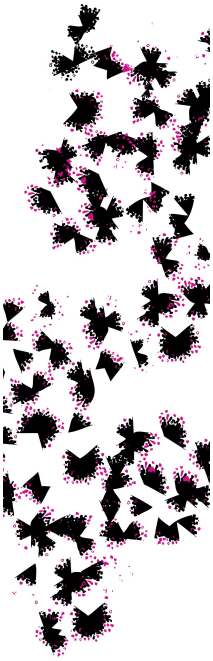

Designing an Actuated Metamorphic Mechanism based on Polyhedral Structures

R.J. (Rik) Koppelman

MSc Report

Committee:
Prof.dr.ir. G.J.M. Krijnen
Dr.ir. J.B.C. Engelen
H. Noshahri, MSc
Dr.ir. R.G.K.M. Aarts

October 2018

041RAM2018
Robotics and Mechatronics
EE-Math-CS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Summary

My theses assignment for the inspection group at the University of Twente is to design an actuated metamorphic polyhedral structure. A metamorphic polyhedral structure can be used as a robot that can change it shape and size. A problem with inspection of tunnels/pipes is that the current robots cannot handle obstacles or diameter changes. The aim of this research is to test if a metamorphic polyhedral structure can be used in tunnels/pipes with varying diameters and that it can achieve a controlled expansion. This project was not only intended for tunnel inspection but could also be used in a broader spectrum. When scaled up or down it could be used for stents in blood vessels or in construction. The literature research and design choices for this project are performed for a 3D metamorphic polyhedral structure, however the design and testing phases handles the 2D case. This choice is made to check if the problem in infrastructure can be solved with this research.

The first step of the research is to investigate the various metaphoric polyhedral structures. With the analysis of the structures, the Hoberman sphere is chosen to be best suited for the problem description. The Hoberman sphere is chosen because it has the highest expansion ratio and can sustain stress better. However it has less space for sensors than other structures. The Hoberman sphere is build up of 2-3 rings connected onto a tetragon with multiple scissor joints. During this research the expansion ratio is checked with respect to the theory. With the Hoberman sphere chosen, a motor set-up is selected where a slider crank mechanism is used. This set-up is selected because it stays inside the sphere while it is expanding and has the highest range of motion.

To control the Hoberman sphere in an enclosed space where it touches the surface of that space, an interaction controller is chosen. This controller can handle forces from the environment, for example an obstacle pushing on the system. To have a stable system with an interaction controller the velocity of the system is necessary. The velocity of a system can be obtained with multiple various methods. One of the methods is using an observer. For an observer to work a model of the system is necessary. A mathematical model of the Hoberman sphere is calculated by using the Lagrangian method. With the Lagrangian method, an equation of motion of the Hoberman sphere was found.

From the mathematical model a physical prototype is build using 3D printing and laser cutting. With a prototype made and the model known, various tests are performed to check if the model has the same behaviour as the prototype. The first test is to check the relation between the angle of the Hoberman sphere and the angle of the motor. The second test is an open loop simulation of the model. In this test an impulse response is set onto the plant to get the crossover frequency of the system. The crossover frequency was found to be 82rad/s. To verify the model a new controller is designed using the crossover frequency in the controller design. This controller is a PID controller. When the model is correct and the velocity of the system is correctly estimated, the PID controller can be replaced with an interaction controller. The final test that is performed on the prototype is a response test with a DC-motor. A reference signal is set onto the system and the angle & current are measured during that process. The results of those measurements are then compared to the simulation.

The main conclusion of this research project is that the model of the Hoberman sphere needs to be improved to better reflect the practice. If the model is further improved the interaction controller can be implemented. The design and selecting of a metamorphic polyhedral structure part of the research project is done successfully. The expansion ratio of the Hoberman sphere is however different from the theory. The theory described an expansion ratio between 2-3, where in practice the ratio is about 1.23. The difference here is mainly due to the added motor set-up. In the closed loop test the error angle signals of the response are compared and the prototype has a RMS value 3 times higher than the simulation. This value needs to be lower when the model is implemented in the interaction controller.

Acknowledgement

Hereby I would like to thank the people who have helped me during my studies and my thesis project. First of all I would like to thank the technicians at the University who helped me with the printing & laser cutting my designs and gave some extra tips. Furthermore I would like to thank dr.ir. Johan Engelen who was my supervisor. He gave me some tips about how to do a master thesis, some information and put me in contact with the right people.

Finally I would like to thank Hengameh Noshahri MSc. who was my main supervisor during my project and who gave me ideas/comments, but still let me take my own path.

Contents

List of Figures

List of Tables

Abbreviation list

RAM	Robotics and Mechatronics
US	Ultra Sound
CCTV	Closed Circuit Television Inspection
pH	potential of Hydrogen
DOF	Degree of Freedom
CoG	Centre of Gravity
EoM	Equation of Motion
CoM	Centre of Mass
PID	Proportional Integral Differential
DC	Direct Current
C	Channel
SS	State Space
RMS	Root Mean Square
EMF	Electromotive Force

List of symbols

Symbol	Description	Unit
V	Potential energy	J
C	Compliance	$\frac{m}{N}$
x	axis, state variable	variable
F	Force	N
R	Rayleigh function	$\frac{Nm}{s}$
b	damping coefficient	$\frac{Ns}{m}$
m_c	Control mass	kg
F_c	Control force	N
k_c	Control stiffness	$\frac{N}{m}$
n	number of scissor joints	-
β	Tameness factor	-
τ	Torque	Nm
L	Lagrangian	J
T	Kinetic energy	J
q	Generalised coordinate	variable
y	Output variable, axis	variable
u	Input variable	variable
P	Power	W
A	State-matrix in state space	-
B	Input matrix in state space	-
I	Current	A
U	Voltage	V
C	Controller	A
A	Actuator	
k_m	Motor constant	$\frac{Nm}{A}$
ω_c	Cross-over frequency	$\frac{rad}{s}$
F_n	Normal force of motor	N
F_h	Force of Hoberman sphere	N
F_r	Force of clevis	N
K_c	Gain of controller	-
K_p	Gain of proportional controller	-
K_i	Gain of integrator controller	-
K_d	Gain of differentiator controller	-
τ_z	Control constant of the zero	-
τ_p	Control constant of the pole	-
τ_i	Control constant of the integrator	-
$\frac{d^2x_{max}}{dt^2}$	Maximum acceleration	$\frac{m}{s^2}$
$\frac{d^3x_{max}}{dt^3}$	Maximum jerk	$\frac{m}{s^3}$
h_m	Range of motion in reference	rad
t_m	Time for range of motion	s

List of lengths and angles

Symbol	Description	Unit
x	position	m
y	position	m
x_v	virtual position	m
x_d	Difference between virtual and actual position	m
l_0	Begin length	m
r_0	Begin radius	m
α	Large angle of scissor part	rad
β	Small angle of scissor part	rad
l_1	Long length of scissor part	m
l_2	Short length of scissor part	m
l_l	Length of prototype scissor part	m
l_h	Height of prototype scissor part	m
θ	Controlled angle of Hoberman sphere	rad
l_3	Length of inner ring Hoberman sphere	m
l_4	Length of middle ring Hoberman sphere	m
l_5	Length of outer ring Hoberman sphere	m
l_6	Length between point 2 and 12 of l_3	m
θ_m	Angle of the motor	rad
θ_s	Angle of the structure	rad
r_1	Radius of ellipse motor set-up	m
r_2	Length from clevis to clevis motor set-up	m
l_7	Length to calculate efficiency motor set-up	m
l_8	Length to calculate efficiency motor set-up	m
ψ	Angle to calculate efficiency motor set-up	rad
ρ	Angle to calculate efficiency motor set-up	rad
λ	Angle to calculate efficiency motor set-up	rad
ϕ	Angle to calculate efficiency motor set-up	rad
$\pm e_{pos}$	max or min error of the angle	rad
$Theta_x$	Angle from reference profile	rad
$Theta_v$	Angular velocity from reference profile	rad
$Theta_a$	Angular acceleration from reference profile	rad
$Theta_e$	Error angle $(Theta_x - \theta)$	rad

1. Introduction

This chapter describes an introduction about the thesis project. When reading about dimensioning of units in this report it is always referred to as, length times width times height ($l \cdot w \cdot h$) in meters unless explicitly written different. Furthermore when reading about expanding/expansion, it also means retracting/retraction unless specified specific.

1.1 Context

A current problem in the infrastructure is the inspection and maintenance of structures that are used for transportation. These structures consists of pipes, tunnels and sewers. The length and the number of tunnels keep increasing. In Japan the length and number increased by about 50% in the last twenty years [1]. To inspect and do maintenance on such structures, robots are developed to improve safety, reliability and to reach difficult accessible places. The robots which are currently used for pipe inspection, cannot continue their movement if there is an obstacle or change in diameter of the pipes.

1.2 Project description

This problem can be solved with the study of shape-changing (metamorphic) robots. Metamorphic robotics is an exciting and relatively new line of research. Simple concepts of Japanese traditional paper-folding art, Origami, have inspired the design of various foldable robots. In addition, polyhedral structures are considered as suitable candidates for foldable robots. Their topological configuration can change by means of changing the connectivity angle of their parts. This research is however not limited to the field of tunnel inspection, but could also be used in other fields.

The goal of this thesis assignment is to design an actuated version of a polyhedral structure in order to achieve a controlled expansion of the mechanism. To achieve this goal the assignment will be split into various research questions:

- Which of the polyhedral structures are most fit, with respect to the requirements, for the problem-set?
- What is the equation of motion of that structure?
- How can the polyhedral structure be actuated, and what actuators are suitable for such movements?
- What control strategy is suited to handle this problem?
- How does the prototype compare to the model of the structure?

1.3 Current and related work

1.3.1 Current way of working

The current tunnel inspection technology can be divided into various parts, namely pipes, sewers and tunnels. These various parts have many similarities however the biggest difference is the size and usages of those parts. Tunnels and sewers are often build out of reinforced concrete and typical defects here are cracks, spalling (surface failure) and efflorescence/leakage. These defects can be detected by a number of methods and a couple of them are given here [1]. First the method is given and then the application is described:

- Visual; a camera/engineer checks the surface and looks for a defect. With a defect found further testing can be done.
- Strength based; Schmidt hammer checks the strength, uniformity and quality of the structure.
- Ultrasound (US); impact hammer checks the strength of the wall by measuring the travel time from impact hammer to detector.

- Magnetic or Electric; sensor that can detect the positioning of the reinforcement. Lack of reinforcement could mean corrosion.
- Thermography; a sensor that measures the thermal radiation emitted from the surface. Non uniform thermal radiation pattern could indicate an abnormality in the underlying structure.
- Radar; ground-Penetrating Radar uses an electromagnetic wave and checks the difference in dielectric constants.
- Endoscopy; an endoscope checks an element for defects deeper in the structure.

All these methods can be used to check defects in tunnels, sewers or pipes. With these methods the inspector still needs to be present and perform the inspections. This process is time consuming, labour dependent (biased) and could be hazardous.

There are robotic systems that can inspect tunnels. These robotic systems are often placed on top of a truck and are using a robotic arm or crane with a robotic arm on top of it. On the robotic arms are sensors and a repair tool, an example can be seen in Figure 1.1. In the example the robotic system uses an ultrasound measuring method to check the strength of the surface.

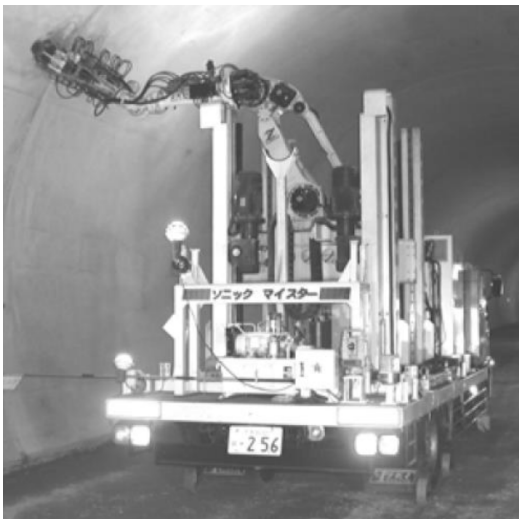

Figure 1.1: Example of a robotic inspection system, [1]

The sewer inspection and replacement is mainly based on structural and hydraulic performance. Information about the performance is received by using a visual method called CCTV, road work and pipe age. The inspection is done by moving a camera which is mounted on top of a vehicle that drives through the sewers. The normal procedure in CCTV inspection is divided into four parts namely, analysing data, coding, condition assessment and prioritizing rehabilitation of sewers. The first two parts are done manually where the engineer checks the data and gives a code to the data. That code gives information about where the defect is and what kind of defect it is. This input is then used by a computer and gives a score of the sewer condition and assesses the condition of the sewer. The last step is to check all the sewers conditions and prioritize which sewer needs to be fixed/replaced. This test is done on average every 8 years [2].

The disadvantages are again that this process is labour intensive and time consuming. Another problem with this method is that the camera system can only check the surface of the sewer and not the deeper structure. An engineer check/evaluate data three times in this method, which is prawn to objectivity [2].

Most of the pipes in a gas distribution network are low pressure pipes. These pipes are currently inspected by leak surveys above ground. This is, as all the examples above, a labour intensive and time consuming job. The major problem with this method however is that it can only find leaks, meaning that there is already a defect. Another problem is that this method does not give any information about the layout [3].

The inspection of high pressure gas pipes is already done with robotic systems. However these systems are not fully autonomous. The main reason that there are already robotic systems is that these pipers are bigger in diameter and have less obstacles or junctions.

1.3.2 Related work

Metamorphic actuated polyhedral structures are a relative new line of research and therefore there is not specific related work that can be explained here. However, metamorphic polyhedral structures are also used in other fields. In space this technology is commonly used to deploy certain structures [4]. There the structure is compressed on earth and launched into orbit. When it is in orbit it is released and folds itself into the full structure, these structures are referred to as inflatable structures [5].

Metamorphic polyhedral structures are also used in construction[6]. This structure uses scissor-assemblies that are connected through hub elements. With this technique structures can easily be assembled and disassembled.

Some polyhedral structures also exist in our nature, namely viruses [7]. The folding mechanism of a virus is until now still not completely figured out. These viruses change their form when there is a change of pH value.

1.4 Other possible field of research

One of the fields of research is biomedical. In hospitals when people have a blocked passageway a stent is placed to unblock the passageway. The stent restores the flow of blood or other fluids. A stent is a mesh tube with a certain structure. When a stent is placed a minimal invasive surgery is done. In this surgery, a tube is placed inside the damaged/blocked passageway with a wrap on the outside of the stent. When the stent is in the correct position the wrap is taken of/retracted and the tube will expand to the wall of the passageway. When the part of the passageway that was completely/partly blocked has been dissolved, the stent will retake the form of the outer wall [8]. The technology that is described in this project could be an alternative for the stent.

1.5 Methodology

In this thesis report a design of an actuated metamorphic polyhedral structure will be investigated. To design the desired structure, first the background information and theory about the various polyhedral structures are described. In the next chapter the design choices will be discussed based on the background information about the polyhedral structures. Here a structure will be chosen and a motor set-up will be designed. After the selection of the structure the EoM is calculated to find a model for the structure. This is done in chapter four. In this chapter the polyhedral structure is schematically explained and further analysed. In this chapter also a design is made of that polyhedral structure and the two different models are compared to each other. From the designed structure a prototype is made and a motor and sensor are selected for that prototype. After the physical prototype is build the controller design is done which is used to verify the found model in chapter four. In the next chapter the measurements and the results are explained. Here the model that is made in chapter four is checked for correctness and compared with measurements done with the prototype. In the last chapter, the conclusions of the research project and the recommendation about further future actions are given.

This research project only handles the 2D case. The background information in chapter two and the choices that are made in chapter three are however done for a 3D case. This choice is made to check if the problem in infrastructure can be solved with this research. When the problem can be solved with this research it can be extended for a 3D case.

2. Background

2.1 Polyhedral structures

2.1.1 General Polyhedra

A polyhedron has a face, edge and a vortex. These polyhedra can be build up of many polygons. A N-gon is a polygon with N sides, where a 8-gon is an octagon. Polyhedral structures can be divided into different categories, where every category has its own constraints. Here are some relevant categories listed with their constraints.

- Regular polyhedra, also known as Platonic solids
  - All faces are identical.
  - Number of faces and edges at each vortex are the same.
  - All angles are the same at all sides.
  - There are five polyhedra that have these properties, namely tetrahedron, cube, octahedron, dodecahedron, icosahedron.
- Semi-regular polyhedra, also known as a Archimedean Solid ([5] & [9])
  - It is vertex-transitive, meaning the number of faces and edges at each vortex are the same.
  - There are a total of thirteen of these polyhedra.

2.1.2 General metamorphic polyhedra

The general metamorphic polyhedra also have categories with constraints. These metamorphic polyhedral structures are categorised beneath and are explained in the rest of this subsection.

- Expandohedra [7]
  - The structure consist of rigid bodies with the same shape and rigid connecting elements.
  - All connections between prism and plate are revolute joints.
  - The whole assembly must have icosahedral rotational symmetry, meaning all chains of connecting elements between adjacent pentagonal prisms are identical and must have a two-fold rotational symmetry.
  - All conditions of compatibility should be satisfied in any position during expansion.
- Jitterbug system or Dipolygonid [9]
  - Each transformation starts from a regular or semi-regular polyhedron but keeps certain rotational symmetry while transforming.
  - The structures after the transformation are not symmetrical with respect to reflections.
  - The structures contain one or two types of polygons with the same edge length.
  - The motion of the vertices is along the intersecting curve of two circumscribed cylinders.
- Orthotropic multibody system [10]
  - The system is unrooted.
  - The system motion is holonomoidal.
  - It is orthothropic.

A method of making a polyhedra metamorphic is by placing a structure between the edges of the polyhedra. This is show in Figure 2.1. The polyhedron is shown in part *a*. The faces of the polyhedron are separated and the edges are copied in part *b*. The edges, a & b are now replaced with a′ & b′ and a″ & b″. In part *c*, the hinges are added. This hinge is attached to every vertex as is shown in part *d*. From there on the whole structure can be build up. A good example of this method is shown in Figure 2.6. With this method an expandohedra is created.

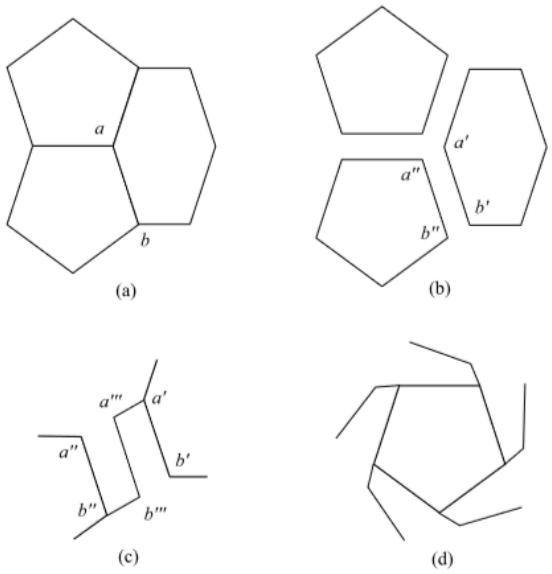

Figure 2.1: Making a polyhedra metamorphic [7]

Another method of making polyhedra metamorphic, is to replace some of the edges with prismatic joints. With the correct placement of the prismatic joints, the polyhedra becomes single DOF. The combination of multiple polyhedra can be done by connecting the edges, faces or vertex and make that into a rigid connection. Different structures can be build that are also single DOF by combining multiple polyhedra. The constraint however is that one of the prismatic joints should be connected/constrained. A single DOF chair could be build with this method [5], see Figure 2.2. The chair example has six prismatic joints within each cube. With those prismatic joints and the constraints the chair example can change form.

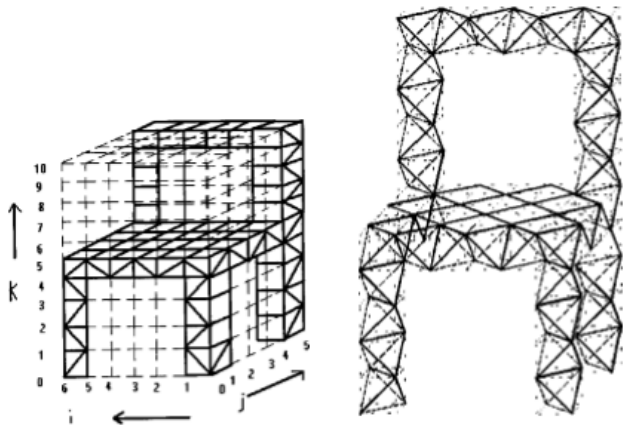

Figure 2.2: Expanding chair [5]

The Jitterbug transform is introduced by Richard Buckminster Fuller. The Jitterbug transform is a polyhedral structure that can transform from one polyhedral to another and is built up using triangles. Each triangle translates with a rotation around its symmetry axis [15]. The Jitterbug transform is not a specific structure but a group of structures that can change their shape and size. In Figure 2.3 an example of a Jitterbug transform is given.

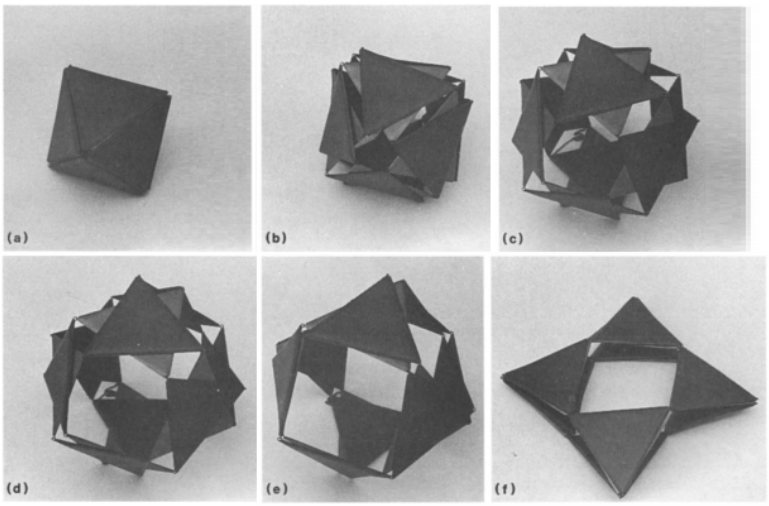

Figure 2.3: A Jitterbug transform of an octahedron [15]

The difference between the expandohedra and the Jitterbug system is mainly that the expandohedra expand with a hinge connected to vertex and the edge. The Jitterbug transform has two structures that can rotate with respect to each other and are all connected to the vertex.

The last category is the orthotropic multibody system. To better explain the orhothropic multibody system and its constraints, see Figure 2.4. An unrooted system has no kinematic constraints to a relative origin (Galilean-frame). It is a free object. A non-holonomoidal motion is a motion that has its final orientation depending on the entire motion history of each subsystem. For a single DOF system this is not the case, because every subsystem has the same motion. This means that the history of every subsystem will give the correct orientation. Another property is that the system is an orthotropoid, which means that König's frame and the Gylden frame coincides. The König's frame is the frame at the CoG with the same axis as the Galilean frame. The Gylden frame is where the internal kinetic energy is least possible with the axis orientation relative to the base body. The advantage of orthotropic multibody systems is that the EoM becomes easier due to the properties. One of the properties is that the internal EoM is independent with respect to the external EoM. Therefore the EoM of a system in this category can be determined independent of the environment. The method described in [10] can be used to describe the EoM of the metamorphic polyhedral structure.

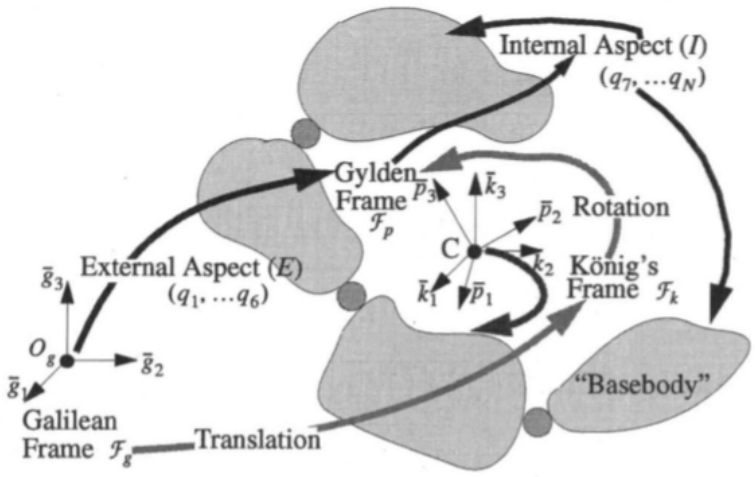

Figure 2.4: The frames of a multibody system [10]

From the literature review that is done for this thesis, there are two different ways of expanding structures, translational and rotational.

2.1.3 Translational expanding structures

Hoberman sphere

The Hoberman sphere is an invention by Chuck Hoberman. He was an engineer, architect and inventor. He invented the Hoberman sphere and other folding structures/toys [11].

The Hoberman sphere is a translational expanding structure that uses a scissor-like mechanism with revolute joints in the structure to expand and retract. Those revolute joints change in one rotational DOF. Due to those joints the Hoberman sphere can change the exterior shape while expanding. Hoberman structures use revolute joints and are based on a tetragon (cube). The tetragon is connected to multiple great circles, which can then expand as a whole [5]. In Figure 2.5 the yellow bars are part of the tetragon and the blue, green and purple are part of the three great circles. The red items in the figure are some of the joints/connectors. The more of the scissor joints are assembled in the great circles the larger the change in diameter becomes. This means that the Hoberman sphere expansion ratio is dependent on the number of scissor joints. There are also different versions of the Hoberman sphere toy, namely the mini and original. The mini Hoberman sphere has four scissor joints per quadrant and the original Hoberman sphere has six scissor joints.

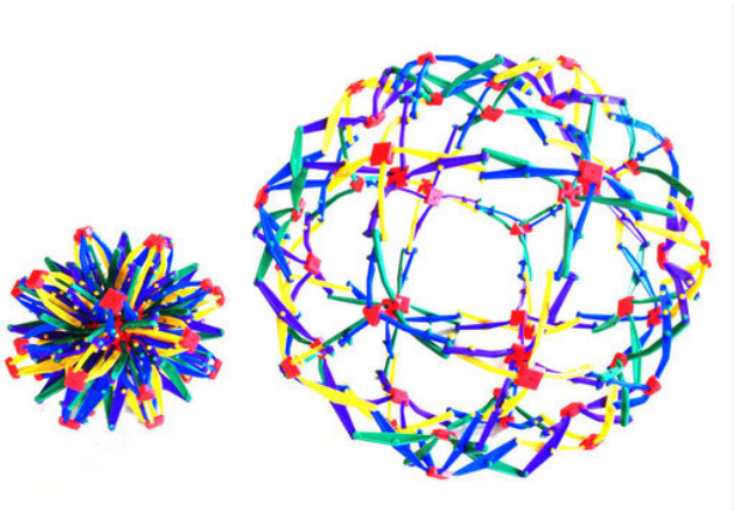

Figure 2.5: Hoberman Sphere [12]

The Hoberman sphere belongs to a special class of multibody systems namely the orthotropic multibody systems, and because of that it has a simplified EoM. The Hoberman sphere has however two downsides, they change faces when expanding and have a large number of moving parts.

2.1.4 Rotational expanding structures

Dodecahedron

As described in subsection 1.3.2, the dodecahedron is used to simulate the behaviour of a virus expanding. A dodecahedron preserves faces but changes its translation and rotation along its symmetry axis. This effect can be more clearly seen in Figure 2.6 where in the first part the dodecahedron is suppressed and in second part it is expanded. In both parts the faces have the same size. The dodecahedron expands by rotation, a certain translation occurs with the rotation of each element. The expanding ratio of the dodecahedron is dependent on the edge length of the structure.

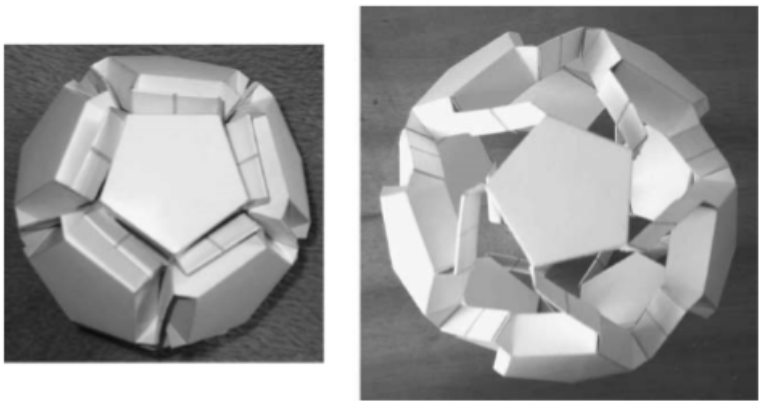

Figure 2.6: Dodecahedron [7]

With the constraints, all the pentagons (each separate element) move with the same velocity and have the same distance to the origin. The dodecahedron increases its circumradius about 77% and a volume increase about 7.2 times. This increase however depends on the edge length and the length from the origin to the vertex of the polyhedra. The dodecahedron is a clear example of a polyhedra using the second technique, where an extra edge is used to expand. The dodecahedron is part of the expandohedra class [7].

