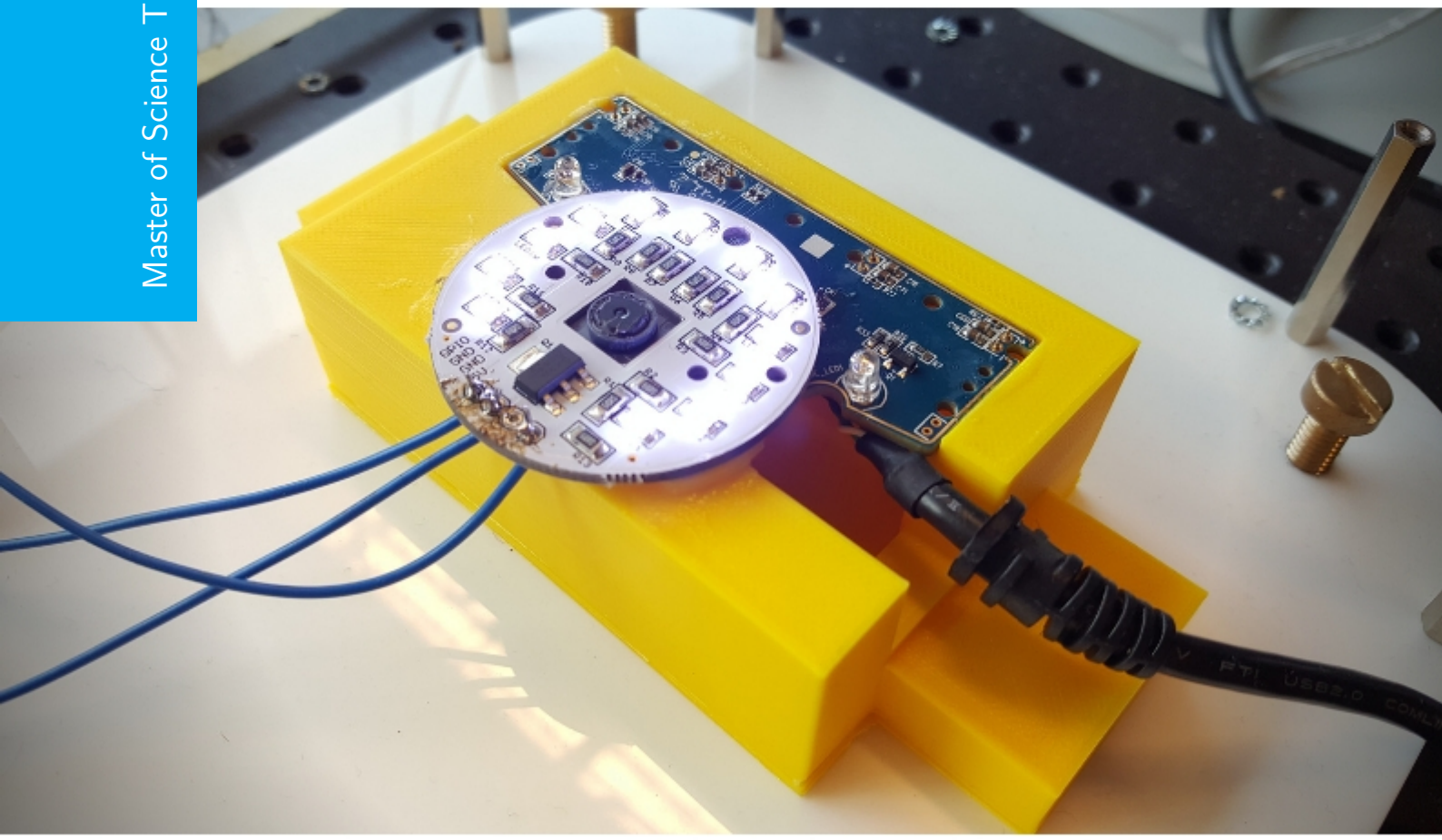
Development and Control of a Low-Cost Precision Positioning System

Exploring the limits of computer vision sensors in precision systems

F.J. Florijn

Master of Science Thesis



TU Delft Delft University of Technology

Delft Center for Systems and Control

Development and Control of a Low-Cost Precision Positioning System

Exploring the limits of computer vision sensors in precision systems

Master of Science Thesis

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

F.J. Florijn

September 17, 2018

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

Abstract

Computer vision has been on the rise over the last decades. The applications of computer vision are mostly found within the domain of robotics and Unmanned Aerial Vehicle's. The use of computer visions in precision industry is new and offers both opportunities and challenges. In high performance precision systems the demands on sensors is set high in terms of precision, delay and sample rate. In [1] a computer vision sensor was developed for the control of a microscopy stage. This development opened a path for computer vision sensors in precision systems. However, the limitations of computer vision sensors are challenging in the domain of control. A computer vision sensor needs to process an image before it can estimate the position of an object. This processing requires time which introduces a time delay into the control system. Due to the nature of the computer vision algorithm and computing devices, the delay in the sensor is time varying. Furthermore, since a controller typically requires a tenfold sample rate of the control bandwidth the limited frame rate of computer vision sensors imposes additional challenges.

In previous work, the delay and limited sample rate of the computer vision sensor limited the control bandwidth of a microscopy to 0.87Hz in translational range. Since high performance precision systems require higher bandwidths a new computer vision sensor is developed in this thesis. An existing microscopy stage is used to test the performance of a computer vision sensor. Ground vibrations in a laboratory environment have a peak value at 10Hz which defines the target bandwidth. In order to reach this target bandwidth a new computer vision sensor is proposed. The sensor uses a cost-effective camera and a computer to run the computer vision algorithm.

By decreasing the delay from 60ms to 10ms a control bandwidth of 7Hz, using PID, is obtained in this work. Since the target bandwidth is 10Hz time delay compensation is considered to improve the performance. By implementing a Filtered Smith Predictor the target bandwidth of 10Hz is reached. The maximum control bandwidth of the controller, 15Hz, was derived in order to inspect the limit of computer vision sensors in precision systems.

Table of Contents

Acknowledgements **xi**

1 Introduction **1**

1-1 Microscopy 1

1-2 State of the Art 1

1-3 Project Objective 3

1-4 System Introduction 3

1-5 Structure 4

2 Paper: Development of a Low-Cost Computer Vision Sensor **5**

3 Discussion **13**

3-1 Actuator System 13

3-2 Sensor System 14

3-3 Control 15

4 Conclusion and Recommendations **17**

4-1 Conclusions 17

4-2 Recommendations 18

4-2-1 Sensor System 18

4-2-2 Control 18

A System Overview **19**

A-1 Delay Overview within the Setup 20

A-1-1 Laboratory Setup 21

B Actuator **23**
B-1 Planar Coil Actuator . . . 23
B-2 Current Control using Pulse Width Modulation (PMW) . . . 25
B-3 Linear Power Amplifiers . . . 26
B-3-1 Linear Power Amplifier Design . . . 26
B-4 Linear Power Amplifiers Laboratory . . . 28
B-4-1 Digital-to-Analogue-Converter . . . 28

C Sensor System **29**
C-1 Marker Identification Using ArUco . . . 30
C-2 Marker map Configuration and Placement . . . 30
C-3 Sensor System Overview . . . 31
C-4 Sensor Resolution and Magnification . . . 32
C-5 Sensor System Results . . . 34
C-5-1 Resolution versus Delay in ArUco . . . 34
C-5-2 Measurement Dispersion at Different Resolutions . . . 34
C-5-3 Delay found in the software pipeline . . . 35
C-5-4 Sample rate measurements using Raspberry Pi . . . 36
C-5-5 Multithreading for increased sample rate . . . 36
C-5-6 Limitation Velocity . . . 37
C-6 Specifications of the Final Sensor System . . . 39

D Control Design **41**
D-1 Time Delays . . . 41
D-2 System Identification . . . 42
D-2-1 Model Identification . . . 42
D-2-2 Signal Reconstruction . . . 43
D-3 PID Benchmark . . . 44
D-3-1 PID synthesis . . . 45
D-3-2 PID Results . . . 46
D-4 Time Delay Compensation . . . 48
D-5 Controller Design and Results . . . 53
D-5-1 Controller Overview . . . 53
D-5-2 Results . . . 56

E Operation Manual **59**
E-1 Sensor Subsystem . . . 59
E-1-1 PlayStation Eye . . . 59
E-1-2 Ubuntu . . . 60
E-1-3 Starting the System . . . 63

F Matlab Code for Filtered Smith Predictor **65**

Bibliography **67**

List of Figures

1-1 Conceptual overview of the microscopy positioning stage. 4

A-1 Overview Stage 19

A-2 Schematic Overview Stage 20

A-3 Top view of the microscopy stage without the mover (left) and the amplifier box which was used as the power supply. 21

A-4 The LTC2668-12 Digital-to-Analogue Converter and the Raspberry Pi 3B. 21

B-1 Schematic overview of the actuator. A single coil consists of two legs through which a current is driven to create a Lorentz force. This configuration of 7 coils and 8 magnets was designed by [1] for an minimum heat production production per unit of force. 24

B-2 Force allocation using the pseudo-inverse $\Phi^{\dagger}$. An example of how the forces are distributed for a 1 Newton force in x-direction when the mover is located at the origin $x = 0, y = 0, \theta = 0$. The seven forces depicted in the figure have a resultant force of 1 N and a torqu of 0 Nm. 25

B-3 Image of the mover (bottom). 16 Magnets are placed in a configuration to create 16 poles. The black lines is the ferrofluid which is gathered at the strongest magnetic fields. At the center the marker map is placed which is used for the computer vision sensor. 26

B-4 Linear Power Amplifier Design where R_1, R_2, R_a and R_s are resistances, V_{in} the input Voltage, Z_l the load impedance which conducts a current I_l. 27

B-5 Amplification rate of the amplifier measured at different set points with a multimeter. 28

C-1 QR-code (left) vs a ArUco marker (right) 29

C-2 A simple illustration of how the ArUco software is able to detect markers in an image [2]. The original image (a) is converted to black white and contours are detected (b). The contours are filtered which result in possible marker candidates (c). From resulting candidates (d) the perspective is removed and divided into grids (e). The grid is used then used to put the image into a binary grid (f) which contain the marker number 31

C-3 Conceptual field of view of the image sensor and an actual image of the image sensor 32
C-4 Schematic overview of the sensor system 32
C-5 Photograph of the image sensor, illumination add-on, printed PCB-holder. 33
C-6 Time delay of the ArUco process at different resolutions of the camera. 34
C-7 Measurement dispersion of the sensor operating at different resolutions of the camera. The standard deviation (σ) of these measurements are 0.39, 0.32 and 0.73μm for 640x480, 320x240 and 180x120 respectively. 35
C-8 Rotational Sensor signal. Arbitrary rotations of the mover and the corresponding sensor signal over time. The signal shows jumps in the value approximately 1 degree (left). Close up of the rotation signal which shows a high resolution for a small rotation. 35
C-9 Delay found within sensor algorithm without the use of multi threading. The time it takes to receive the image from the camera (left), delay caused by ArUco (middle), delay cause by transmitting the position information using UDP (right). 36
C-10 Interval between measurements which is measured via the Raspberry Pi without the use of multi threading. 37
C-11 Frame rate (left axis) and mean time delay of ArUco (right axis) versus the number of threads using a resolution of 320x240 pixels of the camera. 37

D-1 Phase of a pure 0.01 second delay 42
D-2 Bode plot of the frequency response and the estimated model. 43
D-3 Signal reconstruction using First Order Hold and Zero Order Hold 44
D-4 A situation where the computer vision sensor is not able to retrieve the position of the images due to the velocity which causes image blurring. 44
D-5 Bode plots for the system $G(s)$ and the loop gains $L = C(s)G(s)$ for two controller which were design for a control bandwidth of 7 and 10Hz taking into account the maximum delay of 15ms and the average delay of 10ms respectively. 45
D-6 Steps 46
D-7 Steps 46
D-8 Comparison of a step response for a 7Hz and 10Hz controller. Due to the limitation in current it is not possible to further improve the rise time. 47
D-9 Smith Predictor 48
D-10 Smith Predictor 49
D-11 Filtered Smith Predictor response versus a normal Smith Predictor 50
D-12 Response of the FSP with a delay estimate of 10ms. The increasing plant delay (10-15ms) is indicated with a red arrow. 50
D-13 Response of the FSP with a delay estimate of 15ms. The increasing plant delay (10-15ms) is indicated with a red arrow. 51
D-14 Step response of the microscopy with a Filtered Smith Predictor and a close up of the Settling Response. 52
D-15 2-DOF Filtered Smith Predictor 52
D-16 Loop gain C(s)*G(s) which is used to design the controller. Not that the controllers are designed for the non-delayed model. 53
D-17 Closed Loop control sensitivity of the designed controllers. 54
D-18 Step responses of the **10**Hz Filtered Smith Predictor with time delay measurements.. 56

D-19 Step responses for a **15**Hz Filtered Smith Predictor with time delay measurements. 57
D-20 Step responses for a **20**Hz Filtered Smith Predictor with time delay measurements. 57

E-1 Example on how to send '10000000111000000011111010000010' to the SPI interface of the raspberry with Simulink. 62

List of Tables

B-1 OPA549 Specifications 26

C-1 Specifications of the sensor system 39

D-1 Controller Parameters (in radians) for the tested control bandwidths (in Hz) of $C(s) = k_p(1 + \frac{w_i}{s})(\frac{s/w_d+1}{s/w_t+1})(\frac{1}{s/w_f+1})$ 45

E-1 Table with some useful commands which can be used on both Debian (Raspberry Pi) and Ubuntu. 61

Acknowledgements

This work continuous on the work of former TU Delft student Len van Moorsel. With his work he opened up a path for computer visions systems in precision systems. Computer vision is an emerging field in the scientific world and many applications have been and will be using computer vision.
Computer vision is used in everyday technology such as snapchat, traffic cameras and gaming. The idea to extend its use into real-time control of precision systems is ambitious since the challenges are difficult. However, many improvements and developments can be expected of computer vision which will open up paths for even more applications.
I would like to thank my supervisor Hassan HosseinNia for his assistance during my work on this thesis. His expertise in control systems helped to better understand the foundations of my work.
A special thanks goes to Niranjan Saikumar for his help with both the practical aspect and theoretical aspect of my work.
I would like to thank the Mechatronic System Design group for their support during the Monday morning meetings. A special thanks for Jo Spronck for his enthusiastic approach in chairing this meetings.
Rafael Munoz Salinas, the developer of ArUco, thank you for your tips on ArUco and the work you have done.
Tom Hubregsten and Tim Eversdijk for the help with C++ programming.

Delft, University of Technology F.J. Florijn
September 17, 2018

Chapter 1

Introduction

1-1 Microscopy

By placing an object under a microscope the object is magnified such that a small region can be inspected. By magnifying the object the field of view is reduced. In order to move the field of view along the object, commonly a positioning system is used referred to as a positioning stage. By translating the object under the field view the entire objected can be inspected. By means of digital image processing, multiple high resolution images can be stitched together. Using these stitch images, a doctor for example, would be able to inspect the magnified images on a tablet or a computer. Furthermore, by rotating the object while inspecting it under an angle it would possible to construct 3D images and models of the object.

1-2 State of the Art

The most common stages are manually operated by the user although automated microscopes are available in the market. However, the high cost of these stages make them unattractive for commercial use. In order to come up with new affordable technologies for these type of systems, cost-effective sensors and actuators are explored with the Precision and Micro Engineering department of the Delft University of Technology.
In 2014 Max Cafe [3] introduced the first microscopy stage using a planar ferrofluid bearing and a single mover. In his work he presented the advantageous properties of the ferrofluid bearing in a six degrees of freedom stage where the rotation was limited. With the use of laser interferometers he was able to reach control bandwidths of 500Hz. However, the cost of the sensor system alone is estimated to be €30k this stage was not considered cost-effective. Gihin Mok [4] presented his cost-effective stage using an optical mouse sensor in 2015. The €30 sensor was is able to achieve a resolution of up to 3μm keeping the total component cost lower than €200. The stage was able to perform a planar stroke of 10x10mm with a control bandwidth of 10Hz and a small rotation around θ. The performance of the stage was limited due to a large delay (50ms) which was found in the optical mouse sensor.

Haris Habib [5] presented a stage which allowed planar translational range of 9x9mm, controlled at 100Hz, with the use of a position sensitive detector (PSD). A PSD is able to measure the position of a spot light which is projected on a sensitive layer. The rotational range of the mover is 360 degrees which is limited by a cable connection to the power source. The total cost is estimated around €300.
In 2017 Len van Moorsel [1] presented the microscopy stage which he developed during his master thesis project. The microscopy stage was able to achieve translational movements up to 30 mm and an infinite amount of rotation due to the absence of a connection between the moving and static part of the stage. For the actuation of the mover a planar Lorentz force actuator was designed which was optimized to produce a minimal amount of heat production per unit of force. The configuration of the coils and the magnets allows the actuator to produce the force and torque needed for every position of the mover. The position of the mover is detected in 6DOF by a computer vision sensor which is based on the ArUco library [2] although only 3DOF information is used for feedback. Due to the absence of any connection between the sensor and the mover, the mover is able to rotate without limitations which would be can be used for 3D imaging. The use of a computer vision sensors is novel within the precision engineering industry and therefore is an interesting subject for research. Since the cost of the sensor system is estimated to be less than €100 it offers great potential for industry and other applications.
Van Moorsel was able to successfully implement the computer vision sensor in his microscopy stage although mayor challenges were encountered due to the resulting time delay within the feedback loop. The total average delay within the system was identified as 80ms from which 60ms was found in the computer vision algorithm. The 60ms delay resulted in a sample rate of the sensor of only 16Hz which limited the control bandwidth below 1Hz without the use of delay compensation.
The search for a cost-effective sensing technology continuous as all the sensing technologies have their own challenges and limitations in practice.

1-3 Project Objective

Computer vision sensors are widely applied in the area of robotics and UAV's. Computer vision is applied to detect objects and determine their position with respect to a robot. In other applications they are used to determine some global position of a robot or UAV. The sensor requirements for these robotic application differ in terms of sample rate, precision, and delay. However, the fact that it is possible to extract 6DOF information with a single image sensor can reduce the amount of complexity in the development of mechatronic positioning systems. The cost of image sensors and computing devices have been decreasing over the last decades. For this reason the microscopy stage, developed in [1], is subject of this master thesis project. In this project, a new sensor system is developed which further improves the sensing capabilities of a computer vision sensor. Along with a new sensor, this thesis will give an insight into the limitations of computer vision sensors within precision systems. In order to come to a conclusion the project objective and sub-objectives are stated as follows:

Develop a computer vision sensor system for a microscopy positioning stage which allows a minimum control bandwidth of 10Hz
— *What are the physical limitations of a computer vision system?*
— *Which challenges are introduced by a computer vision sensor system in the domain of control.*

1-4 System Introduction

In figure 1-1 the conceptual overview of the microscopy stage is seen. In the top of the figure the microscope is seen which is focused on the object of interest. The mover, consisting of a steel plate with magnets and the ArUco marker map, is able to translate and rotate the object of interest with respect to the microscope. The mover rests upon a ferrofluid bearing which has the advantage of having no stick-slip behaviour. The stage is actuated with a planar Lorentz force actuator using the magnets in the steel plate and coils embedded in a PCB. In [1] challenges were found in the current control using a combination of H-bridges and PWM's. Therefore this combination was replaced by linear power amplifiers. With the use of linear power amplifiers the challenge in the stage were isolated to the challenges found in the sensor. For more information on the amplifiers see appendix B.
A hole in the PCB allows the camera, which is placed under the PCB, to capture images of the target. The target is an ArUco marker map which has shown to be outperforming other marker recognition software ([2], [6]).
The system uses a cost-effective camera, PlayStation Eye, to capture images of the target. The images are processed on a laptop using an Ubuntu operating system. After each image is processed the position information is send to the controller over an internet connection (UDP). The controllers were designed in Simulink and compiled on a Raspberry Pi which can give voltage set points, using an additional DAC, to the linear power amplifiers.

