Department of Precision and Microsystems Engineering

Controller Design for a Freeform Optical Surface Tracking System with a Non-linear Spring

Rolf Bavelaar

Report no	:	2024.043
Coach	:	Tiis van der Werf
Professor	:	Hassan HosseinNia
Specialisation	:	Mechatronic System Design
Type of report	:	Master Thesis
Date	:	03 July 2024

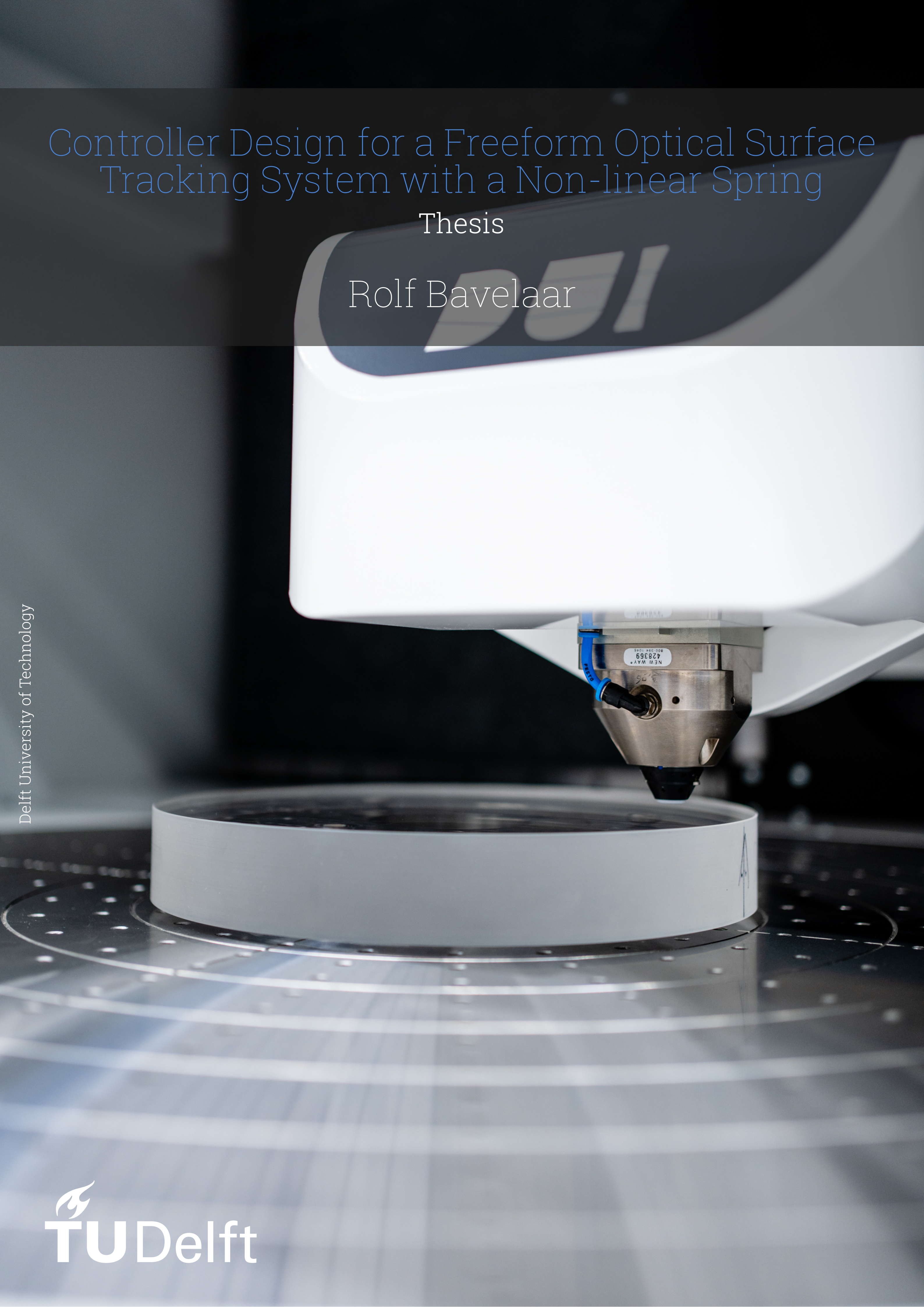
Delft University of Technology

Challenge the future

Controller Design for a Freeform Optical Surface Tracking System with a Non-linear Spring

Thesis

Rolf Bavelaar



Delft University of Technology

TU Delft

Controller Design for a Freeform Optical Surface Tracking System with a Non-linear Spring

Thesis

by

Rolf Bavelaar

Supervisors:	Ir. Tiis van der Werf (Dutch United Instruments)
	Dr. Hassan HosseinNia (TU Delft)
Project Duration:	September, 2023 - July, 2024
Faculty:	Faculty of Mechanical, Maritime and Materials Engineering (3ME)

TU Delft

Preface

This work is submitted in partial fulfillment of the requirements for the degree of Master of Science at the department of Precision and Microsystems Engineering of the Delft University of Technology. The objective of this thesis is to enhance the performance of a freeform optical surface tracking system through an improved control strategy.

The thesis is structured in two main parts. First the main results of the thesis are presented in a paper format. This part, presents the implemented controller modifications, provides a justification for their use and quantifies their impact. Following this research paper an extensive appendix is provided. The appendix presents all the theoretical background and tools used to design and evaluate the implemented controller modifications. The appendix will be referenced throughout the paper in case additional information is of interest to the reader.

Rolf Bavelaar
Delft, July 2024

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
BLS	Base Linear System
CF	Confocal
CI	Clegg Integrator
CGLP	Constant in Gain Lead in Phase
CPS	Cumulative Power Spectrum
DUI	Dutch United Instruments
DF	Describing Function
FORE	First Order Reset Element
GM	Gain Margin
GFORE	Generalized First Order Reset Element
HOSIDF	Higher-Order Sinusoidal-Input Describing Functions
LTI	Linear Time-Invariant
IFM	Interferometer
NMF	NANOMEFOS
NLC	Non-linearity compensator
NSV	Nyquist Stability Vector
MM	Modulus Margin
PM	Phase Margin
PID	Proportional Integral Derivative
PSD	Power Spectral Density
SFC	Spring Force Compensation
SORE	Second Order Reset Element
ZOH	Zero-Order Hold

Controller Design for a Freeform Optical Surface Tracking System with a Non-linear Spring

Rolf Bavelaar [1]

Abstract: This paper presents a controller design approach aimed at optimizing the tracking capability of a freeform optical surface tracking system. It is shown that the second-order disturbance caused by the non-linearity in the system's spring and the changes in surface height is by far the biggest contributor to the tracking error. To reduce this error the paper proposes a position dependent spring force compensation, aiming to use the measurable non-linearity in the spring to effectively linearize the system. Thereafter, it is shown that the tracking error is mainly caused by the change in surface height and an unknown disturbance acting around the bandwidth. Since the change in surface height is known, a feedforward control structure is used to suppress the effect of this disturbance. Following this modification, the unknown disturbance becomes dominant. To suppress the effect of this disturbance a so called 'Constant in Gain Lead in Phase' controller is added that decreases the magnitude of the sensitivity function in the active frequency range of the disturbance, without increasing the sensitivity's magnitude at other frequencies. The proposed controller modifications are all experimentally validated on a freeform optical surface measurement machine through the evaluation of the tracking error during a tilted flat measurement.

Keywords: Non-linear Control, Reset Control, CgLp, Feedforward Control, Mass-Spring-Damper

1. Introduction

As the high-tech sector continues to push its boundaries, the demand for freeform optics is ever increasing. Lasers, augmented reality, advanced imaging and many other technologies, all require optics with complex forms fabricated with a minimal shape uncertainty. It goes without saying, that the more accurate a part can be measured, the more accurate it can be fabricated. An increased demand for high-end optics, therefore goes hand in hand with an increased demand for high-end measurement machines. Dutch United Instruments (DUI) produces such measurement machines. Simply put, their machine operates by moving a distance sensor over an optical component, thereby generating a height map of the component's surface. The highest measurement accuracy is achieved when the distance between the surface and the sensor remains fixed. Consequently, to ensure accurate measurements, the sensor has to move vertically when the surface height changes. In section 2 it will be shown that this vertical movement is generated by a tracking system that can be described as mass-spring-damper with a non-linear spring.

The control problem for this system is essentially a standard position tracking problem. Typically, these types of control problems are solved through linear feedback and/or feedforward control [1]. A conventional approach is to first use feedforward control to suppress the effect of known disturbances as much as possible and thereafter use a linear feedback control strategy, usually PID control, to increase the tracking capability of the system further [2].

When using a linear control structure, Bode's phase-gain relation and the waterbed effect put inherent limitations on the performance of the control system [3], [4], *see appendix C.8 for a more detailed explanation.* To overcome these limitations and increase the tracking capability of a system further, a non-linear control strategy will have to be implemented. In this paper reset control is chosen to further increase the tracking capability of the system [3]. The benefit of using reset control compared to other non-linear control techniques such as sliding mode, fuzzy logic or model predictive control is that, just as linear controllers, reset controllers can be designed in the loop-shaping framework [5]. Reset control in general works very similar to linear control, many reset control elements have a linear counterpart [6]. The difference with linear control is the fact that a subset of the controller states is reset when a pre-determined condition is met [7]. This resetting action causes the controller to be non-linear and capable of overcoming the limits of linear control [6].

[1] Department of Precision and Microsystems Engineering, Delft University of Technology, Delft, The Netherlands

This paper presents a systematic approach to improve the tracking capability of a freeform optical surface tracking system. The approach will largely follow the conventional sequence of feedforward, followed by linear feedback and then non-linear feedback control. It will be shown that in this case however, the non-linearity in the spring is of such magnitude that an initial linearization step is required before this 'conventional approach' can be effectively implemented. Furthermore, the used setup allows for a dual-sided measurement of the mass. In this case both the extension of the spring and the distance between the mass and the optical surface is known. This dual-sided setup will be exploited to boost the tracking capability of the system further. The remainder of this paper is structured as follows: First a detailed description of the system is given, explaining how the system works and how it is currently controlled, section 2. Thereafter, the results of a performance analysis are presented, quantifying the initial performance and generating insight into the root causes of tracking errors, section 3. Next, the implemented controller modifications will be presented and evaluated, section 4. The paper concludes with a brief summary of the implemented modifications and a recommendation for future work, section 5.

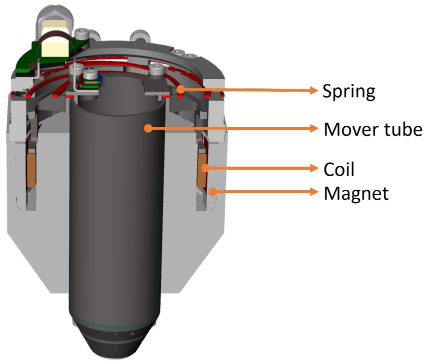
2. System Description

The used tracking system is presented in this section. First the mechanical workings of the system are outlined, see section 2.1. Thereafter, a detailed description of the initial control system is given, section 2.2.

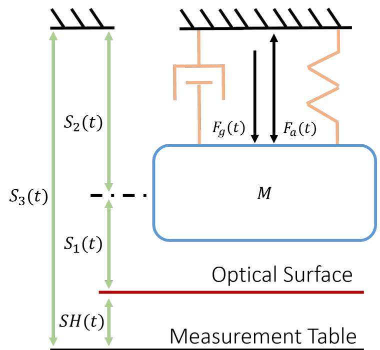
2.1. Mechanical System Description

A section viewed drawing of the system is presented in figure 1a. The system consists of a so-called "Mover tube" that houses the moving distance sensor described in the introduction. This mover tube is in turn connected to a coil placed in a magnetic field, effectively creating a Lorentz actuator that can be used to control the position of the tube. Simplifying the mover tube assembly to a single mass, the system can be represented by the multi-body diagram presented in figure 1b. The measurement of the moving sensor is denoted by S_1. This sensor measures the distance between the moving mass and the optical surface. To mitigate the effects of thermal expansion and other unwanted disturbances, the position of the assembly is carefully monitored by a measurement loop. The final surface height (SH) measurement is the solution of equation 1. The S_1 measurement is obtained using a confocal sensor, this sensor has a 2σ uncertainty of $3.7\ [nm]$ in case of perfect tracking. Here it should be noted that, due to its internal mechanics, the confocal sensor's accuracy decreases when its measurement deviates from zero, hence the need for a tracking system, *see appendix B for a more detailed explanation*. The other measurements in the loop are done using a series of interferometers, capable of achieving a 2σ uncertainty of $2\ [nm]$ [8].

$$SH(t) = S_3(t) - S_1(t) - S_2(t) \tag{1}$$



(a) Annotated section viewed drawing of the surface tracking system.



(b) Schematic representation of the surface tracking system. F_g and F_a represent the gravitational force and actuator force respectively.

Figure 1: Used optical surface tracking system.

2.2. Control System Description

As explained in the previous section, section 2.1, the goal of the control system is to ensure the S_1 measurement remains constant. The initial control system used to achieve this task is presented in figure 2. In the diagram, G represents the plant, which denotes the ratio between the input current and the output position. The constant K denotes the spring constant and the parameters D_i signify the disturbances acting on the system. Del_1 and Del_2 represent the sensor delay of S_1 and S_2 respectively. Following the block diagram presented in figure 2, the S_1 output can be calculated using equation 2. The control system essentially calculates an output force based of the tracking error, which is equivalent to the S_1 measurement. Initially this calculation was done using a standard PID control algorithm, with an extra addition, the spring force compensation (SFC) term, to compensate for the spring force. The SFC term is calculated based on the S_2 measurement and intended to mimic the spring force acting on the system as closely as possible. When the SFC term equals the spring force, the spring compensated plant, G_c, equals a mass-damper system. A mass-damper system has a much higher gain at low-frequencies compared to a mass-spring-damper system. This gain increase will cause the sensitivity function, defined as $Sens(z) = \frac{1}{1+PID(z)G_c(z)Del_1(z)}$, to be smaller. Since the tracking error is proportional to the sensitivity function, see equation 2, this error will reduce as well.

$$S_1(z) = \frac{G_c(z)Del_1(z)D_1(z) + Del_1(z)D_2(z) + D_3(z)}{1 + PID(z)G_c(z)Del_1(z)} \quad (2)$$

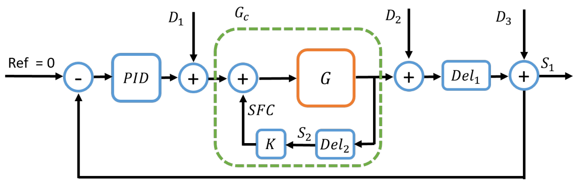


Figure 2: Block diagram representation of the initial control system.

3. Performance Analysis

The goal of this research is to increase the tracking capability a mass-spring-damper system measuring an optical surface. To benchmark this tracking capability, the tracking error during the measurement of a tilted flat surface will be evaluated, see figure 3. When measuring a tilted flat, the system goes through its full range of motion and encounters a tilted surface. These properties make a tilted flat measurement a good indicator of the system's performance on an arbitrary surface.

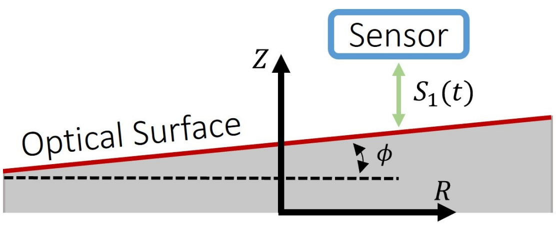
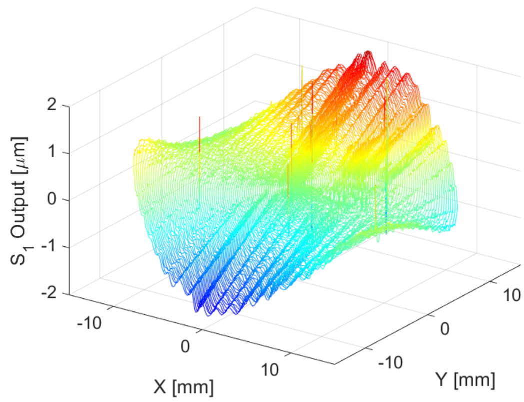
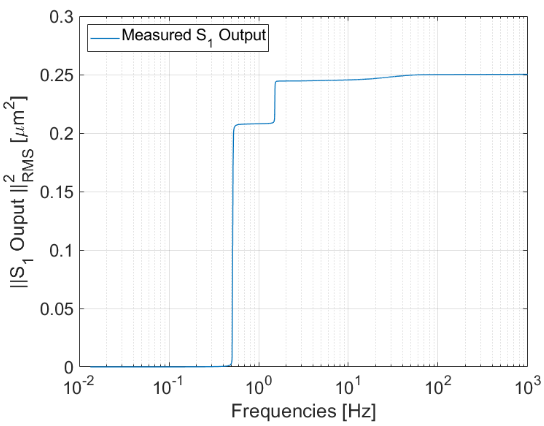


Figure 3: Flat surface tilted ϕ degrees.

During a measurement the optical surface is rotated around the Z-axis while the assembly moves in the negative R-direction at a constant speed, see figure 3. These two movements cause the system to move over the surface in a spiral shape. Figure 4a shows the output of S_1, during a $7°$ tilted flat measurement. The figure shows the spiral trace the sensor makes when scanning the surface. X and Y represent the position of the sensor on the surface and the Z-axis represents the S_1 measurement. To get an indication of the frequency content of the sensor output, the cumulative power spectrum (CPS) is used. Figure 4b shows the corresponding CPS of the S_1 output. *See appendix C.9 for a more in depth explanation about the CPS.*



(a) Initial S_1 output during a $7°$ tilted flat measurement.



(b) Initial CPS of the S_1 output during a $7°$ tilted flat measurement.

Figure 4: Visualization of the initial tracking error during a $7°$ tilted flat measurement.

When the system tracks the surface perfectly, the sensor outputs a value of zero. In figure 4a it can clearly by seen that is not the case. These deviations from zero are the result of tracking errors. In this section, the components causing these tracking errors will be outlined. The goal of this section is to is to generate insights into the effects of these components on the tracking capability of the system. These insights will be used in section 4 to justify the changes and additions to the control system. The modelled sensor outputs shown in this section were generated using a Simulink model structured following the block diagram presented in figure 2. *Further information regarding the simulation can be found in appendix F.*

3.1. Changes in Surface Height

The plant essentially represents the ratio between the input current and the displacement of the spring measured by S_1. When the surface height changes, the position measured by the sensor changes while the current remains the same. In a transfer function this can be described using equation 3. In the equation $SC(t)$ represents the change in surface height over time. K_m, m, b and K represent the motor constant, mass, damping and spring constant respectively. Following the block diagram presented in figure 2, the surface change can be regarded as part of disturbance D_2 at the plant output.

$$G(s) = \frac{K_m}{ms^2 + bs + K} + \mathcal{L}\left\{SC(t)\right\} \tag{3}$$

The deviations in the surface height of the used flats are known to be below $2\ [nm]$ root mean square (RMS). During a tilted flat measurement the system goes through its full range of motion, $\pm 2.5\ [mm]$. Changes in the surface height caused by the tilt of the flat are thus roughly a million times larger than the changes in height caused by irregularities in the surface. Since the irregularities are so small it is assumed the flat is perfectly flat when modelling the surface change disturbance. Given this assumption, the height of the tilted flat in the z-direction follows equation 4, *see appendix A for the derivation*. In the equation θ denotes the amount the flat is rotated around the Z-axis, ϕ indicates the tilt angle of the flat and R represents the distance between the measuring point and the origin, in the xy-plane.

$$z = -\sqrt{\frac{R^2}{sin(\theta)^2 + (cos(\theta)cos(\phi))^2}}\, sin(\phi)cos(\theta) \tag{4}$$

Relevant Frequency Spectrum

R and θ, both change over time. The tilt angle ,ϕ, remains constant. When the tilt angle is close to zero, $cos(\phi)$ is close to 1. This means that the denominator of the square root in equation 4 can be rewritten as shown below.

$$\begin{aligned} sin(\theta)^2 + (cos(\theta)cos(\phi))^2 &\approx sin(\theta)^2 + cos(\theta)^2 \\ sin(\theta)^2 + cos(\theta)^2 &= 1 \end{aligned} \tag{5}$$

This enables the disturbance to be expressed as:

$$z = -Rsin(\phi)cos(\theta) \tag{6}$$

The smaller the tilt angle, the more the surface change resembles a pure cosine wave oscillating at the rotational frequency of the machine. When the surface change is a pure cosine wave, the CPS plot appears as a single step at the rotational frequency. In figure 5, the normalized CPS of the surface change disturbance model is presented for various angles at a rotational frequency of $0.5\ Hz$. The normalization is done by dividing each element in the CPS vector by the vector's maximum value. From the figure it can be concluded that as the tilt angle decreases, the correlations with frequencies other then the rotational frequency, becomes lower as well. The measured optics typically have a tilt angle, anywhere between $0°$ and $10°$. In figure 5 it can be seen that for a $10°$ tilt angle, frequencies other then the rotational frequency are negligible. Since $10°$ is the largest angle used, it will be assumed that the surface change can be approximated as a pure cosine wave oscillating at the rotational frequency of the machine.