Octoid

The octoid is a design made by Wolhart, and can be seen in Figure 2.7. This is a structure that also does not change its face when expanding. The expansion ratio of the octoid is also dependent on the edge length of the structure.

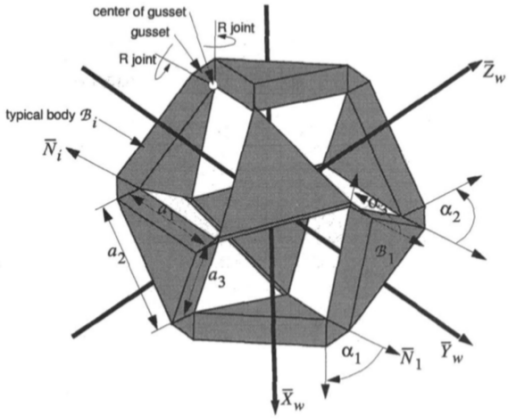

Figure 2.7: Octoid [13]

The octoid consists of eight identical triangles which are connected to twelve gussets to make it a single DOF. Properties of the octoid are that during motion the normal of the triangles does not change. Furthermore the intersection of the two rotational joints stays in one of the x, y or z planes during the motion. The octoid is also part of the orthotropic multibody system [10] as well as the jitterbug system.

Fulleroid

The fulleroid is also a design made by Wolhart. This design is very similar to the octoid and can be seen in Figure 2.8. The fulleroid is build up of triangles that are connected to each other. The expanding ratio of the fulleroid is dependent on the edge length of the structure.

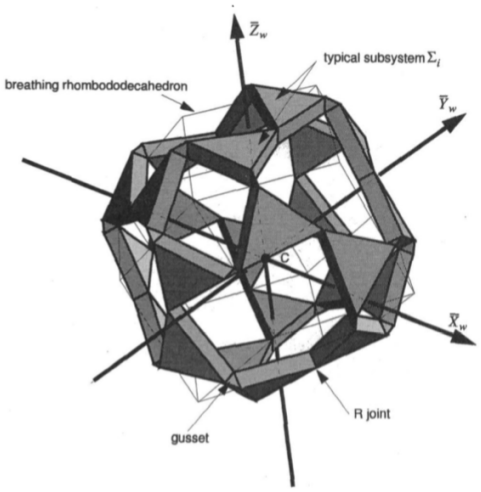

Figure 2.8: Fulleroid [13]

The fulleroid consists of twelve subsystems located on a rhombododecahedron and also has a single DOF. Each subsystem consists of two triangles with a revolute joint. This subsystem is connected with the free vertices to the surface of the rhombododecahedron [14]. The fulleroid is also part of the orthotropic multibody system [10] as well as the jitterbug system.

2.2 Interaction control

A hybrid of position and force control is called interaction control. This means that the system depends on the robot and the environment. With interaction control a robot exchanges energy with the environment.

A way to use interaction control is to control the potential energy of a system, and it is called energy shaping. The control law for a P-action interaction controller is shown in Equation 2.1. In the equation V is the potential energy, C is the compliance, x_d is the difference in position where $x_d = x - x_v$ and the x_v is the virtual position. The difference between this virtual position and a setpoint, is that a setpoint should always be reached. However a virtual position could be reached if there is no external force limiting the system. When the virtual position cannot be reached, the system will exert a force on the environment.

$$V(x_d) = \frac{1}{2C}x_d^2 \qquad F = -\frac{\partial V}{\partial x_d} = -\frac{\partial(\frac{1}{2C}x_d^2)}{\partial x_d} = -\frac{1}{C}x_d = -\frac{1}{C}(x - x_v) \tag{2.1}$$

There are certain conditions that should be met when using energy shaping. These conditions are that the system should be:

- Back drivable system
- Low friction
- Position measurements

In order for energy shaping to work, as shown above, the system should be back drivable. When the environment pushes on the system, the system should respond. A metamorphic polyhedral structure is back drivable by design.

The system will have an oscillatory behaviour with only a spring. This is because the spring will pull the system to the desired position. The system has low friction and so there is almost no action causing the system to stop at the desired position. The system will overshoot and the spring will act in the opposing direction. This continues until the system is at rest. Adding a damper to the system will make the system get at rest faster, and thus solve the problem of the system behaving oscillatory. With that the control law becomes Equation 2.2, where R is the Rayleigh function. In the linear case the solution is given after the equation, where $\frac{dx_d}{dt} = \frac{dx}{dt}$. Now a type of PD controller is designed.

$$R(x_d) = \frac{1}{2}b(\frac{dx_d}{dt})^2$$

$$F = -\frac{\partial V}{\partial x_d} - \frac{\partial R}{\partial \frac{dx_d}{dt}} = -\frac{1}{C}(x - x_v) - \frac{\partial(\frac{1}{2}b(\frac{dx_d}{dt})^2)}{\partial \frac{dx_d}{dt}} = -\frac{1}{C}(x - x_v) - b\frac{dx_d}{dt} = -\frac{1}{C}(x - x_v) - b\frac{dx}{dt} \quad (2.2)$$

Now the problem arises that the $\frac{dx}{dt}$ is normally not measured. The quantity that is measured is the position and when differentiated the velocity is obtained. However this differentiation is done numerically and will result in an increase of noise. Some of the options to solve this problem are:

- Observers
- State Variable Filters
- Damping Injection

With an observer the states of the system can still be found even if these are not available on the output. For this the system needs to be observable. For an observer a model of the system is necessary. With the observer the dynamics of the error can be chosen. A state variable filters uses an integral action, which is less sensitive to noise. With this option however a phase lag is introduced depending on the frequency. Therefore the state variable filter is good at low frequencies. Another method for finding the velocity is by using damping injection and it is schematic shown in Figure 2.9. In the figure there are two models shown (top & bottom). The top model is the desired PD-interaction controller. It has however still the problem that the velocity is not known. The lower model, with the extra mass and stiffness, the velocity issue can be solved. The boxed filled part is part of a simulation and is controlled by the designer. In this set-up the velocity is measured at m_c. This mass is part of the controller and thus the velocity of that mass can be found. The extra stiffness(k_v) is much higher than the original stiffness(k), the mass (m_c) is also much smaller than the original mass(m). With that addition the bottom model mimics the top model, but the velocity can be measured from the simulation [17].

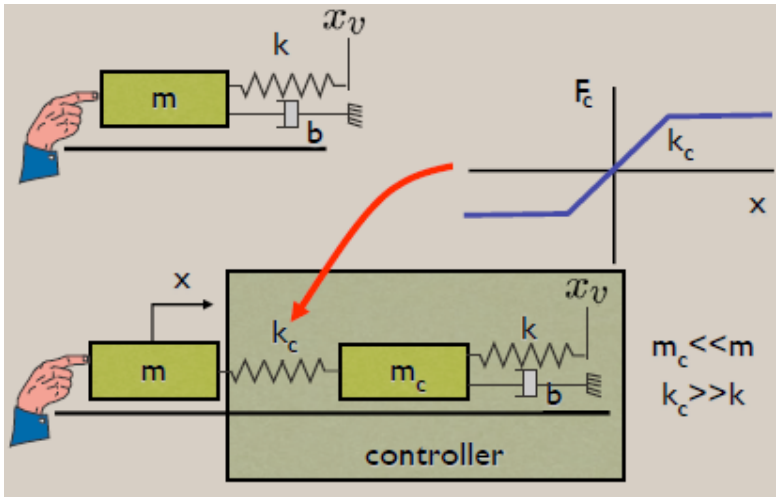

Figure 2.9: Damping injection ([16])

2.3 Conclusion background information

From this chapter it can be concluded that a metamorphic polyhedral structure can be divided into two categories, translational and rotational. For the translational category it is a Hoberman sphere and for the rotational the octoid, fulleroid and dodecahedron. These four structures will be analysed to make a choice which one is suited for the problem set.

Another conclusion can be made about the interaction controller. For a system to have an interaction controller and to be stable, the velocity of that system should be known. This can be solved by using an observer, state variable filter or with damping injection.

3. Design choices

In this chapter the design choices are explained, where a choice is made about the structure, the motor set-up and the controller.

3.1 Choice of polyhedron structure

3.1.1 Characteristics of polyhedral structures

To make a choice about which polyhedron structure is most suited for this research, first the characteristics that are important are described. The characteristics here are not precisely quantifiable because the dimensions are not yet set and some are not known or found in literature. The aim of the characteristics is to determine which of the structures are most suited for the problem description. This is done by comparing the important characteristics with each other. The important characteristics are:

- Expansion ratio
- Stress it can sustain
- Change of face while expanding/retracting
- Easy motor placement

The expansion ratio is defined as the largest ratio divided by the smallest ratio of the mechanism. The characteristics that the faces of a structure don't change while expanding is important for the sensors. The sensors are most likely being placed on the faces of the structure. If the structure does not change its face the entire face could be used for a sensor. However when the structure does change its face, the edge or vertex are only available for the sensor. The area of the face is larger than the vertices or edges and thus the structure that does change its face will have less place for the sensors, which result in less information.

3.1.2 Selection polyhedron structure

To choose the most suitable metamorphic polyhedral structure, the structures themselves need to be compared to each other with respect to the requirements. This is shown in Table 3.1, where the expansion ratio and the complexity are given. The Hoberman sphere expansion ratio depends on the number of scissor joints. As stated in the previous chapter, the mini has 4 and the original has 6 scissor joints. The number of scissor joint from the museum Hoberman sphere is unknown. The complexity gives information about how easy it is to use a motor for that structure. It has to do with the number of moving parts, space within the structure to place a single motor and the complexity of the movement. Also some of the requirements set in the previous section where not available in the literature.

Table 3.1: Comparison between the structures

Polyhedral structure	Expansion ratio	Complexity
Hoberman sphere mini	2.14	Medium
Hoberman sphere original	3.17	Medium
Hoberman sphere museum	4.0	Medium
Dodecahedron	1.77	Low
Octoid	≤ 1.77	Medium
Fulleroid	$\approx$ 1.77	High

The expansion ratio differs for the size of the Hoberman sphere. This is because of the number of scissor joints inside the sphere. With a bigger sphere it is easier to get more scissor joints inside. There was not any information about that specification for the octoid and fulleroid. However for the fulleroid there was a volume increase about the same as the dodecahedron, which would suggest an expansion length of about the same. With the fulleroid being a more complex version of the octoid it would also suggest that the octoids expansion

ratio would be less than the fulleroid

For the Dodecahedron the complexity is low because a single rotational motor could be applied on one of the faces and the whole structure will move. For this reason the octoid and the Hoberman sphere are set to medium. For the Hoberman sphere it is because of the many moving parts but easy move-ability. For the octoid it is because of the more difficult move-ability. This is also the reason why the fulleroid has a high complexity. It has many components that move and has a difficult move-ability.

The rotational structures have the benefit of not changing their face, and therefore not limiting the sensors that could be applied there. The downside however of the rotational structures is that the change in length is dependent on the edge length and thus limiting the expansion.

The amount of stress it can sustain is a requirement, however in the literature review there was no information given about this specification, because it is a property that is dependent on many other parameters. A translational structure while expanding will bump into the surface. A rotational structure will however still rotate while bumping into the surface, creating extra friction on the face itself. This is a negative effect when wanting to use the technology in the field. When the structure is expanding it could damage the face where the sensors are connected.

3.1.3 Conclusion polyhedral structure

With the specification better explained for every structure, the Hoberman sphere is the structure that is selected for this research. The Hoberman sphere has the highest expansion ratio, is better suited for stresses and has easy motor placement options. The only downside of the Hoberman sphere is that it changes it face during expanding. The sensors can however be added to the vertex or edge of the Hoberman sphere, these points are always directed toward the surface.

3.2 Motor set-up selection

The metamorphic polyhedral structures are all single DOF and thus only need one motor to move the entire structure. For the simplified EoM as discussed in the previous chapter it is recommended to place the motor in the CoM. The structure could be actuated by a rotational or a translational motor. In this section various motor set-up will be explained that could be used for the Hoberman sphere with their advantages and disadvantages.

3.2.1 Motor set-up

There are some set-ups possible with the current choice of the polyhedral structure. The first option is that the motor is in the CoM with a connection to one of the joints of the Hoberman sphere. Another option where the motor is not connected in the CoM is to put the motor between the scissor joints. In Figure 3.1 the tetragon is shown of the Hoberman sphere, where the two options are depicted. In the first option the motor could be a translational or a rotational motor. A translational option is to have a rod, that is connected to a scissor joint, drive the Hoberman sphere. When the motor pull/push the rod the whole structure will move. As stated, because the structure is single DOF this should be enough. For the second option a translational motor is then fixed to one end of the scissor joint and it controls the other end. The problem that is already stated is that it is not in the CoM. Another problem is that there is not much room for a motor.

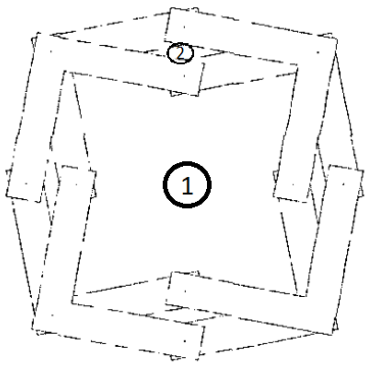

Figure 3.1: Tetragon of a Hoberman sphere [5]

A problem however due to friction, spacing and misalignment, the motor needs to put in more force than necessary. The structure itself also rotates a bit and therefore creates extra friction in the system. A better option is then to use a motor with a certain mechanical set-up that connects two points opposite from one another. In this way the forces are better distributed. Another bonus here is that the CoM is not affected due to symmetry. This motor set-up will have a rotational motor with a mechanical set-up that will move the opposite joints and will be situated in option one.

3.2.2 Requirements motor set-up

For a motor set-up to be selected certain requirements are set. First the requirements are given to decide which motor set-up is suited for the problem. At the end of this section a choice will be made that best fits the requirements. These requirements are only to determine the set-up, so the mechanical part, not to decide a motor.

The requirements for the motor set-up are:

- Be usable within a tunnel or other environments with narrow spacing .
- 3D applicable.
- Fit within the cavity of the Hoberman sphere.
- Mechanism should not exceed the outer Hoberman sphere.
- Being able to use the full range.

Most of the requirements are logical and are related to the boundaries of the Hoberman sphere, however they will be shortly explained here. The total structure is going to be applied in an environment where normally no humans are permitted, and where there is not much extra space, the motor set-up should be applicable for that field. Another property is that the set-up should be fixed/floating inside the Hoberman sphere and stay fixed within the CoM, so 3D applicable. Logically the set-up should fit within the inner sphere and should not exceed the outer sphere. Another requirement is that the set-up should not exceed the outer boundaries when expanding or retracting. If this is not satisfied the set-up is not usable inside a tunnel or other structure. Lastly it is preferable that the set-up could utilize the full range of the Hoberman sphere. This means that a large expansion ratio can be used. In each motor set-up a ratio is mentioned that gives a measure for the expansion of the set-up. The ratio is defined as r/r_0, where r_0 is the minimum radius and r is the radius.

3.2.3 Motor set-up options

Double rack and gear

An option for a motor in the situation where it needs to push and pull in to opposite directions is a rotational motor with double rack with a gear attached to it, this is shown in Figure 3.2. In the figure the double rack goes horizontally so the figure is more clear but in fact the double rack will go within a certain angle where one end is connected to a joint. In the figure the set-up is in its minimum configuration. When one bar is to the left and the other to the right, the set-up is at its maximum. In its maximum the r_0 is twice as high, thus a ratio of 2.

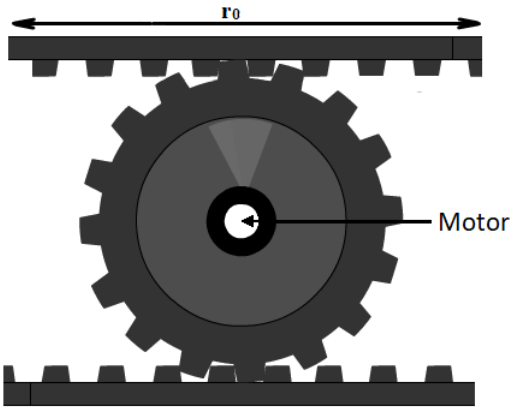

Figure 3.2: Double rack and gear

A problem with this set-up is that the double rack needs guidance to make sure that the double rack stay connected with the toothed gear, making the system more complex. In this set-up the guidance necessary are point guidances, because the racks continuously rotate.

Double spindle and gear

The double spindle is a smaller version of the double rack and gear. In Figure 3.3 the set-up is shown, where on the light blue the connection to the structure is. Furthermore as shown in the figure, the motor will drive a gear and on that gear there are two other gears that are $\pi/2rad$ shifted and connected to the spindle.

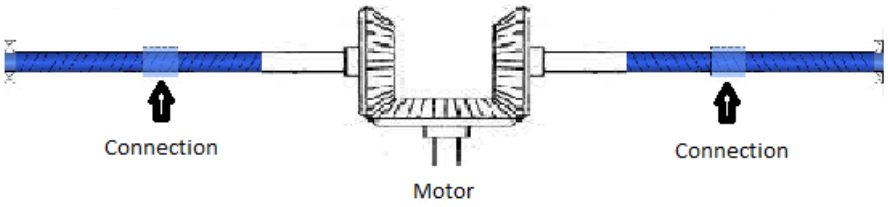

Figure 3.3: Double spindle and gear

The advantages of this design is that the forces are distributed in one line. Another advantage is that set-up could be relatively small and it can drive the full range. However it also has the same disadvantage, namely that the guidance of the spindle will go outside of the boundaries and that there are extra constraints necessary to keep the motor connected to the spindle. The ratio of this set-up is 1, because this set-up cannot retract. This set-up as stated, has a fixed length.

Pressure piston

Another option is a piston idea with a motor attached to a piston rod. This is shown in Figure 3.4. This idea has one large piston where the motor pushes and pulls, and two smaller pistons where the structure is connected.

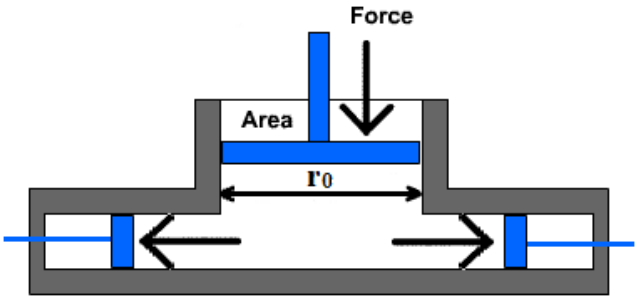

Figure 3.4: Pressure piston

This idea has a lot of advantages, however it has one major downside and that it needs a compressor attached to the Hoberman sphere. Another downside is that this is not good controllable. The advantage of this set-up is that inside the Hoberman sphere it can be very small. However the ratio and range of this design is dependent on the dimensioning of the piston. In this example the ratio is about 2.

Slider crank

Furthermore there is another option with a rotating gear and two bars attached to it, like a train. As is shown in Figure 3.5 the motor is connected to a disc of a certain shape. This disk has two rods connected to the structure which will drive the system.

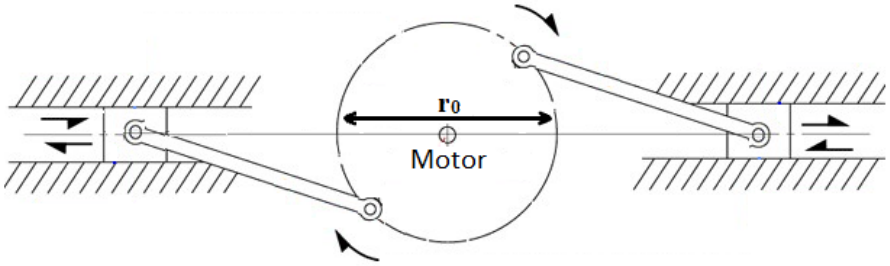

Figure 3.5: Double slider crank

The main advantage of this idea is that it fits almost all of the requirements. The disadvantages here are that the push direction is at an angle and resulting in some loss of force. Furthermore this set-up requires some room within the sphere to let it reach the maximum range. This results in a trade-off between size of set-up and range, so the requirement being able to use the full range is not fulfilled. The ratio here is between 2-3 dependent on how far the disk in the middle can turn without colliding with the rod. The configuration where the bars are vertically opposite from each other, the set-up is at $0rad$ motor angle. If the motor rotates $-\pi/2rad$ the minimum configurations(r_0) is found. When the motor rotates $\pi/2rad$ the maximum configuration is found. So at the motor angle $0rad$ the ratio is 2 and at $\pi/2rad$ the ratio is 3.

There are of course other ideas that are usable within this structure, however these are probably variations of the set-ups that are already described.

3.2.4 Conclusion motor set-up

To make a choice about the motor set-up the requirements that are set are again checked for every set-up. In the description of every set-up the advantages and disadvantages are given. From that it is logical that the slider crank is the best option. The slider crank is the only set-up that stays within the boundaries of the sphere itself, does not need an external power source and has the largest ratio.

3.2.5 Discussion motor set-up

However the difficulty with every set-up is to make it 3D applicable and making it floating yet fixed inside the sphere itself. This can be solved for the slider crank mechanism. When making the set-up 3D applicable the set-up automatically is fixed within the sphere itself. When two planes are made drivable the third axis is already fixed within the sphere. The 3D set-up will not be described for now because the report only handles the 2D case.

A problem occurs however in the 2D case with the slider crank mechanism and the other mechanisms. The motor is not fixed with respect to the frame and therefore it can rotate without moving the mechanism. When the motor is fixed the structure itself will also rotate while expanding, due to the force not being perpendicular to the structure. This will be further explained in subsection 7.2.1.

3.3 Controller

The environment that the Hoberman sphere is probably going to be used in is a confined space, like a tunnel. In that environment most of the sensors need to have contact with the surface. Therefore the Hoberman sphere needs to be controlled inside that tunnel to expand to the diameter of the tunnel itself. Some of the measurement techniques needs to apply a certain force to the surface. With such environment the controller needs to be able to:

- Expand and retract the structure.
- Deal with forces from outside (surface) pushing on the system.
- Keep a certain pressure on the surface to prevent it from falling/retracting
- Keep contact while the sensor set-up is doing some measurements/experiments.

The problem with position controllers is that they do not work correctly when the set-point cannot be reached due to a wall or surface being in the way. With that knowledge an interaction controller is chosen as is described in section 2.2. This controller is able to handle all of the requirements. The interaction controller will get an input from a sensor that measures the distance between the Hoberman sphere and the surface. With that sensor data the controller creates a virtual position that lies further away than the input of the sensor. This creates a force that applies pressure onto the surface. The virtual spring will be attached between the outside of the Hoberman sphere and a point behind the surface of the environment. The Hoberman sphere is categorized as an orthotropic multibody system, which results in that the external EoM can be solved separately from the internal EoM. The controller uses the external EoM.

As described an interaction controller has three conditions that need to be met. The conditions are back drivability, low friction and position measurement. All of these conditions could be met with the current Hoberman sphere and slider crank mechanism. However to use an interaction controller the velocity of the system needs to be known. This can be solved with an observer, state variable filters or damping injection. For now the choice is to use an observer, and for that a model is created. With an observer the error dynamics can be freely chosen if the system is observable.

With the choice for an observer,a model is necessary. The rest of the report is about finding and validating the model. Therefore the choice that is made in this chapter about the interaction controller will not be implemented.

3.4 Conclusion design choices

The conclusions from this chapter are the choices that are made. These choices are the Hoberman sphere as the selected polyhedral structure, the slider crank mechanism as the motor set-up and the interaction controller as the control strategy.

4. Modelling structure

4.1 Introduction

As described in the previous chapter a model is necessary for the observer to find the velocity of the system. In the first part of this chapter the model of the Hoberman sphere will be explained. The mathematics that is used to calculate most of the equations in this chapter are explained in more detail in Appendix A. In this appendix, a simple planar case is modelled to check the method and the equations. The method and equations are written out and later checked with Mathematica, a mathematical software programme. In Appendix B the script is shown where various calculations are performed by Mathematica.

In the second part of this chapter the design of the Hoberman sphere is done and the limitations of the design is checked. This is done in a SolidWorks model where the Hoberman sphere will be designed. With the design of the Hoberman sphere done the motor set-up and parts for the sensor are designed with Solidworks. For the motor set-up the force distributions are also discussed.

The last part is to check the correctness of the mathematical and Solidworks models, by comparing them to each other.

4.2 Planar Hoberman sphere

In the next section the planar Hoberman sphere is described. This planar Hoberman sphere will be called a Hoberman sphere for the remainder of the report unless a 3D version is specified specificity.

The aim of this chapter is to find the EoM and thus the model of the Hoberman sphere. The EoM can be found by calculating the Lagrangian of the system. To calculate the Lagrangian the generalised coordinates needs to be found. This generalised coordinates can be used to calculate the various points of structure and velocity. In this single DOF case the generalised coordinate consists of only one coordinate. To find this variable first the schematics of the Hoberman sphere are explained.

In Appendix B a script is given that calculates the parameters that are needed to find the EoM of the Hoberman sphere. In the script the x & y coordinates of the various positions are derived, dependent on the number of scissor joints (n).

4.2.1 Schematics of structure

In Figure 4.1 a quadrant is shown of the Hoberman sphere. The o in the right corner of the figure is the origin. Two triangles that are connected (red and black) to one another are called a scissor joint. At the vertices of each triangle a black dot is added to symbolise a rotation point. In the figure there are three scissor joints shown. With the figure only being a quadrant of the Hoberman sphere, the number of scissor joint is twelve. The parameters l_3, l_4, l_5 *and* θ are variables of the Hoberman sphere. At every point (1,2 ... until 2n+12) the x & y coordinate is calculated. Every point is located on a circle, where the inner circle has length l_3, the middle circle has length l_4 and the outer circle has length l_5. The various points are situated on those circles. The first circle contains the points 1-12, the middle circle points 13-24 and the outer circle points 25-36. The green triangle in the figure is used to explain the calculation of the various lengths and is shown in the next figure. The points are also always on the dotted line in the figure. The angles of the dotted lines are fixed dependent on the number of scissor joints. The angle $2\pi/n$ in the figure is equal to $\pi/6$, because of twelve scissor joints.

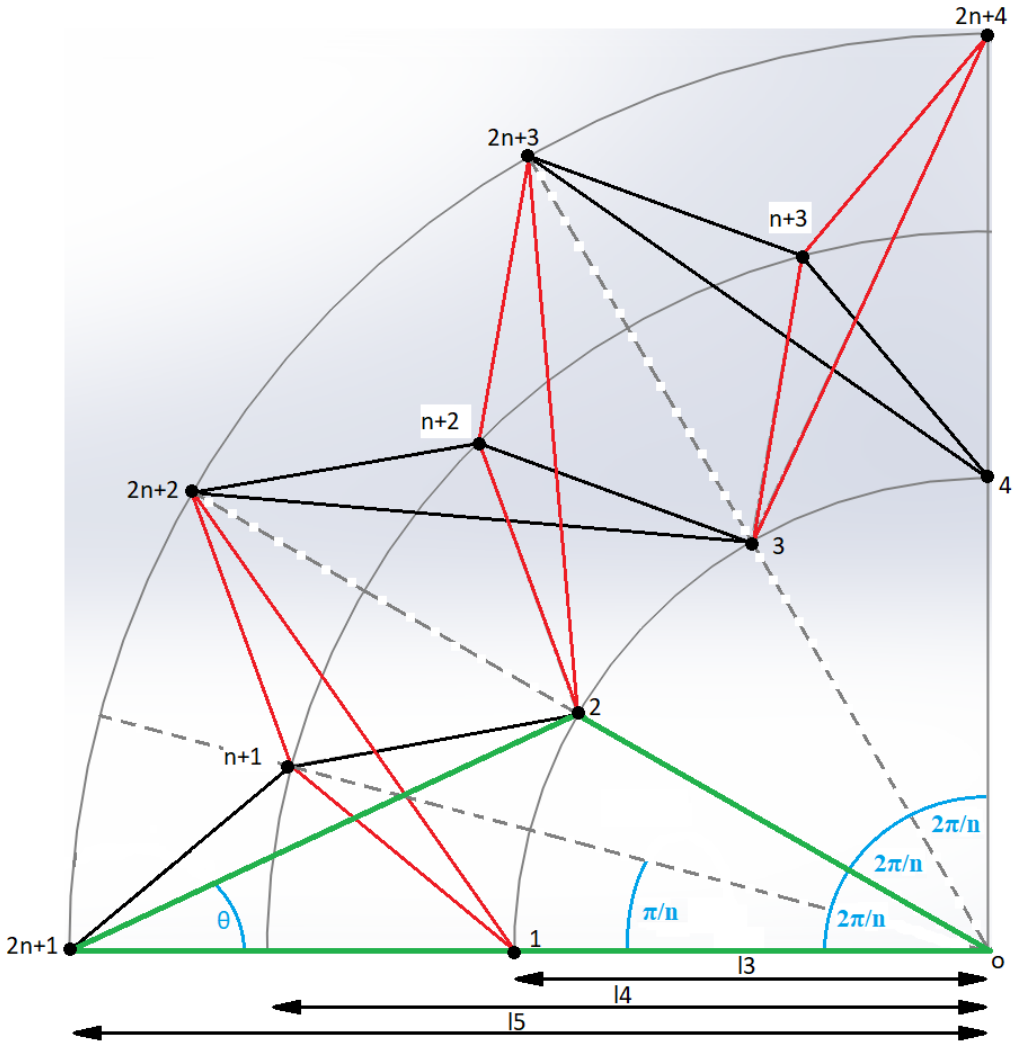

Figure 4.1: Hoberman sphere schematic

In Figure 4.2 a triangle of the Hoberman sphere is shown with the various parameters. The parameters l_1, l_2, α *and* β are constant lengths and angles of the triangles that make up the Hoberman sphere. The shape of the triangle is dependent on the number of scissor joints. In Equation 4.1 the relation between the constants is shown. The length of l_1 *and* l_2 will be determined in section 5.2.