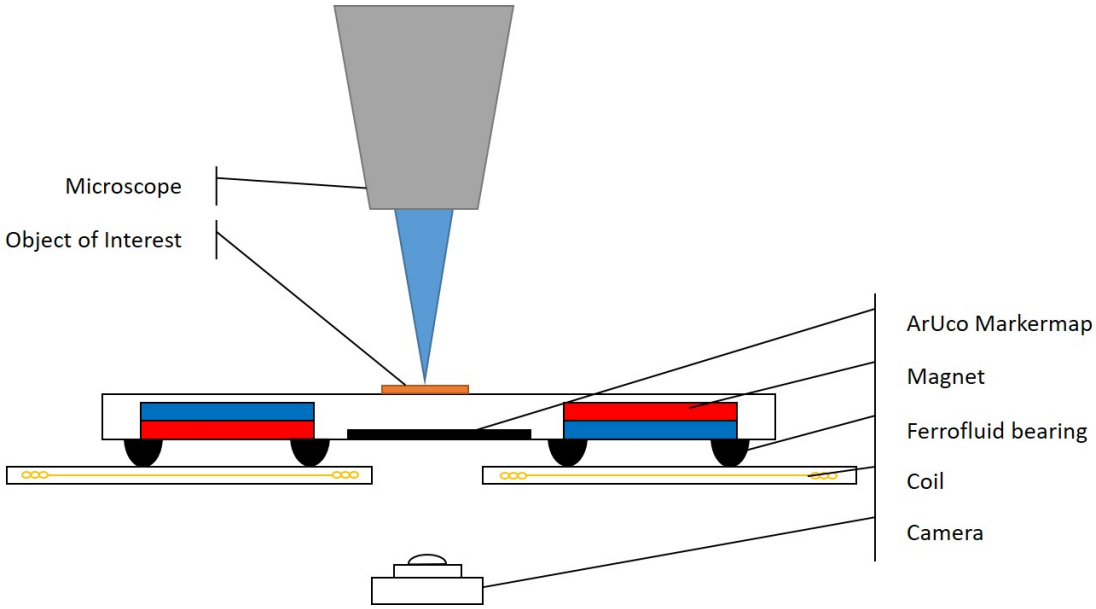


Figure 1-1: Conceptual overview of the microscopy positioning stage.

1-5 Structure

In chapter 2 the work of this thesis is presented in a paper. In chapter 3 the results are further discussed and some more details about the project are given. The conclusions are presented in chapter 4. More background information on the system and the actuator are given in the appendices A and B. More information on the computer vision algorithm, the sensor and the results on the sensor can be found in C. The control design is treated in appendix D and a operation manual of the system with some additional tips is found in appendix E.

Chapter 2

Paper: Development of a Low-Cost Computer Vision Sensor

The scientific contribution of this master thesis project is presented in the form of a paper. The contribution of this work includes the proof of concept of a computer vision sensor for precision applications. An analysis is given on the influential parameters of the computer vision sensor which is useful for future work. Compared to previous result,s a significant improvement of the sensor performance is realized. An additional discussion is presented in the next chapter.

Development of a low-cost micro positioning system using a simple computer vision sensor

Floris Jan Florijn, Niranjan Saikumar and S. Hassan HosseinNia

Abstract— **Precision systems use high performance and expensive sensors. Computer vision can be used for as a cost-effective alternative of these expensive sensors. This paper presents a simple computer vision sensor for a precision positioning system. The developed sensor system is designed for a precision positioning system which requires a 10Hz control bandwidth. Since the use of computer vision leads to time delays in the measurement channel, time delay compensation is applied to improve the control performance. The sensor system developed in this paper shows the feasibility of computer vision sensors for positioning systems up to 15Hz.**

I. INTRODUCTION

The application of computer vision in positioning systems has shown an increase in popularity over the last decade ([1],[2],[3],[4],[5],[6],[7],[8],[9]). Most applications focus on the positioning of unmanned vehicles such as UAVs and robots which do not require high performance sensors. However, in precision mechatronics expensive, high performance sensors are required for feedback control to enable high control bandwidth and precision. The high demands set on the accuracy, precision and sample rates tend to increase the price of sensors. However, some work has been done in the development of low-cost sensors and actuators for precision mechatronics.

In [10] a Position Sensitive Detector was used to sense the position in 3 Degree-Of-Freedom (DOF). The sensor is able to measure the position of light spot which is projected on a light sensitive layer. In [11] an optical mouse sensor was used to measure the displacement in 2 DOF. By comparing the pixel displacement between two subsequent images the sensor is able to the determine the displacement. In [12] a cost-effective computer vision sensor was developed for a positioning stage. The sensor was used to determine the position in 3 DOF (x, y, θ). Fiducial markers, comparable with a simple QR-code, are placed in a map. By making images of the marker map and applying a computer vision algorithm the sensor system is able to determine the position. Computer vision sensor systems have a great potential since they can be used to determine the position in 6 DOF. Delay and sampling rate are the main challenges of computer vision in control. A computer vision algorithm requires time to retrieve information from an image which leads to a time delay. Due to the nature of the algorithm and execution within the computer, the delay is not constant but time varying which introduces additional challenges. Time delays in feedback control are challenging because a time delay introduces additional lag to a system. For high performance precision systems this forms a problem since the additional lag limits the performance in terms of stability and control bandwidth.

The control of time delay systems is a well studied subject ([13],[14],[15]) and many methods are available for the compensation of a delay. The Smith Predictor is a well-known method for the compensation of a constant delay, introduced in [16]. Many modifications on the Smith Predictor exist which try to account for time varying delays. An additional challenge is found in the limited sample rate of computer vision sensors. Due to the shutter time of the camera and the processing of the images the sample rate is limited. The Nyquist theorem states that the sampling frequency must at least be twice the frequency of the frequency dynamics. However, a desired sample rate of control systems is typically a tenfold of the control bandwidth.

Before computer vision sensors can be applied in high performance precision systems, the problems of a time delay and a limited sample rate needs to be solved. By solving these problems computer vision sensors can act as a cost-effective alternative for current sensors.

In this paper a new computer vision sensor system is developed for a positioning system which requires a control bandwidth of 10Hz. The sensor system is analyzed in terms of delay and precision. To account for the tenfold sampling rate of the control system, multi-threading is integrated in the software which doubles the sampling rate to 171. To account for the time varying delay a Filtered Smith Predictor (FSP) is applied [17] and control bandwidths of 15Hz can be reached. By implementing multi threading into the software the sample rate of the sensor system is almost doubled. The implementation of this sensor system on a positioning system illustrates that computer vision sensors can be used for other applications in the same bandwidth region.

This paper is structured as follows: The preliminaries and the problem formulation are given in section 2. The sensor configuration and an analysis in terms of delay and precision are given in section 3. In section 4 the control design and control performance are shown. In section 5 the conclusions and future work are presented.

II. PRELIMINARIES AND PROBLEM FORMULATION

A. Preliminaries

The positioning stage which was developed by [12] is used to analyze the performance of the computer vision sensor developed in this paper. In figure 1 the cross section of the stage is seen. The stage has a translational range of 30 mm in x- and y-direction and an infinite range of rotations around θ_z. The mover consists of a steel plate with 8 embedded magnets, a ferrofluid bearing and a fiducial marker map. The

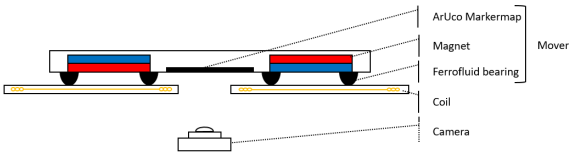


Fig. 1. A cross section of the positioning stage.

ferrofluid is located at the edges of the magnets where the magnetic field is strongest and forms a bearing which has the advantage of having no static friction [19]. The mover rests upon a PCB with a hole in the middle which allows a camera to capture images of the marker map from below.
In figure 2 a schematic top view of the planar motor is depicted. The stator consists of 7 coils which are embedded in a PCB. 8 Magnets are placed in the steel mover and placed on top of the PCB. In figure 2 (a) the actuation principle is seen. By sending a current through a coil a Lorentz force on the coil, and a reaction force on the mover, are created. The two forces of a single coil are combined into one resulting force per coil, see figure 2 (b). By combining the forces a net force or torque can be exerted on the mover.
The resultant force on the mover, created by a single coil, is

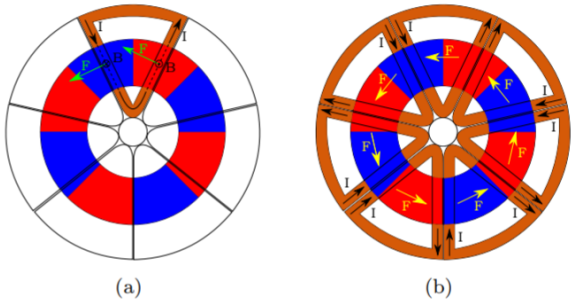


Fig. 2. A schematic overview of the planar axial flux permanent magnet motor.

dependent on the position and orientation of the mover. The motor constants for every position and coil are stored in a position dependent matrix $\Phi(x, y, \theta) \in R^{3\times7}$ which is used by the controller to determine necessary current set points.

B. Sensor Configuration

In this section the practical implementation of the sensor is addressed. In figure 3 a schematic overview of the sensor and actuator is seen. A PlayStation Eye, operating at 187 frames per second (fps), is used as the image sensor. The images are processed on a laptop which sends the position information to the controller which operates on a Raspberry Pi 3B. 7 Linear power amplifiers, with a 2.1 A current limit, are used to control the current in the coils.
In figure 4 an image of the sensor system is seen. The PlayStation Eye casing is removed and placed into a 3D printed holder. A macro lens is placed on the image sensor with an additional illumination add-on.
Open source computer vision software such as ARToolKit,

Fig. 3. Schematic overview of the sensing system and actuator

AprilTag, ARTag and ArUco ([20]),[21]),[22]),[23])) are available for commercial and academic use and all rely on fiducial markers to extract position information. In [24] the benefits of ArUco compared to other algorithms is shown in terms of detection speed and robustness. Multiple markers

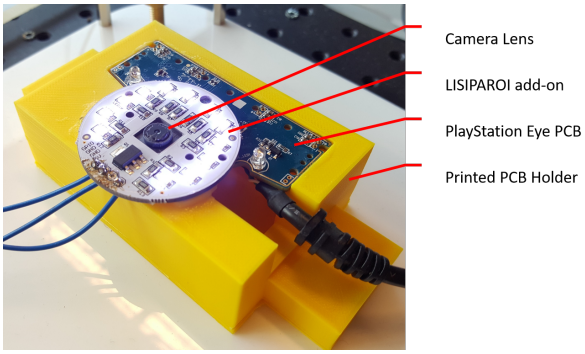


Fig. 4. Configuration of the sensor system.

are placed on a map which is placed on the bottom of the mover. The image sensor is able to compute the position if a single marker is recognized. A comparison of the field of view and an actual image of the image sensor is seen in figure 5.
With an image width of 8.75mm,the length seen by a single pixel is 27μm when the camera is operated at 320x240 pixel resolution. Since the number of markers seen in a single image increases the computation delay, a minimum amount of markers is implemented on the marker map. The width of a square marker is 2.5mm.

C. Problem Formulation

In [12] a delay of 80ms from which 60ms was attributed to the sensor system. The reason for this large delay was attributed to the limited computing power of the Raspberry Pi at which the computer vision algorithm was implemented. The maximum sample rate of the sensor system was limited to 16Hz using the minimum camera resolution of 160x120

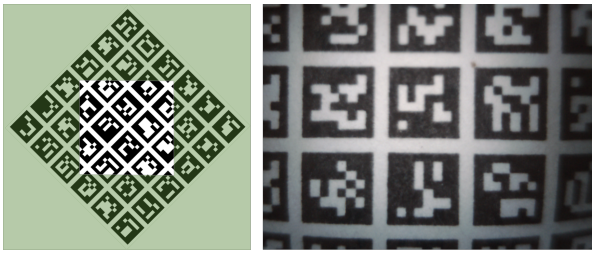


Fig. 5. Conceptual field of view of the marker map and an actual image from the image sensor.

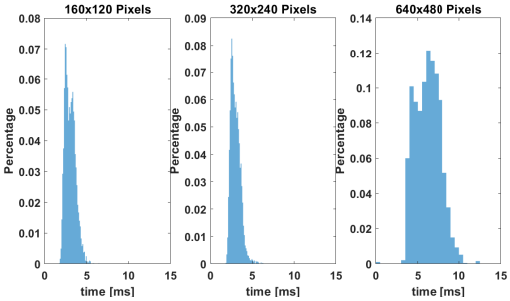


Fig. 6. Time delay results for different camera resolutions

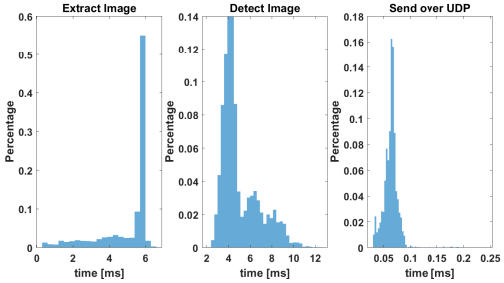


Fig. 7. Time delay results within different processes of the sensor subsystem.

pixels. Due to these limitations in delay and sample rate the control bandwidth was limited to 0.87Hz in translational range without the use of time delay compensation.
To enable a desired bandwidth of 10Hz of the positioning system of 10Hz these problems needs to be addressed. Considering that for most precision system the use of computer is not an issue this paper considers the implementation of the computer vision algorithm on a computer. In order to conclude about the performance of the sensor system this paper addresses the following topics:

1) Development of a cost-effective computer vision sensor with a minimal delay
2) An analysis on the sensor performance in terms of delay
3) The limitations in control without the use of time delay compensation
4) Improving the control performance to the desired bandwidths by applying time delay compensation

By considering these topics this paper will give useful insights in the in the application of computer vision in precision systems.

III. Computer Vision Sensor Configuration and Analysis

In order to improve the sensor system an analysis is needed on the influential parameters which affect the performance. Using a high camera resolution decreases the image length seen by a single pixel which increases the precision of the position estimation. Since the camera resolution affects the delay of the ArUco algorithm, the delay of ArUco against different camera resolutions is seen in figure 6. For the maximum resolution of the camera, 640x480 pixels, the average delay of the algorithm is $6.2ms$. For a resolution of 320x240 pixels the average delay is halved to $3.1ms$. Since the image is segmented in the first step of the algorithm, which is a small part of the algorithm, further decreasing the resolution to 160x120 pixels has no significant effect on the delay ($3.0ms$).

In figure 7 the different delays within sensor subsystem are seen. Extracting an image of the image sensor has an average delay of 5.0ms which is related to the inverse frame rate of the camera sensor $\frac{1}{187} \approx 5$ms. The mean delay of ArUco for these measurements is larger as the delay found in figure 6 since the measurements were obtained in a situation where the mover was moving. It is observed that the communication to the controller, via Internet protocol UDP, has no significant impact on the delay. The average delay of the three processes combined is 10ms.
The increase in delay of a moving target is caused by image blurring. Image blurring affects the quality of an image which results in a larger delay within the computer vision algorithm.

With the reduced delay, the phase lag of the sensor system is significantly reduced by a factor of 6 which allows higher control bandwidth controllers to be designed. However, the effect on the precision must be analyzed in order to inspect the influence of the resolution on the sensor precision.
The measurement dispersion at different resolutions is seen in table I. The measurement dispersion of the largest resolutions are comparable, the smallest resolution has a factor 2 higher measurement dispersion. Without affecting the precision of the computer vision sensor the resolution can be decreased to 320x240 pixels. The issue of the limited sample rate of the sensor remains and is therefore inspected.

Without further decreasing the delay it the sample rate of the sensor is increased by implementing multi threading in

TABLE I
3σ Measurement dispersion at different camera resolutions

Resolution	3σ Dispersion
640x480	1.17 μm
320x240	0.96 μm
160x120	2.19 μm

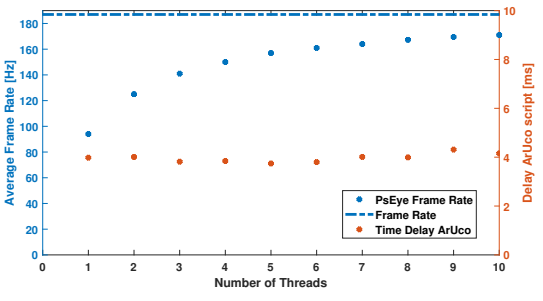


Fig. 8. Time delay and sample rate results by applying multi threading in the software.

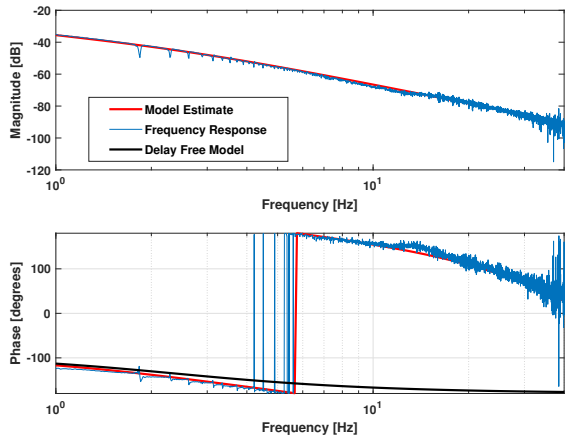


Fig. 9. System frequency response model fit and delay free model.

TABLE II
PARAMETERS FOR THE PID CONTROLLERS

BW	k_p	w_i	w_d	w_t	w_f
7Hz	440	4.8	14.5	131.9	440
10Hz	663	6.8	20.7	188	1885

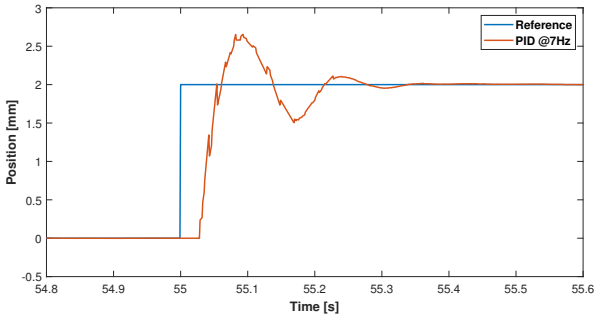


Fig. 10. Step response for a 2mm step with a 7Hz BW PID Controller

the software. The results of implementing multi threading in the software is seen in figure 8. From these results it is clear that that the sample rate is increased by multi threading. Applying ten threads results in a sample rate of 171 samples per second compared to a sample rate of 94 per second without multi threading. The sample rate of 171 samples per second is close to the maximum frame rate of the camera. From 8 it is also visible that the delay found within the algorithm is not affected significantly by applying multi threading. Taking a tenfold sample rate of the control bandwidth into account the effect on the control bandwidth is clear.

IV. Control Design

The improved sensor system has a reduced delay and an increased sample rate. However, the challenge in control still remains since a time varying delay of 10ms still leads to a significant amount of lag at the target frequency of 10Hz. Considering a maximum delay of 15ms the problem of phase lag is best explained by inspecting the phase of a pure 0.15ms delay at 10Hz:

$$\angle e^{-0.015s} = -54\omega = 54^\circ \tag{1}$$

In order to prove if this 54° delay forms a problem the system is identified and the maximum controller bandwidth with a PID controller is defined which shows the need for a delay compensation controller.

A. System Identification

The mechanical system for translations in x-direction is modeled as:

$$G_x(s) = \frac{F_x e^{-\tau s}}{ms^2 + cs} \tag{2}$$

Where F_x is the actuator force, τ the delay, m the mass and c the damping of the ferrofluid. In figure 9 the frequency response of the system with the model estimate and a delay free model is seen. The model estimate is given by:

$$G_x(s) = \frac{F_x e^{-0.11s}}{0.464s^2 + 6.9s} \tag{3}$$

Since the computer vision sensor was used for the measurements the noise increases at higher frequencies due to the limited sample rate. The challenge in control with a measurement delay is clearly visible by inspecting the difference in phase for the delay free model estimate and the model estimate.

B. PID

Using the rules of thumb for designing a PID controller (see equation 4) within the precision industry results in a maximum phase compensation of 55° [26]. Although higher phase compensation can be achieved by increasing the range of w_d and w_t in equation 4 this not advisable since it will increase the gain at higher frequencies. The high gain in the high frequency domain forms a problem since noise gets amplified which deteriorates the performance. Considering the measurement noise and additional noise due to the image blurring when the target is on the move these rules of thumb are maintained.

$$C(s) = k_p(1 + \frac{w_i}{s})(\frac{s/w_d + 1}{s/w_t + 1})(\frac{1}{s/w_f + 1}) \tag{4}$$

Considering the estimated model this would mean the control bandwidth limit is 10Hz. However, if a maximum delay of 15ms is taking into account the control bandwidth (BW) is limited to 7Hz.
The parameters for the 7Hz and 10Hz controller which were designed are seen in table II. The controllers were implemented on the microscopy stage to inspect the response for a step reference. In figure 10 the step response for a 2mm step in x-direction is seen for the 7Hz PID controller. The controller has a 37% overshoot with some oscillations. In figure 11 a step response along with the time delay

measurements is seen for a 10Hz PID controller which shows the challenge in the time varying delay of computer vision sensor. Due to the image blurring when the target is moving the time delay increases. This increase in delay results in oscillations in the response. Since the response is not satisfactory time delay compensation is used to reach the target bandwidth of 10Hz.