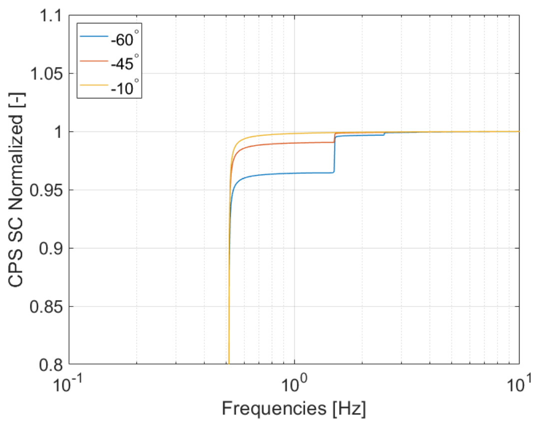


Figure 5: Normalized CPS of the surface change disturbance, for various tilt angles.

Disturbance Evaluation

The previous section showed the surface change can be approximated as a pure cosine wave at the plant output. Following the block diagram presented in figure 2, the contribution of a D_2 disturbance to the S_1 output can be written as:

$$S_1(z) = \frac{Del_1(z)}{1 + PID(z)G_c(z)Del_1(z)} D_2(z) \tag{7}$$

From equation 7 it can be concluded that the errors through the change in surface height are proportional to the sensitivity function multiplied by $Del_1(z)$. In other words, when a cosine wave enters the system a scaled and phase shifted cosine wave is added to the sensor output. The frequency of the wave remains the same. This causes the surface change disturbance to appear as a jump at the rotational frequency in a CPS plot.

To get an indication of the magnitude of this jump, the surface change disturbance is applied in the Simulink model at D_2. The disturbances D_1 and D_3 are set to zero. Using this set-up, the modelled S_1 output presented in figure 6 is obtained. When evaluating the modelled output, as expected, a clear jump can be seen at 0.5 $[Hz]$. When comparing the modelled and measured sensor outputs however, a significant difference between the magnitude of the jumps is present. Generally speaking, this difference in magnitude can be caused by one or a combination of the factors outlined below:

1. **Inaccuracies in the modelled plant gain**: The contribution of D_2 to the sensor output is proportional to the magnitude of the sensitivity function. When the magnitude of the modelled plant differs from the magnitude of the actual plant, the magnitude of the sensitivity function will be different. This difference will in turn cause the magnitude of the jump at the rotational frequency to be different as well.
2. **Presence of other 0.5 [Hz] disturbances**: The jump in the CPS plot reflects the power of the 0.5 $[Hz]$ frequency in the output. If other disturbances act on the system at a 0.5 $[Hz]$ frequency, then these disturbances will contribute to the jump in the CPS as well.
3. **Inaccuracies in the modelled surface height**: The tilted flat will never be placed in the machine exactly as modelled. In practice the flat will always be placed slightly off-center and never exactly at the specified rotation angle. The modelled surface change disturbance will therefore always differ from the actual surface change disturbance.

In the next section, section 3.2, it will be shown that the difference in magnitude of the jump at 0.5 $[Hz]$ is primarily caused by a combination of point 1 and 2.

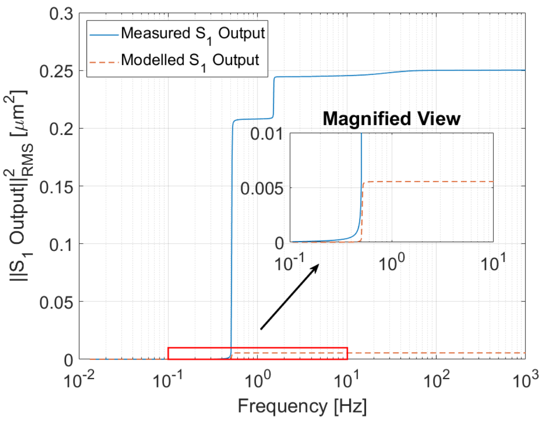


Figure 6: Comparison between the modelled and measured sensor output.

3.2. Non-linearity in the Spring Constant

In the previous section, section 3.1, it is shown that the magnitude of the disturbances at the rotational frequency are much larger in practice compared to the simulation. This difference is caused by the fact that the model does not take the non-linearity in the spring constant into account. The model currently assumes the spring constant remains the same over time. In practice however, the spring constant changes depending on the position of the system. To include this non-linearity, the system's transfer function has to be rewritten from a standard mass-spring-damper system, to the form presented in equation 8. It should be noted, that the presented equation is a combination between the time and frequency domain, x changes over time. This means that the equation can not be used analytically and is only valid in a numerical simulation. Furthermore, the equation assumes all the other parameters remain constant over time, which is not the case in practice. It was found however, that incorporating a position dependence in the spring constant enables the model to sufficiently predict the behaviour of the system.

$$G(s) = \frac{K_m}{ms^2 + bs + K(x)} \tag{8}$$

Quantifying the Non-linearity in the Spring Constant

The non-linearity in the spring constant can be derived from the force balance in the x-direction. Assuming the system is not accelerating, this force balance can be written as:

$$\sum F_x = F_{motor} + F_{spring} + F_{gravity} = 0 \tag{9}$$

When the system is positioned at a $90°$ angle, relative to the ground, gravity does not affect the extension of the spring. In this case, the spring constant can be calculated, using equation 10.

$$K(x) = \frac{F_{spring}}{x} = \frac{-F_{motor} - F_{gravity}}{x} \quad (10)$$

$K(x)$ at position x can therefore be derived by logging the motor force required to place the sensor at position x. In figure 7, the results of such tests across the system's full range of motion are presented. It should be noted that the irregularities around zero are caused by the fact that equation 10 encounters a zero division at this point. The small spikes at $\pm 2.5\ [mm]$ are caused by accelerations in the system. When the system is accelerating, equation 9 does not hold. Using MATLAB's polyfit function a continues approximation of $K(x)$ can be realized. This function essentially determines the coefficients of equation 11 that provide the best fit of the input data. In the equation, $C(i)$ represents a fitted coefficient. The polyfit approximation is used as $K(x)$ in the simulation to account for the non-linearity in the system. A fifth order polynomial was found to generate a sufficient fit. *The MATLAB script used to find the polynomial is shown in appendix G.4*.

$$K(x) = \sum_{i=1}^{k} C(i) x^{k-i} \quad (11)$$

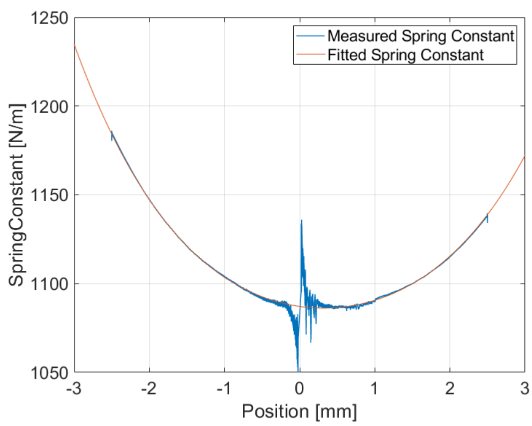


Figure 7: Measured spring constant, with a corresponding polynomial fit.

The CPS of the model with a position dependent spring constant (PDSC) is shown in figure 8. From the figure it can be concluded that including a position dependence in the spring constant allows the model to simulate reality much more accurately. Comparing the modelled and measured CPS, the difference between the curves is now much smaller.

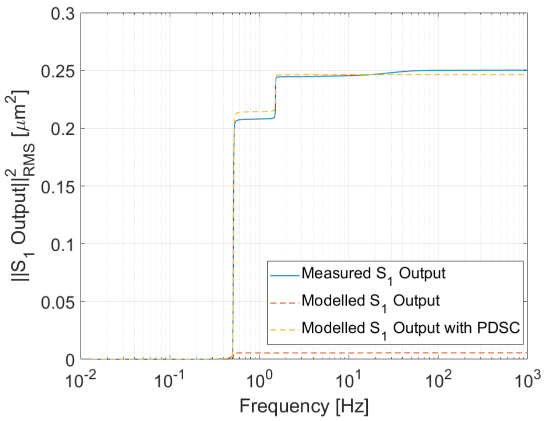


Figure 8: Impact of adding a position dependent spring constant on the accuracy of the model.

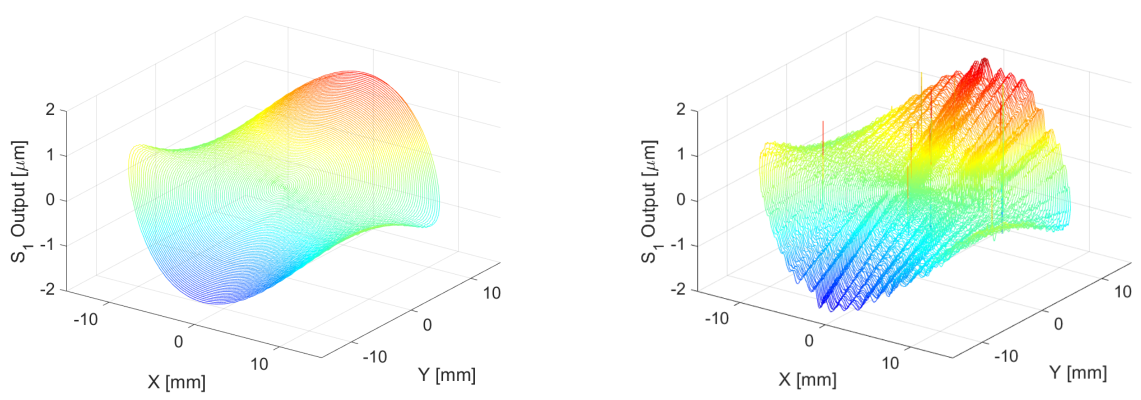
The large increase in the modelled S_1 output, can be explained using equation 12. In equation 12, the non-linearity, $NL(x)$, is essentially reformulated as a disturbance on the plant output, at D_2. The added disturbance is dependent on x and therefore in turn, dependent on the disturbances acting on the system. The non-linearity can be regarded as a second-order disturbance that arises from other disturbances changing the x-position of the system.

$$\begin{aligned} G_c(s) &= \frac{K_m}{ms^2+bs+K(x)-K} = \frac{K_m}{ms^2+bs} + \mathcal{L}\{NL(x)\} \\ \mathcal{L}\{NL(x)\} &= \frac{K_m}{ms^2+bs+K(x)-K} - \frac{K_m}{ms^2+bs} \end{aligned} \quad (12)$$

The modelled outputs presented in figure 8 show the effect of the non-linearity in the spring constant, assuming the surface change disturbance is the only disturbance acting on the system. The simulation shows that the non-linearity in the spring constant, increases the signal power from 0.00361 to $0.247\ [\mu m^2]$, therefore increasing the power by a factor of 68. It should be noted that the model assumes the positioning of the flat to be perfect and the parameters used to model plant do not exactly match the real parameters. These factors will cause a difference in the exact increase in signal power, it is however safe to assume that the errors caused by the non-linearity in the spring and the change in surface height, provide the biggest contribution to the tracking error. Furthermore, with a position dependent spring constant, the CPS of both the model and the measurement show a jump at $1.5\ [Hz]$. This indicates the jump at $1.5\ [Hz]$ is caused by the non-linearity in the spring.

3.3. Position Dependent Errors

When further evaluating the measured CPS of the S_1 output in figure 8. An increase of the CPS



(a) Modelled S_1 output during a 7° tilted flat measurement.

(b) Measured S_1 output during a 7° tilted flat measurement.

Figure 9: Comparison between the measured and modelled tracking error during a $7°$ tilted flat measurement

can be observed in the region between 10 and 100 $[Hz]$. This increase is not present in the CPS of the modelled output, indicating there are other disturbances present that are currently not being modelled. Evaluating the spiral trace of the S_1 output, see figure 9, these additional disturbances can be observed as ripples along the Y-axis.

These ripples always seem to occur at the same X-position. Since the measured flat is tilted around the Y-axis. The flat has the same height for all Y-positions, given a certain X-position. For each given X-position, the system is therefore at the same height as well. It is likely, that factors such as friction add an additional disturbance to the system, whenever the sensor is at a certain height. The exact disturbance added is unfortunately unknown, following the block diagram presented in figure 2 however, it is known that the magnitude of this disturbance is proportional to the magnitude of the sensitivity function. Furthermore, using the CPS plot, the active frequency range of the disturbance can be derived to be between 10 and 100 $[Hz]$. Therefore, to reduce the magnitude of the position dependent disturbances, the magnitude of the sensitivity function will have to be reduced in the region between 10 and 100 $[Hz]$.

3.4. Problem Statement

The objective of this research is to increase the tracking capability of a mass-spring-damper-system with a non-linear spring. In section 3.2, it is shown that the second-order disturbance caused by the non-linearity in the spring and the change in surface height is by far the biggest contributor to the tracking error. The main focus of the controller modifications presented in section 4 will therefore be on decreasing the non-linearity in the system and increasing the system's ability to reject the surface change disturbance. Additionally, section 3.3 showed there is a small part of the tracking error that is caused by position dependent disturbances in the 10-100 $[Hz]$ frequency range. The effect of these disturbances is relatively small, making them a lower priority for improving the system's performance. In section 4 however, it will be shown that as the effect of the non-linearity and surface change disturbance is reduced, the position dependent disturbances become dominant. In order to keep pushing the boundaries of the system's tracking capability, these disturbances will therefore have to be addressed as well.

4. Controller Modifications

In this section the implemented controller modifications will be presented, a justification for the made selection is provided and the impact of the modifications on the tracking capability will be shown. *The implemented modifications are a selection of the options presented in appendix D and E.*

4.1. Position Dependent Spring Force Compensation

In section 3.2, it is shown that the non-linearity in the spring constant, significantly degrades the performance of the system. To counter act this non-linearity, a position dependent spring force compensation (PDSFC) is proposed, *see appendix D.1 for a more detailed explanation*. Following the block diagram presented in figure 10, the constant K is essentially swapped for the position dependent variable $K(x)$ fitted in section 3.2. Feeding back a position dependent variable allows the SFC term to mimic the actual spring force more closely and effectively linearizes the system. This will in turn reduce the disturbances caused by the non-linearity and increase the predictability of the system, as linear evaluation techniques such as Bode plots can now be used more effectively. The combination of an improved tracking capability and an increased predictability of additional modifications, make PDSFC a highly suitable first modification. *The structured text implementation of this modification is shown in appendix H.1.*

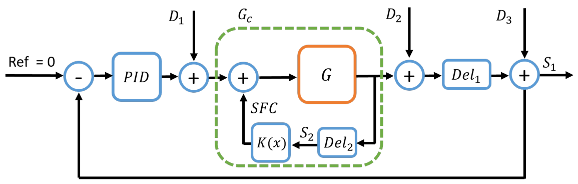


Figure 10: Block diagram illustrating the implemented control setup with PDSFC.

Result

The effect of the modification on a $7°$ tilted flat measurement is visualized in figure 11. Using PDSFC the signal power of the tracking error is reduced from 0.25 to 0.017 $[\mu m^2]$. Furthermore, the step at 1.5 $[Hz]$ has now disappeared. In section 3.2 it is shown that this step is caused by the non-linearity in the spring. The fact that this step is now gone is a clear indication that the non-linearity is largely suppressed.

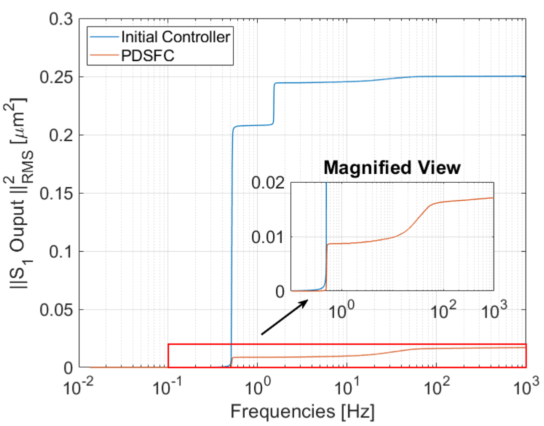


Figure 11: Measured CPS of the tracking error with constant and position dependent spring force compensation.

4.2. Feedforward Control

Assessing the magnified view in figure 11, it can be concluded that after the implementation of the position dependent spring force compensation, the build up of tracking errors essentially consists of two main parts: A jump at the rotational frequency, at 0.5 $[Hz]$, and an exponential increase between 10 and 100 $[Hz]$. In section 3.1, it is shown that the jump at the rotational frequency is caused by the change in surface height of the tilted flat. When a disturbance is known, it is possible to reduce the effect of this disturbance using feedforward control. A feedforward controller can be described using the block diagram presented in figure 12.

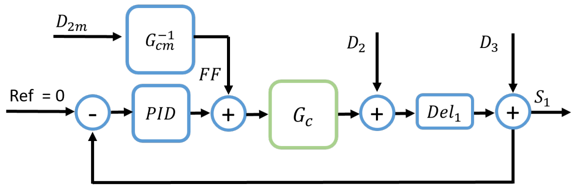


Figure 12: Block diagram illustrating the implemented control setup with a feedforward controller. $G_c(z)$ represents the spring compensated plant.

Only considering the surface change disturbance, $D_2(z) = SC(z)$, the S_1 output can be calculated using equation 13. The subscript m is used to make a distinction between a modelled and real variable.

$$S_1(z) = \frac{G_c(z)Del_1(z)}{1+L(z)}FF(z) + \frac{Del_1(z)}{1+L(z)}SC(z) \tag{13}$$

Where,

$$FF(z) = -SC_m(z)G_{cm}^{-1}(z)$$

$$L(z) = PID(z)G_c(z)Del_1(z)$$

When the plant model is perfect, equation 14 holds.

$$G_{cm}^{-1}(z) * G_c(z) = 1 \tag{14}$$

This means that equation 13 can be simplified to:

$$S_1(z) = \frac{-Del_1(z)}{1+L(z)}SC_m(z) + \frac{Del_1(z)}{1+L(z)}SC(z) \tag{15}$$

When the modelled change in surface height is perfect as well, the two terms in equation 15 cancel each other out and the the sensor output becomes zero. In practice the used models will never be perfect. It is therefore impossible to completely suppress the surface change disturbance. However, when the model has a very close resemblance to the actual system, as is shown to be the case in section 3.2, a performance increase can still be expected using feedforward control. Furthermore, feedforward control does not affect the closed-loop of the control system. This means that the waterbed effect, *described in section C.8*, does not apply. Using feedforward control the effects of a known disturbance can be reduced without having to sacrifice the performance of the control system outside the frequency band of that disturbance. The effect of a known disturbance is essentially reduced, while the impact of other disturbances remains the same. Specifically targeting a significant known disturbance, while keeping the effect of other disturbances the same, makes feedforward control a logical next step in improving the controller.

