Six degree of freedom stage with electromagnetic actuators

Six Degree of freedom stage with electromagnetic actuators

J.A.F.M. Simons (s478530)
N.C.T. de Kleijn (s470851)

DCT 2005.123

DCT traineeship report

Supervisors:
Dr. Hector Gutierrez
Prof. Dr. Ir. M. Steinbuch

Florida Institute of Technology (FIT)
Dept. of Mechanical and Aerospace engineering
Melbourne, Florida, USA

Technische Universiteit Eindhoven
Department of Mechanical Engineering
Dynamics and Control Technology Group

January 17th to April 22nd 2005

Contents

List of Figures

Preface

Long before any plans to go to Florida were ever made, we were both very much looking forward to our international internship. We were both making our own arrangements, but somehow wound up going together. We have known each other for a long time so we knew we would work well together, and we did. After our time abroad is over, it turned out to be much better than we ever hoped.
When we arrived in Melbourne, Florida, we rolled right into our apartment and got to work with Hector. We started together with designing a one degree of freedom test-setup to figure out what the electromagnetic actuators are actually capable of. After this, Niels concentrated on the preparing the control part and making some models. Jasper designed the test-setup in detail and spent two weeks in the FIT-machineshop to manufacture all the parts. After the test-setup was completed, we worked together again with the electrics and the characterizing measurements. Niels then controlled the test-setup to a steady state error of 200 [nm]. Jasper used the behavior of the actuators to design a six degree of freedom motion platform with preloaded actuators. These actuators were also manufactured and tested. This proved the feasibility of the 6DOF design.
Coming from the Eindhoven University of Technology, where all projects are extensively prepared for us and plenty of staff is available to help us, we learned a lot from finally having to do everything ourselves. The literature, the theory, mechanical design, manufacturing, wiring all the electronics and measuring accurately with some repeatability and so forth. We both feel that working in Florida for these four months has brought us a very big step closer to becoming a mechanical engineer.

All of this would not have been possible without the help from some great people we would like to thank: First of all: Doctor Hector Gutierrez, our supervisor from Florida Institute of Technology. Besides all the work he did for us, everything he helped us with and all the trust he showed in our work, he also became a good personal friend. Next: John Lee and Bill Bailey from the FIT-machineshop. They were very open to let us work in their machineshop and after a few days of working together, we got along perfectly. Especially John has helped us a lot with writing some NC-files for some parts and could always answer all our questions. It was great working together. Larry Buist for helping us with the electronics. Also: Professor Maarten Steinbuch from Eindhoven for arranging our placement in Florida and keeping an eye on us from the Netherlands. And Finally, Ben van den Elshout and Stefan Voeten, the two students before us, for showing us around and getting us settled in.

Jasper and Niels

Chapter 1

Introduction

Electromagnetic actuating is a tried and known concept, however, it is currently not being applied much in high-precision machinery. The forces exerted by these actuators depend strongly on material properties (magnetic permeability), the shape of the core and its actuated counterpart; this makes it difficult to apply these actuators. Also, the force depends on the squares of both current and gap, which makes the actuators highly unstable (i.e. decrease of gap results in a quadratically higher force, resulting in an even smaller gap).

A six degree of freedom (6DOF) machine has been designed and built by Lionel Fevre [2]. In this machine, four E-shaped, laminated actuators are used to control a beam horizontally and vertically. Four of these controlled beams connect to a central stage.

In order to characterize these E-shaped actuators, a one degree of freedom test-setup was built (one rotation around a horizontal axis). This setup yielded some results, however, there were many problems with the poor hinge point and the fact that the actuators also had to compensate for gravity. Because of these problems, it was very difficult realize a working controller for this test-setup and for the 6DOF-machine.

Because of this, it was decided to make a new 6DOF design for the manipulation of a tray. But, to be able to do this, proper knowledge of the behavior of these actuators is required. This is why it was decided to design and build a new 1DOF test-setup without the above mentioned problems. The goal of the design is to allow good characterization of the E-shaped and perhaps other actuators. This characterization is the parameter fit to the known relation between force, current and gap:

$$F = K \cdot \frac{I^2}{g^2}$$

Several different concepts were considered, one was chosen and worked out. Design drawings were made and the design was built at the FIT machine shop. After this characterization is done, several concepts are considered for 6DOF motion control. This has lead to the maglev hexapod design which relies on preloaded actuators. Meanwhile several different control strategies are tested and tuned on the 1DOF test-setup. This report will show the feasibility of the maglev hexapod design with the preloaded actuator and working control laws.

Chapter 2

1DOF Design concepts

2.1 Requirements and assumptions

Previous experience at FIT and a paper by David Trumper from MIT [11] show that the test-setup has to be free of play, friction and hysteresis. In order to achieve this, elastic suspension of the actuated part is a logical solution. Both for control and characterization, an accurate measurement is required, preferably non-contact. For characterization, the force exerted by the actuators has to be measured accurately while still keeping a constant gap. Finally, the design should be suited for rotation around a vertical axis (without gravity) and around a horizontal axis (with gravity), so that gravity compensation with either permanent magnets or a separate set of electromagnets will be possible. In the current 6DOF design, travel for the actuators is less than one millimeter. Therefore, this setup will be designed to have an initial gap of 500 [μm] on either side. This allows total travel to be one millimeter. Because most parts for the 6DOF machine were already bought, the decision was made to use the old E-shaped actuators as well as the capacitive gapsensors and a strain gage force sensor. Besides the E-shaped actuator, an actuator with a round core is also designed. All other parts have to be machined.

2.2 Pure translating design

Some of the problems with the old 1DOF setup were caused by poor measurement. The capacitive sensors measured on a stack of laminations; a very rough surface. Also, the shape of the measured gap becomes trapezoidal because of the rotation. Pure translation and a well-made surface would provide a much better measurement. To realize this, a design was made, using a parallel guidance with leafsprings. To prevent statical overdetermination, one of the leafsprings needs to be cut into. Furthermore, to prevent moments on the leafsprings, the force must be applied at the middle of the leafsprings' length. Finally, to provide equal in-plane stiffness, the other leafspring is also cut, to have the same cross-section. These constraints yielded the following design (figure 2.1).

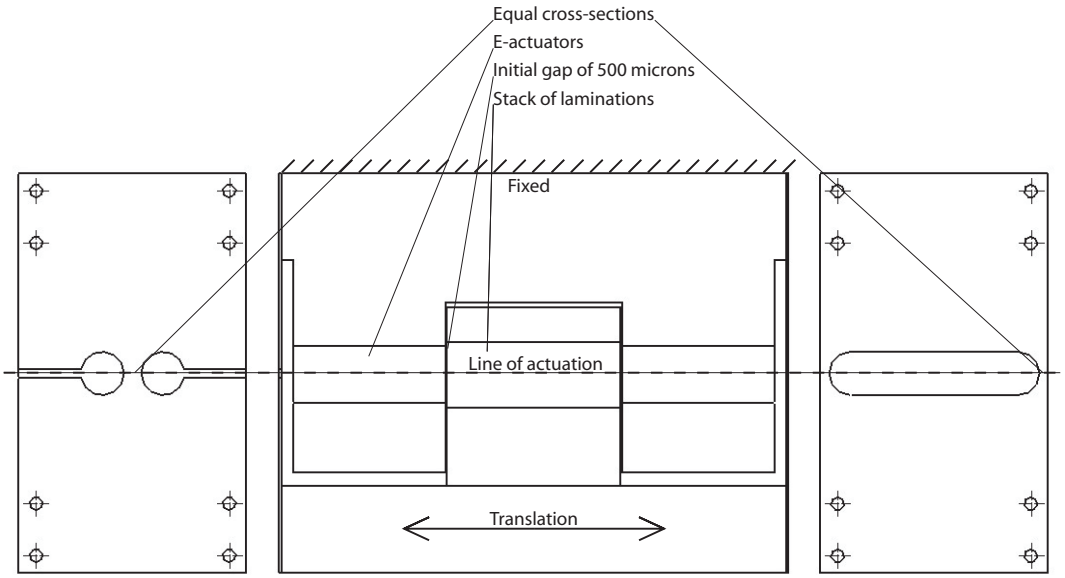


Figure 2.1: Pure translating design

Analysis of this design showed that it meets basic requirements. Vertical travel caused by horizontal movement is negligible $(0,8[\mu m])$. The first eigenfrequency is at 8,5 [Hz]; higher eigenfrequencies start at 300 [Hz]. The stiffness is 1002 [N/m] and can easily be altered by using thicker leaf springs.

2.3 Pure rotating design

The other possibility for 1DOF is a setup with only one free rotation. By designing an arm, fixed to an elastic hinge point, the stack of laminations at the other end of the beam makes a pure rotation. Depending on the length of the beam, this closely resembles a translation. The travel at the end of the beam is still 500 $[\mu m]$ either way. With a beamlength of 400 [mm], this corresponds to an angle of $\pm 1,25$ $[\mu rad]$. Dimensions such as beam-length and -wallthickness, and the hingeblock were set at realistic values and then tuned slightly. This design has its first eigenfrequency ay 9 [Hz] and higher eigenfrequencies from 300 [Hz]. The stiffness is 3650 [N/m] and the capacitive sensor can also be targeted at a smooth milled surface. At these dimensions, neither plastic strain nor fatigue will ever occur.

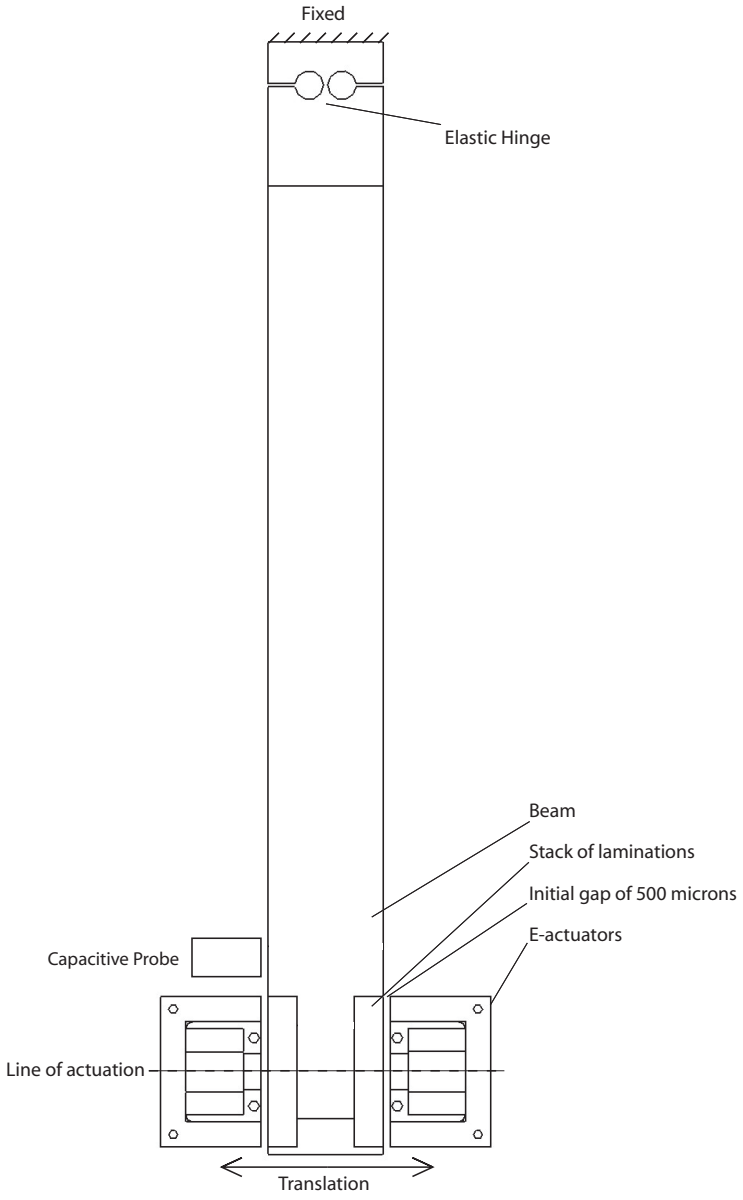


Figure 2.2: Pure rotating design

2.4 Concept choice

Both designs meet basic requirements such as travel, improved guidance and measurement and the ability to operate both horizontally as well as vertically. Nonetheless there are some pro's and cons on both designs:

Leaf spring design
This design always keeps a parallel gap, which will avoid any complications that may occur with a trapezoidal gap. On the other hand, the machining of the leafsprings must be done with EDM, which will be very expensive to do [1]. Also, this design has to be hung from some sort of rigid support, making it even harder to assemble and reach into for making any adjustments. There will be little room for mounting different types of actuators.

Hinged beam design
Although this design does not keep a parallel gap, the angles that do occur are very small, and can be controlled with the length of the beam. Also, this design can be mounted on a base-plate and then be placed on any table. It can be approached from all directions and allows plenty of space for measurement and different actuators.

In effect, the choice of design is dictated by the influence of the trapezoidal gap. Inquiry at the company that produces the capacitive sensors yielded a document concerning this issue [6]. This document provided the following formula:

$$Error = \left(\frac{1 - \sqrt{1 - \left(\frac{r \cdot \Theta}{d}\right)^2}}{2} \right) \cdot d \cdot k$$

with:

r	= radius of probe sensing area	$[m]$
d	= probe/target gap measured at probe center	$[m]$
Θ	= probe/target angle	[rad]
k	= experimentally determined value, default=5	$[-]$

This shows that for the hinged beam design, the resulting error will be approximately 80 [nm]. This error is negligible and can also be predicted very well. Considering the above, the hinged beam design is chosen.

[1] Based on a quotation from Florida Precision Machining

Chapter 3

Hinged beam design

The chosen design is built up out of several imported parts, namely: the flexures, a beam, mounting of a magnetic target and the mounting of the actuators. All of these parts need to be mounted to a base to define their mutual positions and to place the entire setup both horizontally as well as vertically. The design of all these parts is described below, technical drawings are available in appendix A.

3.1 Flexures

One of the requirements for the design of the setup is elastic hinging. This provides motion that is free of play, friction and hysteresis. There are several solutions for constructing an elastic hingepoint, such as a flat leafspring (a), crossed leafsprings (b) and 'hole-hinges' (c) (figure 3.1).

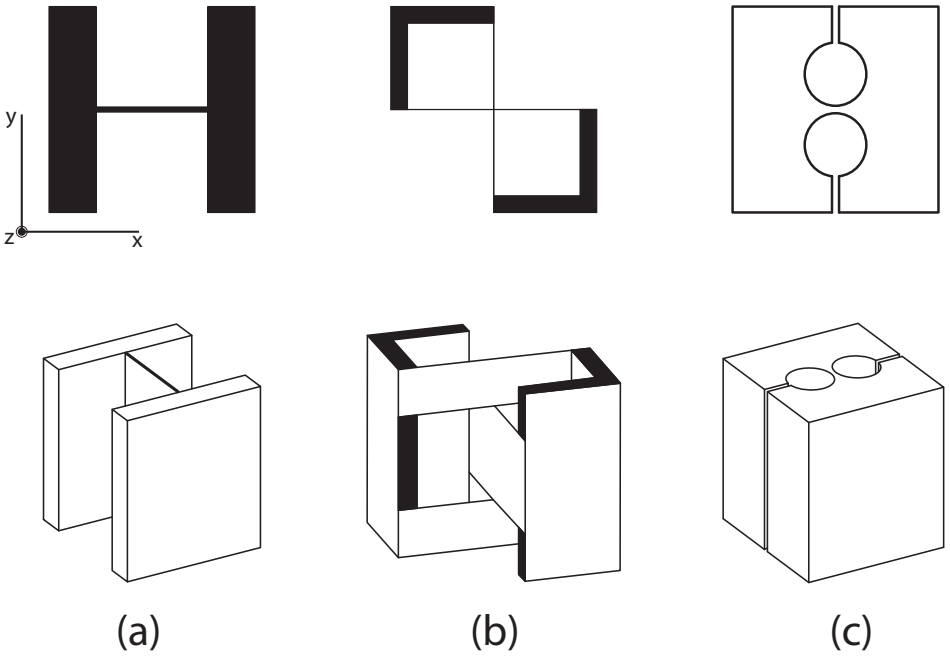


Figure 3.1: Several elastic hinges

Since the hingepoint needs to provide a pure rotation, the leafspring in figure 3.1a will not suffice; this hinge can make a double bend, allowing an additional Y-translation, also, torsional stiffness around the X-axis will be too low. The crossed leafsprings from 3.1b are not very good at supporting forces in Z-direction or moments around the Y-axis. These loads are present in the final setup, caused by the weight of the beam and the magnetic target at its end. This leaves the hinges shown in figure 3.1c. These will not allow a double bend or torsion around the X-axis and support forces in Z-direction better. By creating two of these hinges above one-another, the weight of the beam will be supported much better. Calculations by hand, verified by finite element analysis (FEA) are used to tune the stiffness and strength, leading to the final design shown in figure 3.2.

