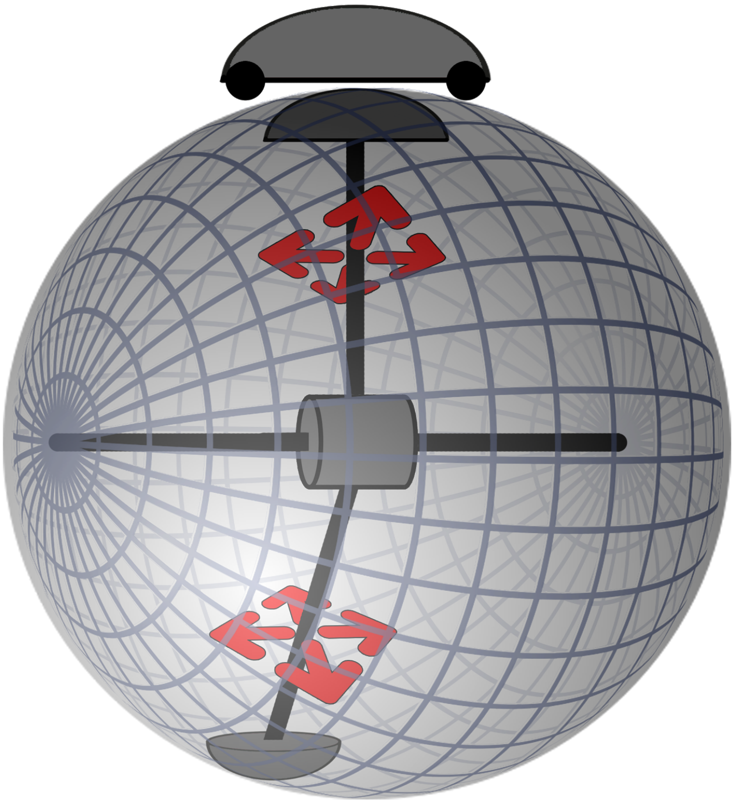


Figure 3.1: Illustration of the system, featuring the pendulum, inverted pendulum, and the load on top.

The Figure 3.1 is an illustration on how the system for this thesis will be formulated. As can be observed in the figure, is that the central axis's sides are mounted to the sphere. In the middle of the central axis there is the connection box that is connecting the pendulum and inverted pendulum to the center axis shaft.

3.1.2 Nominal Parameters and Uncertainties

The parameter values and uncertainties will be described and shown. It is to notice that the parameter values are all artificially made with consideration to a real system. Therefore take into consideration that the parameters are not subjected to practical tests as mentioned in the *Limitations*, chapter 1.3.

A table of the used parameters and their corresponding Real Parametric Uncertainties (RPU). However, due to the number of used parameters some are not considered to have RPU. That is to lessen the computational time needed for all the tests.

Name	Nominal Value	RPU
m_s: Sphere Mass	3.0kg	0%
m_p: Pendulum Mass	12.0kg	0%
m_{p_l}: Pendulum Shaft Mass	0.1kg	0%
m_h: Inverted Pendulum Mass	1.0kg	0%
m_{p_u}: Inverted Pendulum Shaft Mass	0.1kg	0%
m_l: Load Mass	2.0kg	0%

Table 3.1: Nominal values and uncertainties of the mass parameters

Name	Nominal Value	RPU
r_s: Sphere Radius	0.50 Meter	0%
r_p: Pendulum Shaft Length	0.45 Meter	0%
r_h: Inverted Pendulum Shaft Length	0.45 Meter	0%
l: Load Length (Center to sphere's shell)	0.15 meter	0%

Table 3.2: Nominal values and uncertainties of the length parameters

Name	Nominal Value	RPU
φ: Slope of the Ground	0	0
c_l: Load Inertia Coefficient	0.5	20%

Table 3.3: Nominal values and uncertainties of the other parameters

3.1.3 Lagrange's Equation formulation

To make the nonlinear model, one can use one of the given approaches in *Mathematical Model Development*, chapter 2.2. The one which is to be used is *Lagrange's equation* approach. The reason for this is because of that the method uses energy instead of force as *Newton's* approach utilizes, that may be beneficial in this case of having several sub-systems. Another property that is of interest is the generalized forces Q_i. Which was given in the description of the method, chapter 2.3. That givens that one only must considering the virtual work acting on the object in question.

3.1.3.1 Mathematical Model Uncovering

From the illustration of the system given in Figure 3.1, one can see how the model will be structured. While some more aspects will have to be clarified to make the model, there some the assumptions which the model is taking will be needed to be explained, and motivated. To make the model one may think about how to model the spherical robot, some parts of the system that may need to be considered is.

- Central connection box.
- Internal relations between the different sub systems, and masses.
- External forces that effects the system, such as friction.

- The magnetic forces that hold the load in place.
- Inertia's that effects the system.

All these are to consider when formulating the model uncovering. Then one may observe that the connection box is not specified, that is because there exist different ways of making this connection box. Which may be described by that one can have that the pendulum and inverted pendulum are coupled, or decoupled. This gives that for this system one may need to see the two different models to determine which of the models that are suitable for one's system design. The two models are shown in the Figures 4.1, and 4.2 respectively. Forward drive is quite straight forward, a short explanation may be. If the pendulum angle is increasing, then an increased force trying to push the pendulum down (depending on the gravity applied on the pendulum mass, and the angle of the pendulum), that is then applied to the sphere by the central axis shaft, making it begin to rotate. For the case of steering, or more specific for this model type turning. Because as one can observe from the Figure 3.1, the system has a central axis. Therefore, one can't use the same model as for forward, or backward drive. Because of the central axis to be able to steer the system, the system may tilt in order to turn. This can be done by tilting the pendulum sideways, this is shown in the turning model in Figure 4.3.

3.1.3.2 Nonlinear Model formulation

Formulation of the nonlinear mathematical model equations, using *Lagrange's equation*. Which gives that the formulation of the nonlinear mathematical model equations is, with respect to the uncovering models in Figures 4.1, 4.2, and 4.3. May be formulated with *Lagrange's equation*, described in chapter 2.3. The formula for the *Lagrange's equation* is given in equation (2.5). If one replace the *Lagrangian*, equation (2.6), with its individual energy components, namely the kinetic and potential energies ($\mathcal{L} = T - U$). One get that *Lagrange's equation*, may be formulated with the kinetic and potential energies, instead of *Lagrangian*, as shown by equation (3.1).

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_i}\right) - \frac{\partial T}{\partial q_i} + \frac{\partial U}{\partial q_i} = Q_i \tag{3.1}$$

T is the kinetic energy and U the potential energy, while the parameter Q_i is the *generalized forces* that is effecting the system. In this case these are frictional forces, and other external forces, which was explained in the chapter *External Forces*, 2.4. As for the *generalized coordinate's*, (or in this case the states that are considered) named q_i in the equation (2.5), and (3.1). That are of interests for the pendulum drive system and balance system, can be observed in the Figures 4.1, and 4.2. Which is $\theta_s, \theta_h, \theta_p$, and the derivatives, there the different letters stand for, s sphere, h inverted pendulum, p pendulum. The formulation of the kinetic terms (T_i, $i = s, h, p$) and potential terms (U_i, $i = s, h, p$), depends on for which system case one consider, coupled or uncoupled. That is because even if they are similar, they will have some different kinetic and potential energies. Therefore one may need to consider both options, for both cases one may as the Figures 4.1, and 4.2 shows. Must consider the effect of the pendulums shafts, because their effect on the system may or may not be sufficiently small to be neglected from the calculations. That's why they are considered a part of the model and assumed to have their center of mass in the centers of them. For each model shown in the Figures 4.1, 4.2, and 4.3, its formulation of the kinetic, and potential energies for each individual system parts. This gives the following energies for each system parts, for the individual models parts, are as shown in the models I, II, and III.

3.1.3.3 Nonlinear State-space Models

The *Lagrangian*, equation (2.6), is the kinetic energy minus the potential energy. Which gives that the *Lagrangian* for each different model can be described by equation (3.2).

$$\mathcal{L} = T - U = (T_s + T_p + T_h) - (U_s + U_p + U_h) \tag{3.2}$$

Using the *Lagrangian* given in the equation (3.2), with the energy terms shown in the Model's I, II, and III. One may formulate the resulting equations, from using the *Lagrangian* in *Lagrange's equation*, equation (2.5). In a nonlinear state-space form, by the general description is shown in equation (2.24).

$$A \begin{bmatrix} \ddot{\theta}_s \\ \ddot{\theta}_h \\ \ddot{\theta}_p \end{bmatrix} + B \begin{bmatrix} \dot{\theta}_s^2 \\ \dot{\theta}_h^2 \\ \dot{\theta}_p^2 \end{bmatrix} + C \begin{bmatrix} \dot{\theta}_s\dot{\theta}_h \\ \dot{\theta}_s\dot{\theta}_p \\ \dot{\theta}_h\dot{\theta}_p \end{bmatrix} + D \begin{bmatrix} \dot{\theta}_s \\ \dot{\theta}_h \\ \dot{\theta}_p \end{bmatrix} + G = E \tag{3.3}$$

The matrices A, B, C, and D are shown what variables they depend upon in the equation (3.3), however for G and E may not be that apparent. The matrix G is for the nonlinear 'constant' terms, these may contain constants, but also trigonometrical functions. The matrix E contains the external forces that effects the system , which is proportional to the generalized forces in the *Lagrange's equation* (2.5), named Q. Therefore one may say that the matrices A, B, C, D, and G are obtained from the *Lagrange's equation*, equation (2.5), left side. While the matrix E is obtained from the right side after calculating the *virtual work* that the external forces do onto the system. The external forces, which was described in details in the chapter *External Forces*, chapter 2.4. Some of the frictional forces from that can be summarized to the equations (3.4)-(3.7), expressed as torque.

$$\tau_{s_h} = \mu_{s_h} F_{N_h} r_s = \mu_{s_h} \left(F_{N_l} + F_{N_m}\right) r_s \tag{3.4}$$
$$\tau_{d_h} = \mu_{d_h} F_{N_h} \omega_s r_s^2 = \mu_{d_h} \left(F_{N_l} + F_{N_m}\right) \dot{\theta}_s r_s^2 \tag{3.5}$$
$$\tau_{s_g} = \mu_{s_g} F_{N_g} r_s \tag{3.6}$$
$$\tau_{d_g} = \mu_{d_g} F_{N_g} \omega_s r_s^2 = \mu_{d_g} F_{N_g} \dot{\theta}_s r_s^2 \tag{3.7}$$

There F_{M_m} is the force applied by the magnets, and the values of the normal forces F_{N_i}, $i = g, l$ is given as.

$$F_{N_l} = \frac{m_l g}{\cos\left(\theta_h + \theta_s\right)} \tag{3.8}$$
$$F_{N_g} = \frac{m_{tot} g}{\cos\left(\varphi\right)} \tag{3.9}$$

The contributions from the *generalized forces* to the dynamics in the *Lagrange's equation*, equation (2.5), with respect to the states $\theta_s, \theta_h, \theta_p$. Results in different effecting factors in Q dependent on the states q, with the input torques for the pendulums (τ_h, and τ_p). However, an important assumption of the model is that the body is rigid, and therefore there is no slip. This gives by the definition of *virtual work*, that *virtual work* is only done if there is a *virtual displacement*. By the assumption of no slip into the model, gives that the frictions in equations (3.4)-(3.7) do not have any virtual displacement. Giving that they are not contributing towards the *generalized forces* [29]. Not to be confused with the external torques from the motors of the system, τ_p is the torque from the driving motor for the pendulum, and τ_h is the torque from the inverted pendulum motor. They will contribute to the *generalized forces* depending on the angles that they are effected on, that differs from the two cases of coupled, or decoupled model, show in the Model's I, and II. To achieve that the head kept aligned with the inverted pendulum, a magnetic field is used that is acting between the load, and the inverted pendulum. For the sphere to freely roll around on the sphere, using some sort of bearing. This will contribute to some external force being applied onto the sphere, a magnetic field which can be formulated in different ways, which was shortly described in *The magnetic forces*, chapter 2.4. However, for simplicity it is assumed that it has a constant force which is sufficiently large so that the head will be kept in place. Because it is assumed that the sphere is rigid enough that there won't be any virtual displacement, leads to that no virtual work is done. Then it is not contributing to the system, and is included in the equations (3.4)-(3.7), as F_{N_m}, a constant force.

3.1.4 Linear State-space Models

To make the linear state-space models, linearization of the nonlinear models given in the chapter 2.5, equations (2.22)-(2.23), as explained in the chapter 2.6, is appropriate. Then one would need to have a reasonable stationary point, to linearize at. This stationary point must result in that the nonlinear function is equal to zero, and is close to the working, or operating area. That is because of the nonlinear property which is lost in the linearization, which may result in that the further away from that point the larger the error becomes. Therefore, one may argue which stationary point to use, for that reason a general expression of the linearization will be obtained for the models. For the case of the nonlinear state-space model representation given in equation 3.3, (shown here for clarity).

$$A \begin{bmatrix} \ddot{\theta}_s \\ \ddot{\theta}_h \\ \ddot{\theta}_p \end{bmatrix} + B \begin{bmatrix} \dot{\theta}_s^2 \\ \dot{\theta}_h^2 \\ \dot{\theta}_p^2 \end{bmatrix} + C \begin{bmatrix} \dot{\theta}_s\dot{\theta}_h \\ \dot{\theta}_s\dot{\theta}_p \\ \dot{\theta}_h\dot{\theta}_p \end{bmatrix} + D \begin{bmatrix} \dot{\theta}_s \\ \dot{\theta}_h \\ \dot{\theta}_p \end{bmatrix} + G = E$$

To simplify the nonlinear models from trigonometrical functions one may use the *Taylors series*, to express the trigonometrical functions as a number of terms instead, Taylor polynomial, shown in equations (2.25)-(2.26). In this case it may be suitable to use a first-order approximation. This method is called *small-angle approximation*, and as the name say it is an approximation of these trigonometric functions when they have small angels. This gives the approximation of the trigonometric functions, first-order approximation.

$$\sin(\theta) \approx \theta$$
$$\cos(\theta) \approx 1$$

The angle of the ground slope φ, may be argued not to be considered small. This results in that the assumption of small angles could give larger error for the slope, this is assumed not to be of major issue. Using the general nonlinear function formulation, equations (2.22)-(2.23).

$$\dot{x}(t) = f(x(t), u(t))$$
$$y(t) = h(x(t), u(t))$$

Then to linearize the nonlinear model to a linear model of the form.

$$\textit{State (dynamic) equation: } \dot{x}(t) = Ax(t) + Bu(t)$$
$$\textit{Measurement equation: } y(t) = Cx(t) + Du(t)$$

$x(t)$ is the states and $u(t)$ the input signals. Using a stationary point (x_0, u_0) to linearize the nonlinear functions f and h, gives as shown in the equation (2.35), and (2.36). f is the resulting nonlinear function given from *Lagrange's equation* (2.5). While the states that are considered are $\theta_s, \theta_h, \theta_p, \dot{\theta}_s, \dot{\theta}_h, \dot{\theta}_p$, and the input signal are the torque of the motors, τ_h, and τ_p. This gives that the internal states and control signal is defined as shown in the equations (3.10), and (3.11).

$$x(t) = \begin{bmatrix} \dot{\theta}_s & \dot{\theta}_h & \dot{\theta}_p & \theta_s & \theta_h & \theta_p \end{bmatrix}^T \tag{3.10}$$
$$u(t) = \begin{bmatrix} \tau_p & \tau_h \end{bmatrix}^T \tag{3.11}$$

Using the nonlinear state-space equation (3.3), and rewrite it into a more workable form for linearization, into the equation (3.12).

$$\begin{bmatrix} \ddot{\theta}_s \\ \ddot{\theta}_h \\ \ddot{\theta}_p \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = A^{-1}E - A^{-1}B \begin{bmatrix} \dot{\theta}_s^2 \\ \dot{\theta}_h^2 \\ \dot{\theta}_p^2 \end{bmatrix} - A^{-1}C \begin{bmatrix} \dot{\theta}_s\dot{\theta}_h \\ \dot{\theta}_s\dot{\theta}_p \\ \dot{\theta}_h\dot{\theta}_p \end{bmatrix} - A^{-1}D \begin{bmatrix} \dot{\theta}_s \\ \dot{\theta}_h \\ \dot{\theta}_p \end{bmatrix} - A^{-1}G \tag{3.12}$$

The inverse matrix of A can be structured in its elements with the constant terms of $a_{i,j}$, for every i row, and j column.

$$A^{-1} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \tag{3.13}$$

Using the linearization of the state (dynamic) equation (2.35) to linearize the non-linear state-space model (3.12), into the linearized model given in equation (2.31). With respect to the states $\dot{\theta}_s, \dot{\theta}_h, \dot{\theta}_p, \theta_s, \theta_h, \theta_p$, and the input states τ_p, and τ_h, shown in the equations (3.10), and (3.11), with a general stationary point. Gives using that the inverse matrix of A is defined as in equation (3.13), that the linear models becomes as shown in the Models VI, and V.

3.1.5 Inertias of the Models

The inertias of the individual parts of the system, is a question. They may be assumed to be a certain geometrical structure like a cylinder or assumed to be a thin line. This is to be considered for each of the parts, that is because even though some parts may have negligible influence on the total system, it is not known for certain. Therefore the different inertias are assumed to be as shown in equations (3.14)-(3.19), with the assumed form and moment of inertia.

$$\textit{Thin spherical shell:} \quad I_s = \frac{2}{3} m_s r_s^2 \tag{3.14}$$

$$\textit{Unknown geometrical form:} \quad I_l = c_l m_l r_l^2 \tag{3.15}$$

$$\textit{Hemisphere:} \quad I_p = \frac{2}{5} m_p r_p^2 \tag{3.16}$$

$$\textit{Hemisphere:} \quad I_h = \frac{2}{5} m_h r_h^2 \tag{3.17}$$

$$\textit{Narrow straight bar:} \quad I_{p_l} = \frac{1}{12} m_{p_l} \left(\frac{r_p}{2}\right)^2 \tag{3.18}$$

$$\textit{Narrow straight bar:} \quad I_{p_u} = \frac{1}{12} m_{p_u} \left(\frac{r_h}{2}\right)^2 \tag{3.19}$$

The load inertia I_l is having a coefficient c_l to it, that is because it is hard to define exactly what type of geometrical structure the load will have. Therefore, it has an assumed constant coefficient in front of it. Which is done in order to observe the effect of different loads by using RPU on the coefficient c_l for the load. The nominal values, and RPUs are shown in the tables 3.1-3.3.

3.2 Simulation Programs

To evaluate the different control algorithms and observe how the system behaves from them. Simulation will be made, to find how they perform in terms of stabilizing and regulating the nonlinear mathematical model. The questions that the simulation will be made to answer, can be found in chapter *Research Questions*, chapter 1.4. However, because of limitations found in the chapter *Limitations*, chapter 1.3. Only simulations will be made to verify the models and controllers. Therefore, it is of importance to have reliable simulation setup, which to base upon. To synthesis the controllers and simulate the controllers behavior with the nonlinear mathematical models, *MATLAB* & *Simulink* (*By MathWorks*) will be used for the simulation programs.

3.2.1 MATLAB

To build up the models and synthesis the controller algorithms *MATLAB (MathWorks)* will be utilized for this purpose, *MATLAB R2019a*. The main reason is that as mentioned *Simulink* will also be used, which gives the possibility of utilizing some properties that follows. That they can be used together, what the meaning by that is that if *MATLAB* have some parameters values in its workspace then *Simulink* may also use these. This gives that when one synthesis the linear and nonlinear controllers in *MATLAB*, one can use these resulting controllers in the simulation in *Simulink*. Which gives that is becomes easier to test out different settings of controllers, and that one can utilize the functions that *MATLAB* have to synthesis controllers. The linear and nonlinear models will be defined depending on the controller to be used, with or without real parametric uncertainty. *MATLAB* will also be used to determine properties of the models, such as controllability, stabilizability, observability, and detectability. But also, the combination of controller and model to check nominal stability, nominal performance, robust stability, and robust performance. Which are defined in *Linear State-space Model Analysis*, chapter 2.7.1.