Feedforward Controller Design

In section 3.1, it is shown that the surface change disturbance, is essentially a pure cosine wave oscillating at the rotational frequency of the machine. This means that, in the case of a tilted flat, a feedforward controller only has to focus on a single frequency. As explained in the previous section, a feedforward controller essentially multiplies the known disturbance with the inverse of the plant. When only the rotational frequency is considered, this comes down to multiplying the disturbance and offsetting θ with the corresponding gain and phase of the plant inverse at the rotational frequency. The feedforward part of the controller can be described using equation 16. In the equation, M_{ip} and θ_{ip} refer to the gain and phase of the plant inverse at the rotational frequency respectively.

$$FF(t) = \hat{R}(t)sin(\phi)cos(\theta(t) + \theta_{ip}) * M_{ip} \quad (16)$$

Where,

$$\hat{R}(t) = -\sqrt{\frac{R(t)^2}{sin(\theta(t)) + \theta_{ip})^2 + (cos(\theta(t)) + \theta_{ip})cos(\phi))^2}}$$

Evaluating figure 10 it can be observed that the jump in the CPS at $1.5\ [Hz]$ disappears when using PDSFC. In section 3.2 it is shown that this jump at $1.5\ [Hz]$ is caused by the non-linearity in the spring. The fact that this jump no longer exists, is a clear indication that the variable spring force compensation has largely linearized the system. In the feedforward controller it will therefore be assumed that the system is linear and effectively acts as a mass-damper system. *The structured text implementation of the controller can be found in appendix H.2*.

Result

Figure 13 shows the CPS of a $7°$ tilted flat measurement of a controller using PDSFC, with and without feedforward control. In the figure a clear decrease in the jump at the rotational frequency can be observed, indicating the feedforward controller is working as expected. Using the feedforward controller, the signal power of the tracking error is further reduced, from 0.017 to $0.010\ [\mu m^2]$.

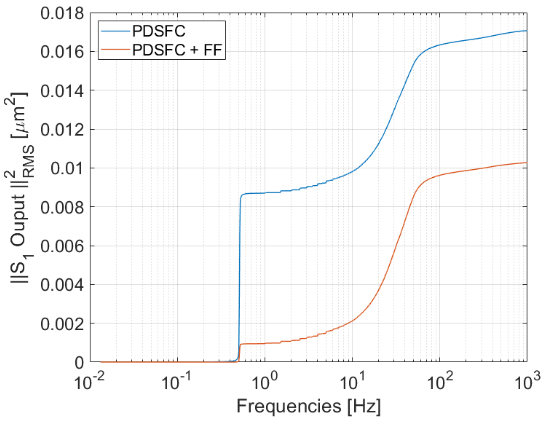


Figure 13: Measured CPS of the tracking error using PDSFC, with and without feedforward (FF) control.

4.3. Reset Control

Evaluating figure 13, it can clearly be observed that the largest contribution to the tracking error is now the exponential increase between 10 and $100\ [Hz]$. As explained in section 3.3, these errors are proportional to the magnitude of the sensitivity function. Therefore, to reduce the effect of these errors, the magnitude of the sensitivity function needs to be reduced in the $10 - 100\ [Hz]$ frequency range. Assessing figure 13, it can be concluded that the steepest slope of the CPS occurs around $40\ [Hz]$, which is the bandwidth. The magnitude around bandwidth is therefore most critical when improving the tracking error. To decrease the magnitude of the sensitivity function around the bandwidth, the phase margin of the system needs to be increased. A higher phase margin, results in a higher modulus margin. *As explained in section C.5*, the inverse of the modulus margin equals the peak in the sensitivity function. The peak in the sensitivity function is located around the bandwidth. A higher modulus margin will therefore result in a lower peak of the sensitivity function around the bandwidth.

When using linear control however, the waterbed effect always has to be considered, *see appendix C.8*. The waterbed effect essentially states that an increase of the sensitivity function in one area will always lead to a decrease of the sensitivity function in another. Re-tuning a linear controller for a higher phase margin will therefore always lead to a decrease of the sensitivity function at lower frequencies. The only way to avoid this trade-off is to implement a non-linear control strategy. As explained in the introduction, section 1, reset control is chosen for this purpose. Within the field of reset control, there many controllers available to choose from *see appendix E.4*. In this case, the addition of a 'Constant in Gain Lead in Phase (CGLP) controller is chosen, as it is specifically designed to generate a broadband phase lead without altering the gain of a system [7]. A block diagram of the new control system is shown in figure 14.

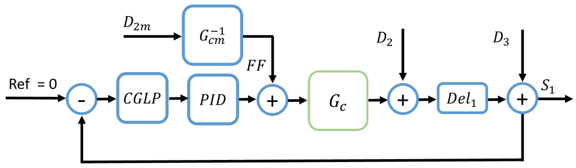


Figure 14: Block diagram illustrating the used control setup with a CGLP controller.

Reset Controller Design

A CGLP is essentially a series combination of a FORE [9] and a lead filter, *see appendix E.6*. These two components can be represented by equation 17. In the equation, the diagonal arrow indicates the state of the FORE is re-set to γ when the reset condition is met. Designing a CGLP, effectively comes down to choosing the parameters γ, α, ω_r and ω_f to achieve the desired performance.

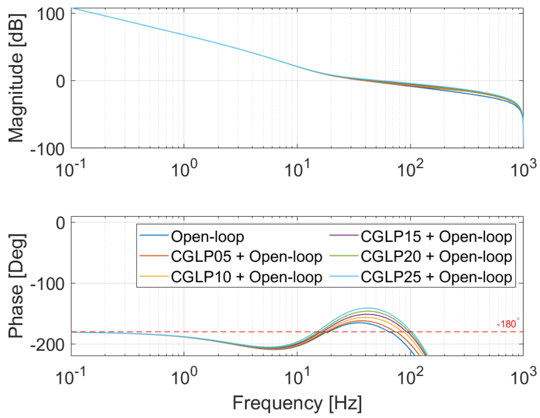
$$FORE(s) = \frac{1}{\frac{\alpha s}{\omega_r}+1} \nearrow \gamma \qquad Lead(s) = \frac{\frac{s}{\omega_r}+1}{\frac{s}{\omega_f}+1} \tag{17}$$

Since a reset controller is non-linear it is impossible to describe its behaviour with a linear transfer function. To still get an indication of the systems behaviour in the frequency domain, a describing function (DF) approximation is used [10]. A DF represents the transfer function between the input and the first harmonic of the Fourier series expansion of the output, *see appendix for a more detailed explanation E.2*. The more dominant the first harmonic is over the higher-order harmonics, the better the DF approximation [6]. When tuning the CGLP, the goal should therefore be to design a controller that meets the set criteria, with the lowest possible magnitude of the higher-order harmonics. To achieve this goal, the tuning procedure presented in [11] is used. *See appendix E.6.1 for a more detailed explanation.*

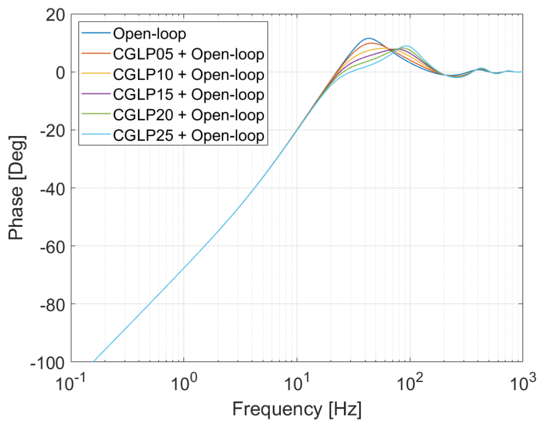
When the magnitude of the higher-order harmonics is increased, the DF approximation becomes more inaccurate. The point where the approximation becomes unreliable and the performance of the system starts to degrade is unknown. To deal with this uncertainty, multiple CGLPs were tested. Figure 15 shows the open-loops and corresponding sensitivity functions of the implemented CGLPs. To gradually build up the non-linearity, five controllers were tested with an additional phase at bandwidth between $0°$ and $25°$. The σ-values, which can be regarded as a measure for the amount of non-linearity [11], associated to the implemented CGLPs are shown in table 2.

	CGLP05	CGLP10	CGLP15	CGLP20	CGLP25
σ	$2.51e^{-06}$	$8.36e^{-06}$	$1.78e^{-05}$	$3.35e^{-05}$	$6.23e^{-05}$

Table 2: Corresponding σ-values of the implemented CGLPs.



(a) Open-loop transfer function.



(b) Sensitivity function.

Figure 15: Bode plot representation of the implemented control structures with various CGLPs. The number added to the CGLPs corresponds to the amount of additional phase at bandwidth.

Stability

A common way to prove stability for reset control systems is through the so-called H_β-condition [12]. This condition however, can be complex to solve and requires a state-space description of the plant, making it dependent on a model of the system [13]. To avoid this dependence and simply the calculations, [14] shows that for the reset control system presented in figure 14, the H_β-condition can be rewritten to prove stability through the evaluation of the Nyquist Stability Vector (NSV). Using the NSV it is possible to determine the stability of the tracking system using a frequency response function measurement of the plant. The NSV, $\vec{N(\omega)}$, is defined by equation 18.

$$\begin{aligned}\vec{N(\omega)} &= [N_x(\omega) + N_x(\omega)]^T \\ &= [\Re(M_1^*(j\omega)M_2(j\omega))\ \Re(M_1^*(j\omega)M_3(j\omega))]^T\end{aligned} \tag{18}$$

Where,

$$M_1(j\omega) = 1 + L(j\omega)(R(j\omega) + C_3(j\omega)$$
$$M_2(j\omega) = L(j\omega)C_s(j\omega)(R(j\omega) - D_r)$$
$$M_3(j\omega) = (1 + L(j\omega)(C_3(j\omega) + D_r))(R(j\omega) - D_r)$$

For the tracking system, the parameters are given by:

$$\begin{aligned}L(j\omega) &= C_1(j\omega)C_2(j\omega)G(j\omega) \\ R(j\omega) &= FORE(j\omega) \\ C_1(j\omega) &= Lead(j\omega) \\ C_2(j\omega) &= PID(j\omega) \\ C_3(j\omega) &= 0 \\ C_s(j\omega) &= 1\end{aligned}$$

The angle of the NSV is defined as $\theta_N(\omega) = tan^{-1}(\frac{N_y(j\omega)}{N_x(j\omega)})$. In [14] it is proven that the zero equilibrium of the tracking system is globally uniformly asymptotically stable if the conditions below hold.

1. The base linear system[2] is stable and the open-loop transfer function does not have any unstable pole-zero cancellation.
2. $-1 < \gamma < 1$
3. $(\theta_2 - \theta_1) < \pi$, with $\theta_1 = \underset{\forall\omega\in\mathbb{R}}{min}\ \theta_N(\omega)$ and $\theta_2 = \underset{\forall\omega\in\mathbb{R}}{max}\ \theta_N(\omega)$
4. $\frac{-\pi}{2} < \theta_N(\omega)$ and/or $0 < \theta_N(\omega) < \frac{3\pi}{2},\ \ \omega \in [0, \infty)$.

The implemented CGLPs all have a stable base linear system and a γ-value between -1 and 1. The stability of the controllers can therefore be derived from the angle of the NSV. In figure 16, this angle is presented for the implemented controllers. The figure shows that θ_n always stays between $0°$ and $180°$, indicating condition 3 and 4 are satisfied and the implemented CGLPs are stable. Figure 16 also shows that $(\theta_2 - \theta_1)$ increases when more phase is added by the CGLP. If this difference exceed $\pi\ [rad]$, the system is either unstable or another method will have to be used to prove its stability. *The script used to calculate the NSV is presented in appendix G.5.*

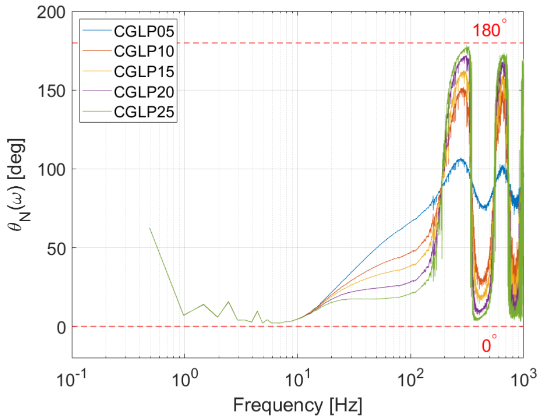


Figure 16: $\theta_n(\omega)$ of the implemented CGLPs.

Result

In figure 17, the CPS plot of a $7°$ tilted flat measurement is presented, for the system using a controller with PDSFC, feedforward control and various CGLPs. The spiral trace of the tracking error corresponding to the best performing CGLP, CGLP20, is shown in figure 18. Evaluating figure 15 it can be observed that the implemented tuning procedure increases the gain around $100\ [Hz]$. The figure also shows that this gain increase becomes larger when the amount of phase added by the CGLP increases. This increase in gain, in turn, increases the slope of the CPS and limits the performance of the CGLP. In figure 17, this effect can clearly be observed when comparing the slope of CGLP20 and CGLP25. Despite these limitations, the CGLP still enables a reduction of the tracking error's signal power. Adding a CGLP the signal power is further reduced from 0.010 to $0.0067\ [\mu m^2]$. *The structured text implementation of the controller can be found in appendix H.3.*

5. Conclusion and Future Work

In this paper a controller design approach is presented to improve the performance of a freeform optical surface tracking system. First, it is shown that a second-order disturbance caused by the non-linearity in the spring and the change in surface height is the primary contributor to the tracking error. To suppress the effect of this disturbance and linearize the system, a position dependent spring force compensation is introduced. Next, a

[2]Base linear system refers to the reset control system when the reset condition is not triggered, *see appendix E.1 for a more detailed explanation.*

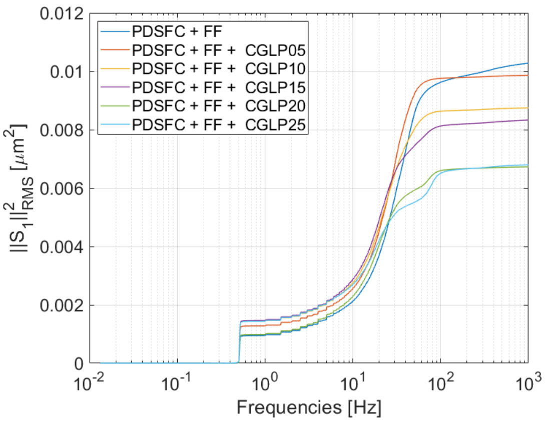


Figure 17: Measured CPS of the tracking error using various CGLPs.

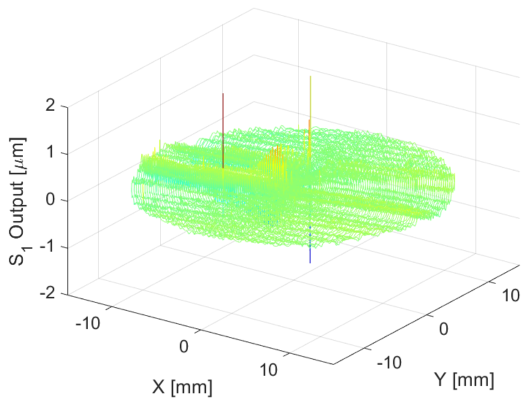


Figure 18: Tracking error during a $7°$ tilted flat measurement with PDSFC, feedforward control and CGLP20.

feedforward control structure is used to suppress the effect of the known disturbances, the change in surface height in this case. Following these modifications, unknown disturbances become the largest contributors to the tracking error. The impact of these disturbances is reduced through the addition of a CGLP controller, aiming to decrease the magnitude of the sensitivity function in the active frequency range of the disturbance. Adding all the modifications together, the signal power of the tracking error is reduced from 0.25 to 0.0067 $[\mu m^2]$.

Additionally, this paper shows that, using a CGLP, it is possible to increase the performance of the tracking system, highlighting the potential of reset control. Here it should be noted that reset control is still a very active field of research, many alternative and additional strategies can still be considered and further developed. The used tuning procedure for example, only considers the phase at bandwidth. Section 4.3, showed that this approach leads to a degradation of the performance around 100 $[Hz]$. Extending the procedure presented in [11] to consider a frequency band instead of only the bandwidth frequency could therefore potentially further increase the system's performance. Furthermore, [7] and [11] show that besides a first order CGLP, a second order CGLP also has the potential to increase the tracking capability of a control system. Additionally, [15] and [16] show that in addition to minimizing the magnitude of the higher-order harmonics, it is possible to shape these harmonics to boost the tracking capability of a control system further.

References

[1] Graham C. Goodwin, Stefan F. Graebe, and Mario E. Salgado. *Control System design*. Prentice-Hall, 2001.

[2] Robert Munnig Schmidt et al. *The design of high performance mechatronics: High-tech functionality by multidisciplinary system integration*. Delft University Press, 2020.

[3] Alfonso Baños and Antonio Barreiro. *Reset Control Systems*. Springer London, 2012.

[4] Karl Johan Åström and Richard M. Murray. *Feedback systems: An introduction for scientists and Engineers*. Princton university press, 2021.

[5] Karst Brummelhuis et al. "Adaptive Feedforward Control For Reset Feedback Control Systems - Application in Precision Motion Control". In: *2021 European Control Conference (ECC)*. 2021, pp. 2450–2457. DOI: 10.23919/ECC54610.2021.9654834.

[6] Niranjan Saikumar, Kars Heinen, and S. Hassan HosseinNia. "Loop-shaping for reset control systems: A higher-order sinusoidal-input describing functions approach". In: *Control Engineering Practice* 111 (2021), p. 104808. ISSN: 0967-0661. DOI: https://doi.org/10.1016/j.conengprac.2021.104808.

[7] Niranjan Saikumar, Rahul Kumar Sinha, and S. Hassan HosseinNia. ""Constant in Gain Lead in Phase" Element– Application in Precision Motion Control". In: *IEEE/ASME Transactions on Mechatronics* 24.3 (2019), pp. 1176–1185. DOI: 10.1109/TMECH.2019.2909082.

[8] L.A. Cacace. "An optical distance sensor : tilt robust differential confocal measurement with mm range and nm uncertainty". English. Phd Thesis 1 (Research TU/e / Graduation TU/e). Mechanical Engineering, 2009. ISBN: 978-90-386-2069-5. DOI: 10.6100/IR653288.

[9] Isaas Horowitz and Patrick Rosenbaum. "Non-linear design for cost of feedback reduction in systems with large parameter uncertainty". In: *International Journal of Control* 21.6 (1975), pp. 977–1001. DOI: 10.1080/00207177508922051.

[10] Yuqian Guo, Lihua Xie, and Youyi Wang. *Analysis and design of Reset Control Systems*. The Institution of Engineering and Technology, 2016, pp. 27–41. ISBN: 978-1-84919-703-8.

[11] Mahmoud Shirdast Bahnamiri et al. "Tuning of CgLp based reset controllers: Application in precision positioning systems". In: *IFAC-PapersOnLine* 53.2 (2020). 21st IFAC World Congress, pp. 8997–9004. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2020.12.2017.

[12] Orhan Beker et al. "Fundamental properties of reset control systems". In: *Automatica* 40.6 (2004), pp. 905–915. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2004.01.004.

[13] Ali Ahmadi Dastjerdi, Alessandro Astolfi, and S. Hassan HosseinNia. "Frequency-domain stability methods for reset control systems". In: *Automatica* 148 (2023), p. 110737. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2022.110737.

[14] Unpublished: S. Ali Hosseini and S. Hassan HosseinNia. "Frequency domain stability approach for reset control systems". In: ().

[15] Nima Karbasizadeh and S. Hassan HosseinNia. "Continuous reset element: Transient and steady-state analysis for precision motion systems". In: *Control Engineering Practice* 126 (2022), p. 105232. ISSN: 0967-0661. DOI: https://doi.org/10.1016/j.conengprac.2022.105232.

[16] Nima Karbasizadeh et al. "Band-Passing Nonlinearity in Reset Elements". In: *IEEE Transactions on Control Systems Technology* 31.1 (2023), pp. 333–343. DOI: 10.1109/TCST.2022.3178043.

A

Derivation Tilted Flat Surface Change

In this appendix an expression is derived for the height of a tilted surface. The derivation is based on the assumptions stated below.

1. The origin of the tilted surface is located at the origin of the coordinate system.
2. The surface is tilted around the y-axis.
3. The surface is perfectly flat. All changes in surface height are caused by the tilt in the surface.