3.2 Beam

The beam connects the magnetic targets to the hingeblock. The length of the beam determines the angle that corresponds with the travel of 1mm at the actuators. The beam needs to be stiff enough against bending around the Y-axis to support its own weight and that of the magnetic targets. Stiffness is also required against bending around the Z-axis, making it much stiffer than the hinges. This minimizes deflection of the beam, caused by actuator-force and the reaction-moment of the hinges. Comparison to standard dimensions led to the choice of a 2"x2" aluminum box-section with a $\frac{1}{8}$" wall-thickness. To ensure perpendicularity to the hingeblock, the vertical walls of the box-section were cut back slightly, leaving contact-surfaces at the outmost material only. The beam mounts to the hingeblock on one side and the magnetic targets on the other, by pulling a threaded rod all the way through the center of the box-section. The length of this rod allows enough strain to pull these pieces together very tight.

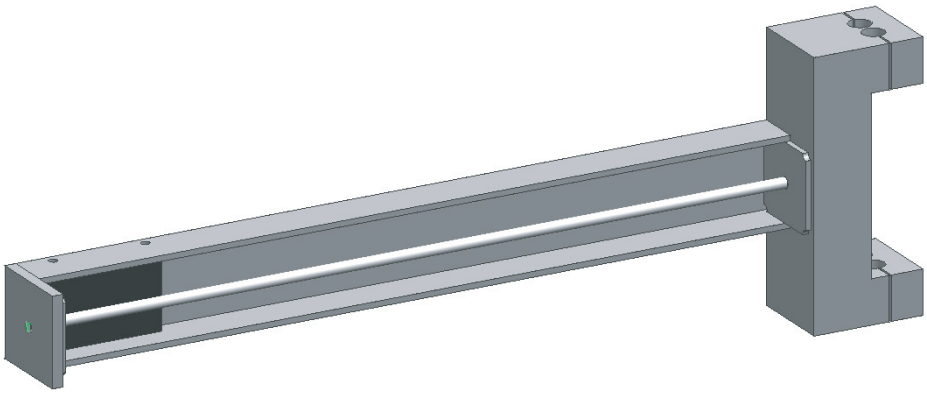


Figure 3.2: Section view of the beam with its mounting

3.3 Magnetic targets and their mounting

In order to close the flux-loop for an electromagnetic actuator, a magnetic target is required. Usually materials with high magnetic permeability are used, such as weak-iron. Enough volume of this material is required to prevent saturation[1]. When a volume of electrically conductive material travels through a magnetic field, eddy currents will initiate in the material. These currents are dissipated in the material causing heat and reducing effectiveness, also electromagnetic drag forces occur due to eddy currents. To reduce these drawbacks, laminated material can be used. In this case, the magnetic target consists of a stack of thin sheets. The sheets are coated, so that they only allow in-plane currents. By placing these laminations perpendicular to the magnetic fieldlines, eddy currents are reduced to a minimum (currents can only run over the thickness of the sheets) and with them, heat-dissipation and drag-forces. The laminations also have a disadvantage; a stack of thin sheets has a rough surface, causing the gap between the actuator and the target to be undefined. To counteract this, the laminations are mounted together, creating a solid block. Then, the relevant surfaced of the blocks are milled to a better quality. The material used, is dictated by the materials that were previously purchased; Orthosil M6 from the Thomas & Skinner company [2].

To reduce weight, two targets of 0,5" thickness are created and mounted within the beam. In order to do this, rectangular pockets are milled out of the vertical walls of the box-section. The end of the beam is closed with a plate thats slots into and over the box-section. This creates a closed box structure with high stiffness. The sides of the beam are milled and polished to provide a good measuring surface for the capacitive sensor. The force-measurement also attaches to the beam.

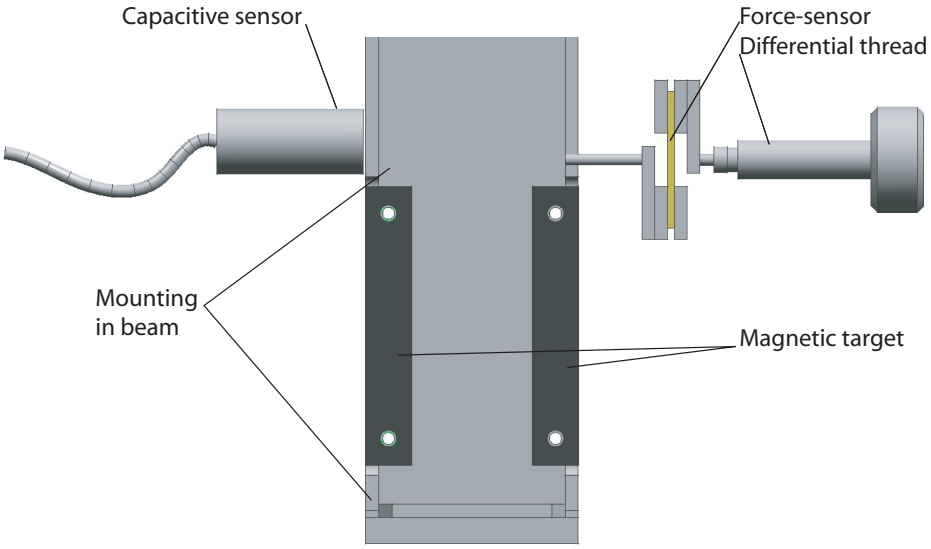


Figure 3.3: Magnetic targets with mounts, gap-sensor and force-sensor

[1]Saturation occurs when all dipoles in the material are oriented along the magnetic fieldlines, from this point on, the actuator-force cannot increase any further, reducing the effectiveness

3.4 Actuators and their mounting

Next, actuators are required on both sides of the beam. Currently, two different types of actuators are designed: the E-shaped actuator and the round actuator. The E-shaped actuator consists of a stack of E-laminations and thus has a laminated core, these laminations have been glued together to create a solid block and were then milled to a smooth surface. The round actuator has a solid soft-iron core.
The coils and bobbins that were used for the E-shaped actuators, are the same ones that were available from previous work [2]. For the round actuators, new bobbins were ordered to match the wire gauge and amount windings from the E-shaped actuators to make a fair comparison.
The attraction force between (electro)magnets and ferrous targets is caused by the gradient of the magnetic energy stored in the air gap, in the direction of motion; the fieldlines can pass through ferrous materials easier (higher μ_r), allowing them to shorten their pathlength. A shorter pathlength corresponds to a lower energy-state, which is wat nature strives for.

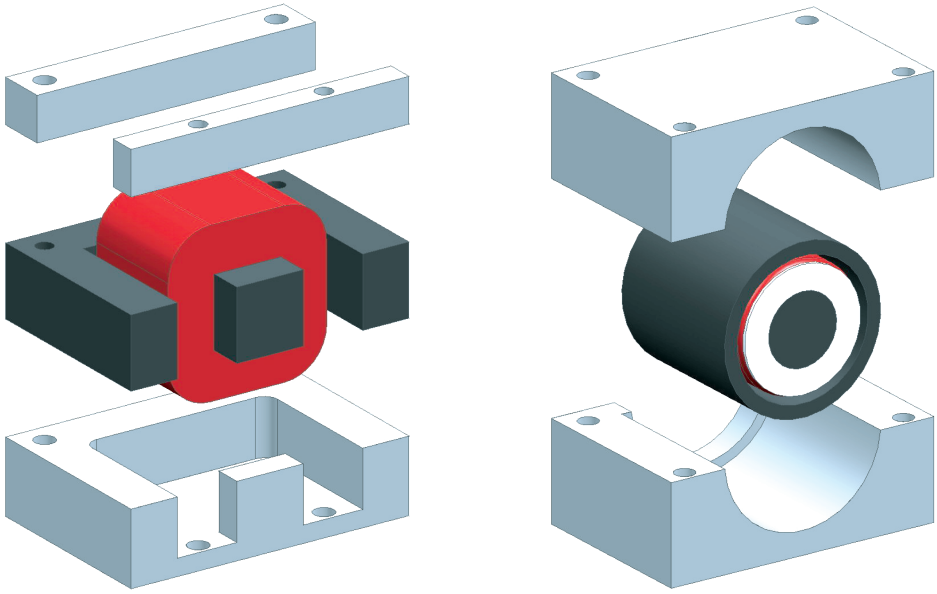


Figure 3.4: E-shaped and round actuator, both with clamping

Because different types of actuators are to be tested, one baseplate has been designed that allows mounting of different clamps for different actuators. Currently, clamps for two types of actuators have been designed but the baseplate allows different actuators in the future.

3.5 Force-measurement

As was said before, in order to characterize the actuators, the force needs to be measured as a result of the current through the actuators and the gap between the magnetic target and the actuator. To fill a grid of measurements points, a constant current is applied and the gap is varied, or vice versa. The constant current can be realized by applying a constant voltage to the amplifiers, but a constant gap is more difficult to realize. The force is measured using strain gages on a thin elastic element. The sensor used is a TBS-40 (thin beam sensor, 40 pounds ($\approx$ 180 [N]) see appendix B.1). This sensor allows inline loading, forcing the thin beam into a double bend (see figure 3.5). Naturally, this double bend causes inline elongation of the setup. In order to compensate for this, and return to the desired gap, a differential thread is designed with a pitch-difference of 0,058 [mm] built up out of an
M6 x 1,0 [mm] and a $\frac{3}{8}$" x 24 UNF ($\approx 1,058$ [mm]).

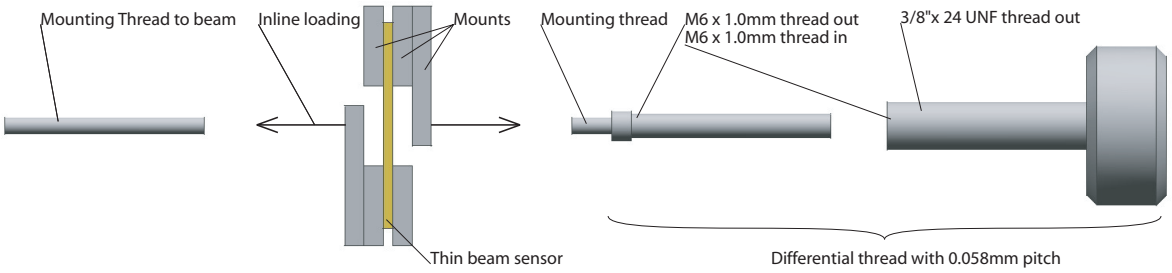


Figure 3.5: Exploded view of the TBS force sensor and the mounts

3.6 Supports

Finally, all the parts mentioned above need to be mounted together. For this a large jig-plate is squared up and patterned with holes. This type of plate has sufficient flatness. Additionally, a set of nine other supports have been designed out of $\frac{1}{2}$" aluminum plate. These supports provide threaded holes for mounting the capacitive sensor and the force-sensor. Also, alignment and spacing are adjustable through the play of the mounting holes. When all of this is assembled, the final design can be seen in figure **??**. Both different types of actuators are mounted in this figure.

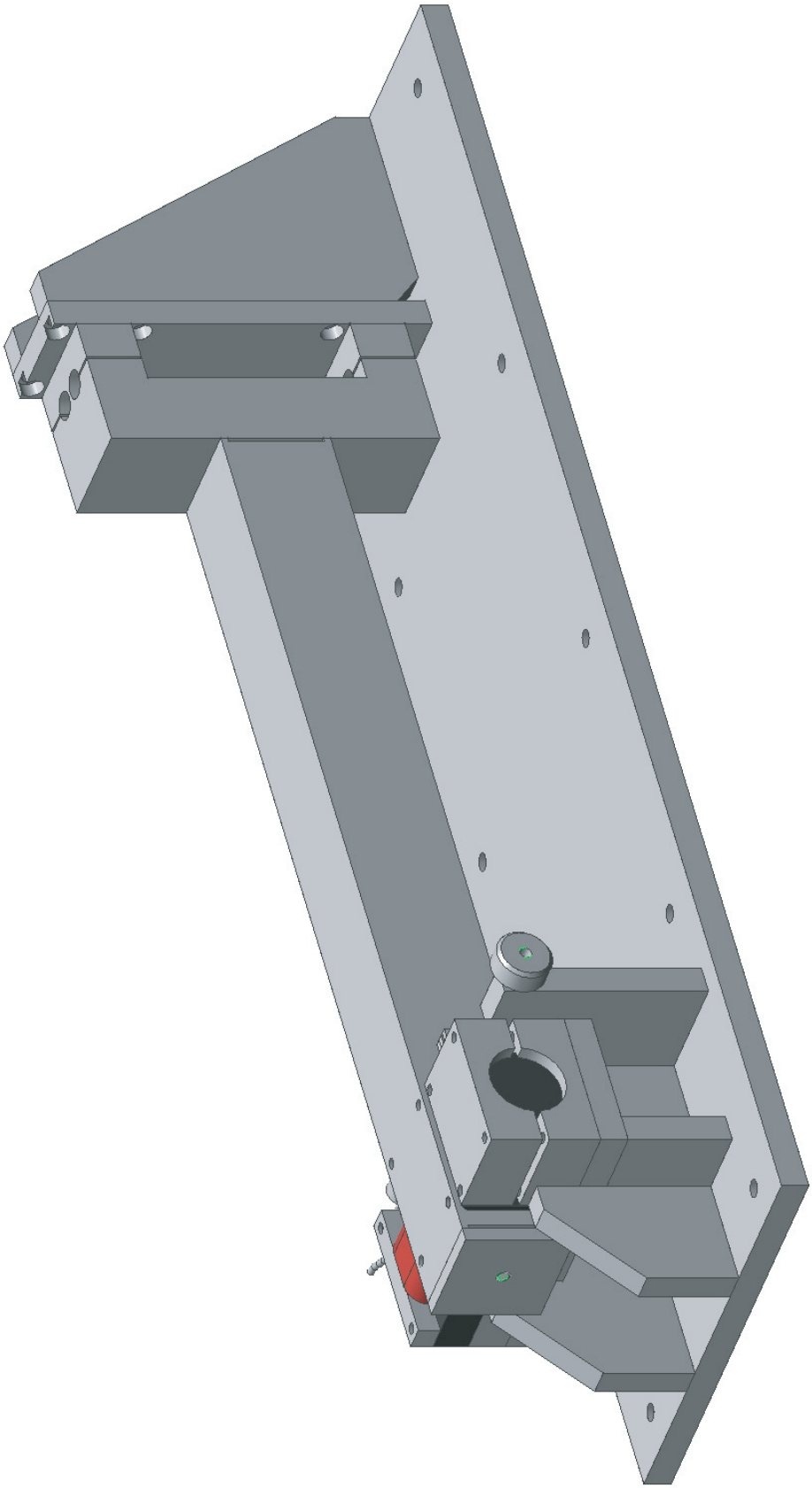


Figure 3.6: The assembled design

Chapter 4

Characterization

To control the system properly good force models of the actuators are very important. In this chapter first a theoretical model of the actuator is derived. Then this model is compared with a fit from experimental results.