C. Delay Compensation

1) Control Architecture: A Filtered Smith Predictor ([25]) is used for the compensation of the time varying delay (see figure 12). In the Filtered Smith Predictor (FSP) architecture $G_m(s)$ is the plant model, C(s) the controller, $\hat{\tau}$ the delay estimate. Not considering the disturbance observer the transfer function from reference to output is given by:

$$\frac{Y(s)}{R(s)} = \frac{C(s)G(s)^{-\tau s}}{1 + \frac{C(s)G(s)e^{-\tau s}F(s)}{1+C(s)G_m(s)(1-F(s)e^{-\hat{\tau} s})}} \tag{5}$$

In the Smith Predictor architecture a constant process disturbance results in a steady state error in the output. For this positioning system two process disturbance act on the system. The first is a gravity force when the mover not being leveled perfectly. Another constant disturbance is caused by the trail forming of the ferrofluid. The ferrofluid trail exerts a magnetic force on the mover which causes a drift is in the steady state error. Due to this drift in the steady state error an additional disturbance observer was implemented which is given by:

$$\hat{d} = \frac{G_m(s)^{-1}}{Q(s)} - \frac{u}{Q(s)} \tag{6}$$

Where Q(s) is a first order low-pass filter with a cut-off frequency off 2.5Hz.

The controller $C(s)$ is designed for the plant $G(s)$ as if this plant was not delayed. The filter F(s) is implemented as a first order low-pass filter with a cut-off frequency of twice the control bandwidth. This filter makes the smith predictor more robust against changes and mismatches in the delay estimate. Furthermore, the sensor signal is filtered close to the controller bandwidth which improves the sensor noise rejection.

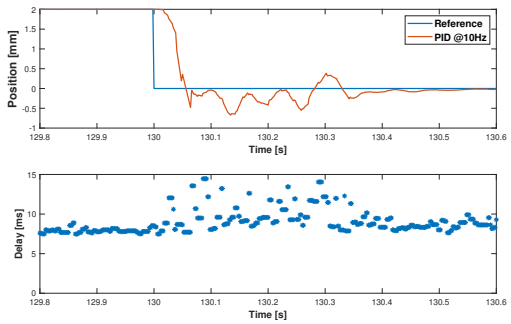


Fig. 11. Step response for a 10Hz PID controller and time delay measurements

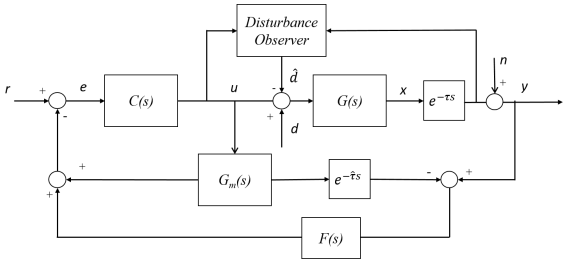


Fig. 12. Architecture of the Filtered Smith Predictor with an additional Disturbance Observer

2) Control Performance of the Filtered Smith Predictor: In figure 13 the step response of a 10Hz and 15Hz controller, using the Filtered Smith Predictor architecture, are seen. The response shows that the oscillations have reduced and a higher control bandwidth can be reach without the affecting the performance in terms of oscillations. The control actuation saturates at a control force of 2.8N and as a result the rise time for both controllers is similar.

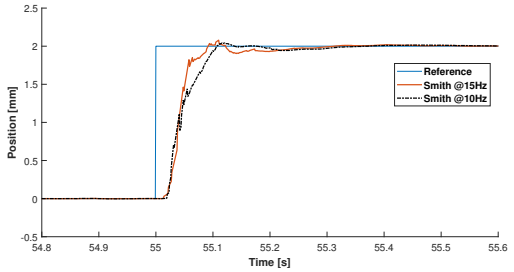


Fig. 13. Step response for a Filtered Smith Predictor with a 10Hz PID controller and a 15Hz PID controller.

Further increasing the control bandwidth of the PID controller has a negative impact on the performance in terms of settling time and oscillations. In figure 14 (a) the step response is seen for the Filtered Smith Predictor with a 20Hz PID controller. For the first step the response is similar to that of the response in figure 13. However, for the second step an increase in oscillations is seen which is explained by looking at the time delay measurements in figure 14 (b). Due to the velocity of the mover the image blurring effect increases. At higher control bandwidths the controller is more sensitive to this sudden increase in delay. In table III the positioning precision over a period of 60s is seen. Due to the lack of movement at a steady position the 20Hz FSP is not affected by image blurring and increasing delay. Increasing the control bandwidth has only a small impact on the positioning precision. The FSP has a small improvement on the positioning which is relatively small since the precision is near to the measurement dispersion of the sensor system.

V. Conclusion

This paper has presented a cost-effective computer vision sensor which can be used for precision positioning systems with a control bandwidth requirement up to 15Hz. The sensor

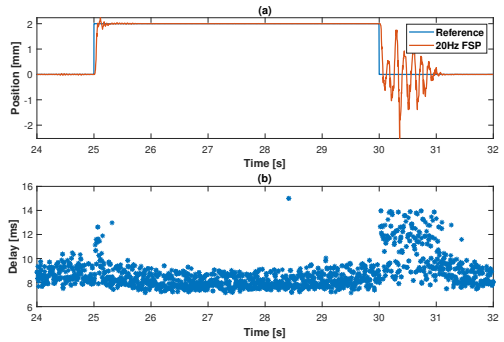


Fig. 14. Step response for a Filtered Smith Predictor for a 20Hz PID controller (a) and the time delay measurements (b).

TABLE III

3σ POSITIONING STABILITY OF DIFFERENT CONTROLLERS MEASURED OVER A PERIOD OF 60S.

Controller	3σ Positioning Precision
7Hz PID	1.52 μm
10Hz FSP	1.46 μm
15Hz FSP	1.46 μm
20Hz FSP	1.42 μm

system has an average delay of 10 ms with a sample of 171 samples per second. Using a PID controller the maximum bandwidth, at which the system has a good response, is 7Hz. By applying a Filtered Smith Predictor the control bandwidth can be doubled. The effect of image blurring leads to a significant increase in delay which is the limiting factor within the controllers.

The decreasing cost of image sensors suggest that high speed cameras can be used in the future for cost-effective sensor systems in high performance precision systems. Besides reducing the average delay high speed cameras have a short shutter time which has a significant impact on the image blurring effect. These high speed cameras can be used for computer vision sensor systems to reduce the variations in delay which allows the design of controllers with a higher bandwidth.

The amount of research which is conducted within the area of computer vision is growing and many research is focused on the improvement of the algorithms. Recent results show that the average detection time of fiducial markers can be reduced to 1ms.

Future work includes the implementation of algorithms on Graphical Processing Units and using other delay compensation methods to further increase the performance of computer vision sensors.

REFERENCES

[1] Alsalam, B.H., Morton, K., Campbell, D.A., & Gonzalez, F. (2017). Autonomous UAV with vision based on-board decision making for remote sensing and precision agriculture. 2017 IEEE Aerospace Conference, 1-12.

[2] Dhiman, V., Ryde, J., & Corso, J.J. (2013). Mutual localization: Two camera relative 6-DOF pose estimation from reciprocal fiducial observation. 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 1347-1354.

[3] Janai, J., Gney, F., Behl, A., & Geiger, A. (2017). Computer Vision for Autonomous Vehicles: Problems, Datasets and State-of-the-Art. CoRR, abs/1704.05519.

[4] Krajnik, T., Nitsche, M.A., Faigl, J., Vanek, P., Saska, M., Preucil, L., Duckett, T., & Mejail, M. (2014). A Practical Multirobot Localization System. Journal of Intelligent and Robotic Systems, 76, 539-562.

[5] Lee, S.J., Tewolde, G.S., Lim, J., & Kwon, J. (2015). QR-code based Localization for Indoor Mobile Robot with validation using a 3D optical tracking instrument. 2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), 965-970.

[6] Pestana, J., Sanchez-Lopez, J.L., Puente, P.D., Carrio, A., & Campoy, P. (2014). A Vision-based Quadrotor Swarm for the participation in the 2013 International Micro Air Vehicle Competition. 2014 International Conference on Unmanned Aircraft Systems (ICUAS), 617-622.

[7] Perez, L., Rodrguez, ., Rodriguez, N., Usamentiaga, R., & Garca, D.F. (2016). Robot Guidance Using Machine Vision Techniques in Industrial Environments: A Comparative Review. Sensors, 16 3.

[8] Sani, M.F., & Karimian, G. (2017). Automatic navigation and landing of an indoor AR. drone quadrotor using ArUco marker and inertial sensors. 2017 International Conference on Computer and Drone Applications (IConDA), 102-107.

[9] Serra, P., Cunha, R., Hamel, T., Cabecinhas, D., & Silvestre, C. (2016). Landing of a Quadrotor on a Moving Target Using Dynamic Image-Based Visual Servo Control. IEEE Transactions on Robotics, 32, 1524-1535.

[10] Habib, H. (2015). Design of a three Degrees of Freedom planar precision stage using a single Position Sensitive Detector. Master thesis TU Delft.

[11] Mok, G. (2015). The design of a planar precision stage using cost effective optical mouse sensors. Master thesis TU Delft.

[12] Moorsel, L. (2017). A planar precision stage using a single image sensor. Master Thesis TU Delft.

[13] Normey-Rico, J. & Camacho, E. (2008). Dead-time compensators: A survey. Control Engineering Practice. 16. 407-428. 10.1016/j.conengprac.2007.05.006.

[14] Normey-Rico, J.E., & Camacho, E.F. (1999). Smith predictor and modifications: A comparative study. 1999 European Control Conference (ECC), 2257-2263.

[15] Richard, J. (2003). Time-delay systems: an overview of some recent advances and open problems. Automatica, 39, 1667-1694.

[16] J. Smith (1957). Closer control of loops with dead time. Chemical Engineering Progress, vol. 53, pp. 217219.

[17] Ingimundarson, A. Robust tuning procedures of dead-time compensating controllers.

[18] Normey-Rico, J., Garcia P. & Gonzalez A (2012). Robust stability analysis of filtered Smith predictor for time-varying delay processes. Journal of Process Control, Volume 22, issue 10, pp. 1975-1984.

[19] Lampaert, S (2018). In-plane friction behavior of a ferrofluid bearing.

[20] Fiala, M. (2005). ARTag, a fiducial marker system using digital techniques. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2, 590-596 vol. 2.

[21] Garrido-Jurado, S., Muoz-Salinas, R., Madrid-Cuevas, F.J., & Marn-Jimnez, M.J. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recognition, 47, 2280-2292.

[22] Abawi, D.F., Bienwald, J., & Dorner, R. (2004). Accuracy in optical tracking with fiducial markers: an accuracy function for ARToolKit. Third IEEE and ACM International Symposium on Mixed and Augmented Reality, 260-261.

[23] Wang, J., & Olson, E. (2016). AprilTag 2: Efficient and robust fiducial detection. 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 4193-4198.

[24] Romero-Ramirez, F.J. (2018). Speeded up detection of squared fiducial markers. Image and Vision Computing, vol. 2018, no. Volume 76, pp. 38-47, 2018.

[25] Ingimundarson, A. Robust tuning procedures of dead-time compensating controllers.

[26] Schmidt, R.M., Rankers A. & van Eijk, J. (2014). High-Tech Functionality by Multidisciplinary System Integration, ISBN 978-1-61499-367-4. Last, F. M. (Year Published) Book. City, State: Publisher.

Chapter 3

Discussion

The goal of this thesis was to further develop the computer vision sensor system in order to inspect the feasibility of computer vision sensors for in precision systems. The first step was to analyze the performance of the different subsystems. After this analysis it was found that the current controller did have an uncertainty for small currents. By replacing the current controller for the actuator the challenges with the sensor system were isolated.

3-1 Actuator System

The actuator which was developed by [1] was not included in the analysis. The complexity of the position matrix is inherent to some errors. Since the controller interpolates for positions which are not stored in the matrix, some in-balance in force allocation can be present. Furthermore, the force allocation which determines how 7 forces are used to generate the set-point forces is not optimal. Since the heat generate in the coils is transferred to the ferrofluid this has a direct effect on the system parameters.

Ferrofluid Bearings

Ferrofluid bearing systems are a promising technology since they offer a bearing which has no static friction. Static friction is challenging in control but the use of ferrofluid has is own challenges.
A ferrofluid bearing is subject to trail forming meaning that some ferrofluid is released from the bearing. This trail forming has an effect on the fly height of the bearing and causes some parasitic forces which are not present in conventional bearing system. The changing fly height of the 8 ferrofluid bearings makes it challenging to precisely level the bearing.
Since the bearings consist of a fluid it is subject to evaporation. During the relatively hot summer of 2018 the evaporation of the ferrofluid had to be replaced regularly. With the hot temperatures of the summer it was also notices that the ferrofluid would get sticky, which led to a higher damping. Higher quality ferrofluid can be used to reduce these effects but

considering the sensor system this would have no significant effect. For high performance precision systems it is common to have a controlled environment which can significantly reduce the variations in the ferrofluid parameters.

3-2 Sensor System

The improvements on the sensor system were significant in terms of delay, precision and sample rate. This improvement is mostly attributed to the use of a computer which increases the computational power significantly compared to a Raspberry Pi.

Nvidia TX2

A Nvidia TX2, which is a high performance computing device, was acquired to facilitate the algorithm of computing power. The specifications of the Nvidia TX2 are promising but complications were encountered in the compatibility between this computing device and the image sensor. Compatible camera's are available but were not considered as a cost-effective alternative.

PlayStation Eye

A €15 PlayStation Eye camera was implement as the image sensor in the subsystem. This camera is widely used in computer vision applications and the frame rate is high for a camera which can be connected to a computer over a USB connection. The limit in shutter time imposed difficulties for the sensing system. Other camera's can offer better performance in terms of shutter time which can significantly improve the performance.

Camera Placement

The camera was placed in a 3D-printed holder which was placed on the base of the stage. The height is adjustable such that the camera is focused on the marker map and the image is not blurred. A caliper was used to align the camera. A misalignment of the camera leads to a error in the position estimation which has an effect on the position dependent motor constant matrix $\Phi(x, y, \theta)$. Using the maximum range of the mover, the position estimation was calibrated in Simulink obtaining a more accurate estimation. Improving the camera placement can have some effect on the accuracy of the actuator.

Camera Lens

A macro-lens from a Raspberry Pi module was used to focus the image sensor on the marker map. The macro-lens is place on the image sensor using a separate 3D-printed holder. Since the lens was not specifically designed for this image sensor this led to a distorted image front. This distorted image front caused an unreliable position estimation of the rotation θ. The availability of macro lenses compatible with image sensors such as the PlayStation Eye is scarce. This limits the practical realizable implementation of a PlayStation Eye as a precision sensor where an accurate measurement of θ is needed.

Multi Threading

Multi threading is a useful tool to improve the sample rate of the sensor system given that enough computer power is available. The delay measurement of retrieving an image of the camera was unreliable due to the implementation of multi threading. The image is retrieved with a single command in C++. Since only one thread is able to communicate with the camera at the same time, other threads had to wait before receiving an image leading to an overestimated delay. The delay of retrieving an image from the camera is therefor estimated with the inverse frame rate of the camera. For this reason the delay measurement is not always accurate since variations in the process of retrieving an image exist. Furthermore, measuring time within computers is not always accurate. During the measuring of the average frame rate it was noticed that the error in time measurement could be as large as 30%. By experimenting with different C++ libraries a more accurate timing library was implemented the software.

Image Blurring

The effect of image blurring is considerably important in real-time computer vision sensing. The detection time of the ArUco algorithm is tripled in some instances due to the image blurring effect. Since image blurring occurs when the marker map moves during the shutter time interval, this effect can be decreased using high speed camera's which have short shutter times. The shutter time ($t_{shutter}$) is also related to the maximum speed (V_{max}) of the mover at which the sensor system is able to estimate its position and is approximated by:

$$V_{max} = l_{bin} \times t_{shutter}$$

where l_{bin} is the length of a single bin in a marker.

Measurement Dispersion

The effect of the measurement dispersion is only measured for the translational estimation of the position. Decreasing the resolution has the logical effect of decreasing the measurement precision although no significant difference between 640x480 and 320x240 pixels was found. The measurement precision can best be increased by increasing the magnification of the lens which, off course, will increase the image blurring effect due to the increase in the image velocity seen by the camera.

3-3 Control

The controller is implemented on a Raspberry Pi which is compatible with Simulink. Implementing the controller on other devices is difficult since the controller needs the position dependent motor constant matrix which is a 8.8Mb data file.

Raspberry Pi

The Raspberry Pi is a powerful computing module with a Broadcom BCM2837 chip. The maximum sample rate of the control system which was tested was 10kHz. Using this sample

rate the amount of measurement data which could be stored was limited. Furthermore, using a high sample rate the Raspberry Pi was more likely to experience an error where the controller script was temporarily shut down. A sample rate of 1kHz was found suitable for the storage of measurement data and reliable operation of the controller. The compatibility with Simulink offers a great cost-effective control device for other control systems.

Control Bandwidth Limitations

The bandwidth limitation without using delay compensation is limited to the amount of lag a controller can compensate. Using the rules of thumb for PID controllers from [7] in performance mechatronics this phase compensation is limited to around 55°. Some experiments were conducted using a higher cut-off frequency of the derivative term which increases the phase compensation. Due to the increase of this parameter the Raspberry Pi would crash after several seconds. It was observed that increasing the sample rate of the controller allowed a larger cut-off frequency of the derivative term but were not implemented due to the data storage limitation.
By implementing a Smith Predictor the phase lag of a delay is removed for the loopgain which allows any controller to be design for the non-delayed system if the complementary sensitivity function is considered. Considering the sensitivity function of a delayed output the control bandwidth limit is given by the crossover frequency of the sensitivity function [8] which, for a pure delay is given by:

$$S(s) = 1 = 1 - e^{-\tau j\omega}, \text{ for } \omega_{max} = \frac{\pi}{3\tau} \tag{3-1}$$

For an average delay of 10ms this would result in a maximum control bandwidth of 16Hz (given that 3-1 is in radians) which is close to the achieved control bandwidth of the microscopy stage.

Positioning Precision

The positioning precision is in the same order of magnitude as the measurement dispersion. In order to improve the precision of the system additional filtering of the sensor signal can be implemented. However, in order to get an exact positioning precision of the system, the measurements should be validated with a high resolution sensor.

Chapter 4

Conclusion and Recommendations

4-1 Conclusions

Computer Vision Sensor System

The average delay from the sensor subsystem is reduced from 60ms to 10ms. The delay reduction is realized by implementing the ArUco software on a laptop and using a PlayStation Eye camera. The time delay remains time varying due to the nature of the algorithm and computations within the computer. A significant increase in delay is caused by blurred images are used to detect the markers. This increase in delay during motion of the mover limits the maximum control bandwidth of the system.
The sample rate of the sensor system was increased from 16Hz to 94Hz. With the implementation of multi threading in the software the sample rate of the sensor system is further increased from 94 tot 171Hz. In terms of 3σ measurement dispersion of the sensor system the performance is increased from 3.9μm to 0.96μm.

Control Performance

Without the use of delay compensation the control performance is increased from 0.87Hz to 7Hz. The positioning precision at this bandwidth, given by the 3σ interval is 1.52μm. The target bandwidth of 10Hz is obtained using a Filter Smith Predictor as a delay compensator. The maximum controller bandwidth, using a Filtered Smith Predictor, is 15Hz. Further increasing the control bandwidth leads to oscillations due to the sensitivity to an increase in delay. The 3σ positioning precision of the Filtered Smith Predictor at a control bandwidth of 10 and 15Hz is 1.46μm. Due to a distorted image front the algorithm was not able to give an accurate estimation of the rotational position of the mover.

4-2 Recommendations

4-2-1 Sensor System

Camera Lens

The measurement estimation of the sensor signal is affected by the distorted image from the camera. The image quality can be improved by using another camera lens. It is expected that the jumps in the rotational sensor signal can be eliminated. However, an adequate lens for the PlayStation Eye image sensor might not be available. Other cameras can be used for the same purpose which have different specifications in order to increase not only the rotational sensor signal.

Choice of Camera

Since the prices of cameras tend to decrease over time it is possible to implement new low-cost cameras which can be implemented in computer vision sensor systems. In the sensor system half of the delay is attributed to the shutter time of the camera. Decreasing the shutter time has a direct affect on the overall delay. Furthermore, by reducing the shutter time the effect of image blurring would be decreased which can be used to either increase the velocity of the mover or as to maintain a more constant delay.

4-2-2 Control

Velocity

The velocity of the mover increases the delay it is recommended to limit the velocity dependent on the application. The velocity of the mover can be limited if a human is in the loop considering the small position changes needed for microscopy applications. Furthermore, higher control bandwidths could be used if path planning is implemented to reduce the maximum velocity.

Precision

In order to increase the precision several methods can be used to filter the sensor signal, for example a Kalman filter. This filtering would make the system more robust against the sensor noise meaning that a better precision can be obtained.

Time Delay Control

Other methods can be used to supply phase to the system without applying any type of delay compensation. Non-linear types of control such as reset control can go beyond the limitation of conventional PID controllers which can be used to increase the bandwidths.

