ISSN 0280-5316
ISRN LUTFD2/TFRT--5750--SE

Controlling the Secondary Mirror Cell of the Euro50 Project using an Inertial Measurement Unit

Olof Sandberg

Department of Automatic Control
Lund Institute of Technology
May 2005

Department of Automatic Control
Lund Institute of Technology
Box 118
SE-221 00 Lund Sweden

Document name
MASTER THESIS

Date of issue
May 2005

Document Number
ISRNLUTFD2/TFRT--5750--SE

Author(s)
Olof Sandberg

Supervisor
Torben Andersen at the Observatory in Lund
Anders Robertsson at Automatic Control in Lu nd
Gustaf Olsson at Industrial Electrical Eng. and Automation in Lund

Sponsoring organization

Title and subtitle
Controlling the Secondary Mirror Cell of the Euro50 Project using an Interial Measurement Unit.
(Reglering av sekundärspegelcellen för Euro50-teleskopet)

Abstract

During the ongoing development of new car models, testing and simulation are used for validation, which involves a large amount of measured data being sampled. To use the tests and simulations in the best way possible, the data collection needs to be effectively worked on.
Currently, the evaluation of sampled data is done primarily through studies of numerous parameters in graphs. To avoid problems choosing the appropriate parameters to study, a graphic representation of the system's functions can be used.
The purpose of the Master's project is to develop a graphical interface, which visualizes data sampled during the development of a climate system and thereby eases the evaluation. The visualization is enabled through a method that visualizes a selection of measured data over a surface in an easy-to-grasp way. A model is developed to calculate temperatures in all places in the compartment as well as in parts of the climate unit. These temperatures are represented with colors. The data is primarily sampled in the car model Volvo XC90.
The visualization tool EasySee is implemented in MATLAB 7.0 release 14 and Java 1.4.2 on a Windows platform. The interpolation is done by MATLAB with Java responsible for the graphical drawing in the user interface and also the user interaction.
This Master Thesis covers the development process of the visualization tool EasySee a Master's project presented by Volvo Technology Corporation, department 6280 Control and Simulation, in Gothenburg. The Master's Thesis consists of the following main chapters: prerequisites, theory of analyzing sampled data and graphical user interfaces, method of development, results and guidelines for future development.This Master project resulted in an application with excellent opportunities to ease the daily work analyzing sampled data.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
83

Recipient's notes

Security classification

The report may be ordered from the Department of Automatic Control or borrowed through:University Library, Box 3, SE-221 00 Lund, Sweden Fax +46 46 222 42 43

Acknowledgements

Several people have been involved in this master thesis. I would like to take the opportunity to thank some of them, especially my supervisors Torben Andersen and Anders Robertsson for the guidance given. Without the aid of my fellow thesis students, Fredrik Bjöörn, Olof Garpinger and Roger Svahn, it would have taken a lot of extra time to overcome all those small obstacles encountered on the way. A special thanks to Gustaf Olsson for valuable input during the whole process.

To Erik, my Beloved Brother, 1983-2004

Contents

1 Introduction

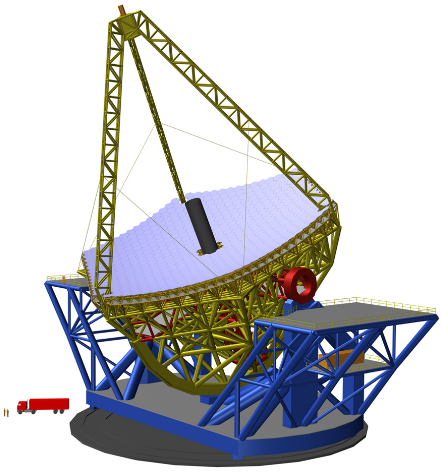

Figure 1.1: A bird's view of the Euro50 Extremely Large Telescope.

In the history of mankind, the sky has always been an inexhaustible source of mystery and dreams. As technology advances, so do the possibilities to reach further and further into the depths of the universe to satisfy our desire to understand and map everything around us. The next step in this evolution is the next generation of telescopes, the extremely large telescopes, or ELTs. The technologies that are involved in the development of these ELTs are, among others, highly advanced adaptive optics, automatic control at different levels and extreme engineering. In this section, a brief explanation of the Euro50 telescope[1] project is presented, as well as some of the features especially important to this particular work.

1.1 The Telescope

The Euro50 project is a European project, aiming to develop, build and make use of an ELT. Basically, the telescope consists of a hexagonal, $50m$ in diameter, primary mirror, built up with 618 smaller hexagonal mirror segments, each with a diameter of $2m$.

This mirror is aspherical, which is one of two possible design methods for mirrors this large. The other design method is a spherical mirror. Both methods have advantages and disdavantages. The main reason for a spherical

[1] More about Euro50 can be found in [1] and general information about telescopes can be found in [3]

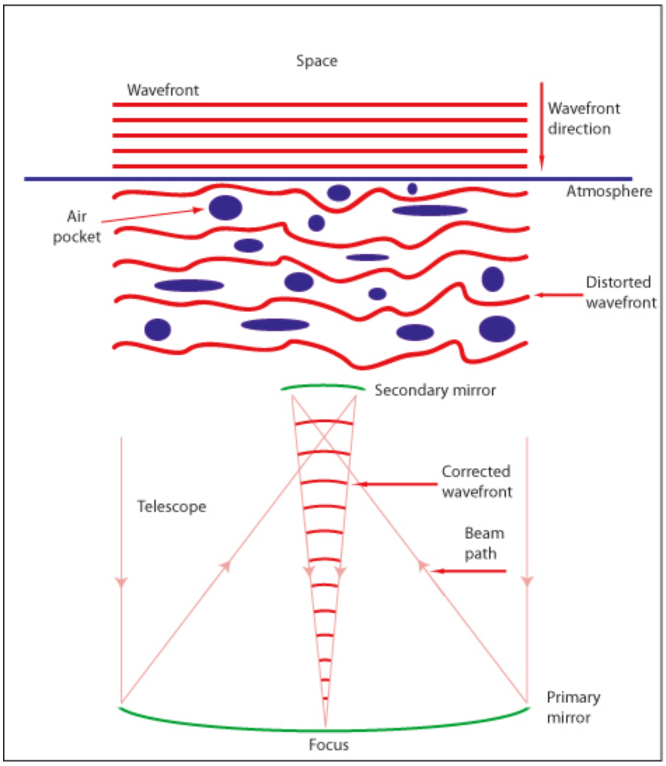

Figure 1.2: The light will be distorted before it reaches the telescope.

mirror is that it can be build up by smaller, *identical* mirror segments, thus lowering the manufacturing cost. On the other hand, a spherical mirror suffers from strong spherical aberration that demand for corrections by 2 or 3 additional mirrors. For each additional mirror, the transmitted light intensity falls by $\approx 10\%$, which directly contradicts the meaning of a telescope of this magnitude. An aspherical mirror on the other hand, must be made out of *unique* mirror segments, which makes it more expensive, and if any of the pieces must be replaced, that exact one must be manufactured again.

From the primary mirror, a tripod structure, made out of Carbon Fiber Reinforced Polymer (CFRP), rises some 60 m above ground level. In the top of this tripod sits the secondary mirror, sometimes referred to as deformable mirror 1 or DM1. The secondary mirror is mounted under a large block, the secondary mirror cell.

Why does one have to build an even larger telescope than the ones already existing? The answer is the evergrowing thirst for knowledge. With an ELT, solar systems can be studied. The ELT would have such an ability to detect light, that a single planet's light would be possible to measure.

In the beginning, the Euro50 will only be capable of detecting light in the infrared region, the K-band, with nearby diffraction limited quality. Later on it is planned to implement features to be able to do so in visible light, the V-band.

1.2 Adaptive Optics

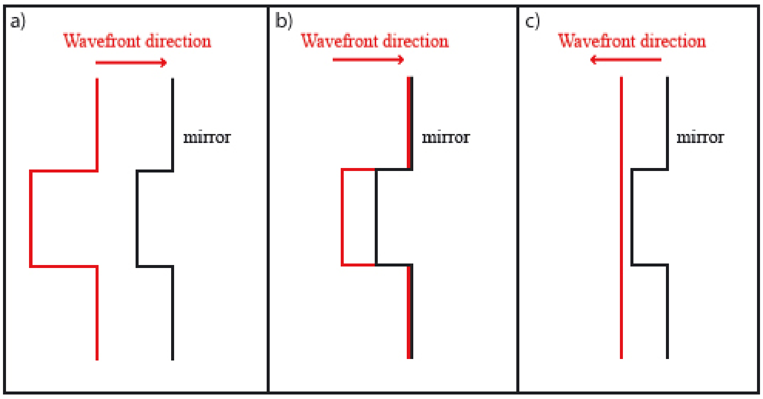

Figure 1.3: By deforming the mirror surface, parallel wavefronts can be reproduced. (In reality, the level differences on the mirror will not be as sharp as on the picture)

To be able to detect such small amounts of light, the need of light with good quality is imperative. This is why the development of adaptive optics began. Adaptive optics has optical components, mirrors in this case, that are capable of deforming their surface, to be able to change the reflected light, see Figure 1.3. The technology is currently under research, and no deformable mirror with a radius greater than about 1 m has been successfully implemented. The secondary mirror of the Euro50, DM1, will be used to reproduce almost perfect wavefronts which have been degraded by the propagation of the light through the atmosphere. Also, it will be used to correct for aberrations from the telescope structure itself and effects from the primary mirror. To know how to deform the mirror properly, a reference signal with the suitable surface must be obtained. Since the light is corrupted by layers in the atmoshphere, see Figure 1.2, the reference signal must be made out of light from a well known source. The light from this source must have traveled roughly the same way as the light waves that are to be corrected. These sources of light are called guide stars, as stars are commonly used as the source of the reference signal. One or several stars can be used in the

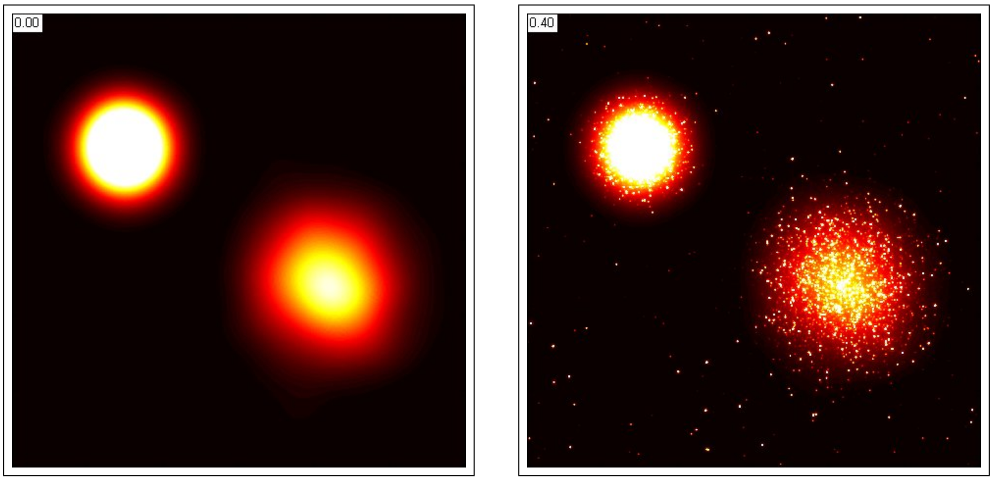

Figure 1.4: Hypothetical field of the sky with two globular clusters.
left) View from a 50 m telescope without adaptive optics.
right) Estimated view from Euro50 with adaptive optics.
Simulation: Ralph Snel

process. The more, the better the reference signal.

1.3 Wavefront Sensor

To be able to realize adaptive optics, a device that can detect the flaws in the wavefronts is needed. For this purpose, the wavefront sensor has been developed. It can detect deficiencies in the wavefronts and thereby the optical aberations. The sensor is built up by an array of smaller lenses. Each lens focuses a small part of the incoming wavefront. By having a sensor at the focal plane of this array of lenses it is possible to see the slope of each small part of the wavefront, thus beeing able to see any perturbation in the entire wavefront.

The wavefront sensor also plays an important role in this master thesis. In fact, it is the only sensor that measures the absolute errors. From its measurements, the wavefront sensor can distinguish the different rotational errors from the lateral errors, thus allowing a separation of five of the six degrees of freedom. The only degree of freedom that the wavefront sensor cannot measure is the rotation around the z-axis (ν-rotation). This is because of the cell's rotational symmetry.

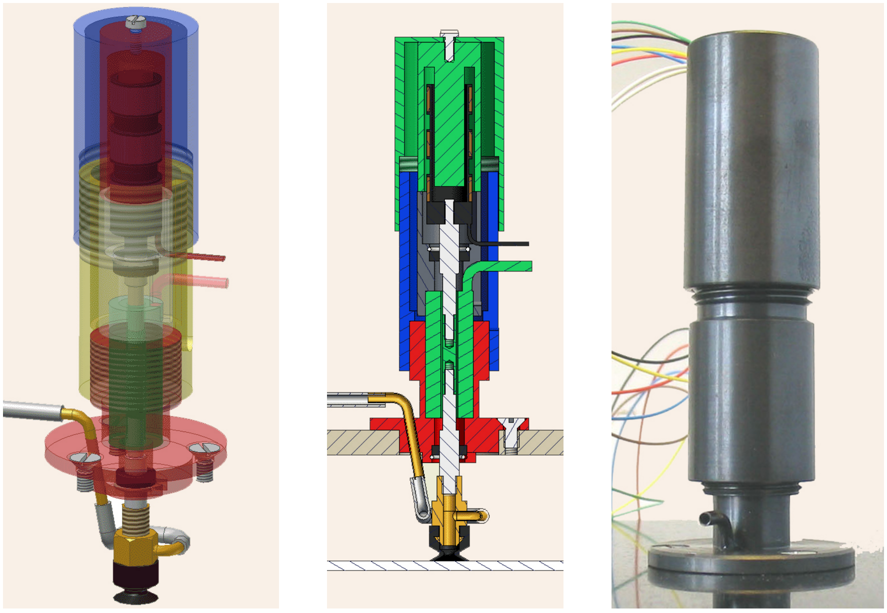

Figure 1.5: A schematic picture of one of the actuators used to shape the form of the deformable mirror.
Left: X-ray view, middle: section, right: photo of prototype.

1.4 Actuators

Even with the aid of all tools mentioned, adaptive optics cannot be realized without an actuator deforming the mirror. Different actuators have been developed in the adaptive optics community. Piezoelectric, magnetostrictive and electrostictive actuators are some that have been tested. Every method has its advantages and drawbacks. For the Euro50 project, another method has been developed. The actuator is an electromagnetic linear motor, a so-called voice-coil, which uses a vacuum cup as the connection to the mirror. Using a vacuum cup as the connection makes is very easy to change an individual actuator if necessary. However, it also makes the connection very soft, unlike an ideal stiff connection that would make the system easier to control. To be able to use this kind of actuator despite its softness, an internal control system must be made.

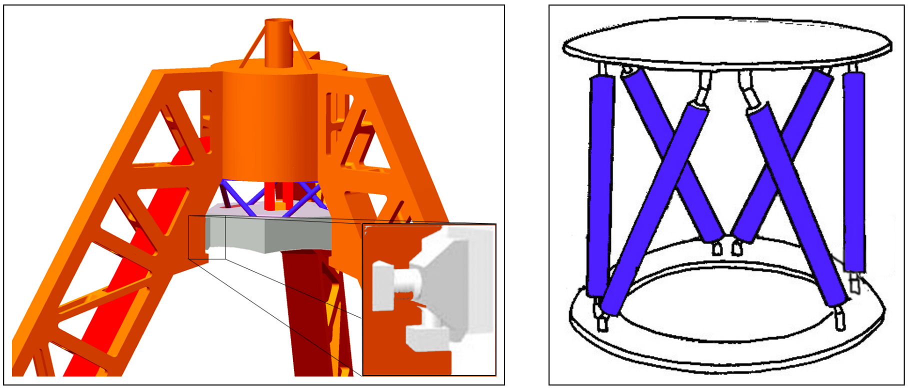

Figure 1.6: **left)** The air bellows (zoomed) and the fixation of the force actuators. **right)** A schematic picture of a hexapod geometry for the actuators.

1.5 The Secondary Mirror Cell

The secondary mirror cell, adjustable in six degrees of freedom, will be located on the top of the tripod structure, resting on 4 air bellows[2], of which 2 are illustrated in Figure 1.6. It is necessary to reduce the influence of the wind loads, otherwise the wind will translate the entire mirror cell some millimeters, which is about two orders of magnitude more than the $\approx 20\ \mu m$ allowed. The reason that there is only one lateral set of air bellows is that the telescope will only tilt in one direction, and therefore only one set of bellows is sufficient. Since the mirror cell has a 6DOF suspension, six actuators will be necessary for full control capabilities. These actuators will be of the force type, which means that there will be pure forces imposed on the cell structure. The actuators will be arranged in a so called hexapod geometry, illustrated in Figure 1.6. By specifications, each of the six actuators can deliver forces in the range of $\pm 1000N$.

[2] 1 lateral and 3 axial

2 Problem formulation

The fundamental problem formulation of this master thesis project is to investigate if it is possible to be able to control the secondary mirror cell within the deviation limit of 20 μm, and if so, how. The basic problem is stated as follows: The mirror cell, located some 60 m vertically above focus, should stay in place with a margin of $\pm 20\ \mu m$. The deviation of the angles of the mirror cell should be no more than $\theta = \arctan \frac{20 \cdot 10^{-6}}{60}^{\circ} \approx 0.2\ arcsec = 0.969\ \mu rad$. The mirror cell has a mass of around $700 kg$, and should be controlled with a bandwidth of $10 Hz$. The above mentioned force actuators will, of course, have a saturation limit, estimated to $1000 N$.

Since this work deals with stochastic problems, the deviation specifications are also stochastic variables. The maximum displacement 20 μm is actually standard deviation $\sigma_{deviation} = 20\ \mu m$. To solve the problem with these specifications, a solution based on the aid of an Inertial Measurement Unit, (IMU), is proposed. One aim of this work is to estimate the specifications needed for the parts of the IMU, and to verify that an implementation by this kind of control theory can be done. One question of the use of only the IMU can be raised. The question is whether the use of an IMU is justified, or if the use of the wavefront sensor would be better.

The warrant of the usage of the IMU is that it will take care of the fast fluctuations faster and better than the wavefront sensor. This because of the IMU's inner structure, the separation of the different degrees of freedom. It is belived that the wavefront sensor cannot separate the different errors as fast and accurately as the IMU. Instead, the wavefront sensor gets to analyse the errors that originate from the IMU's noise properties, which in many ways are slower and less problematic.

2.1 The Inertial Measurement Unit

The main part of this thesis is based on control using an Inertial Measurement Unit. An IMU mainly consists of three gyros and three accelerometers, each designed to take measurements from one of the six degrees of freedom.

2.1.1 The Gyro

Apart from the conventional rotating gyros, there are mainly three different kinds of gyros; piezoelectric, fiber-optic gyros (FOG), and ring laser gyros (LRG). Piezoelectric gyros measure the angular rate by "tuning forks", made out of silicon or quartz, that measure its deformation due to the Coriolis force when rotated. Because they measure the rate by vibrating tuning forks, they

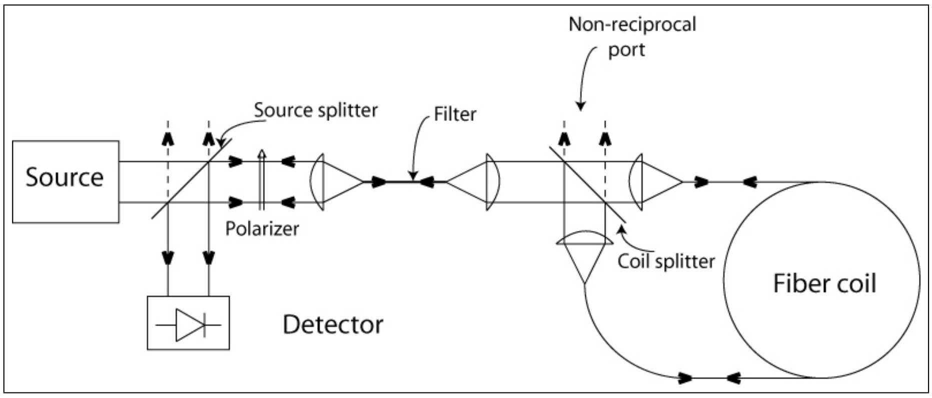

Figure 2.1: The principle for fiber optical gyros.

are easily affected by vibrations from the surroundings, making them difficult to use for the Euro50 project. They are also less precise.

The fiber optic gyros and laser ring gyros use the Sagnac effect to measure the angular rate. Both of these kinds of gyros send identical light waves into opposite ends of a fiber coil (FOG) / laser cavity (LRG). When not rotating, the light has the exact same optical distance to travel in the coil/cavity. Therefore, there will be no phase difference between them when they exit thier optical path into a sensing element. However, when rotating, a phase difference, proportional to the angular velocity will emerge. And from that phase difference, the angular rate is calculated. Fiber optic gyros have been developed for a longer time than laser ring gyros and are still more reliable. As time passes, the LRG will almost certainly get closer to the FOG in means of stability and robustness. The choice of gyro type is therefore a question for the future.

There are a some limitations on gyros that limit their use for all applications. All optical gyros suffer from angular random walk, ARW, which is a stochastic disturbance in the measurements. This is a variable used to measure the noise influence in gyros. This noise influence is made even worse by the fact that the measurement signal has to be integrated before it can be used for position reference, since the raw output is angular velocity and not position.

Another important drawback with all FOG is that there is a bias on the output due to different optical phenomena, [7]. The bias would not have been a problem if it were constant, but there is an uncertainty in the bias too. The limitation is not as severe as for the random walk, but the bias changes slowly over time. Other limitations in the measurements come from axis misalignment errors, temperature dependency and linearity errors. For this application, this does not have the crucial effect it could have. Other telescope projects have tried to use high precision gyros with very little angular random walk together with bias estimation among other techniques. This resulted in telescope observation times around 15 minutes with the maximum resolution of 0.2 *arcsec*. The aim with this project is to be able to keep the minimum resulution 0.2 *arcsec* for as long as the observation requires. To be able to do this with the influences degrading the gyro measurements taken into account, an external position reference must be used. As stated in Section 1.3, the wavefront sensor is used for this purpose.