4.1 Theoretical model

The theory presented in [11] and [13] has been used to derive a theoretical force model of the actuator. There are two different actuator types, the E-shaped and the round actuator. The following force model is used:

$$F = \frac{N^2 \cdot \mu_0}{2 \cdot A} \cdot \frac{I^2}{\left(\frac{R_c}{2 \cdot A \cdot \mu_r} + x \right)^2} \tag{4.1}$$

With:

F	= Force	$[N]$
I	= Current	$[A]$
N	= Number of coil windings	$[-]$
R_c	= Reluctance	$[A/Wb]$
H	= Magnetic field intensity	$[A/m]$
μ	= Permeability of a material ($\mu = \mu_r \cdot \mu_0$)	$[H/m]$
μ_0	= Permeability of free space, $4\pi \cdot 10^{-7}$	$[H/m]$
μ_r	= Relative permeability of a material	$[-]$
S	= Cross section actuator	$[m^2]$
A	= Surface	$[1/m^2]$

The dimensions of the actuators and numerical values of the parameters mentioned above can be found in appendix D.1.

The force model has been derived with the following assumptions:

- Flux losses are negligible. The flux paths pass only through the actuators, targets and airgaps,
- The metal core of the actuator and the target do not saturate,
- There are no eddy current losses,
- The airgap is small compared to the actuator dimensions.

With the m-file in appendix D.1, the force is computed for different airgaps and currents. The theoretical model plot can be seen in figure 4.1. The theoretical values are compared with the experimental values. In section 4.2 more is told about the empirical model. The differences between the theoretical and experimental values are discussed in this section. Flux losses and possible saturation may explain these differences. But the uncertainties in the parameters used in the theoretical model are more important. There are some geometrical uncertainties in the surfaces (A) and therefore in the the cross sections (S). But, more important are the coil windings and the relative permeability of the materials (μ_r). The force depends quadratically on the number of coil windings (equation 4.1). A change in the number of coil windings directly influences the force. A difference of 10 windings could result in a 6 [N] difference in the calculated force. The windings are also hand wound, during the winding process it was difficult to count the number of windings precisely. The relationship with the relative permeability is more difficult to see. A closer look at the term (from equation 4.1):

$$\left(\frac{R_c}{2 \cdot A \cdot \mu_r} + x\right)^2 \tag{4.2}$$

Learns that $\frac{R_c}{2 \cdot A \cdot \mu_r}$ is negligible with respect to x when the relative permeability of the material (μ_r) is above 20000. Therefore small deviations in the relative permeability of the laminated E-shaped actuator are not significant since the permeability is 47000. But, this is significant with the round, soft iron, actuator. The relative permeability of soft iron is around the 4000. So, small deviations in the permeability cause deviations in the calculated force, because $\frac{R_c}{2 \cdot A \cdot \mu_r}$ is no longer negligible.
Because of the uncertainties in the theoretical model, empirical values are desired to fit all the uncertain parameters into one constant.

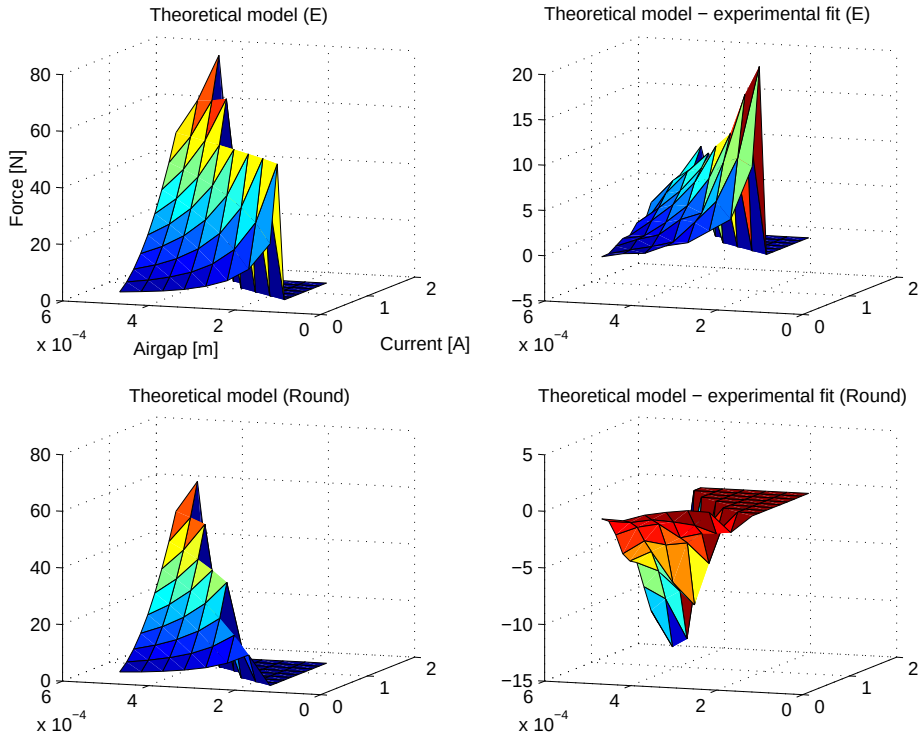


Figure 4.1: Theoretical values compared with experimental values

4.2 Empirical model

4.2.1 Hardware

As was said before, both the current and the airgap determine the force exerted by the actuator. Therefore, the current, airgap and force need to be measured to characterize the actuators. To do so, the following setup is made: First of all, the machine described in chapter 3 is bolted to an airsupported table. The actuator to be tested is mounted on one side of the machine, parallel to the hinged beam in such a manner that about 600 $[\mu m]$ of travel is possible. Next, the force-sensor with differential thread is attached to the beam to match the travel.

4.2.2 Electronics

In order to control and measure, some amplifiers, powersupplies and data-acquisition is required. This is schematically depicted in figure (4.2). To clarify things further and for future reference, all the components are described in detail. For pinouts, see appendix B.2.

- **PC with DAQ-card**
  The PC is fitted with a PCMCIA to PCI board so the DAQ-card can be used. From here, Matlab, labview or C-code can be used to send and receive signals to the TU/e DACS QAD box [9].

- **TU/e DACS QAD**
  The TU/e DACS QAD is a four channel data acquisition unit. It communicates with the PC through a PCMCIA DAQ-card. The QAD is powered by a 9 [V] DC adapter and has two analog-digital (ADC) channels and two digital-analog (DAC) channels. The two ADC channels are used to measure the force and the gap, one DAC-channel is used to control the current through the actuator, the other DAC-channel controls the current through the second actuator during control. All the channels are fitted with BNC-connectors, all grounded to the machine-frame. All the channels range from -2,5 [V] to +2,5 [V]

- **HP Amplifier**
  Because both the force- and gap-signal are unsuited to be read by the QAD directly, they are treated in the HP amplifier. Firstly, the force-signal is too weak (20 [mV] corresponds to $\approx$ 180 [N]) and noisy. This signal is amplified forty times and most noise is removed by the bandwidth of the amplifier (acts as a low-pass filter). Secondly, the signal from the Lion Precision acquisition box range from -10 [V] to +10 [V]. This signal is divided by five to meet the range of the QAD.

- **TBS-40 Force-sensor**
  The force-sensor uses straingages on one side of the bending element. With the mounting kit (TBS-MK2), the elements goes into a double bend when loaded; the straingages then measure both compression and tension. The sensors need to be exited with 10 [V], that has to be created with a -5 [V] and a +5 [V]. The sensor then generates a signal of 2 [mV/V], corresponding to a 20 [mV] signal for full range ($\approx$ 180 [N]). The sensor is mounted to the hinged beam a with threaded rod, it mounts to the machine-frame through a differential thread.

- **Power supply for TBS-40**
  This power supply generates a -5 [V] and +5 [V] for powering the TBS-40 sensor. The power supply is grounded by its 110 [V] powercable. The ground output by the power supply is not used; the 10 [V] are floating. This is done to prevent the disturbances in the outlet-ground from affecting the force-signal.

- **Capacitive gap-sensor**
  The gap-sensor from Lion Precision [2] measures the capacity between its own frontal surface and the target surface. The air between the two surfaces acts as the insulating material. Only the center ring of the sensor actually measures. The sensor threads into the machine-frame and allows adjusting and locking with an extra nut. The sensor is powered and read out by a dedicated acquisition unit from Lion Precision. All the available sensors are calibrated to be used with one specific channel of the acquisition unit. When adjusting or replacing the sensor, it is important to use a sensor with the proper range (500-1500 [μm]) and to make sure that the acquisition unit's indicator lights stay in the green throughout the entire range.

- **Lion Precision acquisition unit**
  As was said, the sensor should be hooked up to the right channel. Next, the unit needs to be set to low accuracy to measure the entire range; the adjusting knobs can be used to set the equilibrium point to zero. Furthermore, each channel needs to be individually grounded to the machine-frame.

- **Transformer for voltage-current amplifiers**
  The voltage-current amplifiers require a 165 [V] DC powersupply. To create this from 110 [V] AC, a transformer is used. This transformer generates upto 16 channels for as many amplifiers. Switching on this transformer generates quite a bit of noise on the local ground at the wall-outlets. Therefore, the measuring circuits are all grounded to each other and the machine-frame, but not to the ground from the wall-outlets. When the amplifiers are not working as expected, the fuses in this transformer are to be checked first.

- **Voltage-current amplifiers**
  To control the current through the actuator-coils, voltage-controlled amplifiers are used. The amplifiers are powered at 165 [V] DC by the transformer and creates a current ranging from 0-3 [A] at 165 [V]. The control signal can range from -10 [V] to +10 [V] DC or PWM. Three adjusting screws on the boards allow changing the current-offset, the gain and the frequency. Earlier research has shown thermal drift on these actuators. This may be counteracted in future by on-board feedback with a current-sensor.

- **Current filter**
  Each filter-board can filter the currents from two amplifiers. This reduces the PWM-noise introduced by the amplifiers.

- **Actuators**
  After the filters, two wires lead the current through the actuator-coils. One should not attach or detach these wires when the transformer is switched on. Also, attention has to be paid to the condition of the coil-wire coating to prevent short-circuits.

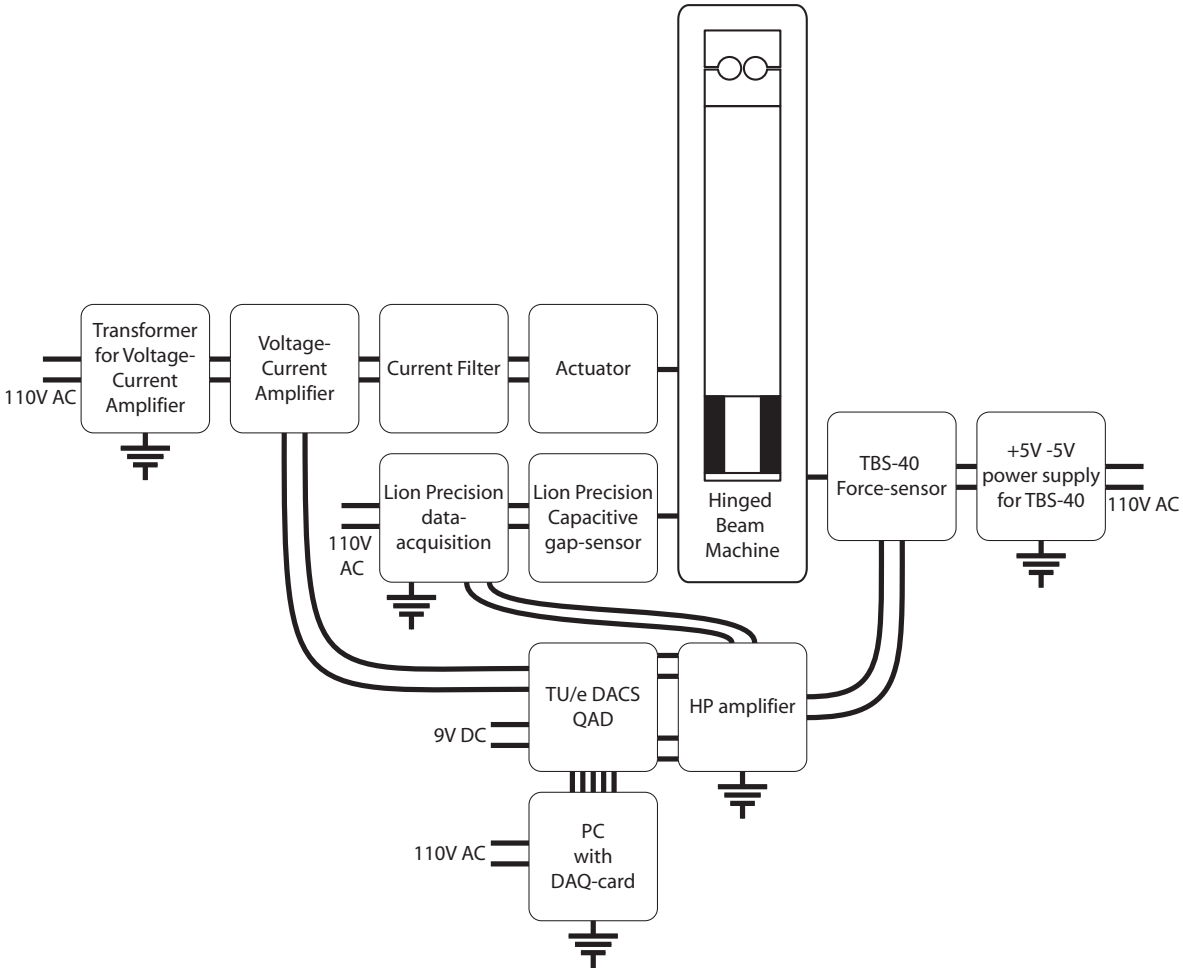


Figure 4.2: Electrical scheme

4.2.3 Simulink setup

The setup described in the section above, allows direct communication between Matlab/Simulink and the machine. In order to do so, a model has been built (figure 4.3). The main block in the model is the TUeDACS/1 QAD/AQ1 interface, this block sends and receives all data to and from the QAD-box. Only the two ADC and two DAC channels are used. A slider gain (i_1 [A]) allows the current to be increased from 0-1,5 [A]. This has to be multiplied by $\frac{5}{3}$ to come to 0-2,5 [V] that will be sent to the voltage-current-amplifier. This signal is also exported to workspace for later use. The force-sensor is hooked up to ADC1; the signal it receives is converted to Newtons [N] by multiplying it with a gain ($177,93[N] \approx 40$ [lbs], $0,8 = 2$ [mV/V] $\times$ 40 (HP AMP)) . Next a slider gain allows the zero-offset to be corrected, a scope allows online monitoring and the signal is also exported to workspace. Finally, the gap-signal is hooked up to ADC2, converted to [μm], adjusted for zero-offset and adjusted to place the equilibrium in the center of the full range (540 [μm]). This signal can also be monitored on the scope and exported to workspace.

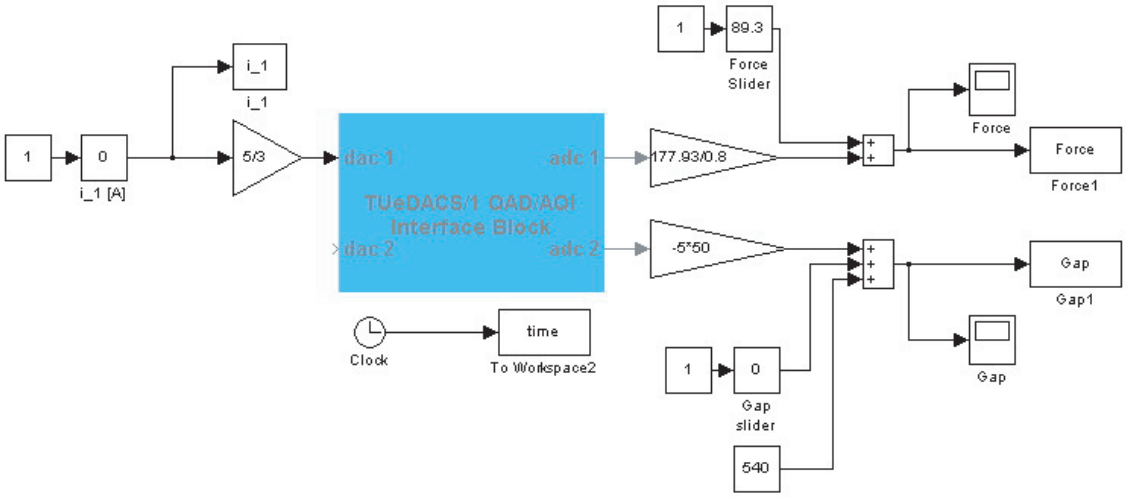


Figure 4.3: Simulink model for force-measurement

4.2.4 Results

Strategy

In order to characterize the actuators in the relevant region, a grid of measuring points was created. The current varies between 0,3-1,5 [A] with 0,15 [A] intervals, the airgap varies between 150-500 [μm] with 50 [μm] intervals. The gridpoints were measured by applying a certain current and then adjusting the gap from 500+ [μm] down to 500 [μm] and down in 50 [μm] steps from there, using the differential thread. This strategy was used at increasing currents. It is not necessary to measure the gridpoints approaching the gap from below rather than above, because there is no hysteresis in the hinge. At higher currents and smaller gaps, the actuator-force becomes larger than the elastic force of the force-sensor, that the beam snaps to the actuator and sticks to it. This is why some of the more 'extreme' points could not be measured.