3.2.2 Simulink

Using the resulting mathematical models and control algorithms from *MATLAB*, one may use *Simulink* to formulate a model based design of the system. The internal dynamics of the system are described by the nonlinear mathematical models IV, and V, depending on which model one wish to test. The models are given by a *MATLAB* function block, to call the *MATLAB* functions which contains the mathematical formulation of the system. This gives that the output is described by the angular acceleration, and therefore is integrated to angular velocity and angular position. Which gives that all tests of the control algorithms have a common building block in the *Simulink* model, which is shown in the Figure 3.2.

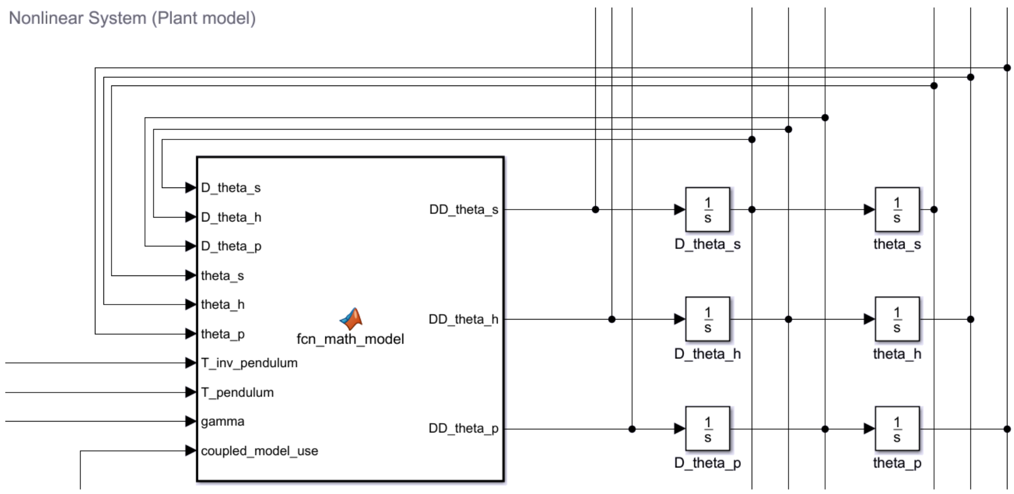


Figure 3.2: The internal dynamics of the nonlinear system, a building structure used in all simulations of the control algorithms.

Inside the *MATLAB* function block, in Figure 3.2, depending if one use the *coupled model*, or the *decoupled model*, the parameter *coupled_model_use* will decide which one to use for testing. This structuring of separating the internal dynamics of the system, and the control algorithm can be viewed as to following the structure found in Figure 2.2a. With the internal dynamics of the system designated as the P in the figure, and K as the controller. As pointed out earlier this formulation is suitable for some control algorithms, this may need to be expanded to add uncertainties, in this case *real parametric uncertainties*. Then the structure observed in Figure 2.2b is more suitable for that. However, the Δ is not a part of the simulation, it is for calculating the control algorithms that needs it.

Measurement Noise The simulations will be done with and without measurement noise, a white Gaussian noise of low magnitude. The noise is make by *MATLAB* as an MATLAB function block in Simulink, the *MATLAB* function 'wgn(m,n,power)' is used to create white Gaussian noise. 'm', and 'n' is defining the noise matrix output size of the *MATLAB* function, while the 'power' is the power of the noise. That may be interpreted in different ways, named 'powertype' which is chosen as to be linear. This function may not be directly design for this, however it does give white Gaussian noise, which will add a realistic touch to the simulations. The choice of power of the white Gaussian noise, which was determined to be very small, just to simulate some static noise, was chosen to be of the power 10^{-4}.

Initial Values & Goal The initial values of the states is determined to be 10% of the steady state, to observe how well it manage to stabilize the system away from the steady state value.

3.3 Model Analysis Evaluation

Analysis of the model is important for the reason of observing if one even may control and stabilize the system. Some parts of the model may contain unobservable states for the controller, and if these states are not stable, the system may become unstable. Therefore, it is of importance to analyze the models of the system, both with and without the controller. Theoretical background of model analysis, with and without controller, can be found in *Model Analysis*, chapter 2.7.

3.3.1 Linear Model Analysis

Analysis of the linear models shown in the Models VI, and VII, with the parameters shown in the tables 3.1-3.3. Is done with the methods given in *Linear State-space Model Analysis*, chapter 2.7.1. In which is done with respect to that the C, and D matrices for the linear state-space representation, given in the equation (2.32). Is defined as in equation (3.20).

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{3.20}$$

Then analysis of the models with the C, and D matrices, gives that one may express the analysis in three stages. That is because there are real parametric uncertainties in the system. Which gives that in order to analyze the different properties of the models, can be done by first checking the nominal linear models without controllers. Followed with the nominal linear models with controllers, and last check the uncertain linear models with controllers.

Nominal Linear Model's Analysis without Controller Analysis of the nominal linear models without controllers, is done before making the controllers to check properties of the model itself. Such as the eigenvalues, controllablity, stabilizabiltiy, observability, and detectability. Which are evaluated by the ways mentioned in the chapter 2.7.1. That is to get some intuition of how the system dynamics is, and if the models can be controlled and stabilized.

Nominal Linear Model's Analysis with Controller This analysis needs a controller to have been synthesized beforehand, which is then used to control the system. That becomes as shown in the Figure 2.2a. This is to observe if the controller can make the system stable, and if the performance specification is upheld, as described in the chapter 2.7.1. Evaluation of the nominal stability (NS), and nominal performance (NP) is followed by its definitions 2.10, and 2.11.

Uncertain Linear Model's Analysis with Controller Similar to the *Nominal Linear Model's Analysis with Controller* case with that one adds the real parametric uncertainties (RPU) found in the parameter tables 3.1-3.3. This gives a new view of the system, that is if one considers that the system is not perfect and will have some defects, imperfection, or errors from the original drawing. Which gives a range of possibilities for different combinations of parameter values, that one may need to consider in some worst-case scenario. The parameters uncertainty may turn out to make the system too difficult to control, costly, and possibly even impossible to control, if the uncertainty is too high. This uncertainty may be added as a block Δ, as is illustrated in the Figure 2.2b. Which is to calculate the robust stability, and robust performance by its definitions 2.12, and 2.13, as shown in the chapter 2.7.1.

3.3.2 Nonlinear Model Analysis

Analysis of the nonlinear models, *coupled model* IV, and *decoupled model* V. With the parameters shown in the tables 3.1-3.3. Could be done with the method given in *Nonlinear State-space Model Analysis*, chapter 2.7.2. However, as mentioned it can be hard to analyze a nonlinear model for various reasons. Therefore, an easier way for this is to assume that if the linear model is stable, then the nonlinear model around the linearization point is also stable. Which may be strengthen by that the stationary point is calculated with the nonlinear models. This gives that the point itself is assumed to have a stable value at that point, and in some neighborhood around the stationary point.

3.4 Control Algorithms Evaluation

In this chapter a walk-through in the how the controllers were made, and the used design parameters. Following the design method in *Control Algorithms*, chapter 2.9.

3.4.1 Linear Control algorithms

The linear control algorithms implementation will be explained and shown here. Theoretical explanations of these algorithms are given in *Linear Control Algorithms*, chapter 2.9.1. However, PID controller will not be made, shown because it is a commonly used controller in some areas, such as industry. As explained in *Simulation Programs*, chapter 3.2, MATLAB and Simulink will be used to simulate the controllers behavior onto the unstable nonlinear system. For the output feedback control algorithms, the C, and D matrices of the linear state-space model is given in the equation (3.20).

3.4.1.1 Linear Quadratic Controllers (LQR, LQI, LQG)

Implementation of the linear quadratic controllers, was done with the MATLABs built in functions, that can use the *Riccarti functions*, and other properties to calculate the controllers. The controllers are depending on the tuning variable matrices, or design parameters, that is shown for each of the three cases of linear quadratic controllers (LQR, LQI, and LQG). How one can solve the different linear quadratic controllers are shown in *Linear Quadratic Controllers*, chapter 2.9.1.2.

Linear Quadratic Regulator (LQR) Linear quadratic regulator, or LQR, the controller is made using MATLABs build in function named 'lqr'. Which takes in the linear system to be controlled together with three design matrix parameters. The design matrix parameters that they name 'Q', 'R', and 'N', is equivalent to the design matrix parameters Q_x, Q_u, and N as described in the *Linear Quadratic Controllers*, chapter 2.9.1.2. They are used to express how one should penalization the different signals and states, with the cost function, as shown in the equation (2.54). As mentioned, one can have a one or two degree of freedom controller, either the error of the system is making the control signal, or the reference and feedback is separately considered. Then added together to make the control signal. To test the full potential of the LQR, an two degree of freedom controller is applied, the gain matrix for the reference is calculated with the equation 2.69. Using cheap control policy for the design choice of the design parameters, this means that one makes it less expensive for the controller to have high input signals, rather than high state values. This gives that the design parameters for the LQR, became as shown in the equation (3.21).

$$\begin{gathered}\textit{Decoupled Model LQR controller parameters}\\ Q = \begin{bmatrix} I_{6\times 6} \end{bmatrix}, \quad R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}, \quad N = \begin{bmatrix} 0_{6\times 2} \end{bmatrix}\end{gathered} \tag{3.21}$$

The resulting simulations with and without noise, is shown in the *Linear Quadratic Controllers*, chapter 4.3.1.

Linear Quadratic Integrator (LQI) Linear quadratic integrator, or LQI, is as one can say an LQR controller with an added integral action. The controller is made using MATLABs build in function named 'lqi', which takes in the linear system to be controlled together with three design matrix parameters. The design matrix parameters that they name 'Q', 'R', and 'N', is equivalent to the design matrix parameters Q_x, Q_u, and N as described in the *Linear Quadratic Controllers*, chapter 2.9.1.2. However, the matrix Q is extended to also include the integral action error penalizations. They are used to express how one should penalization the different signals and states, with the cost function, as shown in the equation (2.54). With that the states x is expanded with the integral action, to z. One can have a one or two degree of freedom controller, either the error of the system is making the control signal, or the reference and feedback is separately considered. Then added together to make the control signal. To test the full potential of the LQI, an two degree of freedom controller is applied, the gain matrix for the reference is calculated with the equation 2.69. Using cheap control policy for the design choice of the design parameters, this means that one makes it less expensive for the controller to have high input signals, rather than high state values. That however gave that the signals became too high causing instability, therefore expensive control policy was determined to be needed. This gives that the design parameters for the LQI, became as shown in the equation (3.22).

Decoupled Model LQI controller parameters

$$\begin{gathered} Q_{states} = \begin{bmatrix} I_{6\times 6} \end{bmatrix}, \quad Q_{integration} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 10^{-10} \end{bmatrix} \\ Q = \begin{bmatrix} Q_{states} & 0 \\ 0 & Q_{integration} \end{bmatrix}, \quad R = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}, \quad N = \begin{bmatrix} 0_{6\times 2} \end{bmatrix} \end{gathered} \tag{3.22}$$

The resulting simulations with and without noise, is shown in the *Linear Quadratic Controllers*, chapter 4.3.1.

Linear Quadratic Guassian (LQG) Linear quadratic Gaussian, or LQG, is as one may say an LQR controller with that one can't observe all the states, therefore one has to estimate them with an state observer. The controller can be made using MATLABs build in function named 'lqg'. That calculate an optimal linear quadratic Gaussian controller, consisting of an LQR (or LQI), and a Kalman filter. However, because of *principle of separation*, definition 2.23, one can separately design the feedback controller and the state observer. Since an LQR has already been made, and to observe how well it improves with a Kalman filter, the LQR from before will be used. While a Kalman filter is made using MATLABs built in function 'Kalman'. The function 'Kalman' takes in three design matrix parameters, namely 'Qn', 'Rn', and 'Nn', which is the covariance matrices for the Kalman filter. The design parameters for the Kalman filter, became as shown in the equation (3.23).

$$Qn = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}, \quad Rn = \begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}, \quad Nn = \begin{bmatrix} 0_{2\times 3} \end{bmatrix} \tag{3.23}$$

The resulting simulations with and without noise, is shown in the *Linear Quadratic Controllers*, chapter 4.3.1.

3.4.1.2 $\mathcal{H}_2$ and $\mathcal{H}_\infty$ Controller

The implementation of the $\mathcal{H}_2$, and $\mathcal{H}_\infty$ controllers is done in a similar way to each other. Following the explanation given in *$\mathcal{H}_2$, and $\mathcal{H}_\infty$ Control*, chapter 2.9.1.3. With that the system structure are defined as a standard problem formulation with uncertainties, shown in the Figure 2.2b. The system block P in the Figure 2.2b, is the nominal open-loop plant state-space model $G_{Nominal}$. With weights to certain inputs, and outputs. This is to filter parts of the signals and express contributions to the plant P. The implementation structure for these two controllers may be seen in the Figure 3.3.

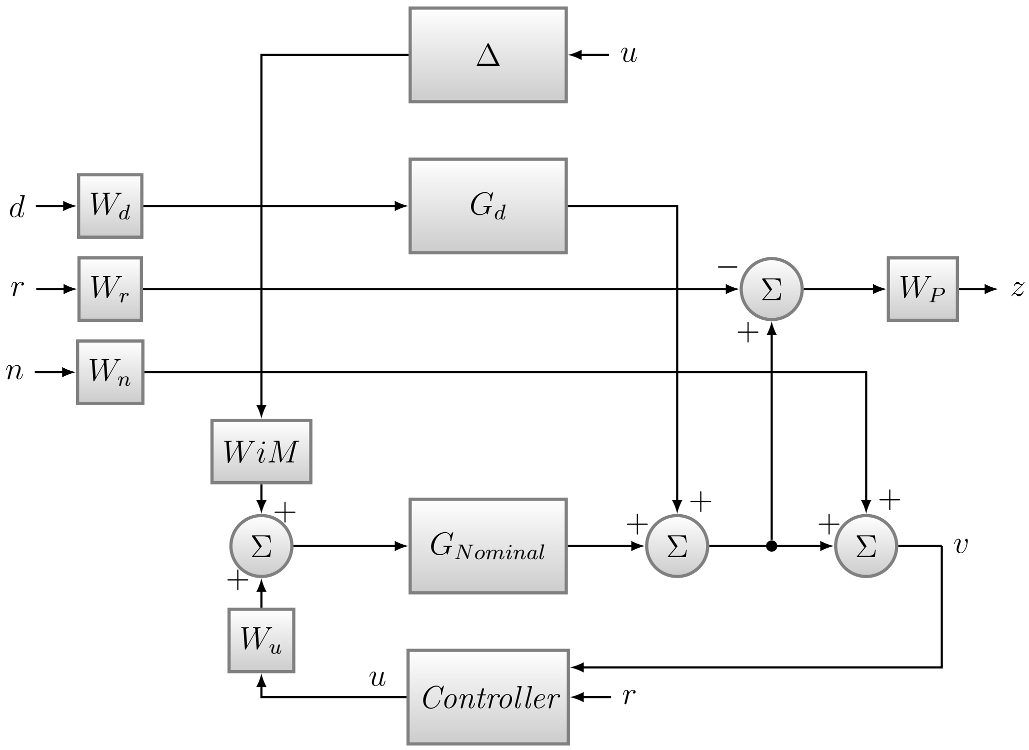


Figure 3.3: Approach of a standard problem formulations of feedback designs, with uncertainty and weighted signals. To construct the system, which is used to defined the $\mathcal{H}_2$, and $\mathcal{H}_\infty$ controllers.

The input to the system which may be denoted as w, contains the individual signals of disturbance d, reference r, and noise n. The output named z, depends on what one wants to have to measure. For this case, it is the error of the system compared to some reference. Each of the signals have a weight block which is as described in the $\mathcal{H}_2$, and $\mathcal{H}_\infty$ controller synthesis theory, chapter 2.9.1.3, a way of tuning the system build up. To find the appropriate weights was done by investigating the uncertain system, more specifically the singular values of the uncertain system. That was to find how the uncertainties was affecting the system, so one could determine the appropriate weights to make the weights rule out unnecessary areas to look at.

What that mean is that for the weight WiM one wants to observe the area where the model is uncertain. For our systems singular values of the uncertain system is shown in the Figure 3.4.

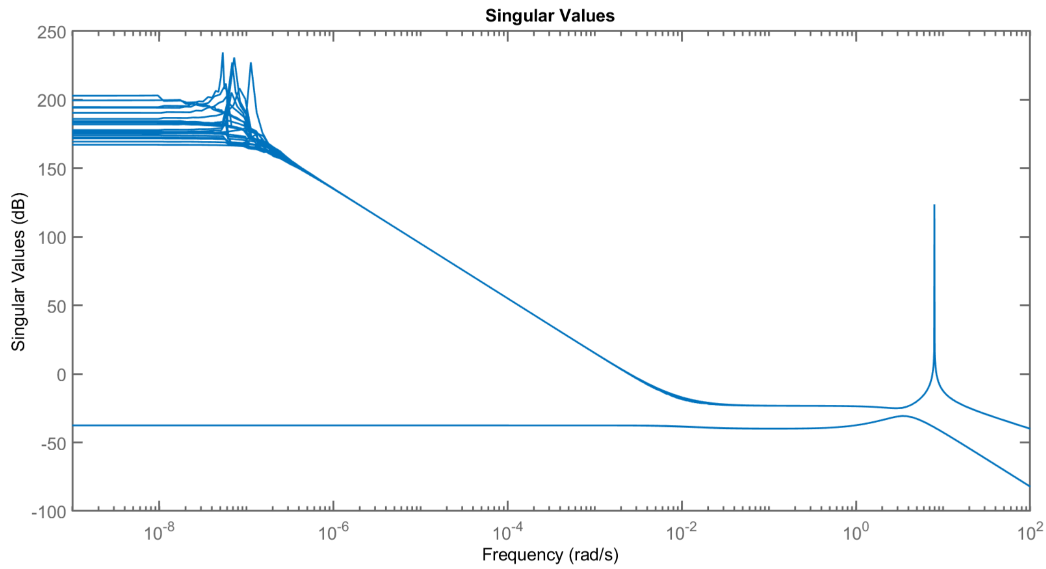


Figure 3.4: The singular value plot of the uncertain system, of the *decoupled model*, that shows areas where the uncertainty is effecting the singular values of the system.

From the Figure 3.4, one can observe the area of frequencies that is needed to be observed to see all uncertain parts of the system. WiM can be obtained with filters on the diagonal. While for the noise, W_n, is assumed to be in the higher frequencies, with low power. The reference is not affected and therefore is identity matrix. While the input signal is changed with coefficients to change the input to the system. For the W_P, that is the performance weight. It works in that the performance is a measurement of how well the system is behaving. The weights of the system used to get the results shown in the chapter 4.3.3, and 4.3.2. Shown in Laplace domain, by the equations (3.24)-(3.26).

$$G_d(s) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad W_d(s) = \begin{bmatrix} \frac{0.01}{10^3 \cdot s+1} & 0 & 0 \\ 0 & \frac{0.001}{10^3 \cdot s+1} & 0 \\ 0 & 0 & \frac{0.01}{10^3 \cdot s+1} \end{bmatrix}, \quad W_r(s) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.24}$$

$$W_n(s) = \begin{bmatrix} \frac{0.01 \cdot s}{s+1} & 0 & 0 \\ 0 & \frac{0.001 \cdot s}{s+1} & 0 \\ 0 & 0 & \frac{0.01 \cdot s}{s+1} \end{bmatrix}, \quad W_P(s) = \begin{bmatrix} \frac{0.8}{s+1} & 0 & 0 \\ 0 & \frac{0.95}{s+1} & 0 \\ 0 & 0 & \frac{0.8}{s+1} \end{bmatrix} \tag{3.25}$$

$$WiM(s) = \begin{bmatrix} 0.005 \cdot \frac{10^6 \cdot s+1}{10^4 \cdot s+1} & 0 \\ 0 & 0.01 \cdot \frac{10^6 \cdot s+1}{10^4 \cdot s+1} \end{bmatrix}, \quad W_u(s) = \begin{bmatrix} 0.8 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.26}$$

The nominal system $G_{Nominal}$ that is defined as the nominal *decoupled model*. Then to make the controllers MATLABs built in functions for $\mathcal{H}_2$, and $\mathcal{H}_\infty$ optimal controllers is used.