Appendix A

System Overview

In this chapter an overview of the microscopy stage is depicted. The working principle will be shortly highlighted as well as the main communication protocols, hardware and software configurations. In figure A-1 a schematic overview of the stage is seen. At the bottom of the figure the camera is seen. The camera is used to capture an image of the ArUco markermap. With help of the ArUco software this image is used the calculated the relative position of the ArUco markermap with respect to the camera. More information on ArUco and the markermap can be found in appendix C. In order to displace the mover to the desired position a Lorentz force actuator is used with which consists of 7 coils and 8 magnets (see appendix B. The coils are embedded in a printed circuit board (PCB) and the magnets are placed within the steel mover. The mover rests upon a ferrofluid bearing which is held in place by the magnets.

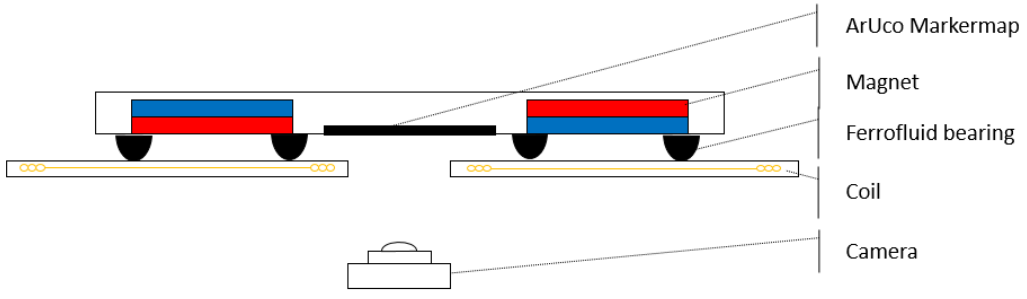


Figure A-1: Overview Stage

A-1 Delay Overview within the Setup

In figure A-2 a schematic overview of the hardware and software is depicted. The schematic is explained from left to right. At first the camera (PlayStation Eye) retrieves an image of the markermap and sends this image to a laptop over a Universal Serial Bus (USB) protocol. The estimated delay during this process is approximately given by the inverse frame rate of the camera, $\frac{1}{187} \approx 5ms$. The ArUco software is implemented on the laptop and there the relative position of the camera w.r.t. the markermap is calculated. Once the position information is available it is send to a Raspberry Pi 3B with a User Datagram Protocol (UDP). The process of of detecting a single image and sending the information to the Raspberry Pi is approximately 5ms not considering any variations in the delay. A real-time Simulink controller is implement on the Raspberry Pi which computes the necessary forces / currents which have to be supplied to the system. The sample rate of the controller is 1000Hz which corresponds to a delay of 1ms.
Once the desired currents are known the Raspberry Pi, it communicates the set-points to a Digital-to-Analogue-Converter (DAC) with a Serial Peripheral Interface using a Serial Peripheral Interface (SPI). The clock rate of the Raspberry Pi operates at a frequency of 16MHz. Sending a single commands requires a 32bit command which corresponds to 32 clock cycles. Since the system uses seven actuators, seven commands must be send. The delay of sending the commands from the Raspberry Pi to the DAC is approximated by:

$$t_{SPI} = \frac{NumberofCommands \times BinaryCommandSize}{Clockrate} = \frac{7 * 32}{16MHz} = 14\mu s$$

The DAC converts the digital information to an analogue voltage output, the rise time of a 5Volt step is less than $10\mu s$ according to the supplier.
The voltage is supplied to a Linear Power Amplifier (LPA) which converts the voltage set-point to the corresponding current. The frequency at which the amplifiers are able to track a reference is at least 25kHz according to [3]. Using this conservative estimate the rise time of the current is approximated by:

$$t_{Amplifier} = 0.35/Bandwidth = 14\mu s.$$

With this estimation the total delay of the actuator subsystem is $38\mu s$, which is only a fraction (3.5%) of the delay which is found in the sensor system.

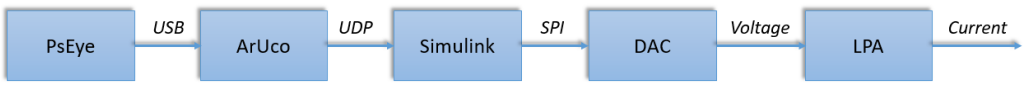


Figure A-2: Schematic Overview Stage

A-1-1 Laboratory Setup

In figure A-3 a a top view of the microscopy stage without the mover is seen as well as the box containing the amplifiers. The whole in middle of the PCB allows the image sensor to capture images of the marker map. A thin layer of plastic is placed upon the PCB to ensure that it is not contaminated with ferrofluid. The brown color of the microscopy stage is caused by ferrofluid residuals.
The amplifier box consists of 7 power amplifiers and two voltage supplies which supply a voltage to the amplifiers of -30/+30 Volts. In figure A-4 the LTC2668-12 DAC (left) is seen

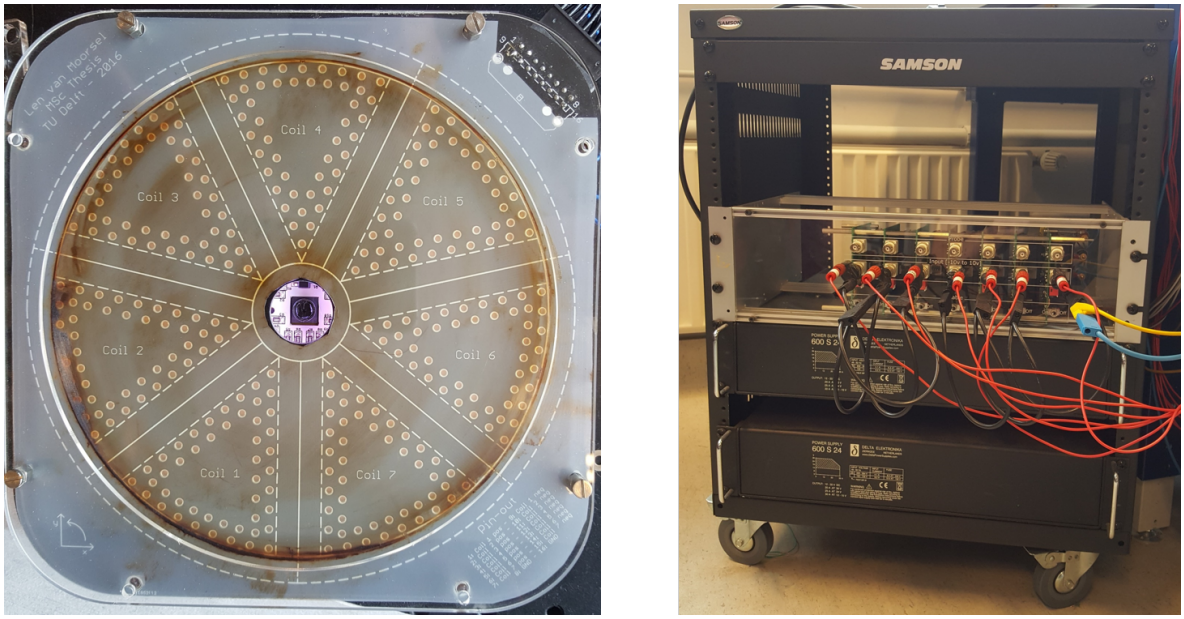


Figure A-3: Top view of the microscopy stage without the mover (left) and the amplifier box which was used as the power supply.

and the Raspberry Pi 3B. An additional voltage supply (-10/+10 Volts) is used to supply the DAC of power. The Raspberry Pi communicates with the DAC using a SPI protocol which uses 4 wires.

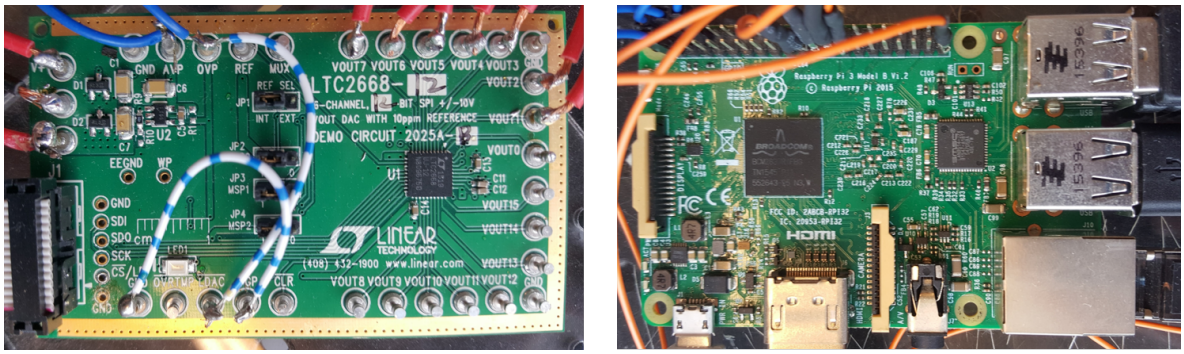


Figure A-4: The LTC2668-12 Digital-to-Analogue Converter and the Raspberry Pi 3B.

Appendix B

Actuator

For the system to move a Lorentz actuator is used which is able to convert current into a force using the Lorentz law.

$$F_l = I_l \times B \tag{B-1}$$

This force is used to move the mover to the desired position. In order to apply force and torque in the necessary directions van Moorsel came up with his own actuator design. With the use of 8 magnets and 7 coils in a planar configuration the system is supply any necessary force and torque for every position. The actuator design was found to be suitable for the application but other problems were encountered in the current controller of the system.

B-1 Planar Coil Actuator

The coil configuration of the microscopy stage is can be seen in figure B-1. In the left of B-1 a current is driven through a single coil which creates a Lorentz force. Since the coils are stationary, a reaction force with opposite direction is resulting on the mover. In the right of B-1 an example is shown of the case where all the 7 coils are used to supply a pure torque to the mover. By regulating the amount of current going through each coil the resulting force or torque can be controlled.

Force Allocation

The force of a current is dependent on both the magnitude of the current and the position of the coil with respect to the mover. A force transition matrix $\Phi(x, y, \theta)$ was introduced to derive the relationship between current and force for different positions. For a single coil at an arbitrary location the force is given by:

$$\begin{bmatrix} F_x \\ F_y \\ F_\theta \end{bmatrix} = \Phi(x, y, \theta)\ I = \begin{bmatrix} kF_x \\ kF_y \\ kF_\theta \end{bmatrix} \begin{bmatrix} I \end{bmatrix} \tag{B-2}$$

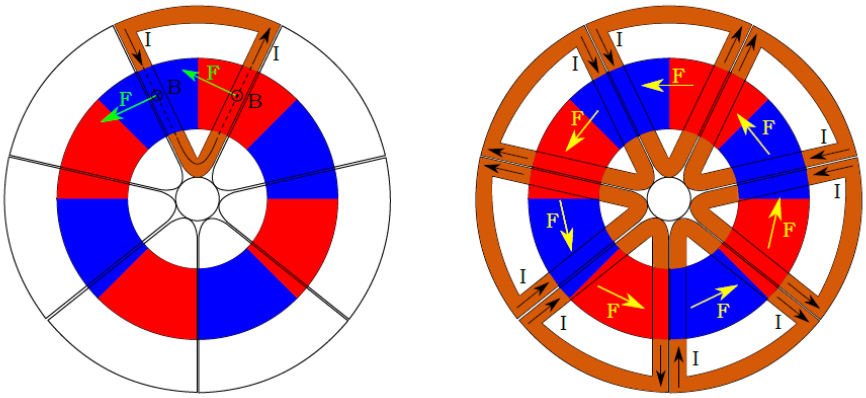


Figure B-1: Schematic overview of the actuator. A single coil consists of two legs through which a current is driven to create a Lorentz force. This configuration of 7 coils and 8 magnets was designed by [1] for an minimum heat production production per unit of force.

Where kF_x, kF_y, kF_θ are linear motor constants, I the current and F_x, F_y, F_θ the resultant forces. For the 7 coils and 8 magnets configuration the transition matrix is given by:

$$F = \Phi(x, y, \theta)\ I \tag{B-3}$$

With $F \in R^{3x1}$, $\Phi \in R^{3x7}$ and $I \in R^{7x1}$.
In order to translate a set point in the force back to the current the pseudo inverse was implemented in [1]:

$$\Phi^\dagger F = \Phi^\dagger \Phi I = I \tag{B-4}$$

where $\Phi^\dagger$ represents the pseudo-inverse $(\Phi^T\Phi)^{-1}\Phi^T$.
With simulations the force allocation was inspected of which an example can be seen in figure B-2. With a desired force of 1 Newton in x-direction it can be seen that this force allocation is far from optimal. Multiple forces counter react to each other and only contribute a small part to the desired resultant force. For this example, a single coil could be used to create the desired force of 1 Newton in x-direction (F_4).
Since the actuator was not considered to be limiting the performance the same pseudo-inverse matrix was used during this project. However, since the actuator was designed for a minimum production of heat a possible topic for future work could be to optimize the force distribution to further minimize the production of heat.

Ferrofluid Bearing System

The damping of the system is provided by ferrofluid. Ferrofluid consists of nanometer-size ferromagnetic particles which are dissolved in a liquid. The ferromagnetic particles are attracted to the location where the magnetic field is strongest. Since this is around the corners of the magnets the ferrofluid is located at the edge of each magnet, see figure B-3. A ferrofluid has no stick-slip behaviour which is advantageous over other types of bearing systems. The damping of a ferrofluid is modelled by:

$$c = \eta_r \eta_n \frac{A_{bearing}}{h_{bearing}} \tag{B-5}$$

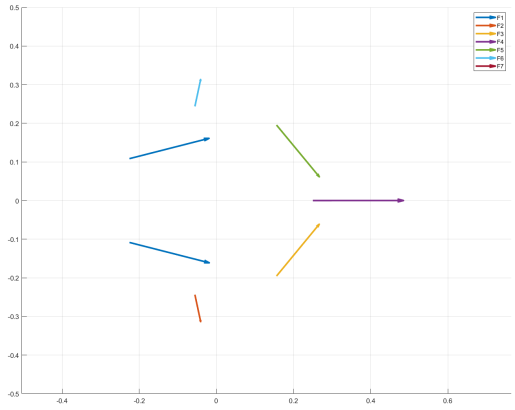


Figure B-2: Force allocation using the pseudo-inverse $\Phi^{\dagger}$. An example of how the forces are distributed for a 1 Newton force in x-direction when the mover is located at the origin $x = 0, y = 0, \theta = 0$. The seven forces depicted in the figure have a resultant force of 1 N and a torqu of 0 Nm.

Where η_r is dependent on the velocity profile, η_n the ferrofluid viscosity, $A_{bearing}$ the area of the contact area and $h_{bearing}$ the height of the ferrofluid bearing [9]. The exact modelling of the ferrofluid is considered to be outside of the scope of thesis.

B-2 Current Control using Pulse Width Modulation (PMW)

In order to regulate the voltage across a coil a pulse width modulator is used in [1]. A PWM uses a rectangular pulse (voltage) where the width of the pulse can be adjusted such that the average voltage over the load is changed. The pulse width is referred to as a duty cycle where a high duty cycle (100%) corresponds to the maximum voltage output of the PWM. Ideally the voltage output of the PWM is linear with its duty cycle although this was found to be inaccurate. To overcome the non-linearity, the duty cycle and the voltage output was measured and compensated with a look-up table. During measurements it was verified that the resistance of the coils increased with approximately 12 % due to the increase in temperature. Since a current is both dependent on the voltage and resistance current feedback was implemented.

$$I = U/R \tag{B-6}$$

The current measurement was found to be inaccurate at low current set-points. A current is measured as an absolute value which makes it difficult to define the direction of the current. Therefor a feedforward controller was used for set-points below 0.5 A. Since this is a feedforward controller it does not take into account any deviations such as the change in resistance due to the temperature. The uncertainties found within this current controller gave enough reason to replace the PWM with a power amplifier.

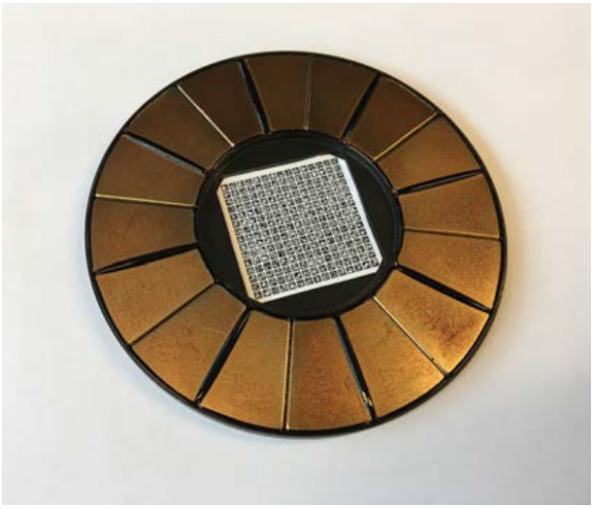


Figure B-3: Image of the mover (bottom). 16 Magnets are placed in a configuration to create 16 poles. The black lines is the ferrofluid which is gathered at the strongest magnetic fields. At the center the marker map is placed which is used for the computer vision sensor.

B-3 Linear Power Amplifiers

Since it is desired to have an accurate current supply to the coils the choice was made to use a linear power amplifier. In order to be able to supply high peak currents a choice was made to use the OPA549 High-Voltage, High-Current Operational Amplifier from Texas instruments which is able to supply a bipolar current (in both directions). The specifications can be seen in table B-1.

Table B-1: OPA549 Specifications

OPA549	
Max constant current output	8A
Peak Currents	10A
Dual Voltage Supply	+-4V to +-30V
Settling Time	20 μs

B-3-1 Linear Power Amplifier Design

In order to supply an bipolar current to a coil the schematic was designed using the OPA549 as can be seen in figure B-4. With this simple schematic the amplification rate between the voltage input and the current output is derived. In practice more decoupling capacitors would be implemented to the voltage supply to filter out peak voltages and get a smooth signal as can be found in the datasheet of OPA549 [ref]?. The current through the impedance Z_l is

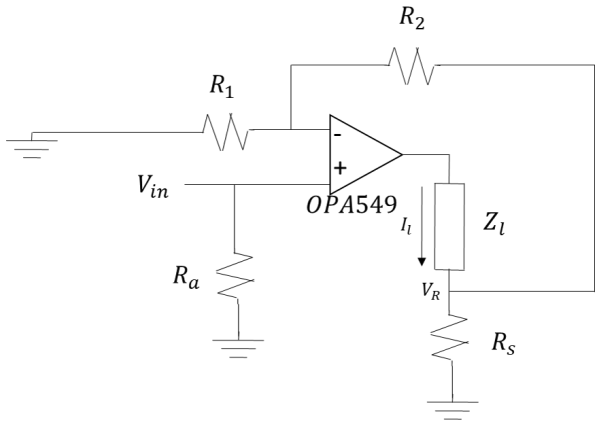


Figure B-4: Linear Power Amplifier Design where R_1, R_2, R_a and R_s are resistances, V_{in} the input Voltage, Z_l the load impedance which conducts a current I_l.

given by:

$$I_l = \frac{V_R}{R_s} + \frac{V_R - V_in}{R2} \tag{B-7}$$

with,

$$\frac{V_R - V_{in}}{R_2} = \frac{V_in}{R_1} \tag{B-8}$$

such that:

$$V_R = V_{in}(\frac{R_2}{R_1} + 1) \tag{B-9}$$

If V_R in B-7 is substituted by B-9 the resulting current through Z_l is given by:

$$I_l = \frac{V_{in}}{R_S}(\frac{R_2}{R_1} + 1) + \frac{V_{in}}{R_1} \tag{B-10}$$

Assuming that the sensing resistance R_s is much smaller than R_1 and R_2 the term $\frac{V_{in}}{R1}$ is small such that current through Z_l is given by:

$$I_l = \frac{V_{in}}{R_s}(\frac{R_2}{R_1} + 1) \tag{B-11}$$

Under the assumption that the sensing resistance (R_s) is small the gain from the input voltage to the output current is given by:

$$1/R_s(\frac{R_2}{R_1} + 1) \tag{B-12}$$

Which could be used to design a proper amplification rate between the voltage input supplied by a digital to analogue converter and the desired output current.
Due to time limits and inexperience with soldering it was decided to let Electronic and Mechanical Support Division (DEMO) handle the manufacturing of 7 power amplifiers. However, due to the high-cost (approx. €2000) off the manufacturing it was decided to use other amplifiers made by a previous student. However, the simple derivation would be use full for anyone designing a simple current output power amplifier and is therefore reported in this thesis.