Model fit

After all the gridpoints were measured for both the E-type and round actuators, two different models were fitted to the measurements.

$$F = K \cdot \frac{I^2}{g^2} \tag{4.3}$$

$$F = K \cdot \frac{I^2}{(g + x_{ad})^2} \tag{4.4}$$

with:

K	= Force constant	$[N \cdot \frac{m^2}{A^2}]$
I	= Current	$[A]$
g	= Airgap	$[m]$
x_{ad}	= Additional gaplength	$[m]$

The first model (equation 4.3) only has one parameter (K) to be fitted; this is done by using a least-squares method, yielding the values listed in table 4.1. The second model is elaborated with an additional gap x_{ad} [1]. Now, two parameters have to fitted to the measured data. A least-squares criterium only yields one equation in K and x_{ad} to be minimized. In order to solve this, the sum of the squared errors is plotted as a function of K an x_{ad} to find its minimum. The M-files used for the fits and the plots are available in appendix D.2. These values can also be found in table 4.1. The values for the actuators acting on a solid target can be found in tabel 4.2. The measurements from both the actuators with a solid, soft iron, target can be found in appendix C.2.

[1] Due to the finite reluctance of the actuator-core (see [11])

	K (E-type)	x_{ad} (E-type)	K (round)	x_{ad} (round)
Model 1	$4{,}5963 \cdot 10^{-6}$	-	$3{,}9853 \cdot 10^{-6}$	-
Model 2	$7{,}95 \cdot 10^{-6}$	$93 \cdot 10^{-6}$	$10{,}60 \cdot 10^{-6}$	$86 \cdot 10^{-6}$

Table 4.1: K and x_{ad} fitted for both models on both actuators with laminated target

	K (E-type)	x_{ad} (E-type)	K (round)	x_{ad} (round)
Model 2	$7{,}2 \cdot 10^{-6}$	$220 \cdot 10^{-6}$	$6{,}00 \cdot 10^{-6}$	$170 \cdot 10^{-6}$

Table 4.2: K and x_{ad} fitted for both models on both actuators with solid target

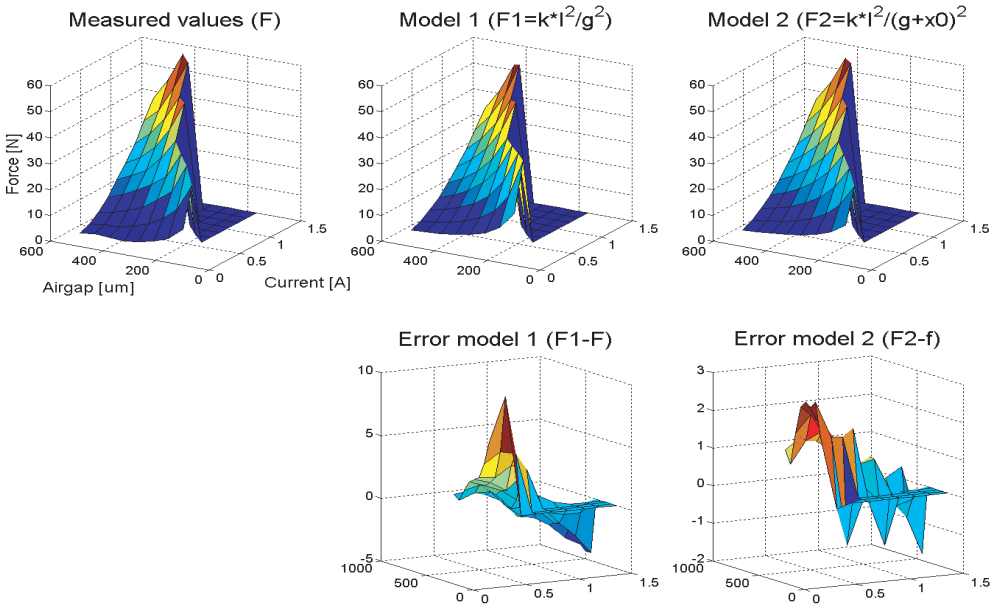


Figure 4.4: E-type actuator with laminated target

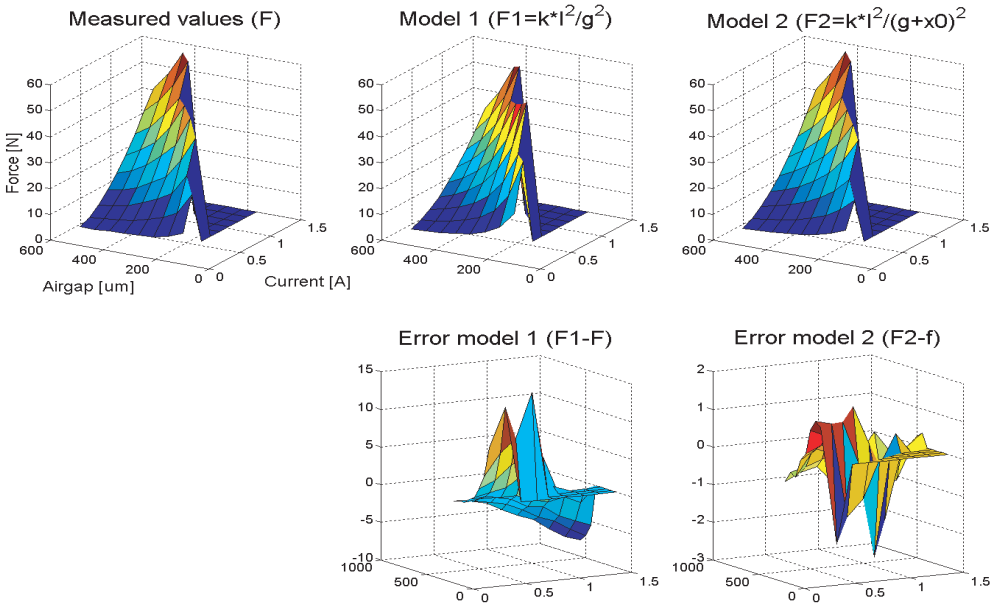


Figure 4.5: Round actuator with laminated target

Finally, an experiment was done to measure the eigenfrequency and damping of the system. The system was released from the extreme left position to vibrate about its equilibrium, results of this can be found in figure 4.6. Counting periods over time showed that the eigenfrequency $\omega_n = 9,5$ [Hz] which is very close to the designed value of 9 [Hz]. Fitting a curve through the minimal values of the vibration, and using equation 4.6 yields the damping-coefficient of $\xi = 0,05$.

$$y = e^{-\omega_n \xi t} \tag{4.5}$$

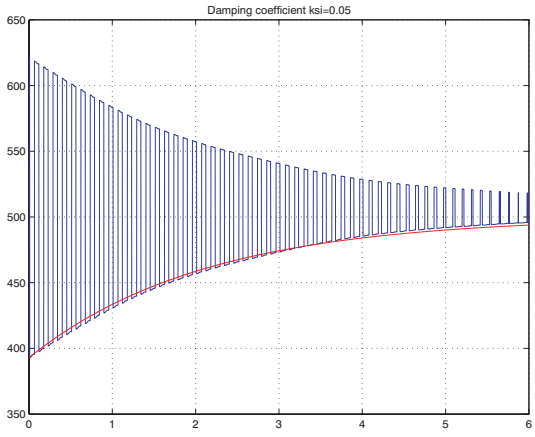


Figure 4.6: Measurement for determining ω_n and ξ

Chapter 5

Control of the 1DOF system

Before using the electromagnetic actuators in a precision machine design, a better understanding of the lay-out and control of electromagnetic actuators is necessary. In the previous chapter, the different actuators have been characterized and the force constants are derived. In this chapter the force constants obtained with the static tests are used to test different control strategies. Both linear and non-linear controllers will be designed, simulated and implemented on the 1DOF system.

A new 1DOF setup has been designed as described in chapter 3. Due to the elastic hinge point, the model of the system is very clean; hysteresis, play and friction are eliminated.

5.1 System Description

The system consists of a beam with one elastic DOF. The system is actuated by two opposing magnetic actuators. The magnetic actuators are described using the force model derived in chapter 4. The distance from the hinge point to the point of actuation is large in comparison to the horizontal travel of the beam. The angular rotation is very small, therefore the system is considered to be a translating mass. The effective mass of the endpoint is determined by computing the rotational moment of inertia of the shaft and the magnetic target at the end, and then dividing this value by the square of the distance from the center of actuation to the rotation axis.

The following equation of motion can be derived:

$$m \cdot \ddot{x} = K_1 \cdot \frac{I_1^2}{g_1^2} - K_2 \cdot \frac{I_2^2}{g_2^2} - K_f \cdot x \tag{5.1}$$

The gaps are defined as:

$$g_1 = x_{01} + x_{ad} + x \tag{5.2}$$

$$g_2 = x_{02} + x_{ad} - x \tag{5.3}$$

With:

x	= Translation beam	$[m]$
x_0	= Initial position of the beam	$[m]$
x_{01}, x_{02}	= Initial distance between target and actuator	$[m]$
K_1, K_2	= Force constants	$[N \cdot m^2/A^2]$
x_{ad}	= Additional gap length	$[m]$
K_f	= Flexure stiffness	$[N/m]$
I_1, I_2	= Currents coil 1 and 2	$[A]$
CM	= Center of mass	$[-]$
m	= Effective mass of the endpoint	$[kg]$

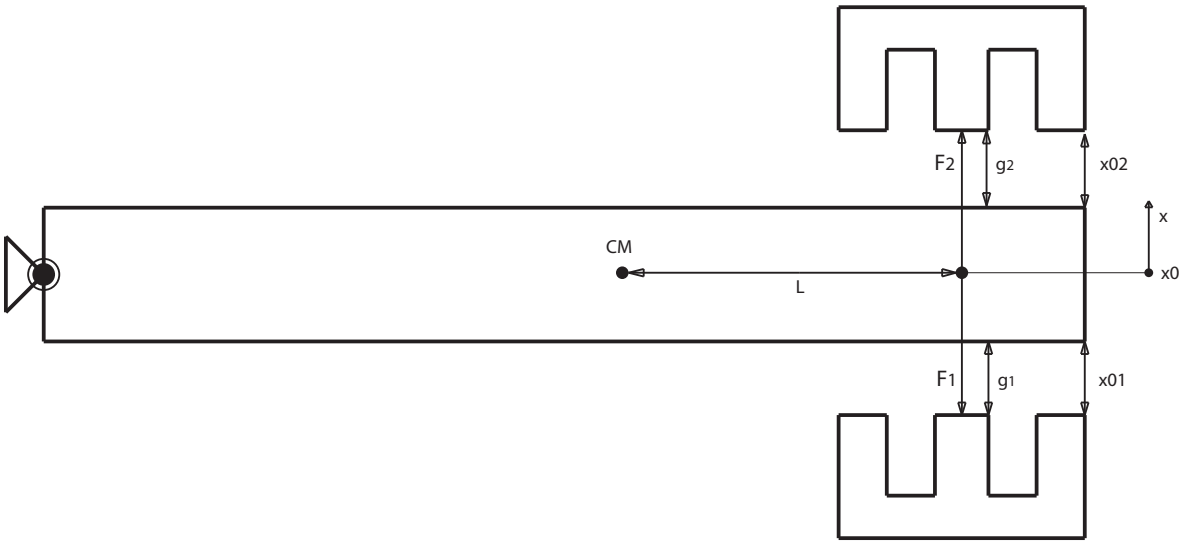


Figure 5.1: Schematic system overview

5.2 Linear control

In this section a linear controller is used to stabilize and control the system. First the system equations are linearized, and a stabilizing controller is found. FRF-measurements are used for loopshaping.

5.2.1 Linearization

A bias current is used to make the system a SISO-system. Both currents, I_1 and I_2, are made bias currents with the same value. The bias current is applied to both coils when the beam is in initial position. In this situation both actuators are attracting the beam with the same force. Now, a control current is introduced. ΔI is the control current. The bias current is successively raised ore lowered with the control current.

$$I_1 = I_b + \Delta I \tag{5.4}$$

$$I_2 = I_b - \Delta I \tag{5.5}$$

The bias current must be greater then the control current, because a current flowing through the coil, positive or negative, will always result in a tractive force. The bias current is chosen to be 0,3 [A]. This will result in the following state space notation, with state variables x (position) and $\dot{x}$ (velocity):

$$\underline{\dot{x}} = \begin{bmatrix} x_2 \\ \frac{K_1}{m} \cdot \frac{(I_b+\Delta I)^2}{g_1^2(x_1)} - \frac{K_2}{m} \cdot \frac{(I_b-\Delta I)^2}{g_2^2(x_1)} - \frac{K_f}{m} \cdot x_1 \end{bmatrix} \tag{5.6}$$

Linearization around the operating point, $x_1 = 0$ and $x_2 = 0$, leads to the following linearized system:

$$\underline{A} = \begin{bmatrix} 0 & 1 \\ \frac{-K_1}{m} \cdot \frac{2 \cdot I_b^2}{(x_0+x_{ad})^3} - \frac{K_2}{m} \cdot \frac{2 \cdot I_b^2}{(x_0+x_{ad})^3} - \frac{K_f}{m} & 0 \end{bmatrix} \tag{5.7}$$

$$\underline{B} = \begin{bmatrix} 0 \\ \frac{K_1}{m} \cdot \frac{2 \cdot I_b}{(x_0+x_{ad})^2} + \frac{K_2}{m} \cdot \frac{2 \cdot I_b}{(x_0+x_{ad})^2} \end{bmatrix} \tag{5.8}$$

$$\underline{C} = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{5.9}$$

$$\underline{D} = \begin{bmatrix} 0 \end{bmatrix} \tag{5.10}$$

With the assumption that both actuators are equal ($K_1 = K_2$) and the bias current is the same in both coils the eigenvalues of the linearized system are:

$$\lambda = \pm\sqrt{4 \cdot \frac{K_1}{m} \cdot \frac{I_b^2}{(x_0 + x_{ad})^3} + \frac{K_f}{m}} \tag{5.11}$$

The eigenvalues of the system depend on the bias current and the force constants of the actuators. The force constants could only change with the use of different actuators. The eigenfrequency can be altered by adjusting the bias current.

5.2.2 System identification

A simple PID controller is used to stabilize the system around the operating point. Due to the flexure, the beams initial position is in the middle of the horizontal range, which made the stabilization around the operating point easier. Once a linear controller is found that stabilizes the system, the sensitivity can be measured. To measure the frequency response, white noise is injected into the system. In figure 5.2 the transfer function of the plant, derived by measuring the sensitivity, can be seen. The dotted line shows the bode plot derived from the linearized system. And the solid line shows the bode plot derived from measuring the sensitivity, while injecting white noise. The bias current was set to 0,3 [A]. According to equation 5.11 one can see that the eigenfrequency increases when the bias current increases. Errors in the measurements and uncertainties in the force constants could explain the difference between the two peaks. In general the measurements match the bode plot obtained with the linearized model.