$\mathcal{H}_2$ **Controller** Was made using MATLABs built in function 'h2syn', which takes in a P the open-loop plant state-space model P shown before. Together with three more inputs namely 'nmeas', 'ncont', and 'opt'. 'nmeas' is the number of measurement output signals of the plant P, which here is equal to three. 'ncont' is the number of control input signals of the plant P, which is equal to two. 'opts' is optional options for the computation of the $\mathcal{H}_2$ controller, more specifically is the 'h2synOptions' in MATLAB. The resulting simulations with and without noise, is shown in $\mathcal{H}_2$ *Controller*, chapter 4.3.2.

$\mathcal{H}_\infty$ **Controller** Similarly, to how the $\mathcal{H}_2$ controller was formulated, with instead using MATLABs built in function 'hinfsyn'. That takes in a P that is the open-loop plant state-space model P, while 'nmeas', and 'ncont' is the same as for the $\mathcal{H}_2$ controller. 'nmeas' is the number of measurement output signals of the plant P, which here is equal to three. 'ncont' is the number of control input signals of the plant P, which is equal to two. 'opts' is optional options for the computation of the $\mathcal{H}_\infty$ controller, more specifically is the 'hinfsynOptions' in MATLAB. An input that differs compared to the $\mathcal{H}_2$ controller, $\mathcal{H}_\infty$ controller may have either 'gamTry', or 'gamRange' as an input. 'gamTry' is for specifying an upper limit on the gamma value of the controller, that may potentially fail to get a controller if too low. While 'gamRange' is to specify an interval for the gamma value to limit the computational time. The resulting simulations with and without noise, is shown in $\mathcal{H}_\infty$ *Controller*, chapter 4.3.3.

3.4.1.3 μ-synthesis (*DK*-iteration)

Implementation of the μ-synthesis controller, or as mentioned when describing the μ-synthesis, in chapter 2.9.1.4, specifically *DK-iteration*. The implementation of this controller was done by defining a system similar with how the $\mathcal{H}_2$, and $\mathcal{H}_\infty$ controllers was defined, showed in the Figure 3.3. Except with no uncertainty block Δ, with that the uncertainty instead is inside the plant P, giving that the real parametric uncertainty is inside the system matrix G. While the other weights are equal to their values in equations (3.24)-(3.26). This gives an appearance similar to the standard problem formulation without uncertainty, seen in Figure 2.2a. To solve this, one can follow the steps of *DK-iteration* given in μ-synthesis, chapter 2.9.1.4, or as done here use MATLABs built in function 'dksyn'. The input 'p' is the uncertain open-loop plant state-space model, which then the real parametric uncertainties need to be a part of the system matrix. While similarly to the $\mathcal{H}_2$, and $\mathcal{H}_\infty$ controllers, 'nmeas' is the number of measurement output signals of the plant P, which here is equal to three. 'ncont' is the number of control input signals of the plant P, which is equal to two. 'opts' optional options for 'dksyn' where one can specify the use of *DK-iteration* or *DGK-iteration*, which is another approach one could use. The resulting simulations with and without noise, is shown in *μ-Synthesis*, chapter 4.3.4. However, due to stability problems of the controller containing unstable parts, only the stable part of the controller is being used for the simulation. Giving that some parts of the system is missing.

3.4.2 Nonlinear Control algorithms

The nonlinear control algorithms implementation will be explained and shown here. Theoretical explanations of these algorithms are given in *Nonlinear Control Algorithms*, chapter 2.9.2. As explained in *Simulation Programs*, chapter 3.2, MATLAB and Simulink will be used to simulate the controllers behavior onto the unstable nonlinear system.

3.4.2.1 Feedback Linearization

The approach to make *Feedback Linearization* is not as strait forward as the linear control algorithms. Theoretical background for this approach is given in the chapter 2.9.2.1. Which can be explained by either *input to state linearization* or *input to output linearization*, depending if one have to consider some *zero dynamics*. Here one have to find the *relative degree*, the *state transformation*, and check the *zero dynamics* to be stable.

Formulating the Problem Using the general form of the models given by the equation (3.3), to formulate the nonlinear equations (2.97)-(2.98). Which is for the formulation of the transformation of the state, $T(x)$, and finding the *relative degree*. The states is then defined as equation (3.27), with its derivatives shown in the equation (3.28).

$$x = \begin{bmatrix} \dot{\theta}_s \\ \dot{\theta}_h \\ \dot{\theta}_p \\ \theta_s \\ \theta_h \\ \theta_p \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} \tag{3.27}$$

$$\dot{x} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} = \underbrace{\begin{bmatrix} -A^{-1}B \begin{bmatrix} x_1^2 \\ x_2^2 \\ x_3^2 \end{bmatrix} - A^{-1}C \begin{bmatrix} x_1x_2 \\ x_1x_3 \\ x_2x_3 \end{bmatrix} - A^{-1}D \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} - A^{-1}G \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}}_{f(x)} + \underbrace{\begin{bmatrix} -A^{-1}E \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{g(x)} u \tag{3.28}$$

Instead of using the *lie derivatives*, the derivatives of the output $y(t)$ is used to determine the *relative degree* and the transformation of the state $T(x)$. With the output set to $y = h(x) = x_4 = \theta_s$, gives the calculation shown in equation (3.29).

$$\begin{aligned} y(t) =& h(x) = x_4 = \theta_s, \quad \sim \textit{no inputs} \\ \dot{y} =& \dot{x}_4 = \dot{x}_4 = x_1 = \dot{\theta}_s, \quad \sim \textit{no inputs} \\ \ddot{y} =& \dot{x}_1 = \ddot{x}_4 = \ddot{\theta}_s = f_1(x) + g_1(x)u, \quad \sim \textit{inputs} \end{aligned} \tag{3.29}$$

This gives that the *relative degree* is equal to two. Which is telling us that there exist some dynamics which is not observable, namely the *zero dynamics*. However, formulating the transformation map $T(x)$ with that is known, then becomes as shown in the equation (3.30).

$$z = T(x) = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ y \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \eta \\ \xi \end{bmatrix} \implies \xi = \begin{bmatrix} \theta_s \\ \dot{\theta}_s \end{bmatrix} \tag{3.30}$$

Zero Dynamics The *zero dynamic* is the unobservable states of the feedback linearized system, defined by equation (2.113). Using this definition with the model may result in difficult equation. Which for this problem is occurring, that's why it cannot be guaranteed to have stable *zero dynamics*, because one can't observe if it is stable or not.

Input Manipulation for Linearization Using the *feedback linearization* to formulate a control signal $u(t)$, that results in that the nonlinear system behaves like a linear one. This control signal can be formulated as mentioned in *Feedback Linearization*, chapter 2.9.2.1. Which gives that one may formulate the new linear looking state-space using a new v control signal.

$$\begin{aligned} \dot{\xi} =& A_c\xi + B_c \underbrace{\gamma(x)[u - \alpha(x)]}_{=v} \\ y =& C_c\xi \end{aligned} \tag{3.31}$$

Giving that one can formulate the control signal $u(t)$ to cancel out the non-linearity. The control signal $u(t)$ may be formulated as shown in equation (3.32).

$$u(t) = \alpha(x) + \beta(x)v \implies v = \beta^{-1}(x)[u - \alpha(x)] \tag{3.32}$$

This gives that $\beta^{-1}(x) = \gamma(x)$, the calculation to formulate the control signal $u(t)$ is shown in the equation (3.33).

$$\begin{aligned} \ddot{y} =& \ddot{\theta}_s = f_1 + g_1 u \\ u =& \alpha(x) + \beta(x)v \\ u =& -g_1(x)f_1(x) + g_1^{-1}(x)v = g_1(x)\left(-f_1(x) + v\right) \\ \implies& \ddot{y} = f_1(x) + g_1\left(-g_1^{-1}f_1(x) + g_1^{-1}v\right) = v \end{aligned} \tag{3.33}$$

This gives that the different parameters β and α becomes as shown in the equation (3.34).

$$\begin{aligned} \alpha(x) =& -g_1^{-1}(x)f_1(x) \\ \beta(x) =& g_1^{-1}(x) \implies \gamma(x) = g_1(x) \end{aligned} \tag{3.34}$$

However, due to that the *zero dynamics* is not guaranteed to be stable, if one tries to control this feedback linearized system one can get that it behaves strangely, or even unstable. That is depending on if the *zero dynamics* is stable or not, therefore another approach is to be considered.

3.4.2.2 Feedback Linearization Controller

As shown in the *Feedback Linearization*, chapter 3.4.2.1, is that to check if the *zero dynamic* is stable or not, will be a hard task. Therefore instead of having to checking the *zero dynamics*, one can use linear output selection for the feedback linearization. Which is described in *Feedback Linearization Control*, chapter 2.9.2.2. Using this approach with the states given in equation (3.27), and its derivative given in the equation (3.28). It is possible to formulate a feedback linearized control using the feedback gain of an LQR controller, as shown in the Figure 2.7. This is as mentioned done by setting the output $h(x)$ equal to a linear function of the states obtained from LQR design. Giving that the nonlinear state-space model becomes as shown in the equations (3.35).

$$\begin{aligned} \dot{x} =& f(x) + g(x)u \\ y =& K_1 x \end{aligned} \tag{3.35}$$

Then the feedback linearization parameters $\alpha(x)$, and $\beta(x)$ can be calculated by using that the output is $h(x) = y = K_1 x$. With that it has a *relative degree* of one, the calculations are shown in the equation (3.36).

$$\begin{aligned} y =& K_1 x \\ \dot{y} =& K_1 \dot{x} \\ =& K_1 \left(f(x) + g(x)u \right) \end{aligned} \tag{3.36}$$

To find a u that makes the system look linear like, could be if one use that u is as shown in equation (3.37).

$$\begin{aligned} u =& \alpha(x) + \beta(x)v \\ =& - \left(K_1 g(x) \right)^{-1} \left(K_1 f(x) \right) + \left(K_1 g(x) \right)^{-1} v \end{aligned} \tag{3.37}$$

The equation (3.37), shows as also mentioned in the description of this approach in chapter 2.9.2.2. That the $\alpha(x)$, and $\beta(x)$ are defined as in equation (3.38).

$$\begin{aligned} \alpha(x) =& -\frac{K_1 f(x)}{K_1 g(x)} \\ \beta(x) =& \frac{1}{K_1 g(x)} \end{aligned} \tag{3.38}$$

The two different LQRs used to make this feedback linearization and controller, using cheap control policy. K, the output, has been chosen to have the design parameters shown in equation (3.39).

$$Q = \left[I_{6\times6} \right], \quad R = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}, \quad N = \left[0_{6\times2} \right] \tag{3.39}$$

While the for the controller to not have to fast control, it does not have as strong values for its design parameters. The design parameters for the LQR controller is shown in the equation (3.40).

$$Q = \left[I_{6\times6} \right], \quad R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}, \quad N = \left[0_{6\times2} \right] \tag{3.40}$$

The resulting simulation with and without measurement noise, can be seen in the *Feedback Linearization Control*, chapter 4.4.1.

4

Results

In this chapter the results from the simulations, and the different properties considered in the evaluation of the different control algorithms are shown. This will include a small explanation of the results, a more detailed analysis is given in *Discussion*, chapter 5. With the conclusions discussed in the *Conclusion*, chapter 6.

4.1 Mathematical Models

Here the resulting linear and nonlinear models will be shown. With the resulting uncovering models, and the energies used for the *Lagrange's equation*. The linear and nonlinear models will be expressed in state-space formats, that was mentioned in the *Mathematical Modelling*, chapter 2.2. For the linear state-space will be given as the equations (2.31)-(2.32). While the nonlinear state-space will be given by the equation (3.3), both are shown below for clarity.

Nonlinear State-space Model

$$A \begin{bmatrix} \ddot{\theta}_s \\ \ddot{\theta}_h \\ \ddot{\theta}_p \end{bmatrix} + B \begin{bmatrix} \dot{\theta}_s^2 \\ \dot{\theta}_h^2 \\ \dot{\theta}_p^2 \end{bmatrix} + C \begin{bmatrix} \dot{\theta}_s\dot{\theta}_h \\ \dot{\theta}_s\dot{\theta}_p \\ \dot{\theta}_h\dot{\theta}_p \end{bmatrix} + D \begin{bmatrix} \dot{\theta}_s \\ \dot{\theta}_h \\ \dot{\theta}_p \end{bmatrix} + G = E$$

Linear State-space Model

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

4.1.1 Models with Energies

The resulting uncovering models of the system, which defines the relations between the different masses, lengths, and more. With its formulation of the kinetic and potential energies for each individual system parts (sphere, pendulum, inverted pendulum, and load). With the short notations (s: sphere, p: pendulum, and h: inverted pendulum, l: load), for the individual models parts.

Model I. *Coupled Model*

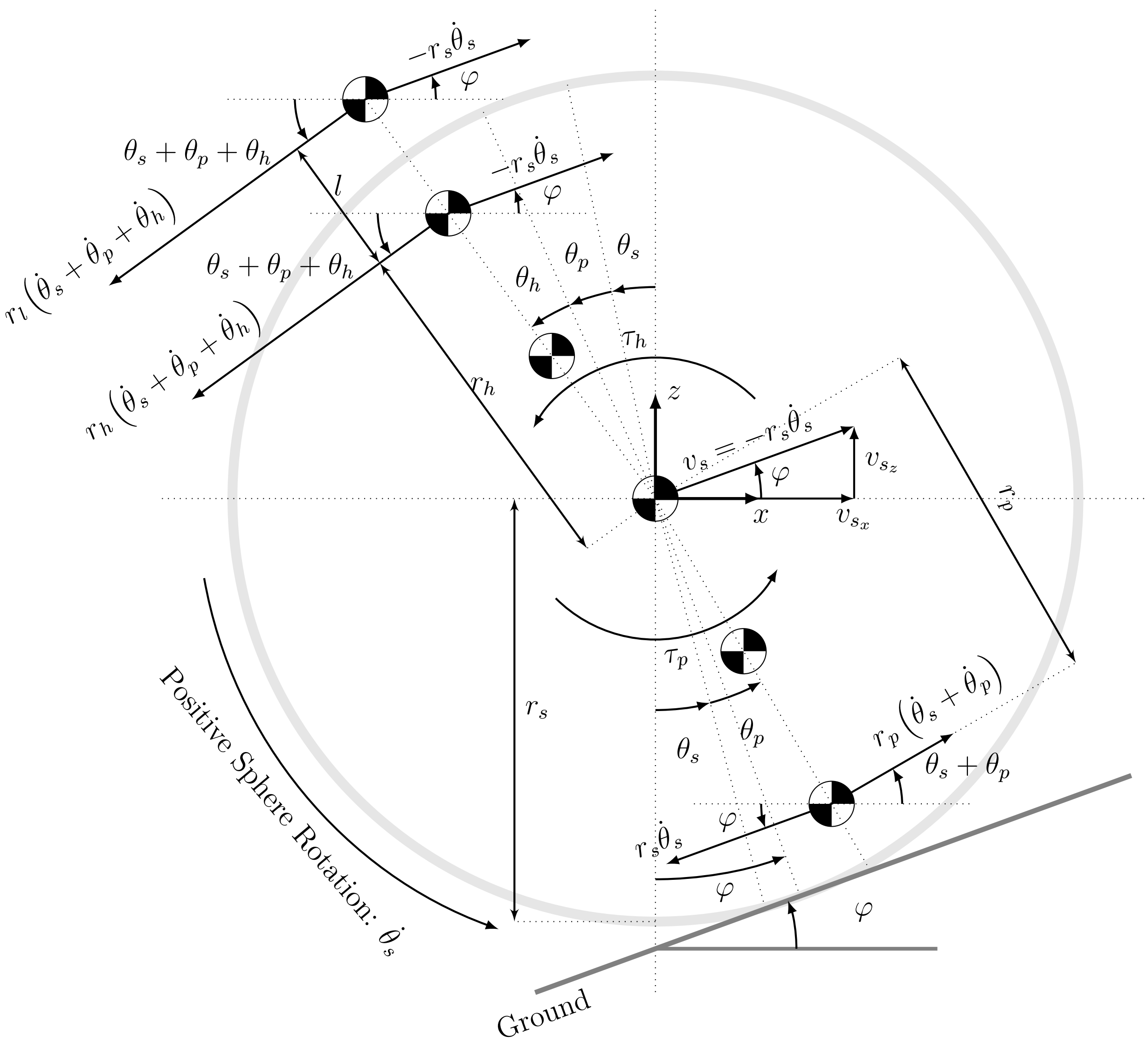


Figure 4.1: Coupled model for the back-, and forward drive of the pendulum driven spherical robot.

$$
\begin{aligned}
T_s =& \frac{1}{2} I_s \dot{\theta}_s^2 + \frac{1}{2} m_s \left(-r_s \dot{\theta}_s\right)^2 \\
T_p =& \frac{1}{2} m_p \left(\left(-r_s \dot{\theta}_s \cos(\varphi) + r_p \left(\dot{\theta}_s + \dot{\theta}_p\right) \cos(\theta_s + \theta_p)\right)^2\right) \\
&+ \frac{1}{2} m_p \left(\left(-r_s \dot{\theta}_s \sin(\varphi) + r_p \left(\dot{\theta}_s + \dot{\theta}_p\right) \sin(\theta_s + \theta_p)\right)^2\right) \\
&+ \frac{1}{2} I_p \left(\dot{\theta}_s + \dot{\theta}_p\right)^2 \\
&+ \frac{1}{2} m_{p_l} \left(\left(-r_s \dot{\theta}_s \cos(\varphi) + \frac{1}{2} r_p \left(\dot{\theta}_s + \dot{\theta}_p\right) \cos(\theta_s + \theta_p)\right)^2\right) \\
&+ \frac{1}{2} m_{p_l} \left(\left(-r_s \dot{\theta}_s \sin(\varphi) + \frac{1}{2} r_p \left(\dot{\theta}_s + \dot{\theta}_p\right) \sin(\theta_s + \theta_p)\right)^2\right) \\
&+ \frac{1}{2} I_{p_l} \left(\dot{\theta}_s + \dot{\theta}_p\right)^2 \\
T_h =& \frac{1}{2} m_h \left(\left(-r_s \dot{\theta}_s \cos(\varphi) - r_h \left(\dot{\theta}_s + \dot{\theta}_p + \dot{\theta}_h\right) \cos(\theta_s + \theta_p + \theta_h)\right)^2\right) \\
&+ \frac{1}{2} m_h \left(\left(-r_s \dot{\theta}_s \sin(\varphi) - r_h \left(\dot{\theta}_s + \dot{\theta}_p + \dot{\theta}_h\right) \sin(\theta_s + \theta_p + \theta_h)\right)^2\right) \\
&+ \frac{1}{2} I_h \left(\dot{\theta}_s + \dot{\theta}_p + \dot{\theta}_h\right)^2 \\
&+ \frac{1}{2} m_{p_u} \left(\left(-r_s \dot{\theta}_s \cos(\varphi) - \frac{1}{2} r_h \left(\dot{\theta}_s + \dot{\theta}_p + \dot{\theta}_h\right) \cos(\theta_s + \theta_p + \theta_h)\right)^2\right) \\
&+ \frac{1}{2} m_{p_u} \left(\left(-r_s \dot{\theta}_s \sin(\varphi) - \frac{1}{2} r_h \left(\dot{\theta}_s + \dot{\theta}_p + \dot{\theta}_h\right) \sin(\theta_s + \theta_p + \theta_h)\right)^2\right) \\
&+ \frac{1}{2} I_{p_u} \left(\dot{\theta}_s + \dot{\theta}_p + \dot{\theta}_h\right)^2 \\
&+ \frac{1}{2} m_l \left(\left(-r_s \dot{\theta}_s \cos(\varphi) - r_l \left(\dot{\theta}_s + \dot{\theta}_p + \dot{\theta}_h\right) \cos(\theta_s + \theta_p + \theta_h)\right)^2\right) \\
&+ \frac{1}{2} m_l \left(\left(-r_s \dot{\theta}_s \sin(\varphi) - r_l \left(\dot{\theta}_s + \dot{\theta}_p + \dot{\theta}_h\right) \sin(\theta_s + \theta_p + \theta_h)\right)^2\right) \\
&+ \frac{1}{2} I_l \left(\dot{\theta}_s + \dot{\theta}_p + \dot{\theta}_h\right)^2 \\
U_s =& -g m_s \theta_s r_s \sin(\varphi) \\
U_p =& -g m_p r_p \cos(\theta_s + \theta_p) - g m_p \theta_s r_s \sin(\varphi) \\
&- g m_{p_l} \left(\frac{1}{2} r_p\right) \cos(\theta_s + \theta_p) - g m_{p_l} \theta_s r_s \sin(\varphi) \\
U_h =& g m_h r_h \cos(\theta_s + \theta_p + \theta_h) - g m_h \theta_s r_s \sin(\varphi) \\
&+ g m_{p_u} \left(\frac{1}{2} r_h\right) \cos(\theta_s + \theta_p + \theta_h) - g m_{p_u} \theta_s r_s \sin(\varphi) \\
&+ g m_l r_l \cos(\theta_s + \theta_p + \theta_h) - g m_l \theta_s r_s \sin(\varphi) \\
Q =& \tau_h (\theta_s + \theta_h + \theta_p) + \tau_p (\theta_s + \theta_p)
\end{aligned}
$$