2.1.2 The Accelerometer

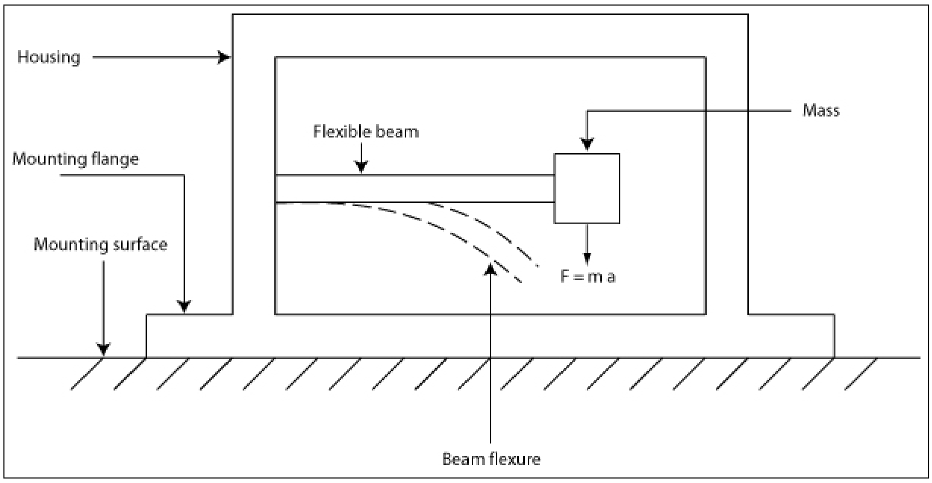

Figure 2.2: Basic setup for a beam type suspended mass accelerometer.

An accelerometer measures the acceleration, and by that the force ($F = m \cdot a$) that the unit is affected by. This is done either mechanically, or electronically. There are different types of accelerometers. One is the suspended mass type. It measures the force exposed to a known mass and thereby the acceleration of the system. A basic setup of this type is seen in Figure 2.2. The suspended mass accelerometer measures the acceleration in one of two main ways. Either by having strain gauges in the flexible beam or by using the mass and housing as the two plates of a capacitor. When the mass is moved, the signal that is proportional to the force is measured.

Another type is the electromechanical force-balance accelerometer. It uses a mass attached to a movable coil together with a permanent magnet to measure the force. When the mass wants to move due to the force, a current

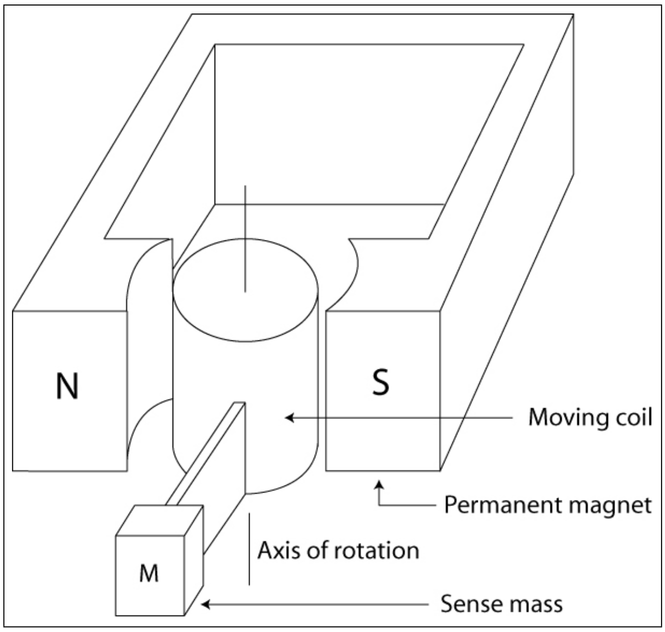

Figure 2.3: Basic setup for a beam type suspended mass accelerometer.

is sent through the coil, thus generating a force to stabilize the mass. This current $I \sim F \sim a$, and the acceleration is calculated. The setup is shown in Figure 2.3.

A third type is the piezoelectrical accelerometer. It uses piezoelectrical material as sensor. This material changes its electrical attributes according to the force exerted on it. Thereby, a signal proportional to the acceleration can be obtained. However, this type can not measure DC signals, making it a less suitable candidate for the Euro50 project.

According to [9], there are a couple of things to consider for the choice of an accelerometer. As the Euro50 is planned to make long time data acquisitions, a good DC response is important. AC accelerometers are typically better suited for fast shock measurements, like the ones in car crash testing. The mounting of the accelerometer to the surface is also important, but since it is most probable that an off-the-shelf IMU is bought, the fastenings of the different components should already have been properly made.

2.2 Noise

One key problem is how to characterize the noise from the accelerometers and gyros. Data sheets from IMU developers define units such as Random

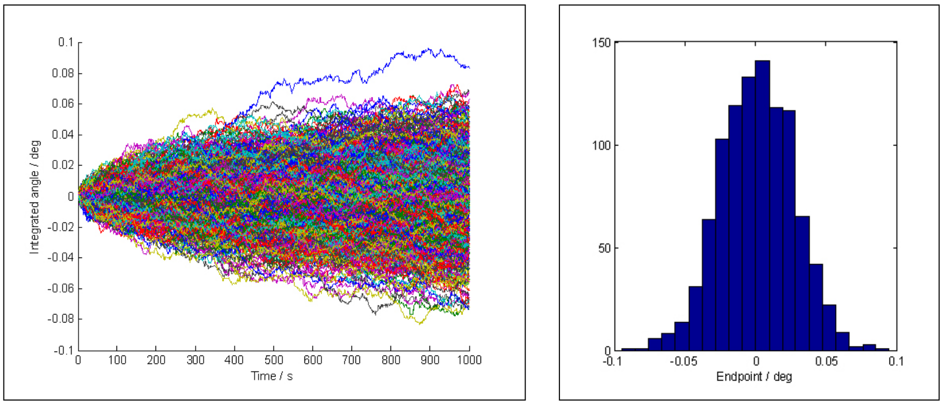

Figure 2.4: **left)** Plot of integration of noise for $1000s$, 1000 trials.
right) Distribution of the end points for the 1000 trials.

Walk[3], Bias Repeatability and Bias Stability with units $[°/\sqrt{h}]$, $[°/h]$ & $[°/h]$ / $[\frac{m/s}{\sqrt{h}}]$, $[\frac{m/s}{h}]$ & $[\frac{m/s}{h}]$ for gyros and accelerometers respectively. The most important one of these is the random walk parameter. It is a measure of how the standard deviation of the angular error (σ_θ) changes over time. This means that the standard deviation of the angular velocity measurement error must depend on time as $\sim \frac{1}{\sqrt{h}}$.

However, the unit of the angular deviation is $[°]$. Therefore, the unit of the standard deviation for the angular deviation must also be $[°]$, not $[°/\sqrt{h}]$. If a normally distributed stochastic variable, as described in [8],

$$X \in N(0, \sigma_{gyro}),$$

is integrated over time with a sampling time h, the result will be a new stochastic variable, Y. This variable will also be normally distributed, but with different attributes. The new variable will be $Y = \sum_{i=1}^{N} h \cdot X_i$, where N is the total number of samples.

The expected value will be

$$E[Y] = E[\sum_{i=1}^{N} h \cdot X_i] = h \cdot \sum_{i=1}^{N} E[X_i] = h \cdot \sum_{i=1}^{N} 0 = 0$$

and the variance will be

$$V[Y] = V\left[\sum_{i=1}^{N} h \cdot X_i\right] = h^2 \cdot \sum_{i=1}^{N} V[X_i] = h^2 \cdot N \cdot \sigma_{gyro}^2.$$

[3] More about ARW is found in [11] and [13].

Since the integration is over time and not over a number of samples, the expression can be written as

$$V[Y] = h^2 \cdot N \cdot \sigma_{gyro}^2 = h \cdot T \cdot \sigma_{gyro}^2.$$

since

$$T = N \cdot h \Leftrightarrow N = \frac{T}{h}.$$

With these attributes, the variable Y will be

$$Y \in N\left(0, \sqrt{h \cdot T} \cdot \sigma_{gyro}\right). \tag{2.1}$$

With the units $[^\circ/s]$, $[s]$ & $[s]$ for σ_{gyro}, h & T respectively, the unit for the standard deviation of Y will be $[^\circ]$. Figure 2.4 illustrates this, where the distribution Y has been plotted over time. Normally distributed noise has been integrated for 1000 s. As seen in the figure, Y is normally distributed. The variance of the end points grows over time. The calculations are the same for the accelerometers, where the only difference is that the unit is $[m/s^2]$ for $\sigma_{accelerometer}$.

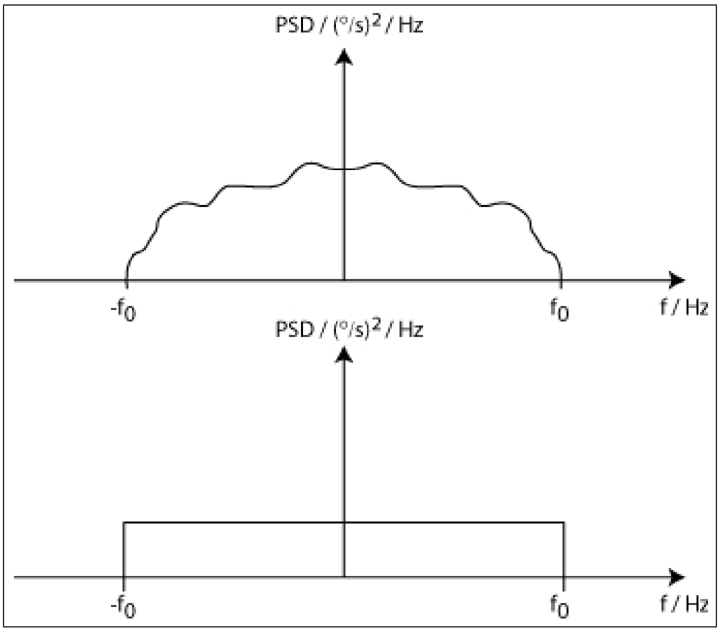

Figure 2.5: **upper)** A schematic figure of an arbitrary angular random walk PSD.
lower) PSD of a white noise angular random walk.
Integrating the PSD (positive frequencies only) gives σ^2 [8].

Another way to look at noise is through its respective frequency content, its Power Spectrum. This gives information on the different components of the noise and can be of great help for understanding noise behavior. Again, looking at data sheets for gyros and accelerometers these Power Spectral Density figures, or PSD figures, are sometimes listed.

A first guess is that the noise from the sensors is white. In practice, it is impossible for a continuous signal to be under influence of white noise since the PSD for white noise is constant for all measured frequencies. Since a continuous signal includes an unlimited range of frequencies, the integral of the PSD, which is the variance σ^2 of the noise, would be unlimited. In discrete time however, white noise is theoretically possible, if there is an upper limit of the frequencies. If there is an upper limit, and in discrete systems such a limit exists, there must be an internal sampling that corresponds to that frequency. According to the Nyquist theorem this sampling frequency must be two times larger than the upper limit frequency.

The question that arises is what will happen to the noise if the user does not use this high sampling frequency when utilizing the IMU. When sampling with a lower frequency, no noise component higher than half this sampling frequency will be visible, if the raw data is filtered through a low pass filter first[4]. When changing the visible noise, the PSD will also change. The new PSD will still be "box shaped", since the noise is still white, but it will have an upper limit that corresponds to the new lower, external, sampling frequency. This means that when integrating the new PSD, the resulting σ will be smaller than the one using a faster sampling. More background information about IMU noise is found in [15], [4] and [2].

2.2.1 Summary

Looking at the noise in these two ways, one of them states that if a faster sampling is utilized, there will be a lesser influence of the noise. On the other hand, the other method states the exact opposite, the noise influence will be lesser the slower the sampling frequency. But, by combining the two, the result will be more understandable. Assume a gyro with an ARW of $x^\circ/\sqrt{h}$ and an internal sampling frequency of $1000Hz$. The sampling time constant will then be $h = 1/1000\ s$. The corresponding standard deviation for the noise will then be some σ. If, for example, $h = 1/200\ s$ is used as the external sampling frequency, instead of $h = 1/1000\ s$, only a fifth of the PSD will be seen as the new sampling is five times slower than the maximal. This means that the integral of the PSD will be five times smaller as well,

[4] This must be done, otherwise there will be aliases from the high frequency noise.

and so will the variance of the noise, $\sigma_{new}^2 = \sigma_{old}^2/5$.

At the same time the first way of looking at the noise concludes that the variance will be five times larger, as the sampling time constant is five times larger. Thus, degrading the sampling frequency will have no influence of the noise coming from the gyro since the two compensates for each other as

$$ARW_{new} = \sigma_{new} \cdot \sqrt{\frac{h_{new}}{T}} = \frac{\sigma_{old}}{\sqrt{5}} \cdot \sqrt{\frac{h_{old} \cdot 5}{T}} = ARW_{old}$$

2.3 Overview of the project

Overcoming the challenges of the noise and the other obstacles calls for a structured project layout. Since all simulations will be made in Matlab and Simulink, a good model of the mirror cell, as well as models for the different noise sources must be derived. Then, different control strategies will be tried out, with the primary aim of suppressing the noise sufficiently. Wind disturbances on site must also be estimated and suppressed. This should be done by using as little control authority as possible. Since the mirror cell is expected to keep its position reference during the observations, the emphasis of this thesis does not lie in optimal step responses, but on the regulation problem.

3 Modeling of the System

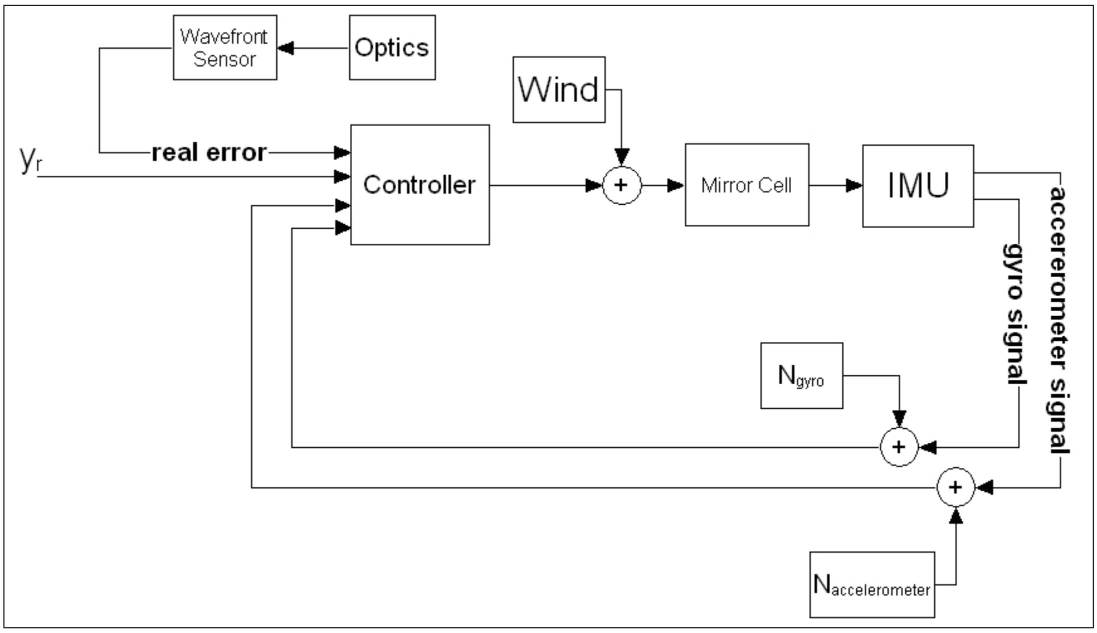

Figure 3.1: A very simplified view of the complete system.

The complete system is displayed in Figure 3.1. In this chapter we will describe the various parts of the system, such as the different sensors and wind influence. The controller will be described and discussed further in the next chapter.

3.1 Mirror Cell Structure

First, a couple of definitions must be made. Since the mirror cell has hexagonal shape, with force actuators fixed at the tripod structure at three of the six corners, see Figure 3.2, it is natural to call these corners 1, 2 & 3. At the other three corners, the force actuators will meet at the horisontal plane of gravity of the cell, so these corners are called A, B & C. By defining the corners in this fashion, the force actuators will be defined accordingly, starting at the leftmost corner (negative x-axis), 1A, 2A, 2B, 3B, 3C, 1C, going counter clockwise.

Also, it is necessary to define the six degrees of freedom used to describe the motion of the mirror cell. These are x-, y-, z-, ϕ-, θ- and ν-axes. The x-, y- and z-axes are self explanatory. The angles are defined as the clockwise rotation around the x-, y- and z-axes, in the order of ϕ, θ and ν.

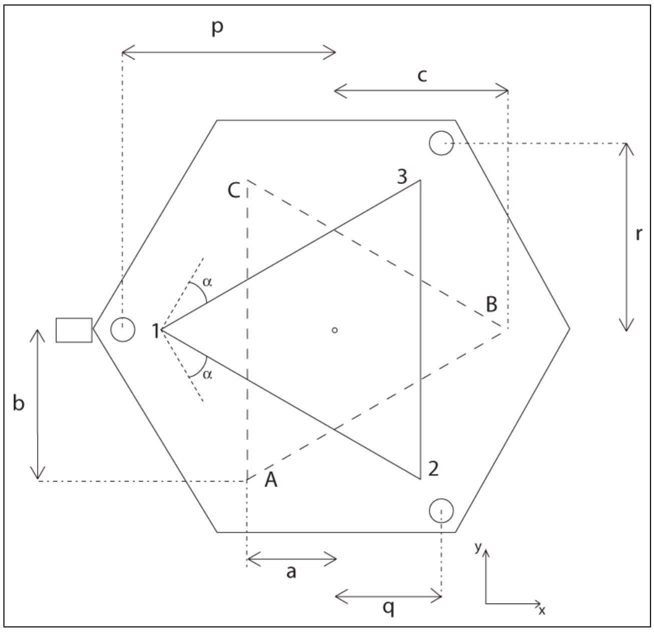

Figure 3.2: Definitions of the different distances for the mirror cell.

Next, there are some distances and angles which also must be defined. These are, as seen in Figure 3.2, the distances from the center of gravity, COG, to the insertion points of the force actuators (a, b & c), the distances from the COG to the different air bellows (p, q & r), also seen in Figure 3.2. Since the force actuators expose their forces in a hexapod structure, each will give force/moment components in all 6 degrees of freedom. For example, the 1A force actuator will have the folllowing component vector,

$$\begin{aligned}\begin{bmatrix} F_x \\ F_y \\ F_z \\ M_\theta \\ M_\phi \\ M_\nu \end{bmatrix} &= \begin{bmatrix} \sin(\alpha)\cdot\sin(\beta) \\ -\cos(\alpha)\cdot\sin(\beta) \\ \cos(\beta) \\ -\cos(\beta)\cdot a \\ \cos(\beta)\cdot b \\ -\sin(\alpha)\cdot\sin(\beta)\cdot b - \cos(\alpha)\cdot\sin(\beta)\cdot a \end{bmatrix} \cdot |F_{1A}| = \\ &= \begin{bmatrix} F_{1A_x} \\ F_{1A_y} \\ F_{1A_z} \\ M_{1A_\theta} \\ M_{1A_\phi} \\ M_{1A_\nu} \end{bmatrix} \cdot |F_{1A}|\end{aligned}$$

where α is the angle as seen in Figure 3.2 and β is the corresponding angle between the z-axis and the force actuator in the hexapod. The angular part of this matrix can of course also be viewed as

$$\begin{bmatrix} M_\theta \\ M_\phi \\ M_\nu \end{bmatrix} = \begin{bmatrix} -F_{1A_z} \cdot a \\ F_{1A_z} \cdot b \\ -F_{1A_x} \cdot b + F_{1A_z} \cdot a \end{bmatrix} \cdot |F_{1A}| \, .$$

By doing the same for all the force actuators, the transformation from the force actuator coordinate system to the 6DOF coordinate system is achieved. This matrix,

$$\begin{bmatrix} \sin(\alpha)\cdot\sin(\beta) & \sin(\beta) & \sin(\alpha)\cdot\sin(\beta) & \sin(\alpha)\cdot\sin(\beta) & -\sin(\beta) & \sin(\alpha)\cdot\sin(\beta) \\ -\cos(\alpha)\cdot\sin(\beta) & 0 & \cos(\alpha)\cdot\sin(\beta) & -\cos(\alpha)\cdot\sin(\beta) & 0 & \cos(\alpha)\cdot\sin(\beta) \\ \cos(\beta) & \cos(\beta) & \cos(\beta) & \cos(\beta) & \cos(\beta) & \cos(\beta) \\ -F_{1A_z}\cdot a & -F_{2A_z}\cdot a & F_{2B_z}\cdot c & F_{3B_z}\cdot c & -F_{3C_z}\cdot a & -F_{1C_z}\cdot a \\ F_{1A_z}\cdot b & F_{2A_z}\cdot b & 0 & 0 & -F_{3C_z}\cdot b & -F_{1C_z}\cdot b \\ -F_{1A_x}\cdot b+F_{1A_y}\cdot a & -F_{2A_x}\cdot b+F_{2A_y}\cdot a & -F_{2B_y}\cdot c & -F_{3B_y}\cdot c & -F_{3C_x}\cdot b+F_{3C_y}\cdot a & -F_{1C_x}\cdot b+F_{1C_y}\cdot a \end{bmatrix}$$

will be used in the simulations to translate the different force signals.

The placement of the inertial sensors also calls for some initial definitions. Proposing an arbitrary placement of the IMU at coordinate x_0, y_0 and z_0 also means there will be initial angles ϕ_0, θ_0 and ν_0 as in

$$\begin{aligned} \phi_0 &= \quad \arctan \tfrac{z_0}{x_0} \\ \theta_0 &= -\arctan \tfrac{z_0}{y_0} \\ \nu_0 &= -\arctan \tfrac{y_0}{x_0} \end{aligned}$$

These expressions are needed later on.

3.2 Sensor Transformation

Since it is the center of gravity of the cell that is to be controlled, that is also what is to be measured. Alas, this is obviously impossible, due to the fact that the COG is in the middle of the solid cell. By measuring the accelerations and angle velocities at a location different from the COG results in additional difficulties.

3.2.1 Rotation

The measured accelerations/velocities of the IMU are not the same accelerations/velocities that the COG "feels". If the cell has a tilt ($\theta \neq 0$) and is exposed of an acceleration in the x-direction (of the coordintate system, not the direction of the cell), the sensors will measure an accereration in the z-direction as well ($a_{z_{measured}} = a_x \cdot \sin\theta$) and the measured acceleration in the x-direction will not be what it should, only $a_{x_{measured}} = a_x \cdot \cos\theta$. For θ, this translation matrix will be

$$\ddot{X}_{measured} = \begin{bmatrix} \ddot{x}_{measured} \\ \ddot{y}_{measured} \\ \ddot{z}_{measured} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}.$$

This sensor translation can be applied for ϕ and ν as well. The resulting rotational transform matrices for all angles will then be

$$\left.\begin{array}{r} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \\ \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \\ \\ \begin{bmatrix} \cos\nu & -\sin\nu & 0 \\ \sin\nu & \cos\nu & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array}\right\} . \tag{3.1}$$

These matrices represent the transformation from the movements of the COG, but what is wanted is the transformation from the sensors to the COG. As for the force actuator matrix, an inversion is necessary. The problem with these transformations is that they do not commute. This means that it does not only matter which the current angles are at the moment, but also in which order they appeared. Since the angles are sufficiently small, a linearisation can be done to solve the problem. For angles $< 1.78°$, the approximation $\sin x \approx x$ and $\cos x \approx 1$ can be utilized[5]. Thus, getting the linerized matrices

[5] The error for an angle should be less than a tenth of the total allowed error ($5.556 \cdot 10^{-5\,\circ}$), *i.e.,* $|\sin x - x| < 5 \cdot 10^{-6}$ (x in rad), a value of $|x| < 0.03107 rad = 1.780°$ is derived.

$$\left.\begin{matrix} \begin{bmatrix} 1 & 0 & \theta \\ 0 & 1 & 0 \\ -\theta & 0 & 1 \end{bmatrix} \\ \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -\phi \\ 0 & -\phi & 1 \end{bmatrix} \\ \\ \begin{bmatrix} 1 & -\nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{matrix}\right\}, \tag{3.2}$$

a transformation between sensors and the COG, with respect to the cell's tip/tilt/rotation, can be done.