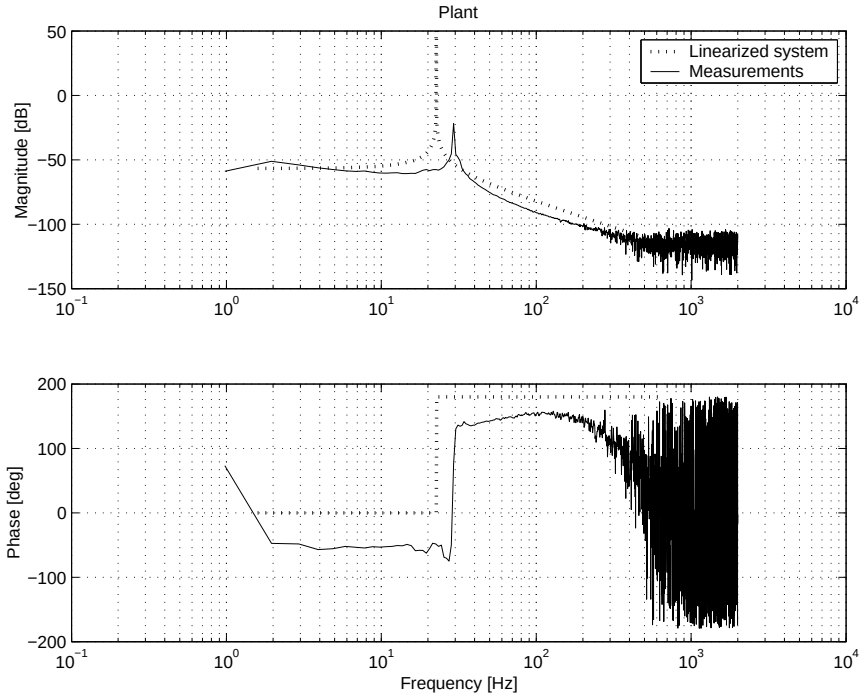


Figure 5.2: Bode plot of linearized system and real system

5.2.3 Simulations and real time results

Since the transfer function is known, a linear controller can be designed with DIET. The controller is analyzed in simulations and implemented on the real system. The controller consists of a gain, a lead/lag filter and an integrator. Despite the nonlinearity of the system, the controller is capable of stabilizing the system around the operating point, tracking a sine wave and performing smooth steps. In simulation the controller is suited for the whole operating range, although the performance in real time is not good. This can be seen in figures 5.3 and 5.4 where the system is tracking a sine wave in simulation and in real time. One can see the bias current and the additional control current. In reality the actuators are not exactly the same, therefore the control current is not symmetrical around zero in the experiments. During the experiments, the non-linearities make the system unstable far away from the linearization point, which can be seen in figure 5.5. The integrator brings the steady state error to 6 [μm]. The steady state error is worse and the overshoot is more then 100 [μm] when performing the 300 [μm] step. The amplifiers are using PWM modulation to convert the voltage (input) to current (output). This causes the high frequency noise in the control currents and the position measurement.

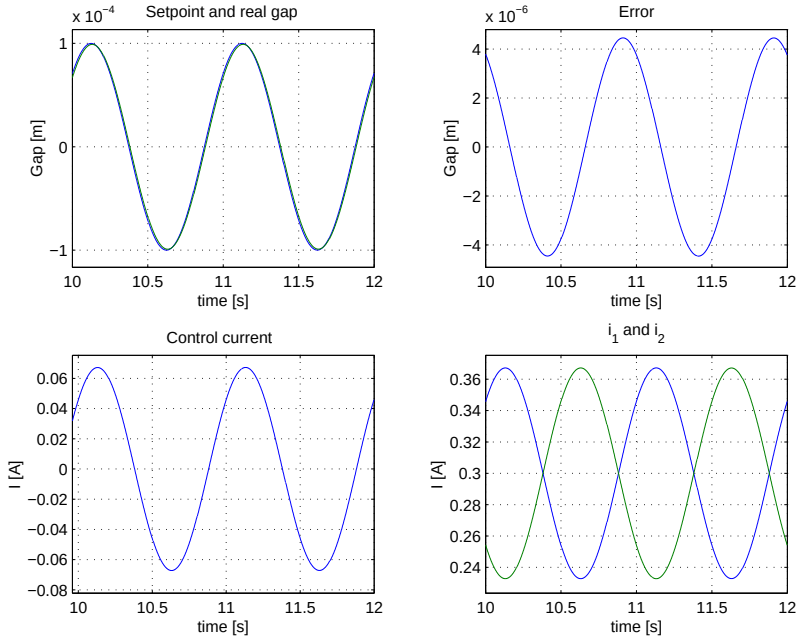


Figure 5.3: Tracking of 1 [Hz] sinusoidal, amplitude 100 [μm] (simulation)

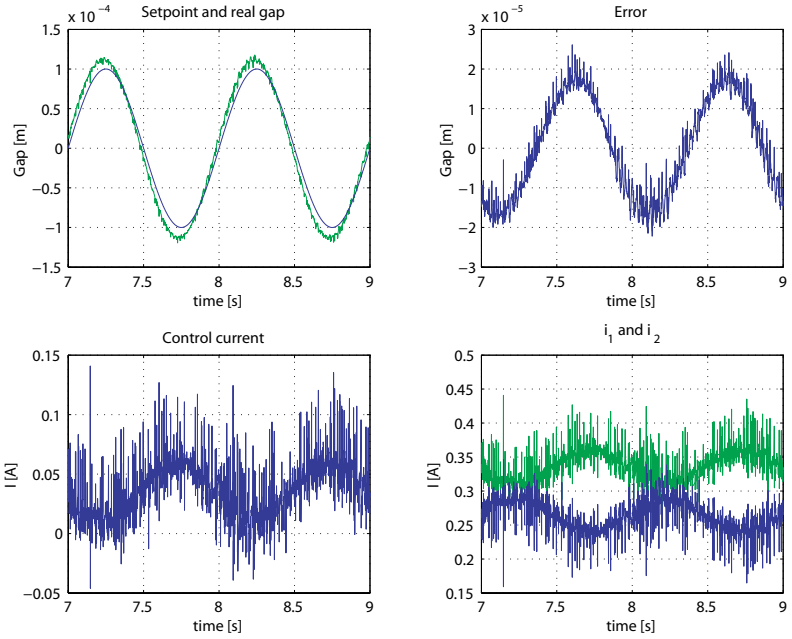


Figure 5.4: Tracking of 1 [Hz] sinusoidal, amplitude 100 [μm] (real time)

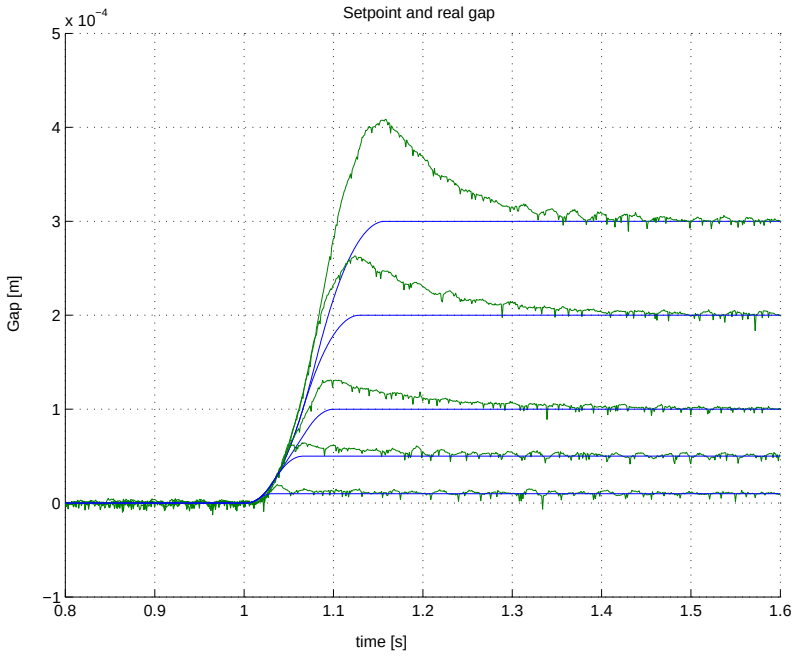


Figure 5.5: Stepresponse

5.3 The amplifiers

The amplifiers convert an input voltage with PWM modulation to a current. There are three adjustment screws on the amplifier board, respectively used for zero, gain and frequency adjustment. A schematic drawing of the connections between amplifier (with or without feedback), filter and power supply can be seen in appendix B.2. The latter, for the frequency adjustment, is not used. The gain offset determines the input (voltage) to output (current) multiplication factor. The TU/eDACS can give out 2,5 [V], in the experiments the maximal current used is 1,5 [A]. So the gain is set to a factor of $\frac{5}{3}$. Figure 5.6 shows that the voltage-ampere relationship can be described with a linear function. During the static experiments the amplifiers sometimes had to deliver 1,5 [A] for a few minutes; no significant heating or drift problems occurred.

5.3.1 Feedback

The tracking performance, when tracking sine waves with frequencies higher then 9 [Hz], is very poor. After doing a quick experiment, it seemed that the amplifiers could not change the current through the coils fast enough. A feedback loop is added to the amplifiers, in which the output current is measured with a current sensor type NT15 (appendix B.3) and the input voltage is controlled to meet the desired output. The electrical scheme of the feedback loop can be seen in appendix B.4. Tests were performed with and without the feedback. During the tests the signal, named "control signal" is sent by the TU/eDACS, this is converted from voltage to desired current with a multiplication in Simulink. The "measurement" is the current through the coil measured with another current sensor. This measured signal should be the same as the control signal. Figure 5.7 shows that the amplifiers perform better without the feedback. The PWM modulation causes noise, and the noise is fed back into the control signal because of the feedbackloop. This, of course, also causes noise in the measured current through the coils. According to this results there feedback is no longer used on the amplifiers. In both subplots the non-linearity of the amplifiers can be seen. The non-linear bends occur especially in a lower current range. There is no compensation for the non-linearity, this is recommended for further error reduction.

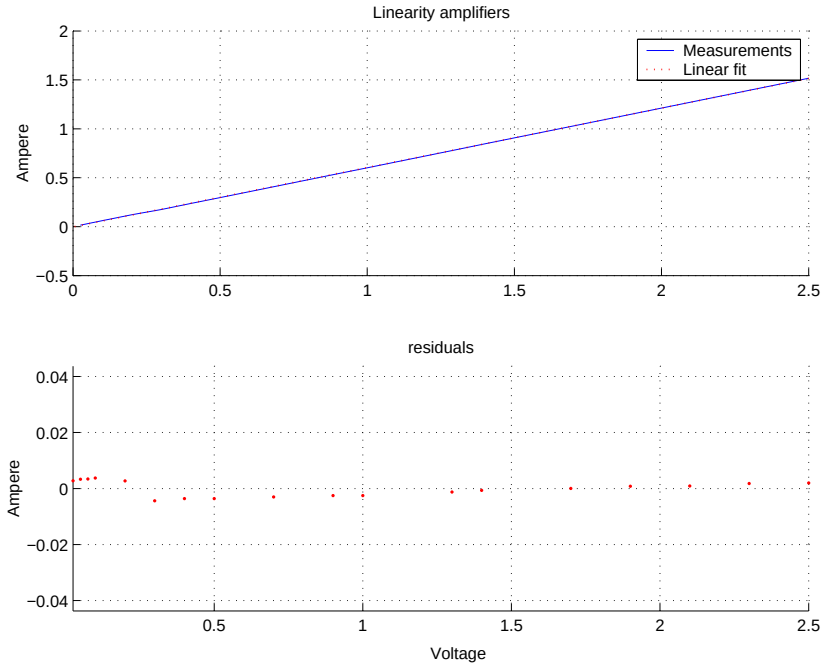


Figure 5.6: Linearity amplifiers

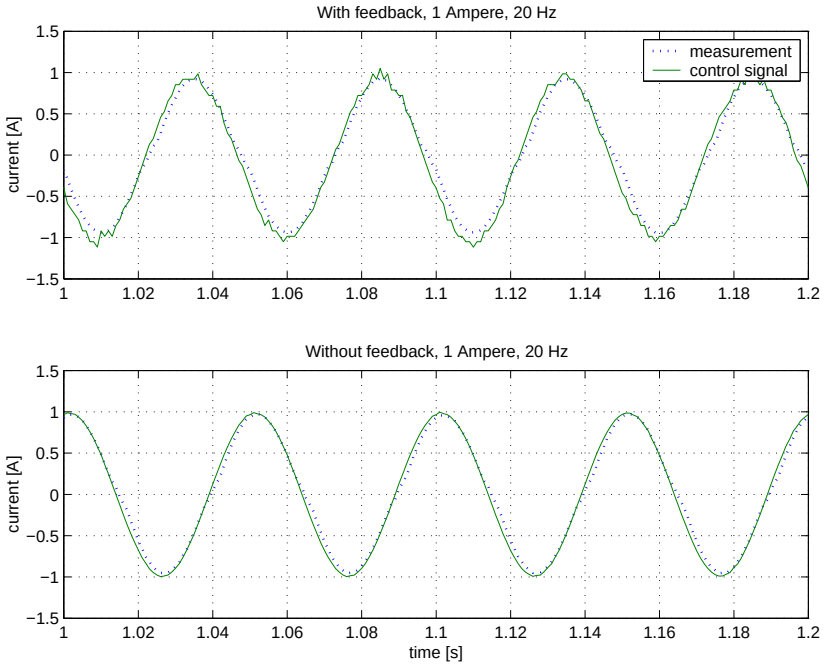


Figure 5.7: Amplifiers tracking a 20 [Hz] sine wave with and without feedback

5.4 Non-linear control

The linear controller, described in chapter 5.2, only performs well in a small operating range around the point of linearization. Literature shows that a feedback linearized system could work in the whole operation range, but will be limited by the accuracy of the model (feedback linearization assumes that the model is accurate and is therefore very sensitive to unmodeled dynamics and parametric errors). Several different control strategies were used to find the best control strategy, the strategies are compared in simulations and in real time. First a linear controller and a LQR controller were implemented on the feedback linearized system. Then two approaches of sliding mode control were implemented on the system. The non-affine sliding mode controller led to the best results and is described in subsection 5.4.5.

5.4.1 Feedback linearization

The goal of the feedback linearization is to linearize the system in terms of the new input v:

$$\underline{\dot{x}} = \begin{bmatrix} x_2 \\ v \end{bmatrix} \tag{5.12}$$

Using equation 5.6 the following transformation linearizes the system:

$$I_1 = g_1(x_1) \cdot \sqrt{\frac{m \cdot v + K_f \cdot x_1}{K_1}} \qquad \text{and:} \quad I_2 = 0 \qquad \text{for:} \quad (m \cdot v + K_f \cdot x_1) < 0 \tag{5.13}$$

$$I_2 = g_2(x_1) \cdot \sqrt{\frac{-m \cdot v + K_f \cdot x_1}{K_2}} \qquad \text{and:} \quad I_1 = 0 \qquad \text{for:} \quad (m \cdot v + K_f \cdot x_1) > 0 \tag{5.14}$$

When using this transformation it is impossible to use both actuators at the same time.

5.4.2 Feedback linearization with a linear controller

The feedback linearized system is controlled with, successively a linear and a LQR controller. The linear controller is roughly the same controller as used to control the non-linear system, the only difference is a higher gain. The linear controller in combination with the feedback linearized system leads to better tracking, with smaller errors. Although, the system has to be tuned for one specific step or sine wave. With this tuning, the error while tracking a sine wave is four times smaller then the error of the linearized system. When the amplitude of the step is changed, the error increases rapidly. This can be seen clearly in figure 5.10. In figures 5.8 and 5.9, one can see the control strategy; one actuator is active and the other is not. The system is tracking a symmetrical sine wave, so the two currents should be the same. Due to model errors and small uncertainties in the placement of the actuators with respect to the beam this is not the case. These errors also make it difficult to stabilize the system in the whole operating range.