Model II. *Decoupled Model*

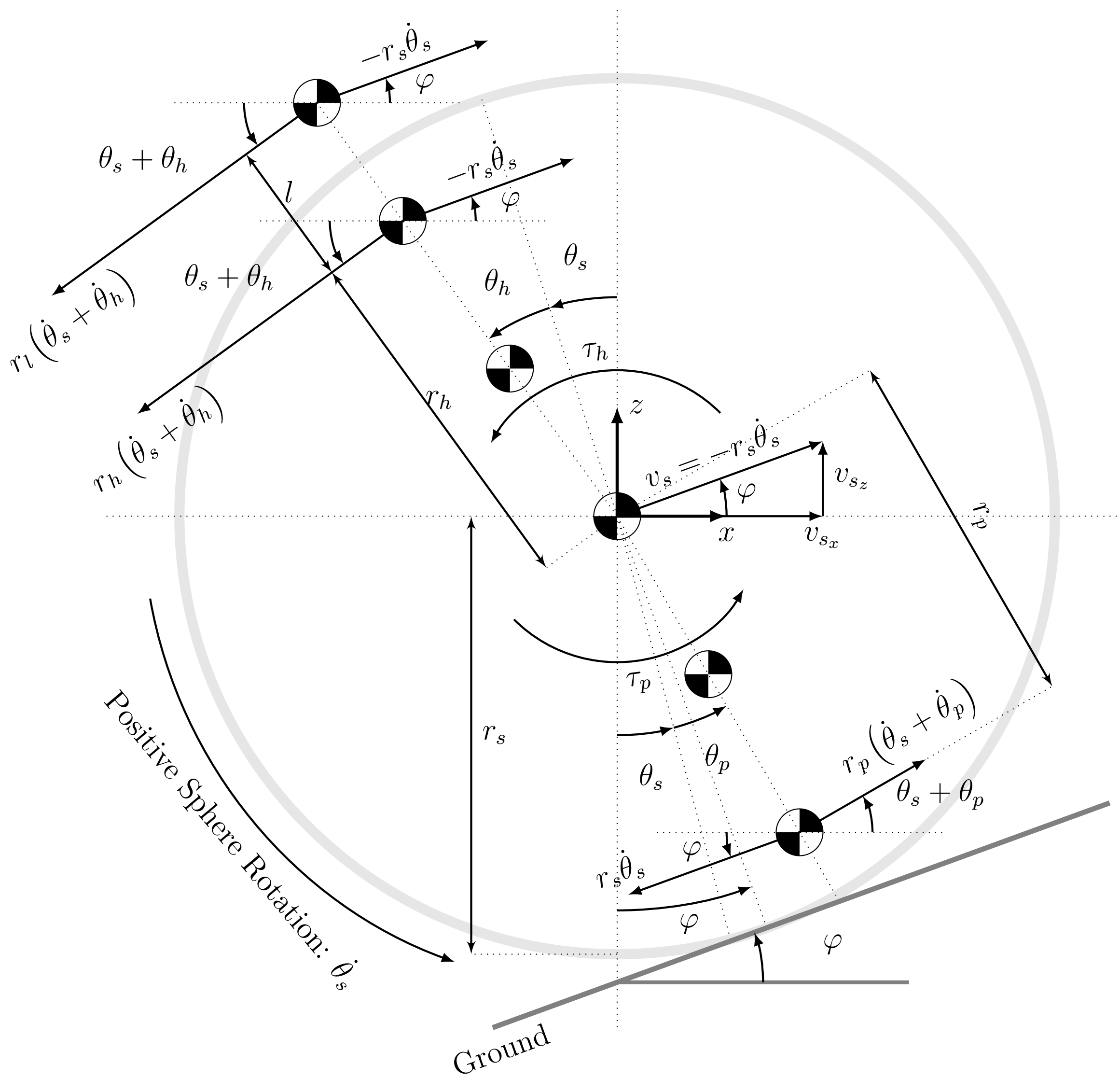


Figure 4.2: Decoupled model for the back-, and forward drive of the pendulum driven spherical robot.

$$
\begin{aligned}
T_s =& \frac{1}{2} I_s \dot{\theta}_s^2 + \frac{1}{2} m_s \left(-r_s \dot{\theta}_s\right)^2 \\
T_p =& \frac{1}{2} m_p \left(\left(-r_s \dot{\theta}_s \cos\left(\varphi\right) + r_p \left(\dot{\theta}_s + \dot{\theta}_p\right) \cos\left(\theta_s + \theta_p\right)\right)^2\right) \\
&+ \frac{1}{2} m_p \left(\left(-r_s \dot{\theta}_s \sin\left(\varphi\right) + r_p \left(\dot{\theta}_s + \dot{\theta}_p\right) \sin\left(\theta_s + \theta_p\right)\right)^2\right) \\
&+ \frac{1}{2} I_p \left(\dot{\theta}_s + \dot{\theta}_p\right)^2 \\
&+ \frac{1}{2} m_{p_l} \left(\left(-r_s \dot{\theta}_s \cos\left(\varphi\right) + \frac{1}{2} r_p \left(\dot{\theta}_s + \dot{\theta}_p\right) \cos\left(\theta_s + \theta_p\right)\right)^2\right) \\
&+ \frac{1}{2} m_{p_l} \left(\left(-r_s \dot{\theta}_s \sin\left(\varphi\right) + \frac{1}{2} r_p \left(\dot{\theta}_s + \dot{\theta}_p\right) \sin\left(\theta_s + \theta_p\right)\right)^2\right) \\
&+ \frac{1}{2} I_{p_l} \left(\dot{\theta}_s + \dot{\theta}_p\right)^2 \\
T_h =& \frac{1}{2} m_h \left(\left(-r_s \dot{\theta}_s \cos\left(\varphi\right) - r_h \left(\dot{\theta}_s + \dot{\theta}_h\right) \cos\left(\theta_s + \theta_h\right)\right)^2\right) \\
&+ \frac{1}{2} m_h \left(\left(-r_s \dot{\theta}_s \sin\left(\varphi\right) - r_h \left(\dot{\theta}_s + \dot{\theta}_h\right) \sin\left(\theta_s + \theta_h\right)\right)^2\right) \\
&+ \frac{1}{2} I_h \left(\dot{\theta}_s + \dot{\theta}_h\right)^2 \\
&+ \frac{1}{2} m_{p_u} \left(\left(-r_s \dot{\theta}_s \cos\left(\varphi\right) - \frac{1}{2} r_h \left(\dot{\theta}_s + \dot{\theta}_h\right) \cos\left(\theta_s + \theta_h\right)\right)^2\right) \\
&+ \frac{1}{2} m_{p_u} \left(\left(-r_s \dot{\theta}_s \sin\left(\varphi\right) - \frac{1}{2} r_h \left(\dot{\theta}_s + \dot{\theta}_h\right) \sin\left(\theta_s + \theta_h\right)\right)^2\right) \\
&+ \frac{1}{2} I_{p_u} \left(\dot{\theta}_s + \dot{\theta}_h\right)^2 \\
&+ \frac{1}{2} m_l \left(\left(-r_s \dot{\theta}_s \cos\left(\varphi\right) - r_l \left(\dot{\theta}_s + \dot{\theta}_h\right) \cos\left(\theta_s + \theta_h\right)\right)^2\right) \\
&+ \frac{1}{2} m_l \left(\left(-r_s \dot{\theta}_s \sin\left(\varphi\right) - r_l \left(\dot{\theta}_s + \dot{\theta}_h\right) \sin\left(\theta_s + \theta_h\right)\right)^2\right) \\
&+ \frac{1}{2} I_l \left(\dot{\theta}_s + \dot{\theta}_h\right)^2 \\
U_s =& - g m_s \theta_s r_s \sin\left(\varphi\right) \\
U_p =& - g m_p r_p \cos\left(\theta_s + \theta_p\right) - g m_p \theta_s r_s \sin\left(\varphi\right) \\
&- g m_{p_l} \left(\frac{1}{2} r_p\right) \cos\left(\theta_s + \theta_p\right) - g m_{p_l} \theta_s r_s \sin\left(\varphi\right) \\
U_h =& g m_h r_h \cos\left(\theta_s + \theta_h\right) - g m_h \theta_s r_s \sin\left(\varphi\right) \\
&+ g m_{p_u} \left(\frac{1}{2} r_h\right) \cos\left(\theta_s + \theta_h\right) - g m_{p_u} \theta_s r_s \sin\left(\varphi\right) \\
&+ g m_l r_l \cos\left(\theta_s + \theta_h\right) - g m_l \theta_s r_s \sin\left(\varphi\right) \\
Q =& \tau_h \left(\theta_s + \theta_h\right) + \tau_p \left(\theta_s + \theta_p\right)
\end{aligned}
$$

Model III. *Turning Model*

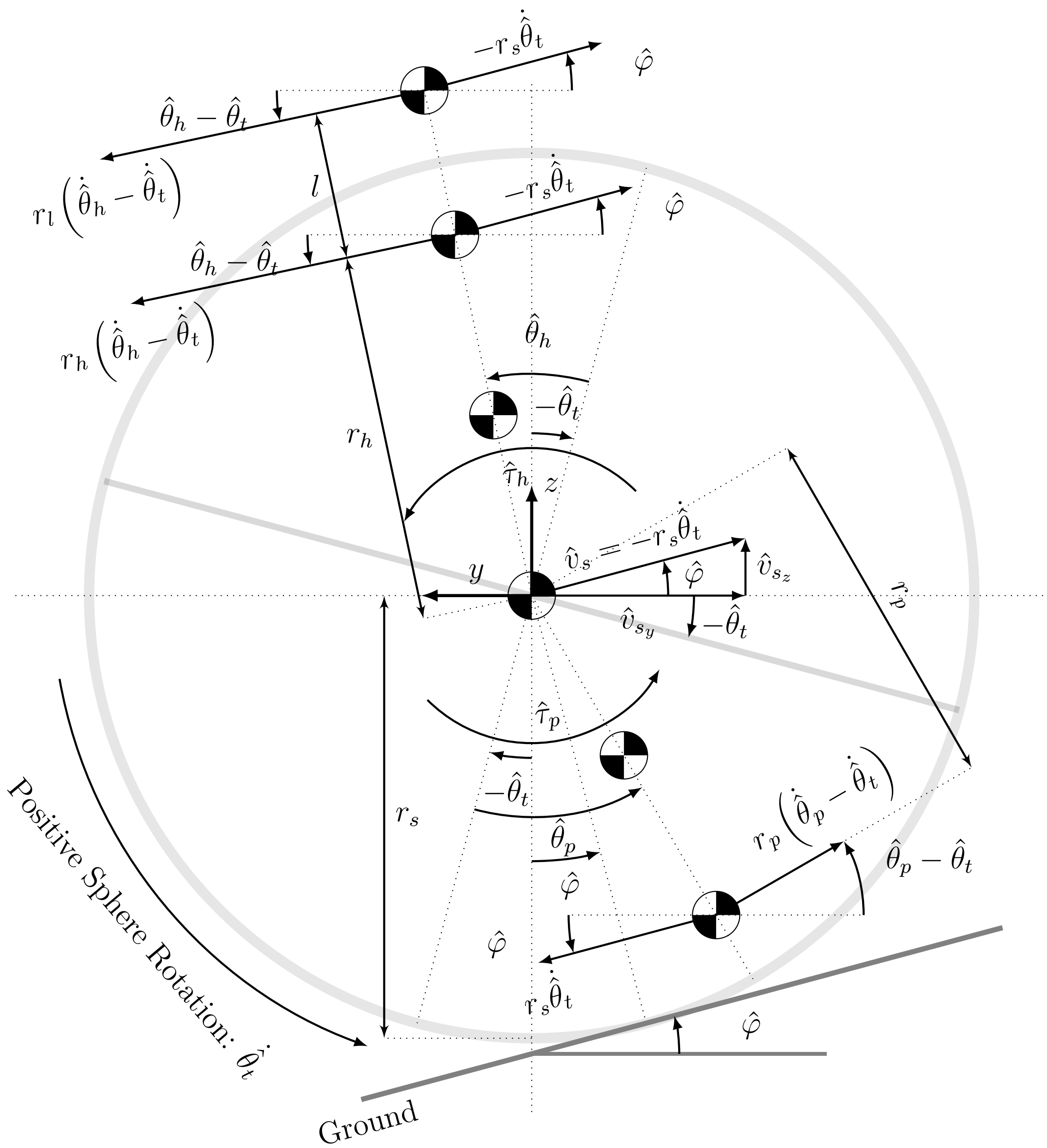


Figure 4.3: Turning model for the pendulum driven spherical robot, which describes the turning radius relationship.

$$
\begin{aligned}
T_s =& \frac{1}{2} I_s \dot{\hat{\theta}}_t^2 + \frac{1}{2} m_s \left(r_s \dot{\hat{\theta}}_t \right)^2 \\
T_p =& \frac{1}{2} \left(\left(r_s \dot{\theta}_s \cos\left(\hat{\varphi}\right) - r_p \left(\dot{\hat{\theta}}_p - \dot{\hat{\theta}}_t \right) \cos\left(\hat{\theta}_p - \hat{\theta}_t \right) \right)^2 \right) \\
&+ \frac{1}{2} \left(\left(r_s \dot{\theta}_s \sin\left(\hat{\varphi}\right) + r_p \left(\dot{\hat{\theta}}_p - \dot{\hat{\theta}}_t \right) \sin\left(\hat{\theta}_p - \hat{\theta}_t \right) \right)^2 \right) \\
&+ \frac{1}{2} I_p \left(\hat{\theta}_p - \dot{\hat{\theta}}_t \right)^2 \\
&+ \frac{1}{2} \left(\left(r_s \dot{\theta}_s \cos\left(\hat{\varphi}\right) - \frac{1}{2} r_p \left(\dot{\hat{\theta}}_p - \dot{\hat{\theta}}_t \right) \cos\left(\hat{\theta}_p - \hat{\theta}_t \right) \right)^2 \right) \\
&+ \frac{1}{2} \left(\left(r_s \dot{\theta}_s \sin\left(\hat{\varphi}\right) + \frac{1}{2} r_p \left(\dot{\hat{\theta}}_p - \dot{\hat{\theta}}_t \right) \sin\left(\hat{\theta}_p - \hat{\theta}_t \right) \right)^2 \right) \\
&+ \frac{1}{2} I_{p_l} \left(\hat{\theta}_p - \dot{\hat{\theta}}_t \right)^2 \\
T_h =& \frac{1}{2} \left(\left(r_s \dot{\theta}_s \cos\left(\hat{\varphi}\right) + r_h \left(\dot{\hat{\theta}}_h - \dot{\hat{\theta}}_t \right) \cos\left(\hat{\theta}_h - \hat{\theta}_t \right) \right)^2 \right) \\
&+ \frac{1}{2} \left(\left(r_s \dot{\theta}_s \sin\left(\hat{\varphi}\right) - r_h \left(\dot{\hat{\theta}}_h - \dot{\hat{\theta}}_t \right) \sin\left(\hat{\theta}_h - \hat{\theta}_t \right) \right)^2 \right) \\
&+ \frac{1}{2} I_h \left(\hat{\theta}_h - \hat{\theta}_t \right)^2 \\
&+ \frac{1}{2} \left(\left(r_s \dot{\theta}_s \cos\left(\hat{\varphi}\right) + \frac{1}{2} r_h \left(\dot{\hat{\theta}}_h - \dot{\hat{\theta}}_t \right) \cos\left(\hat{\theta}_h - \hat{\theta}_t \right) \right)^2 \right) \\
&+ \frac{1}{2} \left(\left(r_s \dot{\theta}_s \sin\left(\hat{\varphi}\right) - \frac{1}{2} r_h \left(\dot{\hat{\theta}}_h - \dot{\hat{\theta}}_t \right) \sin\left(\hat{\theta}_h - \hat{\theta}_t \right) \right)^2 \right) \\
&+ \frac{1}{2} I_{p_u} \left(\hat{\theta}_h - \hat{\theta}_t \right)^2 \\
&\frac{1}{2} \left(\left(r_s \dot{\theta}_s \cos\left(\hat{\varphi}\right) + r_l \left(\dot{\hat{\theta}}_h - \dot{\hat{\theta}}_t \right) \cos\left(\hat{\theta}_h - \hat{\theta}_t \right) \right)^2 \right) \\
&+ \frac{1}{2} \left(\left(r_s \dot{\theta}_s \sin\left(\hat{\varphi}\right) - r_l \left(\dot{\hat{\theta}}_h - \dot{\hat{\theta}}_t \right) \sin\left(\hat{\theta}_h - \hat{\theta}_t \right) \right)^2 \right) \\
&+ \frac{1}{2} I_l \left(\hat{\theta}_h - \hat{\theta}_t \right)^2 \\
T_t =& \frac{1}{2} I_t \dot{\hat{\theta}}_t^2 + \frac{1}{2} m_t \left(r_s \dot{\hat{\theta}}_t \right)^2 \\
U_s =& - g m_s \hat{\theta}_t r_s \sin\left(\hat{\varphi}\right) \\
U_p =& - g m_p r_p \cos\left(\hat{\theta}_p - \hat{\theta}_t \right) - g m_p \hat{\theta}_t r_s \sin\left(\hat{\varphi}\right) \\
&- g m_{p_l} \left(\frac{1}{2} r_p \right) \cos\left(\hat{\theta}_p - \hat{\theta}_t \right) - g m_{p_l} \hat{\theta}_t r_s \sin\left(\hat{\varphi}\right) \\
U_h =& g m_h r_h \cos\left(\hat{\theta}_h - \hat{\theta}_t \right) - g m_h \hat{\theta}_t r_s \sin\left(\hat{\varphi}\right) \\
&+ g m_{p_u} \left(\frac{1}{2} r_h \right) \cos\left(\hat{\theta}_h - \hat{\theta}_t \right) - g m_{p_u} \hat{\theta}_t r_s \sin\left(\hat{\varphi}\right) \\
&+ g m_l r_l \cos\left(\hat{\theta}_h - \hat{\theta}_t \right) - g m_l \hat{\theta}_t r_s \sin\left(\hat{\varphi}\right) \\
U_t =& - g m_t \hat{\theta}_t r_s \sin\left(\hat{\varphi}\right) \\
Q =& \hat{\tau}_h \left(\hat{\theta}_h - \hat{\theta}_t \right) + \hat{\tau}_p \left(\hat{\theta}_p - \hat{\theta}_t \right)
\end{aligned}
$$

4.1.2 Nonlinear State-space Model

The resulting matrices of the nonlinear state-space models. Which is given in the format by equation (3.3), with $A_{i,j}$ is the element in the i row, and j column of matrix A.