A point on a circle in the xy-plane can be defined as presented in equation A.1. In the equation R_c and θ refer to the radius of the circle and the angle of the point with respect to the x-axis respectively. The angle θ, can also be regarded as the amount the flat is rotated around the z-axis.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} R_c cos(\theta) \\ R_c sin(\theta) \\ 0 \end{bmatrix} \tag{A.1}$$

Following equation A.2. This point can be rotated around the y-axis with an angle ϕ, which can be regarded as the tilt of a tilted flat. Equation A.2 gives the position of a point on a tilted flat, in 3D-space.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} cos(\phi) & 0 & sin(\phi) \\ 0 & 1 & 0 \\ -sin(\phi) & 0 & cos(\phi) \end{bmatrix} \begin{bmatrix} R_c cos(\theta) \\ R_c sin(\theta) \\ 0 \end{bmatrix} = \begin{bmatrix} R_c cos(\theta) cos(\phi) \\ R_c sin(\theta) \\ -R_c cos(\theta) sin(\phi) \end{bmatrix} \tag{A.2}$$

The radius in the reference frame of the tracking system, R, can be calculated as presented in A.3.

$$R = \sqrt{x^2 + y^2} = \sqrt{(R_c cos(\theta) cos(\phi))^2 + (R_c sin(\theta))^2} \tag{A.3}$$

The radius of the tip of the probe and the angle θ of the spindle are known. R_c can be derived from these variables by rewriting equation A.3, to the form presented in A.4.

$$R_c = \sqrt{\frac{R^2}{sin(\theta)^2 + (cos(\theta) cos(\phi))^2}} \tag{A.4}$$

The surface height is equivalent to the z-position of the tilted flat and can thus be calculated as:

$$z = -\sqrt{\frac{R^2}{sin(\theta)^2 + (cos(\theta) cos(\phi))^2}} sin(\phi) cos(\theta) \tag{A.5}$$

B
The Confocal Sensor

The accuracy of the system is currently determined by the accuracy of the confocal sensor, as it has the lowest accuracy of the used sensors. The confocal sensor is essentially a distance sensor that uses controlled chromatic aberration to focus each wavelength of a white light source at a different distance. Depending on the distance between the sensor and the surface, the surface will reflect a different color of light. This reflected light is measured by a spectrometer, which is part of the controller of the sensor, and can be regarded as a measure for the distance between the sensor and the surface. An illustration of the working of the confocal sensor is shown in figure B.1.

An angle in the surface, changes the way the light is reflected. This in turn changes the color measured by the spectrometer. To maintain an accurate distance indication, the measurement therefore needs to be calibrated for the tilt angle of the surface. To completely compensate for this tilt error a different correction needs be applied for each tilt angle and each surface height within the sensor's measurement range. The machine studied in this thesis is only calibrated for one distance. When controlling the system, this distance is regarded as the zero reference point. A tracking error or deviation from this reference causes the sensor to be outside its calibrate regime. This in turn causes the tilt correction to be less accurate. An inaccurate tilt correction results in a less accurate measurement.

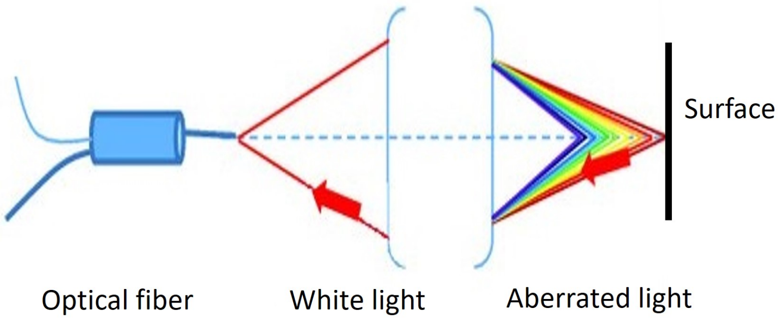


Figure B.1: Illustration of the working principle of the confocal sensor [1].

C Prerequisites

This appendix provides the theoretical basis needed to understand how the tracking system is currently controlled. The theoretical principles and concepts introduced in this appendix are used throughout the thesis to develop a comprehensive understanding of the system and its performance.

C.1. Linear Feedback Control

The system is currently controlled using linear feedback control. Linear feedback control is a specific branch of control theory that deals with linear systems. For a system to be linear it needs to be homogeneous and obey the principles of superposition. When a system is homogeneous, the scaling of an input results in the same scaling of the output. Superposition essentially means that a system's response to multiple inputs, equals the sum of the system responses to each of the inputs individually. The superposition principle is illustrated in equation C.1 below:

$$f(x_1 + x_2) = f(x_1) + f(x_2) \tag{C.1}$$

In linear control it is also assumed that the system variables remain constant over time, which means that the system is time-invariant. In literature this type of system is often referred to as a Linear Time-Invariant (LTI) system. Working with LTI systems is desirable as their properties significantly simplify the modeling and mathematics of systems. In practice however, systems are never linear and variables change over time. Fortunately, in most cases it is possible to linearize a system around its working point and cover deviations from reality through sufficient margins in the controller design. [2]

A standard block diagram of a linear feedback control system in the frequency domain is presented in figure C.1 below. In the diagram $C(z)$ and $G(z)$ refer to the plant and the controller, respectively. Both of these systems are LTI systems. In the control system, errors caused by model inaccuracies, disturbances and noise are accounted for by feeding back the systems measured state, denoted by y. The feedback line enables the controller to recognize errors and in turn compensate for them in its control output.

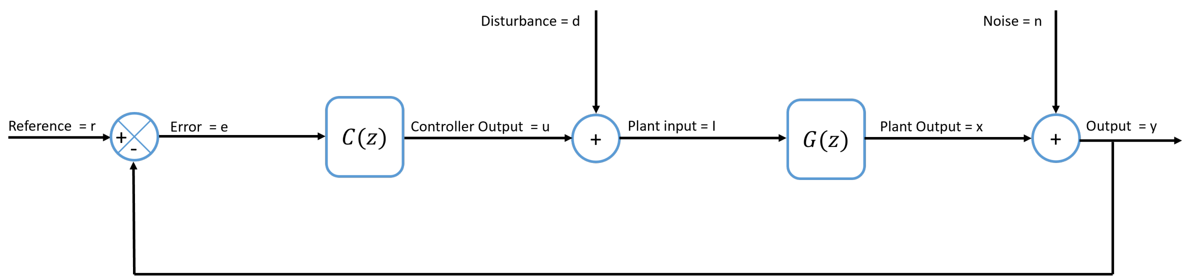


Figure C.1: Standard linear feedback control system.

C.2. Transfer functions

From the block diagram presented in figure C.1, the transfer functions presented below can be derived. These relations will be used in section 3 to evaluate the performance of the tracking system.

The complementary sensitivity function: The complementary sensitivity function, presented in equation C.2, is often referred to as the closed loop transfer function and represents the relation between the system output and the reference input. The complementary sensitivity function generates insight in the system's ability to track a reference signal.

$$T(z) = \frac{y}{r} = \frac{C(z)G(z)}{1 + C(z)G(z)} \tag{C.2}$$

The sensitivity function: The sensitivity function, shown in equation C.3, represents the relation between the output and the noise. It generates insight in the ability of a system to attenuate noise. The sensitivity function can also be regarded as the ratio between the tracking error and the reference, making it a useful indicator of a system's performance. Furthermore, it is worth noting that the complementary sensitivity function is the complement, hence the name, of the sensitivity function. This means $T(z) + S(z) = 1$. As a result, when the sensitivity function is small, the complementary function is large and the other way around.

$$S(z) = \frac{y}{n} = \frac{e}{r} = \frac{1}{1 + C(z)G(z)} \tag{C.3}$$

The process sensitivity function: The process sensitivity function, depicted in equation C.4, represents the relation between output and the process disturbances. The impact of disturbances should be a low as possible. This means that the process sensitivity function needs to be minimized.

$$P(z) = \frac{y}{d} = \frac{G(z)}{1 + C(z)G(z)} \tag{C.4}$$

C.3. Visualisation Tools

To get an impression of the performance and stability of a system, the transfer functions needs to be analyzed. For this analysis, it is useful to visualize the functions. In this section, two commonly used visualizations, the Bode and Nyquist plot, are discussed.

C.3.1. Bode Plot

When dealing with a sinusoidal input in steady state, a LTI system essentially only changes the phase and the amplitude of the input. A Bode plot visualizes this change, by plotting the amplitude gain and phase shift across a series of pre-specified frequencies. The reaction of the system to inputs characterized by a single frequency can therefore conveniently be analyzed in a Bode plot. Using a Fourier series expansion, every input can be approximated as a series of sinusoids. When the inputs are known, it is therefore possible to get an indication of a system's performance by looking at a Bode plot over a range of frequencies. A Bode plot of a mass-spring-damper is shown as an illustration in figure C.2 below.

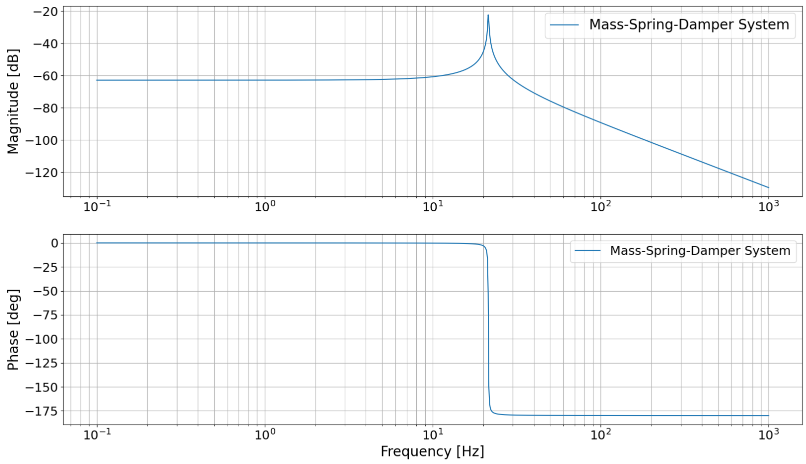


Figure C.2: Illustration of a Bode plot using a Mass-Spring-Damper system.

A Bode plot of the open-loop transfer function, $L(z) = C(z)G(z)$, can provide valuable information on the stability of a system. Looking at the closed loop transfer function, presented in equation C.2, it can be concluded that when the open-loop has a value of -1, equivalently a $0\ [dB]$ magnitude and $-180\ [deg]$ phase shift, that the closed loop transfer function goes to infinity and is thus unstable.

C.3.2. Nyquist Plot

Instead of showing the magnitude gain and phase shift in two plots, it is also possible to visualize the system's response in one plot in the complex plane. This type of visualization is called a Nyquist plot. A Nyquist plot of the open-loop is shown in figure C.3. Compared to a Bode plot, it is less easy to see the exact change of a sinusoidal input in a Nyquist plot. The Nyquist plot does however, offer a much more straightforward way to assess a system's stability.

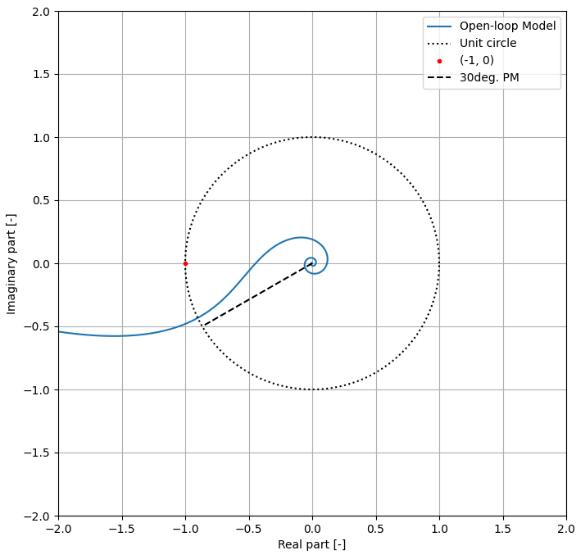


Figure C.3: Illustration of a Nyquist plot using the open-loop transfer function of the tracking system.

The stability of a system can be assessed using the Nyquist stability criterion, which essentially states that a system is stable if the number of encirclements of the $(-1, 0)$ point is lower then the number of poles, which are the roots of the denominator of a transfer function, in the right half plane [3]. The tracking system evaluated in this thesis does not have any poles in the right half plane. This means that the it is stable as long as the open-loop does not encircle the $(-1, 0)$ point in the Nyquist plot.

C.4. The Step Response

It also possible to analyze the performance of a system in the time domain. To measure this performance, the reaction of a system to a step input change is often considered. The reaction can be characterized by the indicators presented below. It should be noted that the definitions of these indicators vary slightly in literature. The definitions presented below, from [3], are the definitions that will be used during this thesis.

- Rise time, t_r: The time it takes a system to reach its new set point for the first time.
- Overshoot, M_p: The maximum amount a system exceeds its new set point value.
- Settling time, t_s: The time it takes a system to enter the deviation band $\pm\delta$. With δ being 2% of the new set point value.

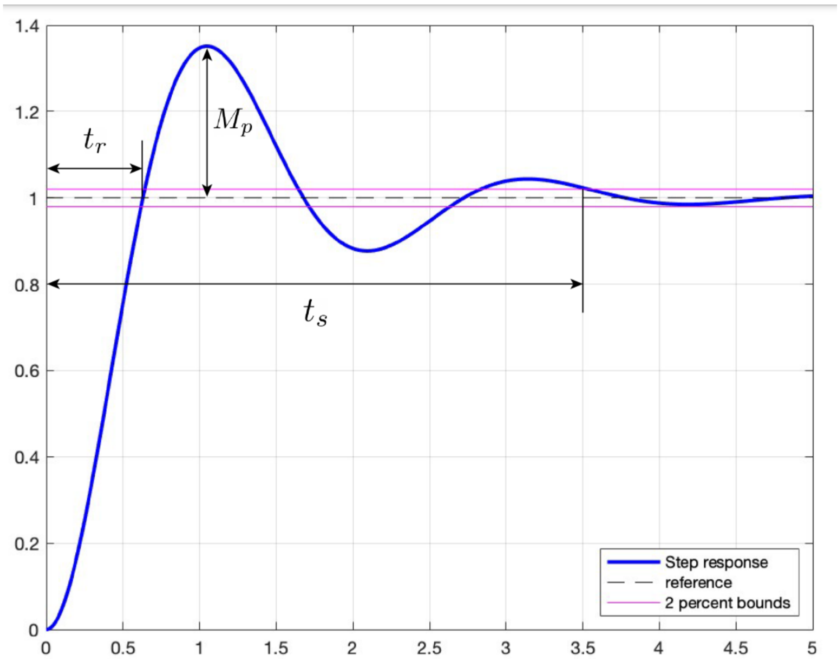


Figure C.4: Visualization step response performance indicators. [4]

C.5. Robustness

The robustness of a system essentially refers to a systems ability to maintain stability and performance when the plant deviates from the modelled plant. In practice, parameters are always changing. The damping of a system for example is dependent on various factors. Variables such as temperature, humidity and friction all influence the damping of a system. Even though the influence of these effects is often only minor, the model of a system will never work exactly the same as the system in practice. In order to account for this difference it is wise to add sufficient margins in the system to ensure stability even when the system does not behave exactly as predicted.

Three frequently used margins are the phase margin, gain margin and the modulus margin [2]. Each of these margins will be discussed shortly. To illustrate the concept, the stability margins of the used tracking system are shown in a Nyquist and Bode plot in the figures below.

- **Gain margin** (GM): The difference between the open-loop gain at the cross-over frequency and the loop-gain at the point where the phase becomes more negative then $-180°$. It can be regarded as the factor by which the feedback loop-gain can increase before the closed-loop system becomes unstable.
- **Phase margin** (PM): The phase of the open-loop gain at the cross-over frequency minus $180°$. Can be regarded as the amount of phase lag that is acceptable before the system becomes unstable.
- **Modulus margin** (MM): The closest distance between the open-loop and the -1 point in the Nyquist plot. When the system encircles the -1 point, the system becomes unstable. The distance to this point is therefore a measure of stability. The MM is also the largest magnitude of the sensitivity function in a Bode plot.

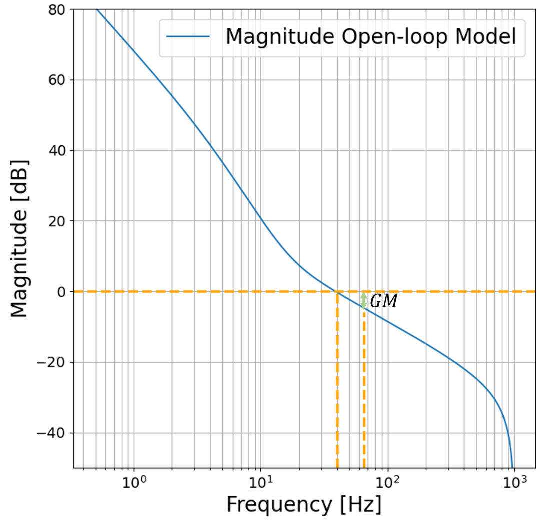


Figure C.5: Visualization gain margin Bode plot.

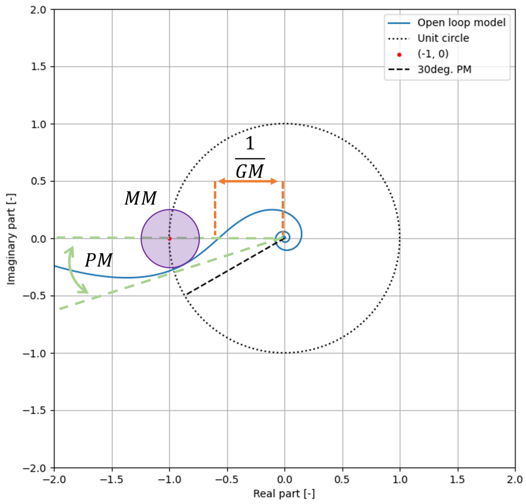


Figure C.6: Visualization stability margins Nyquist plot.

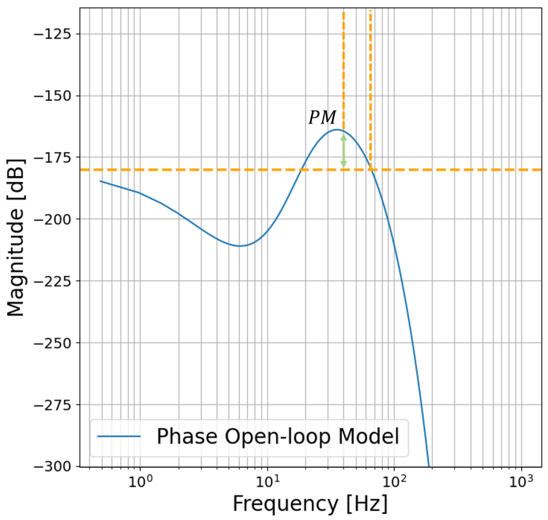


Figure C.7: Visualization phase margin Bode plot.

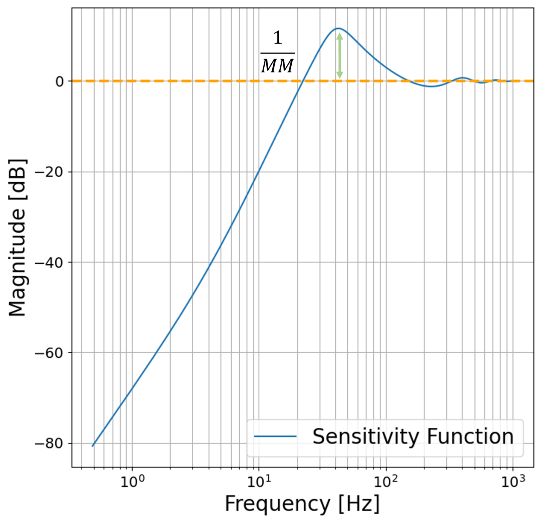


Figure C.8: Visualization modulus margin Bode plot.