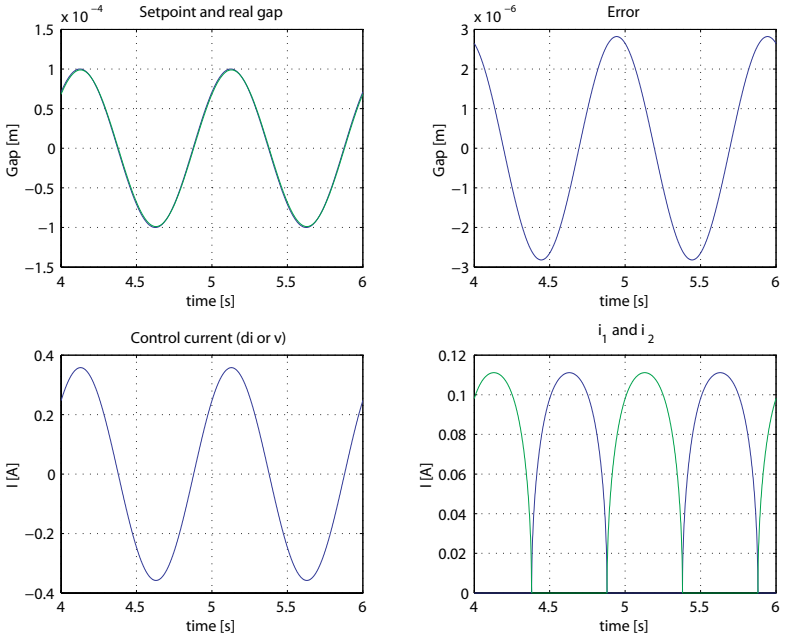


Figure 5.8: Tracking of 1 [Hz] sinusoidal, amplitude 100 [μm] (simulation, PID)

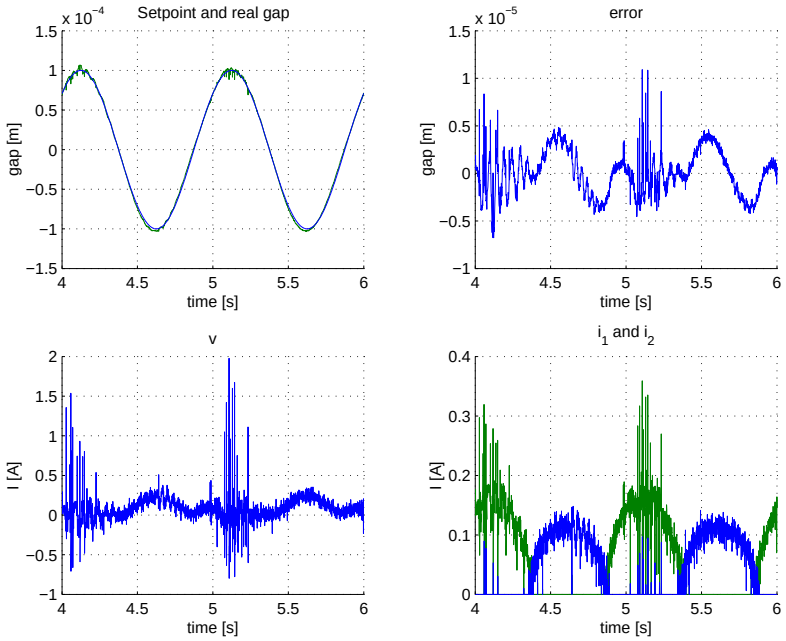


Figure 5.9: Tracking of 1 [Hz] sinusoidal, amplitude 100 [μm] (real time, PID)

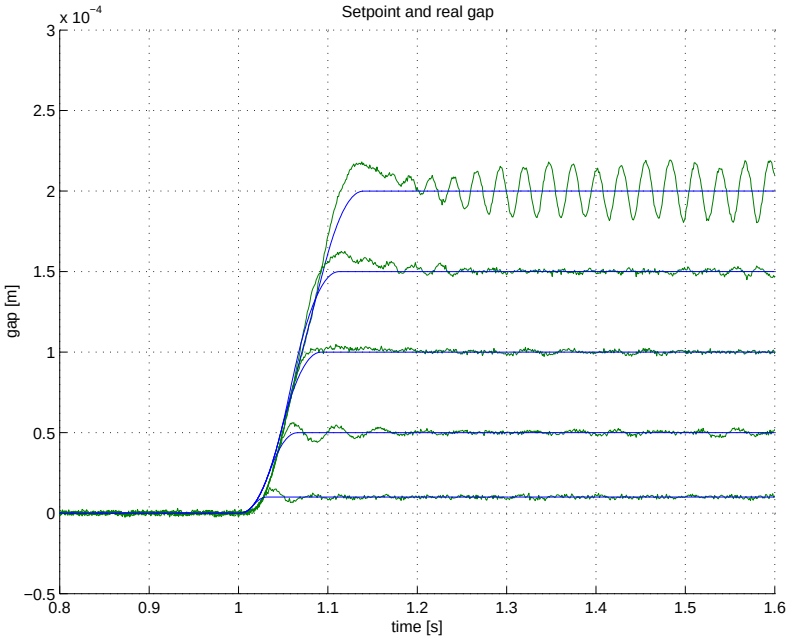


Figure 5.10: Stepresponse, tuned for 100 [μm] step (PID)

5.4.3 Feedback linearization with a Linear Quadratic Regulator

Theory

The linear quadratic regulator (LQR) is the basis of some powerful design methods such as the LQG and H∞ [4]. The LQR is a least squares optimization problem, that has some attractive advantages, namely, guaranteed closed loop stability, robustness and it is simple to compute. The LQR algorithm computes the optimal feedback gain matrix K that allows the feedback law $U_k = -K \cdot X_k$ to minimize the following cost function:

$$J = \frac{1}{2} \sum_{0}^{\infty} \qquad (X_k^T \cdot Q \cdot X_k + U_k^T \cdot R \cdot U_k) \tag{5.15}$$

$X_{k+1} = A_d \cdot X_k + B_d \cdot U_k$ is the constraint information given by the system's dynamics. A_d and B_d are the discrete plant matrices, X_k and U_k are the vectors of states and inputs, respectively. Q and R are symmetric, positive semi-definite matrices that allow to penalize the inputs or states that will be minimized by K.

To achieve better performance with this controller the integral of the tracking error is added as an additional state. Now the integral of the error could be penalized. The augmented system equations are:

$$\frac{d}{dt} \begin{bmatrix} x \\ \dot{x} \\ \int (x-r)dt \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -K/m & -d/m & 0 \\ 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ \dot{x} \\ \int (x-r)dt \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} v \\ r \end{bmatrix} \tag{5.16}$$

Where r is the reference command and v is the output of the linear controller. The Matlab script used for the computation of the gain- and observer-matrix can be found in appendix D.4.

Real time results

The LQR is implemented on the feedback linearized system in simulation and real-time. The results can be seen in figures 5.11 and 5.12. The LQR algorithm has not been optimized. Both the PID controller and the LQR in combination with the feedback linearized system have not led to the desired improvements, in tracking performance, with respect to the linear controller in combination with the linearized system. The feedback linearized system is very sensitive for model uncertainties and unmodelled dynamics. In simulations the controllers in combination with the feedback linearized system perform much better, because there are no model imperfections and therefore the gain can be increased without stability problems. The LQR controller was not perfected due to lack of time; figures 5.11 and 5.12 clearly show that the LQR does not perform very well in this stage of tuning.

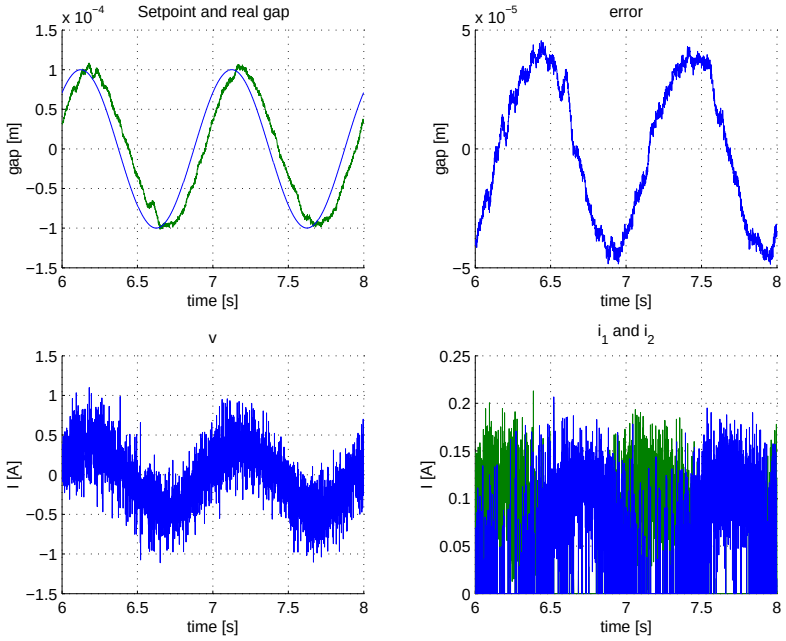


Figure 5.11: Tracking of 1 [Hz] sinusoidal, amplitude 100 [μm] (real time, LQR)

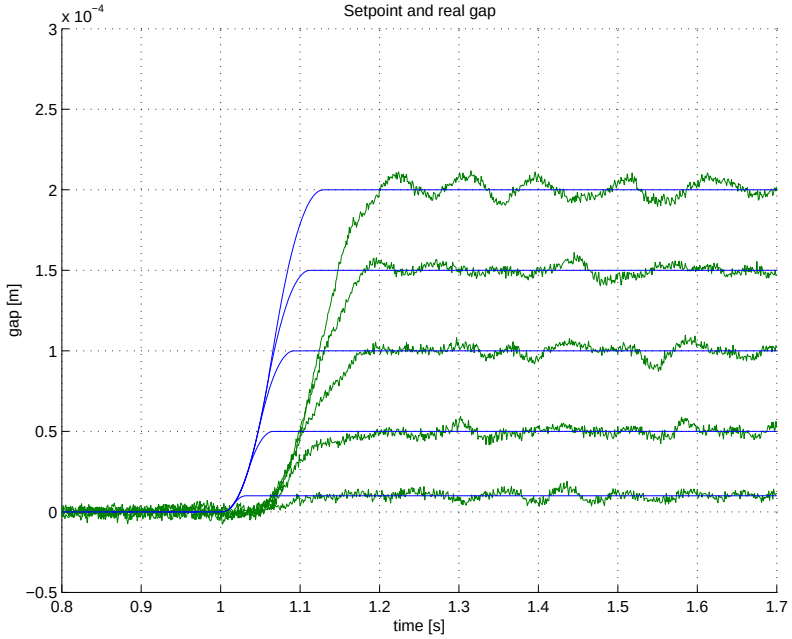


Figure 5.12: Stepresponse (LQR)

5.4.4 System identification toolbox

Once a stabilizing controller with feedback linearization was found, it was possible to use the Matlab system identification toolbox to verify the values for K and x_{ad} derived from the empirical force model (section 4.2). The command "PEM" is used. PEM computes the prediction error estimate of a general linear model. The model can be interpreted to be linear because of the feedback linearization. During the experiments white noise was injected into the system and the two input currets, I_1 and I_2, and the airgap, x, were measured. This is a dynamic experiment, as opposed to the static measurements which were done with the force sensor. The m-file used for the estimation can be found in appendix D.3. A state space model with user-defined free parameters and some known parameters is used for the estimation. The values obtained with the estimation for the force constants, K_1 and K_2 vary from $1 \cdot 10^{-6}$ to $4 \cdot 10^{-6}$, depending on the initial conditions. Higher order fits were also attempted because satisfactory prediction of dynamic validation data was not possible. But the higher order terms were negligible in comparison with the quadratic model fit used in subsection 4.2.4. The values for the force constants obtained with PEM are not used any further.

5.4.5 Sliding mode control of the non-affine system

A sliding mode controller could offer more robustness to model uncertainties. The theory used in is used to implement the sliding mode controller ([4], [5], [8] and [13]). The system is non-affine because of the non-linearity of the input. The system equations can be rewritten as:

$$\dot{x} = f(x) + B \cdot h(I_1, I_2, x) \tag{5.17}$$

With:

$$f_1(x) = x_2 \tag{5.18}$$

$$f_2(x) = \frac{-K_f \cdot x_1}{m} \tag{5.19}$$

$$B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{5.20}$$

$$h(I_1, I_2, x) = \frac{K_1}{m} \cdot \frac{I_1^2}{g_1^2(x_1)} - \frac{K_2}{m} \cdot \frac{I_2^2}{g_2^2(x_1)} \tag{5.21}$$

Theory

At first a region where the system behaves as desired is defined as the sliding surface. Then a control action that takes the system onto the surface and keeps it there has to be determined. The switching control term takes the system to the sliding surface, the equivalent control is used when the system is on the sliding surface. The switching and the equivalent control term are derived separately. To determine the switching term, the sliding surface is defined as follows:

$$S(x,t) = \dot{e} + 2 \cdot \lambda \cdot e + \lambda^2 \int_0^t e \, dt \qquad with \;\; e = (x - r) \tag{5.22}$$

Asymptotic stability is guaranteed if the following condition is met:

$$S^T \cdot \frac{dS}{dt} = S^T \cdot (\frac{\delta S}{\delta x} \cdot f + \frac{\delta S}{\delta t} + \frac{\delta S}{\delta x} \cdot B \cdot h(I_1, I_2, x)) < 0 \tag{5.23}$$

Define:

$$\rho(x) = (\frac{\delta S}{\delta x} \cdot f + \frac{\delta S}{\delta t}) \tag{5.24}$$

Now the sliding condition is:

$$\rho(x) + h(I_1, I_2, x) < 0 \tag{5.25}$$

In a first attempt, the system is initially treated as SISO with only one actuator active at the time, depending on the sign of the tracking error. From this and the sliding condition the following control terms are proposed:

$$I_{1_{sw}} = -g_1(x_1) \cdot \sqrt{\frac{n(x)}{K_1}} \quad and \;\; I_{2_{sw}} = 0 \quad if \;\; S > 0 \tag{5.26}$$

$$I_{2_{sw}} = g_2(x_1) \cdot \sqrt{\frac{n(x)}{K_2}} \quad and \;\; I_{1_{sw}} = 0 \quad if \;\; S < 0 \tag{5.27}$$

To show that equation 5.26 meets the sliding condition 5.25, the case $S > 0$ needs to be analyzed. Since $n(x)$ is an upper bound for $\rho(x)$ it can be established that $-h(I_1, I_2, x) > \rho(x)$ and the sliding condition is met. For $S < 0$ the procedure is similar.

The equivalent control is the control when the system is in the sliding mode. $dS/dt = 0$ yields:

$$\dot{S} = \frac{\delta S}{\delta t} + \frac{\delta S}{\delta x} \cdot \dot{x} = \frac{\delta S}{\delta t} + \frac{\delta S}{\delta x} \cdot (\begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot h(I_1, I_2, x)) = 0 \tag{5.28}$$

From equations 5.24 and 5.28 follows that:

$$h_{eq} = -\frac{\delta S}{\delta t} - 2 \cdot \lambda \cdot f_1(x) - f_2(x) = -\rho(x) \tag{5.29}$$

Solving this for I_{eq} gives:

$$I_{1_{eq}} = g_1(x_1) \cdot \sqrt{\frac{\rho(x)}{K_1}} \quad and \;\; I_{2_{eq}} = 0 \quad if \;\; \rho > 0 \tag{5.30}$$

$$I_{2_{eq}} = g_2(x_1) \cdot \sqrt{\frac{-\rho(x)}{K_2}} \quad and \;\; I_{1_{eq}} = 0 \quad if \;\; \rho < 0 \tag{5.31}$$

It can be shown, that even in the non-affine case a total control action given by the sum of the switching and the equivalent terms satisfy the modified sliding condition (equation 5.25):

$$I = I_{sw} + I_{eq} \tag{5.32}$$

Stability for all four situations can be proven. This control strategy makes it possible that both actuators are active at the same moment. This controller is implemented and has resulted in very good tracking performance, which can be seen in the following paragraph.