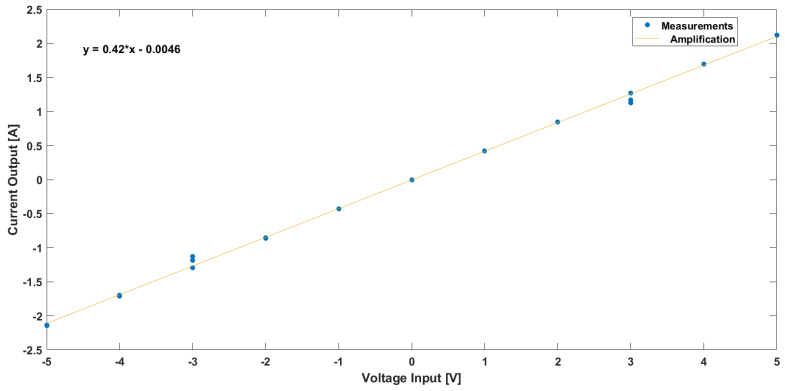


Figure B-5: Amplification rate of the amplifier measured at different set points with a multimeter.

B-4 Linear Power Amplifiers Laboratory

In [3] seven custom amplifiers were manufactured using the OPA548 from Texas Instruments. Every individual amplifier is able to deliver a continuous output current of 3 A. In [3] it was shown that the amplifiers are able to operate at frequencies up to 25kHz. Since the amplification of the amplifiers were unknown the amplification was measured. Input/output measurements at different set points are seen in figure B-5. No explanation has been found for the current spread at an the input voltages of -3/+3 V.

B-4-1 Digital-to-Analogue-Converter

The LTC2668-12 is operated via the Raspberry Pi and supplies a voltage to operate the power amplifiers. The LTC2668-12 is connected to the Raspberry Pi using a SPI connection. The Simulink Raspberry Pi package allows to operate the SPI pins of the Raspberry Pi which was convenient for implementation. More information on how to operate the LTC2668-12 is shown in the operation manual appendix.
The use of the LTC2668-12 allows a 12bit set-point of the voltage. The voltage range at which the LTC2668-12 operates is -5/+5 Volts. Meaning that the resolution at this operating range is given by:

$$V_{res} = \frac{10}{2^{12}} \approx 0.5\text{mV}$$

With the amplification rate of the amplifier and the voltage resolution the resolution of the combined system is:

$$I_{res} = 0.42[\text{A/V}] * 0.5[\text{mV}] \approx 2\text{mA}$$

The LTC2668-12 allows other operating ranges such that the actuator force could be increased if necessary. However, due to the increase in delay at higher velocities the operating range of the LTC2668-12 was not increased.

Appendix C

Sensor System

Applying computer vision in applications has gained popularity in the last decades. Due to the availability of low-cost and fast micro-controllers and developments in pattern recognition has made this technology suitable for a broad range of applications. Computer vision applications can be found within the domain of robotics, unmanned areal vehicles (UAV's), facial recognition, medical diagnosis and others. Different software packages such as ARToolkit, OpenAR and ArUco are freely available for academical and commercial purposes [[10], [11], [12], [2]]. Most of these packages are build upon the Open Computer Vision Library (OpenCV) which is optimized for real-time application and computing efficiency.
Augmented Reality library from the University of Cordoba (ArUco) has been introduced by [2] and has shown to be effective and fast with respect to other methods such as ARToolkit+, ChiliTags and AprilTags [6].
ArUco makes use of a QR-like pattern which is called a marker. An example of a QR-code and an ArUco marker can be found in figure C-1. The black / white squares within a marker function as a binary bit. The ArUco marker in figure C-1 contains 25 bits meaning that it can have over $2^25 \approx 33$ million configurations. Markers with a larger amount of bins are available as well but are not favorable in terms in computational effort.

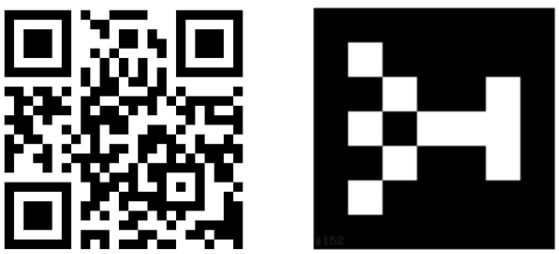


Figure C-1: QR-code (left) vs a ArUco marker (right)

C-1 Marker Identification Using ArUco

In this section a short overview is given of the pipeline that is used by the ArUco software to detect the markers in an image. The relative position of a single marker with respect to the camera is computed with the locations of the four corners of the marker.

1. In figure C-2 the different steps of the ArUco software are depicted.
   (a) **Input image** (C-2-a), the original (colored) input image is seen.
   (b) **Segmentation** (C-2-b), the original input image is converted into black white and divided into multiple segments. Corners and edges are detected by the difference in intensity between the different segments and pixels.
   (c) **Contour Filtering and Extraction** (C-2-c), the contours different objects in the image are extracted. The contours are filtered in order to extract only the relevant contours and discard irrelevant contours.
   (d) **Marker code extraction** (C-2-d), the inner region of every resulting contour is analyzed to check if the contours contain a marker. The perspective is firstly removed by a projection (C-2-e) and divided into grids. The resulting image is then converted in a grid containing binary numbers (C-2-f). Since a 90 degree rotation leads to a different marker 4 different candidates are considered. If the binary grid is valid marker it is accepted otherwise it is discarded as a background element.
   (e) **Corner up-sampling**, The markers are detected in the projected and segmented image and the position of the marker in the original image is found. Since the corners of each markers are used for the estimation of the position it has a direct influence on the precision. Therefore the area around the corner is up-sampled in the vicinity of the corners to achieve a more precise estimation.

The ArUco software consists of various features in order to increase the speed of the detection. The version (ArUco 3.0.1) which is used in this project has a special mode for video sequences in order to speed up the process. One particular effective method of speeding op the process is explained as follows:

- Once a marker is detected in an image the size of the marker with respect to the to image is known. Since the next image in the video sequence is most likely similar to the previous image the threshold, minimum marker-size, is updated. By doing this the algorithm will not try to locate markers in small contours in the image. This adaptive thresholds has shown to improve the detection speed of the software in [6].

C-2 Marker map Configuration and Placement

Many different marker configurations are possible as mentioned before. However, in literature it is mentioned that the performance in terms of robustness and precision is not the same for every marker type / configuration. It is suggested to use the ARUCO_MIP_36h12 marker

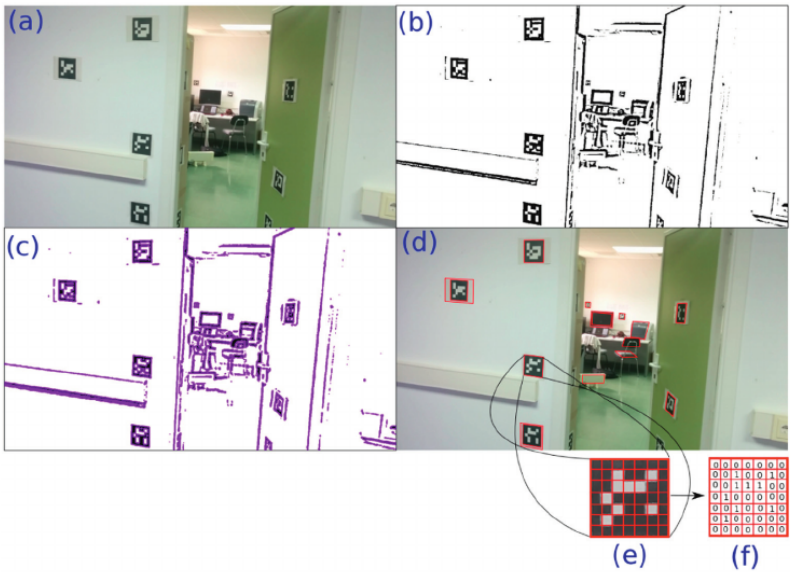


Figure C-2: A simple illustration of how the ArUco software is able to detect markers in an image [2]. The original image (a) is converted to black white and contours are detected (b). The contours are filtered which result in possible marker candidates (c). From resulting candidates (d) the perspective is removed and divided into grids (e). The grid is used then used to put the image into a binary grid (f) which contain the marker number

library since it has shown to have the best results with respect to robustness and precision. The ARUCO_MIP_36h12 marker library consists of 250 different markers which can be used to create a marker map.
In figure C-3 an example can be seen of how the image sensor sees a part of the marker map. The left image depicts how the image sensor sees the marker map as it moves along the image sensor. In the right part an actual image of the marker map as seen by the image sensor is seen.
The marker map consists of 15x15 markers with a single marker size of 2.5mm including some white space between the markers. In [4] it was derived that a minimum field of view of $2\sqrt{2}$ markers was needed to ensure that a single marker was always entirely visible for the image sensor. Since the amount of markers directly affects the delay of the software the marker size was not reduced to enable more markers to be visible to the image sensor. The size of a single bin, (see figure C-2) is $250\mu m$.

C-3 Sensor System Overview

In C-4 an overview of the sensor system is seen. Each image is taken on a PlayStation eye which send the image data to the laptop. With the use of OpenCV it is possible to directly receive the images in C++ and retrieve the position with ArUco. In order to send the information to the controller, a Raspberry Pi 3B, the information is send via the UDP (internet) protocol. In order to assign the Raspberry Pi an IP address, which is needed to receive the information, an additional router is used which is connected to both the laptop and the Raspberry Pi.

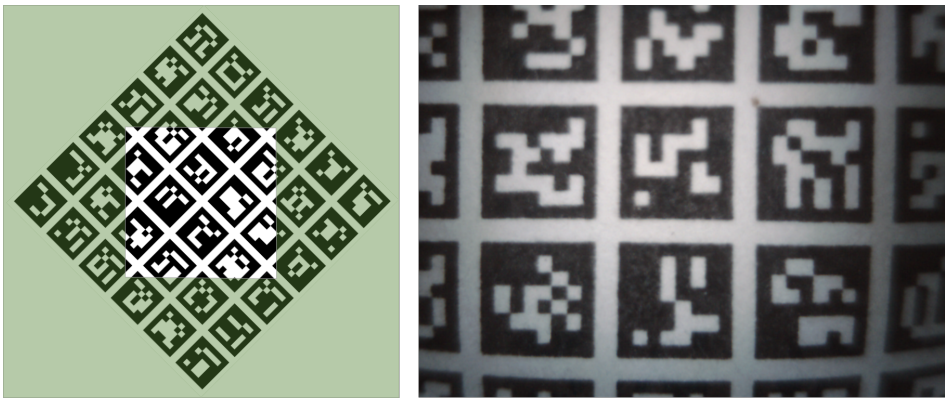


Figure C-3: Conceptual field of view of the image sensor and an actual image of the image sensor

In the beginning of the project a Nvidia TX2 was acquired for the implementation of the ArUco software. However, due to incompatibility with PlayStation Eye a choice was made to implement the ArUco software on a laptop running on Ubuntu. Compatible camera's for the Nvidia TX2 are available starting at a price of €500. A custom hold for the PlayStation

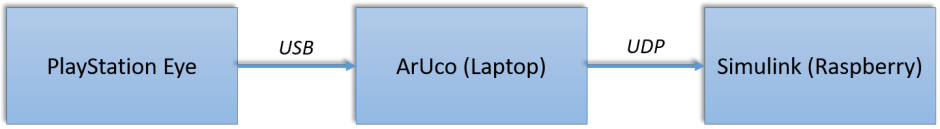


Figure C-4: Schematic overview of the sensor system

Eye circuit board was designed in SolidWorks to place the PlayStation Eye directly under the target. To make sure every image contained enough contrast for the ArUco software the target is illuminated with an Raspberry Pi add-on (LISIPAROI). A photo of the image sensor configuration can be seen in figure C-5.

C-4 Sensor Resolution and Magnification

The PlayStation Eye sensor used in this project has a OV7720 CameraChip^{TM}sensor with a 640x480 pixel arrangement. The size of a single pixel is $6\mu m$ x $6\mu m$ with a total image area of $3984\mu m$ x $2952\mu m$. In order to increase the frame rate of the camera the number of pixels that are read out can be adjusted. When the resolution of the camera is set to 320x240 the camera reads out 4 pixels as a single pixel. This phenomenon is called binning and is used to speed up the frame rate of the camera as well as to reduce the computational effort needed within the ArUco software. Using the 320x240 pixel arrangement the maximum frame rate of the image sensor of 187 frames per second is obtained. A camera resolution of 160x120 was also tested but the results showed no advantage over the 320x240 resolution.
The size of the image that is seen by the camera sensor has approximately a width of $9mm$ which is projected on the image sensor of width $3.984mm$. With this information the magnification factor of the optical system can be derived as the size of the image divided by the

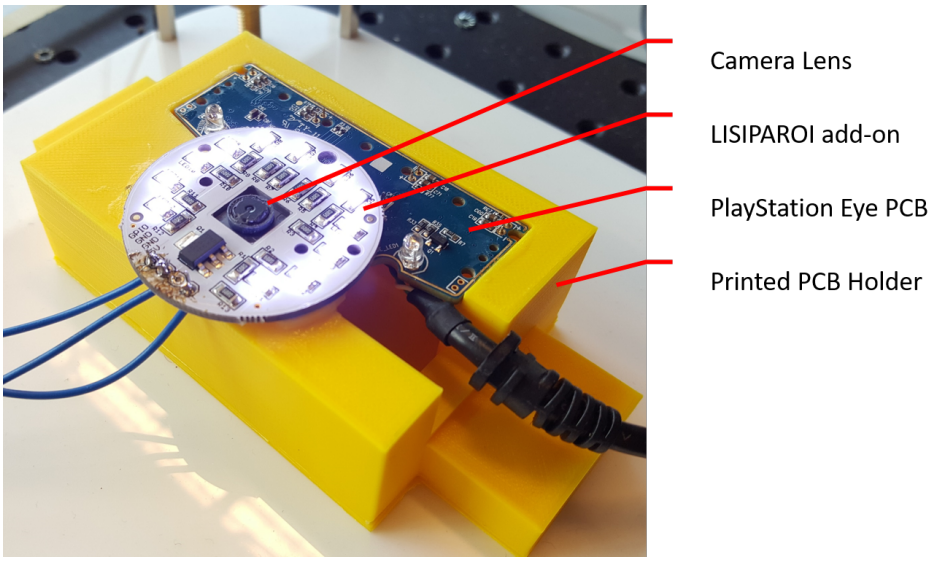


Figure C-5: Photograph of the image sensor, illumination add-on, printed PCB-holder.

size of the object:

$$M = \frac{h_i}{h_o} = 0.46 \tag{C-1}$$

Without using binning, the resolution of a single pixel is given by the image width seen by a single pixel. Since the object width of 8.75 mm is seen by 640 pixels the resolution of the camera system in this configuration is given by $8.75mm/640$ corresponding to $13.7\mu m$. However, by reading out 4 pixels as a single pixel, effectively using 320x240 pixels, the resolution is given by $8.75mm/320$ which is $27.3\mu m$.

C-5 Sensor System Results

The results of the sensor system are treated in this section. Three different camera resolutions are considered in these results in terms of delay and measurement dispersion. For the multi threading results and overall time delay results only one camera resolution, which is used in the final design, is considered.

C-5-1 Resolution versus Delay in ArUco

In figure C-6 the delay, which is found in only the part of the ArUco software, is seen in a normalized histogram at three different resolutions. In this delay data capturing an image and retrieving the data from the PlayStation Eye are not considered. Using the maximum amount of pixels (640x480) the largest delay is found with a mean value of 6.2ms and a σ of 1.5ms. For a resolution of 320x240 this delay is reduced to a mean value of 3ms and a σ of 0.6ms. Surprisingly, reducing the resolution further to a resolution of 160x120 does not affect the delay found in the script much ($\mu = 3$ms, $\sigma = 0.6$ms).

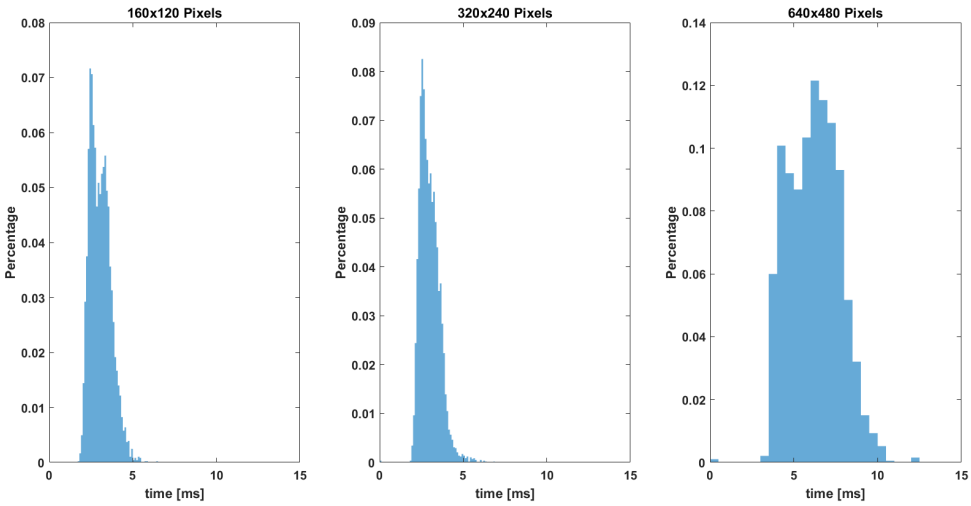


Figure C-6: Time delay of the ArUco process at different resolutions of the camera.

C-5-2 Measurement Dispersion at Different Resolutions

In figure C-7 the measurement dispersion for translations at different resolutions are seen. The measurement dispersion is increasing with almost a factor two comparing the 320x240 to the 240x180. No direct explanation was found for the fact that the 320x240 resolution shows a smaller dispersion compared to that of the 640x480 resolution. In figure C-8 the sensor signal for rotations can be seen for some arbitrary movement of the mover. In the left plot jumps in the sensor signal are observed for two different camera resolutions. In the right figure of C-8 a close up is seen of the sensor signal between one of those jumps. It is seen that the resolution of the sensor, between these jumps, is in the region of 10^{-3} degree. However,

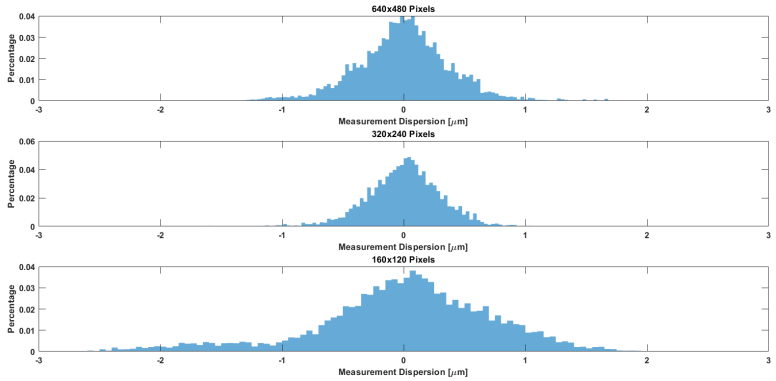


Figure C-7: Measurement dispersion of the sensor operating at different resolutions of the camera. The standard deviation (σ) of these measurements are 0.39, 0.32 and 0.73μm for 640x480, 320x240 and 180x120 respectively.

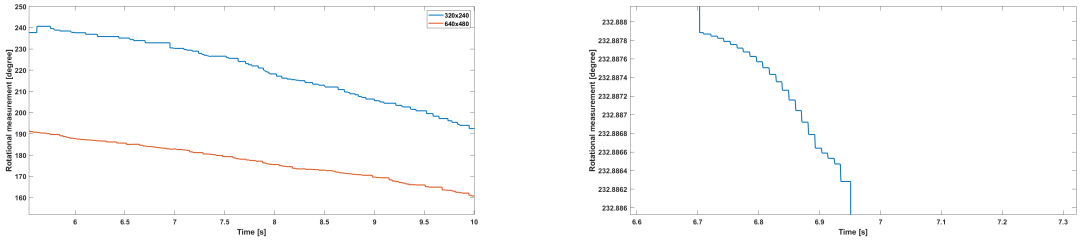


Figure C-8: Rotational Sensor signal. Arbitrary rotations of the mover and the corresponding sensor signal over time. The signal shows jumps in the value approximately 1 degree (left). Close up of the rotation signal which shows a high resolution for a small rotation.

these jumps cause a problem in the overall sensing and control of the rotation of the mover. These phenomena is most likely caused by the distorted image which gets fed to the ArUco software. As can be seen in figure C-3 the image is slightly curved which is caused by the lens which was used in the system. Different methods were explored to solve this issue such as different marker maps and calibrations of the software but none was able to solve this issue. Implementing another lens, which time did not allow, is most likely to resolve or at least reduce these jumps.

C-5-3 Delay found in the software pipeline

The pipeline of the software is divided in to three different sub processes to analyze the contribution of these processes to the overall delay. The following processes are considered:

- **Extract Image**, this is the part in the script where an image is extracted from the PlayStation Eye camera. The camera is only able to send the information to the laptop if a new image is ready to be send.
- **Detect Image**, this process consists of applying the ArUco software to a single image.