The amount of margin needed, is determined by the errors in the model. A multiplicative description of modeling errors is given in equation C.5. In the equation P_n and P refer to the modelled and actual plant respectively.

$$P(s) = P_n(s)(1 + \delta P(s)) \tag{C.5}$$

If the real system is stable, the open-loop does not encircle the -1 point in the Nyquist plot. According to [5], this means that the distance between the real open-loop and the modelled open-loop is always smaller then the MM, as shown in equation C.6. In the equation, $|C(s)P_n(s)\delta(s)|$ represents the difference between the open-loops and $|1 + C(s)P(s)|$ represents the MM of the model.

$$|C(s)P_n(s)\delta P(s)| < |1 + C(s)P_n(s)| \qquad \forall\omega \geq 0 \tag{C.6}$$

Equation C.6 makes it possible to calculate a maximum modelling error that maintains a closed-loop stable plant. The maximum error can be calculated following equation C.7.

$$|\delta P(s)| < \frac{|1 + C(s)P_n(s)|}{|C(s)P_n(s)|} \qquad \forall\omega \geq 0 \tag{C.7}$$

C.6. PID Control

Proportional Integral Derivative (PID) control is the most popular feedback control system in the industry. A time- and corresponding z-domain representation of a PID controller is shown in equation C.8 below. In the equation k_p, k_i and k_d denote the control parameters. T_s represents the sampling time.

$$u[t] = k_p e[t] + k_i \sum_{t=0}^{\infty} e[t] + k_d(e[t] - e[t-1]) \quad \overset{z}{\Leftrightarrow} \quad U(z) = k_p + k_i T_s \frac{z}{z-1} + \frac{k_d}{T_s}(1 - \frac{1}{z}) \tag{C.8}$$

PID controllers are very popular, because they are easy to implement, only three parameters that need to be tuned, and at the same time provide sufficient performance for a large variety of plants. The tuning of the control parameters can be done using the principles of loop shaping, presented in section C.7. Additional filters can also be used the increase the performance further. [2]

C.7. Loop Shaping

The open-loop transfer function is given by $L(z) = C(z)G(z)$. Analyzing the open-loop allows for a direct observation of the impact of controller adjustments, as the controller is only multiplied by the plant. In the transfer functions presented in C.2 this is not the case and the effects of changing the controller are less transparent. In order to achieve the desired performance of a system it is therefore common practise to look at the frequency response of the open-loop when adjusting the controller. Adjusting the controller to generate a desired shape of the open-loop in the frequency domain is called loop shaping.

Controlling a system essentially entails having the capability to place a system in a desired state at any given time. Perfect control implies that the reference input consistently matches the system's output. The complementary sensitivity function presented in subsection C.2, can be regarded as measure of this match. Neglecting the effects of noise and process disturbances, perfect tracking would be achieved when the complementary sensitivity function is equal to 1.

In practice noise and disturbances are always present and they should be suppressed by the control system as much as possible. Section C.2 showed that when the complementary sensitivity function is close to 1, the sensitivity function is close to zero. A sensitivity function close to zero essentially means that noise is not suppressed. In figure C.1 it can be seen that noise is added to the plant output and fed back to the controller. When the noise is not suppressed, the controller follows the noise instead of the reference. This can have a significant impact on the performance.

Fortunately noise is usually mostly present at high frequencies. When designing a controller, the goal is therefore often to have the complementary sensitivity function close to 1 at low frequencies and close to zero at high frequencies. From the equation of the complementary sensitivity function it can be derived that this implies a high open-loop gain at low frequencies and a low open-loop gain at high frequencies. Additionally, to ensure robustness and stability, sufficient phase margin should be added at the cross-over frequency. Putting these factors together, the desired shape of the open-loop transfer function presented in figure C.9 can be derived.

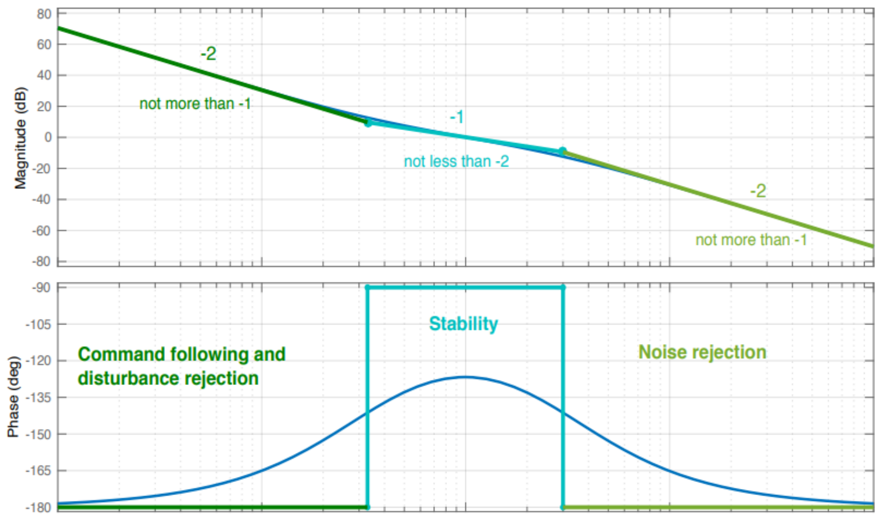


Figure C.9: Desired shape of the open-loop transfer function [6].

Besides tuning the shape of the open-loop, the bandwidth is another important parameter to tune. A higher bandwidth results in a faster settling time. Increasing the bandwidth essentially means moving the open-loop up in the frequency domain. This causes the gain to be higher at each frequency, which in turn leads to a decrease of a systems ability to attenuate noise. As explained in [7], a trade-off thus exists between the settling time and a systems ability to attenuate noise.

C.8. Limits of Linear Control

Disregarding the limits imposed by the capabilities of the sensors and the actuators, using linear control puts inherent limitations on the performance of a system. In this section the limitations most relevant for the tracking system used in this thesis will be discussed.

C.8.1. The Waterbed Effect

One of the most well known limitations of linear control, is the so called 'Waterbed effect' [8]. The waterbed effect can be described mathematically using Bode's sensitivity integral as presented in equation C.9. In the equation p_k represents a pole in the right half plane.

$$\int_0^\infty log|S(i\omega)|d\omega = \pi \sum p_k \tag{C.9}$$

Bode's sensitivity integral is visualized for the used tracking system in figure C.10. Since the tracking system does not have any right half plane poles, according to C.9, the area under and above the zero dB line must be equal. This means that an increase of the sensitivity function in one place will always lead to a decrease in another area. In other words, if you increase the the noise/disturbance attenuation in one frequency range, you have to decrease the attenuation in another. This trade-off is known as the Waterbed effect. [9]

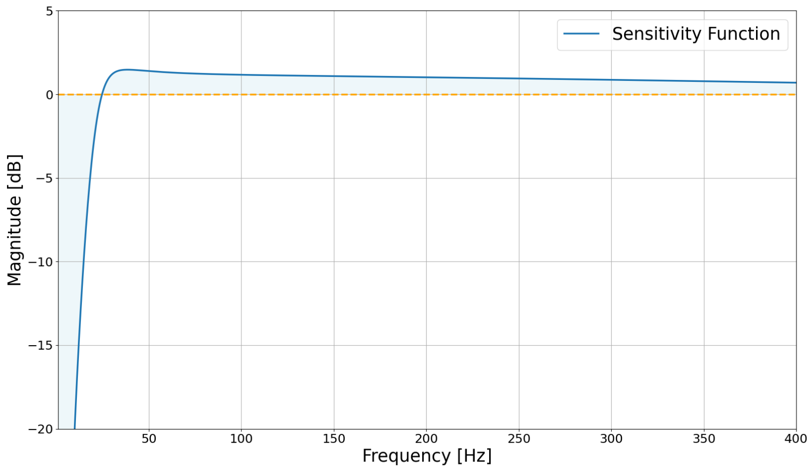


Figure C.10: Visualization of Bode's sensitivity integral using the sensitivity function of the probe.

C.8.2. Bode's Phase-Gain Relationship

When using linear control, the relation between the phase and the gain of a minimum phase system, is uniquely determined by Bode's phase-gain relationship. The relation and the approximated relation are presented in equation C.10 and C.11 respectively. In the equation, n represents the slope of the magnitude in a Bode plot. When loop shaping, see section C.7, the phase-gain relation essentially implies that a slope n always comes with a phase of around $90n°$. This means that when adding for example an integrator, a -1 slope, the phase will always drop 90 degrees.

$$\angle G(j\omega_0) = \frac{1}{\pi}\int_{-\infty}^{\infty} \underbrace{\frac{dln|Gj\omega|}{dln(\omega)}}_{n} ln|\frac{\omega+\omega_0}{\omega-\omega_0}|\frac{d\omega}{\omega} \tag{C.10}$$

$$\angle G(j\omega_0) \approx 90°n \tag{C.11}$$

For non-minimum phase systems, such as systems with delay, the relation becomes even stricter. To get an accurate impression of the phase-lag, one should add the phase lag caused by the delay to the phase-lag calculated through the phase-gain relation. In the case of systems with delay, the phase-gain relation can therefore be regarded as an indication of the maximum achievable phase. In practice, the actual phase will always be lower. [7]

C.8.3. Limitation Imposed by Time Delays

Even with 'ideal' control, time delay can not be removed as this would require input from the future. As explained in section C.2, perfect tracking implies that the complementary sensitivity function equals one for all frequencies. When there is delay in the the system, this equality transforms to $T(s) = e^{-\theta s}$, here θ denotes the time delay. This means that the sensitivity function will be $S(s) = 1-T(s) = 1-e^{-\theta s}$. Using a Taylor series expansion for the exponent, the equality can be transformed to $S(s) \approx 1-(1-\theta s) = \theta s$. In the case of perfect tracking, $S(s) = \frac{1}{L(s)}$, which means $L(s) = \frac{1}{\theta s}$. Using this relation, an upper bound on the bandwidth can be derived as presented in equation C.12. [10]

$$\omega_c < \frac{1}{\theta} \tag{C.12}$$

C.9. Dynamic Error Budgeting

As explained in section C.7, disturbances can have a significant impact on the performance of a system. Many disturbance sources have stochastic properties, which means that they are random and consist of a range of frequencies. A Bode plot represents the response of a system to each single input frequency. The effect of a range of frequencies is more difficult to observe. In order get an indication of the impact of a disturbance source on the performance of a system, the power of that signal can be used. [11]

For a disturbance source with a zero mean, which is often the case for stochastic disturbance sources, the power is equal to the variance of a signal and defined by equation C.13 presented below. In the equation x and T represent the disturbance signal and the time period respectively.

$$||x||_{rms}^2 = \lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{T} x(t)^2 dt \tag{C.13}$$

Parseval's Theorem states that the energy in the time domain equals the energy in the frequency domain [11]. Using this theorem the signal power can be converted to the frequency domain using equation C.14. In the equation, PSD refers to the Power Spectral Density. The PSD is a measure of the power of a signal at a given frequency component and is defined in continues time by equation C.15. The integral of the PSD over all positive frequencies is called the Cumlative Power Spectrum CPS. The CPS represents the square of the RMS error induced by a disturbance source [12]. The MATLAB script used to calculate the CPS is presented in appendix G.6.

$$||x||_{rms}^2 = \lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{T} x(t)^2 dt = \underbrace{\int_{0}^{\infty} PSD(f) df}_{CPS} \tag{C.14}$$

$$PSD(f) = \lim_{T\to\infty} \frac{|X(f)|^2}{T} \tag{C.15}$$

Usually, various disturbances are present in a system. In figure C.11 a closed-loop system with multiple disturbance sources is presented. According to [11], when modeling the disturbance sources in a system, it is possible to add up the influence of each independent source, if the sources are uncorrelated. This property generates insight into the total error and the individual contribution of each independent disturbance source on this error. It allows a designer to allocate budgets to the different disturbance sources, hence the name dynamic error budgeting. 'Dynamic' refers to the use of the frequency dependent models.

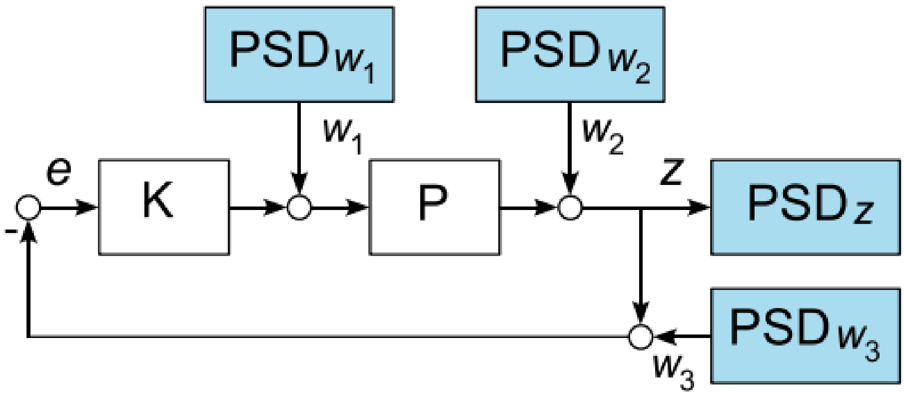


Figure C.11: Closed-loop system with various disturbance sources [11].

D

Linear Control Strategies

The considered linear control strategies will be presented in this appendix. In the final section, section D.5, these strategies are evaluated, providing a justification for the control strategies that were implemented.

D.1. Position Dependent Spring Force Compensation

In section 3.2 it is shown that it is possible to accurately measure the non-linearity in the spring force to model to the behaviour of the tracking system more reliably. Besides an improved model accuracy, section 3.2 also shows that the non-linearity in the spring constant causes a significant degradation in the performance of the spring force compensation. Disregarding the effects of position dependent disturbances, the model of the system without a position dependent spring constant, essentially shows how the system would behave if the fed-back spring constant equals the actual spring constant at all times. In section 3.2 it is shown that when this is the case, the modelled signal power of the S_1 output is 68 times lower then the signal power when using the current spring force compensation.

A logical step to improve the current spring force compensation is therefore, to use the available data on the spring constant and feed back a position dependent variable instead of a constant. In a block diagram this change can be visualized following figure D.1. The used $K_{est}(x)$ equals the polyfit derived in section 3.2. Furthermore, the spring compensated plant, G_c, can be regarded as the closed-loop transfer function of the inner positive feedback loop. Equation D.1 shows how this feedback loop can be rewritten as standard mass-spring-damper plant. The derivation shows that the spring compensated plant can be regarded as a mass-spring-damper where the spring constant equals the difference between the actual spring constant and the estimated spring constant.

$$G_c(s) = \frac{G(s)}{1 - G(s)K_{est}e^{-sDel_2}} = \frac{1}{G(s)^{-1} - K_{est}(x)e^{-sDel_2}} = \frac{1}{ms^2 + bs + K - K_{est}(x)e^{-sDel_2}} \tag{D.1}$$

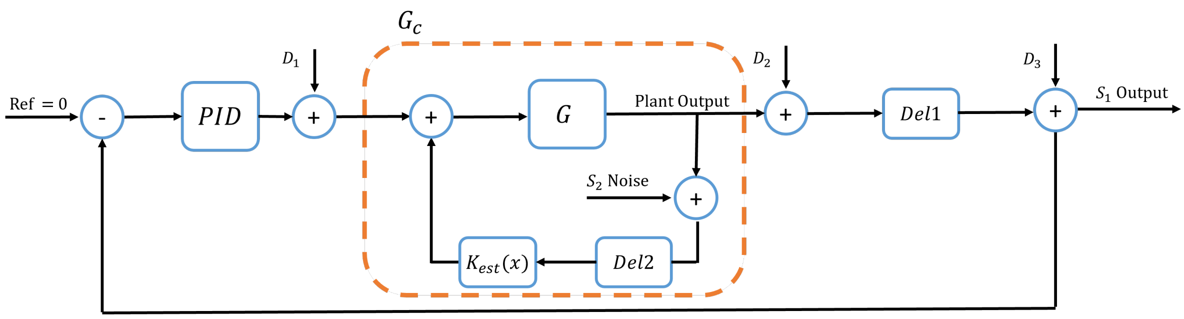


Figure D.1: Block diagram representation of the system with position dependent spring force compensation.

When the polyfit perfectly matches the non-linearity in the spring, the system is converted to a mass-damper system. When the mass and damping in the system remain constant there are no non-linear elements and the position dependent spring force compensation (PDSFC) essentially linearizes the system. A Bode plot of a mass-damper and mass-spring-damper system is shown in figure D.2. The spring force compensation essentially converts the slope of the magnitude plot from 0 to -1 at low frequencies. This steeper slope generates a higher plant gain, and in turn open-loop gain, at low frequencies. Following the block diagram presented in figure D.1, the S_1 output, or tracking error, can be calculated as shown in equation D.2. Evaluating equation D.2, it can be concluded that the tracking error is proportional to the sensitivity function. When the magnitude of the open-loop is higher, the magnitude of the sensitivity function is lower. Compensating the spring force therefore reduces the tracking error of the system.

$$S_1(z) = \underbrace{\frac{1}{1 + PID(z)G_c(z)Del_1}}_{Sensitivity\ Function}(G_c(z)Del_1 D_1(z) + Del_1 D_2(z) + D_3(z)) \tag{D.2}$$

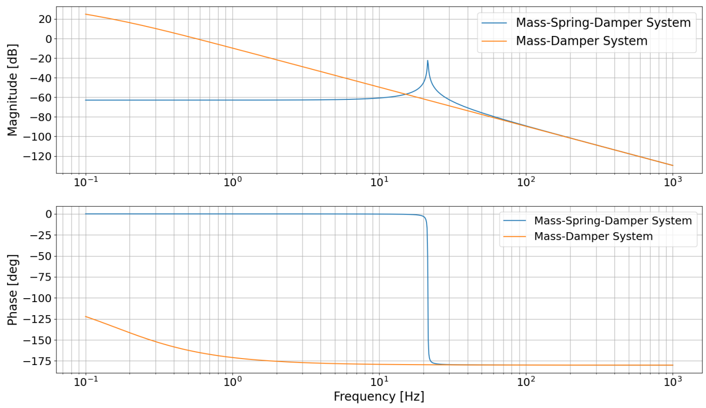


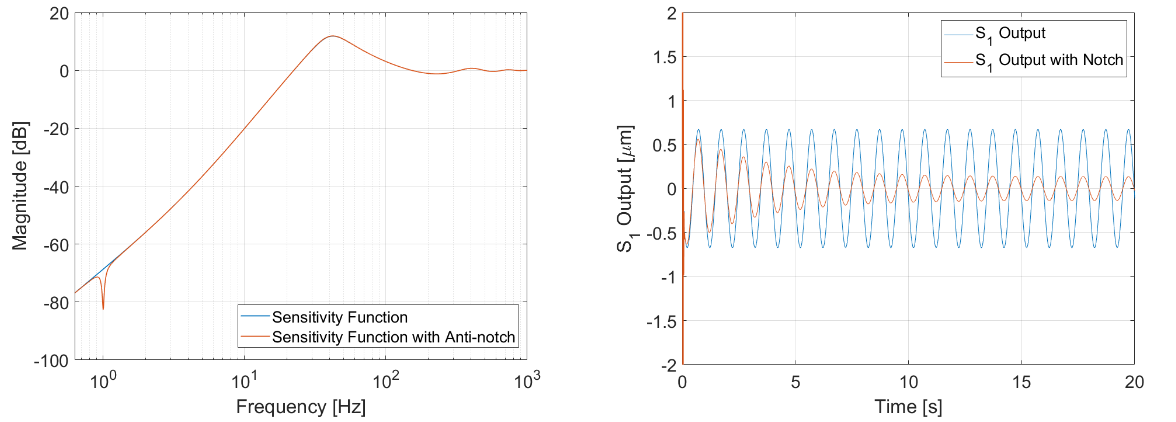
Figure D.2: Illustration of the difference between a mass-damper and a mass-spring-damper system.

D.2. Anti-Notch Filter

In section 3.1 it shown that the surface change disturbance is proportional to the magnitude of the sensitivity function. A common way to decrease the magnitude of the sensitivity function at a specific range of frequencies is through the use of an anti-notch. An anti-notch is a linear filter that adds a peak in the magnitude plot of the open-loop. As a consequence of the open-loop peak, the sensitivity function will have a local valley. A generalized form of a anti-notch filter is presented in equation D.3 below. [2]

$$F(s) = \frac{(\frac{s}{\omega_n})^2 + \frac{s}{Q_1\omega_n} + 1}{(\frac{s}{\omega_n})^2 + \frac{s}{Q_2\omega_n} + 1} \tag{D.3}$$

In the anti-notch filter ω_n determines the location of the the peak, the ratio between Q_1 and Q_2 determines the height of the peak and the absolute value of Q_1 and Q_2 determines the width of the peak. In figure D.3 the sensitivity function and tracking error with and without an anti-notch filter are presented.



(a) Sensitivity function of the system with and without an anti-notch in the controller.

(b) System's response to a tilted flat surface change for a controller with and without an anti-notch.