Simulations and real time results

Sliding mode control of the non-affine system has led to the best results. In some of the following figures the capacitive sensor has been set to "fine measurement". Which improves the performance but this also reduces the range. When measurement are done with laser, they could be improved without the loss of range. C-code used for the simulations and real-time experiments can be found in appendix E.1 and E.2. The arm model is only used in simulation. In the experiments of coarse, this model is replaced with the real system. In figures 5.13 and 5.14 the simulation and real time results of the 1 [Hz] 100 [μm] sine wave can be seen. In real time the tracking error is around plus and minus 4 [μm], which is four times better then the performance of the feedback linearized system. The controller stabilizes the system in a large range which can be seen in in figure 5.15. Steps from 10 up to 400 [μm] are performed. These steps are made towards the second actuator. Of coarse these steps could also be made towards the first actuator. This bring the total range to 800 [μm]. All the steps are stable with a short settling time and a small steady state error (below $\pm$ 1 [μm]). In figure 5.16 a 50 [μm] step is performed, with the capacitive sensor set at fine measurement. This resulted in a tracking error below 2 [μm] and a steady state error of $\pm$ 200 [nm].

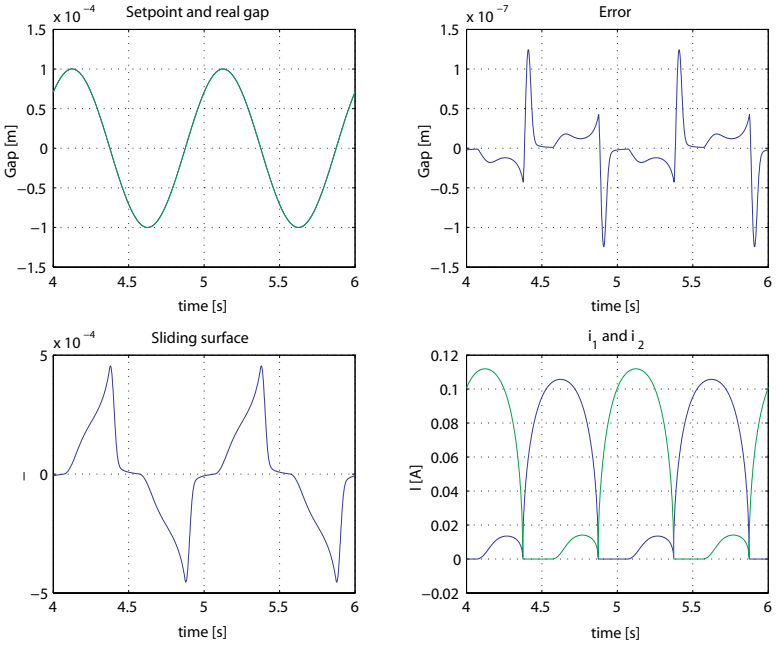


Figure 5.13: Tracking of 1 [Hz] sinusoidal, amplitude 100 [μm] (simulation, SMC)

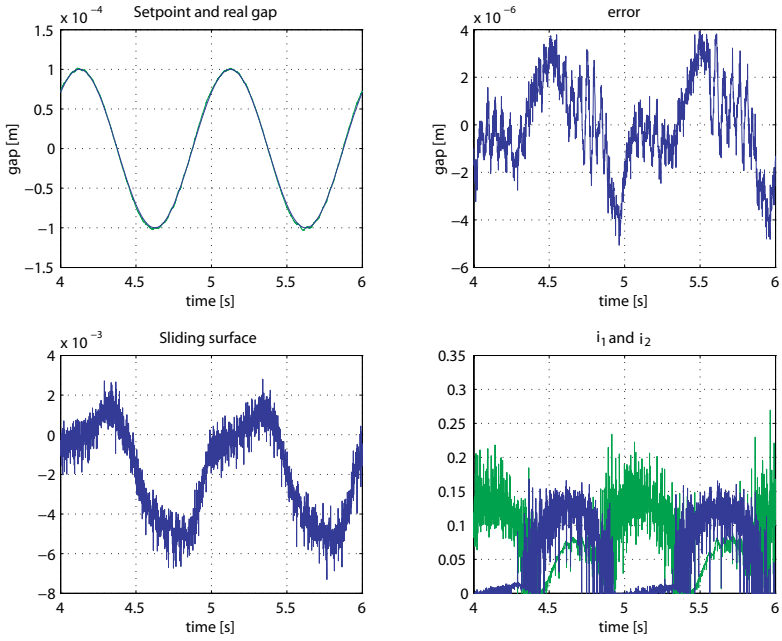


Figure 5.14: Tracking of 1 [Hz] sinusoidal, amplitude 100 [μm] (real time, SMC)

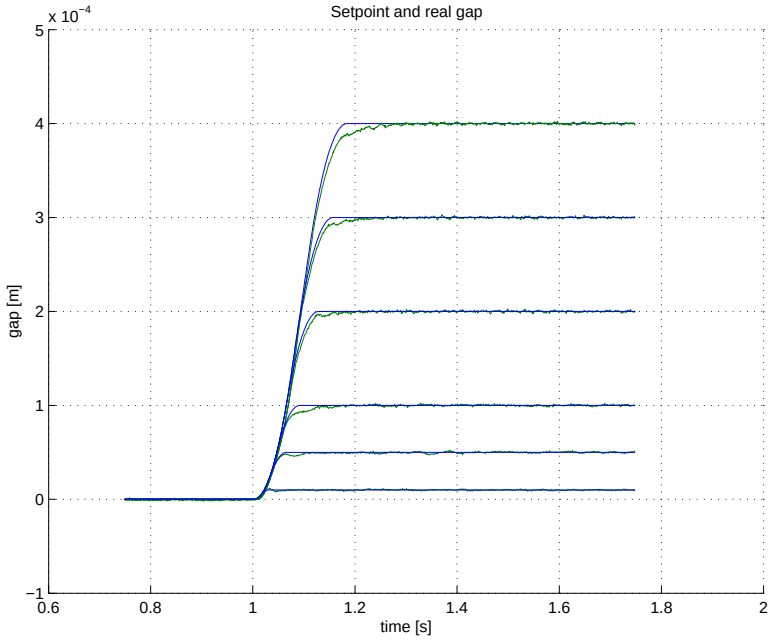


Figure 5.15: Stepresponse (SMC)

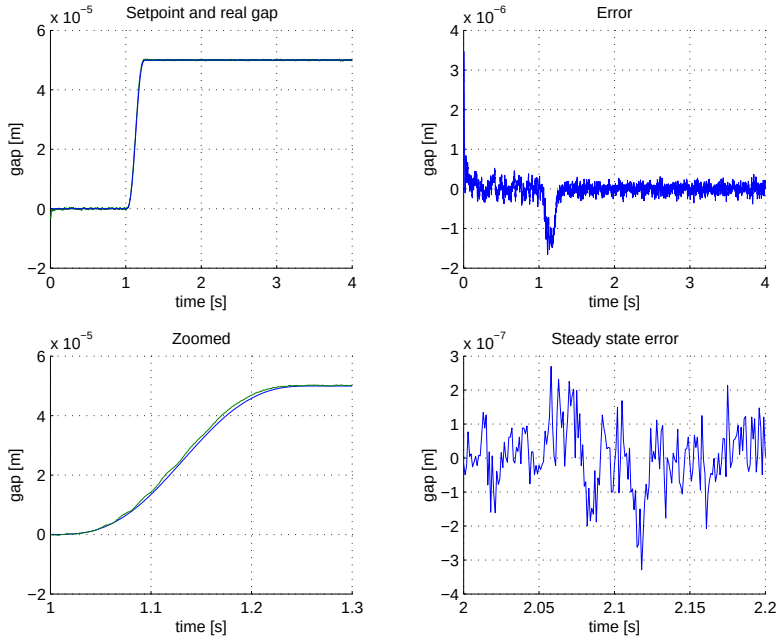


Figure 5.16: Stepresponse (fine measurement, SMC)

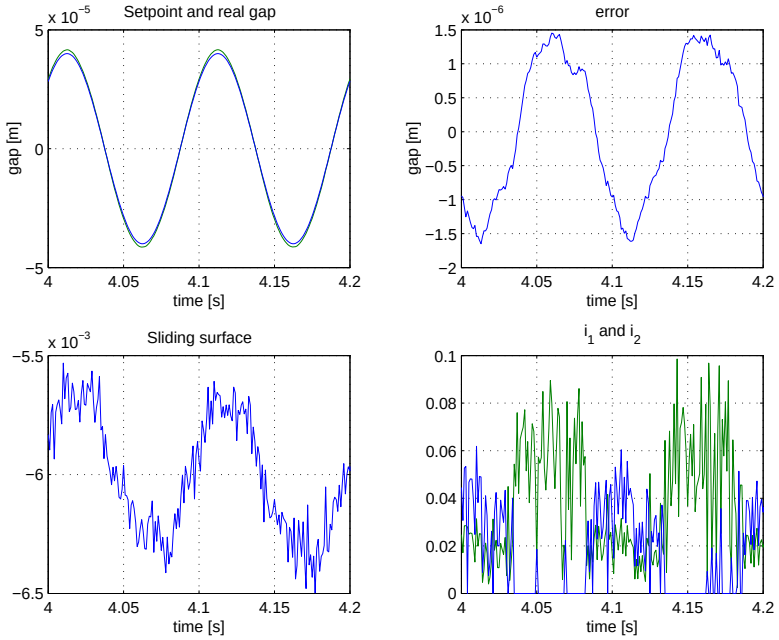


Figure 5.17: Tracking of 10 [Hz] sinusoidal, amplitude 40 [μm] (real time with fine measurement, SMC)

5.4.6 Different actuator configurations

All the mentioned experiments are done with the all four different actuator configurations; the E shaped and the round actuator acting both on a laminated and a solid target. The different experiments are done to look at the differences in performance between the actuator configurations. Because there is no lateral movement between the actuator and its target, Eddy current losses are not significant in this application. Therefore the round actuators are not laminated which makes them cheaper and easier to fabricate. Although, in the experiments no differences occurred between the laminated and solid actuators. Due to the horizontal movement the actuators do not have to compensate for the gravity and therefore the currents are relatively low. In future applications the gravity does not need to be compensated with the actuators used to control the position of the beam, but with an additional actuator or even better; permanent magnets.

5.4.7 Conclusion

Different controllers have been tested. The linear controllers (PID and LQR) used in combination with the non-linear and feedback linearized system did not lead to the desired results. Because of the non-linearities far away from the initialization point and, in case of the feedback linearization, the uncertainties in the force model. The sliding mode control has led to the best performance; in combination with the same force model. The SMC offers stability and good performance in a large operating range. Laminations are no longer necessary for this application. This makes the fabrication of an actuator a lot easier and cheaper.

Chapter 6

6DOF designs concepts

6.1 General concepts

The future of the high precision manufacturing industry brings the usage of extreme ultra violet wavelengths. These wavelengths do not penetrate air, therefore any machine designed to work with these wavelengths needs to operate in high vacuum. Conventional stages use airbearings which are difficult to use in vacuum. These problems may one day be solved by using (electro)magnetic levitation and manipulation. In the following chapter, several such concepts are discussed. One of the concepts is chosen and worked out. Sections will be dedicated to actuation, kinematics, decoupling and position measurement. The stage requires XY-travel of 200 [mm] x 200 [mm] and Z-travel of 1-2 [mm]. There are many ways to move a stage in six degrees of freedom, each with their own advantages and disadvantages. Three different possibilities are discussed:

6.1.1 Old design

Before this research project started, a lot of work had already been done by our predecessors. Lionel Fevre [2] designed and built the first 6DOF stage together with Erik Visser [13]. This stage consists of a tray with four arms. Each arm is suspended by four actuators which allow axial travel. These suspensions are mounted on dual linear motors that provide the coarse travel in XY (see figure 6.1). Linear motor pairs 1 and 2 provide Y motion, pairs 3 and 4 provide X motion. So, a total of 16 electromagnetic actuators and 8 linear motors are used to provide 6DOF travel. Laser interferometry is used to measure the four linear motor positions, yielding X,Y and rZ. At each suspension, two capacitive probes measure horizontal and vertical position of each arm within the suspension, yielding Z, rX and rY measurements. Advantages of this design are cheap production (few surfaces have to be machined for air-bearings) and cheap position measurement (no large L-mirrors are required, all laser interferometry takes place along fixed paths; only small mirrors are required). Disadvantages are overdetermined construction, overactuation, variable deflection of the arms and the usage of unmachined laminations (poor measurement, see chapter 2).

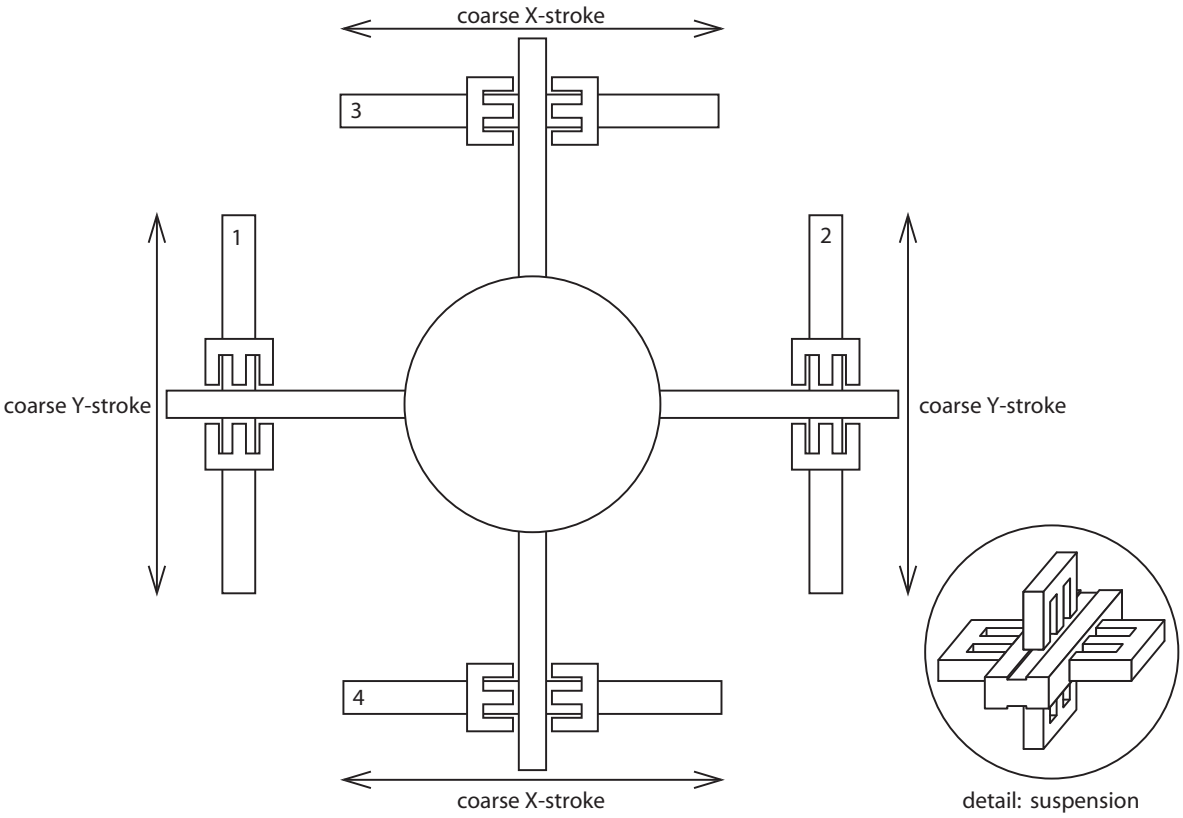


Figure 6.1: The old design for 6DOF motion

6.1.2 H-bridge

A tried and accomplished design for 6DOF manipulation is the H-bridge design. A large beam on two guided provides X-travel, a smaller sled on this large beam provides Y-travel and a final stage on this sled provides Z-travel and three adjustable angles. Sometimes the angle-adjustments are incorporated in X- or Y-travel. This concept can also be used with electromagnetic actuators. In figure 6.2, such a setup is shown. Linear motors provide the long XY-stroke, air bearings in these motors could be replaced by (electro)magnetic actuators. On the final stage, these actuators can take care of the other DOFs. Advantages of these designs are easy decoupling of the DOFs and the large amount of available knowledge on these systems. Disadvantages are expensive L-mirrors and the stacking of errors from the separate stages because they are mounted on top of each other and significant Eddy current losses.