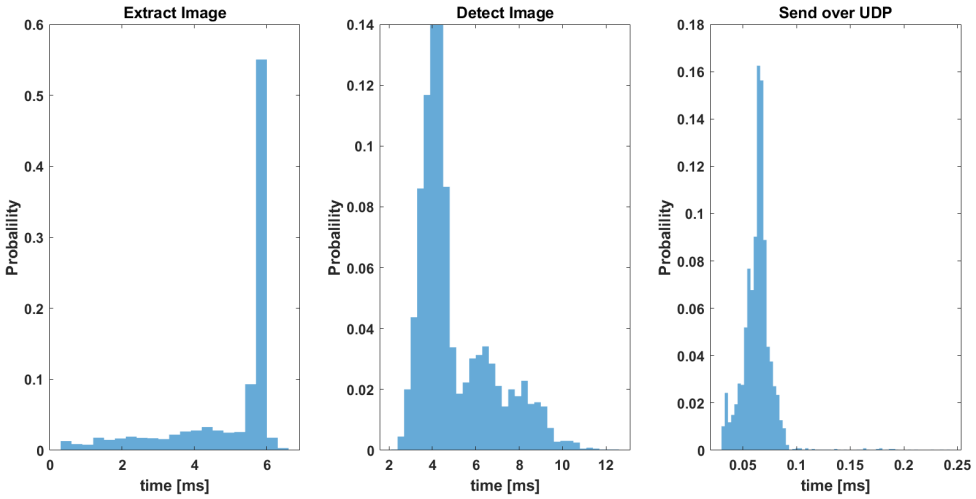


Figure C-9: Delay found within sensor algorithm without the use of multi threading. The time it takes to receive the image from the camera (left), delay caused by ArUco (middle), delay cause by transmitting the position information using UDP (right).

- **Send over UDP**, this represents the time it takes to send the different locations (x, y, θ) to the Raspberry Pi.

The results of the delays found within the software pipeline are seen in figure C-9. Detecting the markers and extracting an image contributes the significant amount of delay into the pipeline. Sending the information to the controller contributes only a small part of the delay. The total mean delay was found to be 10.1ms with a σ of 2.3ms.

C-5-4 Sample rate measurements using Raspberry Pi

During measurements it was found that the internal clock of the C++ program that was used was not always reliable. In order to confirm the delay data, measured with C++, the sample rate of the sensor was measured with the controller (Raspberry Pi). The sample rate of the controller itself was validated with one of the oscilloscopes available in the laboratory. The measured sample rate of the sensor can be seen in figure C-10. It must be noted that for these measurements the mover was given a velocity which impacts the delay as well. The mean sample rate, measured with the controller is 10.8ms with a σ of 3.9ms.

C-5-5 Multithreading for increased sample rate

In order to increase the sample rate of the sensor system a method called multithreading was applied [13]. Since the laptop was not exploiting all the CPU power this method could be employed. With multithreading it is able to start multiple processes at the same. For the sensor pipeline this basically means the every time an image is available the program starts a new thread in order to detect the markers and retrieve the position of the mover.
In figure C-11 the results of multithreading can be seen. The frame rate was measured over

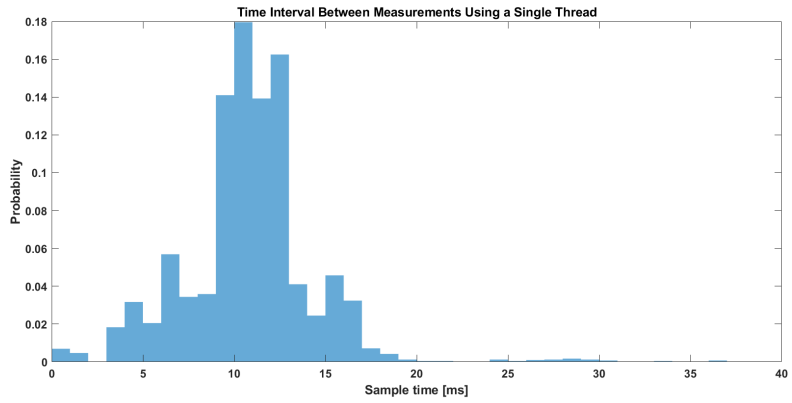


Figure C-10: Interval between measurements which is measured via the Raspberry Pi without the use of multi threading.

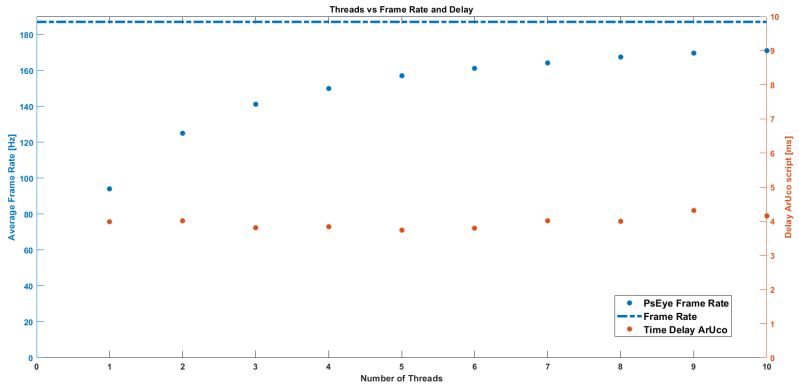


Figure C-11: Frame rate (left axis) and mean time delay of ArUco (right axis) versus the number of threads using a resolution of 320x240 pixels of the camera.

an interval of ten seconds using a range of 1 to 10 threads. As the number of threads increases the frame rate of the sensor converges closer to the actual frame rate of the camera. For these measurements the delay found in ArUco was measured as well to see if the overall delay per measurement increased. In can be seen that applying multithreading barely affects the delay found in ArUco. By applying multithreading the delay in the sensor is not reduced but only the frame rate of the sensor signal is increased.

C-5-6 Limitation Velocity

A limitation was found in the maximum velocity of the mover at which the ArUco algorithm is able to recognize the images. This limitation is derived in appendix D, see figure D-4. The limitation in the speed is related to the shutter time of the camera. The shutter time of the camera is given by:

$$T_{shutter} = 1/fps \approx 5ms \tag{C-2}$$

At a speed of $60mm/s$ the displacement during a single image capture therefore is $60mm/s * 0.005s = 300\mu m$. Given that the length of a single bin of a marker is $250\mu m$ a black bin could

be at the same position as a white neighbouring bin in this time interval. The resulting part of the image would give a blurred output with the mean color value of black and white, grey.

$$V_{max} = l_{bin} \times T_{shutter} = fps \times l_{bin} \tag{C-3}$$

Using this derivation the maximum speed of the mover at which the sensor is able to retrieve its position is given by:

$$V_{max} = 250\mu m \times 187 \approx 45mm/s \tag{C-4}$$

Table C-1: Specifications of the sensor system

Camera	PlayStation Eye @320x240 resolution
Image length seen by a single pixel	27.3μm
ArUco	3.0.1, DM_FAST mode
Maximum Speed	45mm/s
Static delay	μ =10.1ms , σ = 2.3ms
Average sample rate	167FPS
Translational measurement Dispersion	σ = 0.32μm
Magnification	0.48

C-6 Specifications of the Final Sensor System

The delay was found to be larger for higher camera resolutions. However the difference in delay between 320x240 and 160x120 pixels was small. For the measurement dispersion of the sensor signal the resolution of 320x240 gave the best results. Therefore, for the final sensor system a camera resolution of 320x240 is used.
Since the measurements in rotations did not allow an accurate measurement of the rotation this degree of freedom is not considered for the control. The challenges encountered in the measurement in rotation were new in this project and were not encountered in [1].
In the final sensor system 12 threads are employed which results in an average frame rate of 167fps during operation. Applying multithreading led to some challenges in the stability of the software since it crashed occasionally due to segmentation faults. The origin of this problem lies in the fact that some memory is shared and accessed by the different threads. This issue was resolved by locking the memory for the specific thread that used it (see: http://www.cplusplus.com/reference/mutex/mutex/). The overall specifications of the sensor system can be seen in table C-1.

Appendix D

Control Design

This appendix will give an overview of the controllers which were designed in this project. To do so, the problem of a time delay in a control system is addressed such that it is clear to what extend a time delay affects a control system.

D-1 Time Delays

Time delays are often not considered when a controller is designed. Most frequently the time delay is either negligible small or is accounted for in the phase margin of the controller design. Accounting for a time delay in controller design can be difficult since a time delay is represented by an infinite dimensional differential equation. For controller design in time domain often a time delay approximation, such as the padé approximation, used to approximate the delay. In the frequency domain a time delay, of τ seconds, is given by:

$$e^{-\tau j\omega} \tag{D-1}$$

To inspect the effect of a time delay in the frequency domain the magnitude and phase of a time delay is considered:

$$|e^{-\tau j\omega}| = 1 \tag{D-2}$$

and the phase (in radians) is given by:

$$\angle e^{-\tau j\omega} = -\tau\omega \tag{D-3}$$

As is seen the magnitude is not affected by a time delay but a time delay adds phase lag which is proportional to frequency and delay. For this reason the phase margin is sometimes also considered as the delay margin. For a crossover frequency ω_c the phase margin is directly related to the maximum allowable delay which will destabilize the system by $\tau_{max} = PM/\omega_c$. For a standard negative feedback loop the loopgain of a system with a time delay is given by:

$$L = C(s)G(s)e^{-\tau s} \tag{D-4}$$

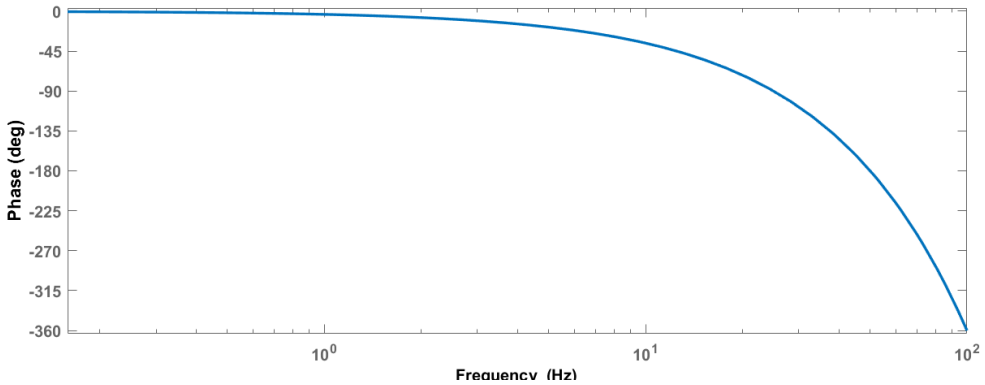


Figure D-1: Phase of a pure 0.01 second delay

To demonstrate the phase due to the delay the phase is plotted against frequency for the loopgain with a unity C(s) and G(s) in figure D-1. It is directly clear that this phase imposes a limited on the achievable control bandwidth of a system since there is a limitation in phase which linear controllers can compensate for. In order to inspect the limitations in the microscopy stage considered in this project the system identification is treated in the next section.

D-2 System Identification

Modelling of a system may cause problems due to modelling errors or neglected dynamics. Therefore a system identification was performed to get an accurate model of the gain and phase of the system. In [1] the system for a single degree of freedom for translations x-direction was modelled by:

$$G_x(s) = \frac{x}{F} = \frac{1}{ms^2 + c_x s} \tag{D-5}$$

Where m is the mass of the mover and c_x the damping coefficient of the ferrofluid. For rotations the system can be modelled by:

$$G_\theta = \frac{\theta}{Torque} = \frac{1}{Is^2 + c_\theta s} \tag{D-6}$$

With I the rotational inertia and c_θ the rotational damping. Since estimate of the rotation was not reliable the rotational control and identification is not included.

D-2-1 Model Identification

For the identification of the model the computer vision sensor is used. Since the sensor operates at a frequency of 167Hz a maximum input frequency of 80Hz was selected. According to the Nyquist sampling rate a signal can be reconstructed if the sampling frequency, f_{sample} is at least twice the maximum frequency of the analogue signal.

$$f_{sample} < \frac{1}{2} f_{signal} \tag{D-7}$$

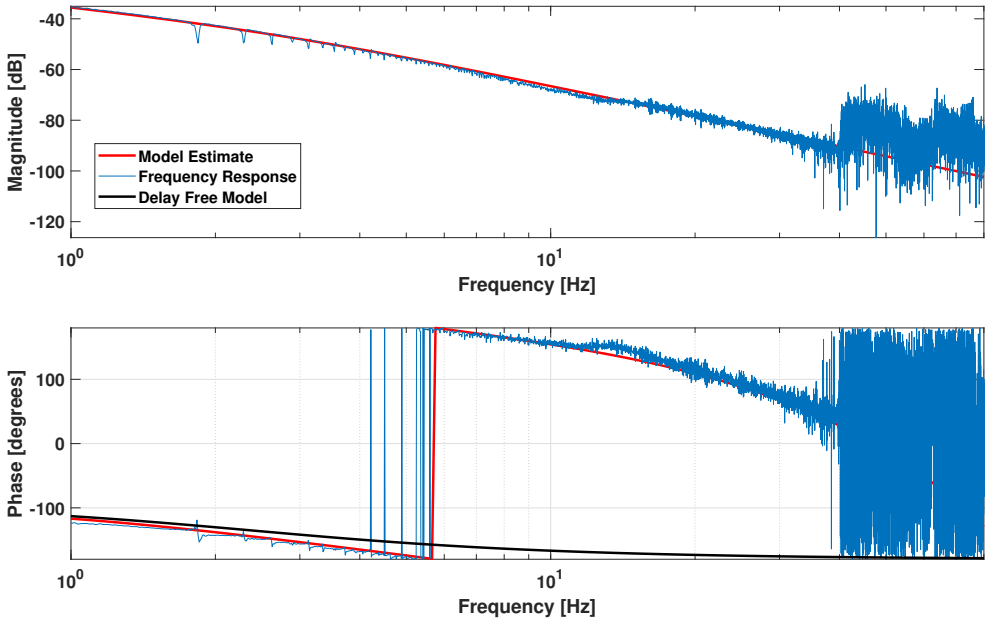


Figure D-2: Bode plot of the frequency response and the estimated model.

As an input, for the system identification, a chirp signal was used with a frequency sweep between 1 and 80Hz. This chirp signal was supplied to the system multiple times in a row and the data of 250 seconds was collected for the system identification.
In figure D-2 the results can be found for the system identification. The model of equation D-5 was fit on the data manually by:

- First matching the gain plot for the parameters specified in equation D-5
- Matching the phase data to the model by introducing a delay in the model: $G_x(s) = \frac{e^{-\tau s}}{ms^2+c_x s}$

The estimation of relative accurate between the frequencies of 1 and 40Hz. At a frequency beyond 40 Hz the system is affected by noise which is most likely caused by the limited sample rate of the sensor used for the identification. At around 14Hz there is a slight hill in the phase for the input output data. The phase shift, caused by the sensor, can also be clearly seen comparing the transfer function estimate of the data and the delay free model in D-2.
Since the sensor was unreliable for rotations (see figure C-8) a system identification for was not performed for the rotation.

D-2-2 Signal Reconstruction

Since the sensor signal inherits a digital sampling behaviour at with a non-uniform sampling frequency the signal is reconstructed in order to get a more accurate estimation of the state of the mover.
By default the signal is held by simulink when it enters the loop and creates a zero order hold

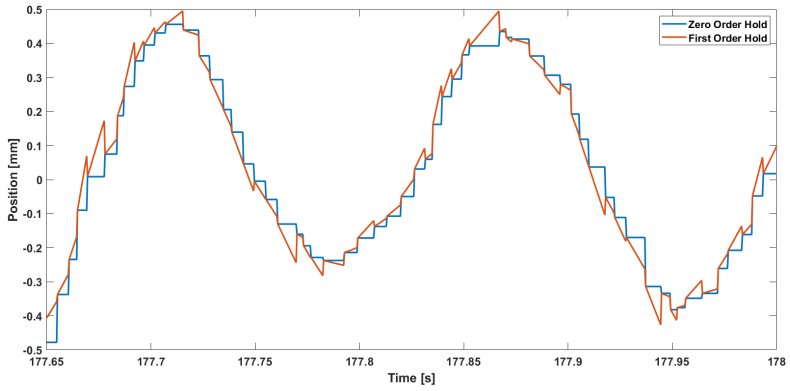


Figure D-3: Signal reconstruction using First Order Hold and Zero Order Hold

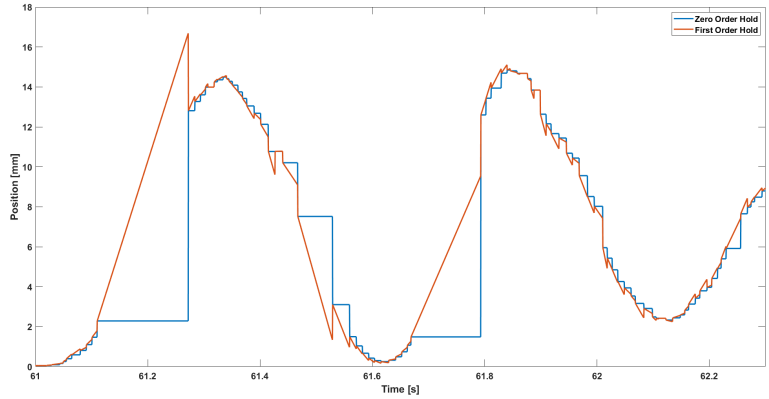


Figure D-4: A situation where the computer vision sensor is not able to retrieve the position of the images due to the velocity which causes image blurring.

behaviour. In order to get a more continuous signal first order hold is applied to the sensor signal (see figure D-3. During a period of constant velocity the first order hold has a more reliable estimation of the state. In some cases it is clear that zero order hold has a smaller error at the moment the next sample comes in. In figure D-4 the zero order hold and first order hold signals are seen in the case that the sensor is not able to retrieve a position of the images. The speed at which the sensor is not able to recognize the markers and retrieve the position was found to occur at mover velocity of 60mm/s and higher. In the case that sensor signal might be lost the first order hold gives a smaller error than the zero order hold.

D-3 PID Benchmark

With the identified model a controller can be designed. At first a benchmark controller is designed to inspect the performance of the system using a PID controller. Knowing that a PID controller can add around 55 degrees of phase to the system the maximum controller bandwidth can be derived. As a rule thumb a phase margin of 30° is desired. Two PID

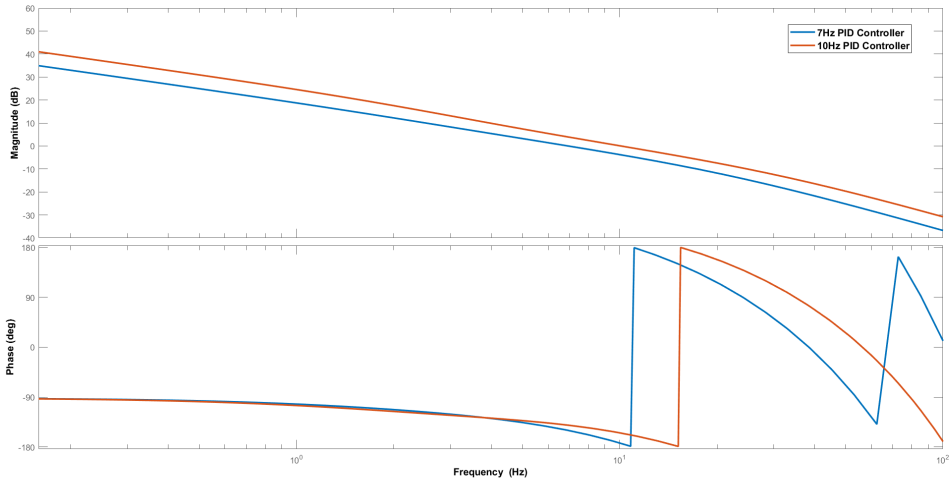


Figure D-5: Bode plots for the system $G(s)$ and the loop gains $L = C(s)G(s)$ for two controller which were design for a control bandwidth of 7 and 10Hz taking into account the maximum delay of 15ms and the average delay of 10ms respectively.

controllers are implemented to inspect the effect of the delay. A controller with a 10Hz cross-over frequency is implemented which takes into account the average delay of 10ms. A second controller with a 7Hz cross-over frequency can be designed considering a maximum delay of 15ms.

D-3-1 PID synthesis

Since the system has a limited amount of current it can supply a PID block from Simulink is used to implement the controller since it allows outputs saturation without winding up the controller output. The controller was designed by the rule of thumb given in [7] and converted to a parallel PID form. The loopgain L, $(C(s)G(s)e^{-\tau s})$ of the corresponding controllers are seen in figure D-5. For the controller with a 7Hz cross-over frequency and taking into account a maximum delay of 15ms the phase margin is 30°. For the 10Hz controller taking into account the average delay the phase margin is 26°.
The following remark is given about the filter coefficient w_t of the derivative term in the PID controller. Increasing w_t, thus increasing the cut-off frequency of the low-pass filter, has a positive effect on the phase margin. However, the controller became unreliable since a large w_t coefficient ($> 10^3$) would crash the Raspberry Pi.