Figure D.3: Implications of adding an anti-notch filter.

Evaluating the tracking error, figure D.3b, it can be concluded that the anti-notch decreases the tracking error of the system, when the surface change disturbance is a simple sine wave. In practice however, it should be possible to measure any arbitrary shape. In such cases, the surface change will not follow a sinusoidal pattern and will have a much smaller correlation to a specific range of frequencies. This makes the use of anti-notches less attractive in practise.

D.3. Feedforward Control

Given that changes in the input of the system are known or measurable, which is the case for the surface change disturbance, it is possible to compensate for these changes using feedforward control. Figure D.4 shows a typical feedforward control architecture for a known disturbance D_2. In the ideal case, the disturbance is multiplied with the inverse of the plant, as presented in the figure D.4. When this is the case, the feedforward input perfectly cancels the disturbance input and the system behaves as if D_2 is not present. It should be noted that this is only the case if the model perfectly matches the actual plant. [7]

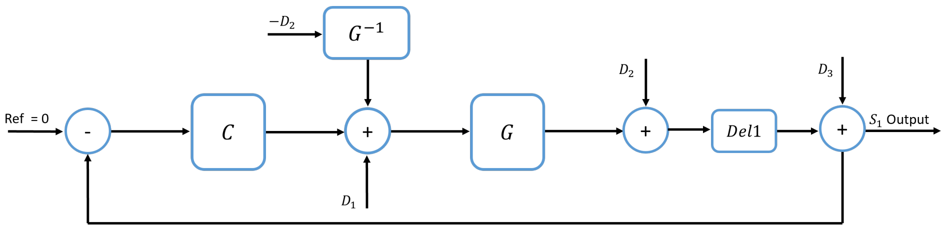


Figure D.4: Block diagram of a typical feedforward control architecture.

The control architecture presented in figure D.4 is unfortunately unrealizable in practise as the inverse of the system's transfer function is improper. When a transfer function is improper, its numerator has a greater order then its denominator. This means that the transfer function goes to infinity as the frequency increases. Infinite gain is impossible to achieve for practical systems. Therefore, to make the transfer function realizable, the function first needs to be made proper. [2]

In the case of a sinusoidal input, only one frequency has to be considered. This means that the transfer function can be simplified to a gain block. Similar as with the anti-notch filter, when a more complex shape is considered, more frequencies will be involved and a simple gain block will not suffice. In this case, the inverse of the plant needs to be used and additional poles will need to be added to make the transfer function proper, see figure D.5. The additional poles reduce the disturbance rejection at high frequencies. However, when the disturbance mainly consists of low frequency components, a performance increase can still be achieved.

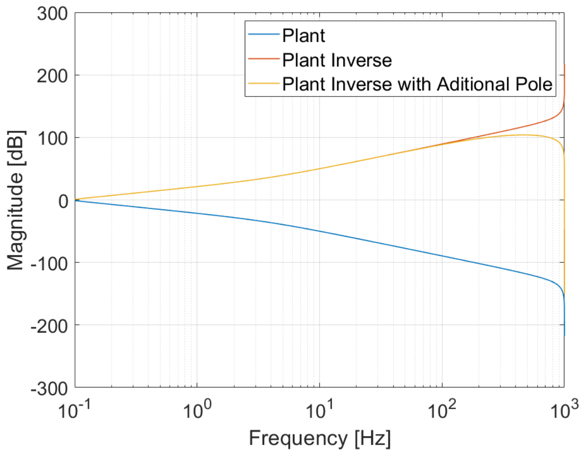


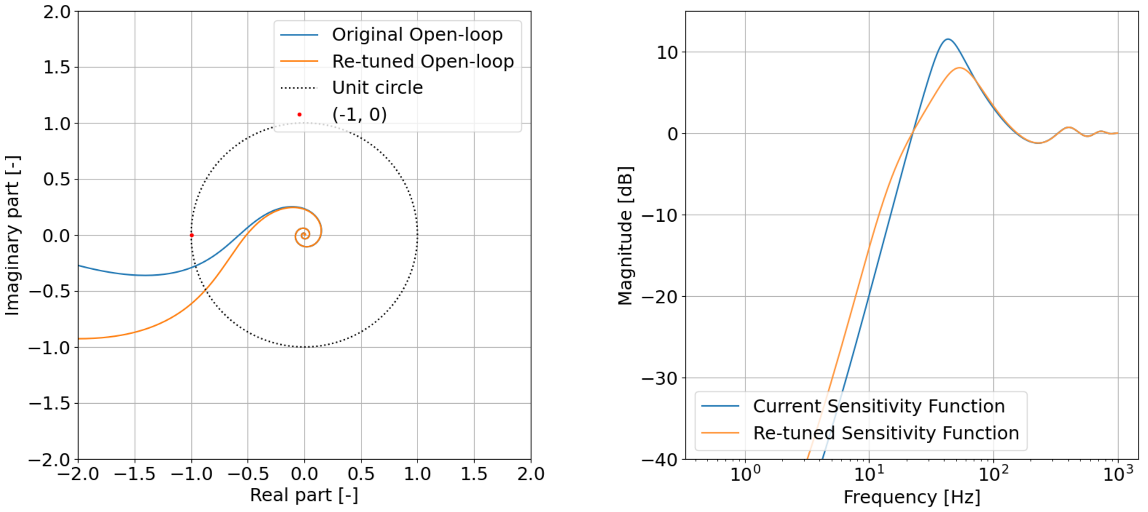
Figure D.5: Illustration of the impact of adding additional poles to the plant inverse.

D.4. Re-Tuning the Controller

When it is not possible to predict a disturbance beforehand, its impact will have to be reduced through feedback control. In section 3.3, it is shown that the unknown disturbances are primarily present in the $10 - 100\ [Hz]$ frequency range. To suppress the effect of these disturbances, the magnitude of the sensitivity function will have to be reduced in this area. In this case the active frequency range of the disturbance is located around the bandwidth. In this area the open-loop transfer function is close to 1. This means that the denominator in the sensitivity function can not be simplified to just the open-loop. The loop-shaping method described in section C.7 therefore does not hold. To reduce the magnitude of the sensitivity function around bandwidth, both the phase and the magnitude of the open-loop have to be considered.

The sensitivity function, equals the inverse of the open-loop minus 1. This causes the distance between the open-loop and the -1 point in a Nyquist plot to equal the inverse of the magnitude of the sensitivity function. In a Nyquist plot, the bandwidth equals the frequency where the open-loop crosses the unit circle. When evaluating this crossing for the Nyquist plot of the tracking system, see figure D.6a, it can be concluded that in order to decrease the magnitude of the sensitivity function, the phase and/or gain margin has to increase. Re-tuning the controller essentially means, re-configuring the transfer function of the controller to achieve these new desired margins.

In figure D.6b, it can be seen that the magnitude of the sensitivity function is greater then one around the bandwidth. Reducing the sensitivity's magnitude in this area, will thus imply a reduction of the area above the zero dB line. The area under the zero dB line is equal to the area above the zero dB line due to the waterbed effect, see section C.8. This means that the magnitude of the sensitivity function has to increase at low frequencies, when the magnitude around the bandwidth is reduced. It is therefore possible to reduce the impact of the unknown disturbances by re-tuning the controller. A reduction will however, inherently amplify the impact of low frequency disturbances. This means that a performance increase can only be achieved when the impact of the disturbances around bandwidth is further reduced then the impact of the low-frequency disturbances is increased.



(a) The effect of re-tuning the controller on the Nyquist plot.

(b) The effect re-tuning the controller on the sensitivity function.

Figure D.6: Visualization of the impact of re-tuning the controller for a greater disturbance rejection capability of frequencies around the bandwidth.

D.4.1. $\mathcal{H}_\infty$ Controller Design

A limitation of tuning is the fact that is does not explicitly generate an optimal controller. Parameters can always be adjusted, and additional filters can always be added. This limitation can be overcome by setting up the controller design as an optimisation problem. In this optimisation a cost function is minimized to generate an optimal controller for a given application. The trick is to set up the cost function in such a way that the minimum of the function correctly reflects the predefined definition of optimal performance. A commonly used methodology for setting up a cost function and finding the optimal controller is called $\mathcal{H}_\infty$ optimisation. [10]

The goal of $\mathcal{H}_\infty$ optimisation is to minimise the $\mathcal{H}_\infty$ norm. The $\mathcal{H}_\infty$ norm can be calculated using equation D.4 and can be interpreted as the peak value of a particular transfer function. Multiplying the transfer function with a frequency dependent weight function enables the $\mathcal{H}_\infty$ norm to be interpreted as the maximum distance to a specified upper bound of the transfer function.

$$||G(j\omega)||_\infty \triangleq \underset{\omega}{max}\ \bar{\sigma}(G(j\omega)) \tag{D.4}$$

In order to generate a controller that provides a fast settling time and sufficient disturbance rejection capabilities, both the sensitivity and complementary sensitivity function have to be considered. It is therefore useful to set up the optimisation as a mixed sensitivity problem. In a mixed sensitivity problem, multiple transfer functions are considered and the $\mathcal{H}_\infty$ can be regarded as the maximum vector length of a vector containing the used transfer functions. The optimisation can be written as presented in equation D.5 and can be calculated using mathematical software packages such as MATLAB [13]. It should be noted that the restriction imposed by the waterbed effect, explained in appendix D.4 still apply. Reducing the magnitude of the sensitivity function around the bandwidth will always cause the magnitude of the sensitivity function to increase at lower frequencies.

$$||\mathbf{G}(j\omega)||_\infty \triangleq \underset{\omega}{max}\ \bar{\sigma}\begin{bmatrix}\omega_1 S\\ \omega_2 T\end{bmatrix} \tag{D.5}$$

D.5. Evaluation Linear Control Strategies

In section 3.2 it is shown that the non-linearity in the spring is by far the biggest contributor to the tracking error. Since this non-linearity has such a big impact and is known, a modification targeting the reduction of this non-linearity has to be included. As explained in section D.2 using the PDSFC, the system is effectively linearized, making it a highly suitable modification to implement.

The tracking error caused by changes in the surface height can be reduced by both a feedforward controller and an anti-notch filter. Applying an anti-notch filter mainly works when the surface change disturbance has a clear correlation with a specific frequency range. This means that it works well for a tilted flat, but makes it less applicable for measuring arbitrary surfaces. Feedforward control does not have this issue and is only dependent on the accuracy of the modelled plant inverse. The plant inverse is especially inaccurate at high frequencies as additional poles are required to make the transfer function proper. These poles cause a feedforward controller to be less effective when the change in surface height has a lot of high frequency components.

When a disturbance is not known beforehand a feedback control strategy needs to be applied. This effectively comes down to changing the transfer function of the controller to generate the shape of the sensitivity function. Here it is important to note that , due to the waterbed effect, an improvement of the control system in one area will always lead to a reduction in performance in another. Therefore, while it is possible to reduce the impact of disturbances in a specific frequency band by re-tuning the controller, an overall performance increase is not a given.

E

Reset Control

Even the best linear control techniques are limited in their performance by the waterbed effect and Bode's gain-phase relation, explained in section C.8. To overcome these limits, non-linear control needs to be implemented. A promising non-linear control strategy is reset control.

Reset control works very similar to linear control. The controllers can be tuned using conventional loop-shaping techniques and many linear control elements have a resetting counter part. The difference is that at some point one of the controller states is re-set when a pre-determined condition is met [9]. This resetting action means that the controller is non-linear and enables the creation of control elements that overcome the limits of linear control. This appendix provides the required tools and background information to design and evaluate the CGLPs presented in section 4.3.

E.1. Reset Control in State Space

When modelling reset control systems it is often useful to describe the system in a state-space representation, as shown in E.1. In the state-space representation, the reset mechanism is described by equation two, where $x_r(t^+)$ and A_ρ represent the state of the controller after the reset and the reset matrix respectively. When the reset condition is not triggered, the system functions as a standard linear systems, as described by equation one and three. This system of equations is often referred to as the base linear system (BLS). [9]

$$\begin{cases} (1) \;\; \dot{x}_r(t) = A_r x_r(t) + B_r(t) & c(t) \neq 0 \\ (2) \;\; x_r(t^+) = A_\rho x_r(t) & c(t) = 0 \\ (3) \;\; u(t) = C_r x_r(t) + D_r e(t) & \end{cases} \tag{E.1}$$

Designing a reset mechanism generally comes down to determining when, by modifying $c(t)$, and how to apply the reset, by modifying A_ρ. Usually $c(t)$ is equal to the tracking error, but it is good to be aware that other signals are possible as well. The reset matrix defines the value of the controller states after the reset. This matrix usually sets the states to zero. In some cases however, it can be useful to only partially reset the controller state.

E.2. Describing Functions

When dealing with non-linear controllers, it is impossible to perfectly describe the behaviour of that controller with a linear transfer function. To still get an indication of the frequency domain representation of a reset controller, it is possible to approximate its behaviour using a sinusoidal input Describing Function (DF). A DF represents the transfer function between the input and the first harmonic of the Fourier series expansion of the output. The DF can be described analytically using equation E.2, the derivation of this equation is presented in [14].

$$H(\omega) = C_r(j\omega I - A_r)^{-1}(I + j\Theta_D(\omega))B_r + D_r \tag{E.2}$$

Where,

$$\begin{aligned}
&\Lambda(w) = \omega^2 I + A_r^2 \\
&\Delta(\omega) = I + e^{\frac{\pi}{w}A_r} \\
&\Delta_r(\omega) = I + A_\rho e^{\frac{\pi}{w}A_r} \\
&\Gamma_r = \Delta_r^{-1}(\omega)A_\rho\Delta(\omega)\Lambda^{-1}(\omega) \\
&\Theta_D(\omega) = \frac{-2\omega^2}{\pi}\Delta(\omega)[\Gamma_r - \Lambda^{-1}(w)]
\end{aligned}$$

Describing functions only provide a valid approximation of the controller behaviour when the first harmonic is dominant over the other harmonics in the Fourier series expansion. To generate greater insight into the behaviour of the controller, the higher-order sinusoidal-input describing functions (HOSIDF) method can be used [15]. The HOSIDF method can be regarded as an extension of the DF analysis. Where, in addition to examining the transfer function between the input and the first harmonic of the output, calculations are extended to include the transfer functions between the input and higher-order harmonics. The HOSIDF can be calculated as outlined in E.3.

$$H_n(w) = \begin{cases} C_r(j\omega I - A_r)^{-1}(I + j\Theta_D(\omega))B_r + D_r & n = 1 \\ C_r(j\omega I - A_r)^{-1}(j\Theta_D(\omega))B_r & odd \;\; n > 2 \\ 0 & even \;\; n \geq 2 \end{cases} \tag{E.3}$$

E.3. The Clegg Integrator

Using reset control it is possible to overcome the limits of linear control and increase the performance of a system. This potential performance increase can intuitively be understood by looking at the integral action of a standard PID controller. As explained in C.6, the integral action is determined by the sum of the error over time. This means that when the error is zero, the controller still outputs a signal, when the sum of the error is not zero. This output causes the unwanted overshoot in the step response and can be eliminated by resetting the error sum, when the error crosses zero. This intuition is the idea behind the Clegg integrator (CI) [16]. A CI is an integrator that is reset to zero when the error is zero. The response of a CI to a sinusoidal input is presented in figure E.2.

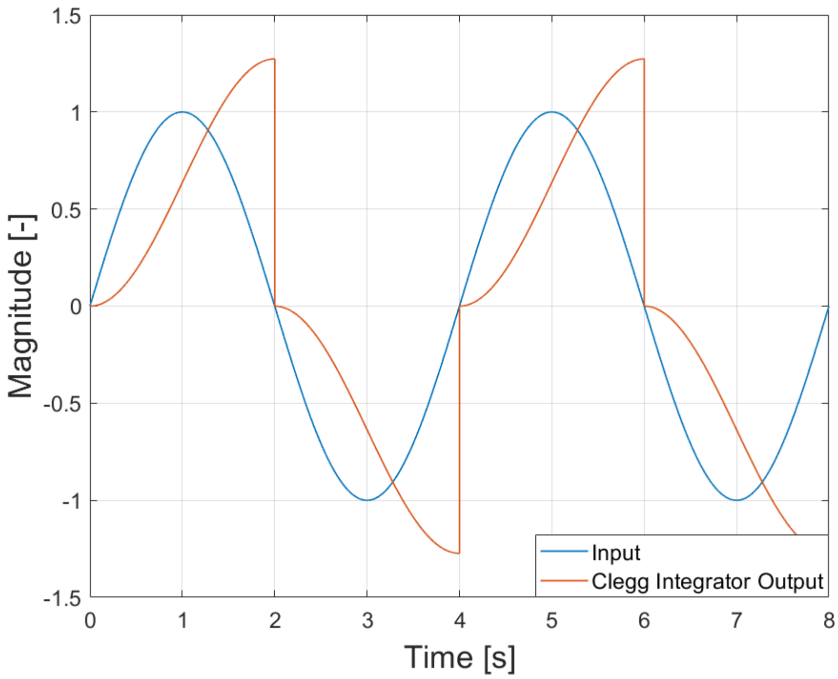


Figure E.1: Output of a CI to a sinusoidal input.

A block diagram representation of the CI is shown in figure E.1. For the CI, $c(t)$ denotes the tracking error. The after-reset value, $a(t)$, is equal to zero. The CI can be regarded as the basis of reset control. Every reset element presented in section E.4 can be described in a block diagram representation based of this block. The state-space matrices for the CI are given by E.4.

$$A_r = 0 \;\; B_r = 1 \;\; A_\rho = 0 \;\; C_r = 1 \;\; D_r = 0 \tag{E.4}$$

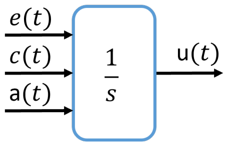


Figure E.2: Block diagram representation of a CI.

Entering the state-space matrices of the CI in equation E.2, results in the DF of the CI presented in equation E.5. The DF is the same as the transfer function of a conventional integrator, with an additional term, $j\frac{4}{\pi}$. This additional term causes the phase of the CI to be $tan^{-1}(\frac{\pi}{4}) \approx 38°$. This means that a CI adds $52°$ compared to a conventional integrator.

$$H_{Clegg}(w) = \frac{1}{j\omega}(1 + j\frac{4}{\pi}) \tag{E.5}$$

In figure E.3, a DF approximation of the CI and a conventional integrator are presented in a Bode plot. In the figure it can be seen that the integrators have the same slope, but a different phase. From this figure it can therefore be concluded that the DF approximation of the CI breaks Bode's gain-phase relation. The small difference in the magnitude can easily be compensated through a decrease in the gain of the CI. When only considering the DF approximation, the CI can be regarded as a integrator with a smaller phase lag.

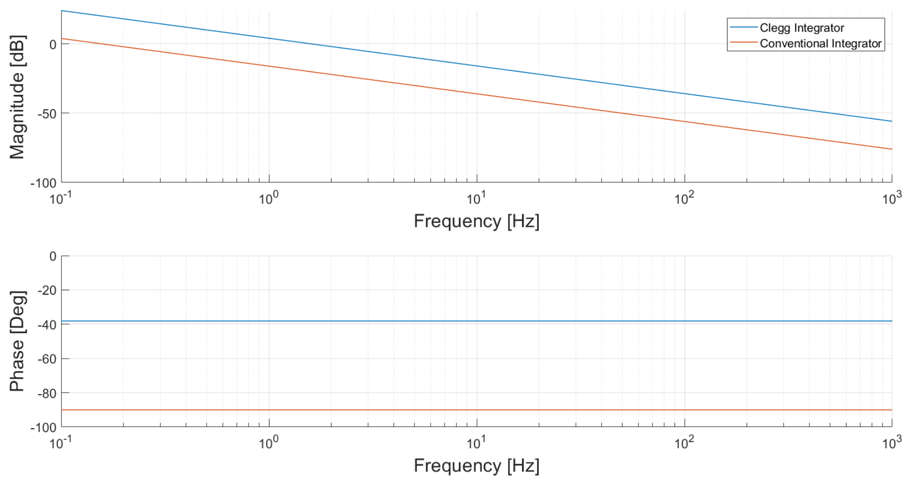


Figure E.3: Comparison of a Clegg and conventional integrator.