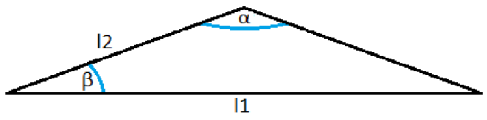

Figure 4.2: Triangle Hoberman sphere

$$\cos(\beta) = \frac{\frac{1}{2}l_1}{l_2} \qquad\qquad l_1 = 2l_2\cos(\beta) \tag{4.1}$$

To find out what the generalised coordinate is for this system, first the various variables are calculated. This can be done by using the x & y coordinates of the various points and trigonometry. In Figure 4.3 a part of Figure 4.1 is shown that better explains the calculation of the various variables. This figure uses the points 2n+1, 2 and the origin.

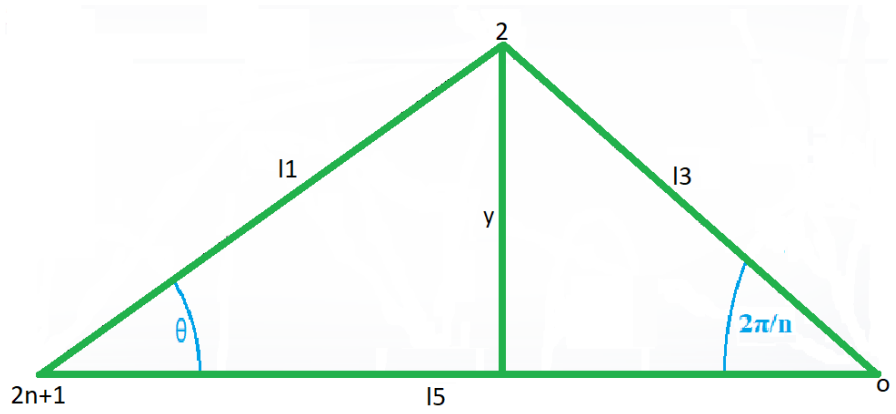

Figure 4.3: Hoberman sphere length calculation

In the figure there are two triangles with the same length y. The equations of the triangles calculating that length y should be equal to each other. The length l_3 is calculated in Equation 4.2. In the equation, y is the y-coordinate of point 2. The two unknowns in this equation are l_3 *and* θ.

$$\begin{aligned} y &= l_1 \sin(\theta) & y &= l_3 \sin(\frac{2\pi}{n}) \\ l_1 \sin(\theta) &= l_3 \sin(\frac{2\pi}{n}) & l_3 &= \frac{l_1 \sin(\theta)}{\sin(\frac{2\pi}{n})} \end{aligned} \tag{4.2}$$

The same method can be used to calculate l_4, where point n+1 in stead of point 2 is used. This is shown in Equation 4.3. The two unknowns are l_4 *and* θ.

$$\begin{aligned} y &= l_2 \sin(\beta+\theta) & y &= l_4 \sin(\frac{1}{2}(\frac{2\pi}{n})) \\ l_2 \sin(\beta+\theta) &= l_4 \sin(\frac{\pi}{n}) & l_4 &= \frac{l_2 \sin(\beta+\theta)}{\sin(\frac{\pi}{n})} \end{aligned} \tag{4.3}$$

The calculation of l_5 is found with the same method, only now the points 2n+1, n+1 and 1 are used. The equation is shown in Equation 4.4, where x(n+1) is the x-coordinate of point n+1 and the length from point 1 to the origin is known (l_3). With the knowledge that the x-coordinate of point n+1 is in the middle of the x-coordinates of points 1 and 2n+1, the length l_5 can be calculated.

$$l_5 = 2 \cdot x(n+1) - l_3 \tag{4.4}$$

With the equations about the various lengths it is shown that all length are dependent on θ. This means that variable θ can describe the full state of the structure and is therefore the generalised coordinate. With the description of the variable the x & y coordinates of the points can be calculated.

The x coordinates are calculated by:

$$\begin{aligned} x(i) &= l_3 \cos((i-1)\cdot(\frac{2\pi}{n})) & 0 > i \leq n \\ x(i) &= l_4 \cos((i-\frac{1}{2})\cdot(\frac{2\pi}{n})) & n > i \leq 2n \\ x(i) &= l_5 \cos((i-1)\cdot(\frac{2\pi}{n})) & 2n > i \leq 3n \end{aligned} \tag{4.5}$$

The y coordinates are calculated by:

$$\begin{aligned} y(i) &= l_3 \sin((i-1)\cdot(\frac{2\pi}{n})) & 0 > i \leq n \\ y(i) &= l_4 \sin((i-\frac{1}{2})\cdot(\frac{2\pi}{n})) & n > i \leq 2n \\ y(i) &= l_5 \sin((i-1)\cdot(\frac{2\pi}{n})) & 2n > i \leq 3n \end{aligned} \tag{4.6}$$

With the x & y coordinates and the generalised coordinate known the Lagrangian can be described.

4.2.2 Lagrangian and state space form

The equations to calculate the Lagrangian and the EoM are shown in Equation 4.7. In the first equation L is the Lagrangian, an energy function that summarizes the dynamics of the system. The other parameters in the equation are the kinetic energy (T) and the potential energy (V). With the Lagrangian the EoM can be found using the Lagrangian equation, see the right equation. The right equation is equal to zero when the system is autonomous.

$$\mathscr{L} = T - V \qquad \frac{d}{dt}\left(\frac{\partial \mathscr{L}}{\partial \frac{d\theta}{dt}}\right) - \frac{\partial \mathscr{L}}{\partial \theta} = 0 \tag{4.7}$$

To describe the Lagrangian of the Hoberman sphere some simplifications are applied to the system. These simplifications are:

- The potential energy in the system does not change.
- All masses are point masses and are situated in the points described at the Hoberman sphere schematics.

With the knowledge that this research only handles 2D, the first simplification is valid. The structure will be tested in the x-y plane, therefore there is no height difference. This is an important simplification for calculating the Lagrangian, because now the potential energy term will disappear. In chapter 7 tests are done to check if this simplification is justified.

To calculate the Lagrangian the kinetic energy should be known. For the kinetic energy the schematics of the Hoberman sphere will be used. In Equation 4.8 the kinetic energy is calculated. The mass is constant and the same for every point because of symmetry, therefore it can be taken out of the summation. The v_i is the velocity of a point in the schematic overview of the Hoberman sphere, see Figure 4.1. Due to the symmetry of the structure the velocities of the point masses corresponding to each length (l_3, l_4 *and* l_5) have the same value. With this symmetry the Hoberman sphere has only three different velocities, one for each ring. Each velocity is dependent on the angle θ.

$$T = \frac{1}{2}m\sum_{i=1}^{3n} v_i^2 = \frac{n}{2}m(v_{l_3}^2 + v_{l_4}^2 + v_{l_5}^2) \tag{4.8}$$

With the kinetic energy known, the Lagrangian is automatic also known because the potential energy is a constant and will disappear when calculation the EoM. The EoM is shown in in Equation 4.9, where the second equation of Equation 4.7 is used. In the equation beneath some constants (l_1, n, α *and* β) are already filled in, which gives the numbers. The lengths and the masses will be measured in the next chapters. More information about the method or calculations can be found in Appendix A & Appendix B.

$$-83.6l_2^2 m\left((1.46\cos(2\theta) + 2.54\sin(2\theta))(\frac{d\theta}{dt})^2 + (-3.22 - 2.54\cos(2\theta) + 1.46\sin(2\theta))\frac{d^2\theta}{dt^2}\right) = 0 \tag{4.9}$$

The result of the Lagrangian is an EoM that is not linear. This is because of the cosine and sine functions that are multiplied by the velocity and acceleration terms. The Lagrangian that is calculated will be used in the modelling of the plant in section 6.5.

4.2.3 Length vs θ

With the Lagrangian and the various relations known, a graph is made that gives the relation between the angle θ and the ring lengths (l_3, l_4 *and* l_5). For this however, the length of the scissor joints is necessary. This is given in section 5.2, where the physical prototype is explained.

In Figure 4.4 the relation is shown between the angle and the various lengths. In the figure the angle θ runs from $0 - \pi/2$rad. In the point based model the outer circle (l_5) can be smaller than the two other circles (l_4 *and* l_3). The minimum of length l_5 is 0.126m at 0rad, and the maximum is 0.253m and is achieved at about $\pi/3$. With these values the expansion ratio for this model is 2.

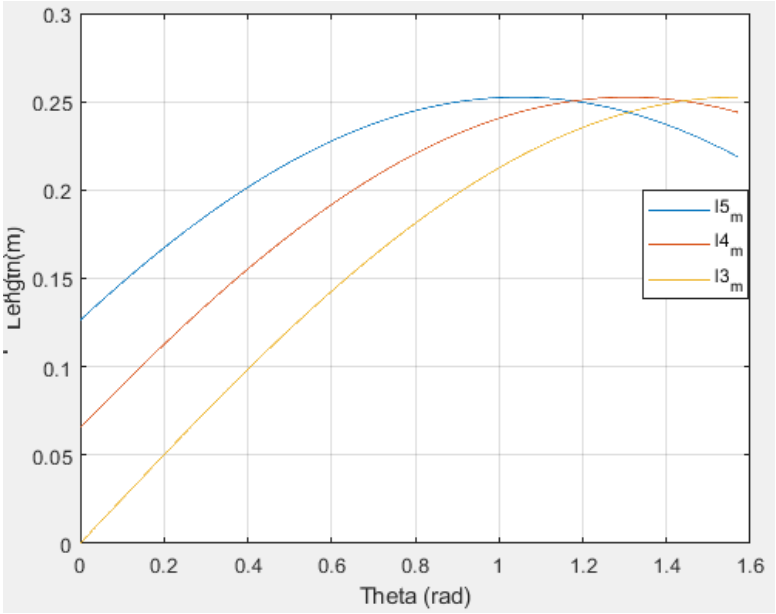

Figure 4.4: Length vs angle θ

The theory suggests in chapter 2 that the Hoberman spheres have an expansion ratio of 2-4. This is however dependent on the number of scissor joints in a quadrant. The Hoberman spheres described in the theory have more scissor joint per quadrant and thus have a larger expansion ratio.

4.3 Design planar Hoberman sphere

With the mathematical model, a point based approach is simulated. Now a different model is created to design the Hoberman sphere. This model will be simulated in such a way that it can be build as a prototype later.

4.3.1 Hoberman sphere

In Figure 4.5 the total SolidWorks model of the Hoberman sphere is shown. In the part to the left the reduced Hoberman sphere is shown and on the right the expanded version is shown.

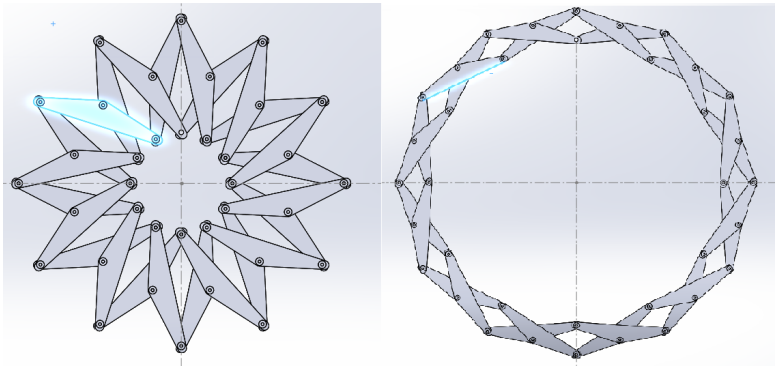

Figure 4.5: Hoberman sphere build up in SolidWorks

In this model the chosen number of scissor joints is twelve. The number of scissor joins is set to twelve because first it needs to be proven that this concept works in a real environment. If this concept works another iteration can be performed to increase the number of scissor joints and thus the expansion ratio. The number of joints is also not larger because the number of moving part also increases, which will result in more friction in the system.

In Figure 4.6 the scissor part is shown which build up the entire Hoberman sphere, were l_1 and l_2 are the same lengths as in the Mathematica model. However the actual length of the total scissor part is equal to l_l and the height l_h, the holes that are used in the drawing are m(i) thread size. When the thread size is increased the size of scissor part should be adapted automatically.

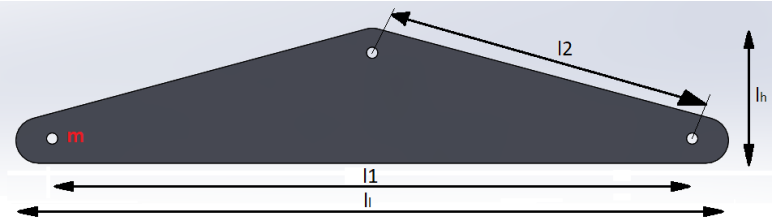

Figure 4.6: Scissor part

In Figure 4.7 the angle θ vs lengths l_5 *and* l_3 is shown. This Hoberman sphere model maximum l_5 is 0.25m and smallest length is 0.14m given a ratio of 1.83. In this figure the range of θ is different. The range goes from 0.04-1.22rad. This is because in this model constraints are added, such that l_5 cannot be smaller than l_4 *or* l_3.

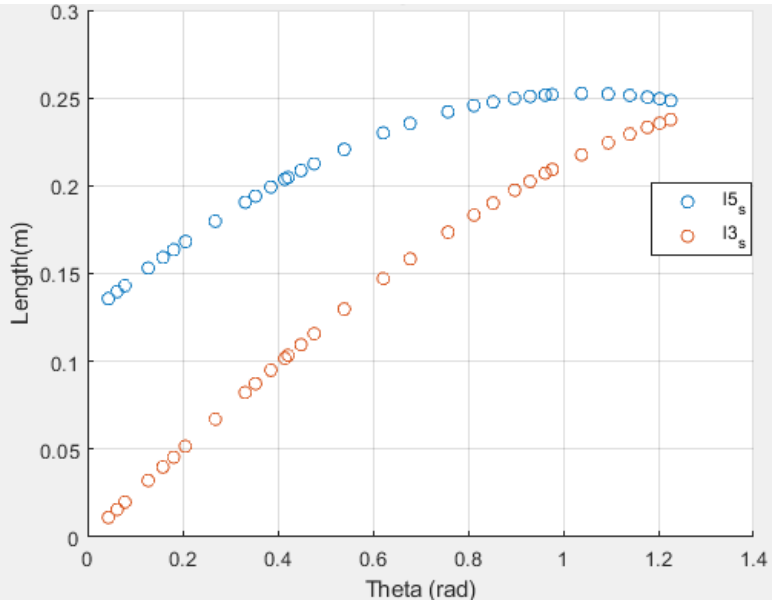

Figure 4.7: Length vs angle θ

In the figure the data is obtained by using the measuring tool in SolidWorks and making the measurements of θ, l_4 & l_5 into sensors. This is done so it can be read easily every time the sphere is changing and updated. With the "move component" option the sphere is expanded, and this is done multiple times. The readings of the sensors are put into Excel and that table is imported into Matlab. This is the reason why the figure gives points and not a solid line.

4.3.2 Sensor

To control the Hoberman sphere set-up a sensor is necessary. The sensor that is going to be used is explained in section 5.4. In this part the SolidWorks model for the sensor is explained.

The sensor consists of two parts, a magnet and a chip. To use the sensor on the Hoberman sphere two modules are designed and 3D printed to measure the angle. The first module is to hold the magnet in place and the second module is to hold the chip and printed circuit board. These two modules are shown in Figure 4.8. The left one is the printed circuit board holder and the right one is the magnet holder. The printed circuit board holder has four pins where the printed circuit board fits onto. The magnet holder module has a hole in it where the magnet can be placed and some glue is added to make sure it stays. Both designs have the same shape as the scissor parts to fit them correctly onto the Hoberman sphere. These shapes are chosen because it is very important for the sensor that both, the chip and the magnet, are correctly aligned with each other and the rotation point.

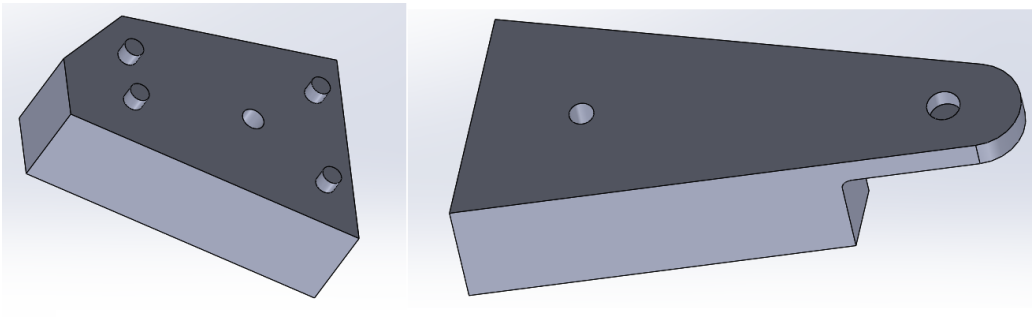

Figure 4.8: Sensor modules

The total sensor set-up is shown in Figure 4.9 where the printed circuit board is also attached to the module. On the left side of the figure the sensor set-up is shown and where it is placed onto the scissor parts. Here it is shown that all modules are aligned with the rotation point. On the right side the spacing is shown between the modules. The maximum distance between the chip and the magnet is 3mm, according to the datasheet [22]. The printed circuit board is 1.6mm thick. To ensure that the distance between the magnet and the chip is smaller than 3mm, a hole is made in the circuit board to lose the 1.6mm. With this solution the distance between the chip and the magnet is now about 1.5mm.

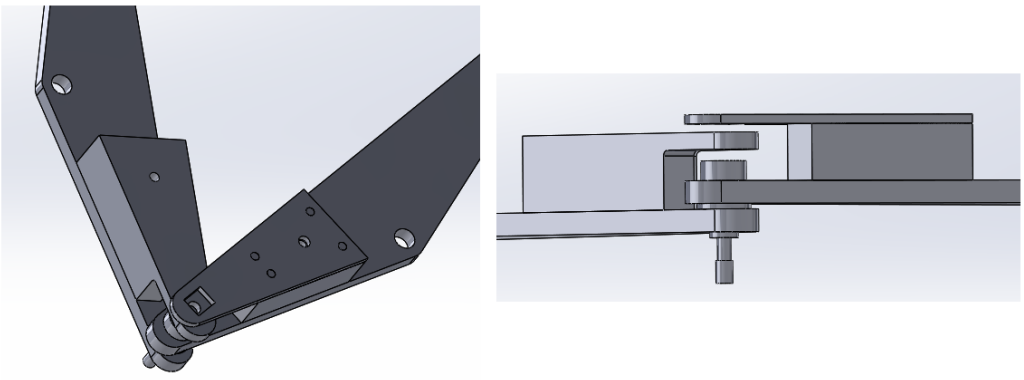

Figure 4.9: Sensor set-up

The sensor has two parts that will be connected onto different scissor parts, where it directly can measure the angle θ. This can be seen in Figure 4.10, where the sensor measures two times the angle θ. The sensor itself is located directly above the 2n+1 point. This point can however be every point on the Hoberman sphere due to symmetry.

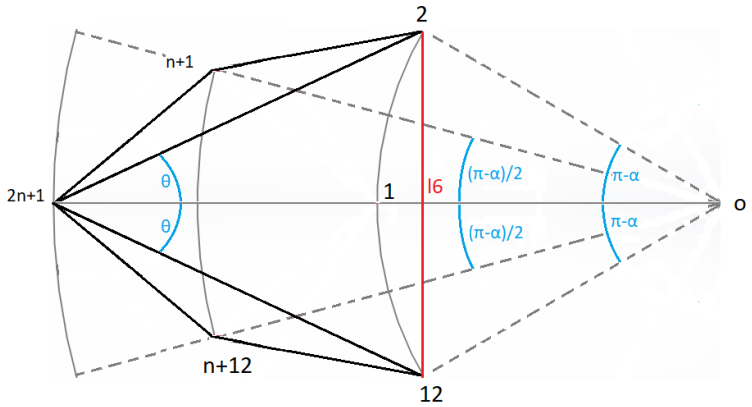

Figure 4.10: The angle measured by the sensor

The length l_6 in the figure is an help line to determine the angle θ, see Equation 4.10.

$$\theta = \sin^{-1}\left(\frac{0.5 * l_6}{l_1}\right) \tag{4.10}$$

4.3.3 Motor set-up

With the SolidWorks Hoberman sphere model given, the motor set-up that is described in section 3.2 can be added to the model. In Figure 4.11 the motor set-up that can be used on the existing Hoberman sphere model is given. The set-up is comprised of 2 parts, the ellipse and the two clevises with a rod in between. As is shown in the top part of the figure the two clevises are not of the same size. This is done to avoid collision, where the one connected to the ellipse has a round shape for that reason. The thickness of the ellipse and the middle blocks within the clevis are 9mm corresponding with the thickness of the Hoberman sphere. The clevis has a fixed connection with the bar and a rotational connection with respect to the Hoberman sphere and the ellipse.

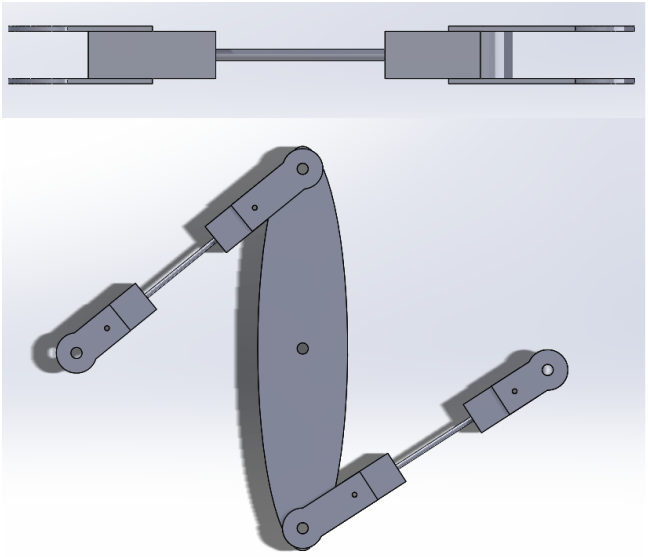

Figure 4.11: Motor set-up model in SolidWorks

The motor itself will be attached to the ellipse, which will be shown in subsection 7.2.1. The set-up itself is connected to the Hoberman sphere by connecting the clevis to a part of the scissor joint. Both clevises will be attached to opposing scissor joints, where the clevis fits onto the bolt and rings that connect it to the Hoberman sphere.

With the motor set-up attached to the Hoberman sphere the expansion ratio drops. With collisions the range of θ decrease to 0.42-0.98rad. The part that cause the collision is the middle part of the clevis which is attached to the ellipse. Due to the collision the minimum length of l_5 is 0.205m and the maximum is 0.252m which results in a 1.23 expansion ratio.

The calculated Lagrangian is however without a motor set-up. To find the relation between the motor set-up and the Hoberman sphere some additional calculations are performed. Figure 4.12 explains the forces and torques that are at play from the motor set-up.

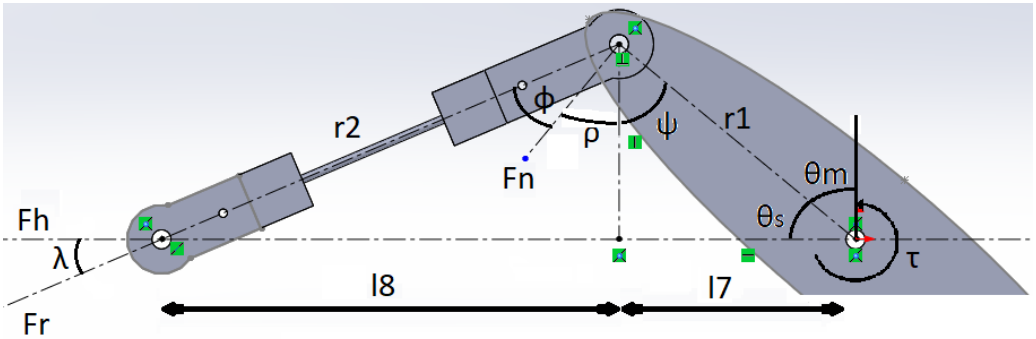

Figure 4.12: Torque and forces of motor set-up

With the figure the different forces can be calculated, see Equation 4.11.

$$\tau_m = F_n r_1 \qquad F_r = F_n \cos(\phi) \qquad F_H = F_r \cos(\lambda) \tag{4.11}$$

In these equations none of the parameters are known yet. However in the next chapter the dimensioning of the scissor part and the motor set-up is set. The parameter θ_m is calculated in subsection 7.2.2 where the relation between θ and θ_m is discussed. In the figure, θ_m is defined in this way such that the other angles can be calculated. With the set parameters, the other parameters can be calculated using the following equations.

$$\begin{aligned} l_3 &= l_7 + l_8 & l_7 &= \cos(\theta_s) r_1 & l_8 &= \cos(\lambda) r_2 & \\ \theta_s &= \frac{\pi}{2} - \theta_m & \psi &= \frac{\pi}{2} - \theta_s & \rho &= \frac{\pi}{2} - \psi & \phi &= \frac{\pi}{2} - \lambda - \rho \end{aligned} \tag{4.12}$$

In the Lagrangian equation, that is calculated in subsection 4.2.2 the system was autonomous. The Hoberman sphere will be used with a motor, so the system is not autonomous. With that fact the Lagrangian equation changes, see Equation 4.13. The τ in the equation is the generalized torque.

$$\frac{d}{dt}\left(\frac{\partial \mathscr{L}}{\partial \frac{d\theta}{dt}}\right) - \frac{\partial \mathscr{L}}{\partial \theta} = \tau \tag{4.13}$$

The τ that is described in the equation is the torque from the motor. However because of the motor set-up there are losses due to force misalignment. These losses will be added in the actuator block inside Simulink. In Equation 4.14 the relation between the forces F_H&F_n is the efficiency of the system. The angles used in the equations are dependent on the motor angle θ_m, which is in turn dependent on the Hoberman sphere angle θ. This is why the efficiency of the motor set-up can be related to the Hoberman sphere angle.

$$A = \cos(\lambda)\cos(\phi) \tag{4.14}$$

With the λ and ϕ known, the efficiency can be calculated with the previous equation and is shown in Figure 4.13. This efficiency is optimal for θ at 0.72rad and a level of 82.3%. These angle are checked with the SolidWorks model, where the angle ϕ is one where F_n and F_r have the same direction so are in line with each other. The angle λ is maximum when the forces F_r and F_h are in line, which is the fully open state. The combination of those angles is thus the efficiency of the motor. The efficiency is almost zero at the fully opened state. That is due the fact that the F_n is almost perpendicular to the F_r.