Table D-1: Controller Parameters (in radians) for the tested control bandwidths (in Hz) of $C(s) = k_p(1 + \frac{w_i}{s})(\frac{s/w_d+1}{s/w_t+1})(\frac{1}{s/w_f+1})$

BW	k_p	w_i	w_d	w_t	w_f
7Hz	440	4.8	14.5	131.9	440
10Hz	663	6.8	20.7	188	1885

D-3-2 PID Results

The controllers given in table D-1 were implemented and tested on a step response. The results of the 7Hz PID controller taking the maximum delay into account is seen in figure D-6. The step response has an overshoot of around 37% and shows some oscillations. In figure

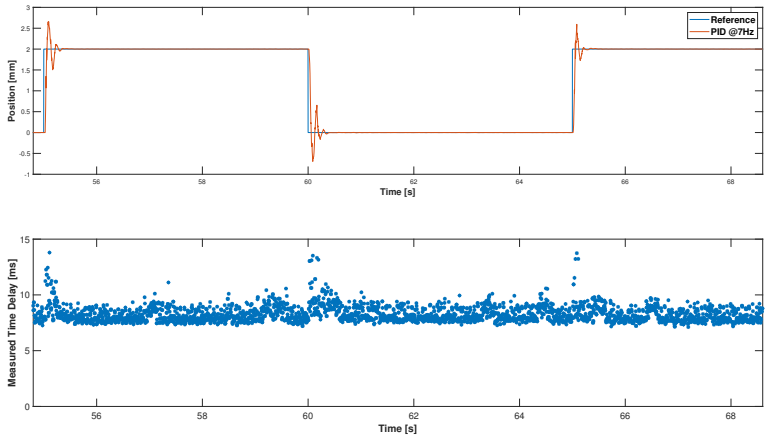


Figure D-6: Steps

D-7 the response for the 10Hz controller which was designed for an average delay of 10ms is seen along with the measured delay. For the first step in the figure the response has a slight improvement compared to the 7Hz controller. However, for the second response it is clear that the performance is affected by oscillations. By inspecting the time delay in the figure it is clear that the oscillations are caused by a temporarily increase in delay during the stepping motion. The increase in delay is caused by image blurring. When the mover has a certain velocity with respect to the camera the image gets blurred. Detection of a blurred image causes an increase in the detection time of the ArUco algorithm. The response of the output

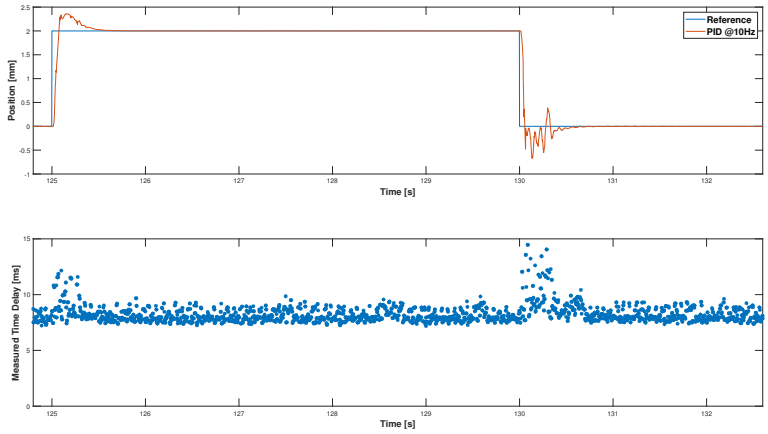


Figure D-7: Steps

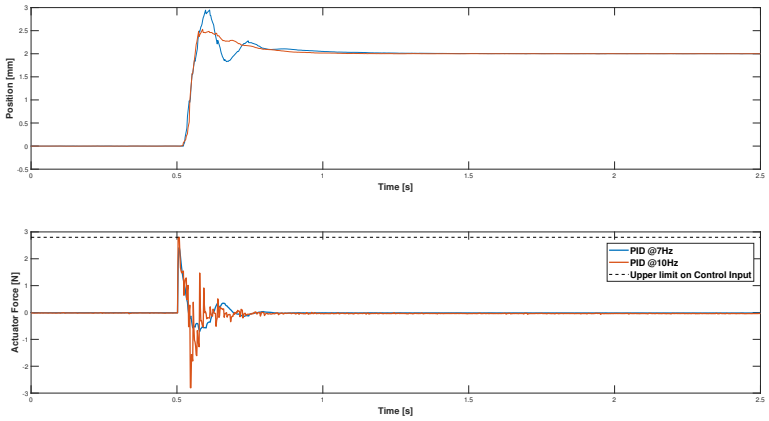


Figure D-8: Comparison of a step response for a 7Hz and 10Hz controller. Due to the limitation in current it is not possible to further improve the rise time.

and actuator force, for both controllers, can be seen in figure D-8. The rise time, defined as the time from 10% to 90% of the reference value, for both controllers are in the order of 30ms. The reason for the rise times being similar is that the controller input is limited to a maximum force of 2.8N due to the limitation in the current supply.

D-4 Time Delay Compensation

From figure D-2 it is clear that a time delay causes problems within feedback control. Control of time delay systems can be difficult and many research has been dedicated to this subject, see [14],[15],[16] and [17]. The emerging field of Networked Control Systems (NCS) has a specific interest in the control of a network induced delay. The properties of a network delay has similar properties to that of the delay found in the computer vision sensor.
A NCS method to compensate for time varying delay was found in [18] but during the project it was concluded that this control architecture acts as a pure feed forward controller. Due to this finding it was decided to implement a more conventional type of delay compensation.
The most accepted control architecture in literature is the Smith predictor which was introduced by J. M. Smith [19]. Since it is widely accepted many research on the smith predictor is available.

Smith Predictor

In D-9 the control architecture of the smith predictor is seen. For the implementation of a smith predictor a plant model $G_m(s)$ and a delay estimate $\hat{\tau}$ is needed. The basic idea behind smith predictor is that if the plant model and the delay estimate is are equal the closed loop transfer function becomes:

$$\frac{y}{r} = \frac{C(s)G(s)e^{\tau s}}{1 + C(s)G_m(s)} \tag{D-8}$$

Under these assumption it is the loop gain of the system is delay free and the controller is design for the delay free plant. Since the loopgain, the denominator is free of the delay the controller C(s) can be designed for the delay free plant $G(s)$. Under these conditions there would be no limitation in control bandwidth for the plant which is considered in this thesis:

$$G_x(s) = \frac{1}{0.464s^2 + 6.9s} \tag{D-9}$$

The problem however, is that a mismatch in the delay causes problems in stability. Many studies focus on the delay sensitivity of the smith predictor. In order to account for the time varying delay a Filtered Smith Predictor is implemented.

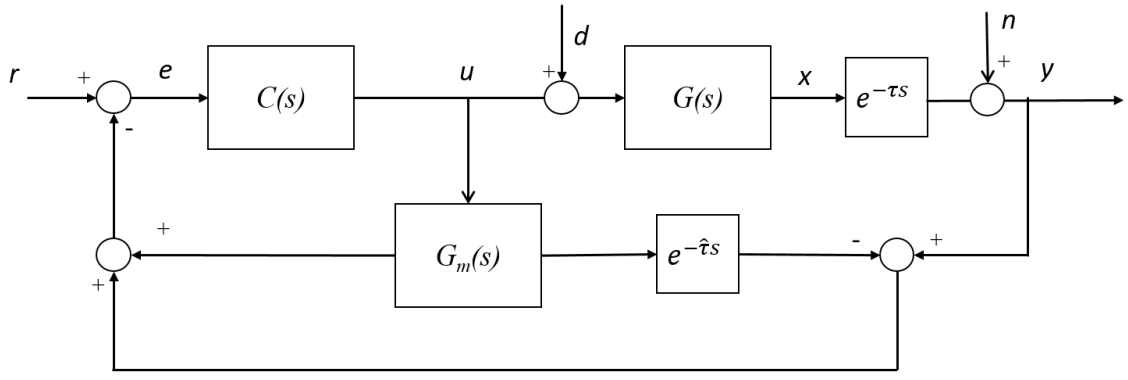


Figure D-9: Smith Predictor

Filtered Smith Predictor

In figure D-10 the Filtered Smith Predictor architecture is shown. By introducing an additional filter F(s) the controller is more stable against variations in the time delay [20]. The filter F(s) is implemented as a low-pass filter however multiple types of filters are mentioned in literature. Another advantage is that the low-pass filter can have a cut-off frequency close to the control bandwidth which allows additional filtering of the sensor signal. The closed

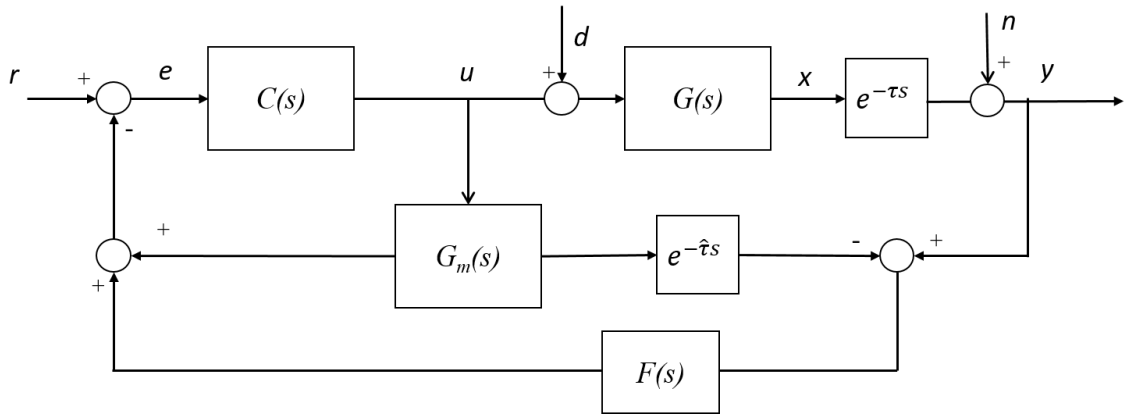


Figure D-10: Smith Predictor

transfer functions of the Filtered Smith Predictor are given by:

$$\frac{y}{r} = \frac{C(s)G(s)^{-\tau s}}{1 + \frac{C(s)G(s)e^{-\tau s}F(s)}{1+C(s)G_m(s)(1-F(s)e^{-\hat{\tau}s})}} \tag{D-10}$$

$$\frac{y}{n} = \frac{1}{1 + \frac{C(s)G(s)e^{-\tau s}F(s)}{1+C(s)G_m(s)(1-F(s)e^{-\hat{\tau}s})}} \tag{D-11}$$

Time Domain Analysis

The Filtered Smith Predictor is analyzed within the time domain to inspect the performance and the effect of the filter. In figure D-11 the response of a filtered smith predictor is compared with a normal smith predictor. For a constant time delay mismatch of 12.5% the FSP shows a slight improvement in the step response. Due to the filter F(s) the noise rejection is slightly delayed but has an improved performance in terms of oscillations.

In figure D-12 the effect is seen of different delays within the plant. The red arrow indicates the increasing delay in the plant. For this result a constant $\hat{\tau}$ was used of 10ms and the plant delay is increased from 10ms to 15ms in steps of 1 ms. Increasing the delay mismatch leads to more oscillations and eventually it will lead to an unstable controller.

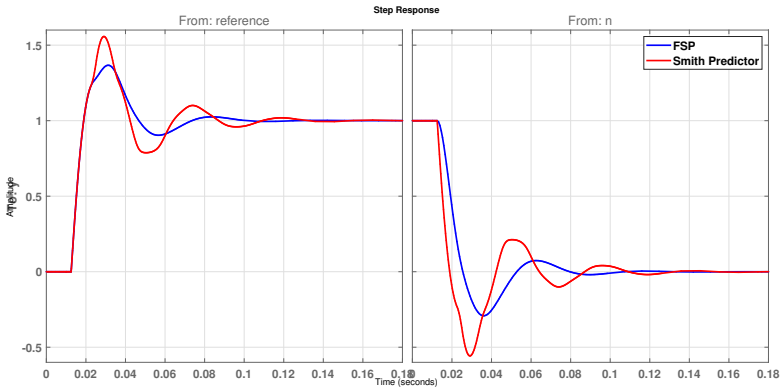


Figure D-11: Filtered Smith Predictor response versus a normal Smith Predictor

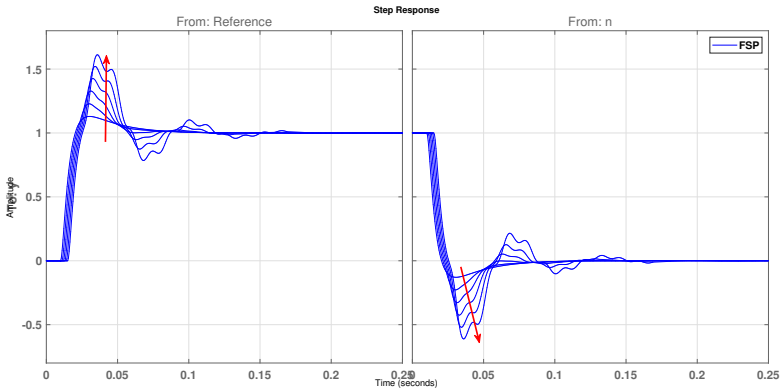


Figure D-12: Response of the FSP with a delay estimate of 10ms. The increasing plant delay (10-15ms) is indicated with a red arrow.

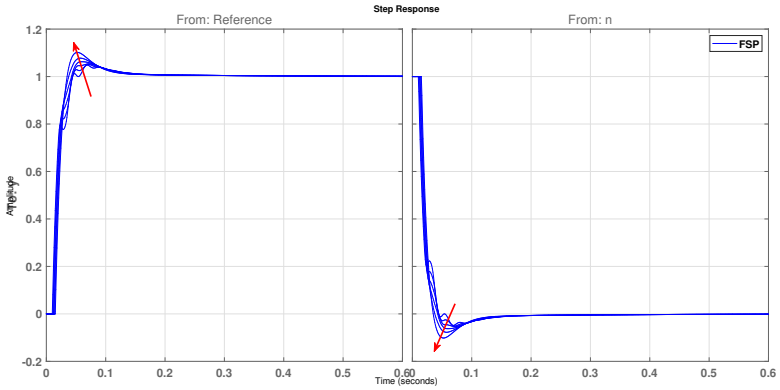


Figure D-13: Response of the FSP with a delay estimate of 15ms. The increasing plant delay (10-15ms) is indicated with a red arrow.

In figure D-13 a similar simulation as in figure D-12 is shown. However, instead of using an estimated delay of 10ms this model uses the maximum delay estimate of 15ms. It is seen that the response is improved significantly by using a larger delay estimate. This indicates that the FSP performance can be improved by not using the average delay but by using a maximum delay.

Process Disturbance Rejection

The smith predictor is known to have a steady state error when a constant process disturbance is acting on the system. If the microscopy stage is not leveled precisely the gravity force acts as a constant process disturbance. Since the gravity is constant the steady state error would be constant, or at least linear, as well. To inspect this a step response of the Filtered Smith Predictor is seen in figure D-14. The overall step response is stable and seems to have a constant steady state error. However, with a closer look to the settling response it is seen that the response has a drift over multiple seconds.
Considering the gravity force does not change the drift is explained due to the trail forming of the ferrofluid bearing. If the mover steps from a certain position to another position a trail of ferrofluid is left behind. Since the ferrofluid is attracted by the magnets there is a force exerted no the mover. Over time the ferrofluid gets pulled to the magnet again and the resultant force on the mover is changing.
A constant steady state error would form no problem for the microscopy stage however the drift is unwanted. Therefore another disturbance observer is added to the control architecture, see figure D-15. The additional disturbance observer compensates the disturbance by estimating the disturbance by:

$$\hat{d} = \frac{G_m^{-1}(s)}{Q(s)} - \frac{u}{Q(s)} \tag{D-12}$$

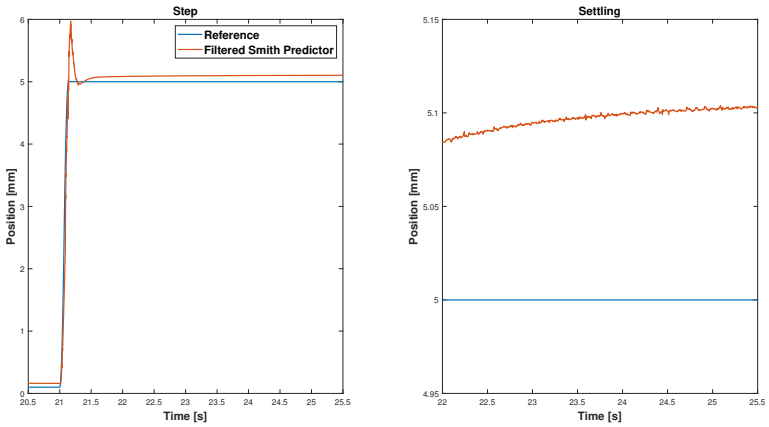


Figure D-14: Step response of the microscopy with a Filtered Smith Predictor and a close up of the Settling Response.

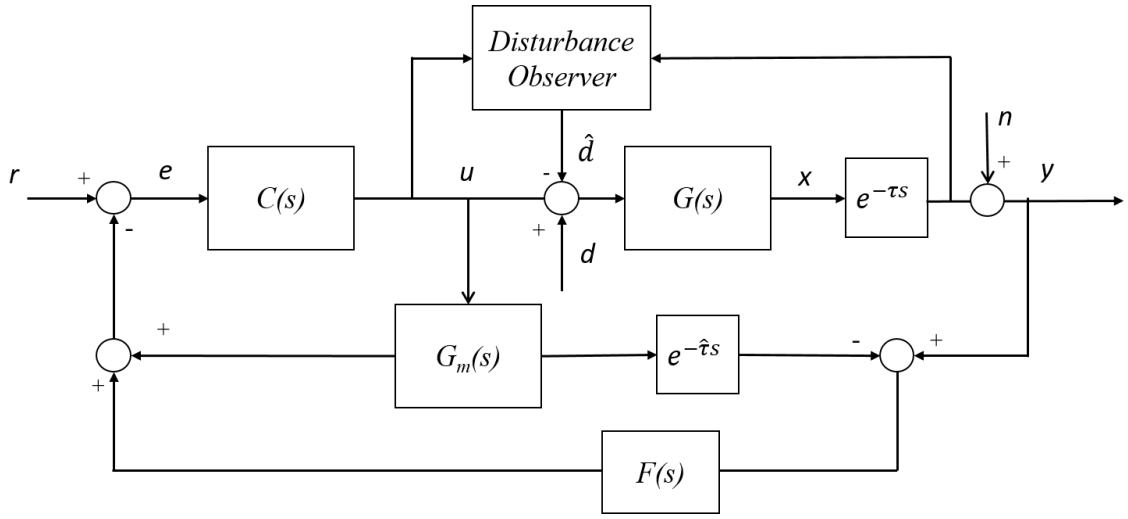


Figure D-15: 2-DOF Filtered Smith Predictor

D-5 Controller Design and Results

In this section the final controller which is implemented in on the system is treated along with the results. The step response of three different controllers are treated in order to see the effect at higher bandwidths.

D-5-1 Controller Overview

The controller design for any kind of Smith Predictor is generally done by designing the controller for the non-delayed plant. Since the time domain analysis showed that in terms of oscillations an over estimated delay is beneficial a delay estimate of 13ms was used for the Filtered Smith Predictor.
A minimum target bandwidth of 10Hz was selected for the controller as a starting point. Higher bandwidth controllers were designed and tested as well in order to inspect the limitations and of the system. For the target bandwidth of 10Hz and the maximum bandwidth of 15Hz the loop gains are seen in figure D-16. In figure D-17 the closed loop control sensitivity

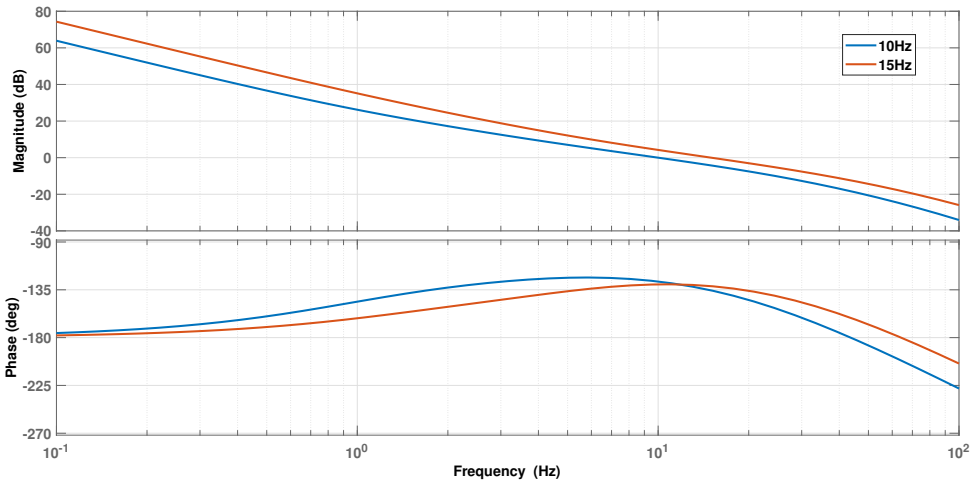


Figure D-16: Loop gain C(s)*G(s) which is used to design the controller. Not that the controllers are designed for the non-delayed model.

functions (see equations D-10 and D-11) are depicted of the controllers. For the increased 15Hz controller the sensitivity peak is 1.95 and for the 10Hz controller 1.75. The closed loop control bandwidth for the given controllers, defined as the -3dB crossing of the complementary sensitivity functions, are 11.3 and 17.9 for. Suggesting that there is an increase in control bandwidth applying a PID in the suggested FSP architecture.