E.4. Reset Elements

From section E.3 it can be concluded that when only considering the DF approximation, the CI adds phase to a system without changing the gain. A big downside of the CI is the fact that the reset action introduces limit cycles into the system when the BLS has a steady state-error. The limit cycles can significantly increase the tracking error [15]. To limit this behaviour and enhance the overall performance of the CI, various reset elements have been developed. In this section a few of the most frequently used reset elements are discussed.

E.4.1. PI + CI

In a conventional CI, the summed error is reset to zero when the reset condition is met. This is referred to in literature as a full reset. Instead of resetting the error sum to zero it is also possible to reset the sum to a percentage of its current value. This so called 'partial reset' decreases the limit cycle behaviour caused by the CI and is the idea behind the PI + CI element [17].

E.4.2. Generalized FORE

Besides an integrator, other filters can also have a resetting counterpart. One commonly used element is the so called First Order Reset Element (FORE) [18]. The FORE is essentially a resetting first order filter. Similar as with the CI it is also possible to partially reset a FORE. In literature this type of filter is classified as a generalized FORE (GFORE) element [15]. The state-space matrices of a GFORE element are presented in E.6.

$$A_r = -\omega_r \quad B_r = \omega_r \quad A_\rho = \gamma \quad C_r = 1 \quad D_r = 0 \tag{E.6}$$

E.4.3. SORE

When loop shaping, second order filters such as notches and low-pass filters are often used. The resetting equivalent of these filters are labelled Second Order Reset Element (SORE) [19]. The SORE can be described in state-space as presented in E.7.

$$A_r = \begin{bmatrix} 0 & 0 \\ -\omega^2 & -2\omega\beta \end{bmatrix} \quad B_r = \begin{bmatrix} 0 \\ \omega^2 \end{bmatrix} \quad A_\rho = \mathbf{0} \quad C_r = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad D_r = \mathbf{0} \tag{E.7}$$

E.5. Evaluation of using Reset Control

When only considering the DF approximations of the reset control elements presented in E.4 a clear increase in performance can be expected compared to their linear counter parts. In [15], it is also stated that reset controllers have shown a significant performance improvement over linear controllers. It should be strongly noted however, that the DF analysis only gives an approximation of the performance.

Figure E.4 shows a few HOSIDFs of the Clegg integrator, in the figure it can be seen that even though the first harmonic has the largest magnitude, it is not significantly higher than the third harmonic. According to [15], this means that the higher order harmonics can not be completely neglected. If only the first order harmonic is considered when tuning a reset controller, a significant deviation can occur between expected and actual performance. In short, reset elements can provide a significant performance increase, but the effects of higher order dynamics should always be considered when tuning the controller.

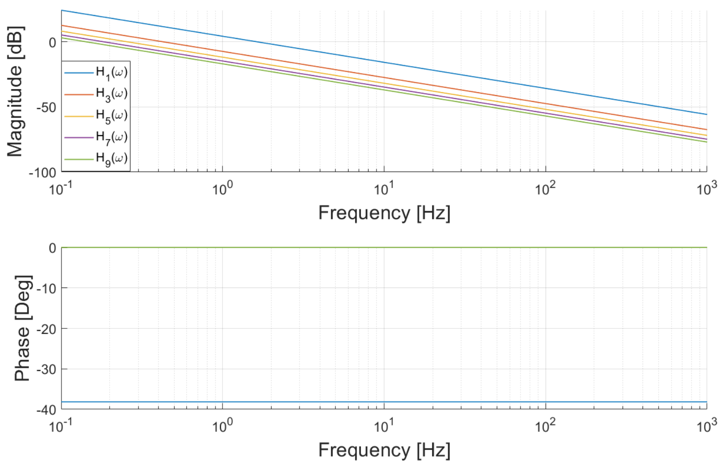


Figure E.4: HOSIDFs of the Clegg integrator. In H_n, n denotes the n^{th} harmonic.

E.6. CGLP

The filters presented in section E.4, mainly focus on reducing the phase lag of an integrator. It is also possible to use reset control to create a broadband phase lead. This is the motivation behind the Constant in Gain lead in phase (CGLP) element introduced in [20]. A CGLP element consists of a reset lag filter in series with an ordinary lead filter, which can both be of first or second order. The reset filter mirrors the magnitude of the lead filter, but has a different phase. When putting these elements together an element with a constant gain and a phase lead is created, hence the name. A bode plot of the DF of a first order CGLP is shown in figure E.5. The state-space matrices are given in E.9. In the equations, α refers to the correction required to match the magnitudes of the FORE and the lead filter.

In section 4.3 is shown that the magnitude of the system's sensitivity function needs to be reduced around the bandwidth. Section D.4, shows that achieving this task through linear control will always lead to an increase of the sensitivity's magnitude at low frequencies. Using a CGLP however, this is not the case as it only changes the phase of a system and not the magnitude. This property makes a CGLP highly suitable for the application and is the reason why it is implemented.

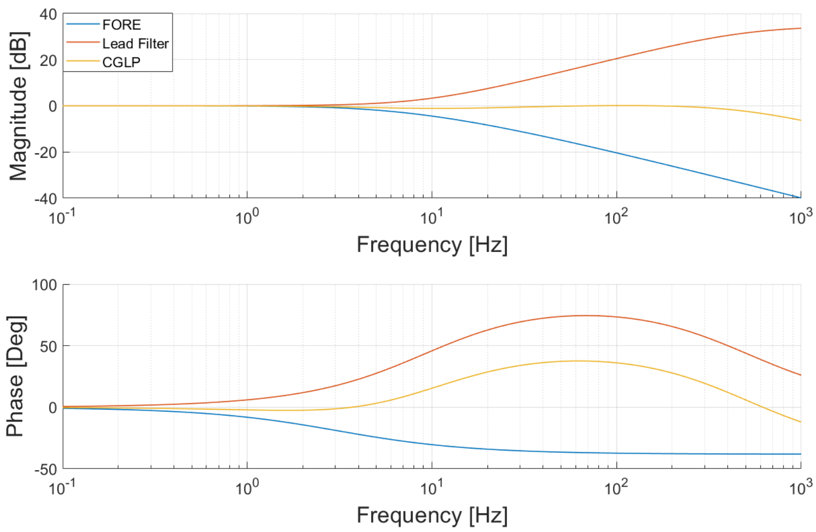


Figure E.5: Bode plot of a CGLP.

$$A_r = \begin{bmatrix} -\omega_r\alpha & 0 \\ \omega_f & \omega_f \end{bmatrix} \quad B_r = \begin{bmatrix} -\omega_r\alpha \\ 0 \end{bmatrix} \quad A_\rho = \begin{bmatrix} \gamma & 0 \\ 0 & 1 \end{bmatrix} \quad C_r = \begin{bmatrix} \frac{\omega_f}{\omega_r} & (1-\frac{\omega_f}{\omega_r}) \end{bmatrix} \quad D_r = \mathbf{0} \tag{E.8}$$

E.6.1. CGLP Controller Design

In section E.6 it is shown that a CGLP is essentially a series combination of a FORE and a lead filter. These two components can be represented by equation E.9. In the equation, the diagonal arrow indicates the state of the FORE is re-set to γ when the reset condition is met. Designing a CGLP, effectively comes down to choosing the parameters γ, α, ω_r and ω_f to achieve the desired performance.

$$FORE = \frac{1}{\frac{\alpha s}{\omega_r} + 1 \nearrow^{\gamma}} \qquad Lead = \frac{\frac{s}{\omega_r}+1}{\frac{s}{\omega_f}+1} \tag{E.9}$$

The DF approximation of a reset controller is only valid, when the first harmonic dominates over the higher-order harmonics. When designing the CGLP, the goal should therefore be to design a controller that meets the set criteria, with the lowest possible magnitude of the higher-order harmonics. In this design it is especially important that the higher-order harmonics are dampened at the lower frequencies, below bandwidth, as these are the frequencies determining the tracking capabilities of a system. To quantify these higher-order harmonics, [21] uses equation E.10 to approximate the HOSIDF presented in section E.2 at low frequencies. In the equation lf refers to low-frequency.

$$H_n(w,n)_{lf} \approx \begin{cases} -CA^{-1}B+D & n=1 \\ j\frac{-2\omega^2}{\pi}(1-\gamma)CA^{-3} & odd \;\; n>1 \\ 0 & even \;\; n>1 \end{cases} \tag{E.10}$$

Inserting the state-space matrices of the FORE, see section 4, into equation E.10, an approximation of the HOSIDF for a FORE is derived [21].

$$|FORE(\omega,n)_{lf}|_{odd\ n>1} \approx \frac{2(1-\gamma)}{\pi}\frac{\omega^2}{(\alpha\omega_r)^2} \tag{E.11}$$

Equation E.11 shows that the magnitude of the HOSIDF is proportional to $\sigma = \frac{1-\gamma}{(\alpha\omega_r)^2}$. [21] also shows, α has to follow equation E.12, to achieve unity gain at high frequencies. The magnitude of the higher-order harmonics, is therefore only dependent on γ and ω_r. To determine the exact values of the parameters, the MATLAB script presented in appendix G.3 used. The script essentially uses brute force to find the combination of γ and ω_r that results in the desired amount of additional phase, while having the lowest possible value of σ.

$$\alpha = \frac{1}{\sqrt{1+F^2}}, \qquad F = \frac{4}{\pi}\frac{1-\gamma}{1+\gamma} \tag{E.12}$$

Setting ω_f higher, allows ω_r to be bigger, while achieving the same amount of additional phase at bandwidth. When ω_r is bigger, σ is smaller. To minimize the higher-order harmonics, ω_f is therefore set at the Nyquist frequency.

References

[1] Marion Agoyan et al. “Confocal chromatic sensor for displacement monitoring in research reactor”. In: *EPJ Web of Conferences* 253 (Jan. 2021), p. 04021. DOI: 10.1051/epjconf/202125304021.

[2] Robert Munnig Schmidt et al. *The design of high performance mechatronics: High-tech functionality by multidisciplinary system integration*. Delft University Press, 2020.

[3] Graham C. Goodwin, Stefan F. Graebe, and Mario E. Salgado. *Control System design*. Prentice-Hall, 2001.

[4] Mees Vanderbroeck. “Damping Analysis of Reset Control Systems: an Analytical Approach”. Master’s Thesis. Delft Universtity of Technology, 2022.

[5] Normey-Rico J E and Camacho E. F. *Control of dead-time processes*. Springer, 2007.

[6] Hassan Hosseinnia and Niranjan Saikumar. “Fractional-order precision motion control for mechatronic applications”. In: Feb. 2019, pp. 339–356. ISBN: 9783110571745. DOI: 10.1515/9783110571745-015.

[7] Karl Johan Åström and Richard M. Murray. *Feedback systems: An introduction for scientists and Engineers*. Princton university press, 2021.

[8] H. W. Bode. *Network analysis and feedback amplifier design*. D. Van Nostrand Company, 1945.

[9] Alfonso Baños and Antonio Barreiro. *Reset Control Systems*. Springer London, 2012.

[10] Sigurd Skogestad. *Multivariable Feedback Control: Analysis and Design*. Wiley India, 2014.

[11] Jan van Eijk Leon Jabben. “Performance analysis and design of mechatronic systems”. In: *Mikroniek* 2 (2011), pp. 5–12.

[12] Wouter Monkhorst. “Dynamic Error Budgeting a design approach”. 2004.

[13] D. Gu, P. H. Petkov, and M. M. Konstantinov. *Robust Control Design with MATLAB®*. Springer, 2013.

[14] Yuqian Guo, Lihua Xie, and Youyi Wang. *Analysis and design of Reset Control Systems*. The Institution of Engineering and Technology, 2016, pp. 27–41. ISBN: 978-1-84919-703-8.

[15] Niranjan Saikumar, Kars Heinen, and S. Hassan HosseinNia. “Loop-shaping for reset control systems: A higher-order sinusoidal-input describing functions approach”. In: *Control Engineering Practice* 111 (2021), p. 104808. ISSN: 0967-0661. DOI: https://doi.org/10.1016/j.conengprac.2021.104808.

[16] J. C. Clegg. “A nonlinear integrator for servomechanisms”. In: *Transactions of the American Institute of Electrical Engineers, Part II: Applications and Industry* 77 (1958), pp. 41–42.

[17] Alfonso Baños and Angel Vidal. “Definition and tuning of a PI+CI reset controller”. In: *2007 European Control Conference (ECC)*. 2007, pp. 4792–4798. DOI: 10.23919/ECC.2007.7068602.

[18] Isaas Horowitz and Patrick Rosenbaum. “Non-linear design for cost of feedback reduction in systems with large parameter uncertainty”. In: *International Journal of Control* 21.6 (1975), pp. 977–1001. DOI: 10.1080/00207177508922051.

[19] Leroy Hazeleger, Marcel Heertjes, and Henk Nijmeijer. “Second-order reset elements for stage control design”. In: *2016 American Control Conference (ACC)*. 2016, pp. 2643–2648. DOI: 10.1109/ACC.2016.7525315.

[20] Niranjan Saikumar, Rahul Kumar Sinha, and S. Hassan HosseinNia. ““Constant in Gain Lead in Phase” Element– Application in Precision Motion Control”. In: *IEEE/ASME Transactions on Mechatronics* 24.3 (2019), pp. 1176–1185. DOI: 10.1109/TMECH.2019.2909082.

[21] Mahmoud Shirdast Bahnamiri et al. “Tuning of CgLp based reset controllers: Application in precision positioning systems”. In: *IFAC-PapersOnLine* 53.2 (2020). 21st IFAC World Congress, pp. 8997–9004. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2020.12.2017.

F

Simulink Model

The Simulink model used to generate the modelled results shown throughout the thesis is presented in this appendix. The model conists of three main parts. First an initialization script is called to set up all the parameters of the model, see appendix G.1. Thereafter, a numerical simulation is ran in Simulink, following the block diagrams presented throughout the paper. Lastly a data processing script is called to evaluate and visualize the results, see appendix G.2. The model uses logs of the radius R and rotation angle θ, collected during a tilted flat measurement, as inputs for the surface change disturbance model.

F.1. Set-up without controller Modifications

The initial control system presented in 2.2 is implemented in Simulink as shown in figure F.1. The change in surface height is added to the system through a function block containing the code snippet presented in F.1. Additionally, the z^{-d} blocks represent the sensor delays. In this initial set-up the spring force compensation is executed through a simple gain block denoted by the triangle containing '-K-'.

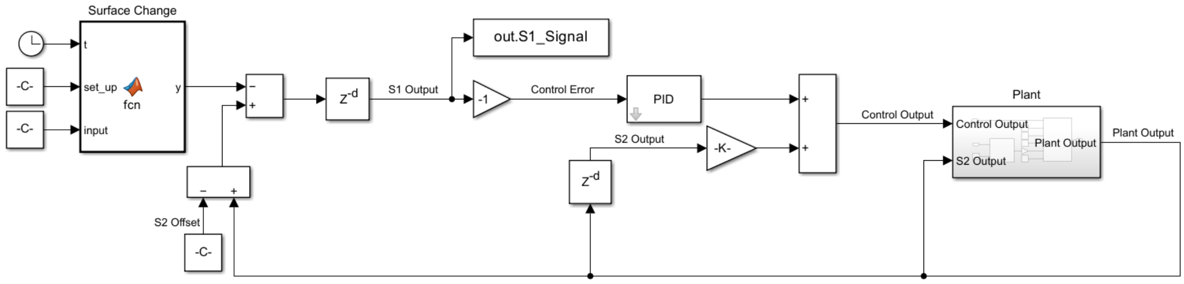


Figure F.1: Simulink model of the system without controller modifications.

```
function z = Surface_Change(t, set_up, input)
    tilt = set_up(1);             % Tilt angle of the tilted flat [rad]
    Ts = set_up(2);               % Sampling time [s]
    i = int64(t/Ts);              % Time stamp [-]
    R = input(i + 1, 1);          % Logged radius at time stamp i [m]
    theta = input(i + 1, 2);      % Logged theta at time stamp i [rad]

    % Radius in the reference frame of the flat
    R_flat = sqrt(R.^2./(sin(theta).^2 + (cos(tilt)*cos(theta)).^2));

    % Surface height at time stamp i
    z = -R_flat.*cos(theta)*sin(tilt);
end
```

Function F.1: Function defining the change in surface height.

The content of the 'Plant' block is presented in figure F.2. As explained in section 3.2, the spring constant changes depending on the position of the system. To incorporate this in the model, Simulink's 'varying transfer function' block is used. The varying spring constant is specified through a function block of the same name. This function block contains the code snippet presented in F.2. The function block essentially specifies the spring constant based off the polynomial fitted in section 3.2.

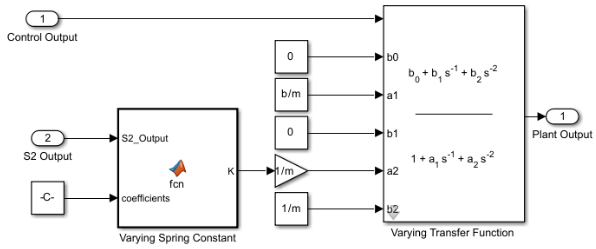


Figure F.2: Plant Block.

```
function K = Varying_Spring_Constant(S2_Output, coefficients)
    K = coefficients(1)*S2_Output.^4 +...
        coefficients(2)*S2_Output.^3 +...
        coefficients(3)*S2_Output.^2 +...
        coefficients(4)*S2_Output +...
        coefficients(5);
end
```

Function F.2: Function defining the varying spring constant.

F.2. Set-up with Controller Modifications

The controller modifications presented throughout the thesis are implemented in Simulink using the model presented in figure F.3. The feedforward block contains the code snippet presented in F.3 and follows a similar structure as the surface change. The difference is the fact that in the feedforward block, the surface change is multiplied by the inverse of the plant, as explained in section 4.2. Figure F.3 also shows the PDSFC block. This block contains the code snippet presented in F.4.

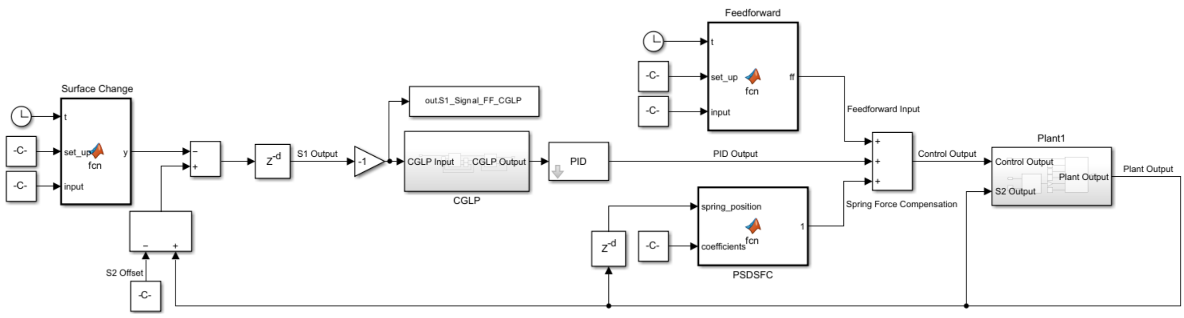


Figure F.3: Simulink model of the system with controller modifications.

```
function ff = Feedforward(t, set_up, input)
    tilt = set_up(1);               % Tilt angle [rad]
    Ts = set_up(2);                 % Sampling time [s]
    offset = set_up(3);             % FF phase offset [rad]
    gain = set_up(4);               % FF gain [-]
    i = int64(t/Ts);                % Time stamp [-]

    R = input(i + 1, 1);            % Logged radius at time stamp i [m]
    theta = input(i + 1, 2) + offset;   % Logged theta at time stamp i [rad]
```

```
    % Radius in the reference frame of the flat
    R_flat = sqrt(R.^2./(sin(theta).^2 + (cos(tilt)*cos(theta)).^2));

    % Surface height at time stamp i
    z = -R_flat.*cos(theta)*sin(tilt);

    % Feedforward gain at time stamp i
    ff = z*gain;
end
```

Function F.3: Function setting up the feedforward input.

```
function Output = PDSFC(spring_position, coefficients)
    K = coefficients(1)*spring_position^4 +...
        coefficients(2)*spring_position^3 +...
        coefficients(3)*spring_position^2 +...
        coefficients(4)*spring_position +...
        coefficients(5);
    Output = K*spring_position;
end
```

Function F.4: Function setting up the position dependent spring force compensation.