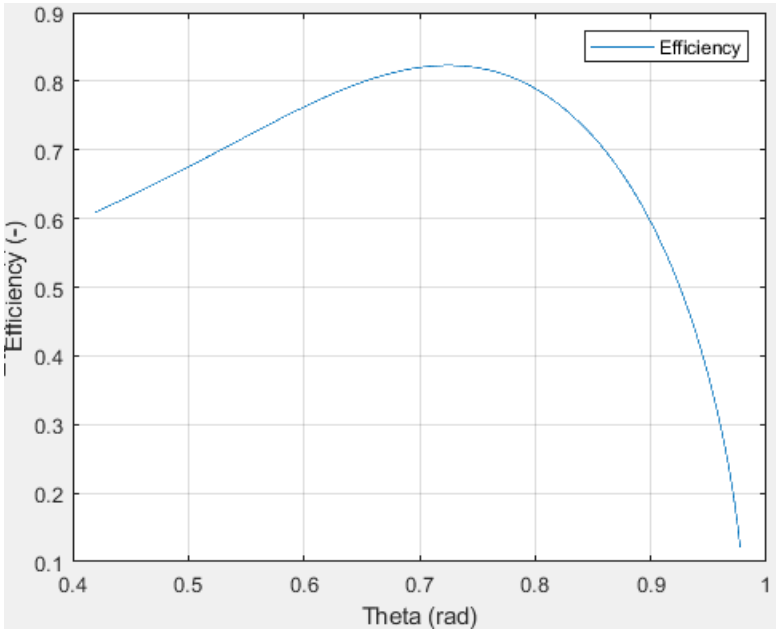

Figure 4.13: Efficiency of the motor set-up

4.4 Checking models

To check if the models that are build in Mathematica and SolidWorks are correct the length l_5 and l_3 are checked with respect to θ. This extra check is done to make sure that the prototype that is going to be build is correct.

Furthermore if the models are not the same there is an error made in on of the models and it should be fixed.

The combined results of the Mathematica and SolidWorks model are shown in Figure 4.14 where l_4 is also plotted to check the relation. For the SolidWorks model only the Hoberman sphere version is shown so not with the motor set-up. As shown in the graph the models are identical, except at the beginning and the end-point. The Mathematica model is defined over the entire range, where the SolidWorks model has a limited range. In the figure the Mathematica model lines are fully defined with an equation, where the SolidWorks model are measurements thus not known over the entire range.

In the figure it stated that l_4 and l_3 can become larger than l_5. In the Mathematica model there are no constraints added to the system on how the movement is, due to it being point based. However due to physical constraints this is not possible in the SolidWorks model. In the Mathematical model the lengths are calculated using coordinates of points and are not constraint with respect to θ.

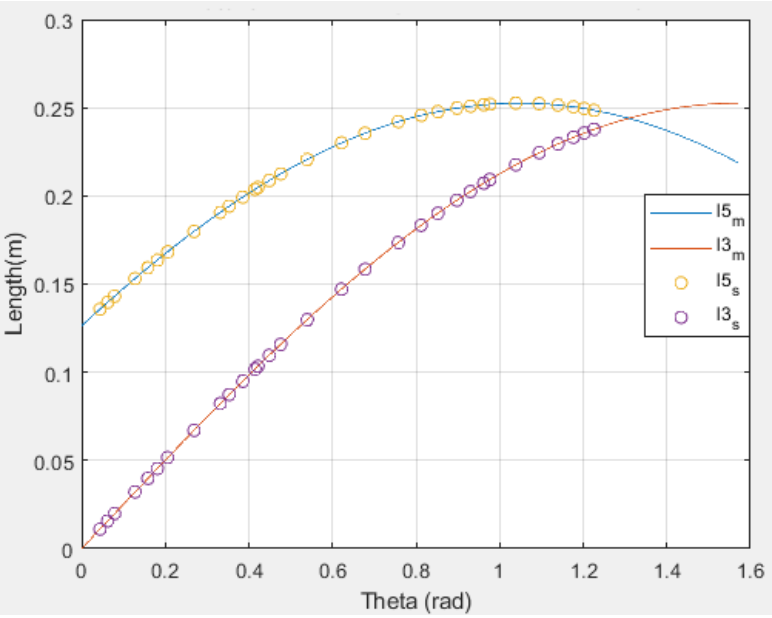

Figure 4.14: Lengths vs θ

4.5 Conclusion modelling structures

The aim of the first part of this chapter is to find the EoM, so that it can be used in a model for the observer. The first conclusion from the modelling of the Hoberman sphere is that the generalised coordinate for this structure is the angle θ. The other important conclusion is that EoM is found and shown in Equation 4.9.

The torque found in the Lagrangian is the torque applied to the Hoberman sphere without a motor set-up. With the motor set-up attached to the Hoberman sphere the efficiency changes, see Figure 4.13.

The mathematical model has an expansion ratio of 2, where the designed SolidWorks model has an expansion ratio of 1.83. With the motor set-up attached to the Hoberman sphere the expansion ratio deteriorates further to 1.23. Furthermore two parts are designed for the sensor to measure the angle θ.

Finally both, the mathematical and design, models have the same relation between the ring lengths and the angle θ.

5. Physical prototype

In this chapter the physical prototypes of the Hoberman sphere are shown and explained. Furthermore a sensor and the motors that are used in testing are explained. In Appendix C the bearing options are shown that are used for one of the prototypes.

5.1 K'nex iteration

The first iteration was one built from K'nex. Due to the symmetry within both the simple planar case and the Hoberman sphere it was possible to create a prototype using K'nex. This K'nex version was made to do some early testing/building during the early stages of the thesis, and it is shown in Figure 5.1. With this iteration an idea can be given about the expansion ratio of the actual Hoberman sphere. Furthermore it can be checked what the influence is when using a motor set-up inside the structure. There are two versions build from K'nex, a small and a bigger version. The difference between the two is the length of the bar, that is l_2 in the schematics of the Hoberman sphere,. In the figure the grey rods on top are there to fix the rotation so that it expands evenly. Furthermore the figure has the same number of scissor joints as the Mathematica model.

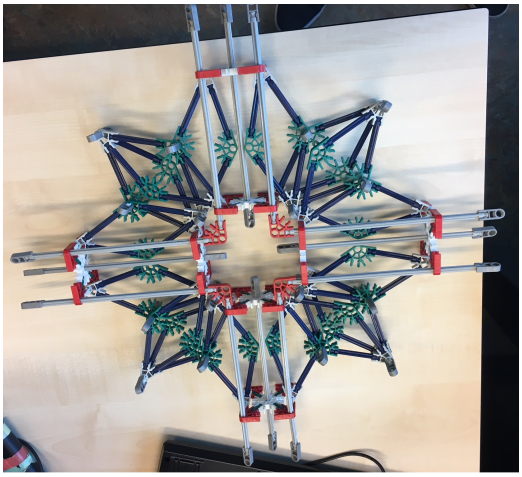

Figure 5.1: K'nex Hoberman sphere

In Table 5.1 the expansion ratio is given of the two K'nex iterations. According to the theory the expansion ratio should be higher. The expansion ratio's of the Hoberman sphere should be above 2, see section 3.1. The K'nex iteration has however a ratio of about 1.25. This low ratio is mainly due to the constraints K'nex has. In the figure the minimum retraction is shown, where the middle part of collides with one another. These parts are there to connect the scissor joints together but constraint the movement.

Table 5.1: K'nex Hoberman sphere planar

K'nex structure	l_2 (m)	$l/l_0()-$
Small	0.057	1.21
Big	0.0873	1.27

The main observation from the K'nex iteration is that the expanding ratio will be lower than the theory describes. This is as explained due to the limitations of the K'nex. However from this experiment the actual prototype will probably also have an expansion ratio of about the same as the K'nex iterations. This is because the space in the middle that is not used by the K'nex iteration will be filled in the prototype by a motor set-up. Therefore the expansion ratio will also be lower than the theory.

5.2 Hoberman sphere prototype

The design model of the Hoberman sphere is already discussed at section 4.2. In this section that model will be 3D printed and checked.

First the size of the scissor parts need to be determined. This is done by checking the different sizes of the scissor parts. The aim here is to have scissor parts that do not bend easily and are not to big (impractical). To check what thickness should be used, three different versions of the scissor part are printed. These versions have different thicknesses, namely 2,3 and 5mm. From these printed version, the 5mm thick scissor part is chosen because it was strong, didn't bend, is not too large and is usable with the bearing options.

In Table 5.2 the parameters of the newly designed scissor part and the washers are given. The scissor part is changed from 5mm to 4mm because of the bearing and bolt options described in the appendix. The 5mm ring is used between the bolt and the first scissor part and between both the scissor parts, the M3 ring is used between the scissor part and the nut.

Table 5.2: Final scissor part 4mm

Structure	$l_1(m)$	$l_2(mm)$	$l_l(mm)$	$l_h(mm)$
Scissor part 4mm	126.28	65.37	140.64	26.65
Ring 5mm	9.5	5		
Ring M3	8	3.2		

In Figure 5.2 the prototype of the Hoberman sphere is shown. In the figure the motor set-up is already attached to the Hoberman sphere. However in the figure it is clearly shown that the expansion ratio is rather limited due to the motor set-up. The maximum of l_5 is 0.251m and the minimum is 0.204m given an expansion ratio of 1.23. This is however as expected because of the result from the SolidWorks model, which should be the same.

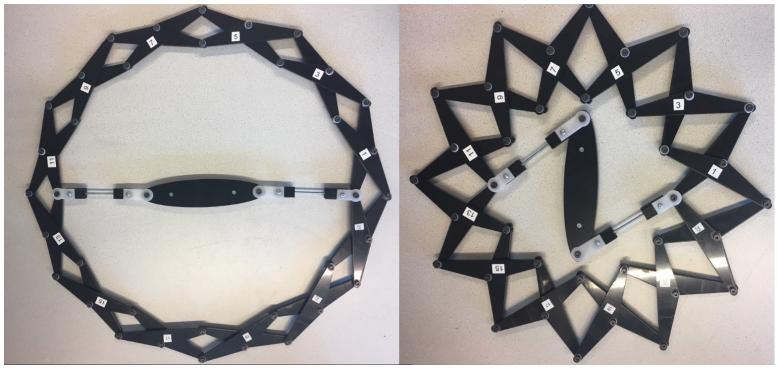

Figure 5.2: Hoberman sphere laser-cut version

From the figure it is shown that the motor set-up is designed not to get to the maximum expansion. Due to the relation between θ and l_5 described in the previous section, the change in length of l_5 decreases when θ is close to the maximum. When θ is almost $\pi/2$rad, there is almost no change in length. Therefore the motor set-up is designed where the change in length is the highest close to the maximum of θ. If the actual maximum of θ is used a lower expansion ratio will be the result. With that the radius of the ellipse is set to 0.08m and the length for clevis to clevis is 0.1294m. These lengths are between the two rotation points.

5.3 Motor

For the actuation of the Hoberman sphere there are two options that are used in various tests.

To find a motor for the various tests some requirements are set. These requirements are limited because of set-up and the use of the Arduino, which have certain limitations. The requirements are:

- A torque of about 0.5Nm.
- A velocity lower than 200RPM.
- Maximum current of 1.2A
- Maximum voltage of 13.5V

The torque requirement of 0.5Nm is set not too high, to make sure that the connection from the motor to the Hoberman sphere does not brake. The velocity is chosen not to be too large because the system itself has a limited range. The maximum range for the motor to move is about π thus when using a RPM of 60 the system could reach its setpoint at 0.5s. This time is however theoretical because the motor still needs to move the load and get to the given velocity. The maximum current and voltage are given because of the Arduino set-up that is being used. For that set-up there are maximum values that are allowed onto the system.

5.3.1 Servo motor

A servo motor has the advantage that it has a built in encoder and control system. For this reason the start-up tests are performed with that type of motor. The servo motor is a VS-11AMB SERVO, with datasheet [20]. This servo is used because it was available at the university, it could apply sufficient torque and was easily added to the Arduino system. The datasheet of the motor was very limiting however the two relevant and almost only information are shown in Table 5.3. In the table the units are converted to the units of the requirements to better compare.

Table 5.3: Relevant specification of VS-11AMB

Specification	at 4.8V no load	at 6.0V no load
Operating velocity	45.5RPM	52.6RPM
Stall torque	2.16Nm	2.55Nm

The connection to the Hoberman sphere is made using an extension of the servo motor itself. This extension is fixed to the axis of the motor and is screwed onto the ellipse, and can be seen in subsection 7.2.1.

The problem however with a servo motor is also one of its advantages, namely that it has already a control system. This means that controller is already fixed for the test that it will be used in.

5.3.2 DC-motor

A DC-motor has standard no encoder or control system on it. Therefore when designing a custom controller, a DC-motor is necessary. The DC-motor that is chosen for the experiments is a Crouzet brushed DC geared motor, [21]. A geared motor is chosen because of the low velocity requirement. This motor has the following specifications:

Table 5.4: Specification of Crouzet

Specification	Value
Operating velocity	140RPM
Torque	0.5Nm
Voltage	12VDC
Maximum usable power	3.8W
Weight	160g

This motor is chosen because it was relatively cheap, it fits the requirements that are set for the motor and it can be used for the various tests. For the DC-motor a flexible beam coupling is bought to make a connection between the motor set-up and the motor itself. The coupling is fixed to the axis of the motor and to a coupling piece that is added onto the motor set-up. This coupling piece is a thin plate with holes added to screw it to the motor set-up and an axis in the middle for the connection to the flexible beam coupling.

5.4 Sensor

The servo motor has an internal encoder that is used as a rotational sensor. However for the DC-motor a sensor is necessary to correctly control the physical prototype. This sensor gives information about θ, the to be controlled variable. This measurement can be done indirectly with an encoder on the motor that measures the angle of the motor or a laser sensor for measuring the expansion length l_5. Both of those measurements give information about the angle θ. However a direct measurement of θ would be better, because of the transformation errors. Furthermore the input of the reference signal is also given in the θ range, see chapter 6.

A direct measurement of θ could be done with rotary sensors. With the Hoberman sphere however it is difficult to connect a rotary sensor to the system because most rotary sensors need to have a shaft that rotates. In the Hoberman sphere the shoulder bolt does not rotate with the sphere, therefore there is no fixed point. With a magnetic rotary position sensor this problem does not occur. Other advantages of the magnetic rotary sensor is that it can measure the angle contactless, has a high accuracy and is relatively cheap. With the sensor being contactless there is no added friction to the system, which is important for the controller. This is why this sensor is chosen to measure the angle θ.

The magnetic rotary sensor works with a chip, namely AS5600 ([22]) and a special rotary magnet. The chip has a 12-bit resolution and the start position is dependent on how the chip is rotated and can be programmed to give full resolution at a set angle range. However for the Hoberman sphere it only needs to have a range of about 0.35-1.22rad. The output of the sensor is a PWM encoded signal. This signal is converted to an angle by a library for the Arduino.

The angle θ is measured by checking the length between two inner bolts as discussed in subsection 4.3.3 and shown in Figure 5.3.

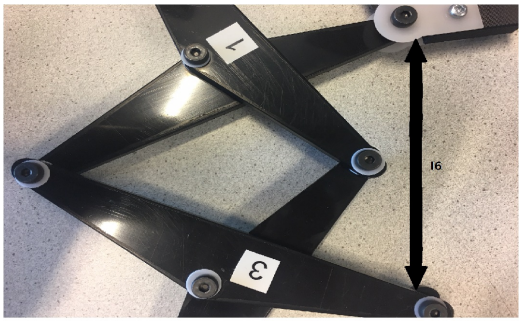

Figure 5.3: Angle measurement

5.5 Conclusion physical prototype

The conclusions from this chapter are that a prototype is successfully made from the SolidWorks model. The prototype has the same expansion ratio of 1.23 as that of the designed SolidWorks model. The sensor that is going to be used on the Hoberman sphere, is a magnetic rotary sensor and is placed at the rotary point of a scissor part.

6. Controller design

6.1 Introduction

From the information in section 2.2 and the choice made in section 3.3, the controller that is going to be used for the inspection robot is an interaction controller. However as explained in that same section, the interaction controller needs a model to correctly estimate the velocity of the system. To validate the model, a position based controller is used. This validity of the model can be checked when it is compared to the prototype with various tests. These test will be described in chapter 7. A position controller is used with the current set-up because it is easily implemented and accessible in both the prototype and the different models. Due to these reasons a position controller will be explained in this chapter. First the controller set-up will be explained and from that each individual component.

6.2 Controller set-up

In this section, the control loop of the Hoberman sphere shown in Figure 6.1 is explained. This controller set-up is designed for the tests performed in section 7.4 where a DC-motor is used. In the figure there is a reference profile, controller (C), actuator (A), plant and a sensor (S). These parts will all be explained in the next sections. The other items in the Simulink model are needed to get the desired measurements. In the figure there are various parameters given, where $Theta_x$ is the angle from the reference profile, the $Theta_e$ is the error between the reference and the output and theta is the angle from the Hoberman sphere. For the plant the input is U and the output is Theta. The U is a input vector for the plant and Theta is the angle from the Hoberman sphere. The plant will be further explained in its section.

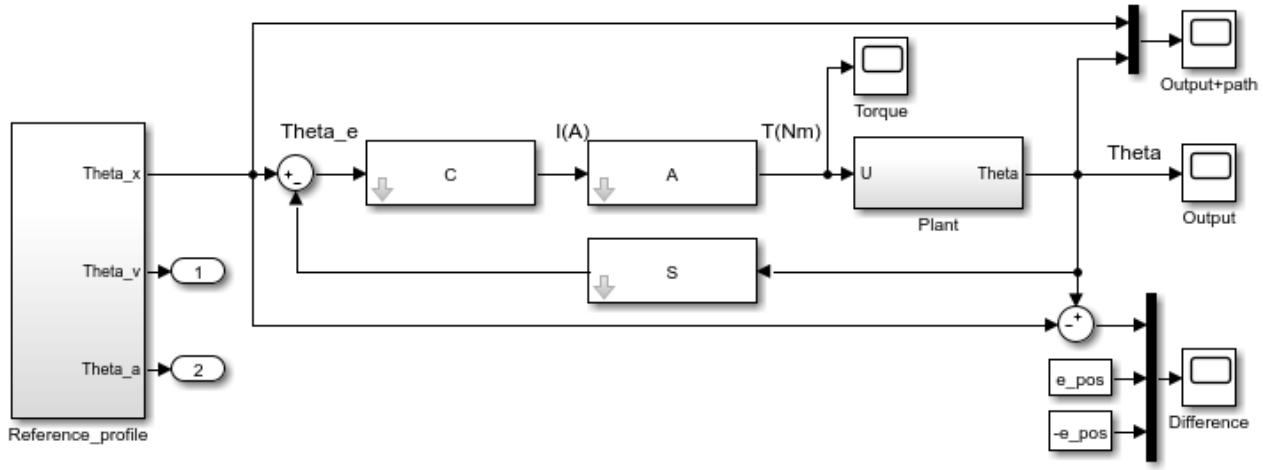

Figure 6.1: Control loop Simulink

With the control set-up there are also some simplifications in the actuating and sensing part.

- The motor is ideal and has no dynamics.
- The sensor is ideal.

6.3 Controller

The controller that is used here will be a PID controller. This controller is chosen because it has the following advantages:

- Loop shaped around crossover frequency (ω_c)
- Increased bandwidth
- Added damping
- Better disturbance rejection

Another reason why the PID controller is chosen and not a more complex position controller has to do with the Arduino, that is used in the testing. The Arduino has limited space available for coding. An advanced controller will take too much space, where also other tasks should be performed by the Arduino. Furthermore there is a library already available about the PID controller that can be used in this set-up. The Arduino set-up will be further explained in the subsection 7.2.1.

The PID controller is shown in Equation 6.1 and the parameters are described in Equation 6.2, [23]. These equations are set in the parallel configuration. In the equation the parameters m, β *and* ω_c are still unknown. The mass and the crossover frequency will be determined in chapter 7, where β is a constant of 10. The parameter β is a tameness factor and adds phase margin to the system.

$$C(s) = K_p + \frac{K_i}{s} + K_d s \tag{6.1}$$

$$K_p = \frac{3m\omega_c^2}{2\sqrt{\beta}} \qquad K_i = \frac{m\omega_c^3}{\beta} \qquad K_d = m\omega_c \tag{6.2}$$

Now the whole controller is dependent on ω_c.

6.4 Reference profile

In Figure 6.2 the Simulink submodel of the reference profile is shown. The reference profile is created for an acceleration signal and is integrated twice to calculated the reference angle.

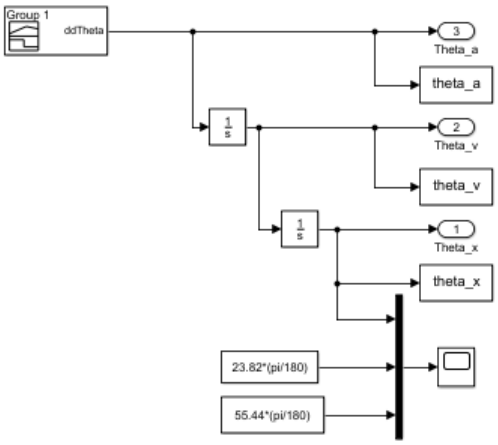

Figure 6.2: Reference profile Simulink

From the knowledge that the Hoberman sphere's needs to expand to make contact with a certain surface, it is known that this is a setpoint problem. For a reference signal there are three common signals, namely skew sine, second order and third order polynomials. These signals will produce low error signals depending on the type of controller and the type of problem. With this already being a setpoint problem, the controller need to be decided. For a PD controller the second order polynomial will results in the lowest error signal and for a PID controller the third order or skew sine [23].

With the PID controller selected, a 3rd order reference profile is the best choice. From a third order polynomial it is known that the acceleration profile looks like a triangle wave, this is shown in Figure 6.3. In the

figure on the right side is the triangle wave and on the left the result of the reference profile on the angle. The reference profile that is selected here has a h_m of 0.56rad and a t_m of 5s. The value for h_m is chosen because that is the range of the Hoberman sphere, see measurement in subsection 7.2.2. This is a fixed/constant value. However the t_m is set for now as five and will differ depending on a measurement and the requirement for the crossover frequency.

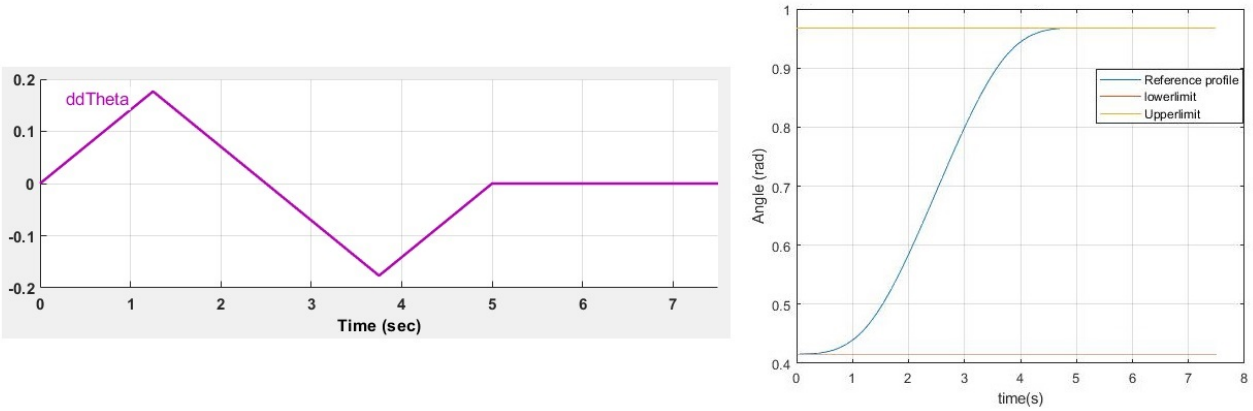

Figure 6.3: Acceleration and angle profile

With the h_m and t_m set for now, the maximum jerk and acceleration can be calculated to fully define the reference profile, see Equation 6.3.

$$\frac{d^2x_{max}}{dt^2} = \frac{8h_m}{t_m^2} \qquad \frac{d^3x_{max}}{dt^3} = \frac{32h_m}{t_m^3} \tag{6.3}$$

6.5 Plant & Actuator

6.5.1 Plant

The plant of the Hoberman sphere is almost completely explained in chapter 4. However in the previous chapter the length of each parts was determined and the mass of the system is given in section 7.1. The length will be filled in and it results in an edited EoM, see Equation 6.4.

$$\tau = m((-0.52 + 0.45\cos(2\theta) - 0.91\sin(2\theta))(\frac{d\theta}{dt})^2 + (1.15 + 0.91\cos(2\theta) - 0.52\sin(2\theta))\frac{d^2\theta}{dt^2}) \tag{6.4}$$

The non-linearity in the system is dealt by using a state space representation of the system and calculate the A and B matrices in the loop. This is shown at Figure 6.4, where in the function block the Lagrangian equation is used to calculate the states. The mux and demux blocks are necessary to only give the correct state, the angle, to the rest of the system. Inside the integrator is the starting angle as an initial condition.

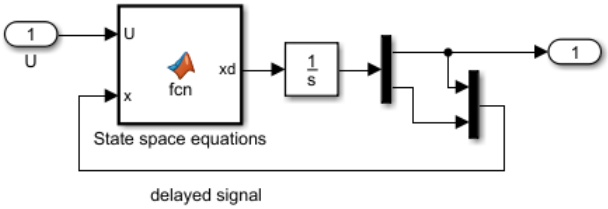

Figure 6.4: Non-linear plant Simulink model

As shown in the figure, the input of the non-linear model is the state vector (x) and input vector (U). The output of the model is the differential of the state vector (xd). To use the model, the EoM needs to be rewritten to fit the non-linear state space description, as shown in Equation 6.5 & Equation 6.6. Equation 6.6 is used in the function block.

$$\frac{d^2\theta}{dt^2} = \frac{1.1\tau + (0.58m\cos(2\theta) + m\sin(2\theta)(\frac{d\theta}{dt})^2)}{m(1.27 + \cos(2\theta) - 0.58\sin(2\theta))} \tag{6.5}$$

$$\begin{aligned} \frac{dx_1}{dt} &= x_2 \\ \frac{dx_2}{dt} &= \frac{1.1U + (0.58m\cos(2x_1) + m\sin(2x_1)x_2^2)}{m(1.27 + \cos(2x_1) - 0.58\sin(2x_1))} \end{aligned} \tag{6.6}$$

6.5.2 Actuator

In simulation block A the actuator of the controller set-up is situated. Also added in this simulation block is the efficiency of the motor set-up that is described in subsection 4.3.3. That actuator in the control set-up is a DC-motor. The servo motor has an internal control set-up so there is no need to create one for that actuator. The motor specifications are explained in subsection 5.3.2. From those specifications the motor constant (k_m) is calculated.

$$I = \frac{P}{U} = \frac{3}{12} = \frac{1}{4}A \qquad\qquad k_m = \frac{\tau}{I} = \frac{0.5}{0.25} = 2Nm/A \tag{6.7}$$

6.6 Conclusion controller design

For the controller set-up a 3rd order reference profile with a PID controller is selected. This combination is selected because of the environment where the Hoberman sphere is used. The controller set-up will be used to verify if the created model from chapter 4 is correct.

For the Lagrangian to be used in the model it needs to be rewritten into state space equations. This non-linear state space equations are shown in Equation 6.5 & Equation 6.6.

7. Measurements & results

In this chapter various tests will be described and explained. In each section the reason why certain tests are performed is discussed. Furthermore the measurement set-up and the results of those measurement will be discussed and analysed. This chapter is divided into various parts. The first part is finding the masses of the system. After that various testing is done on a servo & DC motor. In the final section of this chapter the comparison between the model and the prototype is analysed to check the correctness of the model itself.

7.1 Masses of the Hoberman sphere

For the model and the controller it is important that the masses of the system are known. These masses are inside the equations of the Lagrangian, which the controller needs to control. The mass is necessary to properly determine those functions. In Table 7.1 the masses are shown for various parts of the Hoberman sphere. These masses are measured using a precise scale with an error of 1^{-3}g.

Table 7.1: Masses of the Hoberman sphere

Measurement (-)	Bolt (g)	Triangle (g)	Nut (g)	Scissor joint (g)
1	2.81	13.91	0.45	31.30
2	2.82	13.90	0.44	31.14
3	2.80	13.91	0.45	31.31
4	2.82	13.88	0.45	31.34
5	2.97	13.87	0.44	31.28
Average	2.81	13.89	0.45	31.27

The scissor joints that are measured have only one bolt & nut, thus the equation beneath gives almost the same result as the measured scissor joint (31.27g). The rings are not measured because their individual weight is not significant, from the measurement however the three rings combined is about 0.23g per connection. The total Hoberman sphere is built up of twelve scissor joints. However in these joints there are two triangles, three bolts and 3 nuts necessary.

$$m_{joint} = 2m_{triangle} + m_{bolt} + m_{nut} = 31.04 \quad m_{HS} = 12 \cdot (2m_{triangle} + 3m_{bolt} + 3m_{nut} + 3m_{rings}) = 458.28g \quad (7.1)$$

The mass of the total system is described there, where here the mass of the individual point is used. The mass that should be implemented here is 458.28/36g. This is the total weight of the Hoberman sphere divided by the number of point masses used in the calculation. This mass is set in the Lagrangian equation of the model. With the masses known the total Lagrangian is known and the test on the prototype can be performed.