Model IV. *Nonlinear Coupled Model*

$$
\begin{aligned}
A_{1,1} = {} & I_h + I_l + I_p + I_{p_l} + I_{p_u} + I_s + \left(m_h + \frac{1}{4}m_{p_u}\right) r_h^2 + m_l r_l^2 + \left(m_p + \frac{1}{4}m_{p_l}\right) r_p^2 \\
& + \left(m_h + m_l + m_p + m_{p_l} + m_{p_u} + m_s\right) r_s^2 \\
& + \left(2m_h r_h + m_{p_u} r_h + 2m_l r_l\right) r_s \cos\left(\varphi - \theta_h - \theta_p - \theta_s\right) \\
& - \left(2m_p + m_{p_l}\right) r_p r_s \cos\left(\varphi - \theta_p - \theta_s\right) \\
A_{1,2} = {} & I_h + I_l + I_{p_u} + \left(m_h + \frac{1}{4}m_{p_u}\right) r_h^2 + m_l r_l^2 \\
& + \left(m_h r_h + \frac{1}{2}m_{p_u} r_h + m_l r_l\right) r_s \cos\left(\varphi - \theta_h - \theta_p - \theta_s\right) \\
A_{1,3} = {} & I_h + I_l + I_p + I_{p_l} + I_{p_u} + \left(m_h + \frac{1}{4}m_{p_u}\right) r_h^2 + m_l r_l^2 + \left(m_p + \frac{1}{4}m_{p_l}\right) r_p^2 \\
& + \left(m_h r_h + \frac{1}{2}m_{p_u} r_h + m_l r_l\right) r_s \cos\left(\varphi - \theta_h - \theta_p - \theta_s\right) \\
& - \left(m_p + \frac{1}{2}m_{p_l}\right) r_p r_s \cos\left(\varphi - \theta_p - \theta_s\right) \\
A_{2,1} = {} & I_h + I_l + I_{p_u} + \left(m_h + \frac{1}{4}m_{p_u}\right) r_h^2 + m_l r_l^2 \\
& + \left(m_h r_h + \frac{1}{2}m_{p_u} r_h + m_l r_l\right) r_s \cos\left(\varphi - \theta_h - \theta_p - \theta_s\right) \\
A_{2,2} = {} & I_h + I_l + I_{p_u} + \left(m_h + \frac{1}{4}m_{p_u}\right) r_h^2 + m_l r_l^2 \\
A_{2,3} = {} & I_h + I_l + I_{p_u} + \left(m_h + \frac{1}{4}m_{p_u}\right) r_h^2 + m_l r_l^2 \\
A_{3,1} = {} & I_h + I_l + I_p + I_{p_l} + I_{p_u} + \left(m_h + \frac{1}{4}m_{p_u}\right) r_h^2 + m_l r_l^2 + \left(m_p + \frac{1}{4}m_{p_l}\right) r_p^2 \\
& + \left(m_h r_h + \frac{1}{2}m_{p_u} r_h + m_l r_l\right) r_s \cos\left(\varphi - \theta_h - \theta_p - \theta_s\right) \\
& - \left(m_p + \frac{1}{2}m_{p_l}\right) r_p r_s \cos\left(\varphi - \theta_p - \theta_s\right) \\
A_{3,2} = {} & I_h + I_l + I_{p_u} + \left(m_h + \frac{1}{4}m_{p_u}\right) r_h^2 + m_l r_l^2 \\
A_{3,3} = {} & I_h + I_l + I_p + I_{p_l} + I_{p_u} + \left(m_h + \frac{1}{4}m_{p_u}\right) r_h^2 + m_l r_l^2 + \left(m_p + \frac{1}{4}m_{p_l}\right) r_p^2
\end{aligned}
$$

$$
\begin{aligned}
B_{1,1} =\ & \left(m_h r_h + \frac{1}{2} m_{p_u} r_h + m_l r_l\right) r_s \sin\left(\varphi - \theta_h - \theta_p - \theta_s\right) \\
& - \left(m_p + \frac{1}{2} m_{p_l}\right) r_p r_s \sin\left(\varphi - \theta_p - \theta_s\right) \\
B_{1,2} =\ & \left(m_h r_h + \frac{1}{2} m_{p_u} r_h + m_l r_l\right) r_s \sin\left(\varphi - \theta_h - \theta_p - \theta_s\right) \\
B_{1,3} =\ & \left(m_h r_h + \frac{1}{2} m_{p_u} r_h + m_l r_l\right) r_s \sin\left(\varphi - \theta_h - \theta_p - \theta_s\right) \\
& - \left(m_p + \frac{1}{2} m_{p_l}\right) r_p r_s \sin\left(\varphi - \theta_p - \theta_s\right) \\
C_{1,1} =\ & \left(2 m_h r_h + m_{p_u} r_h + 2 m_l r_l\right) r_s \sin\left(\varphi - \theta_h - \theta_p - \theta_s\right) \\
C_{1,2} =\ & \left(2 m_h r_h + m_{p_u} r_h + 2 m_l r_l\right) r_s \sin\left(\varphi - \theta_h - \theta_p - \theta_s\right) \\
& - \left(2 m_p + m_{p_l}\right) r_p r_s \sin\left(\varphi - \theta_p - \theta_s\right) \\
C_{1,3} =\ & \left(2 m_h r_h + m_{p_u} r_h + 2 m_l r_l\right) r_s \sin\left(\varphi - \theta_h - \theta_p - \theta_s\right) \\
G_{1,1} =\ & g \left(m_p + \frac{1}{2} m_{p_l}\right) r_p \sin\left(\theta_p + \theta_s\right) \\
& - g \left(m_h + m_l + m_p + m_{p_l} + m_{p_u} + m_s\right) r_s \sin\left(\varphi\right) \\
& - g \left(m_h r_h + \frac{1}{2} m_{p_u} r_h + m_l r_l\right) \sin\left(\theta_h + \theta_p + \theta_s\right) \\
G_{2,1} =\ & - g \left(m_h r_h + \frac{1}{2} m_{p_u} r_h + m_l r_l\right) \sin\left(\theta_h + \theta_p + \theta_s\right) \\
G_{3,1} =\ & g \left(m_p + \frac{1}{2} m_{p_l}\right) r_p \sin\left(\theta_p + \theta_s\right) \\
& - g \left(m_h r_h + \frac{1}{2} m_{p_u} r_h + m_l r_l\right) \sin\left(\theta_h + \theta_p + \theta_s\right) \\
E_{1,1} =\ & \tau_h + \tau_p \\
E_{2,1} =\ & \tau_h \\
E_{3,1} =\ & \tau_h + \tau_p
\end{aligned}
$$

Model V. *Nonlinear Decoupled Model*

$$
\begin{aligned}
A_{1,1} &= I_h + I_l + I_p + I_{p_l} + I_{p_u} + I_s + \left(m_h + \frac{1}{4}m_{p_u}\right) r_h^2 + m_l r_l^2 + \left(m_p + \frac{1}{4}m_{p_l}\right) r_p^2 \\
&\quad + \left(m_h + m_l + m_p + m_{p_l} + m_{p_u} + m_s\right) r_s^2 \\
&\quad + \left(2m_h r_h + m_{p_u} r_h + 2m_l r_l\right) r_s \cos\left(\varphi - \theta_h - \theta_s\right) \\
&\quad - \left(2m_p + m_{p_l}\right) r_p r_s \cos\left(\varphi - \theta_p - \theta_s\right) \\
A_{1,2} &= I_h + I_l + I_{p_u} + \left(m_h + \frac{1}{4}m_{p_u}\right) r_h^2 + m_l r_l^2 \\
&\quad + \left(m_h r_h + \frac{1}{2}m_{p_u} r_h + m_l r_l\right) r_s \cos\left(\varphi - \theta_h - \theta_s\right) \\
A_{1,3} &= I_p + I_{p_l} + \left(m_p + \frac{1}{4}m_{p_l}\right) r_p^2 - \left(m_p + \frac{1}{2}m_{p_l}\right) r_p r_s \cos\left(\varphi - \theta_p - \theta_s\right) \\
A_{2,1} &= I_h + I_l + I_{p_u} + \left(m_h + \frac{1}{4}m_{p_u}\right) r_h^2 + m_l r_l^2 \\
&\quad + \left(m_h r_h + \frac{1}{2}m_{p_u} r_h + m_l r_l\right) r_s \cos\left(\varphi - \theta_h - \theta_s\right) \\
A_{2,2} &= I_h + I_l + I_{p_u} + \left(m_h + \frac{1}{4}m_{p_u}\right) r_h^2 + m_l r_l^2 \\
A_{3,1} &= I_p + I_{p_l} + \left(m_p + \frac{1}{4}m_{p_l}\right) r_p^2 - \left(m_p + \frac{1}{2}m_{p_l}\right) r_p r_s \cos\left(\varphi - \theta_p - \theta_s\right) \\
A_{3,3} &= I_p + I_{p_l} + \left(m_p + \frac{1}{4}m_{p_l}\right) r_p^2 \\
B_{1,1} &= \left(m_h r_h + \frac{1}{2}m_{p_u} r_h + m_l r_l\right) r_s \sin\left(\varphi - \theta_h - \theta_s\right) \\
&\quad - \left(m_p + \frac{1}{2}m_{p_l}\right) r_p r_s \sin\left(\varphi - \theta_p - \theta_s\right) \\
B_{1,2} &= \left(m_h r_h + \frac{1}{2}m_{p_u} r_h + m_l r_l\right) r_s \sin\left(\varphi - \theta_h - \theta_s\right) \\
B_{1,3} &= -\left(m_p + \frac{1}{2}m_{p_l}\right) r_p r_s \sin\left(\varphi - \theta_p - \theta_s\right) \\
C_{1,1} &= \left(2m_h r_h + m_{p_u} r_h + 2m_l r_l\right) r_s \sin\left(\varphi - \theta_h - \theta_s\right) \\
C_{1,2} &= -\left(2m_p + m_{p_l}\right) r_p r_s \sin\left(\varphi - \theta_p - \theta_s\right) \\
G_{1,1} &= g\left(m_p + \frac{1}{2}m_{p_l}\right) r_p \sin\left(\theta_p + \theta_s\right) - g\left(m_h r_h + \frac{1}{2}m_{p_u} r_h + m_l r_l\right) \sin\left(\theta_h + \theta_s\right) \\
&\quad - g\left(m_h + m_l + m_p + m_{p_l} + m_{p_u} + m_s\right) r_s \sin\left(\varphi\right) \\
G_{2,1} &= -g\left(m_h r_h + \frac{1}{2}m_{p_u} r_h + m_l r_l\right) \sin\left(\theta_h + \theta_s\right) \\
G_{3,1} &= g\left(m_p + \frac{1}{2}m_{p_l}\right) r_p \sin\left(\theta_p + \theta_s\right) \\
E_{1,1} &= \tau_h + \tau_p \\
E_{2,1} &= \tau_h \\
E_{3,1} &= \tau_p
\end{aligned}
$$

4.1.3 Linear State-space Model

The resulting matrices of the linear state-space models. Which is given in the format by the equations (2.31)-(2.32), with $A_{i,j}$ is the element in the i row, and j column of matrix A.

Model VI. *Linear Coupled Model*

$$
\begin{aligned}
A\left(i,1\right) =& - a_{i,1}\left[b_{1,1}\right]_{x0,u_0}\left(2\dot{\theta}_{s0}\right) - a_{i,1}\left[c_{1,1}\right]_{x0,u_0}\left(\dot{\theta}_{h0}\right) - a_{i,1}\left[c_{1,2}\right]_{x0,u_0}\left(\dot{\theta}_{p0}\right)\\
A\left(i,2\right) =& - a_{i,1}\left[b_{1,2}\right]_{x0,u_0}\left(2\dot{\theta}_{h0}\right) - a_{i,1}\left[c_{1,1}\right]_{x0,u_0}\left(\dot{\theta}_{s0}\right) - a_{i,1}\left[c_{1,3}\right]_{x0,u_0}\left(\dot{\theta}_{p0}\right)\\
A\left(i,3\right) =& - a_{i,1}\left[b_{1,3}\right]_{x0,u_0}\left(2\dot{\theta}_{p0}\right) - a_{i,1}\left[c_{1,2}\right]_{x0,u_0}\left(\dot{\theta}_{s0}\right) - a_{i,1}\left[c_{1,3}\right]_{x0,u_0}\left(\dot{\theta}_{h0}\right)\\
A\left(i,4\right) =& a_{i,1}\left[\left(m_h r_h + \frac{1}{2}m_{p_u} r_h + m_l r_l\right) r_s - \left(m_p + \frac{1}{2}m_{p_l}\right) r_p r_s\right]\left(\dot{\theta}^2_{s0}\right)\\
&+ a_{i,1}\left[\left(m_h r_h + \frac{1}{2}m_{p_u} r_h + m_l r_l\right) r_s\right]\left(\dot{\theta}^2_{h0}\right)\\
&+ a_{i,1}\left[\left(m_h r_h + \frac{1}{2}m_{p_u} r_h + m_l r_l\right) r_s - \left(m_p + \frac{1}{2}m_{p_l}\right) r_p r_s\right]\left(\dot{\theta}^2_{p0}\right)\\
&+ a_{i,1}\left[\left(2m_h r_h + m_{p_u} r_h + 2m_l r_l\right) r_s\right]\left(\dot{\theta}_{s0}\dot{\theta}_{h0}\right)\\
&+ a_{i,1}\left[\left(2m_h r_h + m_{p_u} r_h + 2m_l r_l\right) r_s - \left(2m_p + m_{p_l}\right) r_p r_s\right]\left(\dot{\theta}_{s0}\dot{\theta}_{p0}\right)\\
&+ a_{i,1}\left[\left(2m_h r_h + m_{p_u} r_h + 2m_l r_l\right) r_s\right]\left(\dot{\theta}_{h0}\dot{\theta}_{p0}\right)\\
&+ a_{i,1}\left[g\left(m_h r_h + \frac{1}{2}m_{p_u} r_h + m_l r_l\right) - g\left(m_p + \frac{1}{2}m_{p_l}\right) r_p\right]\\
&+ a_{i,2}\left[g\left(m_h r_h + \frac{1}{2}m_{p_u} r_h + m_l r_l\right)\right]\\
&+ a_{i,3}\left[g\left(m_h r_h + \frac{1}{2}m_{p_u} r_h + m_l r_l\right) - g\left(m_p + \frac{1}{2}m_{p_l}\right) r_p\right]\\
A\left(i,5\right) =& a_{i,1}\left[\left(m_h r_h + \frac{1}{2}m_{p_u} r_h + m_l r_l\right) r_s\right]\left(\dot{\theta}^2_{s0} + \dot{\theta}^2_{h0} + \dot{\theta}^2_{p0}\right)\\
&+ a_{i,1}\left[\left(2m_h r_h + m_{p_u} r_h + 2m_l r_l\right) r_s\right]\left(\dot{\theta}_{s0}\dot{\theta}_{h0} + \dot{\theta}_{s0}\dot{\theta}_{p0} + \dot{\theta}_{h0}\dot{\theta}_{p0}\right)\\
&+ \left(a_{i,1} + a_{i,2} + a_{i,3}\right)\left[g\left(m_h r_h + \frac{1}{2}m_{p_u} r_h + m_l r_l\right)\right]\\
A\left(i,6\right) =& A\left(i,4\right)\\
B\left(i,1\right) =& a_{i,1} + a_{i,2} + a_{i,3}\\
B\left(i,2\right) =& a_{i,1} + a_{i,3}\\
& \textit{Where the integral } i \textit{ is for } i = 1,2,3.\\
A\left(4,1\right) =& 1\\
A\left(5,2\right) =& 1\\
A\left(6,3\right) =& 1
\end{aligned}
$$

Disturbance matrix for the slope of the ground.

$$
\begin{aligned}
B_2(i,1) = &- a_{i,1}\left[\left(m_h r_h + \frac{1}{2}m_{p_u} r_h + m_l r_l\right) r_s - \left(m_p + \frac{1}{2}m_{p_l}\right) r_p r_s\right]\left(\dot{\theta}_{s_0}^2\right) \\
&- a_{i,1}\left[\left(m_h r_h + \frac{1}{2}m_{p_u} r_h + m_l r_l\right) r_s\right]\left(\dot{\theta}_{h_0}^2\right) \\
&- a_{i,1}\left[\left(m_h r_h + \frac{1}{2}m_{p_u} r_h + m_l r_l\right) r_s - \left(m_p + \frac{1}{2}m_{p_l}\right) r_p r_s\right]\left(\dot{\theta}_{p_0}^2\right) \\
&- a_{i,1}\left[\left(2m_h r_h + m_{p_u} r_h + 2m_l r_l\right) r_s\right]\left(\dot{\theta}_{s_0}\dot{\theta}_{h_0}\right) \\
&- a_{i,1}\left[\left(2m_h r_h + m_{p_u} r_h + 2m_l r_l\right) r_s - \left(2m_p + m_{p_l}\right) r_p r_s\right]\left(\dot{\theta}_{s_0}\dot{\theta}_{p_0}\right) \\
&- a_{i,1}\left[\left(2m_h r_h + m_{p_u} r_h + 2m_l r_l\right) r_s\right]\left(\dot{\theta}_{h_0}\dot{\theta}_{p_0}\right) \\
&+ a_{i,1}\left[g\left(m_h + m_l + m_p + m_{p_l} + m_{p_u} + m_s\right) r_s\right]
\end{aligned}
$$

Where the integral i is for $i = 1, 2, 3$.

Model VII. *Linear Decoupled Model*

$$
\begin{aligned}
A\left(i,1\right) =& - a_{i,1}\left[b_{1,1}\right]_{x0,u_0}\left(2\dot{\theta}_{s_0}\right) - a_{i,1}\left[c_{1,1}\right]_{x0,u_0}\left(\dot{\theta}_{h_0}\right) - a_{i,1}\left[c_{1,2}\right]_{x0,u_0}\left(\dot{\theta}_{p0}\right)\\
A\left(i,2\right) =& - a_{i,1}\left[b_{1,2}\right]_{x0,u_0}\left(2\dot{\theta}_{h_0}\right) - a_{i,1}\left[c_{1,1}\right]_{x0,u_0}\left(\dot{\theta}_{s_0}\right)\\
A\left(i,3\right) =& - a_{i,1}\left[b_{1,3}\right]_{x0,u_0}\left(2\dot{\theta}_{p_0}\right) - a_{i,1}\left[c_{1,2}\right]_{x0,u_0}\left(\dot{\theta}_{s_0}\right)\\
A\left(i,4\right) =& a_{i,1}\left[\left(m_h r_h + \frac{1}{2}m_{p_u}r_h + m_l r_l\right)r_s - \left(m_p + \frac{1}{2}m_{p_l}\right)r_p r_s\right]\left(\dot{\theta}^2_{s_0}\right)\\
&+ a_{i,1}\left[\left(m_h r_h + \frac{1}{2}m_{p_u}r_h + m_l r_l\right)r_s\right]\left(\dot{\theta}^2_{h_0}\right)\\
&- a_{i,1}\left[\left(m_p + \frac{1}{2}m_{p_l}\right)r_p r_s\right]\left(\dot{\theta}^2_{p_0}\right)\\
&+ a_{i,1}\left[\left(2m_h r_h + m_{p_u}r_h + 2m_l r_l\right)r_s\right]\left(\dot{\theta}_{s_0}\dot{\theta}_{h_0}\right)\\
&- a_{i,1}\left[\left(2m_p + m_{p_l}\right)r_p r_s\right]\left(\dot{\theta}_{s_0}\dot{\theta}_{p_0}\right)\\
&+ a_{i,1}\left[g\left(m_h r_h + \frac{1}{2}m_{p_u}r_h + m_l r_l\right) - g\left(m_p + \frac{1}{2}m_{p_l}\right)r_p\right]\\
&+ a_{i,2}\left[g\left(m_h r_h + \frac{1}{2}m_{p_u}r_h + m_l r_l\right)\right]\\
&- a_{i,3}\left[g\left(m_p + \frac{1}{2}m_{p_l}\right)r_p\right]\\
A\left(i,5\right) =& a_{i,1}\left[\left(m_h r_h + \frac{1}{2}m_{p_u}r_h + m_l r_l\right)r_s\right]\left(\dot{\theta}^2_{s_0} + \dot{\theta}^2_{h_0}\right)\\
&+ a_{i,1}\left[\left(2m_h r_h + m_{p_u}r_h + 2m_l r_l\right)r_s\right]\left(\dot{\theta}_{s_0}\dot{\theta}_{h_0}\right)\\
&+ \left(a_{i,1} + a_{i,2}\right)\left[g\left(m_h r_h + \frac{1}{2}m_{p_u}r_h + m_l r_l\right)\right]\\
A\left(i,6\right) =& - a_{i,1}\left[\left(m_p + \frac{1}{2}m_{p_l}\right)r_p r_s\right]\left(\dot{\theta}^2_{s_0} + \dot{\theta}^2_{p_0}\right)\\
&- a_{i,1}\left[\left(2m_p + m_{p_l}\right)r_p r_s\right]\left(\dot{\theta}_{s_0}\dot{\theta}_{p_0}\right)\\
&- \left(a_{i,1} + a_{i,3}\right)\left[g\left(m_p + \frac{1}{2}m_{p_l}\right)r_p\right]\\
B\left(i,1\right) =& a_{i,1} + a_{i,2}\\
B\left(i,2\right) =& a_{i,1} + a_{i,3}
\end{aligned}
$$

Where the integral i is for $i = 1, 2, 3$.

$$
\begin{aligned}
A\left(4,1\right) =& 1\\
A\left(5,2\right) =& 1\\
A\left(6,3\right) =& 1
\end{aligned}
$$

Disturbance matrix for the slope of the ground.

$$
\begin{aligned}
B_2\left(i,1\right) = &- a_{i,1}\left[\left(m_h r_h + \frac{1}{2}m_{p_u} r_h + m_l r_l\right) r_s - \left(m_p + \frac{1}{2}m_{p_l}\right) r_p r_s\right]\left(\dot{\theta}_{s_0}^2\right) \\
&- a_{i,1}\left[\left(m_h r_h + \frac{1}{2}m_{p_u} r_h + m_l r_l\right) r_s\right]\left(\dot{\theta}_{h_0}^2\right) \\
&+ a_{i,1}\left[\left(m_p + \frac{1}{2}m_{p_l}\right) r_p r_s\right]\left(\dot{\theta}_{p_0}^2\right) \\
&- a_{i,1}\left[\left(2m_h r_h + m_{p_u} r_h + 2m_l r_l\right) r_s\right]\left(\dot{\theta}_{s_0}\dot{\theta}_{h_0}\right) \\
&+ a_{i,1}\left[\left(2m_p + m_{p_l}\right) r_p r_s\right]\left(\dot{\theta}_{s_0}\dot{\theta}_{p_0}\right) \\
&+ a_{i,1}\left[g\left(m_h + m_l + m_p + m_{p_l} + m_{p_u} + m_s\right) r_s\right] \\
&\textit{Where the integral } i \textit{ is for } i = 1,2,3.
\end{aligned}
$$

4.2 State-space Model Analysis

Here the results from analyzing the linear and nonlinear models, shown in the chapter 4.1, will be shown. The different results surrounding properties of the models will be explained and shown if they hold or not. Theoretical background of the properties and analysis tools are explained in *Model Analysis*, chapter 2.7.