3.2.2 Sensor Translation

As stated in the definitions, it is impossible to place the IMU in the cell's COG. Therefore, there will be problems measuring the angular movements. If the mirror cell tilts $\delta\theta$, the sensors will feel two movements. One angular, same as the real $\delta\theta$, but also one movement in the direction of the tangent of the COG/sensor circle[6]. Depending of the initial placement of the sensors, the influence of this translational error will be more or less noticeable. For a rotation $\delta\theta$, the translational matrices from the movements of the COG to the measurements of the IMU will be

$$\begin{bmatrix} \delta\ddot{x} \\ \delta\ddot{y} \\ \delta\ddot{z} \\ \delta\ddot{\theta} \\ \delta\ddot{\phi} \\ \delta\ddot{\nu} \end{bmatrix}_{measured} = \begin{bmatrix} -\sin(\theta_0+\theta)\cdot|r_\theta| \\ 0 \\ \cos(\theta_0+\theta)\cdot|r_\theta| \\ 1 \\ 0 \\ 0 \end{bmatrix} \cdot \delta\ddot{\theta}.$$

Doing the same for ϕ and ν,

$$\begin{bmatrix} \delta\ddot{x} \\ \delta\ddot{y} \\ \delta\ddot{z} \\ \delta\ddot{\theta} \\ \delta\ddot{\phi} \\ \delta\ddot{\nu} \end{bmatrix}_{measured} = \begin{bmatrix} 0 & 0 & 0 & -\sin(\theta_0+\theta)\cdot|r_\theta| & 0 & -\sin(\nu_0+\nu)\cdot|r_{nu}| \\ 0 & 0 & 0 & 0 & sin(\phi_0+\phi)\cdot|r_\phi| & sin(\nu_0+\nu)\cdot|r_\nu| \\ 0 & 0 & 0 & \cos(\theta_0+\theta)\cdot|r_\theta| & 0 & -\cos(\nu_0+\nu)\cdot|r_\nu| \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \delta\ddot{x} \\ \delta\ddot{y} \\ \delta\ddot{z} \\ \delta\ddot{\theta} \\ \delta\ddot{\phi} \\ \delta\ddot{\nu} \end{bmatrix}$$

[6] The sensors will perform a circular motion when the cell is conducting a tip/tilt/rotation, with the radius of $\sqrt{x_0^2+z_0^2}$ for the θ-axis.

is obtained. The pure x-, y- and z-axes motions will bring no further problems other than the ones discussed previously.

3.2.3 Conclusion

By combining both the rotation and sensor translation matrices, the complete transformation matrix is obtained,

$$\begin{bmatrix} 1+\theta\cdot\phi\cdot\nu & -\nu+\theta\cdot\phi & \theta & -\sin(\theta_0+\theta)\cdot|r_\theta| & 0 & -\sin(\nu_0+\nu)\cdot|r_\nu| \\ \nu & 1 & -\phi & 0 & sin(\phi_0+\phi)\cdot|r_\phi| & \sin(\nu_0+\nu)\cdot|r_\nu| \\ -\theta+\phi\cdot\nu & \theta\cdot\nu+\phi & 1 & \cos(\theta_0+\theta)\cdot|r_\theta| & 0 & -\cos(\nu_0+\nu)\cdot|r_\nu| \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Note that this transformation matrix still represents the transformation *from* the COG *to* the sensors. This is done beacause of the easier visualization in that direction. The desired matrix is simply the inversion, seen in Appendix A.1.

3.3 A Model of the Mirror Cell

Deriving a model for the mirror cell is pretty straight forward. Ideally, the mirror cell can be seen as a hexagonal shaped rigid body with a large mass m. Then, a suitable model is a simple double integrator which is a state space representation of Newton's second law, $\overline{F} = m \cdot \overline{a}$ for the lateral positions. For the tip/tilt measurements the formula $\overline{I} = m \cdot \overline{\omega}$, a rewriting of Newtons second law, is used. Using a second order system for the model is justified by one of the original problem specifications. Saying that the mirror cell should be controlled by a bandwidth of $10Hz$ leads to the fact that frequencies (high) above the bandwidth limit do not have the same impact on the system. The mirror cell eigenfrequencies will be above $10Hz$ by a wide margin.

The force actuators also need to be modeled. Since it is not yet decided which type of force actuator that will be used, a generic type with no dynamics ($y = K_{FA} \cdot u$, *i.e.,* no effects such as friction are accounted for), will be used as a model. By using these two systems as a basic model of the mirror cell and its control actuators, the following system model can be derived. The equations

$$\left.\begin{aligned} \overline{F} &= m \cdot \overline{a} \\ \overline{I} &= m \cdot \overline{\omega} \\ \overline{y}_{FA} &= K_{FA} \cdot u \end{aligned}\right\}$$

lead to the state space formulation

$$\dot{\overline{X}} = \begin{bmatrix} \overline{x}_1 \\ \overline{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \cdot X + \begin{bmatrix} 0 \\ \frac{K_{FA}}{m} \end{bmatrix} \cdot u \qquad (3.3)$$

where

$$\begin{aligned} \overline{x}_1 = & \begin{bmatrix} x & y & z & \theta & \phi & \nu \end{bmatrix}^T \\ \overline{x}_2 = & \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} & \dot{\theta} & \dot{\phi} & \dot{\nu} \end{bmatrix}^T. \end{aligned}$$

3.3.1 Air Bellows

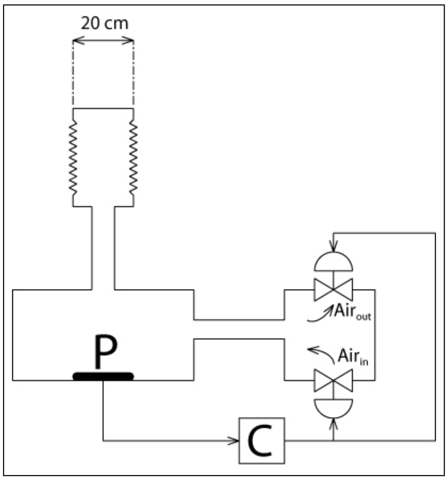

Figure 3.3: The strucural design and control of an air bellow.

To get rid of the stiff connection between the tripod and the mirror cell, soft air bellows are attached, acting as a low pass filter for the mirror cell. The effect is modeled by springs attached to the cell. Although the spring constant is very small, it will affect the mirror cell making the cell less damped. The bellows are designed to only be stiff in their respective direction, *i.e.,* when shifting position in the y-axis, the cell will not feel any of the four bellows. When introducing these bellows, the process changes, making the differential equations of the system[7] look like

$$\left.\begin{aligned} m \cdot a_x = & \ F_x - k_{spring_x} \cdot x \\ m \cdot a_z = & \ F_z - 3k_{spring_z} \cdot z \\ m \cdot I_{yy} = & \ F_\theta - k_{spring_z} \cdot (2r^2 + p^2)\,\theta \\ m \cdot I_{xx} = & \ F_\phi - k_{spring_z} \cdot 2q^2\phi \end{aligned}\right\}. \tag{3.4}$$

The complete state space system can be seen in Appendix A.2. The air bellows are controlled via direct air pressure control. By assuming the equation

[7] The degrees of freedom that will change its system behavior are x, z, θ and ϕ. This is due to the fact that these degrees of freedom have air bellows acting parallel to their respective axes.

of state for an ideal gas, $pV = nRT$, setting all paraameters except p and n to constant values, the expression $p \sim n$ is found. Then V has to be sufficiently large to be assumed to have a constant value. To calculate the volume of the bellows tank, the statement is made, that when compressing the bellow, the volume must not change more than 0.1%. Since the diameter of the bellows is 0.20 m and the expected maximum deviation of each bellow is 5 mm, the demanded volume of the tank will be $0.157m^3$. These figures are based on

$$\begin{array}{r} D = 0.20m \Rightarrow A = 0.1^2\pi m^2 \\ \Rightarrow \delta x = 0.005mm \\ \Rightarrow \delta V = 0.1^2\pi \cdot 5 \cdot 10^{-3} \leq V \cdot 0.001 \\ V \geq \frac{0.1^2\pi \cdot 5 \cdot 10^{-3}}{1 \cdot 10^{-3}} = 0.157m^3 \end{array} . \tag{3.5}$$

3.4 Wind

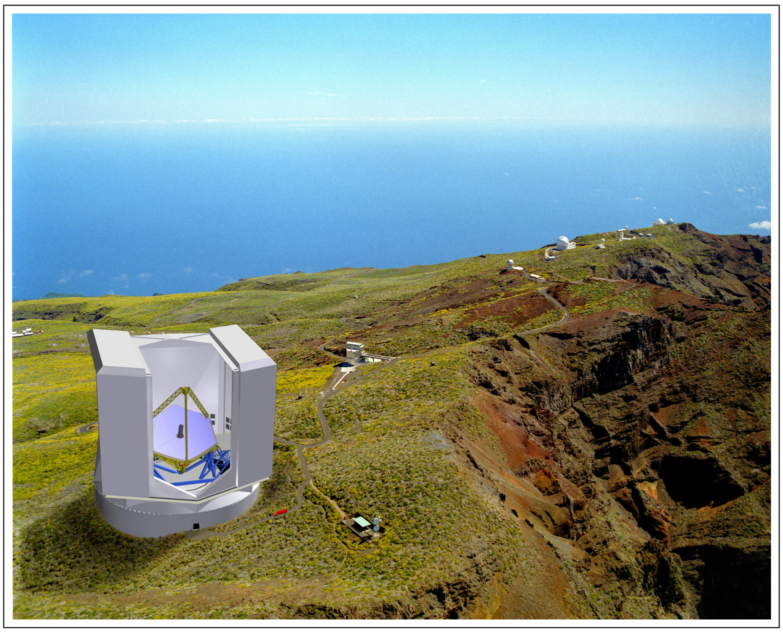

Figure 3.4: One of the proposed locacions of the Euro50 ELT, La Palma, Spain.

With the telescope located on open ground[8], the wind influence is an important issue. The need of a good wind model to test the ability of the controllers to counteract the wind forces is of great importance.

[8] One of the proposed locations is La Palma, Spain.

There are a number of models for deriving the power spectral density of the wind speed. One of the simplest, but powerful enough for this application, is the Davenport model, see [14], [6] and [10]. This model neglects the turbulence on and around the mirror cell, thus only taking the free flow wind velocities into account. The PSD is, according to the Davenport model,

$$PSD(f) = \frac{4u_\star^2}{f} \cdot \frac{X^2}{(1+X^2)^{4/3}}, \tag{3.6}$$

where f is frequency, $u_\star$ the shear velocity and X defined as

$$X = \frac{1200f}{u(10)}.$$

$u(10)$ is the mean velocity 10 m above ground. To be able to calculate the mean wind velocity at 60 m above ground, the equation

$$u(z) = 2.5u_\star \cdot \ln\left(\frac{z}{z_0}\right)$$

is used, where z_0 is seleced on the basis of terrain roughness. This equation can also be used to calculate $u_\star$, when the wind velocity at 10 m is specified. A modification of this model can, and should be made. The wind has some energy at high frequencies, which comes from small eddies with a certain physical size in the air. High spatial frequencies (small eddies) correspond to high temporal frequencies. However, there is an attenuation when the eddies are small in size compared to the mirror cell, which makes the high frequency wind forces on the cell smaller than the original model states. The mirror cell is said to act as a natural low pass filter. To get the modifed PSD, a model for the spatial distribution of the turbulence caused by the eddies has to be applied. The chosen turbulence model is

$$Q = \frac{1}{1+\left(2f\frac{\sqrt{A}}{u(z)}\right)^{4/3}}, \tag{3.7}$$

where A is the cross section area of the mirror cell. To apply this model, the PSD is simply multiplied by Q^2, where Q is defined as above. The PSD for both versions of the model can be seen in Figure 3.5.

For later simulations, a time series of simulated wind speeds is necessary. The wind speed time series is generated according to the **direct cosine method**, as a sum of cosine waves with amplitudes determined by the PSD.

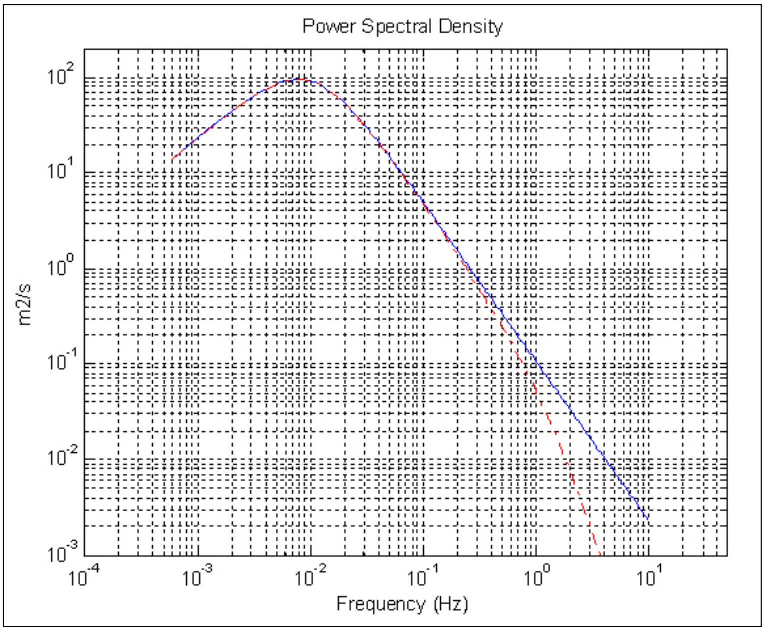

Figure 3.5: Wind power spectral density for the undisturbed (solid curve) and for the attenuated (dashdotted curve) model.

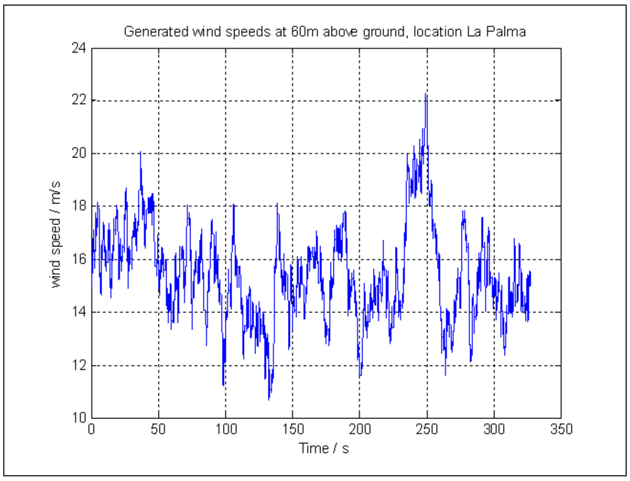

Figure 3.6: A typical time series of wind generated by the Davenport model, realized by the **direct cosine method**.

To calculate the speeds as a random process, random phases are added to all cosine waves.

$$u(t) = \sum_{k=1}^{N} \sqrt{2S(f_k)\Delta f} \cos\left(2\pi f_k t + \phi_k\right) \tag{3.8}$$

Here, Δf is frequency resolution, t time, N the number of frequency values used and ϕ_k uniformly distributed random phase angles. For a realization of wind speeds, see Figure 3.6. For confirming the validity, an estimation of the time series power spectral density is computed. This estimation, together with the Davenport model it should resemble, is shown in Figure 3.7.

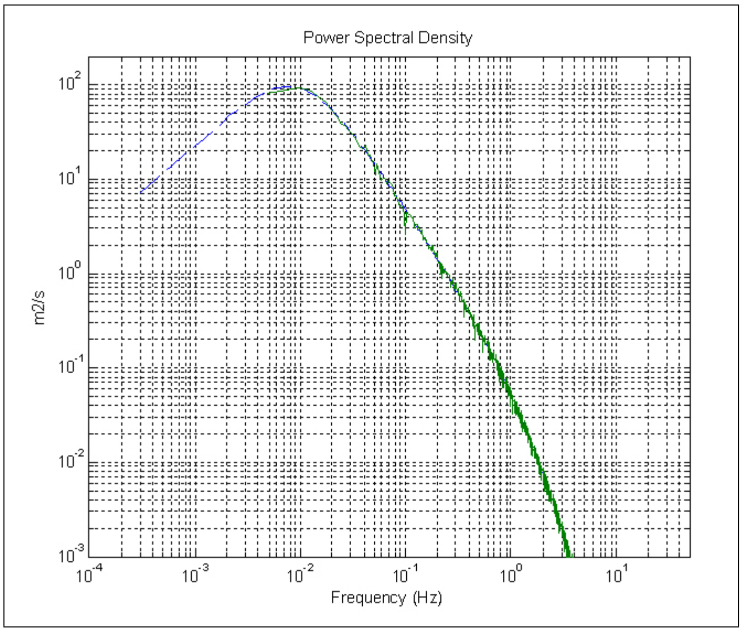

Figure 3.7: Estimated PSD for the wind time series (uneven curve) compared with the Davenport PSD (dashed curve).

3.5 Wavefront Sensor

An important part of the setup is the wavefront sensor, as it is the only sensor that is capable of measuring the actual, and supposedly, noise free[9] measurements. As the sensor measures the actual errors and not the acceleration or the velocity, it is modeled as a pure double integrator without disturbances.

[9] Probably, the meassurements will not be noise free, but as the wavefront sensor is the component with the highest resolution, it is supposed that the noise will be of a lesser magnitude than the resolution needed for this thesis' specifications

In reality, the error is the output, but in the simulations the actual position is the output.

There is no measurement data for the ν-axis. Since the mirror cell is (almost) rotation symmetrical, the wavefront sensor has little or no ability to measure the rotation deviations. This is neglected in the simulations but must be solved for the real implementation. For instance, the rotational movement could be measured by putting encodes or other kinds of sensors on each of the six force actuators. By measuring the position of all six actuators, a rough estimate of the cell position can be obtained. It will not be as accurate as the signals from the wavefront sensor, but since the position of the rotational axis is not as important as the other axes, it will not affect the control of the mirror.

4 Control

An overview of the control system was shown in Figure 3.1. In this section we will describe the details of the different controllers.

4.1 Overview

As stated before, the aim of this work is to be able to control the secondary mirror cell with the aid of an IMU. The IMU is used for feedback information creating a loop, see Figure 4.1.

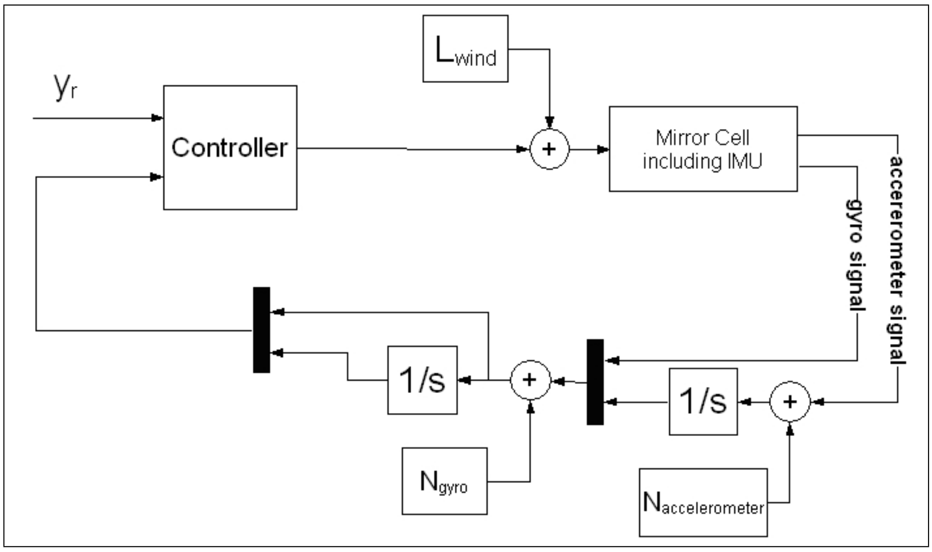

Figure 4.1: The IMU model from Figure 3.1 is now made explicit.

Since the information obtained from the IMU is influenced by noise and bias, it cannot be trusted for very low frequencies. By retrieving information from the wavefront sensor, the errors of the different degrees of freedom[10], can be found. This creates a new *outer* loop, which utilizes the wavefront sensor's ability to detect the actual errors. Together, these two loops form the basic control loops, as can be seen in Figure 4.2.

These loops will later be completed with a feed forward system & model following to be able to handle changes in the control signal and a compensation link, to improve the phase margin.

With this configuration, the inner loop is forseen to control fast and relatively large events while the outer loop is designed to take care of slow

[10] All except the ν-axis.

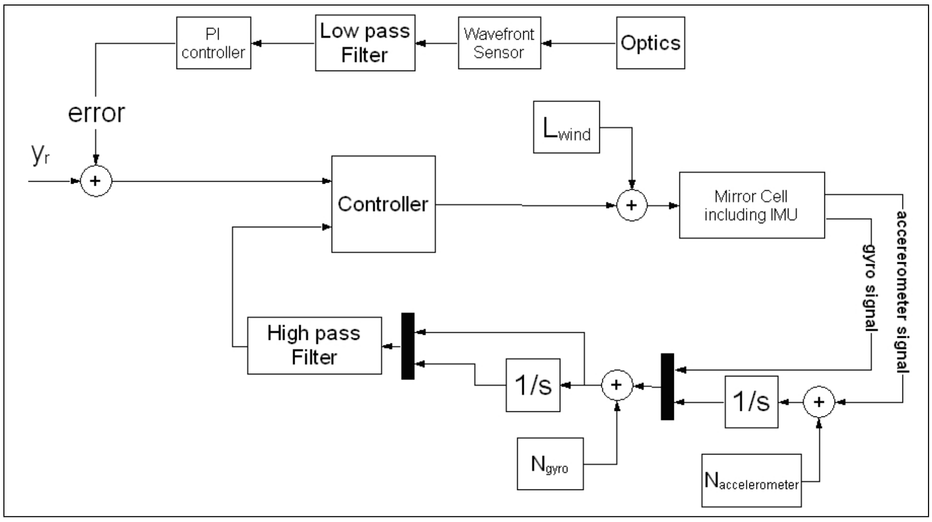

Figure 4.2: Together, the wavefront sensor and the IMU are used to form the basic control loops.

dynamics. Inserting a lowpass filter of second order before the outer PI-controller makes the outer loop react only on slow signals. The same action is taken in the inner loop, but with a second order highpass filter before the respective controller. These highpass/lowpass filters were tuned to a cut off frequency of $0.5/0.1Hz$, which seemed suitable and later proved to work.