7.2 Servo motor

With the servo motor and set-up, the designed Hoberman sphere can be checked if everything works correctly. This is done with the servo motor because it has already a built in encoder and controller. With the servo motor various tests can be performed and the design & software can be checked. During this stage of the research the sensor described in section 5.4 is not yet available.

7.2.1 Test set-up

In Figure 7.1 the first test set-up of the Hoberman sphere is shown. In this test set-up there is the designed Hoberman sphere, motor set-up, Arduino and a supporting structure. The supporting structure is used to fix the motor, such that the rotation of the motor results in a movement of the sphere. Another fixation is placed

on the Hoberman sphere itself, which is shown in the figure by two red arrows. This is done to prevent the Hoberman sphere from rotating, instead of translating. These fixations however are allowed for now because when it is a full 3D application the motor will be internally fixated. Furthermore these fixations are necessary because the model only handles translations. The fixation on the Hoberman sphere does introduces extra friction to the system.

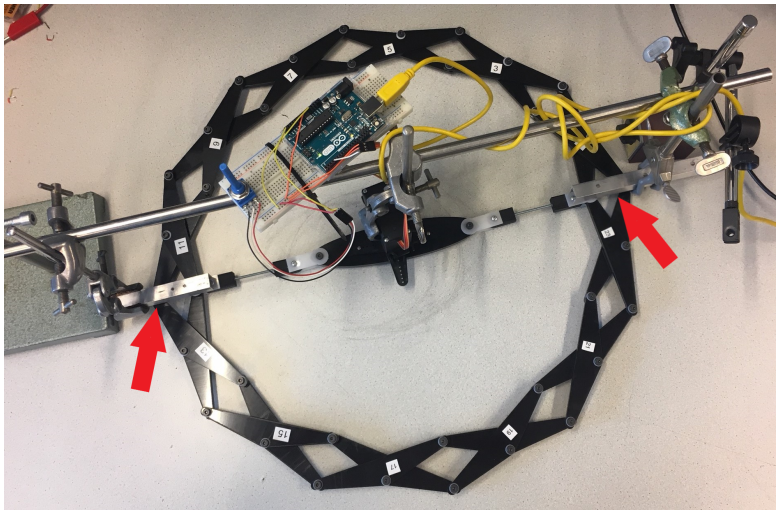

Figure 7.1: Measurement set-up

In the figure there is also an Arduino that is connected to the motor and the laptop. This Arduino set-up is put on top of the test set-up such that it doesn't cause a collision. The Arduino is used to move/control the Hoberman sphere via a potentiometer that rotates the motor within a range of $0 - \pi$rad. The connection diagram is shown in Figure 7.2, where the potentiometer is an input for the Arduino (pin A5) and gives a signal to the motor (pin D3). The voltage is made variable using the potentiometer and with that variable voltage the angle can be set on D3. With this connection diagram the potentiometer controls the Hoberman sphere and the first impressions could be done about the Hoberman sphere. With these impressions the limitations of the Hoberman sphere are found out and the supporting structure where added.

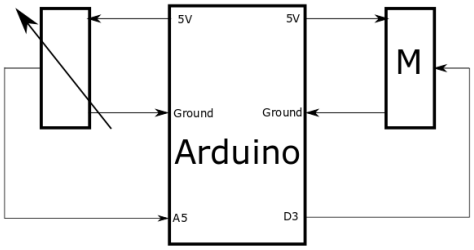

Figure 7.2: Arduino test set-up

7.2.2 Angle comparison test

Measurement plan

In this test the relation between the angle θ and the angle of the motor (θ_m) is going to be investigated. The reference signal that is used in the control set-up uses the variable θ. However the motor in the control set-up uses a different angle variable. The motor can change from $0 - \pi$rad while the Hoberman sphere moves from 0.42-0.98rad. To correctly control a motor in the various set-ups used in this research, the relation needs to be found between the angle of the motor, θ_m and the angle of the Hoberman sphere θ. This conversion will then be added to the model. With the servo motor used in this set-up the angle of the motor is already known due to the encoder of the motor.

From the Arduino script, the encoder signal is read out during the measurement. The measurement is performed for the forward and backward motion. This test is performed for both motions because it is expected

that there is a difference. This expected difference is mainly due to play in the system, but could also be from other sources.

Results

The total measurement is shown in Table 7.2. In the table it is shown that there is a difference between the forward and the backward motion of the Hoberman sphere, as expected. When performing the measurement it was noticed that the difference between the forward and backward motion was mainly due to play in the system.

Table 7.2: Angle measurement

	Forward		Backward	
Measurement (-)	Angle θ (rad)	Angle motor (rad)	Angle θ (rad)	Angle motor (rad)
1	0.42	0	0.42	0
2	0.43	0.12	0.45	0.09
3	0.47	0.24	0.49	0.19
4	0.52	0.37	0.54	0.30
5	0.58	0.52	0.58	0.40
6	0.64	0.63	0.63	0.51
7	0.69	0.75	0.70	0.63
8	0.77	0.91	0.75	0.73
9	0.83	1.03	0.80	0.82
10	0.86	1.12	0.85	0.93
12	0.90	1.22	0.88	1.03
13	0.94	1.36	0.93	1.19
14	0.97	1.50	0.95	1.31
15	0.97	1.62	0.97	1.41
16	0.97	1.75	0.97	1.64
17	0.95	1.92	0.97	1.78

From the measurements a graph is set up in Figure 7.3 to see the differences more clearly. As shown in the figure the backward motion looks like a delayed version of the forward motion. The measurements above approximate $\pi/2$ are not taken into account due to symmetry.

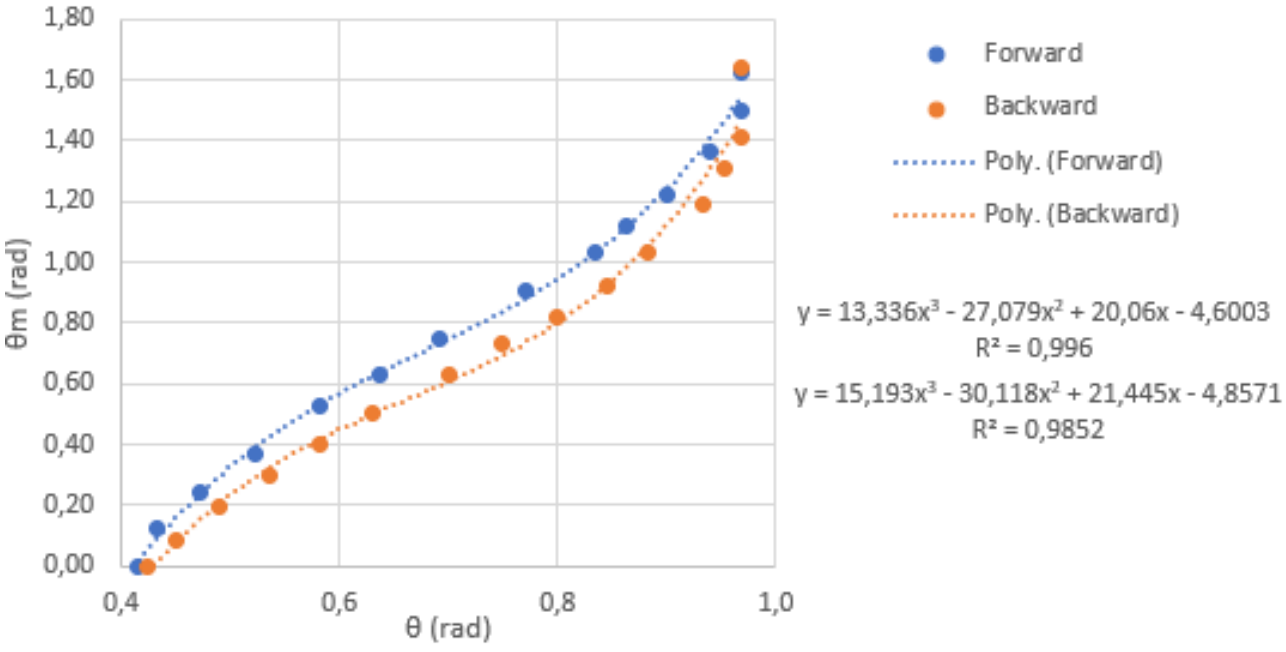

Figure 7.3: Angle comparison between forward and backward motion

From the individual datasets, two trend lines are created as shown in the figure. The forward motion trend line will be chosen as the conversion block to calculate the motor position. This conversion is necessary because

the DC-motor does not contain a rotary encoder at the motor. The main reason to choose the forward path is because it is more accurate and that most of the time the Hoberman sphere will go to the setpoint and will expand to that position and not retract.

The difference between the two datasets is calculated using the trend lines. The largest difference between the two is at $\theta \approx 0.75rad$ with a difference of $\theta_m = 0.14rad$.

7.2.3 Discussion

The servo motor test showed what the relation is between θ and θ_m. Furthermore some limitations of the Hoberman sphere and the set-up were found out. With this knowledge the other tests can be performed and the simulation can be adjusted with the relation between the angles θ&θ_m.

However the communication between Matlab and Arduino was also checked in this test, and it is quite limited. The idea is to use the Arduino to handle all the external signals (sensors) and feed this to Matlab. In Matlab the analysis is done and from there Matlab gives the signal as an input to the Arduino that gives it to the motor. The main problem however, is that Matlab is not good dealing with real time, while the Arduino is. Getting the input from Arduino to Matlab and having Matlab converting the correct time for each measurement is problematic. To solve the timing issues multiple ideas where tried out without success:

- Using the timing from Matlab self (tic & tac commands and other tools).
- Using Simulink with the Arduino support blocks.
- Using the serial port via Matlab serial command, without directly connecting to Arduino.

With no clear solution for this problem the other tests will only be done in Arduino, so without the connection to Matlab. After the test, the Arduino data is used and analysed in Matlab.

7.3 Open loop impulse response

A test that is performed on the open loop of the model is an impulse response test to find the crossover frequency. This impulse response is done on the simulated open loop, where a delta function is set onto the plant. The output data from this impulse response is transformed with a Fourier transform. This Fourier transformed signal is used in a frequency response. From the frequency response, the crossover frequency can be found. With the crossover frequency known the PID controller can be finalized [23]. In this test there is no reference signal, controller or motor used.

7.3.1 Measurement plan

The open loop Simulink model is shown in Figure 7.4, where A_1 is the motor constant and the loop contains the A_2 setting where the efficiency is stored. This part of the actuator is in the loop because it is dependent on the angle θ. A saturation is added to the loop that ranges from 1-0.01. This saturation is added to make sure that there is no multiplication with zero. From section 6.2 the actuator block is now changed into two different parts because of the implementations of the efficiency of the motor set-up. The input for the open loop model is an impulse response or a delta function. It has an infinitely high peak at an infinitely small time interval.

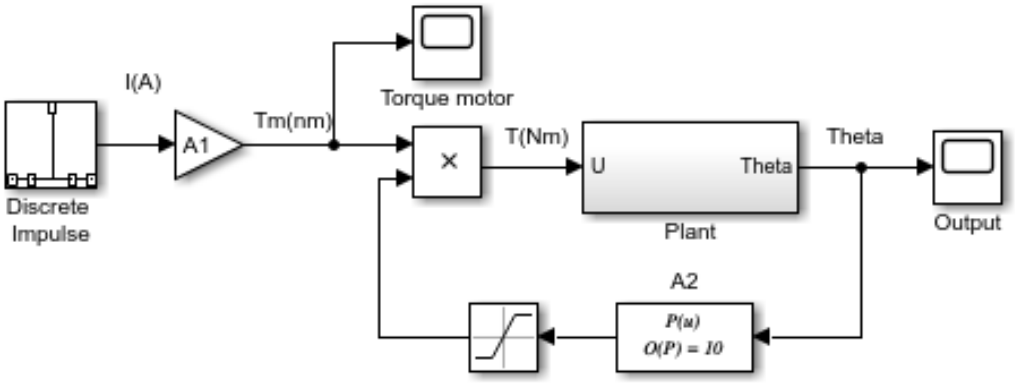

Figure 7.4: Open loop Simulink model

7.3.2 Results

The frequency response is shown in Figure 7.5. As shown in the figure the crossover frequency can be found where the amplitude is equal to 1 (10^1), so a 82 rad/s.

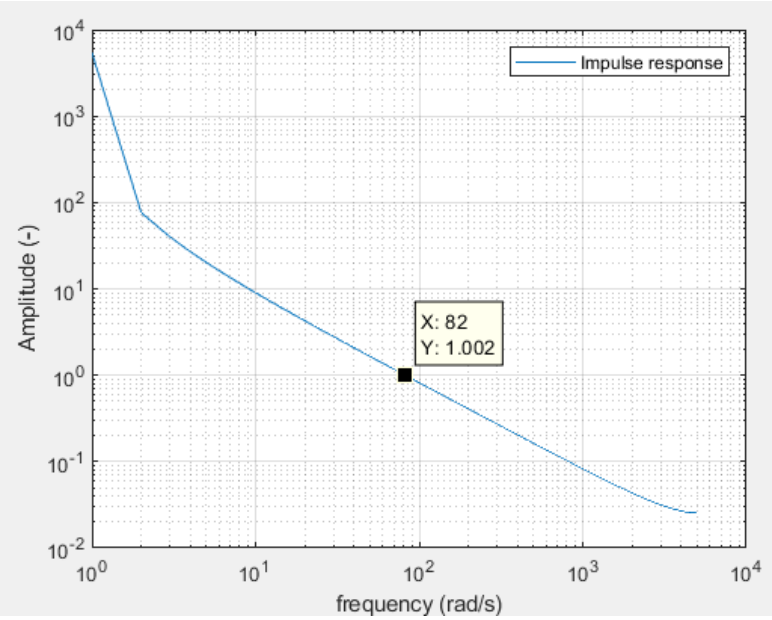

Figure 7.5: Open loop frequency response

With the crossover frequency known, the PID-controller can be fully designed. The values for the PID controller are given in Equation 7.2. These PID controller settings will be used in the prototype and the model to check the correctness of the model. This will be tested in the next sections.

$$K_p = \frac{3m\omega_c^2}{2\sqrt{\beta}} = 40.602 \qquad K_i = \frac{m\omega_c^3}{\beta} = 350.946 \qquad K_d = m\omega_c = 1.0439 \tag{7.2}$$

7.3.3 Discussion

The frequency response of the open loop system does not give the expected behaviour, because in the model there is only a mass present. With only a mass in the model the frequency response should be a linear signal. In section 8.1 a more detailed discussion is given about this subject.

7.4 DC-motor

The DC-motor tests can now be performed because the controller is known. In these test the open loop of the prototype is tested further. In the open loop test on the prototype the motor is attached to the motor set-up. The other tests with the DC-motor are performed closed loop to check the response of the system.

7.4.1 Measurement set-up

For the DC-motor a different measurement set-up is used compared to the servo motor set-up. For both the open and closed loop the same set-up is used. In this set-up the Arduino is connected to the rotational sensor, described in section 5.4 and to an external power supply. This external power supply is necessary because the DC-motor works on $12VDC$ and the Arduino can only handle 5V. To make the DC-motor run on the Arduino, a Adafruit motor shield V2.3 is added on top of it. Some of the motor requirements are set due to the use of the Arduino with the Adafruit motor shield. This motor shield is capable of running multiple different motors at the same time. However the motor shield only allows up to 13.5VDC of voltage supply and 1.2A of current for the motors. Furthermore the H-bridge on the motor shield does not have a current measurement. Therefore an oscilloscope is added to the set-up, to measure the current through and voltage over the motor. The current and voltage are necessary to find the torque that is applied to the motor. With the torque known, a comparison can be made between the prototype and the model. From the comparison the model can be checked and verified.

The connection diagram of the measurement set-up is shown in Figure 7.6. In the connection diagram it is shown that the sensor is connected to D3, a PWM pin on the Arduino. This is necessary for the Arduino to read the sensor correctly. In the figure the oscilloscope is not visible as a block but as different channels (C) and the external trigger. With the different channels the voltage (C1 & C2) over and current (C3 & C4) through the motor can be measured. A resistor of 1Ω is added to the electrical circuit to measure the current through the motor. With the resistance being 1Ω, the voltage should be equal to the current. The external trigger can be used to synchronise the Arduino with the oscilloscope measurement.

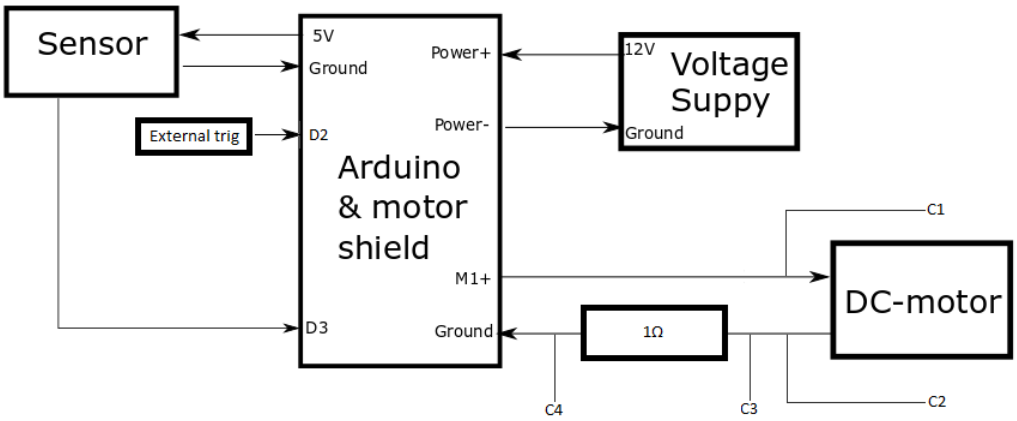

Figure 7.6: Arduino DC-motor test set-up

In the code that is used by Arduino, the motor is controlled by an PID controller, see Appendix E. The PWM signal of the sensor is converted to an angle. The PID controller in the code uses a reference and controls the motor to that reference. Both the controller and the angle conversion are part of a library. However as shown in the appendix, both library needed some changes to make it compatible with the DC-motor. In the Arduino code the reference is set to from 0.42rad as a starting angle and 0.91rad as the end angle.

The oscilloscope has four channels and one mathematical option where mathematical operations can be performed on the different channels. For that reason the mathematical channel is used to calculate the difference between C3 & C4 to get the current through the motor. The other channels only display the duty cycle of the motor and thus only differ from $\pm 0 - 12V$ depending on the direction of the motor. The motor can be set to forward, backward and release, where it moves in the forward or backward motion. However the release options is there to disable the motor. The external trigger is used to get the timing of the Arduino code the same as the oscilloscope to better compare the results. In the code D2 is triggered every loop and therefore give a trigger for the oscilloscope.

For the oscilloscope it is important that the ground of every channel probe is properly grounded. Without correct grounding the error when measuring is about 200mV, about the same range as the to be measured current. This error is measured by turning on the power supply but not enabling the motor in the software. When turning on the power supply, the oscilloscope already measures a voltage. This is the error voltage that can be reduced by grounding the system correctly. With the correct grounding the measuring error is reduced to about ±25mV. The data acquisition inside the scope is set to high resolution, to get a less noisy signal. This data acquisition is a setting in the oscilloscope and is given for reproducibility.

A downside of the oscilloscope measurement is that the data cannot be continuously read from the scope. The data that is available is only what is on the screen of the oscilloscope itself. To measure the entire test the time scale of the oscilloscope must be set higher dependent on the test itself. This however will also influence the data per second.

7.4.2 Open loop DC-motor

Measurement plan

The first test with the DC-motor is to check the open loop response. The reason for this test is to find the differences between the prototype and the simulated model.

The input on the prototype will be a step function that works for 1s. This will be when the motor is put

into forward/backward motion in the software and is released after 1s. The duration is added to the test because the Hoberman sphere is only allowed to move πrad, otherwise it will collide with the motor set-up. In this test the voltage and the current will be measured by the oscilloscope. The Arduino is only used to measure the angle of the motor and release the motor itself after 1s. The test will be set in three different phases namely start-up, measurement and die-out. In the start-up phases of 1s, there could be some initial effects on the system. With a 1s delay, the initial effect should be gone. In the measurement phase the system gets the 1s step function. The die-out phase starts when the system comes to rest again. At each step of the test the current and angle are measured.

After the open loop test, the current data is set as an input in the Simulink model. This is the same model as in the open loop test, see Figure 7.4. The only difference is that the impulse response block is changed to a from workspace block, where the current data is set. With this the simulated model gets the same input as the prototype to better compare the two. With this input added to the model and both of the angles measured/simulated, the differences between the prototype and the model can be shown.

From the relations described in section 6.5 about the actuator forces, it is known that the Hoberman sphere needs higher torques at certain angles. With that information, the current applied to the motor should be different in certain situations. For that reason this test consist of four different configurations that are checked. The configuration are two in the extension phase with close (0.96rad) and optimal configuration (0.72rad). The other configuration are in the retraction phase with an open (0.42rad) and optimal configuration again. The optimal configuration is defined where the efficiency from the motor set-up is the highest. The two optimal configurations are in between the open and closed states.

Results open loop test on prototype

First the open loop test on the prototype is checked. The result of this test is shown in Figure 7.7, where the individual configurations are displayed with respect to time. In the figure it is clear that the first second was used to get rid of start up noises. After the start-up phase the measurement of the individual configurations are shown, where every configuration has the same input. The closed configuration (blue) has only a range of 0.12rad, whereas the optimal expanding (orange) configuration has a range of 0.85. The optimal expanding configuration travels through the fully open state and goes to the closed state again. The reason that the optimal expanding configuration has a larger range is because of the efficiency of the motor set-up discussed in subsection 4.3.3. The closed configuration starts at a point where the efficiency of the motor set-up is lower than the optimal configuration. The optimal configuration start at the optimal angle, however the efficiency decreases from that point on. It most likely got through the point where there efficiency is about zero due to the momentum it has.

The behaviour changes however in the retraction phase, where the open (grey) configuration has also a large range, about 0.57rad. It was expected that the open configuration, where the efficiency is low, should have a small range. The same goes for the optimal retracting (yellow) configuration where the range was expected to be larger.

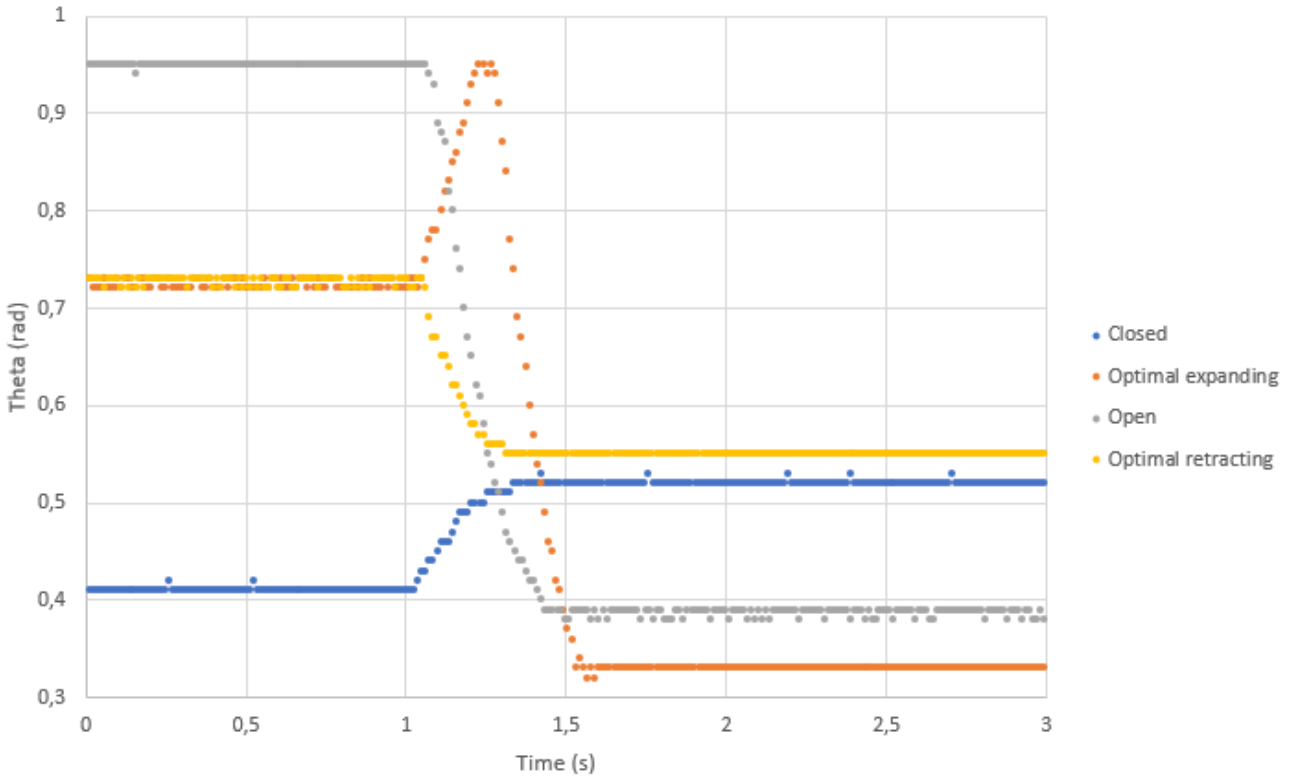

Figure 7.7: Open loop prototype angle measurement

In Table 7.3 the results are shown with some extra parameters. The signal applied to the motor is a step function of 1s , however it looks like the step function only last 0.5s. This is shown in column "Time for range (s)" of the table. From the time and the range a velocity can be calculated. As expected is that the configuration with the highest velocity has the highest range. The strange behaviour with the step function is further explained in the next part.

Table 7.3: Prototype angle comparison

Configuration	Range (rad)	Time for range (s)	Velocity (rad/s)
Closed	0.12	0.37	0.32
Optimal exp	0.85	0.54	1.57
Open	0.57	0.47	1.21
Optimal ret	0.18	0.31	0.58

Results open loop comparison

In the last open loop test the current data from the oscilloscope is set onto the input of the Simulink model. The input and output data is shown in Figure 7.8. This data is from the optimal expanding state but this behaviour is observed at every state. In the input data it is shown that the current is still above 0.4A for the full time of which the step function is active. The first 0.5s of the measurement the motor does move and thus a fluctuating current. After that 0.5s the current becomes constant at about 0.4A but the motor does not move anymore. In the Arduino the state of the motor is also checked and the motor is not switched off during this time.

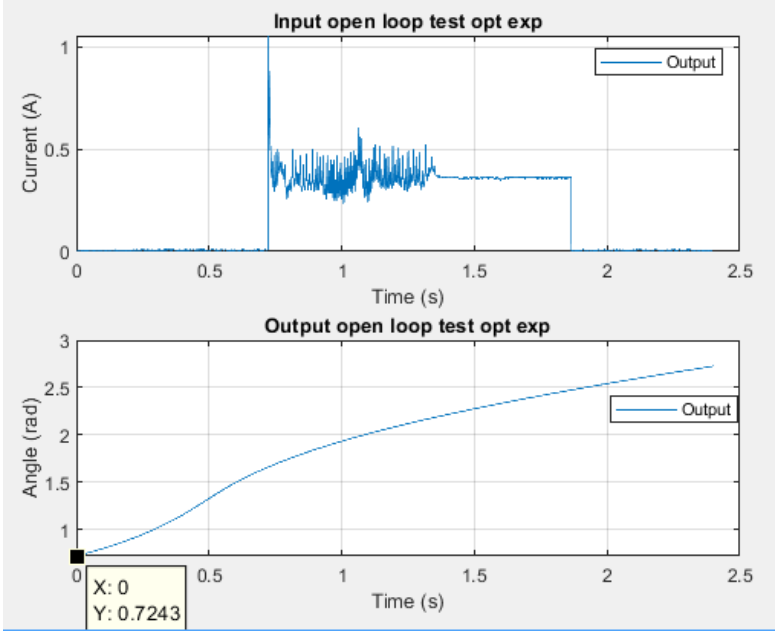

Figure 7.8: Open loop prototype input on Simulink model

The output data in Figure 7.8 starts at the optimal angle and from there rises almost independent of the input. The input has until about 0.7s almost no input (0.01A), but still according to the model the Hoberman sphere can move. The maximum and last value of the output is 2.73rad, which would imply a range of 2.00rad. Furthermore after there is almost no current on the system, the system still rises. This implies that there is no friction and/or damping in the simulated system, which is the case.

Some of the results of the test are displayed in Table 7.4. The closed state of the simulated model measurement has the a range of 6.46rad. The used model in the simulation is a point based model and has therefore no constraints what the maximum range could be. Due to the model being point based and that there is no friction and/or damping in the system, the angle θ can become very large. The differences between the different configurations of the model are because of the motor set-up. With the closed state configuration, the starting angle is also the beginning of the motor efficiency range. With the closed state configuration the entire range of the motor set-up is used whereas other configuration only uses a part of that set-up. After the range of the motor set-up the system gets a value that is about zero.