4.2.1 Linear State-space Analysis

From the approach for linear state-space model analysis described in *Linear State-space Model Analysis*, chapter 2.7.1. Analysis of the models before having a controller, with the two linear models of *coupled model* VI, and *decoupled model* VII. Then gives the following analysis results with the nominal parameter values given in the tables 3.1-3.3.

Eigenvalues Analysis The analysis of the eigenvalues of the linear state-space models, gave that the eigenvalues for the two different models become as shown in table 4.1.

Nominal Coupled Model's Eigenvalues		Nominal Decoupled Model's Eigenvalues
-11.9065		$-4.6597 \cdot 10^{-10} + 7.9557i$
$+11.9065$		$-4.6597 \cdot 10^{-10} - 7.9557i$
-6.1074		$+4.4873$
$+6.1074$		-4.4873
$-3.0130 \cdot 10^{-8}$		$+1.1796 \cdot 10^{-7}$
$+3.0130 \cdot 10^{-8}$		$-1.1796 \cdot 10^{-7}$

Table 4.1: The nominal model's eigenvalues.

Reachability & State Controllability The controllability for the two models is evaluated with the controllability matrix in equation (2.41). This gives for the *coupled model* VI, that the model is not controllable due to that the controllability matrix is rank deficient. Which raises the question if the uncontrollable states are stabilizable or not. However, for the *decoupled model* VII, the controllability matrix is of full rank and therefore is controllable.

State Stabilizability As described in the definition of state stabilizability, definition 2.5, the uncontrollable states need to be stable for the models to be stabilizable. The *decoupled model* VII are controllable and therefore is also stabilizable. However, for the *coupled model* VI which is not controllable, with a rank deficiency of the controllability matrix, with two states being uncontrollable. From the eigenvalues of the model, shown in the table 4.1, it can be seen that the eigenvalues are given in pairs, with a stable and a unstable one. This gives that the *coupled model* VI is not stabilizable and in so can't be guaranteed to be stable while controlled.

State Observability The observability for the two models is evaluated with the observability matrix in equation (2.42). For both models *coupled model* VI, and *decoupled model* VII, the observability matrix is of full rank. This gives that both models are observable.

State Detectability As described in the definition of state detectability, definition 2.7, the unstable states need to be observable for the models to be detectable. With that both models are observable, both systems are detectable.

4.2.2 Nonlinear State-space Analysis

As described in *Nonlinear State-space Model Analysis*, chapter 2.7.2. It can be hard to analyze a nonlinear state-space model, due to various phenomenon's that may occur by the non-linearity. Many methods exists to analyze a nonlinear system, such as the named *Lyapunov's Stability theorem*, theorem 2.2. However, because it may be hard to find a suitable Lyapunov function, therefore as mentioned in *Nonlinear Model Analysis*, chapter 3.3.2. It will be assumed that the nonlinear state-space will behave as the linear state-space in a neighborhood around the stationary point.

4.3 Linear Control Algorithms

Here the simulation results from the linear control algorithms will be presented, the angular accelerations, angular velocities, angles, and the control input signals will be presented. By that the linear model analysis results from the chapter 4.2.1, only the *decoupled model* is used, for that the *coupled model* is not guaranteed to be stabilizable. Two of each simulation will be done, one is for noise free situation and the other is with measurement noise. For the output feedback control algorithms, the C and D matrices of the linear state-space model is given by the equation (3.20).

The simulations results are shown in a certain order, with that five different plots are done with the first one from above. That first from above shows the angular accelerations. Then the second shows the angular velocity The third shows the angles of the sphere (θ_s), and pendulum (θ_p). The fourth shows the angle of the inverted pendulum (θ_h). While the fifth shows the control signals in torque, namely the torque input of pendulum (τ_p), and inverted pendulum (τ_h).

4.3.1 Linear Quadratic Controllers (LQR, LQI, LQG)

The three different ways to define a linear quadratic controller resulting simulations, will be presented with and without measurement noise. The approach to get these results are given in *Linear Quadratic Controllers*, chapter 3.4.1.1.

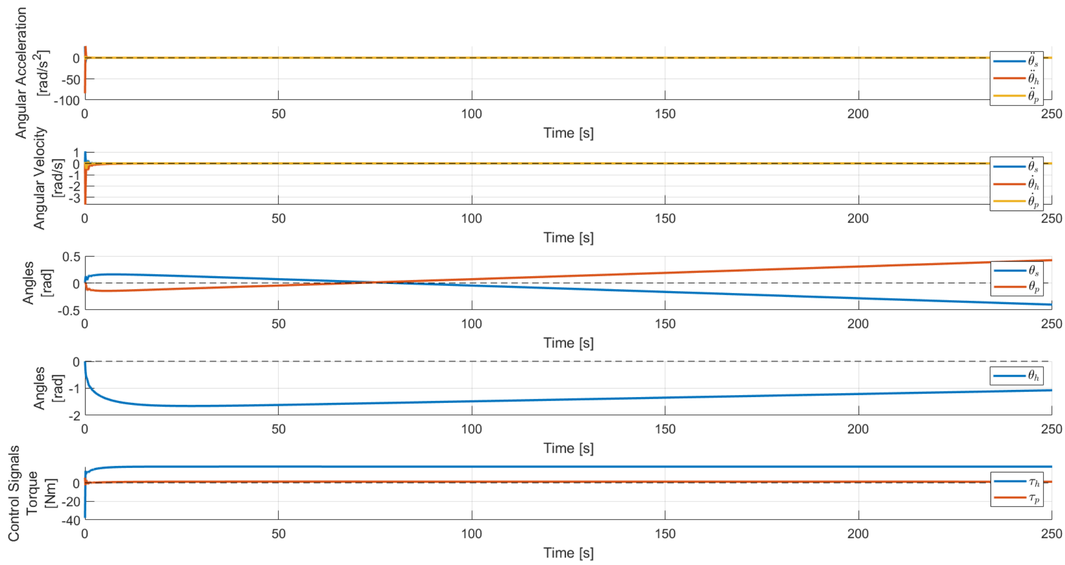


Figure 4.4: LQR Controller, simulation with the nonlinear *Decoupled model* without measurement noise.

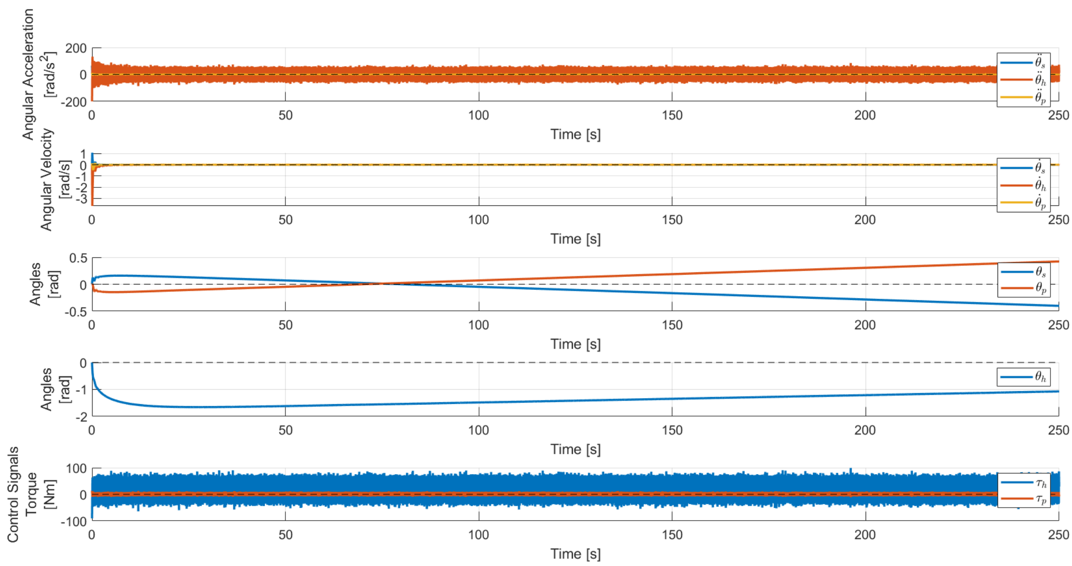


Figure 4.5: LQR Controller, simulation with the nonlinear *Decoupled model* with measurement noise.

From the simulations, Figures 4.4 and 4.5, one can observe how well the LQR stabilize the system with and without measurement noise. With that the LQR is upholding nominal stability (NS) and nominal performance (NP). From the simulation without noise, Figure 4.4, one can observe that the inverted pendulum angle is converging towards zero while moving forward. With some rather high values for the angle of the inverted pendulum before beginning to converge towards zero. It does have a steady control signal compared to with noise, Figure 4.5. Which shows that with and without noise, the angles are behaving similar. However, with noise the control signal and angular acceleration is affected in rather sporadic changes. With that the noise effect on the inverted pendulum seem to be the most problematic.

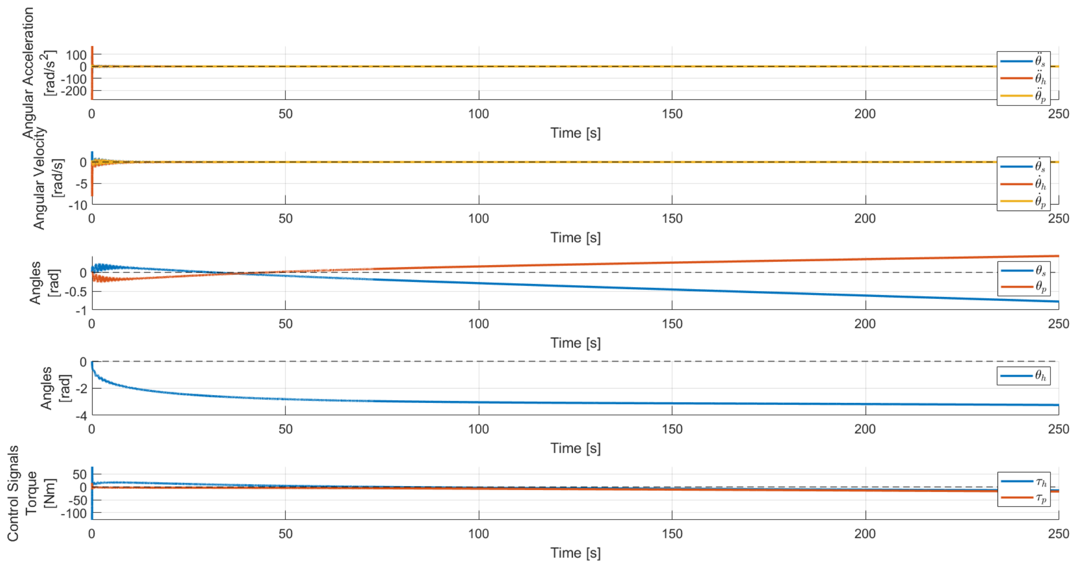


Figure 4.6: LQI Controller, simulation with the nonlinear *Decoupled model* without measurement noise.

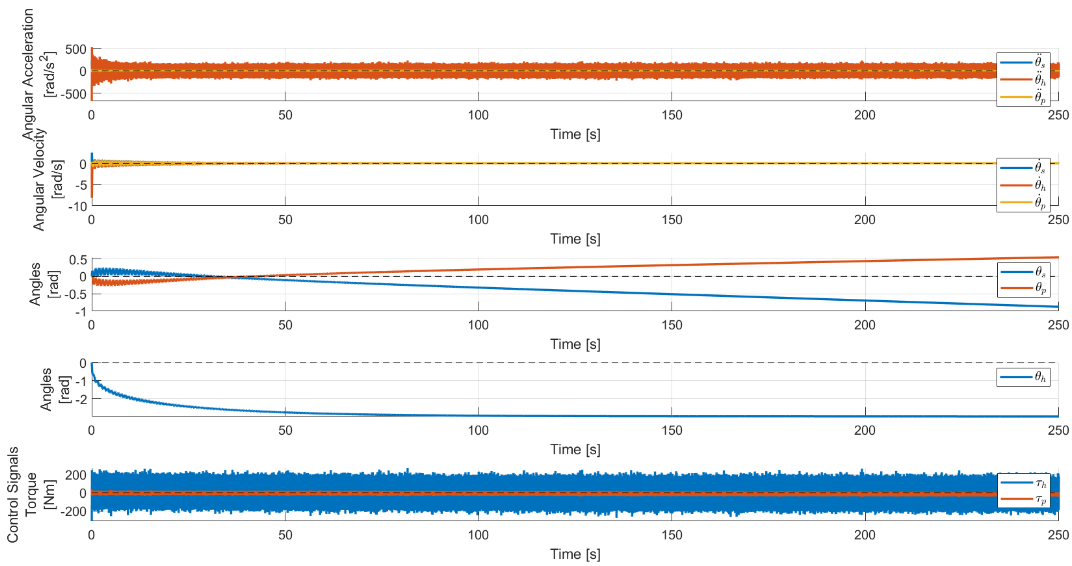


Figure 4.7: LQI Controller, simulation with the nonlinear *Decoupled model* with measurement noise.

From the simulations, Figures 4.6 and 4.7, one can observe how well the LQI stabilize the system with and without measurement noise. With that the controller is giving nominal stability (NS), even though it does not seem to be performing as intended. From the simulation without noise, Figure 4.6, one can observe that the angle of the inverted pendulum is around the value of π radian. Which then may be interpreted as that the inverted pendulum is hanging as a pendulum. Giving that the LQI could not stabilize the load. Similar to the LQR result, with noise the angular acceleration and control signal is affected in rather sporadic changes, as can be observed from Figure 4.7. With that the noise effect on the inverted pendulum seem to be the most problematic. However, here it has higher magnitude for the angular accelerations, angular velocity, and control signal.

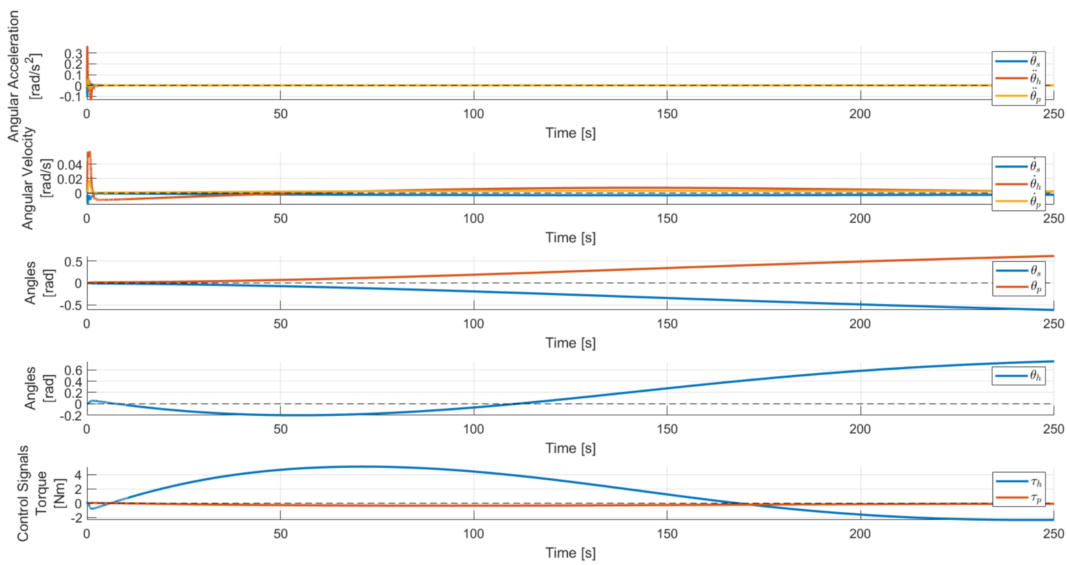


Figure 4.8: LQG Controller, simulation with the nonlinear *Decoupled model* without measurement noise.

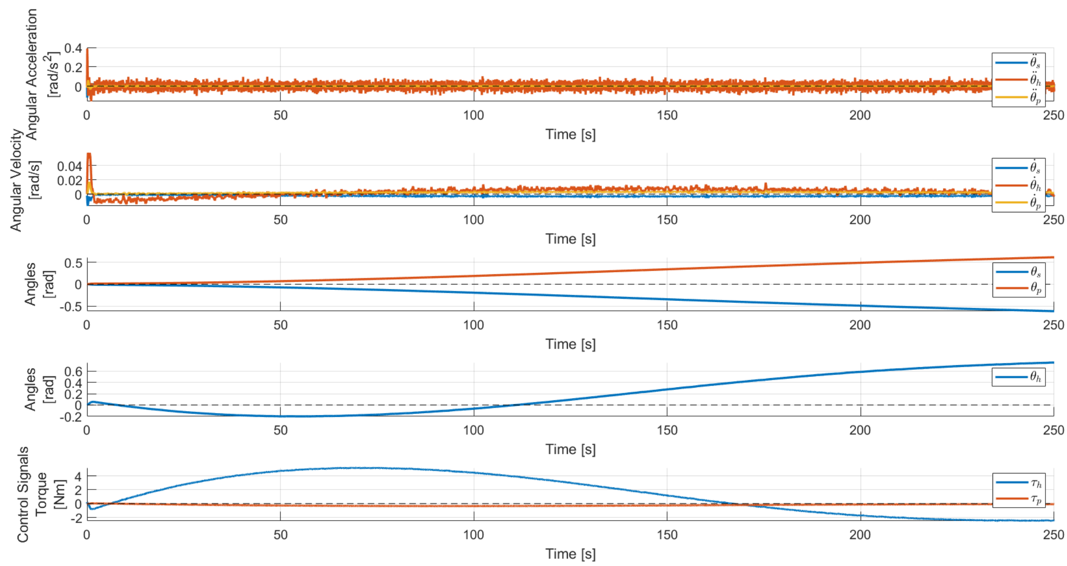


Figure 4.9: LQG Controller, simulation with the nonlinear *Decoupled model* with measurement noise.