Now, a definition of the control strategy must be made, the movement of the mirror cell and its coordinate system. Since the telescope is to be tilted and rotated, and the IMU will sense these movements, the mirror cell has to be aligned with the telescope at all times. This can be done by retrieving tilt/rotation information from the sensors of the telescope and just substracting these numbers from the command signal of the cell.

4.2 Derivation of alternative controllers

Here, theoretical descriptions of the different controller structures are found.

4.2.1 PI control

One of the simplest and most widely spread controllers is the PID-controller. The control law is built up from the three parts (Proportional, Integral, Derivative). The first part, P, acts as $u_P(t) = K \cdot e(t)$, where e is the control error $y_{ref}(t) - y(t)$. This leads to the problem of a stationary error, as the

P-controller needs an error to generate a control signal[11]. To remedy this, the I-part is included in the control law as $u_I = \frac{K}{T_i}\int_0^T e(t)dt$. Since the I-part will add (or substract) to the control signal as long as there is an error, whereas the stationary error will go to zero. How fast depends on T_i, which can be interpreted as the time horizon for the I-part. The D-part will add to the control signal depending on the time derivative of $e(t)$, as $u_D = -KT_d \cdot \frac{de(t)}{dt}$ Since the D-part reacts on the derivative of the control error, it is very noise sensitive. As the main purpose of this thesis is to suppress noise, the D-part will not be included in the controller. More information is found in [5].

4.2.2 Linear Quadratic Control

As with all automatic control, the systems may be implemented in continuous or discrete time. For this thesis a discrete Linear Quadratic, or LQ-controller has been derived, making it easier to implement digitally later. Stationary LQ control has many similarities with pole placement state feedback, but the control law $u(k) = -Lx(k)$ is derived in a completely different way. The state feedback controller's parameters are computed with the aid of the desired closed loop behavior, the stationary LQ-controller's parameters are computed by minimizing a loss function. This loss function, typically looking as

$$J = \int_0^{Nh} |x(t)|^2 \, dt = \int_0^{Nh} x(t)^T x(t) dt,$$

is used to weight the magnitudes of the states. Because all states may not have the same dimension, the loss function of the states more generally looks like

$$J = \int_0^{Nh} x(t)^T Q_{1c} x(t) dt,$$

where Q_{1c} is a symmetric positive semidefinite matrix. The states at the end time can be penalized in the same way, by introducing

$$\begin{aligned} J = E\Bigg(& \textstyle\int_0^{Nh} \big(x(t)^T Q_{1c} x(t) + 2x^T(t) Q_{12c} u(t) \\ & + u^T(t) Q_{2c} u(t)\big)\, dt + x^T(Nh) Q_{0c} x(Nh)\Bigg) \\ = E & \left(\textstyle\int_0^{Nh} \begin{bmatrix} x^T(t) & u^T(t) \end{bmatrix} Q_c \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} dt + x^T(Nh) Q_{0c} x(Nh) \right). \end{aligned} \tag{4.1}$$

[11] Here there are integrators in the process which helps the control signal following, but the integrator in the controller is still needed to get rid of load disturbances.

Q_c in this equation consists of four sub matrices,

$$Q_c = \begin{bmatrix} Q_{1c} & Q_{12c} \\ Q_{12c}^T & Q_{2c} \end{bmatrix},$$

where Q_{0c}, Q_{1c} and Q_{2c} are symmetric and positive semidefinite. Since the sampling period is periodic and the control signal is constant over the whole period[12], a discretization of the system is easily made, with the sampling period h. The control problem is now to derive the control law that minimizes the loss function in (4.1). Since the LQ-controller, in this case, should be in discrete-time, the loss function needs to be sampled. By using the fact that u is constant over the sampling period, the sampled loss function will be of the form

$$J(x,u) = \begin{bmatrix} x^T & u^T \end{bmatrix} \begin{bmatrix} Q_x & Q_{xu} \\ Q_{xu}^T & Q_u \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}, \tag{4.2}$$

where

$$Q_x = \int_{kh}^{kh+h} \Phi^T(s,kh) Q_{1c} \Phi(s,kh) ds \tag{4.3}$$

$$Q_{xu} = \int_{kh}^{kh+h} \Phi^T(s,kh)(Q_{1c}\Gamma(s,kh) + Q_{12c}) ds \tag{4.4}$$

$$Q_u = \int_{kh}^{kh+h} (\Gamma^T(s,kh) Q_{1c} \Gamma(s,kh) + 2\Gamma^T(s,kh) Q_{12c} + Q_{2c}) ds \tag{4.5}$$

The next step is to find the minimum of this loss function, (4.2), with respect to u. If these prerequisites are fulfilled, there exists an L satisfying

$$Q_u L = Q_{xu}^T$$

such that (4.2) can be rewritten as

$$J(x,u) = x^T(Q_x - L^T Q_u L)x + (u + Lx)^T Q_u (u + Lx). \tag{4.6}$$

(4.6) is minimized by using the control law $u = -Lx$. Also, L is unique if Q_u is positive definite. By inserting this control law into (4.6), it is seen that the minimum is

$$J_{min} = x^T(Q_x - L^T Q_u L)x$$

What this means in reality is that the controller will drive the states to their desired values as fast as possible with as little control power as possible. The speed and power of this is determined by the original Q_c. If

[12] This control strategy is called zero order hold

Q_{1c} is relatively larger than Q_{2c}, the controller is allowed to use more control authority to drive the states to their values. One rule of thumb to find Q_c is to let the diagonal elements in Q_{1c} be the inverse of the maximum allowed deviation of each state. The same can be applied for the control signal u.

So far, it has been shown, by rough hand calculations, that an L that minimizes the discretized loss function can be found. L itself is still to be derived. The way it is derived is by using the **principle of optimality** and the concept of **dynamic programming**. In short terms, dynamic programming uses backward iteration to determine the optimal control signal for each step. By starting at time $N-1$, the last control signal can be derived. Next, the signal from time $N-2$ to $N-1$ is found. By going back in time, step by step, u from time N to time 0 is derived.

Consider a system described by

$$\begin{aligned} x(kh+h) =& \Phi x(kh) + \Gamma u(kh) \\ y(kh) =& Cx(kh), \end{aligned} \tag{4.7}$$

that is, a system with no influence of either process or measurement noise. By using dynamic progamming, thus letting $u(kh)$ be a function of $x(kh), x(kh-h), \ldots$. Introducing the **discrete-time Ricatti equation**

$$\begin{aligned} S(kh) =& \Phi^T S(hk+h)\Phi + Q_x - \left(\Phi^T S(kh+h)\Gamma + Q_{xu}\right) \cdot \\ & \cdot \left(\Gamma^T S(kh+h)\Gamma + Q_u\right)^{-1} \left(\Gamma^T S(kh+h)\Phi + Q_{xu}^T\right) \end{aligned} \tag{4.8}$$

with the end condition $S(N) = Q_0$, the unique control stategy $u(kh) = -L(kh)x(kh)$ can be derived, where

$$L(kh) = \left(Q_u + \Gamma^T S(kh+h)\Gamma\right)^{-1} \left(\Gamma^T S(kh+h)\Phi + Q_{xu}^T\right). \tag{4.9}$$

Since this controller is time varying, and needs to be computed every time step, only the stationary controller[13] is used in this thesis. Otherwise, since the mirror cell should be able to follow a command signal, the L from time kh to time $kh + N$, would have to be calculated every time the command signal changes, where N is the control time horison.

4.2.3 Linear Quadratic Gaussian Control

In the LQ section, an optimal controller was derived. This was for the deterministic case, with no noise or disturbances affecting the system. This,

[13] When calculating the $S(kh)$, iterating until a constant S is obtained gives the stationary LQ-controller

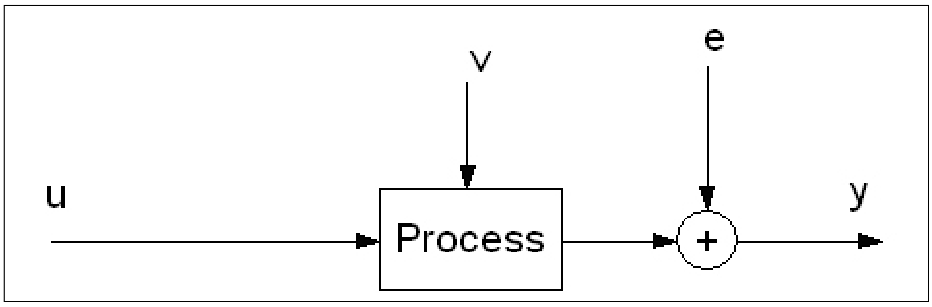

Figure 4.3: Noise modeled to enter the system.

however, is not how reality works. The strong noise influence of the gyros/accelerometers has been discussed earlier. It would be of great importance if the signals from the IMU could be filtered in such a way that the output of the filter would be more trustworthy than the raw output. One such filter is the Kalman filter based on the Linear Quadratic Gaussian, or LQG, design. The system used for the LQ design, (4.7), is now extended with process and measurement noise,

$$\begin{aligned} x(kh+h) =& \Phi x(kh) + \Gamma u(kh) + v(kh) \\ y(kh) =& Cx(kh) + e(kh) \end{aligned} \tag{4.10}$$

These disturbances may be viewed as noise entering the system as in Figure 4.3. The idea behind the LQG design is to let the Kalman filter know the variance of these white noise sources, and then estimate what the real output should be. To derive such a filter a one-step-ahead estimator on the form

$$\hat{x}(kh+h|kh) = \Phi\hat{x}(kh|kh-h) + \Gamma u(kh) + K(kh)\left(y(kh) - C\hat{x}(kh|kh-h)\right) \tag{4.11}$$

is proposed[14]. By combining (4.10) and (4.11), the estimation error $\tilde{x} = x - \hat{x}$ will be

$$\begin{aligned} \tilde{x}(kh+h) =& \Phi\tilde{x}(kh) + v(kh) - K(kh)\left(y(kh) - C\hat{x}(kh|kh-h)\right) \\ =& \left(\Phi - K(kh)C\right)\tilde{x}(kh) + v(kh) - K(kh)e(kh) \\ =& \begin{bmatrix} I & -K(kh) \end{bmatrix} \left(\begin{bmatrix} \Phi \\ C \end{bmatrix} \tilde{x}(kh) + \begin{bmatrix} v(kh) \\ e(kh) \end{bmatrix} \right) \end{aligned} \tag{4.12}$$

The aim of using LQG control is to minimize the variance of the estimation error, $P(kh)$.

$$P(kh) = E\left(\tilde{x}(kh) - E\tilde{x}(kh)\right)\left(\tilde{x}(kh) - E\tilde{x}(kh)\right)^T \tag{4.13}$$

[14] The notation $\hat{x}(kh|kh-h)$ means that the current estimation $\hat{x}(kh)$ is based on the previous measurements up until and including time $kh - h$.

Using the fact that $\tilde{x}(kh)$ is independent of both $v(kh)$ and $e(kh)$, it follows that the variance of the estimation error is minimized with respect to $K(kh)$ by

$$K(kh) = \left(\Phi P(kh)C^T + R_{12}\right)\left(R_2 + CP(kh)C^T\right)^{-1}, \tag{4.14}$$

where R_1 and R_2 are the variance of v and e respectively, and R_{12} is the covariance between v and e. The variance of the estimation error will then be

$$\begin{aligned} P(kh+h) =& \Phi P(kh)\Phi^T + R_1 \\ &- \left(\Phi P(kh)C^T + R_{12}\right)\left(R_2 + CP(kh)C^T\right)^{-1}\left(CP(kh)\Phi^T + R_{12}^T\right) \\ P(0) =& R_0 \end{aligned} \tag{4.15}$$

This estimator has the same appearance as the L vector in the LQ-controller, where only the stationary Kalman filter is used. The design of the L matrix and the K matrix are in fact dual operations. By iterating (4.15) until $P(kh+h) = P(kh)$, the K obtained from (4.14).

4.2.4 Air Bellows

Since the air bellows are included in the process model, there must be a model of the air bellows included in the simulations as well. As stated in Section 3.3.1, the bellows are modeled as integrators, with the transfer function

$$G_p(s) = \frac{k_{bellow}}{s}. \tag{4.16}$$

These bellows are supposed to relieve the extra pressure from the force actuators, caused by the force of gravity. The setup for this controller is a little bit different. The output y from the bellows system is added to the total force applied to the process, thus offloading that force from the force actuators. The system should not drive the output to equal the input, but drive the input to zero while maintaining the output. By modeling the input (the force from the actuators) as the error signal $y_r - y$, a closed loop is created without the need of internal feedback. The calculations can still be made as usual, assuming a negative feedback. Since it is a simple process, a P-controller, with the gain K_p, is sufficient to drive the one state to zero.

$$\begin{aligned} G_0 =& \frac{k_{bellow}K_P}{s} \Rightarrow \\ G_{CL} =& \frac{k_{bellow}K_P}{s + k_{bellow}K_p} \end{aligned}$$

The bandwidth of $0.5H$ makes $K_P = \frac{0.5 \cdot 2\pi}{k_{bellow}}$ a good choice as the controller gain.

4.2.5 Model Following

When introducing a step response for a system, an unnaturally big error occurs, which may cause actuators to saturate. By introducing model following, the effective command signal will instead be the ideal step response signal. The model is done by creating a system with the specified attributes, *e.g.,* desired bandwidth and damping. In that case, a model of second order can look like

$$G(s) = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}. \tag{4.17}$$

The output signal from this model will then be the command signal to the process, yielding a well suited signal for the system to follow.

4.2.6 Feed Forward

Since common control theory often deals with the regulation problem[15] and this thesis includes the servo problem[16], a feed forward link is favourably implemented as a part of the controller. This is done by calculating the preferable control signal in open loop, according to step response specifications, to bring the states to their desired values. By looking at the process in open loop, the system

$$y = G \cdot u_{ff} = G \cdot G_{ff} \cdot u_c$$

is valid. If the wanted system behavior is G_m, u_{ff} should ideally be $u_{ff} = \frac{G_m}{G} u_c$, rendering

$$y = G \cdot u_{ff} = G \cdot \frac{G_m}{G} u_c = G_m \cdot u_c \tag{4.18}$$

as the desired feed forward open loop system. Of course the model G_m must be stable. Also, the pole excess of the model must not be less than the pole excess of the process, and unstable zeros in the process must also be zeros in the model [12].

4.2.7 Compensation Link

Sometimes, the phase margin[17] for a system is too small and cannot be compensated for by changing the controller parameter values. In that case, a phase lead compensation link can provide the aid needed. By creating

[15] The control has one goal, to bring all states to zero.

[16] A specified command signal should be followed.

[17] The phase at the cut-off frequency + 180°, or the margin down to 180°, is the phase margin.

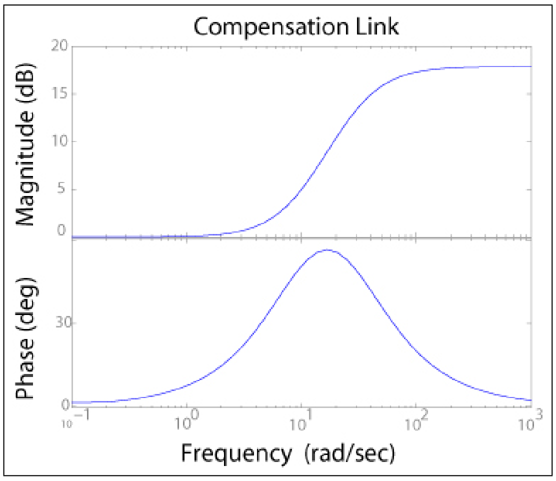

Figure 4.4: By creating a lead filter the phase margin can be increased.

a transfer function in a special way, an improvement in the phase at the specified frequency can be achieved. For example, the transfer function

$$G(s) = \frac{\frac{s}{10}+1}{\frac{s}{28}+1}$$

will result in the lead filter shown in Figure 4.4. The problem with lead filters are, as the figure shows, that the gain increases as well as the phase, making the system more sensitive to noise.

4.2.8 Integral Action

When having disturbancies acting on a system, it is necessary to be able to estimate the position of the states as accurately as possible. This can be done in several ways. The easiest way would be to connect an integrator to the signal $y_r - y$. In this thesis a different approach has been used. This method is based on the extended use of a Kalman filter, *e.g.,* like the one used in Section 4.2.3. Assuming a disturbance acting on the process, the system looks as

$$\begin{aligned}\frac{dx}{dt} &= Ax + B\left(u+v\right)\\ y &= Cx\end{aligned}$$

where v is the disturbance, in this case the wind. Modeling v as

$$\begin{aligned} \frac{dw}{dt} &= A_w w \\ v &= C_w w, \end{aligned}$$

with w as the states of v. By introducing these states in the system, the augmented system

$$z = \begin{bmatrix} x \\ w \end{bmatrix}$$

is created. The complete system now looks like

$$\begin{aligned} \tfrac{dz}{dt} &= \begin{bmatrix} A & \beta C_w \\ 0 & A_w \end{bmatrix} z + \begin{bmatrix} \beta \\ 0 \end{bmatrix} u \\ y &= \begin{bmatrix} C & 0 \end{bmatrix} z. \end{aligned} \tag{4.19}$$

Sampling this system leads to the discrete-time system

$$\begin{aligned} z(kh+h) &= \begin{bmatrix} \Phi & Phi_{xw} \\ 0 & \Phi_w \end{bmatrix} z(kh) + \begin{bmatrix} \Gamma \\ 0 \end{bmatrix} u(kh) \\ y(kh) &= \begin{bmatrix} C & 0 \end{bmatrix} z(kh). \end{aligned} \tag{4.20}$$

Introducing the observer injection

$$\begin{bmatrix} K \\ K_w \end{bmatrix} (y(kh) - C\hat{x}(kh)) \tag{4.21}$$

makes the estimated system to be

$$\begin{aligned} \begin{bmatrix} \hat{x}(kh+h) \\ \hat{w}(kh+h) \end{bmatrix} &= \begin{bmatrix} \Phi & \Phi_{xw} \\ 0 & \Phi_w \end{bmatrix} \begin{bmatrix} \hat{x}(kh) \\ \hat{w}(kh) \end{bmatrix} + \\ &+ \begin{bmatrix} \Gamma \\ 0 \end{bmatrix} u(kh) + \begin{bmatrix} K \\ K_w \end{bmatrix} (y(kh) - C\hat{x}(kh)) \qquad (4.22) \\ y(kh) &= \begin{bmatrix} C & 0 \end{bmatrix} z(kh). \end{aligned}$$

Using the new control law

$$u(kh) = -L\hat{x}(kh) - L_w\hat{w}(kh) \tag{4.23}$$

ensures that v goes to zero in the output y if the disturbance has been modeled correctly. The closed loop system will look like

$$\begin{aligned} x(kh+h) =& (\Phi - \Gamma L)x(kh) + (\Phi_{xw} - \Gamma L_w)w(kh) - \Gamma L\tilde{x}(kh) - \Gamma L_w\tilde{w}(kh) \\ w(kh+h) =& \Phi_w w(kh) \qquad (4.24) \\ \tilde{x}(kh+h) =& (\Phi - KC)\tilde{x}(kh) + \Phi_{xw}\tilde{w}(kh) \\ \tilde{w}(kh+h) =& \Phi_w\tilde{w}(kh) - K_w C\tilde{x}(kh), \end{aligned}$$

where

$$\begin{aligned} \tilde{x}(kh) &= x(kh) - \hat{x}(kh) \\ \tilde{w}(kh) &= w(kh) - \hat{w}(kh) \end{aligned}$$

i.e. the estimation error. Left is the decision of the vectors L_w and K. As Equation 4.24 implies, the ideal selection of L_w would be if $\Phi_{xw} - \Gamma L_w = 0$. This may not be possible, but calculating L_w as

$$L_w = \Gamma^{-1\star}\Phi_{xw}, \qquad (4.25)$$

where $\Gamma^{-1\star}$ denotes the pseudo inverse of Γ, is a good guess. Also, using pole placement as a way of chosing proper K and K_w, gives the system described by the estimation error good closed loop behavior.

If the disturbance is completely unknown, a pure integal can be implemented to ensure that the stationary error goes to zero. In that case, Φ_{xw} becomes Γ, and by using Equation 4.25, it is intuitive to see that L_w becomes 1. In this thesis, both the approach with the wind model from Section 3.4 and with an unknown disturbance have been tested with the wind time series computed from the wind model, to see if there is a significant difference.

4.3 Cascade Control

Asymptotic design methods give good overview of the process to be controlled. As the process can be assumed to be a pure double integrator[18], the magnitude plot will be a straight line with negative slope 40 *dB/decade*. By using the velocity signals from the IMU (one mathematical integration of the axial components) to create a feedback loop, it is easy to tune the chosen controller to the desired bandwidth. By doing the same with the IMU's position signals the full inner cascaded controller setup is complete. The controller setup can be seen in Figure 4.5.

[18] The modeled spring constants from the air bellows are so small in comparison, that they may be neglected here

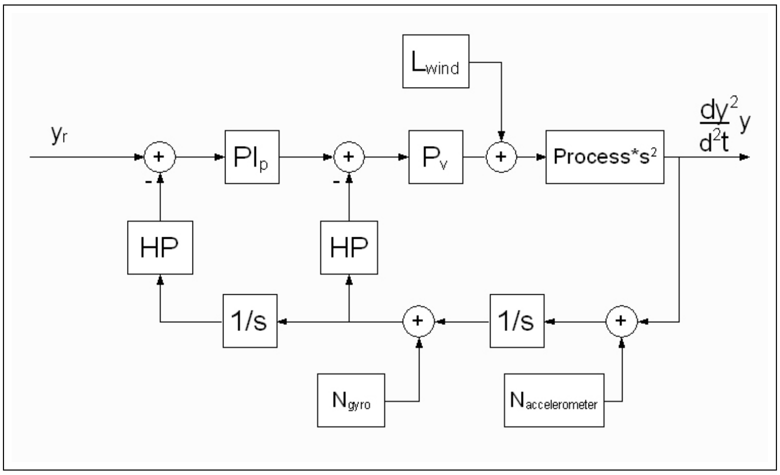

Figure 4.5: The inner controller setup using cascaded PI/P controllers.

When tuning the controllers it was found that the inner (velocity) controller gave as good results when implemented just as a P-controller as it was when implemented as a PI-controller. This might be explained by saying that integrational control of the velocity signal is basically the same thing as proportional control of the position signal (which is integrated one time from velocity).