Table 7.4: Range comparison prototype and model

State	Range prototype (rad)	Range model (rad)	Model/prototype (-)
Closed	0.12	6.46	55.76
Optimal exp	0.84	2.00	2.37
Open	0.57	1.65	2.92
Optimal ret	0.18	1.94	10.57

7.4.3 DC-motor response test

Measurement plan

In this test the closed loop response is checked of the system with a DC-motor attached. This is done by setting the system in the begin position (homing) and from there let the PID controller, control the system to the reference position. While this is done the current will be measured. The current is measured to calculate the torque that the motor applies to the system using the relation between torque, motor constant and current. This torque is necessary to evaluate the model and the prototype with each other. The test is performed with three different measurement times, namely 5 &2,5 &1s. This is added to see if there is a difference in response due to the time component.

The reference that is applied to the motor is changed a bit to add homing to the system. The first second of the reference profile, the system will have time to go to the begin angle (homing). From that starting point the third order polynomial will be applied to the system. This third order polynomial has a range from 0.42-0.91rad. After the third order reference profile is applied another phase is added to it for the system to die out.

This results in a reference profile with a start-up, the measurement and a die-out phase.

To implement the third order motion profile on the Arduino, numerical integration is used. From a third order motion profile it is known that the jerk is ±constant. From that fact the angle can be found by integrating the jerk three times. This is done by using the trapezoidal numerical integration shown in Equation 7.3, where a & b are time instances and f(a) is the value of function f at time a.

$$f(x) = (b-a) \cdot \frac{f(a)+f(b)}{2} \tag{7.3}$$

To get the desired third order motion profile, Equation 6.3 is used to find the maximum jerk. When the jerk is implemented as Figure 7.9, the desired reference profile is applied. The figure shows the jerk profile for the 5s measurement time. With this jerk profile an one second homing and 4s die-out phase is added. The values of the jerk are not the actual values. These are set to one to make the figure more clear. After each integration, an integration constant is added. However the begin velocity and acceleration are zero. The angle has an integration constant, namely the starting angle.

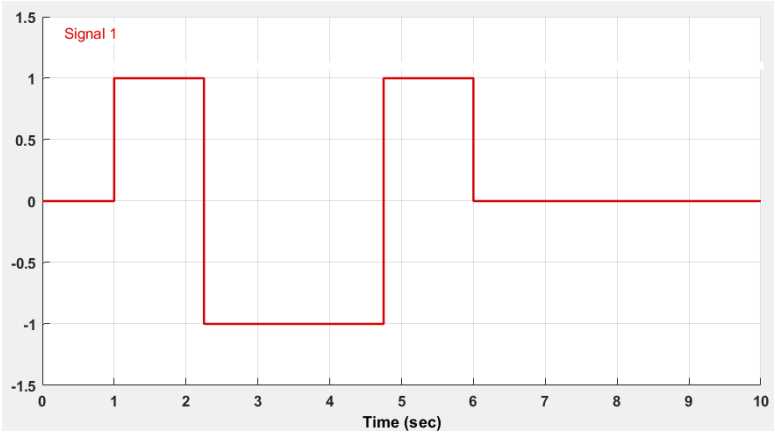

Figure 7.9: Jerk profile of the reference

Response results

The responses of the closed loop with a DC-motor is shown in Figure 7.10, where every measurement time has its own subfigure. In each subfigure the reference profile is shown and the response of the prototype to that profile. The reference profile is divided into three phases, namely start-up, measurement and die-out.

The PID settings from the simulation are not good enough for the prototype itself. The response of the prototype is slow, has overshoot and oscillates after the final setpoint is reached. This behaviour occurs at every measurement. The oscillations are caused by integral wind-up. The integral action is too high for the prototype and therefore the system overshoots. This occurs because the motor switches direction and a high opposing signal is than applied to the system. This is better shown in Figure 7.11, where the control input is shown. Another problem that strengthens the integral wind-up is that there is some spacing when switching between direction, see the hysteresis effect at Figure 7.3. In the beginning, at about an angle of 0.45rad, the movement stops and goes again. This could be due to the stick slip effect or the difference between the forward and backward motion, as discussed in subsection 7.2.1.

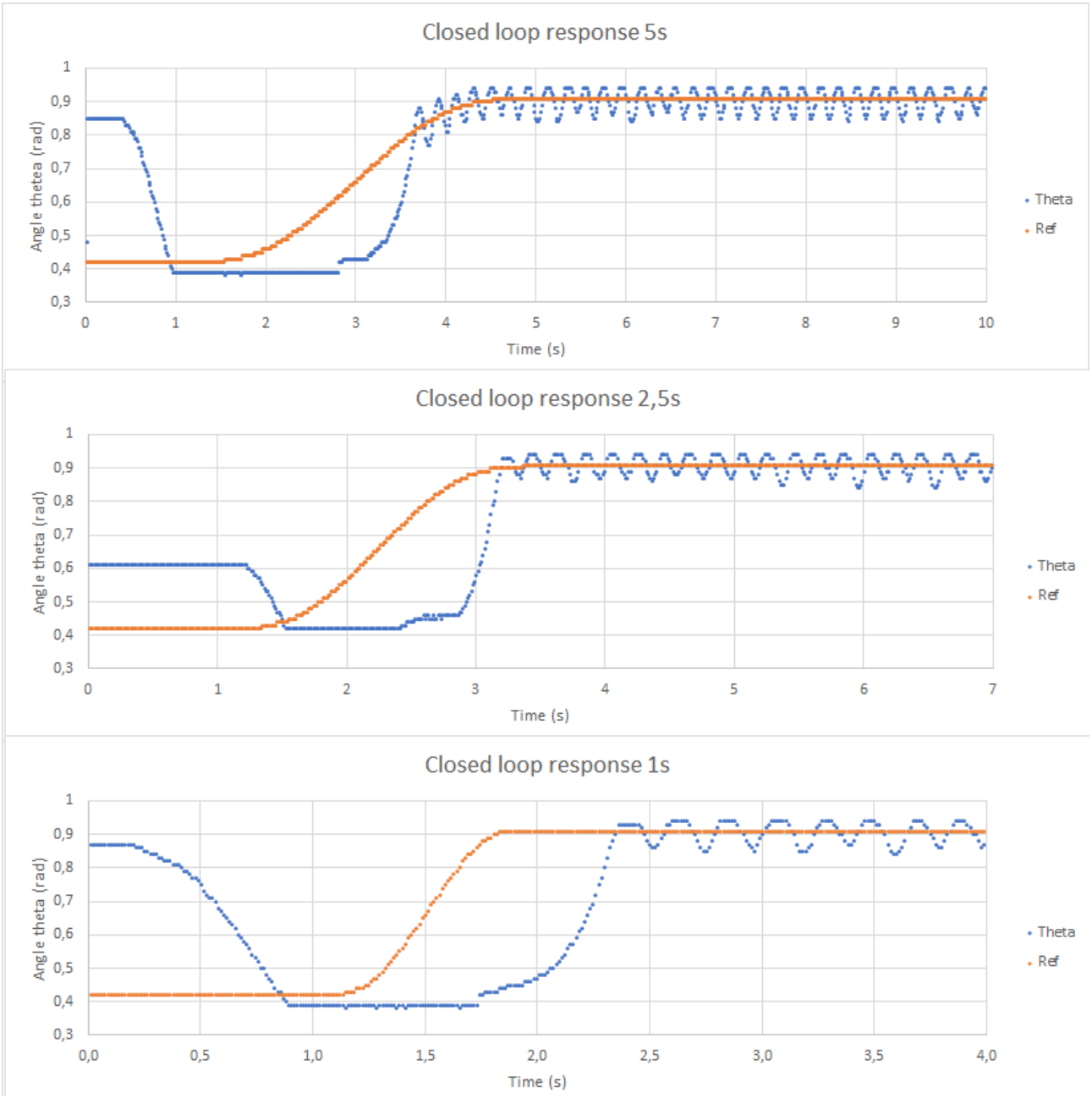

Figure 7.10: Closed loop response

The 5s and 1s measurement responses look like they respond well to the homing signal, within 0.2s. The 2,5s measurement respond after the homing phase is completed, at about 1.2s, to the same reference profile. This is due to the control power not being high enough for the 2,5 measurement at the homing phase. The begin angle of the 2,5s signal is closer to the starting angle resulting in a lower difference between reference and error. This effect is again better shown in Figure 7.11.
A better PID controller for the prototype would be one with a higher K_p, lower K_i and higher K_d. The higher K_p will result in a better following behaviour because the gain is in the beginning of the reference profile too low. A lower K_i will result in less overshoot and the higher K_d will result in a faster response. For the prototype this PID controller still results in a stable system. It is stable because the oscillation do not increase, however this is not a desired behaviour. The wanted behaviour is a system that reaches its end value without oscillations as fast as possible. From this information it can be concluded that the model, on which the PID controller is based, is not good enough.

The output of the PID controller is shown in Figure 7.11. The PID output has a value between $0 - 255$, meaning 255 is maximum power. For the Hoberman sphere to move, from the previous figure and this figure combined, it looks like the PID action needs to be above about 70. This is one of the reasons why this PID controller gives problems for the prototype. For the 1s and 5s measurement the controller output reaches the value 70 well

within the homing phase, whereas the 2,5s only reaches that value at about 1s. In the part where the control output fluctuates between a positive and negative number, the integral wind-up is clearly visible.

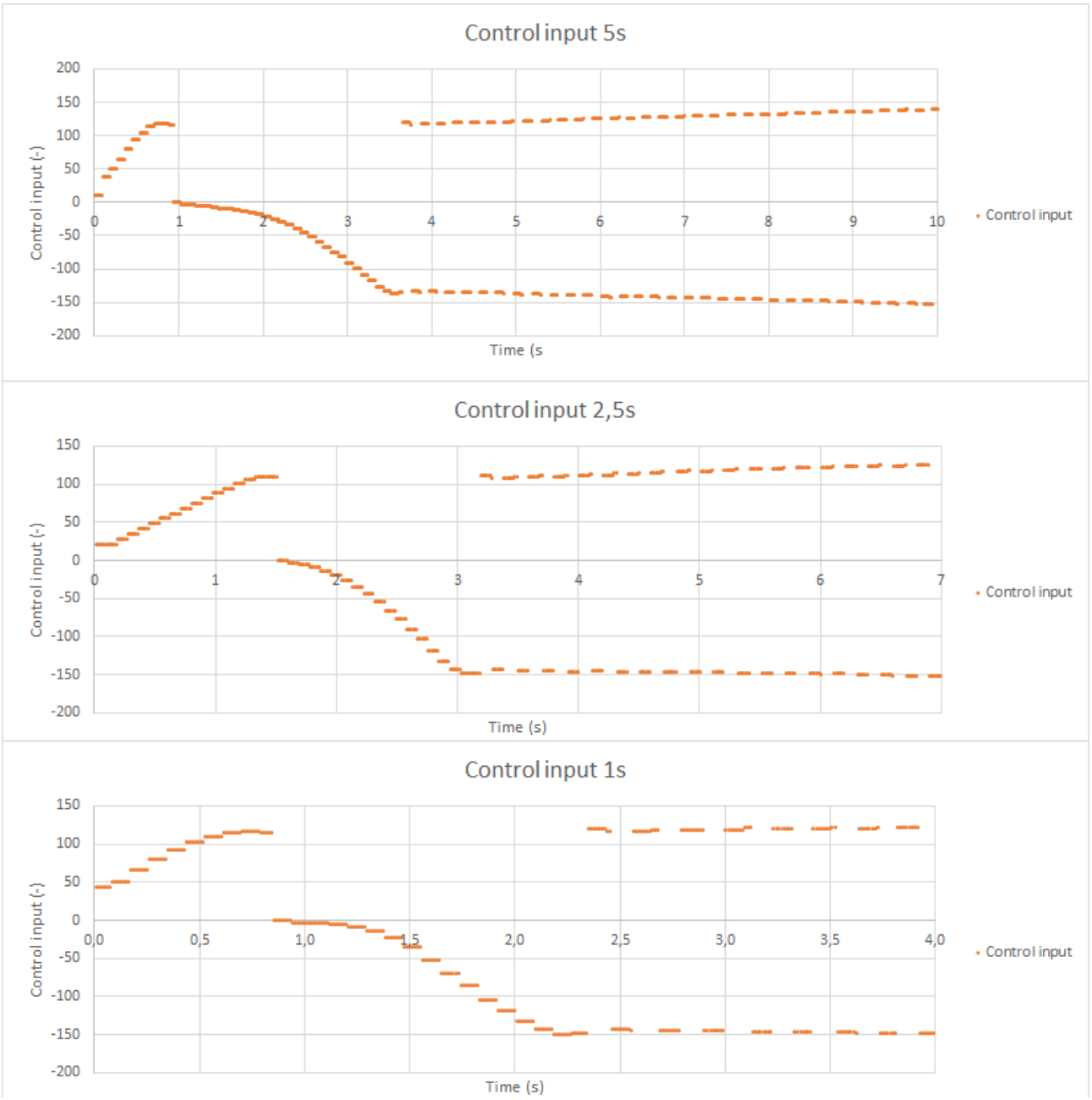

Figure 7.11: Closed loop control input

In Figure 7.12 the error of the responses are shown. From all the responses the final error is about the same, with the same oscillatory behaviour. The differences between the measurements is mainly that the error is slightly higher at the reference profile part of the response because the PID controller is not fast enough. This means that the error will increase for faster systems.

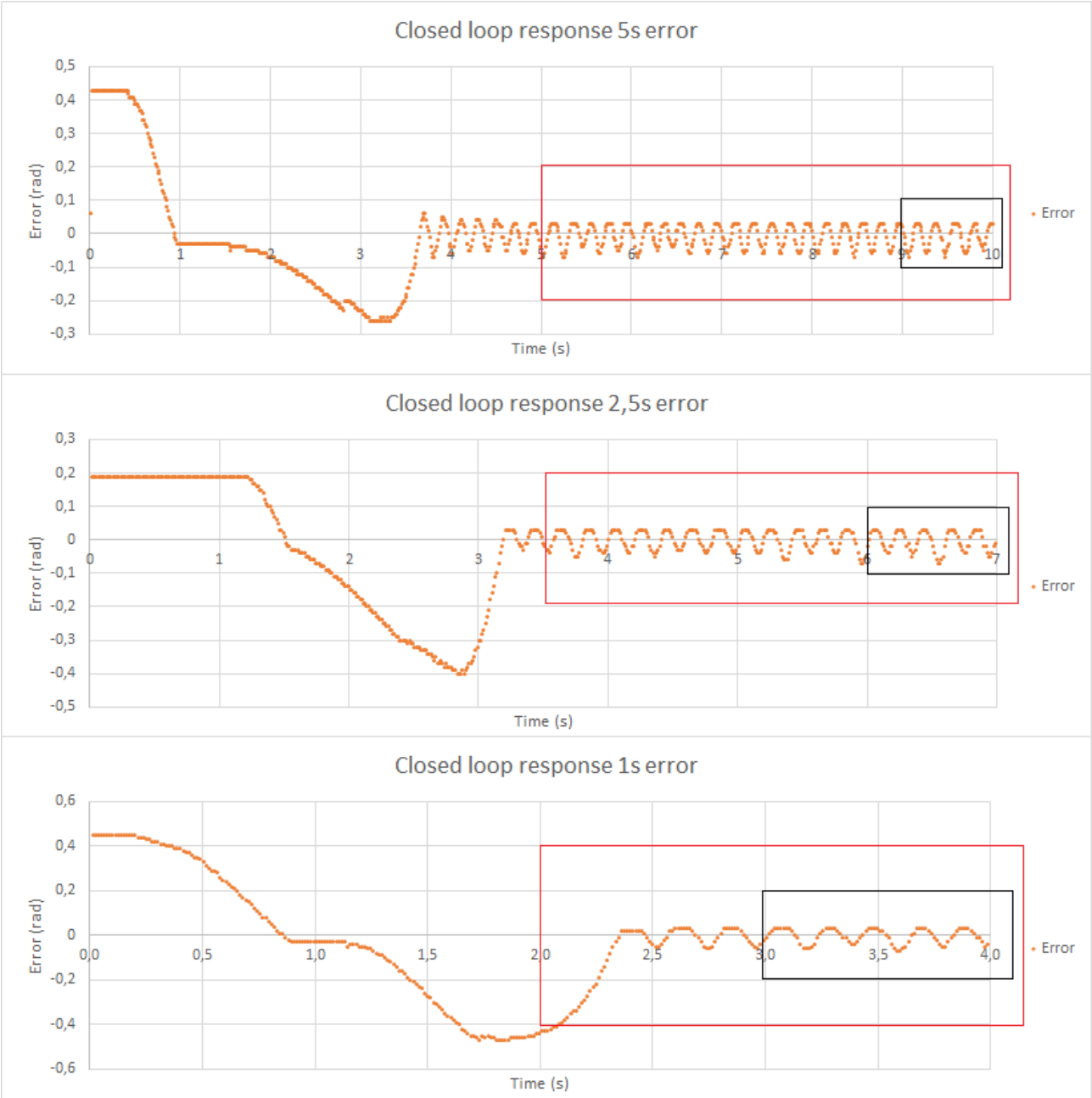

Figure 7.12: Closed loop error

The values of the error are given in Table 7.5. In the table the differences between the measurement is more clearly shown. The ± maximum errors are divided into three different time periods to better see the differences. The first time period is the overall measurement, so the entire graph. The second measurement is where the reference reaches its final value to the end, the red box. The last is one second before the end to the end, the black box. As shown in the table, the response with a reference profile of 1s has the highest error (-0.47rad). The final error is for all the responses the same, with an oscillatory behaviour with ± maximum error of 0.03 to -0.07 rad.

Table 7.5: Closed loop response, error comparison

Measured from		Error 5s (rad)	Error 2.5s (rad)	Error 1s (rad)
Overall	min	-0.26	-0.40	-0.47
	max	0.43	0.19	0.45
Ref starts to end	min	-0.07	-0.07	-0.47
Red box	max	0.03	0.03	0.03
1s before to end	min	-0.07	-0.07	-0.07
Black box	max	0.03	0.03	0.03

To find the current and the torque values of the response, the oscilloscope data is been used. With the measurement done over the different phases, the voltage is measured over channel 3 &4. This result is shown in Figure 7.13. The voltage over the 1Ω resistor has a relation that looks quite similar to the error profile, only flipped. This is again logical because when the error increases the PID controller output, more torque is demanded and thus a higher current. The current positive and negative side is dependent on the direction of the motor (forward/backward). These directions can be set by the Arduino code and that is why the current data is flipped compared to the error signal.

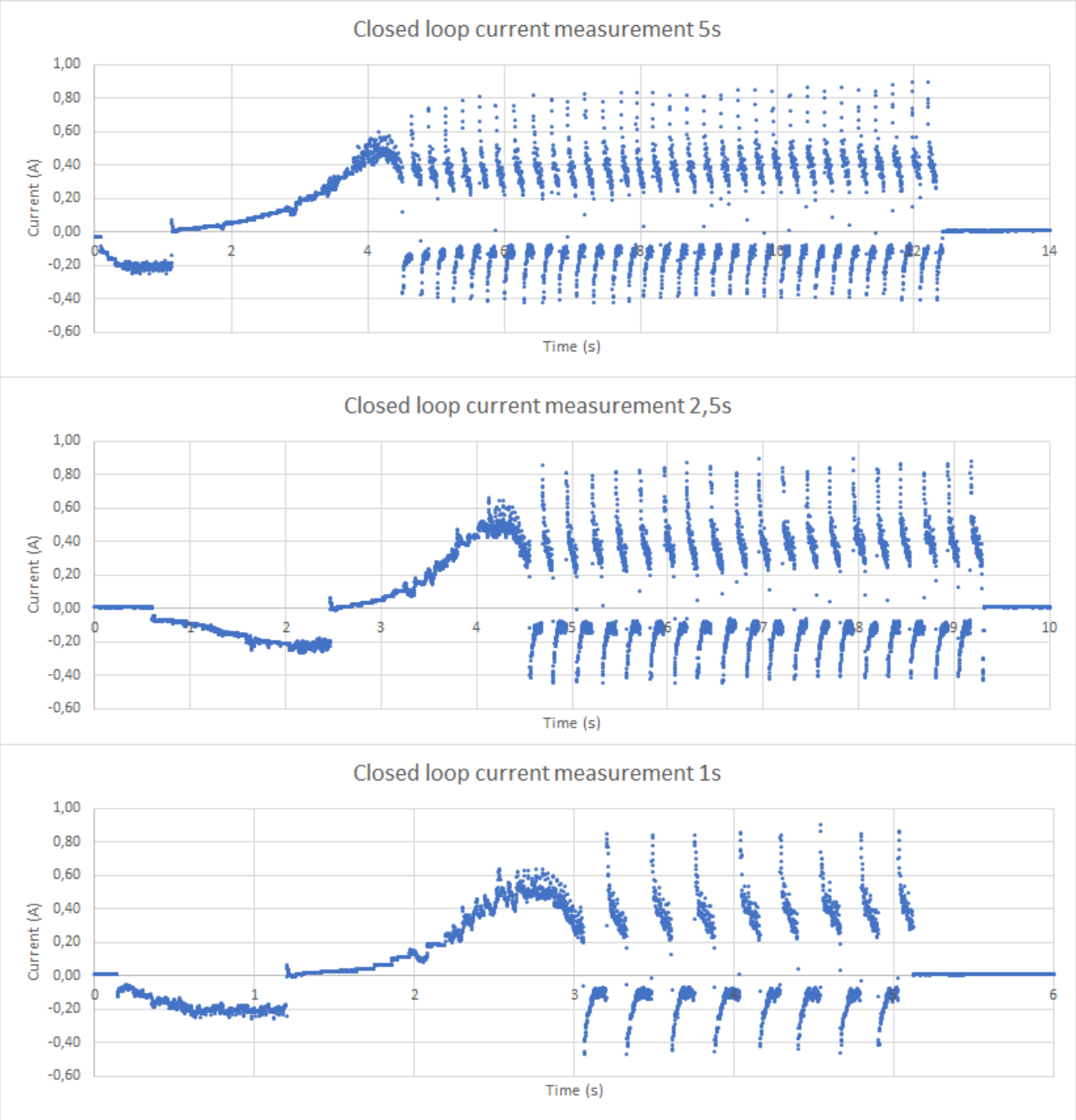

Figure 7.13: Closed loop current measurement

From the figure it is shown that the current exceeds the maximum power that was available according to the specification. The specification [21] states that the maximum power is 3.90W, which results in a current of 0.33A. The peak value here for the current is about 0.90A independent of the different measurements. The place where the peak current occurs is at the beginning of the oscillations. Every time the motor switches between direction a peak is shown in the plot. This peak has to do with the motor standing still and trying to start up. Without the oscillations the maximum current would be about 0.6A. This is however still about twice the amount, the motor should be able to handle.

The torque is automatically then also known because the motor constant is already described and was set to 2. Therefore the torque plot will be twice as high as the current plot. Again this goes against the specifications of the motor because the torque will be higher then 0.5Nm. With the maximum current know, the maximum torque will be about 1.8Nm.

7.4.4 Discussion

The main problem with the DC-motor testing was to get a correct measurement of the current or torque. The current measurement is now done with an oscilloscope, but other methods were also tried. This current measurement is necessary because the Adafruit motor shield does not have a current sensing pin on it. Furthermore the voltage output from the motor shield is not given. This voltage that is sendout by the motor cannot be directly added to the Arduino itself because the Arduino can only handle voltages between $0 - 5V$ and the motor sends out voltages from $\pm 0 - 12V$. The $\pm$ is added because, depending on the direction of the motor it is either positive or negative. These voltages are not allowed onto the analogue pin of the Arduino.

Another problem is that the reference cannot be set to the maximum position because the current Arduino script cannot handle that position. The script will keep increasing the PID action which results in an overshoot to the maximum position, see Appendix E for the used script. In the script the forward and backward motion are dependent on the angle θ. If θ is smaller than the reference the script gives the command forward. When θ is larger than the reference the script gives the command backward. The problem is that at the maximum angle both directions have a smaller angle. This results in that the PID-controller will push the system trough the maximum angle to the minimum angle on the other side and it will still keep increasing the controller output.

A couple of reasons that the current is higher then expected are that:

- A measuring error because the system has still a grounding error of about $\pm 25mV$ at every channel.
- The maximum power allowed is not correctly given in the datasheet and from that a wrong motor constant is calculated.
- A correction for the losses due to Back-EMF is not done. If this is done the resulting current will also be lower.

7.5 Prototype vs model

7.5.1 Comparing prototype vs simulated model

The controller shown in section 6.2 is changed here to add the motor set-up correctly. This is shown in Figure 7.14. This new set-up is added to compare the closed loop responses of the prototype and the simulated model. In this set-up the T_m is the torque from the motor and T_p is hte torque that is delivered to the plant. The angle at which the simulated model starts is dependent on the previous test prototype starting angle. The starting angle of the prototype will be implemented into the integrator of the plant.

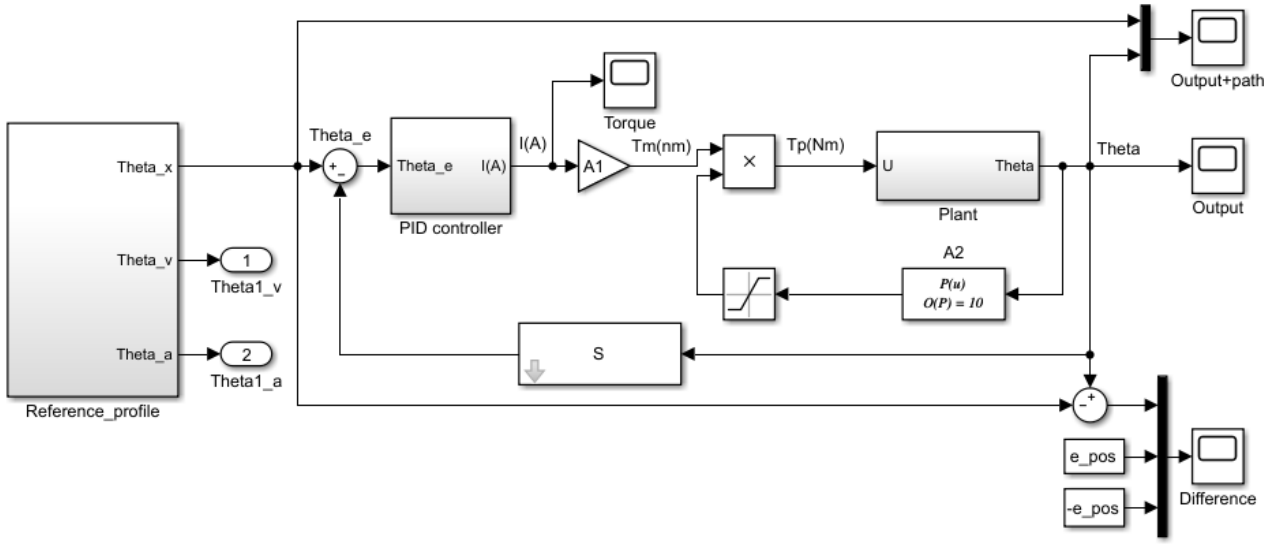

Figure 7.14: Simulink model dc motor

The test is divided into two parts. In the first part the response of both models are compared and in the second test the torques are compared. The torque that is necessary to compare the two models is after block A1, T_p. With these two comparisons the correctness of the model can be checked.

7.5.2 Results

Here the prototype is compared to the model created in Mathematica and Matlab. In Figure 7.15 the response of both the prototype and the model with a measurement time of 5s is shown to better compare them. The 5s measurement time is chosen because there is almost no difference with respect to the other measurements. As is shown in the figure, the starting angle is comparable. At both models the same PID settings and the same reference profile is used.

With the same settings there are clear differences between the model and the prototype. Where the model can reach the desired position, the prototype oscillate around it. This is again duo to the differences between the two. In the model there is no friction inside the structure and also no friction from outside sources. In the model the backward and forward motions are identical, whereas the prototype has some spacing. Furthermore the model has an ideal sensor, actuator and the actuator has no power limitations.

The model has some overshoot in the beginning. This is expected with the large K_i action. The model does follow the signal very accurate. The error in every measurement time is at the end about zero.