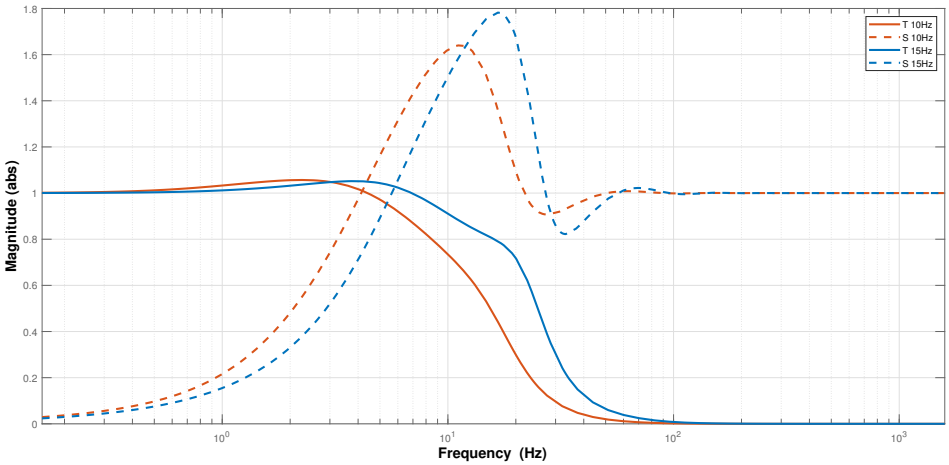


Figure D-17: Closed Loop control sensitivity of the designed controllers.

Remark: Delay Update

Considering that the delay is measured within the computer vision algorithm it was considered to use this delay and feed it into to the delay estimate of the Smith Predictor. Using this measurement a more accurate time delay estimation is used in the controller.
Multiple experiments were conducted in order to validate that the performance was improved using a delay update. From figure D-7 it is clear that the variations of in the time delay are considerably large during stepping motion. Therefore an additional low-pass filter, in order to average the delay was implemented as well.
Updating the time delay, with or without the additional low-pass filter, did not increase the stability or the performance of the system. The results are therefore not included.

D-5-2 Results

The step response for multiple steps and time delay measurements of the 10Hz Filtered Smith Predictor is seen in figure D-18. It is seen that the delay increases during the stepping motions. Since the delay affects the response of the system not all the steps are similar in terms of overshoot and rise time. The 3σ positioning precision for this controller is 1.46μm which was measured during a 60s interval in which the controller had to keep the mover at the origin. In figure D-19 the step results for the 15Hz controller are seen along with the

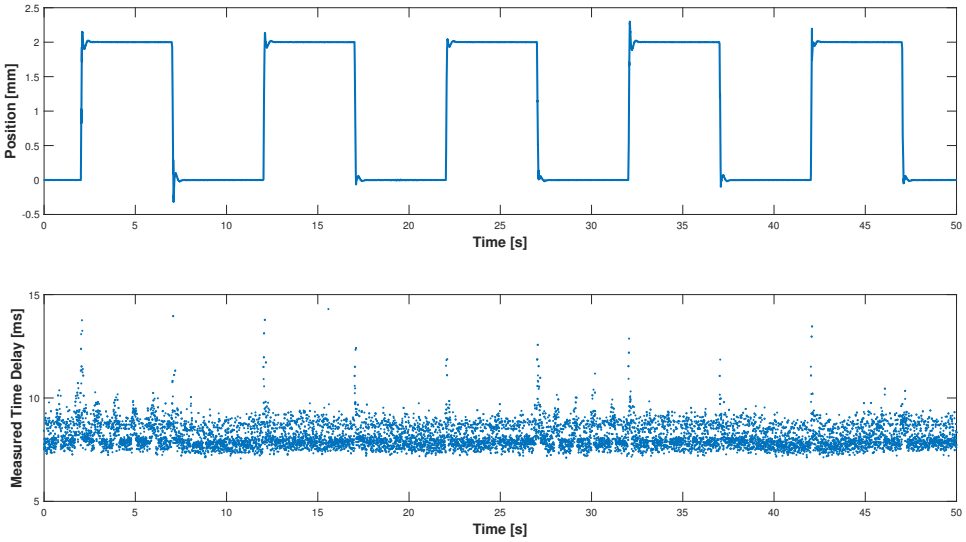


Figure D-18: Step responses of the **10**Hz Filtered Smith Predictor with time delay measurements..

time delay measurements. For this controller the same is said about the overshoot in terms of inconsistency due to the sensor delay. For the positioning stability the same precision is obtained as the 10Hz FSP controller, 1.46μm. Note that the 3σ measurement dispersion is 0.96 μm which is closed to the positioning precision of the controllers.

At higher control bandwidths the controller gets more sensitive to the time varying delay. To illustrate this the step responses for a 20Hz controller are depicted in figure D-20. For the first steps 2 steps the response is normal. But when the controller encounters larger delays oscillations occur. Since these oscillations cause additional image blurring, the delay increases which leads to additional image blurring.

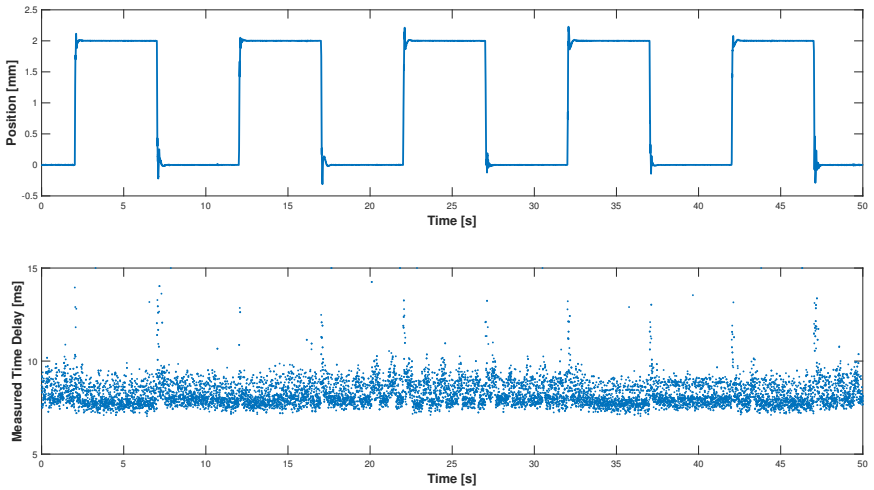


Figure D-19: Step responses for a **15**Hz Filtered Smith Predictor with time delay measurements.

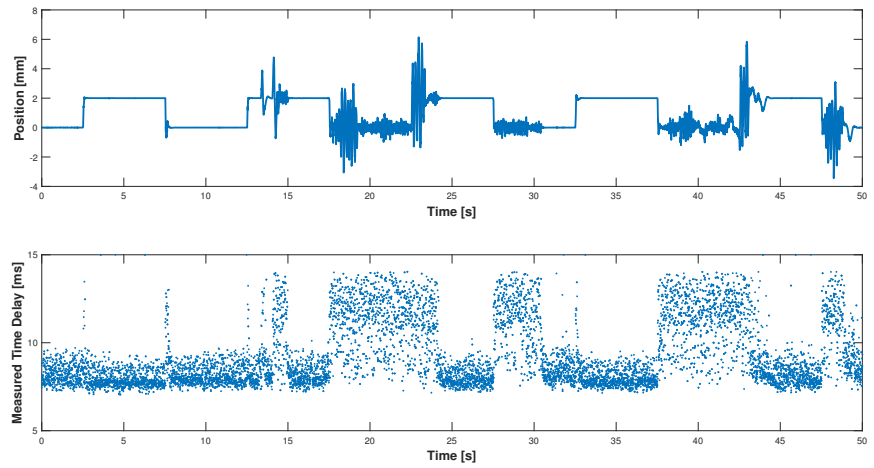


Figure D-20: Step responses for a **20**Hz Filtered Smith Predictor with time delay measurements.

Appendix E

Operation Manual

This operation manual is written in order to make the system accessible to others who may want to conduct experiments or research on this setup or subject. Since the system uses various kinds of software programs and languages this appendix will act as a guide on how to start up and change parameters in the system. The operation manual is divided into separate sections to show how to operate the different subsystems individually.

E-1 Sensor Subsystem

The sensor subsystem uses the following hardware systems in order to detect the position of the mover and send to the Raspberry Pi:

- PlayStation Eye
- Laptop running on Ubuntu 16.04
- Router
- Raspberry Pi
- LISIPAROI header for the Raspberry Pi

E-1-1 PlayStation Eye

The PlayStation Eye is an affordable webcam which can be easily operated through the necessary commands in C++. The PlayStation Eye comes with a USB cable which can directly plugged into the laptop. In order for the ArUco program to work appropriately the marker map was illuminated with a LISIPAROI header.
During the project effort was made to access the PlayStation Eye using a laptop running on Windows 10. There are examples where people are successful in operating the PlayStation

Eye on Windows 10 using C++ however this imposed mayor difficulties during this project. Ubuntu is recommended in combination with the PlayStation Eye since no additional drivers have to be installed and it can be directly accessed with OpenCV. Some useful commands can be found the following enlistment which uses the video capture API from OpenCV.

```
1 TheVideoCapturer.set(CV_CAP_PROP_FPS,fps);          // Set frame rate
2 TheVideoCapturer.set(CV_CAP_PROP_FRAME_HEIGHT, 240);// Set frame height
3 TheVideoCapturer.set(CV_CAP_PROP_FRAME_WIDTH, 320); // Set frame width
4 TheVideoCapturer >> Image;                          // Retrieve an Image
```

E-1-2 Ubuntu

As mentioned the ArUco software is operated through Ubuntu. The following software programs are needed in order to operate the PlayStation Eye and send the information to Raspberry Pi:

- CMake
- OpenCV
- ArUco

Ubuntu and OpenCV can be simple installed by consulting the internet. For the installation of ArUco follow the steps which are found on the official webiste of ArUco: https://www.uco.es/investiga/grupos/ava/node/26.
Three different examples, which come with the installation of ArUco, were used and modified in this project, namely:

- aruco_calibration
- aruco_create_markermap
- aruco_test_markermap

The examples can be accessed via the commands which are depicted in table E-1 which contains some other useful commands when operating Ubuntu or Debian (the OS for the Raspberry Pi). For further information one should visit the website of ArUco.
The Aruco_Test_Markermap.cpp example in the ArUco script was used as basis program for this thesis. A UDP communication protocol was implemented and as well some code to operate with the PlayStation Eye. In order for the Raspberry Pi to receive the information the router must be connected to both the laptop and the Raspberry Pi. It is recommend to turn of wifi during operation and only use the LAN connection. The Raspberry Pi package of simulink allows the use of a block diagram which receives information via UDP. In order to transmit and receive the information the port numbers of the sending script and the simulink block must match. The target is illuminated with the LISIPAROI which automatically turns on if the Raspberry Pi has power.

Table E-1: Table with some useful commands which can be used on both Debian (Raspberry Pi) and Ubuntu.

Command	Comment
cd FolderName	Go to FolderName directory
cd ..	go up one directory
./executable	execute the executable
sudo ./executable	execute executable as superuser
ctrl+alt+t	open a terminal
top	inspect running processes
sudo kill 2122	Shut down process with PID number 2122

Raspberry Pi and Simulink

It is recommended to use MATLAB 2017b or a more recent version of MATLAB to operate the Raspberry Pi via Windows. Operating the Raspberry Pi via a Linux distribution is not supported. Connecting the Raspberry Pi with MATLAB is straightforward with the tutorial provided by MathWorks.

Operating the LTC-2668

Since the operating the LTC-2668 is not straight forward an example is given in figure E-1. In order to to send a binary code to the digital to analogue converter the SPI Register Write Raspberry Pi block can be used that comes with the Raspberry Pi Simulink package. The block accepts a integer number and converts this to a 32-bit binary. The specific commands for the LTC-2668 can be found in the data sheet from the supplier. In order to merge three different binaries a vector concentenate block is used. After the three binaries are merged the binaries are converted back again to a decimal number. The illustrated example sends a command to the LTC-2668 to limit the output to a maximum of 5 Volts and a minimum of -5 Volts.
The output of the LTC2668-12 needs to be reset in order to stop the amplifiers from supplying current to the system. The current will produce heat and eventually this could destroy the system. In order to reset the LTC2668-12 the following can be done:

- Connect the pin CLR (clear) to ground with a conducting material. This will reset the LTC2668-12 entirely.
- Reset the output to zero by opening the reset program with the following command: sudo ./Reset_DAC_to_zero.elf.
- Disconnect the power supply to the LTC2668-12. Note that this does not reset the output set points but shuts off the power.

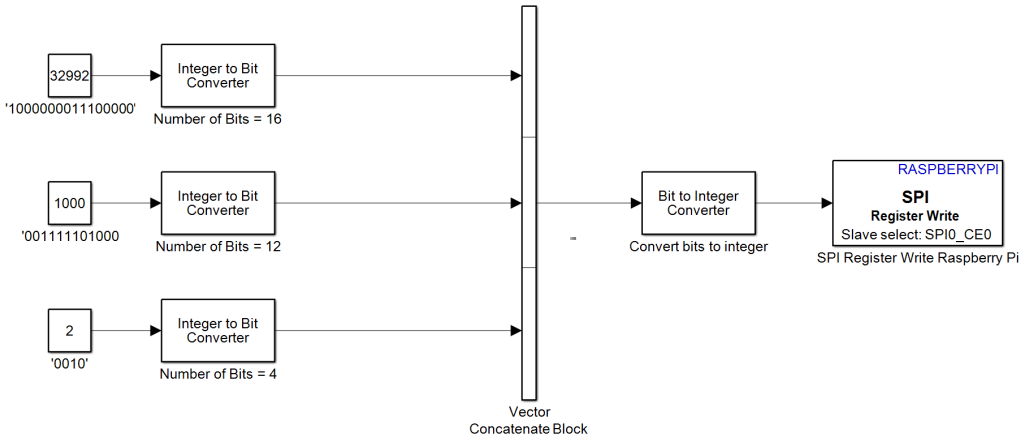


Figure E-1: Example on how to send '10000000111000000011111010000010' to the SPI interface of the raspberry with Simulink.

E-1-3 Starting the System

Before starting the system the following should be checked:

- LTC2668-12 connected to Amplifiers
- Laptop connected with Raspberry Pi through router
- LTC2668-12 Output set to zero

If all is connected the following steps are used to start up the system:

- Start up the sensor on Ubuntu using the following command: "sudo ./aruco_test_markermap live:1 map1515MIP36h12.yml CameraCalibrationFile.yml -s 0.0025"
- Start up the controller on the Raspberry by opening, for example, 10HzSmith.elf with the command: "sudo ./10HzSmith.elf". To stop the controller press CTRL+C in the terminal.

The ArUco algorithm needs a calibration file (CameraCalibrationFile.yml) and the configuration of the marker map(map1515MIP36h12.yml). Good luck!

Appendix F

Matlab Code for Filtered Smith Predictor

```
%% Filtered Smith Predictor
clear all; close all; clc;
s = tf('s');
wcf = 10*2*pi;            %Controller Bandwidth
Lwcf = 15*2*pi;           %Filter Cut-off
%Filter
F = 1/(s/Lwcf+1);         %Filter
F.InputName = 'dy';
F.OutputName = 'dp';
% Plant
P = exp(-0.015*s) * 1/(0.464*s^2+6.9*s);
P.InputName = 'u';
P.OutputName = 'y0';
% Plant Model
Gp = 1/(0.464*s^2+6.9*s);
Gp.InputName = 'u';
Gp.OutputName = 'yp';
%Plant Delay
Dp = exp(-0.01*s);
Dp.InputName = 'yp'; Dp.OutputName = 'y1';
% Overall plant
S1 = sumblk('ym = yp + dp');
S2 = sumblk('dy = y0 - y1');
Plant = connect(P,Gp,Dp,F,S1,S2,'u','ym');
% Design PID controller with 30Hz bandwidth and 60 degrees phase margin
Options = pidtuneOptions('PhaseMargin',60);
C = pidtune(Plant,pidstd(100,30,3000,100),wcf,Options);
C.InputName = 'e';
C.OutputName = 'u';
% Closing the loop
Sum1 = sumblk('e = r - yp - dp');
```

```
32 Sum2 = sumblk('y = y0 + n');
33 Sum3 = sumblk('dy = y - y1');
34 T = connect(P,Gp,Dp,C,F,Sum1,Sum2,Sum3,{'r','n'},'y');
35 % Perturbed models
36 P1 = exp(-0.01*s) * 1/(0.464*s^2+6.9*s);
37 P2 = exp(-0.0125*s) * 1/(0.464*s^2+6.9*s);
38 P3 =exp(-0.015*s) * 1/(0.464*s^2+6.9*s);
39 P4 =exp(-0.0175*s) * 1/(0.464*s^2+6.9*s);
40 P5 =exp(-0.02*s) * 1/(0.464*s^2+6.9*s);
41 Plants = stack(1,P,P1,P2,P3,P4,P5);
42 T1 = connect(Plants,Gp,Dp,C,F,Sum1,Sum2,Sum3,{'r','n'},'y'); % Smith
43 figure
44 step(T1,'black');
```

Bibliography

[1] L. van Moorsel, *A planar precision stage using a single image sensor.* Msc. Thesis TU Delft, 2017.

[2] S. Garrido-Jurado, R. MuÃśoz-Salinas, F. Madrid-Cuevas, and M. MarÃŋn-JimÃľnez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280 – 2292, 2014.

[3] M. Cafe, *Nanometer precision Six Degrees of Freedom Planar Motion Stage with Ferrofluid Bearings.* Msc. Thesis TU Delft, 2014.

[4] G. Mok, *The design of a planar precision stage using cost effective optical mouse sensors.* Msc. Thesis TU Delft, 2015.

[5] H. Habib, *Design of a three Degrees of Freedom planar precision stage using a single Position Sensitive Detector.* Msc. Thesis TU Delft, 2015.

[6] F. J. Romero-Ramirez, R. MuÃśoz-Salinas, and R. Medina-Carnicer, "Speeded up detection of squared fiducial markers," *Image and Vision Computing*, vol. 76, pp. 38 – 47, 2018.

[7] R. M. Schmidt, *The Design of High Performance Mechatronics -2nd Revised Edition.* IOS Press, 2014.

[8] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design.* IOSJohn Wily and Sons Ltd, 2005.

[9] S. Lampaert, *Planar Ferrofluid Bearings Modelling and Design Principles.* Msc. Thesis TU Delft, 2015.

[10] J. Wang and E. Olson, "Apriltag 2: Efficient and robust fiducial detection," *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4193–4198, 2016.

[11] D. F. Abawi, J. Bienwald, and R. Dörner, "Accuracy in optical tracking with fiducial markers: an accuracy function for artoolkit," *Third IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 260–261, 2004.

[12] M. Fiala, "Artag, a fiducial marker system using digital techniques," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, pp. 590–596 vol. 2, 2005.

[13] D. Heck, *Programming with Threads: Synchronization and Communication.* 2011.

[14] R. Sipahi, *Time Delay Systems: Methods, Applications and New Trends.* Springer, 2012.

[15] J. J. Loiseau, *Topics in Time Delay Systems.* Springer, 2009.

[16] Q.-C. Zhong, *Robust Control of Time-delay Systems.* Springer, 2006.

[17] T. Insperger, *Time Delay Systems.* Springer, 2017.

[18] K. Natori, T. Tsuji, K. Ohnishi, A. Hace, and K. Jezernik, "Time-delay compensation by communication disturbance observer for bilateral teleoperation under time-varying delay," *IEEE Transactions on Industrial Electronics*, vol. 57, pp. 1050–1062, 2010.

[19] J. Smith, "Closer control of loops with dead time," *Chemical Engineering Progress*, vol. 53, pp. 217–219, 1957.

[20] J. E. Normey-Rico, P. Garcia, and A. Gonzalez, "Robust stability analysis of filtered smith predictor for time-varying delay processes," *Journal of Process Control*, vol. 22, no. 10, pp. 1975 – 1984, 2012.