From the simulations, Figures 4.8 and 4.9, one can observe how well the LQG stabilize the system with and without measurement noise. With the LQR used here is upholding nominal stability (NS) and nominal performance (NP), and that the performance seems from the figures to be improved. From the simulation without noise, Figure 4.8, one can observe that the inverted pendulum angle is around the desired angle of zero radian. Trying to converge with some degree of success, as can be seen by the control signal. The magnitudes of all the angular accelerations, angular velocities, and control signals are not as high compared to LQR and LQI. With that is has a steady increase of the angles for the pendulum and sphere. Even with noise it still behaves similar as without, with some small effect on the angular acceleration and angular velocity. With that the noise effect on the inverted pendulum seem to be the most problematic.

4.3.2 $\mathcal{H}_2$ - Controller

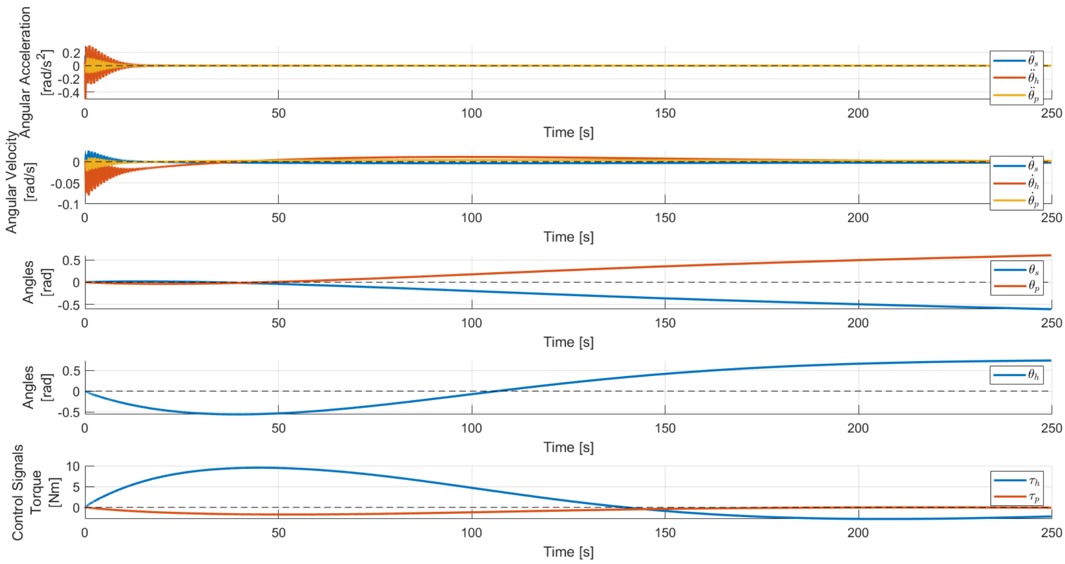


Figure 4.10: $\mathcal{H}_2$ Controller, simulation with the nonlinear *Decoupled model* without noise. Gave an γ value equal to 2.7717, with NS:True, NP:False, RS:False, RP:False.

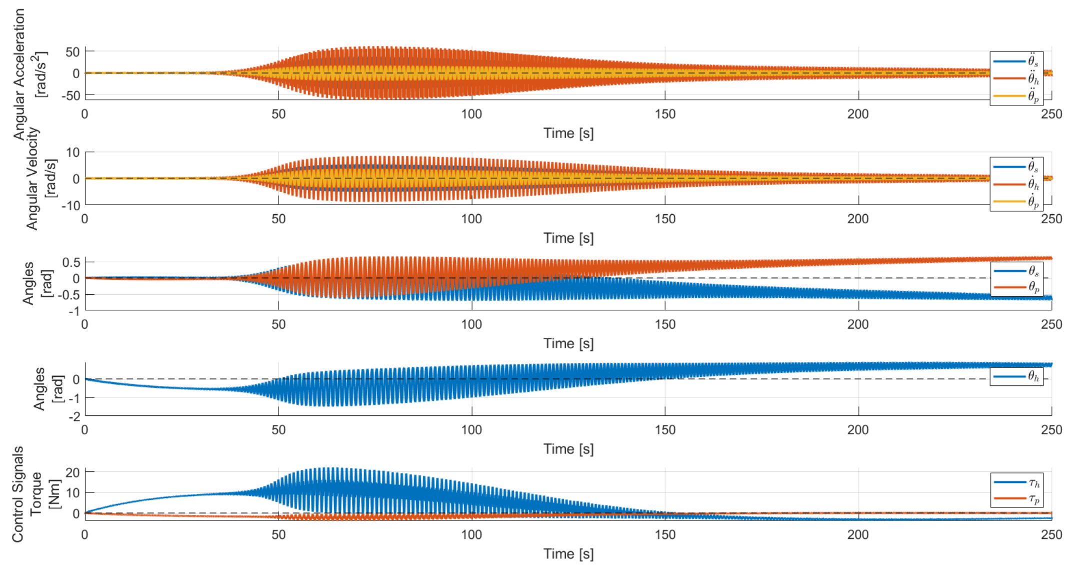


Figure 4.11: $\mathcal{H}_2$ Controller, simulation with the nonlinear *Decoupled model* with noise. Gave an γ value equal to 2.7717, with NS:True, NP:False, RS:False, RP:False.

From the simulations of the $\mathcal{H}_2$ controller, Figures 4.10 and 4.11, one can observe how well the controller stabilize the system with and without measurement noise. This controller got a γ value of 2.7717, which may be rather high. This is also shown to only uphold the nominal stability (NS). While the other, nominal performance, robust stability, and robust performance, is does not being uphold by this $\mathcal{H}_2$ controller. Which results in that the even though the angles of the system is behaving in a way one would want, in the noise free case, as can be observed in Figure 4.10. The effect of adding some measurement noise gives the system some sporadic changes for all the values, as can be observed in Figure 4.11. With that the noise effect on the inverted pendulum seem to be the most problematic. Resulting in that it is not a guaranteed way to ensure performance and robustness of the system. Even if for the noise free case the angles, angular velocity, angular accelerations, and control signals, have appreciated values of magnitude and behavior.

4.3.3 $\mathcal{H}_\infty$ - Controller

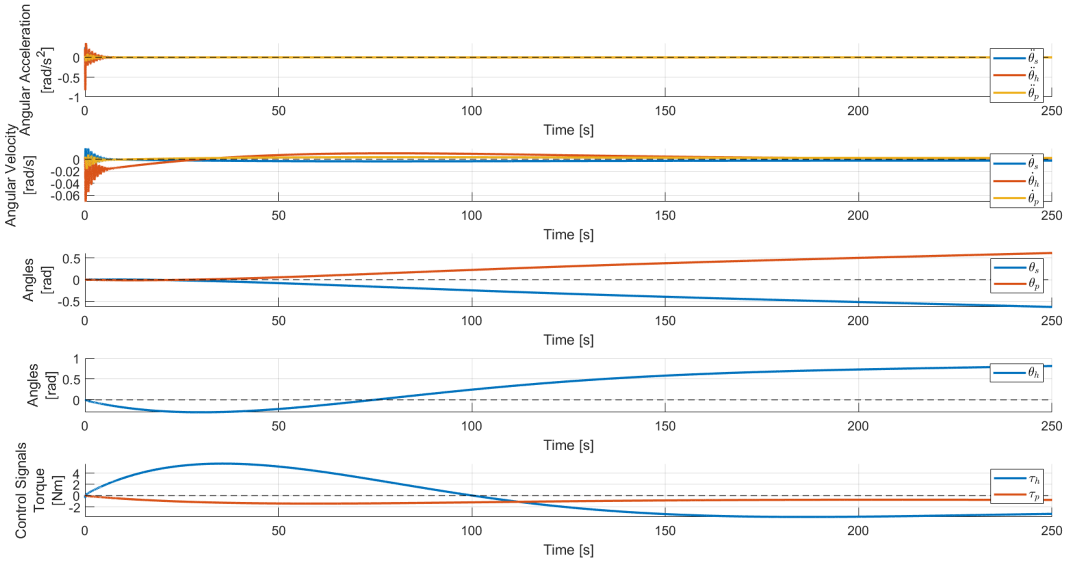


Figure 4.12: $\mathcal{H}_\infty$ Controller, simulation with the nonlinear *Decoupled model* without noise. Gave an γ value equal to 0.9772, with NS:True, NP:True, RS:True, RP:True.

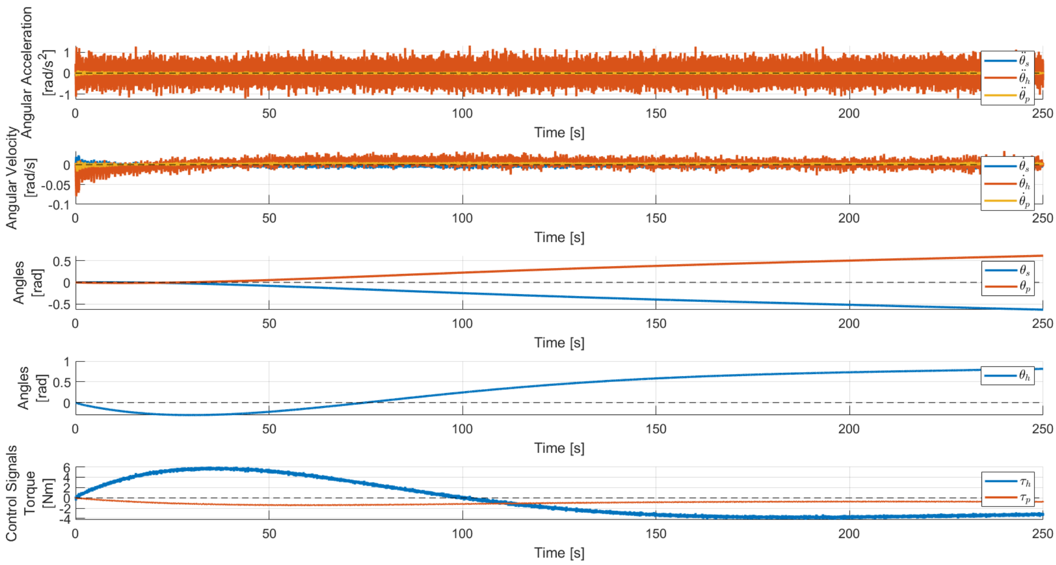


Figure 4.13: $\mathcal{H}_\infty$ Controller, simulation with the nonlinear *Decoupled model* with noise. Gave an γ value equal to 0.9772, with NS:True, NP:True, RS:True, RP:True.

From the simulations of the $\mathcal{H}_\infty$ controller, Figures 4.12 and 4.13, one can observe how well the controller stabilize the system with and without measurement noise. This controller got a γ value of 0.9772, that may be regarded as being close to be high. But is an appreciated value to have. This is also shown to uphold all the conditions of nominal stability (NS), nominal performance (NP), robust stability (RS), and robust performance (RP). That results in that for the noise free case, Figure 4.12, the angles, angular velocity, angular accelerations, and control signals, have appreciated values of magnitude and behaviour. With that the measurement noise may add some difficulties so the angular acceleration and angular velocity. That however is not effecting the results when adding measurement noise, as can be observed in Figure 4.13. Both the cases with and without measurement noise, does have similar form in its signal curves. With that the noise effect on the inverted pendulum seem to be the most problematic.

4.3.4 μ-synthesis ($\boldsymbol{DK}$-iteration)

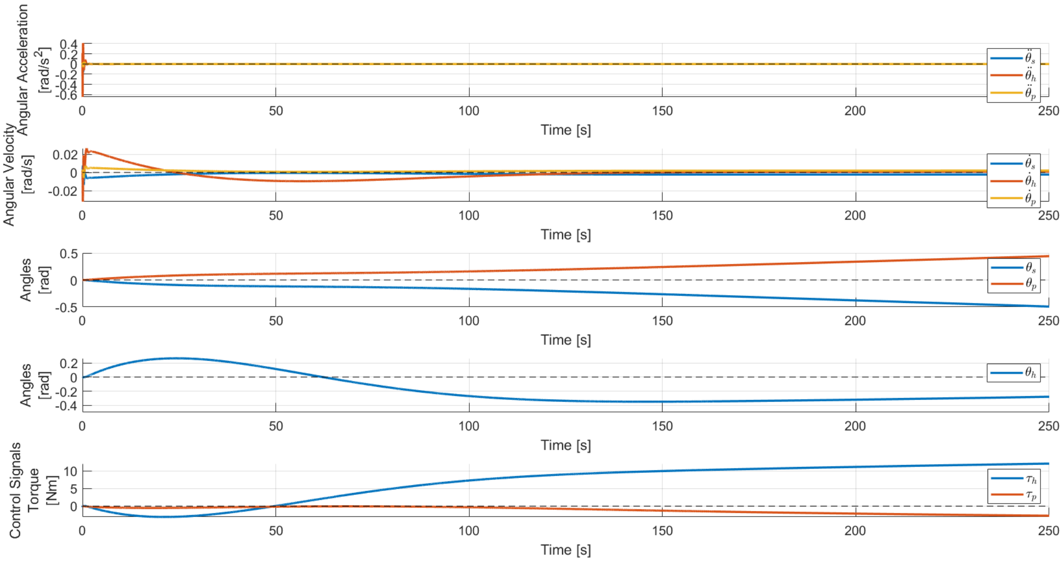


Figure 4.14: μ-synthesis Controller, simulation with the nonlinear *Decoupled model* without noise. Gave an μ value equal to 1.0837, with NS:True, NP:True, RS:True, RP:False.

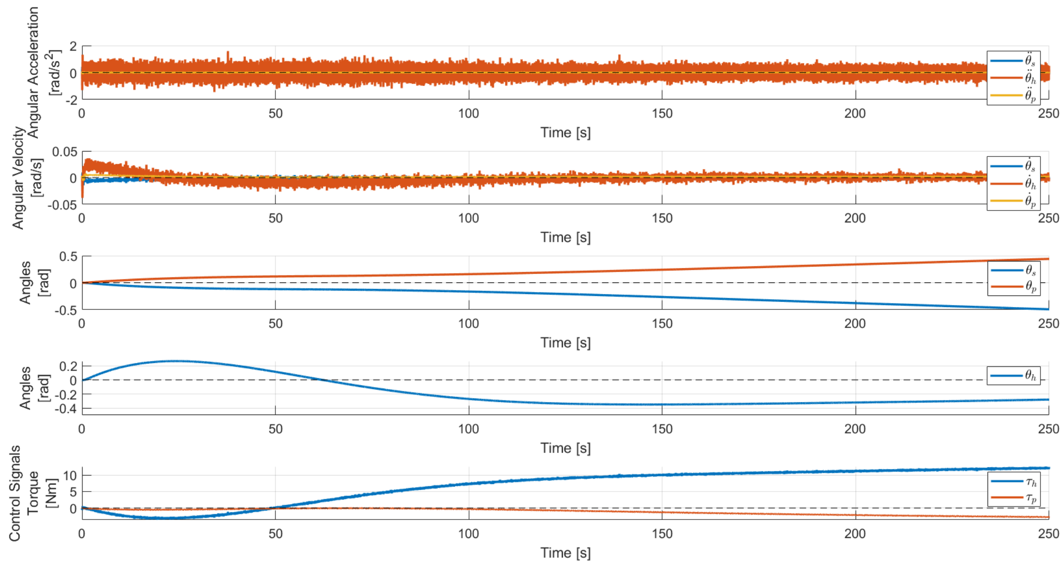


Figure 4.15: μ-synthesis Controller, simulation with the nonlinear *Decoupled model* with noise. Gave an μ value equal to 1.0837, with NS:True, NP:True, RS:True, RP:False.

From the simulations of the μ-synthesis Controller, Figures 4.14 and 4.15, one can observe how well the controller stabilize the system with and without measurement noise. This controller got a μ value of 1.0837, that is a little too high value for some criteria. Which is observed from that the robust performance is not being upheld. While nominal stability (NS), nominal performance (NP), and robust stability (RS) is upheld As mentioned in *μ-synthesis (DK-iteration)*, chapter 3.4.1.3. Due to stability problems of the controller containing unstable parts, only the stable part of the controller is being used for the simulation. Giving that some parts of the controller is missing. By the simulation without noise, Figure 4.14, one can observe that the signals is behaving as intended. With that it tries to have the angle of the inverted pendulum zero, with a smaller angle error compared to the other algorithms. While all the signals seem to be small in magnitude, with a steady increase of the angles for the pendulum and sphere. Giving that the robot is moving while keeping the load on top, with some not to high input control signals. Comparing if one adds noise, it seems that not a major change in behavior happens. Exempt some quite influential noise effect on the inverted pendulum, that seem to be the most problematic.

4.4 Nonlinear Control Algorithms

Here the simulation results from the nonlinear control algorithms will be presented, the angular accelerations, angular velocities, angles, and the control input signal's will be presented. By that the linear model analysis results from the chapter 4.2.1. Only the *decoupled model* is used, for that the *coupled model* is not guaranteed to be stabilizable. Two of each simulation, one for noise free situation and the other is with measurement noise.

4.4.1 Feedback Linearization Control

Following the implementation of the feedback linearization control with the use of linear input selection, given in *Feedback Linearization Controller*, chapter 3.4.2.2 Gave that this controller with its settings was not able to control the nonlinear decoupled system. While the reason may be argued for, this may be the result of a to narrow selection of parameters that was needed to be found in order to get a working controller for this system. That is more explained in the discussion of the feedback linearization controller, given in chapter 5.4.2.

5

Discussion

This chapter will discuss the different results of the control algorithms. With the simulation results in the *Results*, chapter 4.

5.1 State-space Models

For all the control algorithms tested and evaluated, a common question may be how well the state-space models relates to a real system of its kind. Providing with a fair model for the comparison of all these control algorithms. For this case that is a relevant question, due to that only simulation is used and not some practical tests to verify the simulation results. However, this will be hard to show by that no practical tests of the system are done. With that the careful calculations of the nonlinear system with *Lagrange's equation* will have to suffice. Together with that the behavior of the system is as thought. Such as that the angles of the pendulum and sphere is of opposite signs and has closely the same magnitude. Resulting in that the system is in motion and is increasing its distance traveled.

While the level of non-linearity in the models could be increased. That however would increase the computational time, therefore was kept at this level of non-linearity. This also applies to that only one parameter was chosen to be uncertain. One could have used so that all the parameters would have some uncertainty. That would then increase the computational time and resources needed to simulate. Focus more on to making a controller for the model, rather than to compare the different algorithms. Therefore, it was determined that as few uncertain parameters was to be used. With that one important parameter was to be uncertain, which in this case was determined to be the load coefficient. Since the unstable part of the models would contribute the most to the system. That in this case was the inverted pendulum, and because the load could be of different shapes.

The model made for turning has not been used and verified, which is because of several reasons. Mostly because of time restraints and that it was enough for the comparison to be done on just the stabilization part. Argued one could say that it would be a better comparison if one had used tracking performance comparison of the algorithms. Tracking was potentially to be used for comparison. However, due to limits which is why the model of turning is only given as an uncovering model with its energies.

5.2 Model Analysis

The analysis of the different models of *coupled model* and *decoupled model*. Showed that both the *coupled model* and *decoupled model* was observable, and therefore both are detectable. While the *decoupled model* was controllable, the *coupled model* was not. Then because of the eigenvalues it was determined that the *coupled model* was not guaranteed to be stabilizable. Therefore, was not used in the tests. The reason to why this model was not stabilizable it due to rank deficiency of the controllability matrix and the eigenvalues of the system. The eigenvalues was shown to be in a pair of an stable and a unstable ones, seen in table 4.1. It may also be observed that two of the columns of the linear *coupled model* is equal, seen in model VI. By that it may be that a fully coupled system of this type is not an appropriate way to construct such mechanic. With that the *decoupled model* was controllable and therefore stabilizable. A potential mix of a decoupled and a coupled system could work. That however is another topic to research.

5.3 Linear Control Algorithms

In this chapter discussion of the linear control algorithms is mad. Which the results and implementation of these controllers will be discussed.