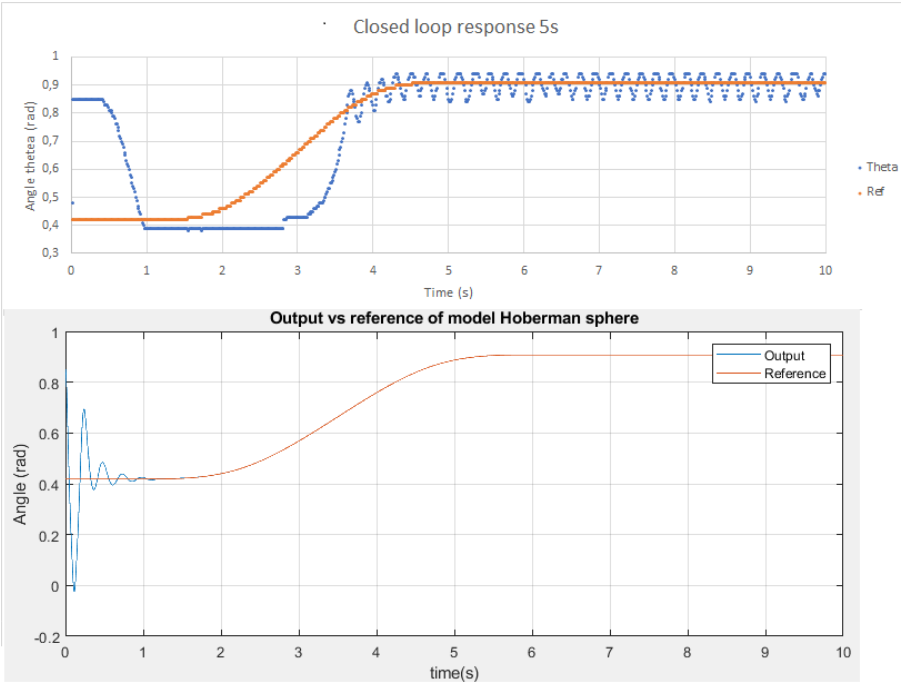

Figure 7.15: Model response comparison

In Figure 7.16 only the currents of 5s measurement is shown. This is done because the results from the current measurement are already discussed and there are not much differences between the measurements. As expected the current of the simulation is very large compared to the prototype. This is logical when the response is checked. The current is that high because the error is very large resulting in a high output from the controller which results in a very fast reduction of the error.

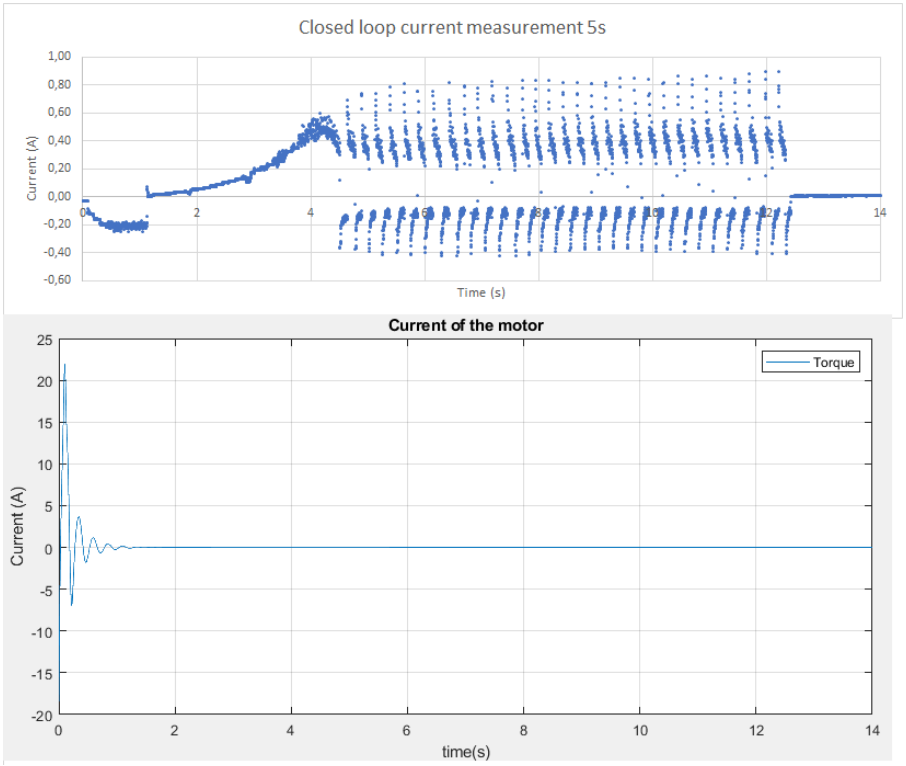

Figure 7.16: Model torque comparison

The error of both signals are given in Figure 7.17, where the error of both the simulation and the prototype are shown. The error is about zero after 1s for the simulations, where the prototype minimum error begins at about 3.5s. The RMS value of the prototype is 0.14rad and of the simulations is 0.05rad, which means that the prototype RMS value is 3 times higher. The closer the RMS value is to zero, the better the controller follows the given reference profile. From the comparison between the RMS values of the prototype and the model, it can be concluded that the model is not the same as the prototype. With an improved model the controller design changes. The new controller would better suit the prototype and with that a lower RMS value should be found. The lower RMS value should also be better compared to the model itself. When the behaviour of the response and the RMS values are comparable the model is good enough.

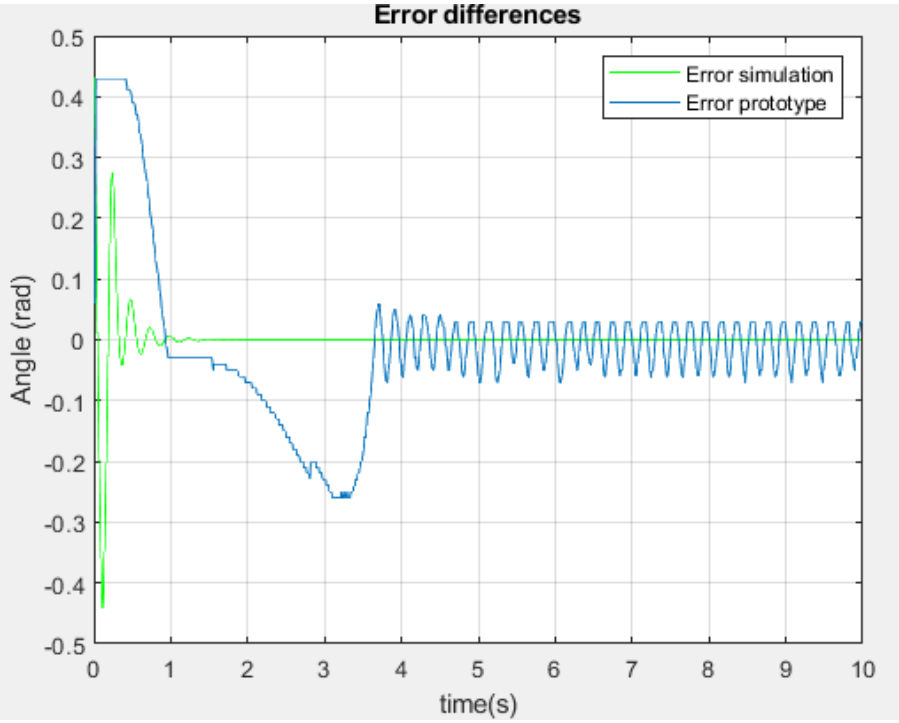

Figure 7.17: Model response error

7.6 Conclusion measurement & results

For this chapter multiple conclusions where found. With the servo motor test set-up the relation between the motor angle and the angle of the Hoberman sphere was found. Another conclusion is that the open loop frequency response has a cross-over frequency of 82 rad/s. With the cross-over frequency the PID controller was designed.

The closed loop response of the prototype was stable. However there was an oscillatory behaviour when trying to reach the end value. This oscillation did not increase over time. This behaviour was due to integrator wind-up. The measured current was found to be 2 times larger than the expected current.

The main conclusion from this chapter was that the found model in chapter 4 is not good enough. The RMS value where 3 times larger than the model. The PID controller is calculated using the model. When the model was good enough the controller should have been able to correctly control the prototype resulting in a much lower RMS value.

8. Conclusions & Recommendations

In this chapter the research question that is given in chapter 1 is given answer to. Some of the answers are already given in the various chapters, however here are all the conclusions given. In this chapter also the reflection and recommendation are given.

8.1 Reflection

A thing that I would have done differently now, is to not exclude the rotational polyhedral structures after the literature research. One of the reasons the translational structure, Hoberman sphere, was chosen is because of the expansion ratio. This expansion ratio that was checked during this project, was lower than the theory described. With this knowledge the rotational structures could also be a candidate for this research.

For the motor set-up a better idea would have been to use two plates instead of the current set-up. One of the plates is on top of the ellipse and the other on the bottom. The connections will still be the same. With this idea the range before collision will be larger. This can be implemented in the next iteration.

In the open loop frequency response I made a mistake with the analysis to get the cut-off frequency. The impulse response used in the analysis was not an unit impulse. With this mistake the cut-off frequency and the controller where determined. The analysis about the correctness of the model is still valid, however a different controller should have been designed and because of that the experiments should have been redone. The open loop frequency response is made again in Appendix F.

In the same appendix also the influence of back-EMF is discussed. The measurement shown in Figure 7.13 should have been compensated for the influence of the back-EMF. In the report it is briefly mentioned that it has a negative influence on the measurement. If back EMF would have been implemented a more accurate and lower current would have been measured.

8.2 Conclusion

From the literature review, the Hoberman sphere is chosen as the most fit polyhedral structure with respect to the requirements. In Table 8.1 the comparison is made between the Hoberman spheres that are described throughout this report. In the table the star(*) represents that it was not clear what method was used to measure the expansion ratio. From the table it is clearly shown that the theory does not represents the practise. This is mainly because in practice the motor set-up is added within the sphere itself, which compromises the expansion ratio. Furthermore the Hoberman spheres described in the theory have more scissor joints, which increases the expansion ratio.

Table 8.1: Hoberman sphere expansion ratio comparison

Hoberman sphere	$l_{5min}\ (m)$	$l_{5max}\ (m)$	Expansion ratio (-)
*Mini**	0.07	0.15	2.14
*Original**	0.12	0.38	3.17
*Museum**	0.75	3	4
Knex small	0.37	0.44	1.21
Knex big	0.48	0.61	1.27
Mathematica	0.1263	0.253	2
Solidworks without collision	0.136	0.252	1.83
Solidworks with collision	0.205	0.252	1.23
Lasercut iteration	0.204	0.251	1.23

The mathematical description of the Hoberman sphere is found, with an EoM as a result, see Equation 6.4.

The most suited motor set-up is found with the double slider crank. This mechanism is selected because it stays within the sphere itself while expanding/retracting and has the highest ratio between 2-3.

From the open loop test that is performed, the crossover frequency is found to be $82 rad/s$. With that a PID controller was designed which made the closed loop simulation is stable. When implementing this onto the prototype a stable but oscillating response was found. This means that the model is not good enough to be used in an interaction controller, and thus the model should be improved. The model can be improved by adding friction from the joints or changing from point mass to a rigid body mass. The RMS value from the error of the response of the prototype is 3 times larger than the simulation. This means that the prototype does not follow the given reference position as well as the simulation. The RMS value of the prototype needs to be comparable with the model when the model is implemented in the interaction controller.

The main conclusion from this research is that the found Hoberman sphere model should be expanded to better reflect the practise. The interaction controller cannot be implemented correctly with the current model. The current model does not give the correct velocity that is necessary for the interaction controller. This controller is according to the specifications best suited for the problem set. The design phase of this research is done successfully with a prototype as the end result.

8.3 Recommendations

In this section the recommendations are given for this thesis report.

- Change 2D Mathematica model to make a better estimation of the physical system.
  - Add friction from the joints and environment
  - Change the point mass to a rigid body perspective.
  - Add back-EMF to the actuator part of the model.
  - Add dependency of motor constant to angle θ because of small range.
- Change controller to an interaction controller when a good model is found
  - The current control strategy is to control the angle θ. To use the Hoberman sphere inside tunnels or other environments the controllable parameter needs to change from the angle θ to l_5.
  - With the interaction controller a virtual point can be set where that is slightly larger than l_5 to make it stick to the surface of the environment.
- Make different models available for 3D Hoberman sphere.
  - The Mathematica model should be extended to a full 3D version where the potential energy component also needs to be added.
  - The motor set-up should be slightly changed to add another ellipse and some gears so that one motor can still apply the torque to the system.
  - With these changes the controller needs to be adjusted with different mass value.
- Make connection between Arduino and Matlab.
  - More advanced controllers can be tested more easily.
  - Better synchronised data between the various devices used.

A. Modelling simple planar case

A.1 Simple planar case with four moving point masses

In this section first the aim is the find the EoM and to check the controllability of the system. This is done by first getting the coordinates of the structure, from there build the Lagrangian. With the Lagrangian the state space solution can be found and the controllability can be checked.

A.1.1 Schematics of structure

Simplification to the simple planar case are:

- All lengths are the same (l).
- There is no gravity because the structure is in the x-y plane.
- All masses are point masses and are situated in points 2, 6, 10 & 14, this results in no inertia.
- The points 1, 2 & 3 are rigidly connected, as are the other quadrants.
- The point 0 is fixed in the translational domain.

The simple planar structure is shown in Figure A.1, where the different points of the mechanism are also shown. In the figure the point 0 is fixed to the world and has a revolute joint. A revolute joint is also attached to point 4, 8 &12. As described in the simplifications the other points are rigid bodies. As expected, there is a lot of symmetry, like in the Hoberman sphere, that simplifies equations as shown in the rest of this section.

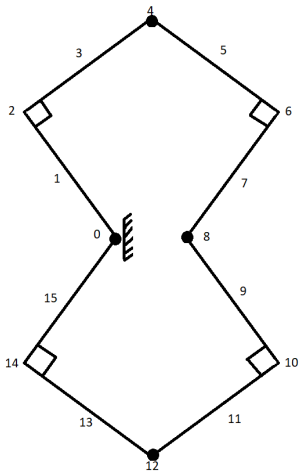

Figure A.1: Points of the simple planar case

In Figure A.2 the different angles are shown that builds up the simple planar structure. With this figure the entire system can be analysed mathematically.

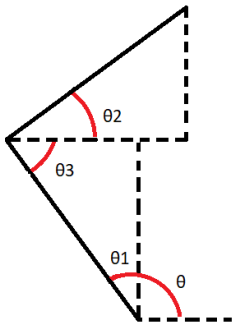

Figure A.2: Angles of the simple planar case

With the use of trigonometry and the combination of Equation A.1 & Equation A.2 it is shown that every point of the structure can be described as a function of θ.

$$\begin{aligned} \theta &= \frac{\pi}{2} + \theta_1 & \pi &= \theta_1 + \theta_3 + \frac{\pi}{2} & \frac{\pi}{2} &= \theta_2 + \theta_3 \\ \theta_3 &= \frac{\pi}{2} - \theta_1 & \theta_2 &= -\frac{\pi}{2} + \frac{\pi}{2} + \theta_1\theta_1 = \theta_2 & & \end{aligned} \tag{A.1}$$

$$\sin(\theta_1) = \sin(\theta - \frac{\pi}{2}) = -\cos(\theta) \qquad cos(\theta_1) = \cos(\theta - \frac{\pi}{2}) = \sin(\theta) \tag{A.2}$$

In Equation A.3 & Equation A.4 the different points are shown with their mathematical relation. Every point can be build up with the a combination of the first four coordinates due to symmetry.

$$\begin{aligned} 1 &= \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} & 6 &= \begin{bmatrix} 2x_4 - x_2 \\ y_2 \end{bmatrix} & 11 &= \begin{bmatrix} 2x_4 - x_3 \\ -y_3 \end{bmatrix} \\ 2 &= \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} & 7 &= \begin{bmatrix} 2x_4 - x_1 \\ y_1 \end{bmatrix} & 12 &= \begin{bmatrix} x_4 \\ -y_4 \end{bmatrix} \\ 3 &= \begin{bmatrix} x_3 \\ y_3 \end{bmatrix} & 8 &= \begin{bmatrix} 2x_4 \\ 0 \end{bmatrix} & 13 &= \begin{bmatrix} x_3 \\ -y_3 \end{bmatrix} \\ 4 &= \begin{bmatrix} x_4 \\ y_4 \end{bmatrix} & 9 &= \begin{bmatrix} 2x_4 - x_1 \\ -y_1 \end{bmatrix} & 14 &= \begin{bmatrix} x_2 \\ -y_2 \end{bmatrix} \\ 5 &= \begin{bmatrix} 2x_4 - x_3 \\ y_3 \end{bmatrix} & 10 &= \begin{bmatrix} 2x_4 - x_2 \\ -y_2 \end{bmatrix} & 15 &= \begin{bmatrix} x_1 \\ -y_1 \end{bmatrix} \end{aligned} \tag{A.3}$$

$$\begin{aligned} x_1 &= \frac{1}{2} l \cos(\theta) & y_1 &= \frac{1}{2} l \sin(\theta) \\ x_2 &= l \cos(\theta) & y_2 &= l \sin(\theta) \\ x_3 &= l \cos(\theta) + \frac{1}{2} l \sin(\theta) & y_3 &= l \sin(\theta) - \frac{1}{2} l \cos(\theta) \\ x_4 &= l(\cos(\theta) + \sin(\theta)) & y_4 &= l(\sin(\theta) - \cos(\theta)) \end{aligned} \tag{A.4}$$

A.1.2 Lagrangian

The Lagrangian equation is given by Equation A.5. In the equation V is the potential energy and because the simplification is made that there is no gravity, this term will disappear.

$$\mathscr{L} = T - V = \frac{1}{2} m v_i^2 - V \tag{A.5}$$

Due to the symmetry of the structure the following velocities are equal to each other.

$$v_2 = v_{14} \qquad v_6 = v_{10} \tag{A.6}$$

To simplify some equations further in the Lagrangian, these trigonometry rules are needed, see Equation A.7.

$$\cos^2(\theta) + \sin^2(\theta) = 1 \qquad \cos(\theta)\sin(\theta) = \frac{1}{2}\sin(2\theta) \qquad \cos(\theta)\cos(\theta) = \frac{1}{2}(\cos(2\theta) + 1) \tag{A.7}$$

For the Lagrangian the different velocities needs to be calculated, this is done at Equation A.8 & Equation A.9.

$$v_2^2 = \dot{x_2}^2 + \dot{y_2}^2 = l\dot{\cos}(\theta)^2 + l\sin(\theta)^2 = -l\dot{\theta}\sin(\theta)^2 + l\dot{\theta}\cos(\theta)^2 = l^2\dot{\theta}^2(\sin^2(\theta) + \cos^2(\theta)) = l^2\dot{\theta}^2 \tag{A.8}$$

$$\begin{aligned} v_6^2 &= (2\dot{x_4} - \dot{x_2})^2 + \dot{y_2}^2 = (2l\cos(\theta) + 2l\dot{\sin}(\theta) - l\cos(\theta))^2 + (l\dot{\sin}(\theta))^2 = (l\cos(\theta) + 2l\dot{\sin}(\theta))^2 + (l\dot{\sin}(\theta))^2 \\ &= (-l\dot{\theta}\sin(\theta) + 2l\dot{\theta}\cos(\theta))^2 + (l\dot{\theta}\cos(\theta))^2 = l^2\dot{\theta}^2\sin^2(\theta) + 4l^2\dot{\theta}^2\cos^2(\theta) - 4l^2\dot{\theta}^2\sin(\theta)\cos(\theta) + l^2\dot{\theta}^2\cos^2(\theta) \\ &= l^2\dot{\theta}^2(\cos^2(\theta) + \sin^2(\theta)) + 4l^2\dot{\theta}^2\cos^2(\theta) - 4l^2\dot{\theta}^2\sin(\theta)\cos(\theta) = l^2\dot{\theta}^2 + 4l^2\dot{\theta}^2\cos^2(\theta) - 4l^2\dot{\theta}^2\sin(\theta)\cos(\theta) \end{aligned} \tag{A.9}$$

With the velocities known the Lagrangian and the kinetic energy can be calculated, see the equation beneath.

$$\begin{aligned} T &= \frac{1}{2}m(v_2^2 + v_6^2 + v_{10}^2 + v_{14}^2) = m(v_2^2 + v_6^2) \\ &= m(l^2\dot{\theta}^2 + l^2\dot{\theta}^2 + 2l^2\dot{\theta}^2(\cos(2\theta) + 1) - 2l^2\dot{\theta}^2\sin(2\theta)) \\ &= m(4l^2\dot{\theta}^2 + 2l^2\dot{\theta}^2\cos(2\theta) - 2l^2\dot{\theta}^2\sin(2\theta))) = 2ml^2\dot{\theta}^2(2 + \cos(2\theta) - \sin(2\theta)) \end{aligned} \tag{A.10}$$

With the Lagrangian the EoM can be found out using the Lagrangian equation. In the equation τ are the torques that act on the system.

$$\frac{d}{dt}\left(\frac{\partial \mathscr{L}}{\partial \dot{\theta}}\right) - \frac{\partial \mathscr{L}}{\partial \theta} = \tau \tag{A.11}$$

In Equation A.12 the different parts of the Lagrangian equation are calculated.

$$\begin{aligned} \frac{\partial \mathscr{L}}{\partial \dot{\theta}} &= 4ml^2\dot{\theta}(2 + \cos(2\theta) - \sin(2\theta)) \\ \frac{d}{dt}(\frac{\partial \mathscr{L}}{\partial \dot{\theta}}) &= 4ml^2\ddot{\theta}(2 + \cos(2\theta) - \sin(2\theta)) - 8ml^2\dot{\theta}^2(\sin(2\theta) + \cos(2\theta)) \\ \frac{\partial \mathscr{L}}{\partial \theta} &= -4ml^2\dot{\theta}^2(\sin(2\theta) + \cos(2\theta)) \end{aligned} \tag{A.12}$$

When combining the previous two equations and filling it in, Equation A.13 is the solution.

$$\begin{aligned} \tau^T &= 4ml^2\ddot{\theta}(2 + \cos(2\theta) - \sin(2\theta)) - 8ml^2\dot{\theta}^2(\sin(2\theta) + \cos(2\theta)) + 4ml^2\dot{\theta}^2(\sin(2\theta) + \cos(2\theta)) \\ &= 4ml^2\ddot{\theta}(2 + \cos(2\theta) - \sin(2\theta)) - 4ml^2\dot{\theta}^2(\sin(2\theta) + \cos(2\theta)) \end{aligned} \tag{A.13}$$

A.1.3 State-space form

To work with the Lagrangian equation of motion the equation is rewritten in state space form. This can be done by applying the knowledge from digital control engineering.

In Equation A.14 the standard form of the SS is given.

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{A.14}$$

In this simple planar one DOF case, the states of the system are θ & $\dot{\theta}$. This is shown in the equation beneath.

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} \rightarrow \begin{bmatrix} \dot{x_1} = \dot{\theta} = x_2 \\ \dot{x_2} = \ddot{\theta} \end{bmatrix} \tag{A.15}$$

For the state space solution the $\ddot{\theta}$ needs to be known. For this the Lagrangian is rewritten to:

$$\begin{aligned} -4ml^2\ddot{\theta}(2+\cos(2\theta)-\sin(2\theta)) + 4ml^2\dot{\theta}^2(\sin(2\theta)+\cos(2\theta)) + \tau = 0 \\ 4ml^2\dot{\theta}^2(\sin(2\theta)+\cos(2\theta)) + \tau = 4ml^2\ddot{\theta}(2+\cos(2\theta)-\sin(2\theta)) \\ \ddot{\theta} = \frac{4ml^2\dot{\theta}^2(\sin(2\theta)+\cos(2\theta)) + \tau}{4ml^2(2+\cos(2\theta)-\sin(2\theta))} \end{aligned} \tag{A.16}$$

Here the same equation is rewritten into state space coordinates.

$$\dot{x_2} = \frac{4ml^2x_2^2(\sin(2x_1)+\cos(2x_1)) + \tau}{4ml^2(2+\cos(2x_1)-\sin(2x_1))} \tag{A.17}$$

However because the system is not linear the equations needs to be linearised. This linearisation is done around an equilibrium, where the e stand for the equilibrium. Where in the equilibrium the derivatives of the states and inputs should be equal to zero.

$$\begin{aligned} &\tau_e = 0 \qquad \dot{x_{1e}} = 0 \qquad \dot{x_{2e}} = 0 \\ &\dot{x_{1e}} = x_{2e} = 0 \qquad \dot{x_{2e}} = \frac{4ml^2x_{2e}^2(\sin(2x_{1e})+\cos(2x_{1e})) + \tau_e}{4ml^2(2+\cos(2x_{1e})-\sin(2x_{1e}))} = \frac{0(\sin(2x_{1e})+\cos(2x_{1e})) + 0}{4ml^2(2+\cos(2x_{1e})-\sin(2x_{1e}))} = 0 \end{aligned} \tag{A.18}$$

In Equation A.19 the different parameters are also rewritten to the equilibrium parameters so it can be used in the linearisation.

$$\begin{aligned} &x_1 = x_{1e} + \delta x_1 \qquad x_2 = x_{2e} + \delta x_2 \qquad \tau = \tau_e + \delta\tau \\ &\dot{x_1} = 0 + \delta\dot{x_1} = x_2 = \delta x_2 \qquad \dot{x_2} = 0 + \delta\dot{x_2} \qquad \tau = 0 + \delta\tau \end{aligned} \tag{A.19}$$

These partial derivatives are taken of these function at their equilibrium points.

$$\begin{aligned} \delta\dot{x_2} &= f(x_{1e}, x_{2e}, \tau_e) + \frac{\partial f}{\partial x_1} + \frac{\partial f}{\partial x_2} + \frac{\partial f}{\partial \tau} \\ \frac{\partial f}{\partial x_1} &= \frac{\partial}{\partial x_{1e}} \frac{4ml^2x_{2e}^2(\sin(2x_{1e})+\cos(2x_{1e})) + \tau_e}{4ml^2(2+\cos(2x_{1e})-\sin(2x_{1e}))} = \frac{8ml^2x_{2e}^2(1+\cos(2x_{1e})-\sin(2x_{1e})) + \tau_e(\cos(2x_{1e})+\sin(2x_{1e}))}{2ml^2(2+\cos(2x_{1e})-\sin(2x_{1e}))^2} \\ &= \frac{8ml^2*0^2(1+\cos(2*0)-\sin(2*0)) + 0*(\cos(2*0)+\sin(2*0))}{2ml^2(2+\cos(2*0)-\sin(2*0))^2} = 0 \\ \frac{\partial f}{\partial x_2} &= \frac{\partial}{\partial x_{2e}} \frac{4ml^2x_{2e}^2(\sin(2x_{1e})+\cos(2x_{1e})) + \tau_e}{4ml^2(2+\cos(2x_{1e})-\sin(2x_{1e}))} = \frac{8ml^2x_{2e}(\sin(2x_{1e})+\cos(2x_{1e})) + \tau_e}{4ml^2(2+\cos(2x_{1e})-\sin(2x_{1e}))} \\ &= \frac{8ml^2*0(\sin(2*0)+\cos(2*0)) + 0}{4ml^2(2+\cos(2*0)-\sin(2*0))} = 0 \\ \frac{\partial f}{\partial \tau} &= \frac{\partial}{\partial \tau_e} \frac{4ml^2x_{2e}^2(\sin(2x_{1e})+\cos(2x_{1e})) + \tau_e}{4ml^2(2+\cos(2x_{1e})-\sin(2x_{1e}))} = \frac{1}{4ml^2(2+\cos(2x_{1e})-\sin(2x_{1e}))} \\ &= \frac{1}{4ml^2(2+\cos(0)+\sin(0))} = \frac{1}{12ml^2} \\ \delta\dot{x_2} &= 0+0+0+\frac{1}{12ml^2}\delta\tau \end{aligned} \tag{A.20}$$

With the linearisation done, the state space equation can be filled in.

$$\begin{aligned} \begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{12ml^2} \end{bmatrix} u \\ y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} u \end{aligned} \tag{A.21}$$

With this new state space solution the controllability can be checked. This state space solution can however not be used in the control loop because the angle is not kept at zero but is changing from $0 - \pi/2$rad. However the controllability can still be checked.

The theory for controllability and reachability states that:

- The system is completely reachable if and only if $im(P) = \Re^n$
- The system is controllable in k steps if and only if $im(A^k) \subseteq im(P)$

From the theory it can be stated that if the system is reachable it is also controllable. In the theory P is the reachability matrix that is described and calculated beneath.

$$P = \begin{bmatrix} B & AB \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{12ml^2} \\ \frac{1}{12ml^2} & 0 \end{bmatrix} \tag{A.22}$$

As shown in the reachability matrix is that the system is full rank which means that the system is reachable thus controllable.