The results, with respect to step response and the use of control authority, from this first setup is seen in Figure 4.6. By adding the model following and the compensation link, the step response can be improved dramatically, as shown in Figure 4.7

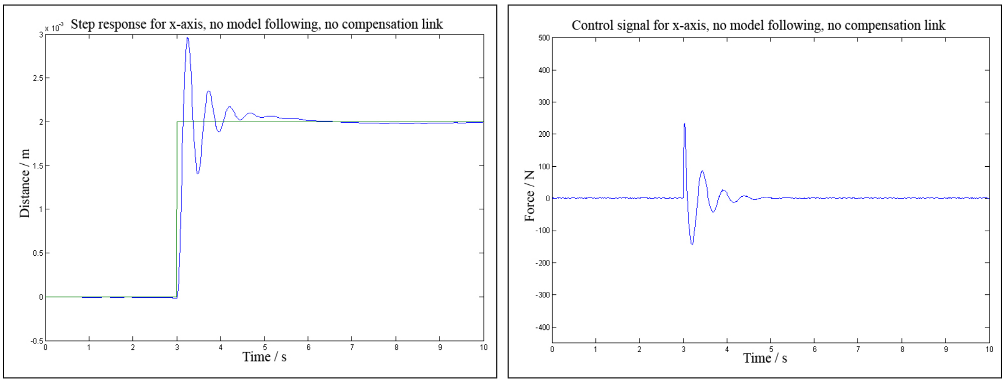

Figure 4.6: Step response of $2 \cdot 10^{-3}$ m with corresponding control signal, no compensation link or model following.

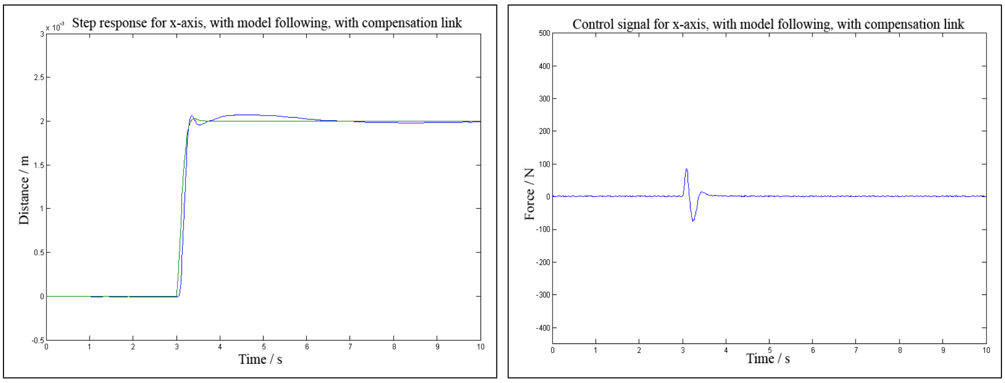

Figure 4.7: Step response of $2 \cdot 10^{-3}\ m$ with corresponding control signal, both compensation link and model following.

4.3.1 Transfer Functions

Important information can be obtained from the different transfer functions of the system. In this section the different transfer functions, both for open and closed loops as for noise and load disturbances are presented.

Since the loops are controlled by cascaded PI-controllers, analytical expressions for the open/closed loops are relatively easy to derive. The following example is for the x-axis. The shape of the transfer functions for the other axes are similar, and therefore not presented. As the inner control loop looks as Figure 4.5, the open loop transfer function from y_{ref} to y will be

$$\begin{aligned} Y(s) &= \frac{HP}{s^2} \cdot Process \cdot P_v s^2 \left(PI_p Y_r(s) - sY(s)\right) \Rightarrow \\ G_{innerOpen}(s) &= \frac{HP \cdot Process \cdot P_v \cdot PI_p}{1 + HP \cdot Process \cdot P_v s} \end{aligned}$$

The transfer function above, in this form, is valid for all axes. The closed loop,

$$\begin{aligned} Y(s) &= \frac{HP}{s^2} \cdot Process \cdot P_v s^2 \left(-sY(s) + PI_p \left(Y_r(s) - Y(s)\right)\right) \Rightarrow \\ G_{innerClosed}(s) &= \frac{HP \cdot Process \cdot P_v \cdot PI_p}{1 + HP \cdot Process \cdot P_v \left(PI_p + s\right)} \end{aligned}$$

will therefore also be valid for all axes. A load/wind disturbance will have the following effect on the closed system:

$$\begin{aligned}
Y(s) &= \frac{HP}{s^2} \cdot Process \cdot s^2 \left(L_{wind} + P_v \left(-sY(s) - PI_p \cdot Y(s)\right)\right) \Rightarrow \\
G_{L\rightarrow Y}(s) &= \frac{HP \cdot Process}{1 + HP \cdot Process \cdot P_v \left(PI_p + s\right)}
\end{aligned} \tag{4.26}$$

This is also valid for all axes. Since the noise will not be measurable in the inner loop, it is not meaningful to derive an expression for it. Only the noise transfer function for the outer loop will be derived. The transfer function for the outer loop, modeled as Figure 4.8, will look like

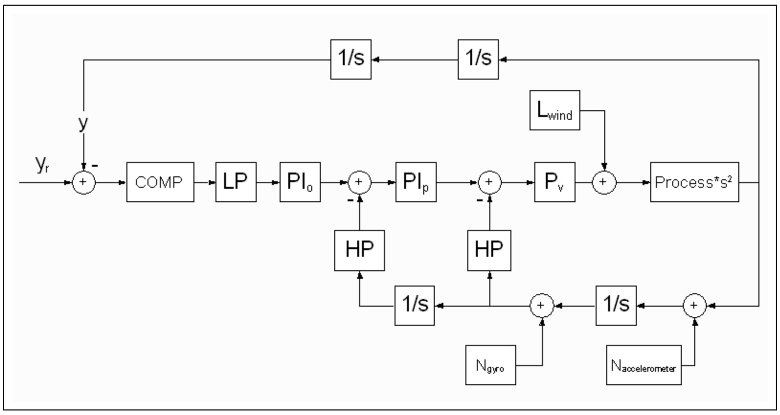

Figure 4.8: The outer controller setup, modeled with two integrators (upmost) instead of the natural double integration that the wave front sensor will perform.

$$\begin{aligned}
Y(s) &= \frac{1}{s^2} Process \cdot s^2 \Bigg(L_{wind}(s) + P_v \Bigg(-HPsY(s) - \frac{HP}{s} N(s) + \\
&\quad + PI_p \Bigg(-HP \cdot Y(s) - \frac{HP}{s^2} N(s) + PI_o LP \cdot COMP \cdot Y_r(s)\Bigg)\Bigg)\Bigg) \Rightarrow \\
&(1+ \; Process \cdot HP \cdot P_v s + Process \cdot PI_p P_v HP)\, Y(s) = \\
&= Process \cdot L_{wind}(s) - \frac{Process \cdot P_v \left(HP \cdot s + PI_p HP\right)}{s^2} N(s) + \\
&+ Process \cdot P_v \cdot PI_p PI_o LP \cdot COMP \cdot Y_r(s) \Longrightarrow \\
G_{outerOpen}(s) &= \frac{Process \cdot P_v PI_p PI_o LP \cdot COMP}{1 + Process \cdot HP \cdot P_v s + Process \cdot PI_p P_v HP}
\end{aligned}$$

and the closed loop transfer functions from the noise N, wind load L_{wind} and reference signal y_r to y, by using the same technique, will be

$$G_{Y_r \to Y} = \frac{Process \cdot P_v PI_p PI_o LP \cdot COMP}{1 + P_v Process\left(HPs + PI_p HP + PI_p PI_o LP \cdot COMP\right)}$$

$$G_{N \to Y} = \frac{Process \cdot P_v\left(HPs + PI_p HP\right)}{s^2\left(1 + P_v Process\left(HPs + PI_p HP + PI_p PI_o LP \cdot COMP\right)\right)}$$

$$G_{L_{wind} \to Y} = \frac{Process}{1 + P_v Process\left(HPs + PI_p HP + PI_p PI_o LP \cdot COMP\right)}$$

4.3.2 Bode Plots and Power Spectral Densities

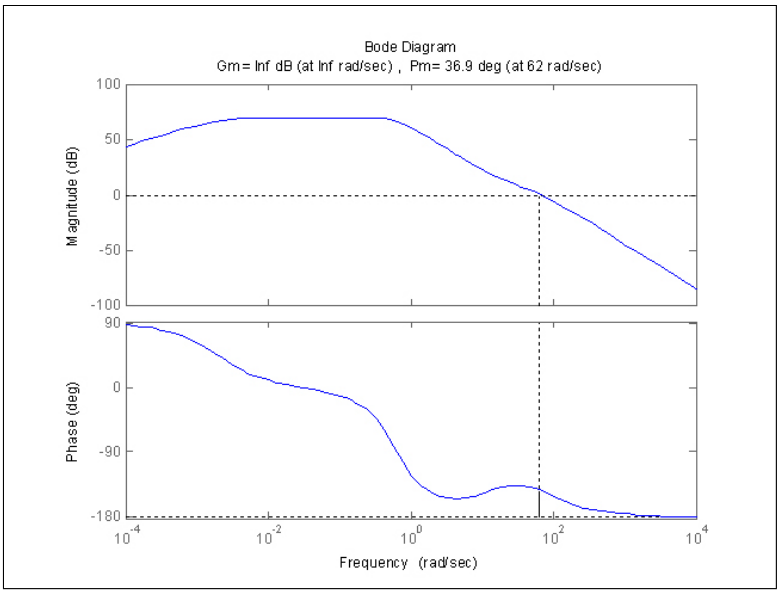

Figure 4.9: Bode plot for the inner controller setup, open system. The vertical line indicates the phase margin.

While previous sections gave insight in how the transfer functions were derived, they do not give valuable information of how the system behaves, nor do they tell how the system is influenced by noise and disturbances in reality. By inserting the values of the different block parameters, Bode plots can easily be calculated. Interpreting these Bode plots helps to understand the expected system frequency responses. The inner loop, that is controlled by one PI- and one P-controller, was specified to work with a bandwidth of $\approx 10Hz \approx 63\ rad/s$. As seen in Figure 4.9, this is accomplished.

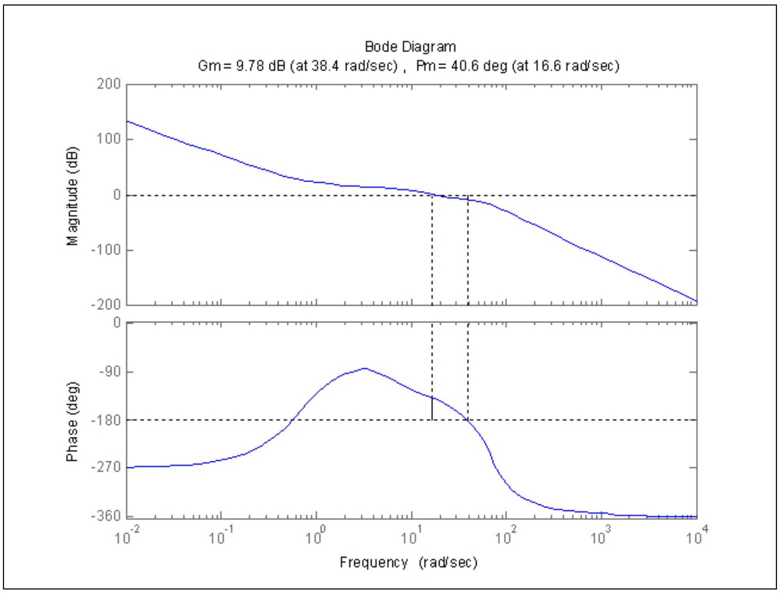

Figure 4.10: Bode plot for the outer controller setup, open system.
The vertical lines indicates the phase and gain margins.

The outer loop is specified to work at a bandwidth of roughly $2-3Hz \approx 12-19\ rad/s$ This is also accomplished, as seen in Figure 4.10. It is also important to know how disturbances affect the system, especially as it is known that the measurements from the IMU are strongly influenced by noise. The four plots in Figure 4.11 show the closed loop behavior for the inner and outer controller setup.

Bode plots are good tools when it comes to frequency response, but another important issue is the actual frequency content of the outgoing noise and load disturbancies. According to IMU developer data sheets, the noise of both gyros and accelerometers is white. Thus the PSD of the noise is the square of the transfer function, *i.e.,* the square of the Bode plots in Figure 4.11. However, the wind disturbance is not white and therefore it gives extra information to calculate the outgoing PSD of the wind, when the incoming wind has the spectral distribution according to the PSD in Figure 3.7. $Y(s) = G(S) \cdot U(s)$ is valid for any linear system. From this, the equation for spectral density can be obtained as $PSD\left(Y\left(s\right)\right) = G(s)\overline{G}(s) \cdot PSD\left(U\left(s\right)\right)$, which also is valid for all linear systems. $\overline{G}(s)$ is the conjugate of $G(s)$. By using this relation together with the PSD from Figure 3.5 together with the transfer function from Equation 4.26, the PSD of the effect of the wind can be calculated, see Figure 4.12.

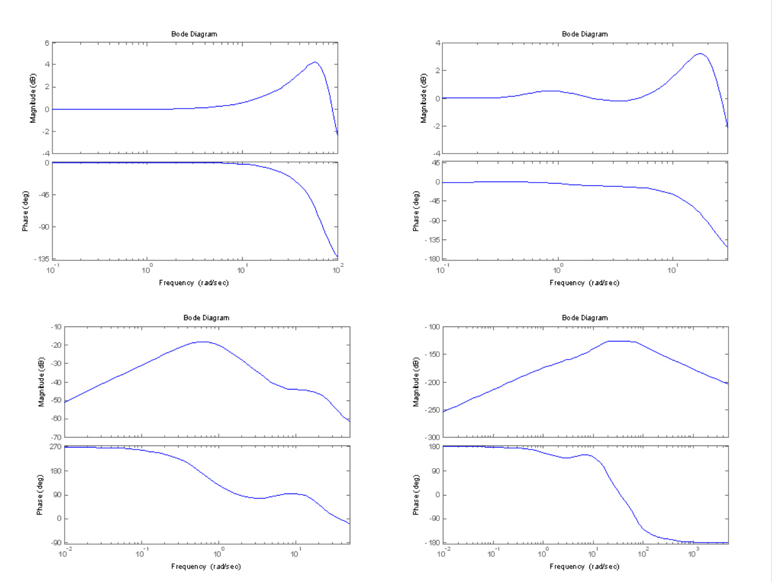

Figure 4.11: Closed loop behavior: **upper left)** $y_{ref} \rightarrow y_i$ inner loop,
upper right) $y_{ref} \rightarrow y$ outer loop,
lower left) $N \rightarrow y$ outer loop,
lower right) $L \rightarrow y$ outer loop.

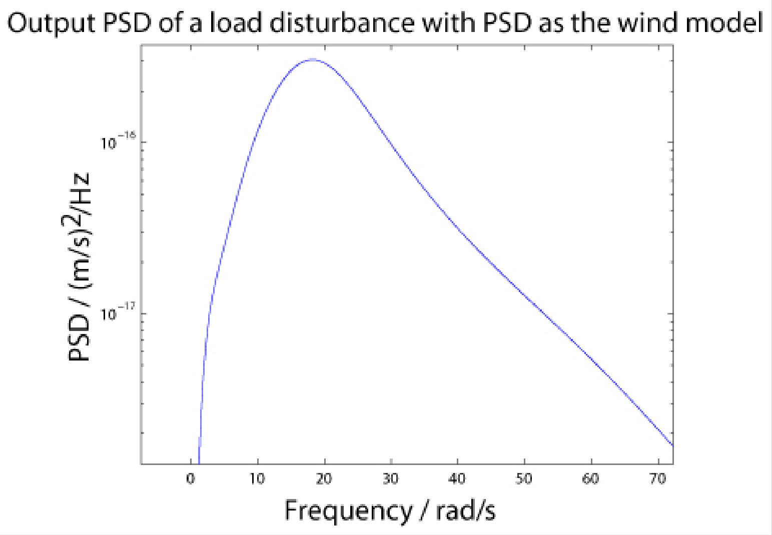

Figure 4.12: The resulting PSD of the wind when using the Davenport model from Section 3.4.

4.4 Linear Quadratic Control

By combining the theory in Section 4.2.2 and Section 4.2.8 a powerful controller is derived[19], which is designed to use as little control authority while keeping the output y as close to the reference y_r as possible.

4.4.1 Structure of the controller

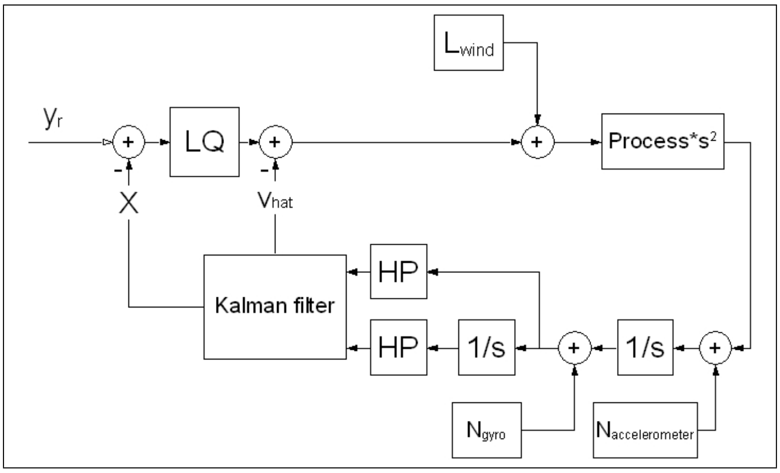

Figure 4.13: The inner controller structure when using LQ/LQG control.

Comparing Figure 4.8 with Figure 4.14 it can be seen that the only difference in appearence is that the inner PI- & P-controllers have been replaced by a Kalman filter and a LQ-controller. But there is more to it than what first meets the eye. First of all, as stated in Section 4.2, the LQ/LQG-controllers are implemented in a discrete control system. By itself, this does not bring any impovement in system stability or controllability. On the contrary, using certain sample times may cause the system to be unstable. But as the controller, whichever is to be chosen, is implemented digitally, it must be discretized. The discrete LQ/LQG-controller itself does not differ from its continuous equivalent.

As suggested above, the choice of a sample time is very important. With the Nyquist theorem in mind, the sampling frequency must be at least the double of the bandwidth. This is not the ultimate selection and a better choice of the sampling period $h = \frac{1}{f_{sampling}}$ is to follow the rule of thumb

[19] No Bode plots of the system using this controller will be presented, due to the similarities with the following LQG-controller.

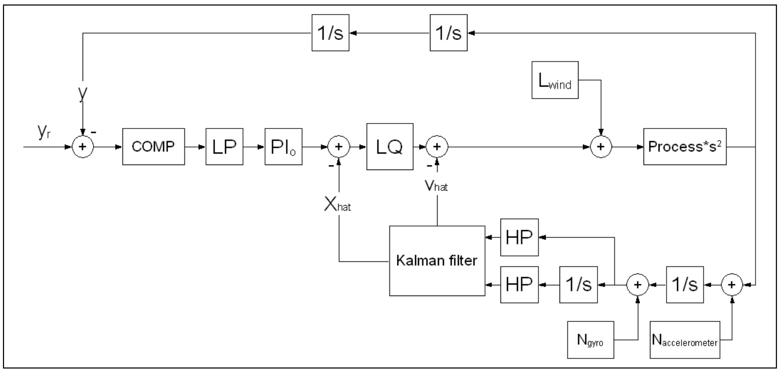

Figure 4.14: The outer controller structure when using LQ/LQG control.

$0.2 \leq w_c h \leq 0.6$, where w_c is the cut-off frequency[20] in rad/s. w_c for the inner (fastest) system is, according to Figure 4.18, 13.3 Hz. This leads to $h = 0.005$ as a proper choice of the sampling time.

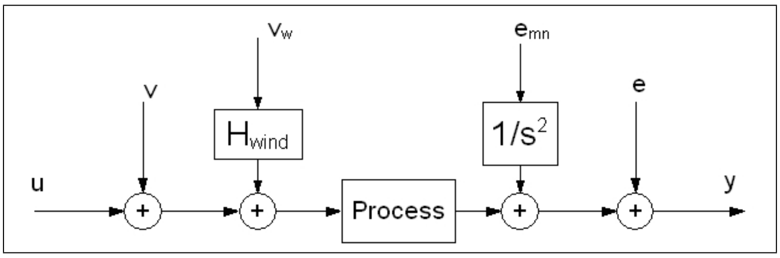

Figure 4.15: The extended noise model, including accelerometer/gyro measurement noise and wind disturbances. The model in the figure is for accelerometers, there is only on integrator in the transfer function for e_{mn} for the gyros, as they measure velocity.

The LQG-controller was derived in the following manner: First, modeling of the different noise sources was necessary. Besides the ones in Figure 4.3, the measurement noise from the accelerometers/gyros and the disturbance from the wind is taken into account, see Figure 4.15. A remark is that the wind model was implemented only for the x-, y- and z-axes. The wind is not expected to have the same impact on the rotational axes, and was therefore

[20] The frequency where the phase goes below 180°.

left out[21]. This leads to the extended system

$$\begin{aligned}
\begin{bmatrix} X(kh+h) \\ X_w(kh+h) \\ X_{mn}(kh+h) \end{bmatrix} &= \begin{bmatrix} \Phi & \Gamma C_w & 0 \\ 0 & \Phi_w & 0 \\ 0 & 0 & \Phi_{mn} \end{bmatrix} \begin{bmatrix} X(kh) \\ X_w(kh) \\ X_{mn}(kh) \end{bmatrix} + \\
&+ \begin{bmatrix} \Gamma \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \Gamma v(kh) \\ \Gamma_w v_w(kh) \\ \Gamma_{mn} e_{mn}(kh) \end{bmatrix} \qquad (4.27) \\
y(kh) &= \begin{bmatrix} C & 0 & C_{mn} \end{bmatrix} \begin{bmatrix} X(kh) \\ X_w(kh) \\ X_{mn}(kh) \end{bmatrix} + e(kh)
\end{aligned}$$

where

$$\begin{bmatrix} \Gamma v(kh) \\ \Gamma_w v_w(kh) \\ \Gamma_{mn} e_{mn}(kh) \end{bmatrix}$$

from here on denoted $\tilde{v}$, corresponds to v in Figure 4.3. When deriving a LQG-controller, the Kalman filter needs to know the different variances and covariances of the different noise components. In this case, all noise sources (v, v_w, e_{mn} & $e(t)$) are supposed to be independent of each other[22]. The covariance matrices for the system will then look like

$$\begin{aligned}
R_1 = R_{\tilde{v}} = E\left[\tilde{v}\tilde{v}^T\right] &= \begin{bmatrix} \Gamma R_v \Gamma^T & 0 & 0 \\ 0 & \Gamma_{v_w} R_{v_w} \Gamma_{v_w}^T & 0 \\ 0 & 0 & \Gamma_{e_{mn}} R_{e_{mn}} \Gamma_{e_{mn}}^T \end{bmatrix} \qquad (4.28) \\
R_2 = R_e &
\end{aligned}$$

By tuning the parameters, a single controller that handles the necessary dynamics for the mirror cell was derived. The step responses, without/with compensation link and model following, together with corresponding control signal is seen in Figures 4.16 and 4.17.

4.4.2 Bode Plots and Power Spectral Densities

Since the pulse transfer functions are in discrete-time, they are less intuitive than the continuous-time transfer functions. Therefore, only the relevant

[21] Since the Davenport model models the wind as a "layer" moving in the air with the same velocity everywhere, it will not create moments on the cell, therefore not affecting θ, ϕ or ν.

[22] It seems reasonable that the wind has nothing to do with the IMU's measurement errors.