5.3.1 PID - Controller

Even though PID controllers was not tested, it is still a good start point to have. That is because its relatively easy to formulate in different ways to control a system. With that the main principles of the PID controller is the different properties of *proportional*, *integral*, and *derivative*. Which can be to some extent read in *PID - Controller*, chapter 2.9.1.1. That can be related to other algorithms, such as LQR as using *proportional*, and LQI as using both *proportional* and *integral*. However, a PID is made in a SISO way, that may be a negative property of using PID control, and its similar controller. This would limit the controllers view of the many input states considered. That is because as one may observe from the linear quadratic case, the controller is dependent on knowing several states and not just one.

5.3.2 Linear Quadratic Controllers (LQR, LQI, LQG)

The linear quadratic controllers, which results one may observe from the many figures in *Linear Quadratic Controllers (LQR, LQI, LQG)*, chapter 4.3.1. LQ controllers are using a cost function that is quadratic, (hence the name), which as explained in *Linear Quadratic Controllers (LQR, LQI, LQG)*, chapter 2.9.1.2. The LQ controllers are relatively easy to formulate, with that there exist solvers for them. This gives that one only more or less needs to consider the design parameters of these controllers, or weight matrices. For the implementation of the LQ-controllers was relative simple with the use of *riccarti equation*, which was used in the MATLAB build in functions. The design parameters choices were mostly just tested out for different values, until a good enough performance compared to other tests was found.

Linear Quadratic Regulator From the implementation in *Linear Quadratic Controllers (LQR, LQI, LQG)*, chapter 3.4.1.1. When making the LQR *cheap control* was used to define the weights matrices. This is to make it cheap for the controllers to have large values of their control signals, namely the torques. Which was chosen in order to make the controller more aggressive. By the results of the LQR controller in *Linear Quadratic Controllers (LQR, LQI, LQG)*, chapter 4.3.1. It can be observed from the Figures 4.4, and 4.5, that it was able to perform in a good way. However, somewhat slow even though that was considered when choosing *cheap control* policy. Then as can be seen from the Figure 4.5 is that noise is effecting in some disturbing manners. Making the control signal and angular acceleration of the inverted pendulum sporadic with high magnitudes. Resulting in that even with low measurement noise, the impact of the noise in the LQR is not appreciated.

Linear Quadratic Integrator For the LQI controller, which implementation is given in *Linear Quadratic Controllers (LQR, LQI, LQG)*, chapter 3.4.1.1. A potential problem of using an integral action is that one may end with an integral windup problem. That may have disturbed some of the results of the LQI controller. As can be seen by that the LQI controller could not maintain the inverted pendulum standing as the LQR controller could, shown by the Figures 4.6, and 4.7. The integral windup problem was not compensated in these tests. The problem with the simulation as mentioned in *Simulink*, chapter 3.2.2. Is that the states are given by integrating the angular acceleration, giving rise to a potential problem with the integral action.

Linear Quadratic Gaussian For the LQG controller, which implementation is given in *Linear Quadratic Controllers (LQR, LQI, LQG)*, chapter 3.4.1.1. This controller is using an LQR controller with a Kalman filter to estimate the states. That is because it is not always possible to measure all states, therefore the states of the velocity of the sphere, angle of the inverted pendulum, and the angle of the pendulum was thought of being reasonable to be measured. Then they were used as the output ($y(t)$) of the system. Which by the results of the LQG, Figures 4.8, and 4.9, seems to be sufficient to control the system. This controller is using the existing LQR controller with adding a Kalman filter, to easier compare the two. By that one could separately calculate the two, by the *principle of separation*, definition 2.23. With that comparing the LQR and LQG results, gives that the added Kalman filter gave improved performance for all the states and control signals. As can be observed from the noise free case Figures 4.4, and 4.8. With added noise the LQR had some difficulties maintaining a good control signal and angular acceleration. While for the LQG the measurement noise is not affecting the states and control signals that much. Mostly only noticeable by the angular acceleration and angular velocity of the inverted pendulum. With low magnitude, as can be observed from the Figure 4.9.

5.3.3 $\mathcal{H}_2$ & $\mathcal{H}_\infty$ - Controllers

$\mathcal{H}_2$ & $\mathcal{H}_\infty$ controllers takes some more work than the LQ-controllers. Due to that one need to define a system as shown in the Figure 3.3. The weights of the system may take some time to determine appropriate values for. They may however provide with better control for general purposes and for perturbations of the systems. That is because it does take into consideration uncertainty of the system, in this case real parametric uncertainty. This gives that one can consider the robustness of the system with controller, both robust stability (RS), and robust performance (RP). As explained in the approach of using these controllers, chapter 3.4.1.2. The weights for the system was the same for both controllers and was chosen by the way of using the singular values of the uncertain system. The weights were determined so the nominal, and robust conditions where uphold as much as possible. However, for the $\mathcal{H}_2$ controller the robustness parts was hard to achieve, in comparison to the $\mathcal{H}_\infty$ controller. Which may be explained in how the robustness properties of robust stability, and robust performance is determined. The $\mathcal{H}_\infty$ norm of the system is lower than one, when these are upheld. The results of the simulations of the $\mathcal{H}_2$ controller in chapter 4.3.2, and for the $\mathcal{H}_\infty$ controller in chapter 4.3.3.

$\mathcal{H}_2$ Controller This controller only upheld nominal stability, with an γ value of 2.7717. With the results shown in Figure 4.10, and 4.11. Even though it has some appreciated curves with the states and control signals. If one adds low measurement noise all the states and control signals becomes sporadic with a not so appreciated magnitude. Giving that the $\mathcal{H}_2$ controller may give some appreciated performance if no noise is present. However, in reality there will be some noise, and that gives that the $\mathcal{H}_2$ controller with these settings may not be that good compared to some other algorithms. However, it is to notice that the design parameters to make this can be altered to improve it. Thought that may take more time and resources.

$\mathcal{H}_\infty$ Controller This controller upholds all the conditions namely nominal stability, nominal performance, robust stability, and robust performance. With an γ value of 0.9772. This is shown in the results of the $\mathcal{H}_\infty$ controller, Figures 4.12, and 4.13. That compared to the $\mathcal{H}_2$ controller, which has similar curves for the states and control signal. This has smoother values to boost, and the effect of the measurement noise is not that overwhelming. Giving that this controller is handling measurement noise well and is upholding robustness for the system. Meaning that this controller will handle perturbations of the system as well, as holding the performance.

5.3.4 μ-synthesis (DK-iteration)

μ-synthesis, or more specifically the DK-iteration, as mentioned in the μ-analysis, chapter 2.9.1.4. Was that there does not exist a direct way to use μ to synthesise a controller, DK-iteration was one that could be used instead. Using this DK-iteration, as mentioned in the implementation in chapter 3.4.1.3, gave the results shown in the figures in chapter 4.3.4. Using the system defined for the $\mathcal{H}_2$ & $\mathcal{H}_\infty$ controllers, with the same weight values. Therefore, it takes approximate the same amount of design time to create the system. However, with the exception that for the μ-synthesis there does not exist a uncertain block Δ, because it is inside the plant model. This gives that one does not have to determine the WiM weight, and so one less weight to determine compared to the $\mathcal{H}_2$ & $\mathcal{H}_\infty$ controllers. The DK-iteration is an iterative process and therefore can take its time, depending on one's settings for the calculation of the controller. Due to stability problems of the controller containing unstable parts, only the stable part of the controller is being used for the simulation. Giving that some parts of the controller dynamics is missing.

Simulation Results This μ-synthesis controller upholds the conditions of nominal stability, nominal performance, and robust stability. With an μ value of 1.0837, that explains why the robust performance was not uphold, by that the μ is above 1. The results of using this controller on the system, can be observed by the Figures 4.14, and 4.15. That shows improved performance of the states especially the angle of the inverted pendulum, compared to the other algorithms. While even with added measurement noise it still has the improved performance. However, with some disturbance to the angular acceleration and angular velocity of the inverted pendulum. While still having the same performance with the angle of the inverted pendulum. Even though this controller used is missing some of its dynamics, it still performs as good or even better than the other algorithms.

5.4 Nonlinear Control Algorithms

In this chapter discussion of the nonlinear control algorithms is made, namely the linear output selection for feedback linearization. With a discussion around the feedback linearization.

5.4.1 Feedback Linearlization

Feedback linearization which was calculated in *Feedback Linearization*, chapter 3.4.2.1. Was shown to result in difficulty to determine if the *zero dynamics* was stable or not. Therefore it was determined to not guarantee that the *zero dynamics* was stable. Were another approach was used, however, this does not mean that this approach is not valid. That is because if one can find an easier way to determine if the *zero dynamics* is stable, one may then use that system together with some controller.

5.4.2 Feedback Linearization Controller

Using the another approach of "linear output selection for feedback linearization" [37], gave that one could get stable *zero dynamics*. This gives as mentioned in the approach, *Feedback Linearization Controller*, chapter 3.4.2.2, that this approach is using a LQR as the output of the system. With an LQR as the controller of the system, that was to have cheap control policy. Gave the results shown in the *Feedback Linearization Control*, chapter 4.4.1. However, that was shown to not being able to control the nonlinear system. The reason for this is because the local stability region obtained by the optimal output selection is small. While some other unstable points may have a large region of attraction. Resulting in that to find the potentially narrow local (or global) stable points region of attraction may be needed to make the controller work for this system. Which is another topic to research more into. This may also be observed by that for all the other controllers, both with and without noise the inverted pendulum is hard to stabilize. Giving that for a system that tries to consider the nonlinear system and not the linearized system around a stationary point, may be provided with greater difficulties. When one must consider the effects of non-linearity may potentially give even when one makes the nonlinear system look linear like. It could potentially give rise to other problems with the unstable and stable points, with regions of attraction and repulsion.

6
Conclusion

6.1 State-space Models

By that only simulations are done, to test if the models are behaving like the real system is hard to investigate. Therefore, the only way to observe if the assumptions of the state-space models are appropriate. Is to use the simulation results and observe if the models are behaving as intended from the knowledge of the mechanical system. By that as discussed in *State-space Models*, chapter 5.1. One can conclude that the *decoupled model* is working as intended. By that it may be assumed that the *coupled model* is also working as intended, with that they are similar in structure.

6.2 Model Analysis

It may be summarized that for the different models, *coupled model*, and *decoupled model*, they have different analytic properties. The *decoupled model* is controllable, and observable. While the *coupled model* is only guaranteed to be observable. Therefore because the*coupled model* is not guaranteed to be stabilizable, it is not used for simulation.

6.3 Control Algorithms

About the control algorithms used in the thesis, gives that a less complex controller may be enough if one's demand of the control system is not high. The less complex controllers are easier to implement and tune, gives that they are more cost effective even if the performance requirement isn't high, or in focus. While for the robust control algorithms, they can be formulated such that the performance is kept. This comes with more complex implementation and tuning of the design parameters, with a potential higher cost of implementation as a result. That, however, should not be focused, since if one wish to guarantee a certain performance in a real system. It is expected to contain noise and perturbations of the system, with model uncertainties. That is because the models are simplified giving rise to some model uncertainty, and uncertainties when constructing the real system, is certain to occur. That form the discussion about the results of the algorithms, chapters 5.3, and 5.4. It may be observed that even less complex controllers will give good performance (LQR), and with some added complexity (LQG) it may even handle noise to the system. But with more complex controllers resulting in improved performance and noise handling, with robustness to perturbations and uncertainties of the system.

6.4 Future Work

Some future work that could be investigated.

Modeling

- The interaction of the magnetic field onto the sphere, where adding this would make a more extended models, than the general ones. It may have a possibility of impacting the stability of the system.
- A new model that is a mix of the *coupled model*, and *decoupled model*, would be a potential investigation, if one can make such a model.

Analysis

- Find out if the *zero dynamics* of the feedback linearization is guaranteed stable or not.
- Investigate the nonlinear models stability aspects, with for example *Lyapunov's Stability theorem*.

Control

- It would be interesting to see the impact of the slope into the controllers. This would give realistic simulations.
- Tracking performance as a measurement for comparison between the algorithms would be the next step. Tracking would potentially make the control more difficult, then just moving forward and keeping the load on top of the shell.
- Fixing the problem for the linear quadratic integrator, the integral windup problem. So that it may give its expected behavior of removing the steady state error.
- Solving the problem of the optimal output selection feedback linearization. Finding the small local stability region obtained by the optimal output selection. To observe how well the nonlinear control algorithms is compared to the linear ones.
- A wider range of controllers would be of interest for the comparison. Such as to add the PID to the comparison, and other controllers as LQGI, MPC, and some more variation of nonlinear ones.
- Real application tests of the algorithms on a spherical robot with load platform, would show how they perform than just simulation. Showing how accurate the simulations is and in so the models.

Bibliography

[1] J.L.Tate, *Toy*, 1893.

[2] R. E. Cecil, *Mechanical toy*, 1909.

[3] J. E. Reilley, *Spherical vehicle*, 1941.

[4] H. William, *Marine vehicle*, 1889.

[5] J. Suomela and T. Ylikorpi, "Ball-shaped robots: An historical overview and recent developments at tkk", vol. 25, Jan. 2005, pp. 343–354. DOI: 10.1007/978-3-540-33453-8_29.

[6] B. P. DeJong, E. Karadogan, K. Yelamarthi, and J. Hasbany, "Design and analysis of a four-pendulum omnidirectional spherical robot", *Journal of Intelligent & Robotic Systems*, vol. 86, no. 1, pp. 3–15, 2017, ISSN: 1573-0409. DOI: 10.1007/s10846-016-0414-4.

[7] S. Mahboubi, M. Fakhrabadi, and A. Ghanbari, "Design and implementation of a novel hybrid quadruped spherical mobile robot", *Robotics and Autonomous Systems*, vol. 61, no. 2, pp. 184 –194, 2013, ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2012.09.026.

[8] S.-S. Ahn and Y.-J. Lee, "Novel spherical robot with hybrid pendulum driving mechanism.", *Advances in Mechanical Engineering*, vol. 2014, ISSN: 16878140.

[9] M. R. Azizi and J. Keighobadi, "Point stabilization of nonholonomic spherical mobile robot using nonlinear model predictive control", *Robotics and Autonomous Systems*, vol. 98, pp. 347 –359, 2017, ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2017.09.015.

[10] A. A. Kilin, E. N. Pivovarova, and T. B. Ivanova, "Spherical robot of combined type: Dynamics and control", *Regular and Chaotic Dynamics*, vol. 20, no. 6, pp. 716–728, 2015, ISSN: 1468-4845. DOI: 10.1134/S1560354715060076.

[11] Y. Li, M. Yang, H. Sun, Z. Liu, and Y. Zhang, "A novel amphibious spherical robot equipped with flywheel, pendulum, and propeller", *Journal of Intelligent & Robotic Systems*, vol. 89, no. 3, pp. 485–501, 2018, ISSN: 1573-0409. DOI: 10.1007/s10846-017-0558-x.

[12] M. ZHENG, Q. ZHAN, J. LIU, and Y. CAI, "Control of a spherical robot: Path following based on nonholonomic kinematics and dynamics", *Chinese Journal of Aeronautics*, vol. 24, no. 3, pp. 337 –345, 2011, ISSN: 1000-9361. DOI: https://doi.org/10.1016/S1000-9361(11)60040-X.

[13] E. Kayacan, Z. Y. Bayraktaroglu, and W. Saeys, "Modeling and control of a spherical rolling robot: A decoupled dynamics approach", *Robotica*, vol. 30, no. 4, 671–680, 2012. DOI: 10.1017/S0263574711000956.

[14] T. J. Ylikorpi, A. J. Halme, and P. J. Forsman, "Dynamic modeling and obstacle-crossing capability of flexible pendulum-driven ball-shaped robots",

Robotics and Autonomous Systems, vol. 87, pp. 269 –280, 2017, ISSN: 0921-8890. DOI: `https://doi.org/10.1016/j.robot.2016.10.019`.

[15] F. R. Hogan and J. R. Forbes, "Modeling of spherical robots rolling on generic surfaces", *Multibody System Dynamics*, vol. 35, no. 1, pp. 91–109, 2015, ISSN: 1573-272X. DOI: `10.1007/s11044-014-9438-3`.

[16] M. Roozegar and M. J. Mahjoob, "Modelling and control of a non-holonomic pendulum-driven spherical robot moving on an inclined plane: Simulation and experimental results", *IET Control Theory Applications*, vol. 11, no. 4, pp. 541–549, 2017, ISSN: 1751-8644. DOI: `10.1049/iet-cta.2016.0964`.

[17] M. Roozegar, M. Ayati, and M. J. Mahjoob, "Mathematical modelling and control of a nonholonomic spherical robot on a variable-slope inclined plane using terminal sliding mode control", *Nonlinear Dynamics*, vol. 90, no. 2, pp. 971–981, 2017, ISSN: 1573-269X. DOI: `10.1007/s11071-017-3705-9`.

[18] T. Ohsawa, "Geometric kinematic control of a spherical rolling robot.",

[19] Y. Bai, M. Svinin, and M. Yamamoto, "Dynamics-based motion planning for a pendulum-actuated spherical rolling robot", *Regular and Chaotic Dynamics*, vol. 23, no. 4, pp. 372–388, 2018, ISSN: 1468-4845. DOI: `10.1134/S1560354718040020`.

[20] M. Roozegar, M. Mahjoob, and M. Jahromi, "Optimal motion planning and control of a nonholonomic spherical robot using dynamic programming approach: Simulation and experimental results", *Mechatronics*, vol. 39, pp. 174 –184, 2016, ISSN: 0957-4158. DOI: `https://doi.org/10.1016/j.mechatronics.2016.05.002`.

[21] T. B. Ivanova, A. A. Kilin, and E. N. Pivovarova, "Controlled motion of a spherical robot with feedback. i", *Journal of Dynamical and Control Systems*, vol. 24, no. 3, pp. 497–510, 2018, ISSN: 1573-8698. DOI: `10.1007/s10883-017-9387-2`.

[22] ——, "Controlled motion of a spherical robot with feedback. ii", *Journal of Dynamical and Control Systems*, vol. 25, no. 1, pp. 1–16, 2019, ISSN: 1573-8698. DOI: `10.1007/s10883-017-9390-7`.

[23] N. Masaki, "Control system for a spherical robot", PhD thesis, 2008.

[24] R. Berg, N. Brundin, D. Hedin, T. Suphona, M. Thander, and I. Wikenstedt, *Bb-8*, 2017.

[25] A. Andersson, C. Andersson, G. Andersson, J. Andrén, J. Laurell, and F. Åvall, *Bb-8*, 2017.

[26] O. Almér, H. Andersson, G. Boquist, E. Forsström, P. Hansegård, and L. Johansson, *Bb-8*, 2017.

[27] V. Adler, D. Johansson, A. Olsson, S. Simmons, and N. Svensson, *Vidareutveckling och konstruktion av sfärisk robot*, 2018.

[28] F. Scheck, *Mechanics. [electronic resource] : From newton's laws to deterministic chaos.* Ser. Graduate Texts in Physics. Springer Berlin Heidelberg, ISBN: 9783662554906.

[29] A. Boström, *Rigid body dynamics*, 2018.

[30] C. C. de Wit, H. Olsson, K. J. Åström, M. Gäfvert, and P. Lischinsky, "Friction models and friction compensation", eng, European Journal of Control, vol. 4, no. 3, 1998, ISSN: 1435-5671.

[31] N. Ida, *Engineering electromagnetics. [electronic resource].* Springer International Publishing, ISBN: 9783319078069.
[32] H. K. Khalil, *Nonlinear systems.* Upper Saddle river : Prentice Hall, cop. 2002., ISBN: 0-13-067389-7.
[33] K. J. Åström and R. M. Murray, *Feedback systems. [electronic resource] : An introduction for scientists and engineers.* Princeton University Press, ISBN: 1400828732.
[34] S. Skogestad and I. Postlethwaite, *Multivariable feedback control : Analysis and design.* Hoboken, NJ : John Wiley, cop. 2005., ISBN: 978-0-470-01168-3.
[35] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory", *TRANS. ASME, SER. D, J. BASIC ENG*, p. 109, 1961.
[36] K. Zhou, J. C. Doyle, and K. Glover, *Robust and optimal control.* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996, ISBN: 0-13-456567-3.
[37] G. Szederkényi, K. Hangos, J. Bokor, and T. Vámos, "Linear output selection for feedback linearization", *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 109 –114, 2002, 15th IFAC World Congress, ISSN: 1474-6670. DOI: `https://doi.org/10.3182/20020721-6-ES-1901.00268`.