The content of the CGLP block is presented in figure F.4. The CGLP is implemented through two function blocks containing the code snippets presented in F.5 and F.6. The function blocks essentially execute the state-space equations of both the FORE and lead element in discrete time.

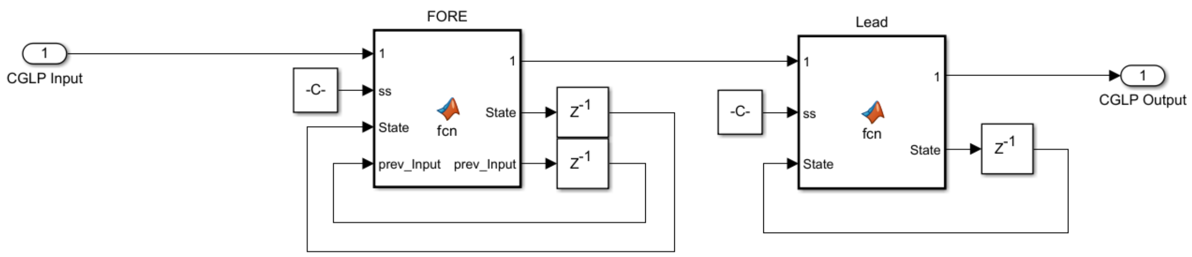


Figure F.4: CGLP Block.

```
function [Output, State, prev_Input] = FORE(Input, ss, State, prev_Input)
    % Retreive the state-space matrices of the FORE element
    A = ss(1);
    B = ss(2);
    C = ss(3);
    D = ss(4);
    Ar = ss(5);

    % Detect a zero crossing
    if (prev_Input > 0 && Input < 0) || (prev_Input < 0 && Input > 0)
        Zero_Cross = true;
    else
        Zero_Cross = false;
    end

    % Determine wheter the system has to reset and calculate the output
    if Zero_Cross == true
        New_State = Ar * State;
        Output = 0;

    else
```

```
        New_State = A * State + B * Input;
        Output = C * State + D * Input;
    end

    State = New_State;
    prev_Input = Input;

end
```

Function F.5: Function defining the FORE block.

```
function [Output, State] = lead(Input, ss, State)
    % Retreive the state-space matrices of the lead element
    A = ss(1);
    B = ss(2);
    C = ss(3);
    D = ss(4);

    % Update the controller state and calcualte the output
    New_State = A * State + B * Input;
    Output = C * State + D * Input;

    State = New_State;
end
```

Function F.6: Function defining the lead block.

G

MATLAB Scripts

The Matlab scripts used to design and evaluate the proposed controller modification are presented in this appendix.

G.1. Simulink Model Initialization

The Simulink model is initialized using the script presented below. The parameters of the mass-spring-damper system and sensor delays defined in the script were fitted using a frequency response function measurement. Additionally, the variable 'coefficients' represents the coefficients of the polyfit retrieved using the 'Spring Constant Fitter' presented in appendix G.4. The parameters of the CGLP are obtained using the 'CGLP Design Tool' presented in appendix G.3.

```
clc; clear; close all;

% Setting up the plant variables
m = 0.07530947;          % Mass [kg]
b = 0.153014;            % Damping [kg/s]
K = 1088.44497480419;    % Stiffness [N/m]

% Define the the coefficients of the polyfit used to simulate
% the position dependent spring constant.
coefficients = [476435757771.356;
                -492876272.681827;
                8733030.26781660;
                -5971.05064248660;
                1088.44497480419];

Ts = (1/2000);           % Sampling time [s]
Rot_v = pi;              % Rotational speed [Rad/s]
Del1 = 'Confidential';   % Delay on the confocal sensor [-]
Del2 = 'Confidential';   % Delay on the interferometer [-]

s = tf('s');
z = tf('z', Ts);

% Set up the spring compensated plant
plant = 1/(m*s^2 + b*s + K);
plantD = c2d(plant, Ts, 'zoh');
KplantD = feedback(plantD, -K*z^(-Del2));

% Evaluate the transfer functions at the rotational frequency
eval_Kplant = evalfr(1/KplantD,exp(Rot_v*1j*Ts));

```

```
% Define the feedforward gain and phase offset
KplantFF_gain = abs(eval_Kplant);
KplantFF_phase = angle(eval_Kplant);

% Set up the PID controller
kp = 2660.689;
I = 200000;
D = 17.0;

% Discrete time PID controller
PID = kp + I*Ts*(z/(z-1)) + (D/Ts)*(1 - 1/z);

% Load the measurement data
folder = 'Specify Folder Here';
path1 = sprintf('%s\\%s.mat', folder, 'time');
path2 = sprintf('%s\\%s.mat', folder, 'R_axis_pos');
path3 = sprintf('%s\\%s.mat', folder, 'Theta');
path4 = sprintf('%s\\%s.mat', folder, 'CF_Output');
path5 = sprintf('%s\\%s.mat', folder, 'IFM_pos');

time_log = (load(path1).time)/1000;        % Time log [s]
R_axis_pos = load(path2).R_axis_pos;       % Radius log [m]
Theta = load(path3).Theta;                 % Theta log [rad]
BaseMeas_CF = -1*load(path4).CF_Output;    % S1 measurement log [m]
BaseMeas_IFM = load(path5).IFM_pos;        % S2 measurement log [m]

tilt = deg2rad(-6.801);         % Tilt angle of the flat [rad]
dur = time_log(end);            % Duration of the simulation [s]

% Set up the FORE in state-space in discrete time
wc = 40*2*pi;
wr =  156.556;
wf =  25.00000 * wc;
gamma =  -0.00949;
F = (4/pi) * (1 - gamma) / (1 + gamma);
alpha = 1 / sqrt(1 + F^2);
w_alpha = wr/alpha;
delta = 0;

% Set up the BLS for the FORE in discrete time
BLS_FORE_C = 1 / ((s / w_alpha) + 1);
BLS_FORE_D = c2d(BLS_FORE_C, Ts,'tustin');

% Retreive the state-space matrices of the FORE
[BLS_FORE_D_num, BLS_FORE_D_den] = tfdata(BLS_FORE_D, 'v');
[A_FORE_D, B_FORE_D, C_FORE_D, D_FORE_D] = tf2ss(BLS_FORE_D_num, BLS_FORE_D_den);
Ar_FORE = gamma;
FORE_D_val = [A_FORE_D, B_FORE_D, C_FORE_D, D_FORE_D, Ar_FORE];

% Set up a lead filter in discrete time
lead_tf = (((s/wr) + 1) / ((s/wf) + 1));
leadD = c2d(lead_tf, Ts, 'tustin');

% Retreive the state-space matrices of the lead filter
[num_lead, den_lead] = tfdata(leadD);
[A_lead_D, B_lead_D, C_lead_D, D_lead_D] = tf2ss(num_lead{1}, den_lead{1});
Lead_D_Val = [A_lead_D, B_lead_D, C_lead_D, D_lead_D];
```

G.2. Simulink Data processing

The output of the Simulink model is processed using the script presented below. The spiral traces and CPS plots presented troughout the thesis are all generated using this script.

```
close all;

% Retreive the position of the sensor
x = R_axis_pos.*cos(Theta)*1e3;
y = R_axis_pos.*sin(Theta)*1e3;

% Retrieve the Simulink Data
Model_Time = out.tout;
S1_Signal = out.S1_Signal.Data;
CF_Signal_VKPlantFF = out.CF_Signal_VKPlantFF.Data;
S1_Signal_FF_CGLP = out.S1_Signal_FF_CGLP.Data;

% Isolate the spiral trace
indices = find(R_axis_pos > 0.0 & R_axis_pos < 0.0150);

% Calculate the CPS of the retreived signals
[freqs_BaseMeas, CPS_BaseMeas] = CPS_Calc(BaseMeas_CF(indices), Ts);
[freqs_S1_Signal_FF_CGLP, CPS_S1_Signal_FF_CGLP] = ...
    CPS_Calc(S1_Signal_FF_CGLP(indices), Ts);

% Plot the Spiral trace
figure(1)
limz = 2;
fontsize = 18;
set(gcf, 'Position', [100, 100, 800, 600]);
set(gca,'FontSize', fontsize);

% Plot the spiral trace of the measurement
color_line3(x(indices), y(indices), BaseMeas_CF(indices)*1e6, ...
            BaseMeas_CF(indices)*1e6);

% Plot the spiral trace of the model without modifications
color_line3(x(indices), y(indices), S1_Signal(indices)*1e6, ...
            S1_Signal(indices)*1e6);

% Plot the spiral trace of the model with modifications
color_line3(x(indices), y(indices), S1_Signal_FF_CGLP(indices)*1e6, ...
            S1_Signal_FF_CGLP(indices)*1e6);
xlabel('X [mm]'); ylabel('Y [mm]'); zlabel('S_1 Output [\mum]');
xlim([-15,15]), ylim([-15,15]); zlim([-limz,limz]);
grid on; view(35, 33); title('');
colormap('jet'); clim([-limz,limz]);

% Plot the CPS of the retreived signals
figure(2)
set(gcf, 'Position', [100, 100, 800, 600]);
semilogx(freqs_BaseMeas, CPS_BaseMeas*1e12); hold on;
semilogx(freqs_S1_Signal_FF_CGLP, CPS_S1_Signal_FF_CGLP*1e12); hold on;
xlabel('Frequency [Hz]'); ylabel('||S_1 Output||^{2}_{RMS} [\mum^2]');
legend('Measured S_1 Output', 'Modelled S_1 Output', ...
       'Modelled S_1 Output with PDSC', 'Location', 'northwest');
grid on; title('');
set(gca,'FontSize', fontsize);
set(findall(gcf,'type','line'),'linewidth',1.5);
```

G.3. CGLP Design Tool

The MATLAB script presented below is used to determine the parameters of a CGLP. The script takes in the bandwidth, Nyquist frequency and required additional phase. The script then uses a brute force approach, to calculate the combination of parameters that yields the lowest magnitude of the higher-order harmonics. The magnitude of the higher-order harmonics is calculated based on the method outlined in section E.6.1.

```
clear; clc; close all;

% Specify required additional phase (AP) at bandwidth
AP = 25;

% Specify the bandwidth and nyquist frequency of the system
wc = 40*2*pi;
wf = 1000*2*pi;

% Specify the search area of the optimizer
n = 80;      % Number of gamma values searched
m = 300;     % Numer of wr values searched

% Set up the search area for the gamma values
gamma = linspace(-0.75, 0.75, n);

% Set up empty arrays for the sigma and wr values
sig = zeros(1, n);
Wr_min = zeros(1, n);

% Loop over the gamma values in the search area
for j = 1:n
    F = (4/pi) * (1 - gamma(j)) / (1 + gamma(j));
    alpha = 1 / sqrt(1 + F^2);

    % Caculate the wr that generates the required AP
    Wr_min(j) = Wr_Calculator(alpha, gamma(j), wf, wc, AP, m);

    % Add the corresponding sigma value to the sig array
    sig(j) = (1 - gamma(j)) / (alpha*Wr_min(j))^2;
    fprintf('%.d / %d \n', j, n);
end

% Retrieve the minimum sigma value
[min_value, min_index] = min(sig(:));

% Print the result
wr = Wr_min(min_index);
fprintf('wr =  %.5f * (2*pi);\n', wr/(2*pi));
fprintf('wf =  %.5f * wc;\n', wf/ wc);
fprintf('gamma =  %.5f;\n', gamma(min_index));
fprintf('Sig =  %.5e;\n', min_value);

function Wr_min = Wr_Calculator(alpha, gamma, wf, wc, AP, n)
    % Set up the wr search area
    A = linspace(0, 5, n);

    % Set up an empty array for the phase errors
    PhaseError = zeros(1, n - 1);

    % loop over the wr values in the search area
    for i = 2:length(A)
        wr = (wc/A(i));
```

```

        % Calculate the phase at bandwidth of a CGLP
        % with the given parameters.
        CGLP = CGLP_Setup(alpha, gamma, wr, wf, wc);
        CGLP_Angle = rad2deg(angle(CGLP));

        % Calculate the phase error of the CGLP
        PhaseError(i - 1) = abs(AP - CGLP_Angle);
    end

    % Calculate the minimum phase error
    [min_value, min_index] = min(PhaseError);

    % Calculate the corresponding wr value
    Wr_min = wc / A(min_index);
end


function CGLP_Val = CGLP_Setup(alpha, gamma, wr, wf, wc)
    % Set-up a FORE in state-space
    w_alpha = wr/alpha;
    A_FORE = -w_alpha; B_FORE = w_alpha; C_FORE = 1; D_FORE = 0;
    Ar_FORE = gamma;
    Sys_FORE = ss(A_FORE, B_FORE, C_FORE, D_FORE);
    FORE_Val = hosidfcalc(Sys_FORE, Ar_FORE, 1, wc);

    % Set up a lead filter
    s = tf('s');
    lead_tf = ((s/wr) + 1) / ((s/wf) + 1);

    % Evaluate the lead filter at bandwidth
    eval_lead = evalfr(lead_tf, wc*1j);

    % Set up the CGLP
    CGLP_Val = FORE_Val * eval_lead;
end
```

G.4. Spring Constant Estimator

The function presented below is used to make a poly-metric fit of the position dependent spring constant. The script uses the procedure outlined in section 3.2 to produce the approximation.

```
function Result = Spring_Constant_Estimator(Data, start_time, end_time)
    % Select the time range of an up or down movement
    time_range = (Data.Time >= (Data.Time(1) + start_time)...
                 & (Data.Time <= Data.Time(1) + end_time));

    % Retreive the S2 data
    conversion1 = 'Confidential';
    S2_signal = double(Data.Interferometer_Probe)*conversion1;

    % Set up a time vector
    time = Data.Time(time_range) - Data.Time(1) - start_time;

    % Set up position vector
    S2_pos = S2_signal(time_range);

    % Convert uint16 to doubles
    F_motor = double(Data.Probe_ActualForce_Counts(time_range));
    F_motor(F_motor > 32768) = F_motor(F_motor > 32768)  - 65536;

    % Convert the Force counts to Newtons
    conversion2 = 'Confidential';
    F_motor = F_motor * conversion2;

    % Calculate the offset
    zero_crossings = diff(sign(S2_pos)) ~= 0;

    % Find zero crossings
    zero_index = find(zero_crossings);
    offset = -1*F_motor(zero_index);

    % Delete the elements close to zero for better fit
    F_motor1 = F_motor(abs(S2_pos) > 0.2e-4) + offset(1);
    S2_pos1 = S2_pos(abs(S2_pos) > 0.2e-4);
    spring_constant = (F_motor1)./(S2_pos1);

    % Approximate the spring constant using a polymetric fit
    coefficients = polyfit(S2_pos1, spring_constant, 4);
    x = linspace(-0.004, 0.004, 1000);
    spring_approx = coefficients(1)*x.^4 + ...
                    coefficients(2)*x.^3 + ...
                    coefficients(3)*x.^2 + ...
                    coefficients(4)*x + ...
                    coefficients(5);

    Result = {time,...
              S2_pos,...
              F_motor,...
              S2_pos1,...
              spring_constant,...
              spring_approx,...
              x,...
              coefficients,...
              F_motor1};
end
```

G.5. θ_N calculator

The MATLAB script presented below is used to calculate the angle of the NSV. In the script, G represents the measured frequency response of the plant.

```
function Theta_N = Theta_N_calculator( s, wr, wf, gamma, PID, G)
    % Set-up the lead filter
    lead = ((s/wr) + 1) ./ ((s/wf) + 1);

    % Set-up the FORE
    F = (4/pi) * (1 - gamma) / (1 + gamma);
    alpha = 1 / sqrt(1 + F^2);
    w_alpha = wr/alpha;
    FORE = 1 ./ ((s / w_alpha) + 1);

    C1 = lead;
    C2 = PID;
    C3 = 0;
    R = FORE;
    Dr = 0;

    % Specify the open-loop
    L = C1.*C2.*G;

    % Set up the shaping filter
    Cs = 1;

    % Set up the Nyquist stability vector
    M1 = 1 + L.*(R + C3);
    M2 = L.*Cs.*(R - Dr);
    M3 = (1 + L.*(C3 + Dr)).*(R - Dr);

    Nx = real(conj(M1).*M2);
    Ny = real(conj(M1).*M3);

    % Calculate Theta_N
    Theta_N = rad2deg(atan(Ny ./ Nx));

    % Determine the quadrant and compute the tangent accordingly
    Theta_N(Nx <= 0 & Ny >= 0) = 180 + Theta_N(Nx <= 0 & Ny >= 0);
    Theta_N(Nx <= 0 & Ny <= 0) = 180 + Theta_N(Nx <= 0 & Ny <= 0);

end
```

G.6. Cumulative Power Spectrum Calculator

The MATLAB function presented below is used to calculate the CPS plots shown in the thesis.

```
function [freqs_cps, cps] = CPS_Calc(Data, Ts)
    n = length(Data);               % Number of Data points [-]
    Fs = 1/Ts;                      % Sampling frequency [Hz]

    % Check if the number of data points is even or odd
    if mod(n, 2) == 0
        RFFT = (1: (n / 2));                   % even
    else
        RFFT = ((1: floor(n / 2) + 1));       % odd
    end

    % Set up a frequency vector containing only the positive frequencies
    freqs = RFFT * 1.0 / (n * Ts);

    % Calculate the FFT
    FFT_Data = fft(Data);
    FFT_Data = FFT_Data(RFFT);

    % Caclulate the PSD
    PSD_Data = real(FFT_Data .* conj(FFT_Data)) ./ ((n * Fs) / 2);

    % Set up an empty vector for the CPS values
    cps = zeros(1, length(PSD_Data)  - 1);

    % Calulate the CPS
    integral = 0;
    for i = 2:length(cps)
        step = freqs(i+1) - freqs(i);

        % Trapezoidal integration
        step_int = 0.5 * step * (PSD_Data(i) + PSD_Data(i - 1));
        cps(i) = step_int + integral;
        integral = integral + step_int;
    end
    freqs_cps = freqs(2:end);
end
```

H

Structured Text Implementations

The used tracking system is controlled using a programmable logic controller (PLC) programmed in structured text. The structured text implementations of the presented controller modifications are shown in this appendix.

H.1. Position Dependent Spring Force Compensation

```
// Caculate spring constant corresponding to the current position
spring_approx := coefficients[1]*EXPT(fMeasured_Position_m, 4) +
                 coefficients[2]*EXPT(fMeasured_Position_m, 3) +
                 coefficients[3]*EXPT(fMeasured_Position_m, 2) +
                 coefficients[4]*fMeasured_Position_m +
                 coefficients[5];

// Calculate the stiffness compensation
StiffnessCompensation_N := spring_approx * fMeasured_Position_m;
```

H.2. Feedforward Control Force

```
// Offset theta by the angle of the plant inverse
Theta_rad := Theta_rad + Theta_offset_rad;

// Calculate the surface height of the flat corresponding
// to the current position of the system
SH:= -1 * SQRT( EXPT( R_axis_pos_m, 2 ) / (EXPT( SIN( Theta_rad ), 2 ) +...
         EXPT( COS( Theta_rad ) * COS( Tilt_angle_rad ), 2 )) ) *...
         COS( Theta_rad ) * SIN( Tilt_angle_rad );

// Calculate the feedforward input
FF := SH * FF_Gain;
```

H.3. First Order Lead Filter

```
// Calculate the filter ouput
New_State_lead := A_lead * State_lead + B_lead * input;
output := C_lead * State_lead + D_lead * input;

// Update the state
State_lead := New_State_lead;
```

H.4. First Order Reset Element

```
// Detect a zero-crossing of the input signal
IF (previousInput_FORE > 0 AND input < 0) OR (previousInput_FORE < 0 AND input > 0) THEN
        Zero_Cross := TRUE;
ELSE
        Zero_Cross := FALSE;
END_IF

// Calculate the filter output
IF Zero_Cross = TRUE THEN
        // Re-set the filter
        New_State_FORE := Ar_FORE * State_FORE;
        output := 0;
ELSE
        // Linear operation
        New_State_FORE := A_FORE * State_FORE + B_FORE * input;
        output := C_FORE * State_FORE + D_FORE * input;
END_IF

// Update the variables
State_FORE := New_State_FORE;
previousInput_FORE := input;
```