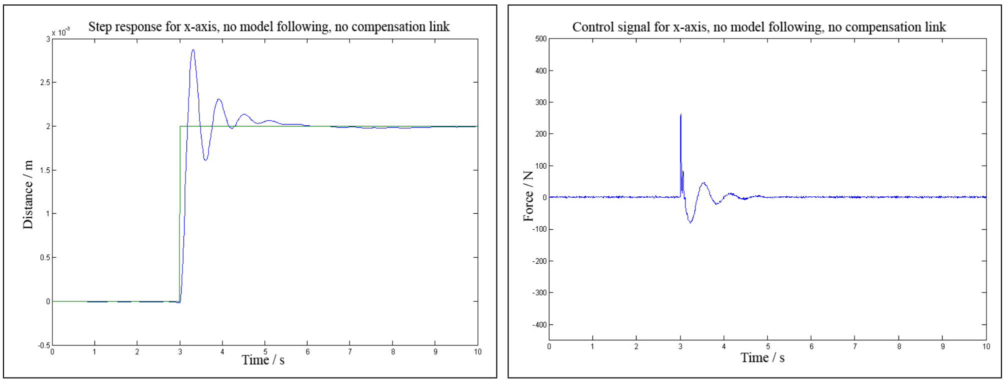

Figure 4.16: Step response of $2 \cdot 10^{-3}$ m with corresponding control signal, no compensation link or model following, LQG-controller used.

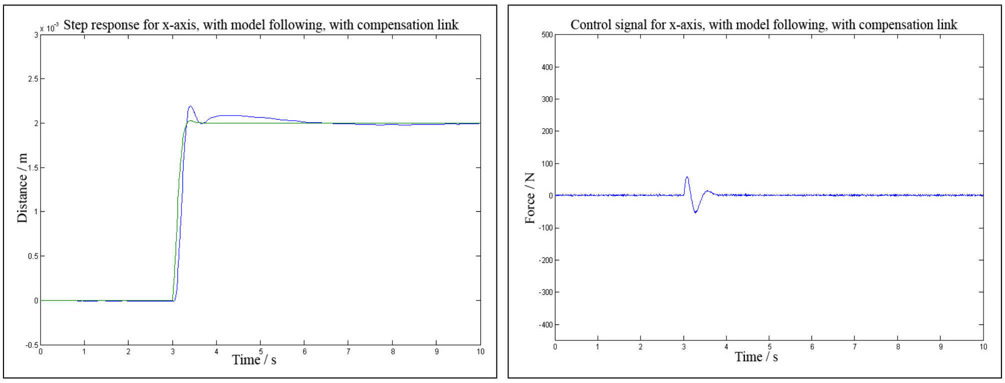

Figure 4.17: Step response of $2 \cdot 10^{-3}$ m with corresponding control signal, both compensation link and model following, LQG-controller used.

Bode plots will be presented. As in Section 4.3.2, the corresponding Bode plots (for the x-axis) will be presented here. Bode plots from the other axes will not be presented, as they are similar to the x-axis.

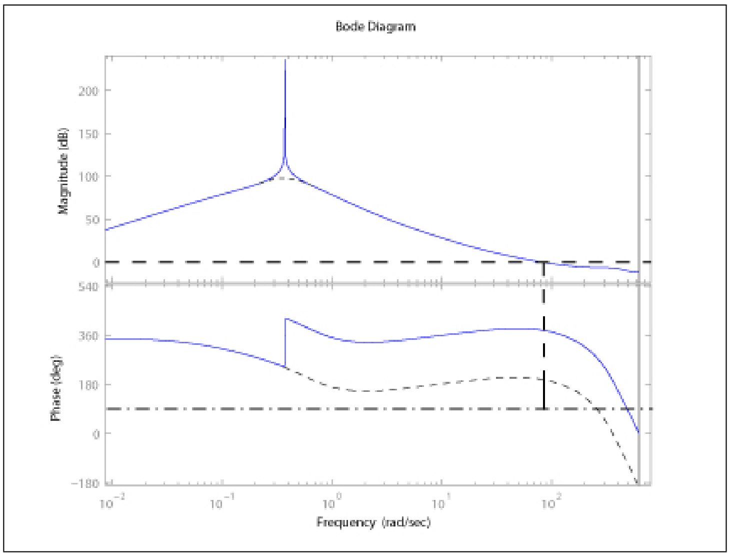

Figure 4.18: Bode plot of the open inner control loop. The phase is biased, and the plot should follow the dotted curve. Due to phase unwrapping issues, error occurs. Dash-dotted line indicates the real $-180°$.

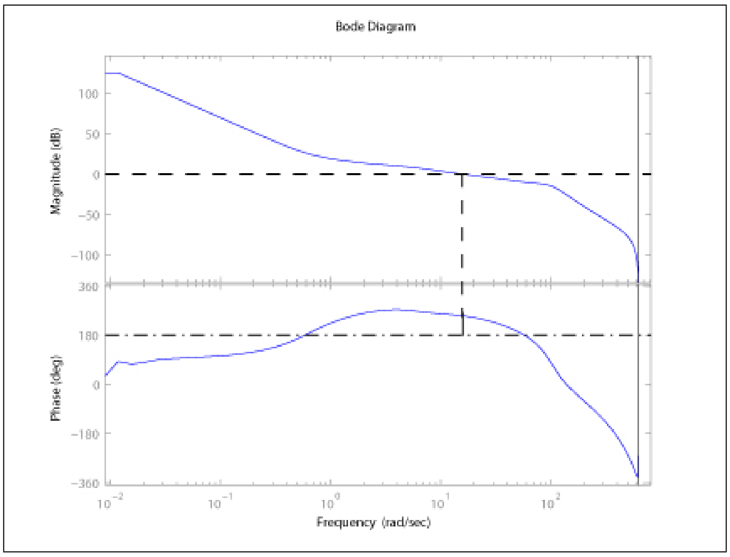

Figure 4.19: Bode plot of the open outer control loop. The phase is biased, dash-dotted line indicates the real $-180°$.

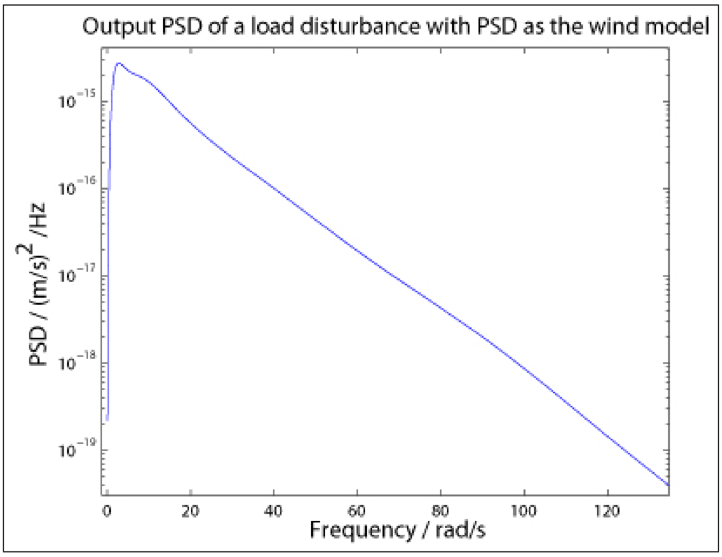

Figure 4.20: The resulting PSD of the wind when using the Davenport model from Section 3.4.

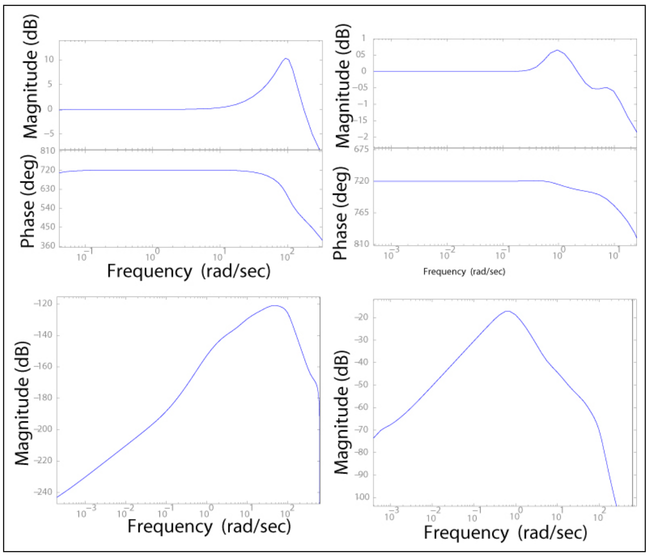

Figure 4.21: Closed loop behavior: **upper left)** $y_{ref} \rightarrow y_i$ inner loop,
upper right) $y_{ref} \rightarrow y$ outer loop,
lower left) $N \rightarrow y$ outer loop,
lower right) $L \rightarrow y$ outer loop.

5 Results

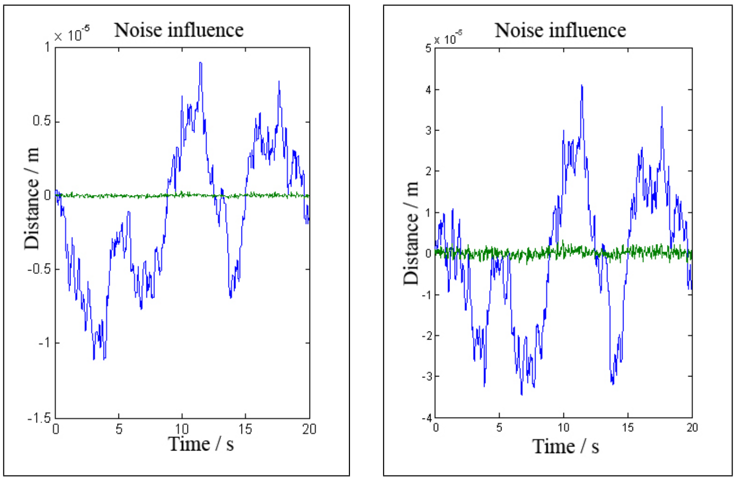

Figure 5.1: **left)** Simulations with the best available "off-the-shelf" gyro (curve with small excursions) and accelerometer (curve with large excursions).
right) Simulations with good enough "off-the-shelf" gyro (curve with small excursions) and accelerometer (curve with large excursions).
The controller used is the cascaded PI. Note the scaling!

Using noise data from existing IMU's has been an essential part of this thesis. The procedure was to take the best[23] "off-the-shelf" IMU hardware currently available from the leading manufacturers. Then, fine tuning of the different controllers were made. At steady state, the performance was measured with respect to their RMS values, calculated as

$$RMS = \sqrt{\sum_{i=1}^{N} \frac{(x_i - r_i)^2}{N}}$$

where $x_i - r_i$ is the difference between the current measurement x_i and the reference value r_i for that sample. This is in fact the equation for the estimation of the standard deviation. The RMS value can therefore be directly compared to the specification mentioned in Section 2. The next step was to increase the noise variance, and thereby the angular random walk as discussed

[23] With respect to noise specifications

in Section 2.2, until the specifications were just met. In Figure 5.1, a caparison between the best[24] ”off-the-shelf” IMU and a IMU just good enough[25] can be seen. The values for random walk are $1.00 \cdot 10^{-3} \frac{c^\circ}{\sqrt{h}}$ & $3.55 \cdot 10^{-3} \frac{m/s}{\sqrt{h}}$ for the best IMU and $10.8 \cdot 10^{-3} \frac{c^\circ}{\sqrt{h}}$ & $16.4 \cdot 10^{-3} \frac{m/s}{\sqrt{h}}$ for the one good enough[26].

As discussed in Section 2.1, the impact of the parameters other than the random walk are neglected due to their insignificance compared to the random walk. Therefore they are only included in the simulations as a term in the noise bias, which is set higher than it will be in reality. The controllers handle the bias very well and they are assumed to handle all the different disturbance sources coming from the IMU.

[24] Yielding $\sigma_{gyro} = 0.802 \cdot 10^{-7\,\circ}$ and $\sigma_{accelerometer} = 0.406 \cdot 10^{-5}\ m$.

[25] Yielding $\sigma_{gyro} = 8.71 \cdot 10^{-7\,\circ}$ and $\sigma_{accelerometer} = 1.828 \cdot 10^{-5}\ m$.

[26] Just below the specification.

6 Conclusion

While it is possible to obtain satisfactory results with both control strategies, there are some issues that need to be discussed. Both strategies have their pros and cons. While, in comparison, the cascaded PI-controller is much easier to understand and easy to tune to a desired bandwidth and damping, the tuning has to be done every time a single parameter changes its value. That means that as time goes by, and the force actuator design gets decided upon, the whole process of tuning the controller for every degree of freedom must be done all over again. As to the LQG-controller, all this is done in a single initializing program. Since all controller parameters are optimized automatically, it is easy to change the different processes, *e.g.,* the model for the cell or the design of the force actuators, and the new controller parameters will be updated accordingly.

The fact that all different controllers yield almost identical results is not strange. An LQG-controller is designed to compute the most probable value of the signal when it it degraded by known, or at least well estimated, disturbances. Since the noise used in this thesis is white, the LQG-controller cannot correct for it, as the expected value for integration of white noise still is zero. Thus, the LQG-controller has to trust the measurements and cannot improve the results.

The information from all IMU developers were that the noise from both gyros and accelerometers is band limited white. Probably, this is not entirely true, real white noise is very rare. The decision of the most favourable controller will have to wait until an IMU unit is bought, and proper noise identifiction has been made. Then, if the noise in fact is white, the cascaded PI/P-controller is a good choice because of its simplicity and intuitive setup. Else, the LQG-controller is probably the best choice. Independently of the outcome of the noise identification, the most important conclusion of this thesis is that it is possible to reach the desired specifications with the technology available today. In the coming years, advancement of the gyro/aceclerometer technologies will probably occur, making room for further improvements.

7 Further research

The LQG-controller is the favourable controller in the sense that it is already discretized and prepared for handling noise and load disturbances. If these should prove to be non-white, it should preferably be selected as the one used in the coming integration of the telescope software. Since time did not allow for any wind or noise identification, it is essential that such data are collected and analysed.

The simulations made are optimized for the specifications given when this thesis was made. These specifications, such as the rating of the force actuators, denoted `maxForce` in the Matlab files, may very well be changed to allow more force than the current $1000N$. If that is the case, the specifications for the model following and feed forward systems may need to be adjusted.

Also, when the decision of which type of force actuators that is to be used is made, that has to be taken into account.

References

[1] Torben Andersen et al. EURO 50, A 50 m Adaptive Optics Telescope, 2003. Lund Observatory, Lund, Sweden.

[2] Zhigang Feng and Harry Huang. An improved adaptive noise cancelling method, 1993. Shanghai Jiao Tong University, URL: http://ieeexplore.ieee.org/iel2/1086/7848/00332239.pdf.

[3] Roger A. Freedman and William J. Kaufmann III. Universe, sixth edition, 2002.

[4] Wilmar Hernández. Improving the response of an accelerometer by using optimal filtering, 2000. Department of electronic and instrumentation, Polytechnical University of Madrid, URL: http://www.sciencedirect.com/science?_ob=MImg&_imagekey=B6THG-42BSJ47-2-22&_cdi=5282&_user=745831&_orig=search&_coverDate=01%2F05%2F2001&_qd=1&_sk=999119996&view=c&wchp=dGLbVtb-zSkWz&md5=4aaed3b663cf2020821f910f41b7624b&ie=/sdarticle.pdf, (Accepted 25 October 2000).

[5] Tore Hägglund. Reglerteknik AK (in Swedish), Lecture notes, Department of Automatic Control, Lund University, Lund, 2002.

[6] Attilio Iannuzzi and Aoulo Spinelli. Artificial Wind Generation and Structural Response, 1987. Part of the Journal of Structural Engineering, Vol 113, No 12, Dec 1987, 2382-2398.

[7] Hervé C. Lefévre. Fundamentals of the Interferometric Fiber-Optic Gyroscope, 1996. Optical Review Vol. 4, 20-27.

[8] Georg Lindgren and Holger Rootzén. Stationära stokastiska processer (in Swedish), Department of Mathematical Statistics, Lund University, Lund, fifth edition, 2003.

[9] Dan Romanchik. Seven tips for making better accelerometer measurements, 2003. part of Test & Measurement World, sep 2003, A4-A5.

[10] Simiu Scanlan. Wind Effects on Structures: an introduction to wind engineering, 1978. John Wiley & sons, ISBN 0-471-02175-X.

[11] Dr. Walter Stockwell. Angle Random Walk, 2005. URL: http://www.xbow.com/Support/Support_pdf_files/AngleRandomWalkAppNote.pdf.

[12] Karl J. Åström and Björn Wittenmark. Computer Controlled systems, theory and design, 1997. Prentice Hall, third edition, ISBN 0-13-314899-8.

[13] Dr.-Ing. E. v. Hinueber. If you investigate in an inertial measurement system, 2005. URL: `http\protect\kern+.2222em\relax//www.imar-navigation.de/englishside/beisp_engl/decision_assistant.htm`.

[14] Patrick T. Wallace. Telescope Control Systems, 1995. SPIE-The International Society for Optical Engineering, ISBN 0-8194-1832-3.

[15] S. Woolven and D. B. Reid. IMU Noise Evaluation for Attitude Determination and Stabilization in Ground and Airborne Applications, 1994. URL: `http://ieeexplore.ieee.org/iel2/1083/7468/00303396.pdf`.

A Appendix A - Matrices

A.1 The Complete Transformation Matrix

$$T = \begin{bmatrix} a_{11} & a_{21} & a_{31} & a_{41} & a_{51} & a_{61} \\ a_{12} & a_{22} & a_{32} & a_{42} & a_{52} & a_{62} \\ a_{13} & a_{23} & a_{33} & a_{43} & a_{53} & a_{63} \\ a_{14} & a_{24} & a_{34} & a_{44} & a_{54} & a_{64} \\ a_{15} & a_{25} & a_{35} & a_{45} & a_{55} & a_{65} \\ a_{16} & a_{26} & a_{36} & a_{46} & a_{56} & a_{66} \end{bmatrix}$$

where

$$a_{11} = \frac{\cos^2(\phi)\cdot\cos(\nu)\cdot\cos(\theta)+\theta\cdot\phi\cdot\nu+\cos(\theta)\cdot\phi^2\cdot\cos(\nu))}{\cos^2(\theta)\cdot\cos^2(\nu)\cdot\cos^2(\phi)+\cos^2(\theta)\cdot\cos^2(\nu)\cdot\phi^2+\theta^2\cdot\phi^2\cdot\nu^2+\cos^2(\phi)\cdot\nu^2\cdot\cos^2(\theta)+\cos^2(\phi)\cdot\nu^2\cdot\theta^2+\theta^2\cdot\cos^2(\nu)\cdot\phi^2+\theta^2\cdot\cos^2(\nu)\cdot\cos^2(\phi)+\phi^2\cdot\cos^2(\theta)\cdot\nu^2}$$

$$a_{21} = \frac{\cos(\phi)\cdot\nu}{\phi^2\cdot\nu^2+\cos^2(\phi)\cdot\nu^2+\cos^2(\nu)\cdot\phi^2+\cos^2(\nu)\cdot\cos^2(\phi)}$$

$$a_{31} = \frac{\cos(\theta)\cdot\phi\cdot\nu-\theta\cdot\phi^2\cdot\cos(\nu)-\cos^2(\phi)\cdot\cos(\nu)\cdot\theta}{\cos^2(\theta)\cdot\cos^2(\nu)\cdot\cos^2(\phi)+\cos^2(\theta)\cdot\cos^2(\nu)\cdot\phi^2+\theta^2\cdot\phi^2\cdot\nu^2+\cos^2(\phi)\cdot\nu^2\cdot\cos^2(\theta)+\cos^2(\phi)\cdot\nu^2\cdot\theta^2+\theta^2\cdot\cos^2(\nu)\cdot\phi^2+\theta^2\cdot\cos^2(\nu)\cdot\cos^2(\phi)+\phi^2\cdot\cos^2(\theta)\cdot\nu^2}$$

$$a_{41} = \frac{-\phi\cdot\mathit{conj}(r_\theta\cdot\cos(\theta_0+\theta))\cdot\cos(\theta)\cdot\nu+\phi^2\cdot\mathit{conj}(r_\theta\cdot\cos(\theta_0+\theta))\cdot\theta\cdot\cos(\nu)+\cos^2(\phi)\cdot\cos(\nu)\cdot\theta\cdot\mathit{conj}(r_\theta\cdot\cos(\theta_0+\theta))+\cos^2(\phi)\cdot\cos(\nu)\cdot\cos(\theta)\cdot\mathit{conj}(r_\theta\cdot\sin(\theta_0+\theta))+\phi\cdot\mathit{conj}(r_\theta\cdot\sin(\theta_0+\theta))\cdot\theta\cdot\nu+\phi^2\cdot\mathit{conj}(r_\theta\cdot\sin(\theta_0+\theta))\cdot\cos(\theta)\cdot\cos(\nu)}{\cos^2(\theta)\cdot\cos^2(\nu)\cdot\cos^2(\phi)+\cos^2(\theta)\cdot\cos^2(\nu)\cdot\phi^2+\theta^2\cdot\phi^2\cdot\nu^2+\cos^2(\phi)\cdot\nu^2\cdot\cos^2(\theta)+\cos^2(\phi)\cdot\nu^2\cdot\theta^2+\theta^2\cdot\cos^2(\nu)\cdot\phi^2+\theta^2\cdot\cos^2(\nu)\cdot\cos^2(\phi)+\phi^2\cdot\cos^2(\theta)\cdot\nu^2}$$

$$a_{51} = \frac{\cos(\theta)\cdot\nu\cdot\phi\cdot\mathit{conj}(r_\phi\cdot\cos(\phi_0+\phi))+\cos^2(\theta)\cdot\nu\cdot\cos(\phi)\cdot\mathit{conj}(r_\phi\cdot\sin(\phi_0+\phi))-\theta\cdot\phi^2\cdot\cos(\nu)\cdot\mathit{conj}(r_\phi\cdot\cos(\phi_0+\phi))-\cos^2(\phi)\cdot\cos(\nu)\cdot\theta\cdot\mathit{conj}(r_\phi\cdot\cos(\phi_0+\phi))+\theta^2\cdot\cos(\phi)\cdot\mathit{conj}(r_\phi\cdot\sin(\phi_0+\phi))\cdot\nu}{\cos^2(\theta)\cdot\cos^2(\nu)\cdot\cos^2(\phi)+\cos^2(\theta)\cdot\cos^2(\nu)\cdot\phi^2+\theta^2\cdot\phi^2\cdot\nu^2+\cos^2(\phi)\cdot\nu^2\cdot\cos^2(\theta)+\cos^2(\phi)\cdot\nu^2\cdot\theta^2+\theta^2\cdot\cos^2(\nu)\cdot\phi^2+\theta^2\cdot\cos^2(\nu)\cdot\cos^2(\phi)+\phi^2\cdot\cos^2(\theta)\cdot\nu^2}$$