B. Mathematica code

```
<< ToMatlab`
SetAttributes[verbose, HoldAll];
verbose@expr_ :=
Module[{res = expr},
If[res =!= Null,
CellPrint@
Cell[BoxData@ToBoxes@res, "Output", CellTags -> "ans =",
ShowCellTags -> True]]];
HoldPattern@verbose@Set[lhs_, rhs_] :=
CellPrint@
Cell[BoxData@ToBoxes[lhs = rhs], "Output",
CellTags -> ToString@Unevaluated@lhs <> " =",
ShowCellTags -> True];
$Pre = verbose;
(*The text above is to ensure that the outputs are the variables \
themselfs for easy checking*)
Clear[\[Theta], m, \[Alpha], \[Beta], l2, x, l3, l4, l5, L1, L2, L3, \
L, r, sys, eq]
SetAttributes[m, Constant]
(*m=480/36;*)
n = 12;
\[Alpha] = \[Alpha] /. Solve[(Pi - \[Alpha])*n == 2*Pi];
\[Beta] = (Pi - \[Alpha])/2;

(*SetAttributes[l2,Constant]*)
l2 = 0.06537; (*It is 0.06537 according to SW*)
l1 = 2*Cos[\[Beta]]*l2;
l3 = (l1*Sin[\[Theta][t]])/Sin[Pi - \[Alpha]]
(*Niet constant maar is herschreven in C1 en C2 * Theta*)
l4 = (l2*Sin[\[Beta] + \[Theta][t]])/Sin[1/2*(Pi - \[Alpha])]

(*The calculation of the x coordinate of the whole system*)
For[i = 1, i <= n, i++ , x[i] = l3*Cos[((i - 1)*(Pi - \[Alpha]))]]
For[i = n + 1, i <= 2*n, i++ ,
x[i] = l4*Cos[((i - 1/2)*(Pi - \[Alpha]))]]
l5 = Simplify[2*x[n + 1] - l3]
For[i = 2*n + 1, i <= 3*n, i++ ,
x[i] = l5*Cos[((i - 1)*(Pi - \[Alpha]))]]

(*The calculation of the y coordinate of the whole system*)
For[i = 1, i <= n, i++ , y[i] = l3*Sin[((i - 1)*(Pi - \[Alpha]))]];
For[i = n + 1, i <= 2*n, i++ ,
y[i] = l4*Sin[((i - 1/2)*(Pi - \[Alpha]))]];
For[i = 2*n + 1, i <= 3*n, i++ ,
y[i] = l5*Sin[((i - 1)*(Pi - \[Alpha]))]];

(*The calculation of the velocity, the v here is v^2*)
For[i = 1, i <= 3*n, i++, v[i] = D[x[i], t]^2 + D[y[i], t]^2]
(*Print["v",i,"=",Simplify[v[i]]]*)
T = Simplify[Sum[1/2 m*Simplify[v[i]], {i, n*3}]]

L1 = Simplify[D[T, D[\[Theta][t], t]]];
L2 = Simplify[D[L1, t]];
L3 = Simplify[D[T, \[Theta][t]] /. D[D[\[Theta][t], t]] -> Constant];
L = Simplify[L2 - L3] /. Constant -> D[D[\[Theta][t], t]]
LL = Simplify[N[L, 3]]

eq = Simplify[Solve[L == \[Tau], D[D[\[Theta][t], t], t]]]
eq1 = N[eq, 2];
sys = StateSpaceModel[{L == \[Tau][t]}, {{\[Theta][t],
0}, {\[Theta]'[t], 0}}, {{\[Tau][t], 0}}, {\[Theta][t]}, t]
sys1 = N[StateSpaceModel[{L == \[Tau][t]}, {{\[Theta][t],
0}, {\[Theta]'[t], 0}}, {{\[Tau][t], 0}}, {\[Theta][t]}, t], 3]

ToMatlab[eq];
ToMatlab[eq1];
```

C. Bearing and bolt options

Due to the fact that the Hoberman sphere has a lot of joints and thus multiple moving parts, friction needs to be minimized. This can be done with different bolt and different bearing options. One of the requirements of the interaction controller was also that there was low friction.

C.1 Sholderbolt

The bolt that is used in the Hoberman sphere is a shoulder bolt, shown in Figure C.1. The advantage of the shoulder bolt is that the middle part (L) is smooth and results in less friction. The limitation however of the shoulder bolt is that the minimum tread thickness is M2,5 and L is 10mm. This results in the dismissal of option 2mm and 3mm as an usable scissor part.

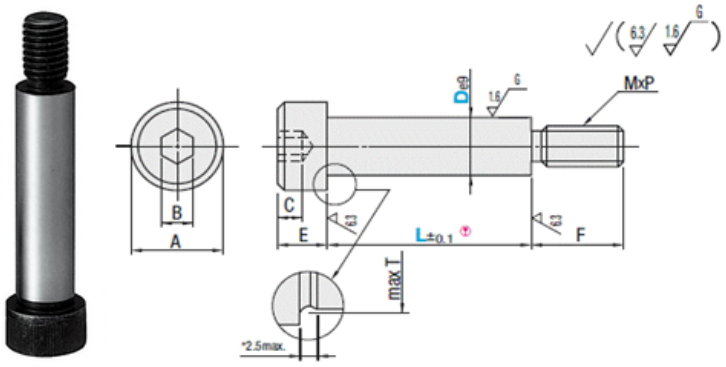

Figure C.1: Sholderbolt with specifications, ([18])

With the knowledge of a minimum L of 10mm, the 5mm scissor part is redesigned into a 4mm part. In the scissor joint two scissor parts and a bearing can be used with the shoulder bolt to fill the 10mm, resulting in 2mm for the bearings.

C.2 Sleeve bearing

An option for a bearing is the use of sleeve bearings, shown in Figure C.2. The material that the bearing is made up of has a very low friction coefficient which makes is very useful in the Hoberman sphere. This bearing will be placed in between the two scissor parts and this will result in the need of two bearings per connection.

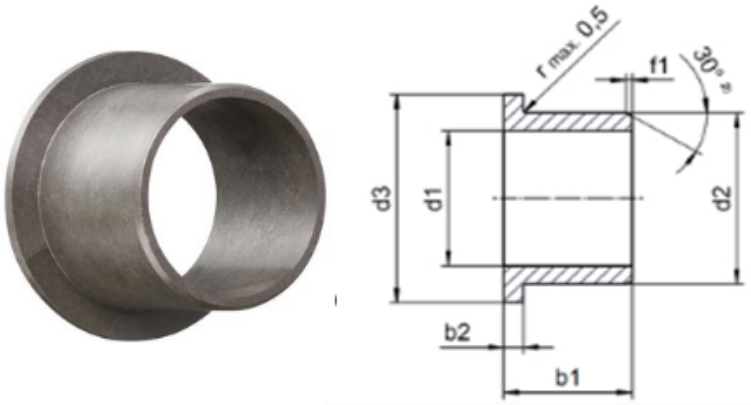

Figure C.2: Sleeve bearing with specifications, ([19])

This bearing specification also has some limitations, namely that d1 has a minimum of 3mm with a stepsize of 1mm. The d1 here should be equal to the D of the shoulder bolt. Another limitation is that b1 should be equal or less than the thickness of the scissor part (4mm), for the reason that the scissor part will be loose on the bearing. Knowing that the thickness of the scissor part is 4mm there is 2mm left within the shoulder bolt to be used for the bearing. This results in a limitation that b_2 should be equal or less than 1mm. If b_2 is smaller than 2mm an extra washer is necessary to fill up the remaining space.

C.3 Washer bearing

The cheaper and easier options is to use a washer instead of the sleeve bearing. With the flexibility here at the university of being able to print/lasercut materials at the desired size this option is chosen for now. The main reason for this option is as stated before, being more flexible, cheaper and there is no shipping time that delays the experiment. When concluded in the experiment that the friction in the system is a problem, the option of the sleeve bearing will be used. This is explained in chapter 7.

C.4 Choice bolt and bearings

With the limitations described, there are a few options left for the shoulder bolt in combination with either the sleeve or washer bearing. For the shoulder bolt the options are either a M2,5 with 4mm D or a M3 with 5mm D. From the size thickness check it was recommended to use the M3 version instead of the M2 or M2,5 version. When applying that here only the M3 with 5mm D is left.

With that there are two options for the sleeve bearings that fit the previously set criteria which is shown in Table C.1. In the table the part-numbers are shown and the only difference between the options is 0.5mm in b_1. The logical choice is here to use the GFM-0506-04, because it fills the whole scissor part.

Table C.1: Sleeve bearing options

Part-number (-)	$d_1(mm)$	$d_2(mm)$	$d_3(mm)$	$b_1(mm)$	$b_2(mm)$
GFM-0506-04	5	6	10	4	0.5
GFM-0506-04	5	6	10	3.5	0.5

With that there is still a 1mm gape that needs to be filled with a washer because of the 0.5mm b_2 parameter. This 1mm is divided into two 0.5mm washers that can be added on the bolt side and the nut side. Where the bolt side has a d1 of 5mm and a d_2 of 10mm and the nut side has a d_1 of 3.2mm and a d_2 of 8mm.

D. Servo motor response test

D.1 Servo response

The last test with the servo motor is to check its response when given a certain reference profile. In this test the aim is to find the response of the system with respect to the servo motor with the build in controller. Another aim of this test is to check the connection between Matlab and Arduino, and if this is usable.

In this test Matlab gives the input for the Arduino and the potentiometer is left out. The input for the motor is to follow the reference profile that is adapted to different simulation time, to check the response. The different simulation times are 5, 2.5, 1, 0.5, 0.25s, and are given to Equation 6.3 to determine the new maximum acceleration that needs to be set in the reference profile block. In this test the result of the previous test will be used to change the Hoberman sphere coordinates of the reference profile to the motor coordinates. The reference position is set by Matlab to Arduino by the use of a toolbox (Matlab support package for Arduino Hardware). In here basic commands can be sent by Matlab to Arduino.

The expectation is that the servo motor cannot achieve the 0.25s simulation time. This is because in the datasheet, [20] the motor can move at a velocities of $0.22sec/(\pi/3)$ at 4.8V, this means for $\pi/2rad$ the minimum amount fo time the motor can achieve that is 0.33s.

Now the response of the servo motor is checked. In Figure D.1 the response is shown of the servo motor with a simulation time of 5s. The right figure gives the response and the left gives the error between the trajectory and the Hoberman sphere. The aim here is to have a setpoint error of zero after the simulation time. As can be seen in the figure this is achieved.

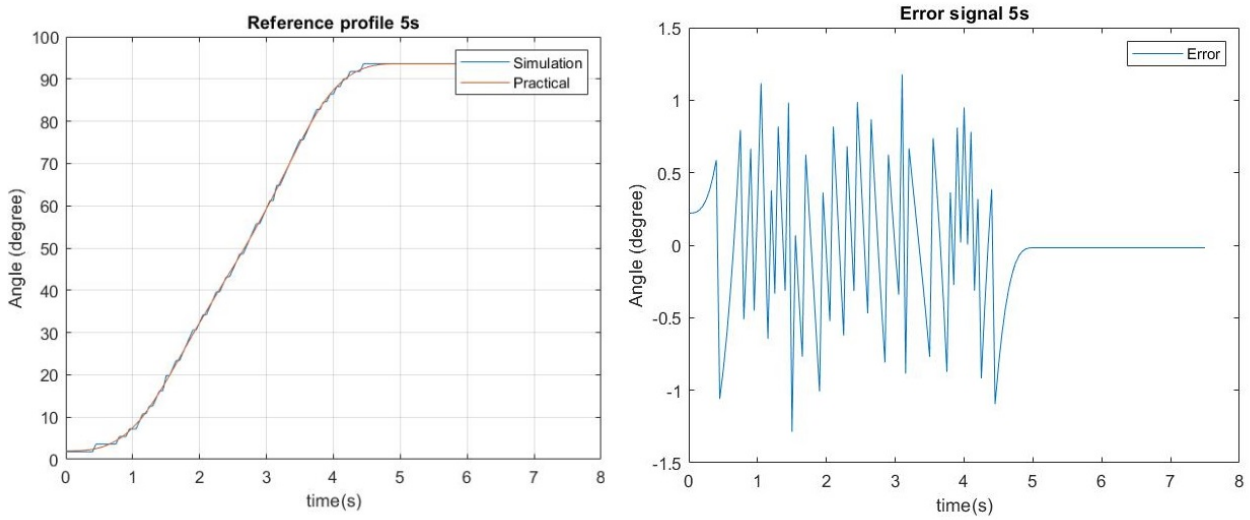

Figure D.1: Response of the servo motor within 5s

From the figure it can be concluded that the servo motor uses a PID controller. This can be concluded because there is no steady state error, which can be assigned to an integral action. The derivative action can also be seen because the controller is fast in its response and has some overshoot at different point of the plot. What is also observed is that the system stops and goes every time a setpoint is reached. This can have multiple reasons, where one is the accuracy of the encoder, but also the stick slip effect could have a contribution. The accuracy of the encoder is unknown and it has influence on the measurement however the stop and go behaviour occurs with a step larger than 0.02rad. This is an indication that it is more likely due to friction in or on the system.

The next step is to shorten the simulation time and see if the system can handle that. The response of the faster simulation time is shown in Figure D.2, and it shows that the system still can follow the input but the error is not zero. Here the previous found stop and go behaviour is almost gone, which suggest that it was the stick slip effect and the friction in/on the system that causes that behaviour. This can be concluded because now the time is shorter and thus there is a larger control power to overcome that effect.

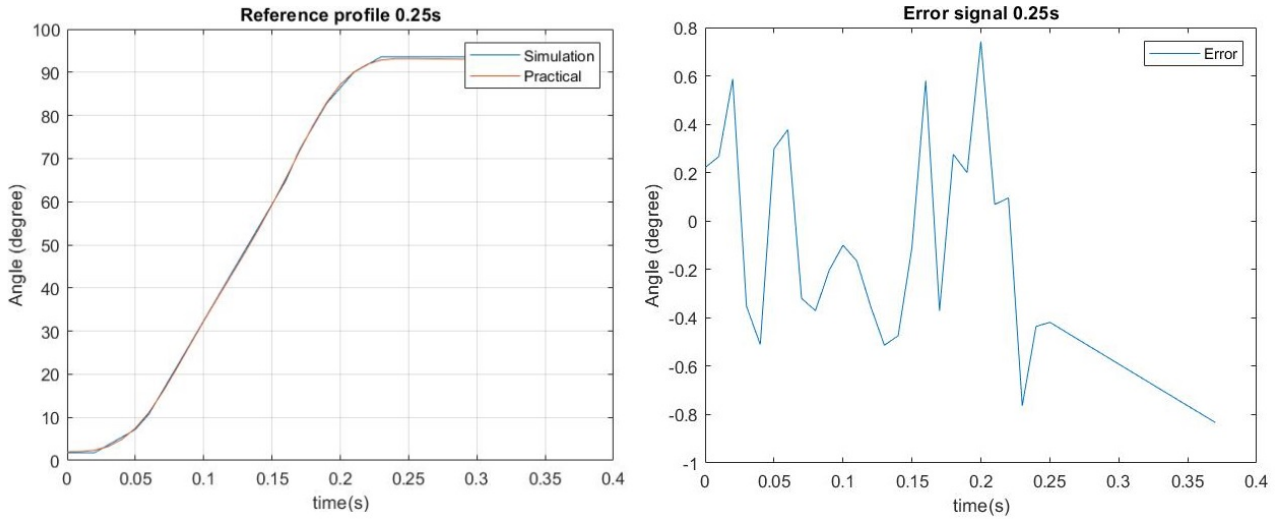

Figure D.2: Response of the servo motor within 0.25s

However according to the specification of the motor the system should not be able to follow the set trajectory. The operating velocity of the servo motor is 45.5RPM which results in 0.33s for $\pi/2rad$ change in angle without a load. The time for the motion was set for 0.25s. During the measurement it was also noticed that the system does not correctly responds to the simulation time of Matlab. The system sees the setpoint it is given and goes there even if it is outside of the simulation time. This means that the time axis that is shown in the plot is not correct.

The problem with the time axis is also the reason why the PID controller does not get a steady state error of zero. This is because the controller does not have the time after the setpoint being reached to do anything. The final setpoint is given when the loop is already finished. The fastest from the measurements done where the error is about zero is with a simulation time of 1s. For the servo motor this seams to be the fastest it can go with a load attached to it.

E. Arduino code

```

#include <Wire.h>
#include <Adafruit_MotorShield.h>
#include "utility/Adafruit_MS_PWMServoDriver.h"
#include <PID_v1.h>
#define AS5600  0x36

#define PIN_OUTPUT 2
#define PIN_OUTPUT 3
#define PI 3.14159265358979323846264338327 95

Adafruit_MotorShield AFMS = Adafruit_MotorShield();
Adafruit_DCMotor *myMotor = AFMS.getMotor(1);

double setpoint, input, output, vel_output;
double kp = 40.602, ki = 350.9457, kd = 1.0439;

int rise_time = 0, fall_time = 0, pwm_value = 0, period = 0;
double angle, angle_f, angle_s;

double MOTION_T = 5000000, T_HOMING = 1000000, REF_END = 52.0/180.0*PI, REF_START = 24.0/180.0*PI;
double t = 0, t_loop = 0, t_elapse = 0, t_begin = 0, ref = REF_START, t_ref=0;
double J1 = 0.0, J2 = 0.0, HM=0.0, a=0.0, a_old=0.0, v=0.0, v_old=0.0, x=0.0, x_old=0.0, x_new=REF_START;

double MAX_SPEED = 255;

PID myPIDForward(&input, &output, &setpoint, kp, ki, kd, DIRECT);
PID myPIDBackward(&input, &output, &setpoint, kp, ki, kd, REVERSE);

void setup() {
Serial.begin(115200);
// read PWM for angle measurement
attachInterrupt(digitalPinToInterrupt(PIN_OUTPUT), rising, RISING);
//turn the PID on
myPIDForward.SetMode(AUTOMATIC);
myPIDBackward.SetMode(AUTOMATIC);
AFMS.begin();
digitalWrite(2, LOW);
HM=REF_END-REF_START;
J1 = 32*HM/pow(((MOTION_T-T_HOMING)/1000000),3); //0.1438 MOTION_T=5
J2 = -32*HM/pow(((MOTION_T-T_HOMING)/1000000),3); //0.1438 MOTION_T=5;
}

void loop() {
digitalWrite(2, HIGH);
uint32_t ts1 = micros();
referenceGen(t);
t_ref=(t/1000000)-1;
setpoint = ref;
input = angle;

if (t <= 2 * MOTION_T) {
if (angle < ref) {
myPIDForward.Compute();
vel_output = limit_velocity(output);
myMotor->setSpeed(vel_output);
Serial.print("PID if:"); Serial.print(vel_output); Serial.print('\t');
myMotor->run(FORWARD);
} else {
myPIDBackward.Compute();
vel_output = limit_velocity(output);
myMotor->setSpeed(vel_output);
Serial.print("PID else:"); Serial.print(vel_output); Serial.print('\t');
myMotor->run(BACKWARD);
}
delay(10);
digitalWrite(2, LOW);
uint32_t ts2 = micros();
t_loop = ts2 - ts1;
t = t + t_loop;
Serial.print("Angle:"); Serial.print(input); Serial.print('\t');
Serial.print("Ref:"); Serial.print(setpoint); Serial.print('\t');
Serial.print("t:"); Serial.println(t);
}
else {
```

```
myMotor->run(RELEASE);
}
}

void referenceGen(double t_gen) {
if (ref < REF_END) {
if (t < T_HOMING) {
ref = REF_START;
}
else {
if(t<((MOTION_T-T_HOMING)*0.25+1000000) &&(t>T_HOMING)){
a_old = a;
a = t_loop/1000000*J1+a_old;
v_old=v;
v = v_old+t_loop/1000000*((a+a_old)/2.0);
x_old=x;
x =x_old+t_loop/1000000*((v+v_old)/2.0);
x_new=REF_START+x;
}

if((t>=(MOTION_T-T_HOMING)*0.25+1000000) &&(t<(MOTION_T-T_HOMING)*0.50+1000000)){
a_old = a;
a = t_loop/1000000*J2+a;
v_old=v;
v = v_old+t_loop/1000000*((a+a_old)/2.0);
x_old=x;
x =x_old+t_loop/1000000*((v+v_old)/2.0);
x_new=REF_START+x;
}

if((t>=(MOTION_T-T_HOMING)*0.5+1000000) &&(t<(MOTION_T-T_HOMING)*0.75+1000000)){
a_old = a;
a = t_loop/1000000*J2+a;
v_old=v;
v = v_old+t_loop/1000000*((a+a_old)/2.0);
x_old=x;
x =x_old+t_loop/1000000*((v+v_old)/2.0);
x_new=REF_START+x;
}

if((t>=(MOTION_T-T_HOMING)*0.75+1000000)&&(t<MOTION_T)){
a_old = a;
a = t_loop/1000000*J1+a;
v_old=v;
v = v_old+t_loop/1000000*((a+a_old)/2.0);
x_old=x;
x =x_old+t_loop/1000000*((v+v_old)/2.0);
x_new=REF_START+x;
}

ref = x_new;
}
}
else {
// finish the motion
ref = REF_END;
}


}

double limit_velocity( double velocity ) {
if (velocity > MAX_SPEED) {
vel_output = MAX_SPEED;
}
else {
vel_output = output;
}
return vel_output;
}

void rising() {
attachInterrupt(digitalPinToInterrupt(PIN_OUTPUT), falling, FALLING);
int curr_time = micros();
period = curr_time - rise_time;
rise_time = curr_time;
}

void falling() {
attachInterrupt(digitalPinToInterrupt(PIN_OUTPUT), rising, RISING);
fall_time = micros();
pwm_value = fall_time - rise_time;
angle_f = (4351.0 / period) * pwm_value - 128.0;
angle_s = map(angle_f, 0, 4087, 4087, 0);
angle = (angle_s / 4095 * 360 / 2 - 58)/180.0*PI;
}
```

F. Extra work

In this appendix some extra work is discussed which is not implemented in the research.

F.1 Open loop response

In Figure F.1 the new open loop model is shown. In this model a 1/ts term is added to get the unit impulse response. This is checked by integrating the signal from the impulse response and its equal to one.

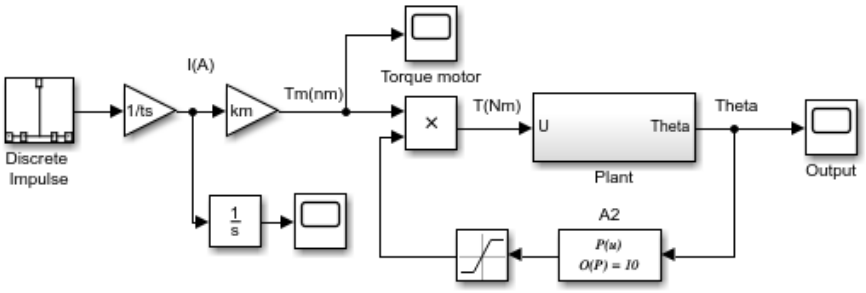

Figure F.1: Open loop model

From the new model a frequency response is determined and the result is shown in Figure F.2. In the frequency response now a more linear response is shown which is consistence with a response of a single mass.

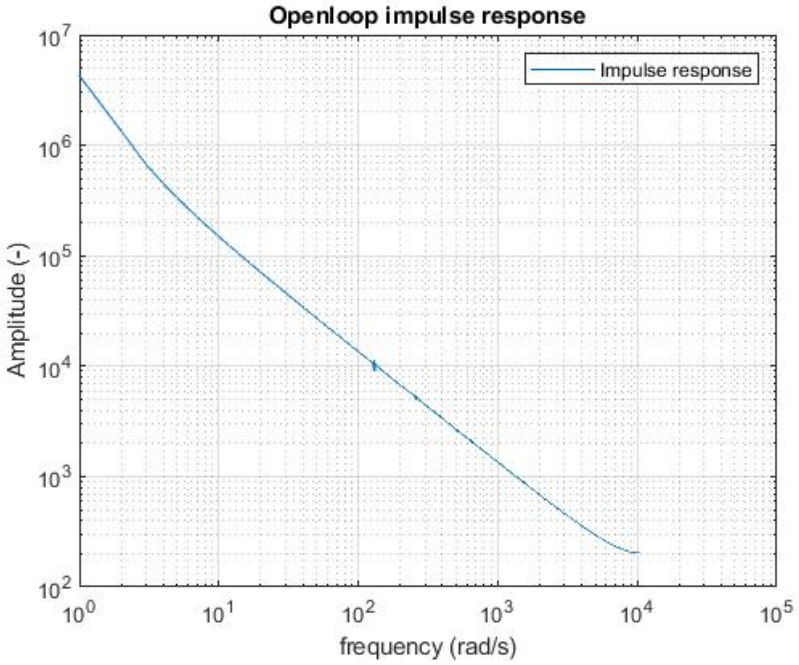

Figure F.2: Open loop unit impulse response

F.2 DC-motor

From the measurement in Figure 7.8, the internal resistance of the motor can also be calculated. This is done with the supply voltage and the stall current. The stall current is the current that flows when the motor is turned on but the motor does not starts rotating. In the figure that is the peak current when the motor is just turned on. The value that is used for the stall current is the average of the different configuration used in that test.

$$R_{internal} = \frac{V_{supply}}{I_{stall}} = \frac{12}{1.05} = 11.43\Omega \tag{F.1}$$

This internal resistance can be used to calculated the back EMF from the motor with the following equation:

$$V_{BEMF} = V_{supply} - IR_{internal} \tag{F.2}$$

Bibliography

[1] C. Balaguer, R. Montero, J.G. Victores, S. Martinez, A. JardÃşn, *Towards Fully Automated Tunnel Inspection: A Survey and Future Trends*, ISARC 2014

[2] J. Caradot, P. Rouault, F. Clemens, F. Cherqui, Evaluation of uncertainties in sewer condition assessnment, Structure and Infrastructure Engineering, 14:2, 264-273, DOI: 10.1080/15732479.2017.1356858, 2017

[3] E. Dertien, S. Stramigioli, K. Pulles, Development of an inspection robot for small diameter gas distribution mains, International Conference on Robotics and Automation, 2011

[4] P.A. Bernhardt, C. Hoberman, D.L. Hysell, A.C. Nicholas, M.W. Nurnberger, Expandable Signal Calibration Target, Patent No: US 7948425B2, May 24 2011

[5] S.K. Agrawal, S. Kumar, M. Yim, J.W. Suh, *Polyhedral single degree-of-freedom expanding structures*, Proceedings 2001 ICRA. IEEE International Conference Robotics and Automation, 2001

[6] C. Hoberman, Folding covering panels for expanding structures, Patent No.: 6,834,465 B2

[7] F. KovÃącs, T. Tarnai, P.W. Fowler, S.D. Guest, A class of expandable polyhedral structures, International Journal of Solids and Structures 41 1119-1137, 2004

[8] https://www.healthline.com/health/stent, Reviewed by: D. Sullivan PhD, 2017

[9] H.F. Verheyen, The complete set of Jitterbug transformers and the analysis of their motion, Computers Math. Applic. Vol. 17, No1-3 pp 203-250, 1989

[10] F.M.J. Pfister, S.K. Agrawal, Analytical Dynamics of Unrooted Multibody Systems with Symmetries, ASME 440/vol. 121 September 1999

[11] O.M. O'Reilly, P.C. Varadi, Hoberman's Sphere, *Euler Parameters and Lagrange's Equations*, Journal of Elasticity, 2000

[12] http://www.instructables.com/id/Hoberman-Sphere/, 2017

[13] S.K. Agrawal, S. Kumar, M. Yim, *Polyhedral single degree-of-freedom expanding structures: Design and Prototypes*, Journal of Mechanical Design, sep 2002 vol 124/473

[14] K.Wohlhart, Kinematics and dynamics of the Fulleroid, Multibody system dynamics 1: 241-258, 1997

[15] H.F. Verheyen, The complete set of Jitterbug transformers and the analysis of their motion, Computers Math. Applic. Vol 17, NO 1-3 pp. 203-205, 1989

[16] S. Stramigioli, Modern Robotics: Lecture 5 Position control and Lecture 6 Interactive control [Powerpoint slides], Course at University Twente 2016

[17] S. Stramigioli, Creating Artificial Damping By Means Of Damping Injection, In Proceedings of the ASME Dynamic Systems and Control Division, pages 601âĂŞ606, 1996

[18] https://uk.misumi-ec.com/vona2/detail/110300249140/?CategorySpec=00000230806

[19] https://www.igus.com/product/64

[20] http://www.produktinfo.conrad.com/datenblaetter/275000-299999/275098-da-01-en-Quarter_Scale_Servo_VS_11_AMB.pdf

[21] https://docs-emea.rs-online.com/webdocs/0030/0900766b80030030.pdf

[22] https://ams.com/documents/20143/36005/AS5600_DS000365_5-00.pdf/649ee61c-8f9a-20df-9e10-43173a3eb323

[23] J. van Dijk, R. Aarts, *Analytical one parameter method for PID motion controller settings*, IFAC conference on Advances in PID control, March 2012