$$a_{61} = \frac{\cos^2(\theta)\cdot\cos(\phi)\cdot\mathit{conj}(r_\nu\cdot\cos(\nu_0+\nu))\cdot\nu+\cos^2(\phi)\cdot\cos(\nu)\cdot\cos(\theta)\cdot\mathit{conj}(r_\nu\cdot\sin(\nu_0+\nu))+\theta^2\cdot\nu\cdot\cos(\phi)\cdot\mathit{conj}(r_\nu\cdot\cos(\nu_0+\nu))+\theta\cdot\nu\cdot\phi\cdot\mathit{conj}(r_\nu\cdot\sin(\nu_0+\nu))+\cos(\theta)\cdot\phi^2\cdot\cos(\nu)\cdot\mathit{conj}(r_\nu\cdot\sin(\nu_0+\nu))}{\cos^2(\theta)\cdot\cos^2(\nu)\cdot\cos^2(\phi)+\cos^2(\theta)\cdot\cos^2(\nu)\cdot\phi^2+\theta^2\cdot\phi^2\cdot\nu^2+\cos^2(\phi)\cdot\nu^2\cdot\cos^2(\theta)+\cos^2(\phi)\cdot\nu^2\cdot\theta^2+\theta^2\cdot\cos^2(\nu)\cdot\phi^2+\theta^2\cdot\cos^2(\nu)\cdot\cos^2(\phi)+\phi^2\cdot\cos^2(\theta)\cdot\nu^2}$$

$$a_{12} = -\frac{\cos^2(\phi)\cdot\nu\cdot\cos(\theta)-\theta\cdot\phi\cdot\cos(\nu)+\phi^2\cdot\cos(\theta)\cdot\nu}{\cos^2(\theta)\cdot\cos^2(\nu)\cdot\cos^2(\phi)+\cos^2(\theta)\cdot\cos^2(\nu)\cdot\phi^2+\theta^2\cdot\phi^2\cdot\nu^2+\cos^2(\phi)\cdot\nu^2\cdot\cos^2(\theta)+\cos^2(\phi)\cdot\nu^2\cdot\theta^2+\theta^2\cdot\cos^2(\nu)\cdot\phi^2+\theta^2\cdot\cos^2(\nu)\cdot\cos^2(\phi)+\phi^2\cdot\cos^2(\theta)\cdot\nu^2}$$

$$a_{22} = \frac{\cos(\phi)\cdot\cos(\nu)}{\phi^2\cdot\nu^2+\cos^2(\phi)\cdot\nu^2+\cos^2(\nu)\cdot\phi^2+\cos^2(\nu)\cdot\cos^2(\phi)}$$

$$a_{32} = \frac{\cos(\theta)\cdot\phi\cdot\cos(\nu)+\theta\cdot\phi^2\cdot\nu+\cos^2(\phi)\cdot\nu\cdot\theta}{\cos^2(\theta)\cdot\cos^2(\nu)\cdot\cos^2(\phi)+\cos^2(\theta)\cdot\cos^2(\nu)\cdot\phi^2+\theta^2\cdot\phi^2\cdot\nu^2+\cos^2(\phi)\cdot\nu^2\cdot\cos^2(\theta)+\cos^2(\phi)\cdot\nu^2\cdot\theta^2+\theta^2\cdot\cos^2(\nu)\cdot\phi^2+\theta^2\cdot\cos^2(\nu)\cdot\cos^2(\phi)+\phi^2\cdot\cos^2(\theta)\cdot\nu^2}$$

$$a_{42} = -\frac{\phi\cdot\mathit{conj}(r_\theta\cdot\cos(\theta_0+\theta))\cdot\cos(\theta)\cdot\cos(\nu)+\phi^2\cdot\mathit{conj}(r_\theta\cdot\cos(\theta_0+\theta))\cdot\theta\cdot\nu+\cos^2(\phi)\cdot\nu\cdot\theta\cdot\mathit{conj}(r_\theta\cdot\cos(\theta_0+\theta))+\cos^2(\phi)\cdot\nu\cdot\cos(\theta)\cdot\mathit{conj}(r_\theta\cdot\sin(\theta_0+\theta))-\phi\cdot\mathit{conj}(r_\theta\cdot\sin(\theta_0+\theta))\cdot\theta\cdot\cos(\nu)+\phi^2\cdot\mathit{conj}(r_\theta\cdot\sin(\theta_0+\theta))\cdot\cos(\theta)\cdot\nu}{\cos^2(\theta)\cdot\cos^2(\nu)\cdot\cos^2(\phi)+\cos^2(\theta)\cdot\cos^2(\nu)\cdot\phi^2+\theta^2\cdot\phi^2\cdot\nu^2+\cos^2(\phi)\cdot\nu^2\cdot\cos^2(\theta)+\cos^2(\phi)\cdot\nu^2\cdot\theta^2+\theta^2\cdot\cos^2(\nu)\cdot\phi^2+\theta^2\cdot\cos^2(\nu)\cdot\cos^2(\phi)+\phi^2\cdot\cos^2(\theta)\cdot\nu^2}$$

$$a_{52} = \frac{\cos(\theta)\cdot\cos(\nu)\cdot\phi\cdot\mathit{conj}(r_\phi\cdot\cos(\phi_0+\phi))+\cos^2(\theta)\cdot\cos(\nu)\cdot\cos(\phi)\cdot\mathit{conj}(r_\phi\cdot\sin(\phi_0+\phi))+\theta\cdot\phi^2\cdot\nu\cdot\mathit{conj}(r_\phi\cdot\cos(\phi_0+\phi))+\cos^2(\phi)\cdot\nu\cdot\theta\cdot\mathit{conj}(r_\phi\cdot\cos(\phi_0+\phi))+\theta^2\cdot\cos(\phi)\cdot\mathit{conj}(r_\phi\cdot\sin(\phi_0+\phi))\cdot\cos(\nu)}{\cos^2(\theta)\cdot\cos^2(\nu)\cdot\cos^2(\phi)+\cos^2(\theta)\cdot\cos^2(\nu)\cdot\phi^2+\theta^2\cdot\phi^2\cdot\nu^2+\cos^2(\phi)\cdot\nu^2\cdot\cos^2(\theta)+\cos^2(\phi)\cdot\nu^2\cdot\theta^2+\theta^2\cdot\cos^2(\nu)\cdot\phi^2+\theta^2\cdot\cos^2(\nu)\cdot\cos^2(\phi)+\phi^2\cdot\cos^2(\theta)\cdot\nu^2}$$

$$a_{62} = -\frac{-\cos^2(\theta)\cdot\cos(\phi)\cdot\mathit{conj}(r_\nu\cdot\cos(\nu_0+\nu))\cdot\cos(\nu)+\cos^2(\phi)\cdot\nu\cdot\cos(\theta)\cdot\mathit{conj}(r_\nu\cdot\sin(\nu_0+\nu))-\theta^2\cdot\cos(\nu)\cdot\cos(\phi)\cdot\mathit{conj}(r_\nu\cdot\cos(\nu_0+\nu))-\theta\cdot\cos(\nu)\cdot\phi\cdot\mathit{conj}(r_\nu\cdot\sin(\nu_0+\nu))+\phi^2\cdot\cos(\theta)\cdot\nu\cdot\mathit{conj}(r_\nu\cdot\sin(\nu_0+\nu))}{\cos^2(\theta)\cdot\cos^2(\nu)\cdot\cos^2(\phi)+\cos^2(\theta)\cdot\cos^2(\nu)\cdot\phi^2+\theta^2\cdot\phi^2\cdot\nu^2+\cos^2(\phi)\cdot\nu^2\cdot\cos^2(\theta)+\cos^2(\phi)\cdot\nu^2\cdot\theta^2+\theta^2\cdot\cos^2(\nu)\cdot\phi^2+\theta^2\cdot\cos^2(\nu)\cdot\cos^2(\phi)+\phi^2\cdot\cos^2(\theta)\cdot\nu^2}$$

$$a_{13} = \frac{\theta\cdot\cos(\phi)}{\theta^2\cdot\phi^2+\cos^2(\phi)\cdot\cos^2(\theta)+\cos^2(\phi)\cdot\theta^2+\phi^2\cdot\cos^2(\theta)}$$

$$a_{23} = -\frac{\phi}{\phi^2+\cos^2(\phi)}$$

$$a_{33} = \frac{\cos(\theta)\cdot\cos(\phi)}{\theta^2\cdot\phi^2+\cos^2(\phi)\cdot\cos^2(\theta)+\cos^2(\phi)\cdot\theta^2+\phi^2\cdot\cos^2(\theta)}$$

$$a_{43} = \frac{-\mathit{conj}(r_\theta\cdot\cos(\theta_0+\theta))\cdot\cos(\theta)+\mathit{conj}(r_\theta\cdot\sin(\theta_0+\theta))\cdot\theta)\cdot\cos(\phi}{\theta^2\cdot\phi^2+\cos^2(\phi)\cdot\cos^2(\theta)+\cos^2(\phi)\cdot\theta^2+\phi^2\cdot\cos^2(\theta)}$$

$$a_{53} = -\frac{\theta^2\cdot\phi\cdot\mathit{conj}(r_\phi\cdot\sin(\phi_0+\phi))-\cos(\phi)\cdot\mathit{conj}(r_\phi\cdot\cos(\phi_0+\phi))\cdot\cos(\theta)+\mathit{conj}(r_\phi\cdot\sin(\phi_0+\phi))\cdot\phi\cdot\cos^2(\theta)}{\theta^2\cdot\phi^2+\cos^2(\phi)\cdot\cos^2(\theta)+\cos^2(\phi)\cdot\theta^2+\phi^2\cdot\cos^2(\theta)}$$

$$a_{63} = -\frac{\mathit{conj}(r_\nu\cdot\cos(\nu_0+\nu))\cdot\theta^2\cdot\phi-\cos(\phi)\cdot\mathit{conj}(r_\nu\cdot\sin(\nu_0+\nu))\cdot\theta+\phi\cdot\cos^2(\theta)\cdot\mathit{conj}(r_\nu\cdot\cos(\nu_0+\nu))}{\theta^2\cdot\phi^2+\cos^2(\phi)\cdot\cos^2(\theta)+\cos^2(\phi)\cdot\theta^2+\phi^2\cdot\cos^2(\theta)}$$

$$a_{14} = 0$$

$$a_{24} = 0$$

$$a_{34} = 0$$

$$a_{44} = 1$$

$$a_{54} = 0$$

$$a_{64} = 0$$

$$a_{15} = 0$$

$$a_{25} = 0$$

$$a_{35} = 0$$

$$a_{45} = 0$$

$$a_{55} = 1$$

$$a_{65} = 0$$

$$a_{16} = 0$$

$$a_{26} = 0$$

$$a_{36} = 0$$

$$a_{46} = 0$$

$$a_{56} = 0$$

$$a_{66} = 1$$

A.2 The Complete State Space System

$$\dot{X} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -k_{spring_x} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -3k_{spring_z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -k_{spring_z} \cdot (2r^2 - p^2) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -k_{spring_z} \cdot 2q^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot X + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{k_{FA}}{m} \\ \frac{k_{FA}}{m} \\ \frac{k_{FA}}{m} \\ \frac{k_{FA}}{I_{yy}} \\ \frac{k_{FA}}{I_{xx}} \\ \frac{k_{FA}}{I_{zz}} \end{bmatrix} \cdot u$$

B Appendix B - Matlab Code

B.1 Initializing

```
% A m file to initialize the mirror cell system.
%
% Schematic picture of the mirror cell hexapod structure. 1,2 & 3 is the
% rods of the tripod where the force actuators start out from. The forces
% are exposed in A, B & C in the plane of gravity. Therefore, there are no
% moments in x and y directions. The force are exposed from number to
% letter, 1->Cand is numbered from 1-6 in the matrix A like
% 1A,2A,2B,3B,3C,1C. Theta is the rotation round the y-axis, phi round the
% x-axis and nu round the z-axis.
%
%       ------------
%      / C________3 \  view from above
%     /  /        \  \                . is center of gravity (cog)
%    /  /          \  \               a = dist.  x dir from cog to A (C)
%   / 1/     .      \B \   y ^        b = dist.  y dir from cog to A (C)
%   \  \            /  /     |        c = dist.  x dir from cog to B
%    \  \          /  /      |        alpha & beta are angles of departure
%     \  \________/  /       |
%      \_A________2_/        -------->
%                                    x

path(path,'E:\Olof\Projekt\Modell2 (bellows)');
path(path,'E:\Olof\Projekt\Modell2 (bellows)\Discrete');
path(path,'E:\Olof\Projekt\Modell2 (bellows)\PrimitiveSystem');

alpha = pi/6;   %angle of departure for the force actuators (rad)
beta = 0.8702;
global h maxForce A A_inv k_FA;

a = 0.830;      %distances
b = 1.450;
c = 1.700;

m = 699.957;    %mass of the mirror cell

k_FA = 10;      %The different constants
k_springX = 10*9.81;
```

```
k_springZ = 10*9.81;
k_air = 10; %The transformation constant for the pressure of the bellows;
springs = [k_springX k_springZ]';
h = 0.005;

% Initialising the sensor, distances and angles.
global theta0 phi0 nu0 r_theta r_phi r_nu x0 y0 z0;
x0 = 0.5;
y0 = 0.3;
z0 = 0.7;
theta0  = atan(z0/x0);
phi0  = -atan(z0/y0);
nu0  = -atan(y0/x0);
r_theta = sqrt(x0^2 + z0^2);
r_phi = sqrt(y0^2 + z0^2);
r_nu = sqrt(x0^2 + y0^2);


% The distances from cog to the bellows
p = 2.2;
q = 2;
r = p/2;

%The inertial momentum vector (from model)
I = [7.618e2 7.618e2 1.212e3];

% The distance from DM1 to focus
focalLength = 60;

% The maximum force that can be exerted by the force acuator
maxForce = 1000;

%Splitting of components of the forces and moment to karthesian coordinates
Fz   = -cos(beta);

F1Ax = sin(alpha)*sin(beta);
F1Ay = -cos(alpha)*sin(beta);
F1Cx = sin(alpha)*sin(beta);
F1Cy = cos(alpha)*sin(beta);
M1A_theta = -a*Fz;
M1C_theta = -a*Fz;
```

```
M1A_phi   = b*Fz;
M1C_phi   = -b*Fz;
M1A_ny    = -F1Ax*b + F1Ay*a;
M1C_ny    = F1Cx*b + F1Cy*a;

F2Ax = -cos(alpha - pi/6)*sin(beta);
F2Ay = -sin(alpha - pi/6)*sin(beta);
F2Bx = sin(alpha)*sin(beta);
F2By = cos(alpha)*sin(beta);
M2A_theta = -a*Fz;
M2B_theta = c*Fz;
M2A_phi   = b*Fz;
M2B_phi   = 0;
M2A_ny    = -F2Ax*b + F2Ay*a;
M2B_ny    = F2Bx*0 - F2By*c;

F3Bx = sin(alpha)*sin(beta);
F3By = -cos(alpha)*sin(beta);
F3Cx = -cos(alpha - pi/6)*sin(beta);
F3Cy = sin(alpha - pi/6)*sin(beta);
M3B_theta = c*Fz;
M3C_theta = -a*Fz;
M3B_phi   = 0;
M3C_phi   = -Fz*b;
M3B_ny    = F3Bx*0 - F3By*c;
M3C_ny    = F3Cx*b + F3Cy*a;



%The force matrix
A=[ F1Ax      F2Ax      F2Bx      F3Bx      F3Cx      F1Cx;
    F1Ay      F2Ay      F2By      F3By      F3Cy      F1Cy;
    Fz        Fz        Fz        Fz        Fz        Fz;
    M1A_phi   M2A_phi   M2B_phi   M3B_phi   M3C_phi   M1C_phi;
    M1A_theta M2A_theta M2B_theta M3B_theta M3C_theta M1C_theta;
    M1A_ny    M2A_ny    M2B_ny    M3B_ny    M3C_ny    M1C_ny];

%The inverse of A
A_inv = inv(A);
```

t’

B.2 Initialize Cascaded Control

```
%-----------------------------------------------------------------------
%-----------------Creating the Controllers-----------------------------
% Properties of the velocity PI-controller: x,y
Kvx = 50000/k_FA;  %50000;
Tivx = 50;    %50;
% Properties of the position PI-controller: x,y
Kpx = 80;     %80;
Tipx = .07;    %.07;
% Properties of the outer PI-controller: x,y
Kox = 4.5;      %6
Tiox = .5;     %.5

% Properties of the velocity PI-controller: z
Kvz = 50000/k_FA;
Tivz = 50;
% Properties of the position PI-controller: z
Kpz = 50;
Tipz = .05;
% Properties of the outer PI-controller: z
Koz = 5;
Tioz = 1;

% Properties of the velocity PI-controller: theta, phi
Kvtheta = 60000/k_FA;  %60000;
Tivtheta = 5e10;    %5;
% Properties of the position PI-controller: theta, phi
Kptheta = 60;     %60;
Tiptheta = .05;    %0.05;
% Properties of the outer PI-controller: theta, phi
Kotheta = 5;      %10
Tiotheta = 10;     %2

% Properties of the velocity PI-controller: nu
Kvnu = 60000/k_FA;
Tivnu = 5e10;
% Properties of the position PI-controller: nu
Kpnu = 60;
Tipnu = .05;
% Properties of the outer PI-controller: nu
```

```
Konu = 5;
Tionu = 10;

% Properties of the bellow PI-controller: x
Kbx = 30;
Tibx = 1;
% Properties of the bellow PI-controller: z
Kbz = 30;
Tibz = 1;

%======================THE BELLOW P-CONTROLLER===========================
%========================================================================
% The pressure process (y = k_air/s * u)
G_bellow = [k_air/s 0; 0 k_air/s];

% Properties of the bellow P-controller: x (bandwidth 0.5 Hz)
Kbx = 0.5*2*pi/k_air;
% Properties of the bellow P-controller: z (bandwidth 0.5 Hz)
Kbz = 0.5*2*pi/k_air;

%Compenstation link
comp = (s/10 + 1)/(s/28 + 1);
COMP = [comp 0 0 0 0 0;
       0 comp 0 0 0 0;
       0 0 comp 0 0 0;
       0 0 0 comp 0 0;
       0 0 0 0 comp 0;
       0 0 0 0 0 comp];

 PIv = [Kvx*(s*Tivx+1)/(s*Tivx) 0 0 0 0 0;
        0 Kvx*(s*Tivx+1)/(s*Tivx) 0 0 0 0;
        0 0 Kvz*(s*Tivz+1)/(s*Tivz) 0 0 0;
        0 0 0 Kvtheta*(s*Tivtheta+1)/(s*Tivtheta) 0 0;
        0 0 0 0 Kvtheta*(s*Tivtheta+1)/(s*Tivtheta) 0;
        0 0 0 0 0 Kvnu*(s*Tivnu+1)/(s*Tivnu)];
PIvss = ss(PIv);

Pv = [Kvx 0 0 0 0 0;
      0 Kvx 0 0 0 0;
      0 0 Kvz 0 0 0;
      0 0 0 Kvtheta 0 0;
      0 0 0 0 Kvtheta 0;
      0 0 0 0 0 Kvnu];

PIp = [Kpx*(s*Tipx+1)/(s*Tipx) 0 0 0 0 0;
       0 Kpx*(s*Tipx+1)/(s*Tipx) 0 0 0 0;
       0 0 Kpz*(s*Tipz+1)/(s*Tipz) 0 0 0;
       0 0 0 Kptheta*(s*Tiptheta+1)/(s*Tiptheta) 0 0;
       0 0 0 0 Kptheta*(s*Tiptheta+1)/(s*Tiptheta) 0;
       0 0 0 0 0 Kpnu*(s*Tipnu+1)/(s*Tipnu)];
PIpss = ss(PIp);

PIo = [Kox*(s*Tiox+1)/(s*Tiox) 0 0 0 0 0;
       0 Kox*(s*Tiox+1)/(s*Tiox) 0 0 0 0;
       0 0 Koz*(s*Tioz+1)/(s*Tioz) 0 0 0;
       0 0 0 Kotheta*(s*Tiotheta+1)/(s*Tiotheta) 0 0;
       0 0 0 0 Kotheta*(s*Tiotheta+1)/(s*Tiotheta) 0;
       0 0 0 0 0 Konu*(s*Tionu+1)/(s*Tionu)];
PIoss = minreal(ss(PIo));

PIb = [Kbx*(s*Tibx+1)/(s*Tibx) 0;
       0 Kbz*(s*Tibz+1)/(s*Tibz)];
PIbss = minreal(ss(PIb));

% Creating the model system (according to specs.)
w_m = 2*2*pi;
Z_m = 1;
numModelC = w_m^2;
denModelC = [1 2*w_m*Z_m w_m^2];
model = tf(numModelC, denModelC);
MODEL = [model 0 0 0 0 0;
        0 model 0 0 0 0;
        0 0 model 0 0 0;
        0 0 0 model 0 0;
        0 0 0 0 model 0;
        0 0 0 0 0 model];

% Low- & highpass filters of order 2 based on the butterworth filter
w_lowpass = .9*2*pi;
w_highpass  = .1*2*pi;
[num_low,den_low] = butter(2, w_lowpass, 's');
[num_high,den_high] = butter(2, w_highpass, 'high','s');
```

```
LPtf = tf(num_low,den_low);
HPtf = tf(num_high,den_high);

LP = [LPtf 0 0 0 0 0;
      0 LPtf 0 0 0 0;
      0 0 LPtf 0 0 0;
      0 0 0 LPtf 0 0;
      0 0 0 0 LPtf 0;
      0 0 0 0 0 LPtf];
LPss = ss(LP);

HP = [HPtf 0 0 0 0 0;
      0 HPtf 0 0 0 0;
      0 0 HPtf 0 0 0;
      0 0 0 HPtf 0 0;
      0 0 0 0 HPtf 0;
      0 0 0 0 0 HPtf];
HPss = ss(HP);

% Lowpass filter for the Air Bellow control
w_lowpassBellow = .5*2*pi;
[num_lowBellow,den_lowBellow] = butter(2, w_lowpassBellow, 's');
LPtfBellow = tf(num_lowBellow,den_lowBellow);
LPb = [LPtfBellow 0 0 0 0 0;
       0 LPtfBellow 0 0 0 0;
       0 0 LPtfBellow 0 0 0;
       0 0 0 LPtfBellow 0 0;
       0 0 0 0 LPtfBellow 0;
       0 0 0 0 0 LPtfBellow];
LPssb = ss(LPb);

% Integrator
Integrator = [1/s 0 0 0 0 0;
              0 1/s 0 0 0 0;
              0 0 1/s 0 0 0;
              0 0 0 1/s 0 0;
              0 0 0 0 1/s 0;
              0 0 0 0 0 1/s];
Integratorss = ss(Integrator);
```

B.3 Initialize Linear Quadratic Control

```
% This m-file takes the continuous state space model and discretizes
% it with the sampling time h, using zoh.
% Then, a discrete LQ-controller is derived.


%=====================INITIALISING=====================================
%=====================================================================
global k_FA;
w = 10*2*pi;
Z = .6;
m = 700;
s = tf('s');
z = tf('z');

%Creating the tilt/rotation for the telescope, a total of 70 deg, 3 deg/s
a_TiltRotation=[0 .5*ones(1,6/h) zeros(1,(69/3-12)/h) -.5*ones(1,6/h-1)
zeros(1,23/h+1)]'; %-3*ones(1,2/h-1) 0 ...
              %zeros(1,(69/3-2)/h) 3*ones(1,1/h-1) 0]';
time_TiltRotation = [0:h:138]';
tiltRotation = [time_TiltRotation [a_TiltRotation; zeros(1,46*2/h)']];

% Low- & highpass filters of order 2 based on the butterworth filter
w_lowpass = 0.9*2*pi;
w_highpass  = 0.1*2*pi;
[num_low,den_low] = butter(2, w_lowpass, 's');
[num_high,den_high] = butter(2, w_highpass, 'high', 's');

% Creating the needed LTI systems for SimuLink
Integrator = [1/s 0 0 0 0 0;
              0 1/s 0 0 0 0;
              0 0 1/s 0 0 0;
              0 0 0 1/s 0 0;
              0 0 0 0 1/s 0;
              0 0 0 0 0 1/s];
IntegratorD = c2d(Integrator,h);

DerivativeD = [(z-1)/(z*h) 0 0 0 0 0;
               0 (z-1)/(z*h) 0 0 0 0;
               0 0 (z-1)/(z*h) 0 0 0;
               0 0 0 (z-1)/(z*h) 0 0;
               0 0 0 0 (z-1)/(z*h) 0;
               0 0 0 0 0 (z-1)/(z*h)];

LPtf = tf(num_low,den_low);
HPtf = tf(num_high,den_high);

LP = [LPtf 0 0 0 0 0;
      0 LPtf 0 0 0 0;
      0 0 LPtf 0 0 0;
      0 0 0 LPtf 0 0;
      0 0 0 0 LPtf 0;
      0 0 0 0 0 LPtf];
LPD = c2d(LP,h);

HP = [HPtf 0 0 0 0 0;
      0 HPtf 0 0 0 0;
      0 0 HPtf 0 0 0;
      0 0 0 HPtf 0 0;
      0 0 0 0 HPtf 0;
      0 0 0 0 0 HPtf];
HPD = c2d(HP,h);

%======================THE SYSTEMS=========================================
%=========================================================================
process_x = k_FA / (s^2*m + k_springX);
process_y = k_FA / (m*s^2);
process_z = k_FA / (s^2*m + 3*k_springZ);
process_theta = k_FA / (s^2*I(2) + k_springZ*(2*r^2+p^2));
process_phi = k_FA / (s^2*I(1) + k_springZ*2*q^2);
process_nu = k_FA / (I(3)*s^2);

sys_x = ss([0 1; -k_springX/m 0], [0; k_FA/m], [1 0], 0);
sys_y = ss([0 1; 0 0], [0; k_FA/m], [1 0], 0);
sys_z = ss([0 1; -3*k_springZ/m 0], [0; k_FA/m], [1 0], 0);
sys_theta = ss([0 1; -k_springZ*(2*r^2+p^2)/I(2) 0], [0; k_FA/I(2)], [1 0], 0);
sys_phi = ss([0 1; -k_springZ*2*p^2/I(1) 0], [0; k_FA/I(1)], [1 0], 0);
sys_nu = ss([0 1; 0 0], [0; k_FA/I(3)], [1 0], 0);

% Creating the discrete system
% x
```

```
sysD_x = c2d(sys_x,h);
H_x = tf(sysD_x);
PHI_x = sysD_x.a;
GAMMA_x = sysD_x.b;
%y
sysD_y = c2d(sys_y,h);
H_y = tf(sysD_y);
PHI_y = sysD_y.a;
GAMMA_y = sysD_y.b;
%z
sysD_z = c2d(sys_z,h);
H_z = tf(sysD_z);
PHI_z = sysD_z.a;
GAMMA_z = sysD_z.b;
%theta
sysD_theta = c2d(sys_theta,h);
H_theta = tf(sysD_theta);
PHI_theta = sysD_theta.a;
GAMMA_theta = sysD_theta.b;
%phi
sysD_phi = c2d(sys_phi,h);
H_phi = tf(sysD_phi);
PHI_phi = sysD_phi.a;
GAMMA_phi = sysD_phi.b;
%nu
sysD_nu = c2d(sys_nu,h);
H_nu = tf(sysD_nu);
PHI_nu = sysD_nu.a;
GAMMA_nu = sysD_nu.b;

% Creating the model system (according to specs.)
w_m = 2*2*pi;
Z_m = 1;
numModelC = w_m^2;
denModelC = [1 2*w_m*Z_m w_m^2];
[Am, Bm, Cm, Dm] = tf2ss(numModelC, denModelC);
modelSysC = ss(Am, Bm, Cm, Dm);
modelSysD = c2d(modelSysC,h);
Hm = tf(modelSysD); %Pulse transfer function for the model
[numModelD,denModelD] = TFDATA(Hm,'v');
```

```
GAMMA = [GAMMA_x GAMMA_y GAMMA_z GAMMA_theta GAMMA_phi GAMMA_nu];

PHI = [PHI_x PHI_y PHI_z PHI_phi PHI_theta PHI_nu];

MODELD = [Hm 0 0 0 0 0;
          0 Hm 0 0 0 0;
          0 0 Hm 0 0 0;
          0 0 0 Hm 0 0;
          0 0 0 0 Hm 0;
          0 0 0 0 0 Hm];


%Creating the feed forward systems
% x
Hff_x = Hm / H_x;
[numFf_x,denFf_x] = TFDATA(Hff_x,'v');
[Aff_x, Bff_x, Cff_x, Dff_x] = tf2ss(numFf_x,denFf_x);
% y
Hff_y = Hm / H_y;
[numFf_y,denFf_y] = TFDATA(Hff_y,'v');
[Aff_y, Bff_y, Cff_y, Dff_y] = tf2ss(numFf_y,denFf_y);
% z
Hff_z = Hm / H_z;
[numFf_z,denFf_z] = TFDATA(Hff_z,'v');
[Aff_z, Bff_z, Cff_z, Dff_z] = tf2ss(numFf_z,denFf_z);
% phi
Hff_phi = Hm / H_phi;
[numFf_phi,denFf_phi] = TFDATA(Hff_phi,'v');
[Aff_phi, Bff_phi, Cff_phi, Dff_phi] = tf2ss(numFf_phi,denFf_phi);
% theta
Hff_theta = Hm / H_theta;
[numFf_theta,denFf_theta] = TFDATA(Hff_theta,'v');
[Aff_theta, Bff_theta, Cff_theta, Dff_theta] = tf2ss(numFf_theta,denFf_theta);
% nu
Hff_nu = Hm / H_nu;
[numFf_nu,denFf_nu] = TFDATA(Hff_nu,'v');
[Aff_nu, Bff_nu, Cff_nu, Dff_nu] = tf2ss(numFf_nu,denFf_nu);

FFD = [Hff_x 0 0 0 0 0;
       0 Hff_y 0 0 0 0;
       0 0 Hff_z 0 0 0;
       0 0 0 Hff_theta 0 0;
       0 0 0 0 Hff_phi 0;
       0 0 0 0 0 Hff_nu];



%=====================THE LQ DESIGN======================================
%========================================================================
% The weiting matrixes, continuous
%The inverse of the squared allowed deviations of the states
Q1c_x = [2e8 0; 0 10000];
%The inverse of the squared allowed deviations of the command signal
rho_x = k_FA^2*1e-4;
Q2c_x = [rho_x];
Q12c_x = [0 0]';
Q0_x = 1e9;

%The inverse of the squared allowed deviations of the states
Q1c_theta = [2e8 0; 0 1000];  %3.33e-7 4e5
%The inverse of the squared allowed deviations of the command signal
rho_theta = k_FA^2*1e-4;
Q2c_theta = [rho_theta];
Q12c_theta = [0 0]';
Q0_theta = 1e9;

% Discretizing the cont. Q
Q1_x = PHI_x' * Q1c_x * PHI_x * h; %[Integral(PHI'*Q1c*PHI)ds]kh->kh+h
%[Integral(PHI'*(Q1c*GAMMA + Q12c))ds]kh->kh+h
Q12_x = PHI_x' * (Q1c_x*GAMMA_x + Q12c_x) * h;
%[Integral(GAMMA'*Q1c*GAMMA + 2*GAMMA'*Q12c + Q2c))ds]kh->kh+h
Q2_x = (GAMMA_x' * Q1c_x * GAMMA_x + 2*GAMMA_x' * Q12c_x + Q2c_x) * h;
Q_x = [Q1_x Q12_x; Q12_x' Q2_x];

% Discretizing the cont. Q
%[Integral(PHI'*Q1c*PHI)ds]kh->kh+h
Q1_theta = PHI_theta' * Q1c_theta * PHI_theta * h;
%[Integral(PHI'*(Q1c*GAMMA + Q12c))ds]kh->kh+h
Q12_theta = PHI_theta' * (Q1c_theta*GAMMA_theta + Q12c_theta) * h;
%[Integral(GAMMA'*Q1c*GAMMA + 2*GAMMA'*Q12c + Q2c))ds]kh->kh+h
Q2_theta = (GAMMA_theta' * Q1c_theta * GAMMA_theta + 2*GAMMA_theta' ...
* Q12c_theta + Q2c_theta) * h;
```

```
Q_theta = [Q1_theta Q12_theta; Q12_theta' Q2_theta];

% The LQ-controller
% x
% Solves the algebraic Ricatti equation
[Lx,Sx,ex] = dlqr(PHI_x,GAMMA_x,Q1_x,Q2_x,Q12_x);
% y
[Ly,Sy,ey] = dlqr(PHI_y,GAMMA_y,Q1_x,Q2_x,Q12_x);
% z
[Lz,Sz,ez] = dlqr(PHI_z,GAMMA_z,Q1_x,Q2_x,Q12_x);
% theta
[Ltheta,Stheta,etheta] = dlqr(PHI_theta,GAMMA_theta,Q1_theta,...
Q2_theta,Q12_theta);
% phi
[Lphi,Sphi,ephi] = dlqr(PHI_phi,GAMMA_phi,Q1_theta,Q2_theta,Q12_theta);
% nu
[Lnu,Snu,enu] = dlqr(PHI_nu,GAMMA_nu,Q1_theta,Q2_theta,Q12_theta);

L = [Lx; Ly; Lz; Ltheta; Lphi; Lnu];

% The wind transfer function, according to windGenerator.m
pwr = 2;
[B_p1,A_p1] = butter(pwr/2,1.4e-1,'s');
[B_p2,A_p2] = butter(pwr/2,8e0,'s');
[B_z1,A_z1] = butter(1,1.4e-2,'high','s');
sysL = minreal(10.8*tf(conv(B_z1,conv(B_p1,B_p2)),...
conv(A_z1,conv(A_p1,A_p2))));
ssL   = ss(sysL);
sysLD = minreal(c2d(sysL,h));
ssLD  = minreal(c2d(ssL,h));
PHI_L = ssLD.a;
GAMMA_L = ssLD.b;
C_L   = ssL.c;
pwr = pwr+1;
[num_wind, den_wind] = tfdata(ssL,'v');


% The new extended model due to the wind states
%x
sys_xi = ss([sys_x.a sys_x.b*C_L; zeros(pwr,2) ssL.a], ...
[sys_x.b; zeros(pwr,1)], [sys_x.c zeros(1,pwr)],0);
```

```
sysD_xi = c2d(sys_xi, h);
PHI_xw = sysD_xi.a(1:2,3:2+pwr);
GAMMA_xi = sysD_xi.b(1:2,1);
L_wx = pinv(GAMMA_x)*PHI_xw;
PHIw_x = sysD_xi.a(3:2+pwr,3:2+pwr);
%y
sys_yi = ss([sys_y.a sys_y.b*C_L; zeros(pwr,2) ssL.a], ...
[sys_y.b; zeros(pwr,1)], [sys_y.c zeros(1,pwr)],0);
sysD_yi = c2d(sys_yi, h);
PHI_yw = sysD_yi.a(1:2,3:2+pwr);
GAMMA_yi = sysD_yi.b(1:2,1);
L_wy = pinv(GAMMA_y)*PHI_yw;
PHIw_y = sysD_yi.a(3:2+pwr,3:2+pwr);
%z
sys_zi = ss([sys_z.a sys_z.b*C_L; zeros(pwr,2) ssL.a], ...
[sys_z.b; zeros(pwr,1)], [sys_z.c zeros(1,pwr)],0);
sysD_zi = c2d(sys_zi, h);
PHI_zw = sysD_zi.a(1:2,3:2+pwr);
GAMMA_zi = sysD_zi.b(1:2,1);
L_wz = pinv(GAMMA_z)*PHI_zw;
PHIw_z = sysD_zi.a(3:2+pwr,3:2+pwr);

%theta
PHI_thetai = [PHI_theta GAMMA_theta; 0 0 1];
GAMMA_thetai = [GAMMA_theta; 0];
C_thetai = [sysD_theta.c 0];
L_wtheta = [1 zeros(1,pwr-1)];
PHI_thetaw = [sysD_theta.b zeros(2,pwr-1)];
PHIw_theta = [1 zeros(1,pwr-1); zeros(pwr-1,pwr)];
%phi
PHI_phii = [PHI_phi GAMMA_phi; 0 0 1];
GAMMA_phii = [GAMMA_phi; 0];
C_phii = [sysD_phi.c 0];
L_wphi = [1 zeros(1,pwr-1)];
PHI_phiw = [sysD_phi.b zeros(2,pwr-1)];
PHIw_phi = [1 zeros(1,pwr-1); zeros(pwr-1,pwr)];
%nu
PHI_nui = [PHI_nu GAMMA_nu; 0 0 1];
GAMMA_nui = [GAMMA_nu; 0];
C_nui = [sysD_nu.c 0];
L_wnu = [1 zeros(1,pwr-1)];
```

```
PHI_nuw = [sysD_nu.b zeros(2,pwr-1)];
PHIw_nu = [1 zeros(1,pwr-1); zeros(pwr-1,pwr)];

C = [sys_x.c sys_y.c sys_z.c sys_theta.c sys_phi.c sys_nu.c];
%PHI_Xw = [PHI_xw PHI_yw PHI_zw PHI_thetaw PHI_phiw PHI_nuw];

% The observer
Z_1 = 0.9; Z_2 = 0.85; Z_3 = 0.8;
w_o = 50;
% Create the polynomial to the eigenvalues
p1 = [1 2*Z_1*w_o w_o^2];     %continous pole placement
p2 = [1 2*Z_2*w_o w_o^2];
p3 = [1 2*Z_3*w_o w_o^2];
p4 = [1 w_o];

P = [exp(h*roots(p1)); exp(h*roots(p2)); exp(h*roots(p4))];
Pi = [exp(h*roots(p1)); exp(h*roots(p4))];
global Tt;
Tt = 10; %Tracking time constant for the anti-windup feature
K_xi = place(sysD_xi.a’,sysD_xi.c’,P)’;
K_yi = place(sysD_yi.a’,sysD_yi.c’,P)’;
K_zi = place(sysD_zi.a’,sysD_zi.c’,P)’;
K_thetai = [place(PHI_thetai’,C_thetai’,Pi)’; zeros(pwr-1,1)];
K_phii = [place(PHI_phii’,C_phii’,Pi)’; zeros(pwr-1,1)];
K_nui = [place(PHI_nui’,C_nui’,Pi)’; zeros(pwr-1,1)];
K = [K_xi(1:2); K_yi(1:2); K_zi(1:2); K_thetai(1:2);
K_phii(1:2); K_nui(1:2);];

%======================THE OUTER PI-CONTROLLER===========================
%========================================================================
% Properties of the outer PI-controller: x,y
Kox = 3;      %16
Tiox = .3;     %.4
% Properties of the outer PI-controller: z
Koz = 3;      %6
Tioz = .4;    %.4
% Properties of the outer PI-controller: theta, phi
Kotheta = 3;      %10
Tiotheta = .5;     %2
% Properties of the outer PI-controller: nu
Konu = 3;
```

```
Tionu = 1;

PIo = [Kox*(s*Tiox+1)/(s*Tiox) 0 0 0 0 0;
       0 Kox*(s*Tiox+1)/(s*Tiox) 0 0 0 0;
       0 0 Koz*(s*Tioz+1)/(s*Tioz) 0 0 0;
       0 0 0 Kotheta*(s*Tiotheta+1)/(s*Tiotheta) 0 0;
       0 0 0 0 Kotheta*(s*Tiotheta+1)/(s*Tiotheta) 0;
       0 0 0 0 0 Konu*(s*Tionu+1)/(s*Tionu)];
PIoD = c2d(PIo,h);

%Compenstation link
comp = (s/10 + 1)/(s/28 + 1);
COMP = [comp 0 0 0 0 0;
       0 comp 0 0 0 0;
       0 0 comp 0 0 0;
       0 0 0 comp 0 0;
       0 0 0 0 comp 0;
       0 0 0 0 0 comp];
COMPD = c2d(COMP,h);

%=====================THE BELLOW P-CONTROLLER===========================
%=======================================================================
% The pressure process (y = k_air/s * u)
G_bellow = k_air/s;
G_bellowD = [c2d(G_bellow,h) 0;
             0 c2d(G_bellow,h)];

% Properties of the bellow P-controller: x (bandwidth 0.5 Hz)
Kbx = 0.5*2*pi/k_air;
% Properties of the bellow P-controller: z (bandwidth 0.5 Hz)
Kbz = 0.5*2*pi/k_air;
```

B.4 Initialize Linear Quadratic Gaussian Control

```
% Creating the double/single integrators, a small distance from the origin.
% The command KALMAN cannot handle pure integrators.
H_xyz = ss(1/((s+1e-4)^2));
H_xyzD = c2d(H_xyz,h);
H_tpn = ss(1/(s+1e-6));
H_tpnD = c2d(H_tpn,h);
```

```
% The wind transfer function, according to windGenerator.m
pwr = 2;
[B_p1,A_p1] = butter(pwr/2,1.4e-1,'s');
[B_p2,A_p2] = butter(pwr/2,8e0,'s');
[B_z1,A_z1] = butter(1,1.4e-2,'high','s');
sysL = minreal(10.8*tf(conv(B_z1,conv(B_p1,B_p2)),...
conv(A_z1,conv(A_p1,A_p2))));
ssL   = ss(sysL);
sysLD = minreal(c2d(sysL,h));
ssLD  = minreal(c2d(ssL,h));
PHI_L = ssLD.a;
GAMMA_L = ssLD.b;
C_L   = ssL.c;
pwr = pwr+1;
[num_wind, den_wind] = tfdata(ssL,'v');


% Creating the extended system including the measurement noise and wind
% disturbance
LQGsys_xe = ss([sys_x.a sys_x.b*C_L zeros(2); zeros(pwr,2)...
 ssL.a zeros(pwr,2);zeros(2,pwr+2) H_xyz.a], [sys_x.b; zeros(pwr+2,1)],...
 [sys_x.c zeros(1,pwr) H_xyz.c],0);
LQGsysD_xe = c2d(LQGsys_xe,h);
LQGsys_ye = ss([sys_y.a sys_y.b*C_L zeros(2); zeros(pwr,2) ...
 ssL.a zeros(pwr,2);zeros(2,pwr+2) H_xyz.a], [sys_y.b; ...
 zeros(pwr+2,1)], [sys_y.c zeros(1,pwr) H_xyz.c],0);
LQGsysD_ye = c2d(LQGsys_ye,h);
LQGsys_ze = ss([sys_z.a sys_z.b*C_L zeros(2); zeros(pwr,2) ...
 ssL.a zeros(pwr,2); zeros(2,pwr+2) H_xyz.a], [sys_z.b; zeros(pwr+2,1)], ...
 [sys_z.c zeros(1,pwr) H_xyz.c],0);
LQGsysD_ze = c2d(LQGsys_ze,h);
sysD_thetae = c2d(ss([sys_theta.a zeros(2,1); zeros(1,2) H_tpn.a], ...
[sys_theta.b; 0], [sys_theta.c H_tpn.c], 0), h);
sysD_phie = c2d(ss([sys_phi.a zeros(2,1); zeros(1,2) H_tpn.a], ...
[sys_phi.b; 0], [sys_phi.c H_tpn.c], 0), h);
sysD_nue = c2d(ss([sys_nu.a zeros(2,1); zeros(1,2) H_tpn.a], ...
[sys_nu.b; 0], [sys_nu.c H_tpn.c], 0), h);

% The covariance matrices
R1_xyz = [1e6*[0.0001 0.00; 0.00 1] zeros(2,pwr+2); zeros(pwr,2) s...
sLD.b*1e2*ssLD.b' zeros(pwr,2); zeros(2,pwr+2) H_xyzD.b*7e-5*H_xyzD.b'];
```

```
R1_tpn = [1e2*eye(2) zeros(2,1); 0 0 H_tpn.b*pi^2/180^2*5e-8*H_tpn.b'];
R2 = 1e-8;

% The cost matrices
Q1c_x = [1e10 0; 0 1e2];
Q1c = diag(1e15*ones(1,pwr+4));
Q1c(1:2,1:2) = Q1c_x; Q1c(end-1,end-1) = 1e13; Q1c(end,end) = 1e13;
Q1_x = LQGsysD_xe.a' * Q1c * LQGsysD_xe.a * h;
Q1_y = LQGsysD_ye.a' * Q1c * LQGsysD_ye.a * h;
Q1_z = LQGsysD_ze.a' * Q1c * LQGsysD_ze.a * h;
Q2c_x = 1e-2;
Q2_x = (GAMMA_x' * Q1c_x * GAMMA_x + 2*GAMMA_x' * Q12c_x + Q2c_x) * h;
Q1c_thetaLQG = [Q1c_theta*300 zeros(2,1); 0 0 1e8];
Q1_theta = sysD_thetae.a' * Q1c_thetaLQG * sysD_thetae.a *h;
Q1_phi = sysD_phie.a' * Q1c_thetaLQG * sysD_phie.a *h;
Q1_nu = sysD_nue.a' * Q1c_thetaLQG * sysD_nue.a *h;


% Creation of the controllers, with the aid of designlqg.m, from the
% department of automatic control, Lund Institute of Technology
[ctrl_x, L_x, obs_x, K_x, Kf] = designlqg(LQGsysD_xe, Q1_x, Q2_x, R1_xyz, R2);
[ctrl_y, L_y, obs_y, K_y, Kf] = designlqg(LQGsysD_ye, Q1_y, Q2_x, R1_xyz, R2);
[ctrl_z, L_z, obs_z, K_z, Kf] = designlqg(LQGsysD_ze, Q1_z, Q2_x, R1_xyz, R2);
[ctrl_theta, L_theta, obs_theta, K_theta, Kf] = ...
designlqg(sysD_thetae, Q1_theta, Q2_theta, R1_tpn, R2);
[ctrl_phi, L_phi, obs_phi, K_phi, Kf] = ...
designlqg(sysD_phie, Q1_phi, Q2_theta, R1_tpn, R2);
[ctrl_nu, L_nu, obs_nu, K_nu, Kf] = ...
designlqg(sysD_nue, Q1_nu, Q2_theta, R1_tpn, R2);
```