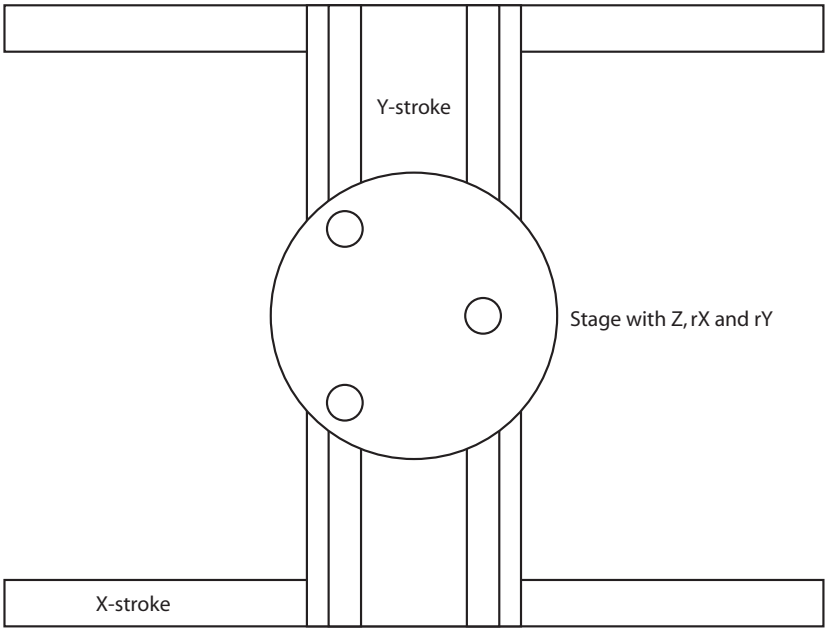


Figure 6.2: H-bridge design using electromagnetic actuators

6.1.3 Combined stages

Another concept is to separate the 6DOF motion into a coarse XY-stroke and a small stage that has small and accurate 6DOF travel. By doing this, large travel can be achieved using cheap and relatively inaccurate actuators. The smaller stage mounted on top this coarse stage, can provide enough travel to compensate the errors of the coarse stage and will also have the other four DOFs. Advantages of this type of design are cheap XY-motion, and the fact that accuracy is only required at the small 6DOF stage over small range of motion. Also, by proper placement of mirrors, the absolute position of the tray can be measured; most existing stages do not measure directly to the tray. Disadvantages are that errors from the XY-stage in other directions than X or Y are transmitted to the smaller stage, and measurement of the two separate stages have to be added up to yield an absolute position.

6.1.4 Concept choice

A good advantage of the combined stage design concept, is that two problems (long travel and very accurate 6DOF travel) can be separated. In effect, both stages can be designed separately, and either solution can be used on its own. Moreover, the old design can easily be improved by reducing the four arms to three and designing a statically determined suspension rather than the 16 actuators used now. However, the main disadvantages will remain. Also, with the H-bridge design it is difficult to meet to the set requirements. It will be extremely difficult to realize large travel with this new type of actuators, at high accuracy (sub-micron). In the concept of the seperate stages, only small ranges of travel have to be achieved with high accuracy, which is more feasible with these actuators. Therefore, this report will elaborate on this concept.

6.2 Small stage concepts

After the concept of separate stages is chosen, concepts for both stages need to be evaluated. This reports concentrates on the design of the accurate 6DOF small stage because the 2DOF XY-stage can be considered trivial due to its low requirements. In this section, three different concepts are discussed:

6.2.1 Oil-immersed box

Earlier research by Stephen J. Ludwick and David L. Trumper [10] has shown that electromagnetic E-actuators are capable of achieving nanometer accuracy in 6DOF motion control. Using an aluminum block, milled out to match the density of the oil it floats in, they achieved very high vibration damping and small actuator forces (which leads to low heat-dissipation in the system). A total of 12 actuators and 6 capacitive probes are used to control the system kinematically. Although their design only allows travel of 100 [μm] cubic, it could be scaled to reach 1-2 [mm] cubic travel.

6.2.2 Elastic mounting

A totally different way to allow motion without friction, play or hysteresis is elastic suspension. In principle this is a difficult thing to do, because any physical attachment to the fixed world will suppress at least one the the 6DOFs, while all need to be free. In order to realize this anyway, a body can be elastically mounted to allow one ore more DOFs. Next another body can be attached to the first to allow the remaining DOFs. One possible model of this concept can be seen in figure 6.3. The outer triangle is suspended with three folded leafsprings. This allow X, Y and rZ travel in the plane of the triangle. The inner triangle is suspended in the outer, by using three sets of parallel struts. This allows travel in Z and small rX and rY travel. By placing two actuators all sides of the outer triangle X, Y and rZ can be actuated. Three actuators acting on all corners of the inner triangle actuate Z, rX and rY. This design has a thermal center.

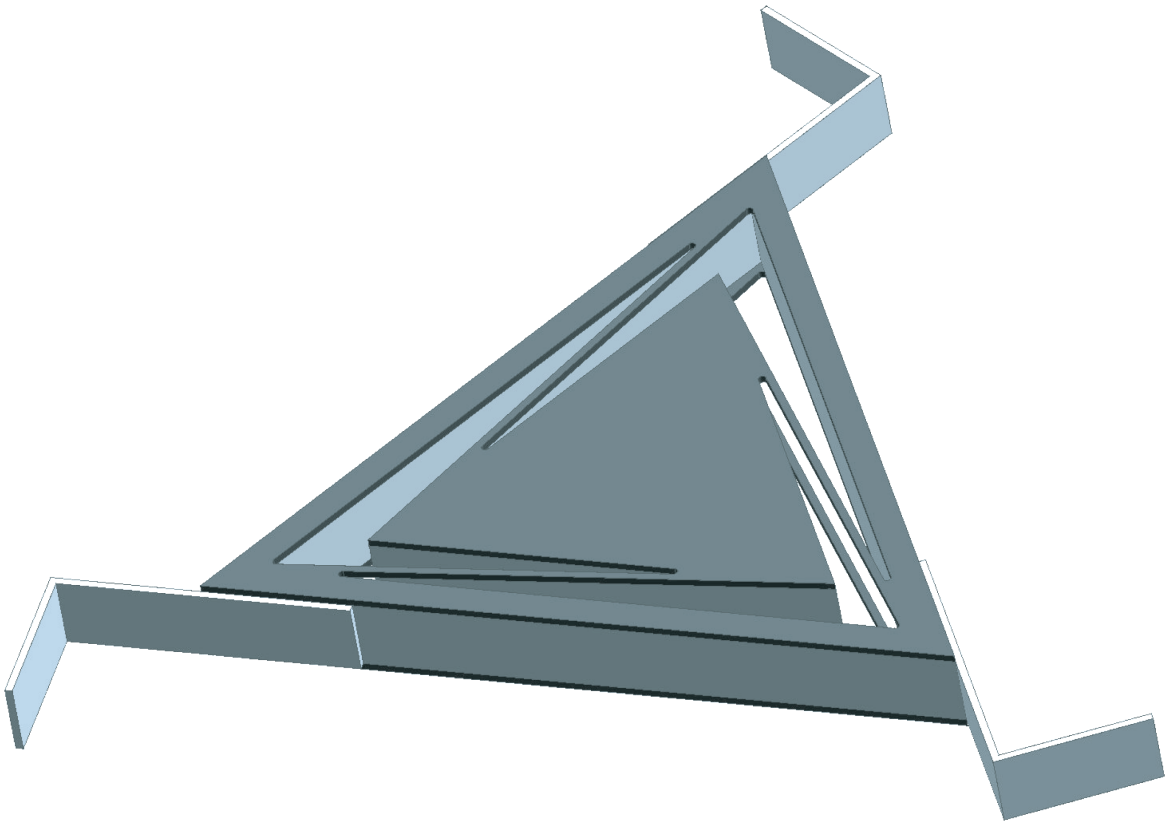


Figure 6.3: The center body allows travel in 6DOF

6.2.3 Maglev Hexapod

Another established concept for 6DOF motion control, is the hexapod or Stewart platform. This concept uses six linear actuators (usually lead-screws or pneumatic/hydrolic cilinders) mounted in three A-frames (triangles, touching at the bases, see figure 6.4). The eletromagnetic actuators discussed in chapter 4 are also capable of dictating a length, therefore, six of these actuators in a hexapod configuration could control a floating tray in 6DOF. Because the actuators can only exert attractive forces, they would have to be mounted above the tray. In order to resolve this, the attractive actuators are combined a pair of repelling permanent magnets. Not only can the actuators be mounted underneath the tray now (allowing free access from above for any production-process), but a new type of actuator is created. These actuators can exert both attractive as well as repulsive forces, and unlike the stand-alone actuators, they are stable by nature (i.e. when the distance between to permanent magnets increases, the force decreases allowing the floating part to come back down due to gravity and vice versa). By actually preloading every actuator with permanent magnets, rather than suspending the entire platform, all actuators can push ànd pull, allowing 6DOF motion control (see figure 6.5).

Figure 6.4: Example of a hexapod system with lead-screws

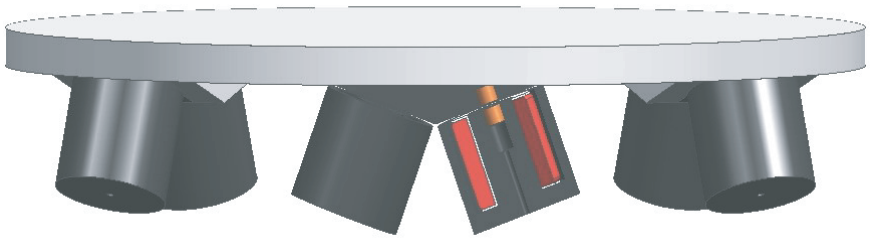


Figure 6.5: Maglev hexapod with four pairs of permanent magnets at each actuator

6.2.4 Concept choice

Although the fluid-suspended design has many advantages (high damping, no direct contact to the noisy fixed world, low actuator forces, kinematical drive and straightforward decoupling), the fact that it is suspended in a fluid medium is a determining factor. This design is to be suited for operation in vacuum conditions, any fluid would evaporate because at least some surface area of the fluid has to be exposed. The elastically suspended system meets the requirements for travel and vacuum conditions, but the required 9 or 12 actuators complicate the design and some of the elastic parts can only be manufactured by EDM or waterjetting, making it impossible to manufacture in the FIT machineshop for testing. Besides the above mentioned disadvantages for the first two concepts, the hexapod concept is favorable because of its following advantages:

- The tray is floating in air, isolated from ground vibrations.
- The tray does not require any power, so no wires are attached.
- The tray is kinematically suspended and actuated.
- The design has a thermal center.
- The preloaded actuators are naturally stable.
- All the suspension can be built under the tray, allowing free access from above.
- Low tray-mass and permanent magnet preloading allows for low control currents and therefore, low heat dissipation.
- Absolute positioning is possible with adequate placement of laser target mirrors.

Chapter 7

Maglev Hexapod analysis

7.1 Kinematics and Dynamics

In order to analyse the behavior of the maglev hexapod design, a dynamic model with the equations of motion are required. This information will be used later for control purposes. To do so, a fixed frame $\underline{\vec{e}}^0$ and a body-fixed frame $\underline{\vec{e}}^1$ are introduced. Next, the rotation-matrix $\underline{A}^{10}$, position-vector $\vec{r}_{CM}$ and the angular velocity vector $\vec{\omega}$ are determined.

$$\underline{A}^{10}(\theta_1, \theta_2, \theta_3) =$$

$$\begin{bmatrix} \cos\theta_2 \cos\theta_3 & \cos\theta_2 \sin\theta_3 & -\sin\theta_2 \\ -\cos\theta_1 \sin\theta_3 + \sin\theta_1 \sin\theta_2 \cos\theta_3 & \cos\theta_1 \cos\theta_3 + \sin\theta_1 \sin\theta_2 \sin\theta_3 & \sin\theta_1 \cos\theta_2 \\ \sin\theta_1 \sin\theta_3 + \cos\theta_1 \sin\theta_2 \cos\theta_3 & -\sin\theta_1 \cos\theta_3 + \cos\theta_1 \sin\theta_2 \sin\theta_3 & \cos\theta_1 \cos\theta_2 \end{bmatrix} \tag{7.1}$$

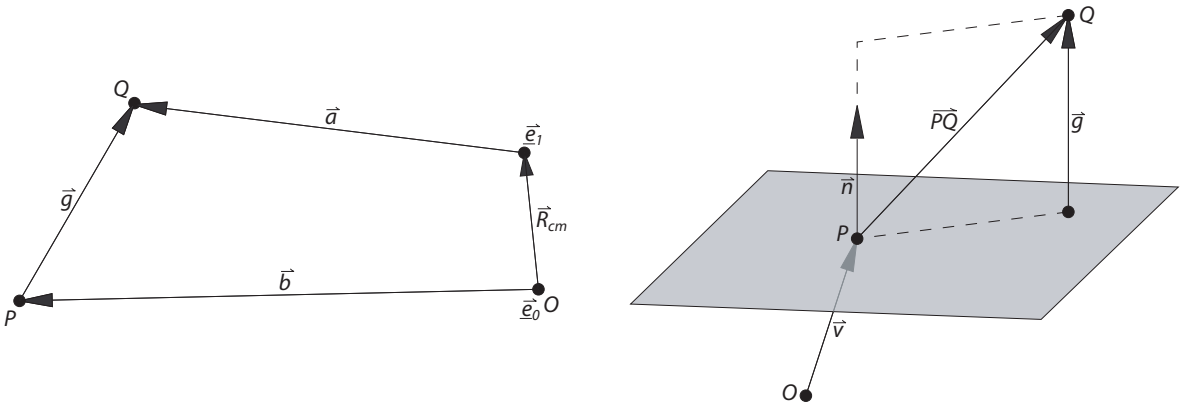
$$\vec{r}_{CM} = [x\ y\ z]\,\underline{\vec{e}}^0 \tag{7.2}$$

$$\vec{\omega} = \underline{\omega}^{1^T}\,\underline{\vec{e}}^1$$

$$\underline{\omega}^1 = \begin{bmatrix} \dot{\theta_1} - \dot{\theta_3}\sin\theta_2 \\ \dot{\theta_2}\cos\theta_1 + \dot{\theta_3}\sin\theta_1\cos\theta_2 \\ -\dot{\theta_2}\sin\theta_1 + \dot{\theta_3}\cos\theta 1\cos\theta_2 \end{bmatrix} \tag{7.3}$$

With this, the following vectors are defined:

- Body-fixed vectors $\vec{a}_i$ to each magnetic target
- Body-fixed normalvectors $\vec{v}_i$ for each target-surface
- Fixed frame vectors $\vec{b}_i$ to each actuator center



These vectors are used to determine the perpendicular distance from each actuator-center to each corresponding magnetic target as a function of x, y, z, rx, ry and rz and some parameters. These six distances are six airgaps that uniquely determine the position of the tray in 6DOF, therefore, these decoupling formulas can be used for simulation and control. The forces F_i also act along these vectors. This leads to the three translational equations of motion (7.4, 7.5 and 7.6) and three rotational equations of motion (7.7). The M-file used for computing all kinematics can be found in appendix D.5

$$m\ddot{x} = \sum_{j=1}^{6} F_j \vec{n}_j \cdot \vec{e}_1^0 \tag{7.4}$$

$$m\ddot{y} = \sum_{j=1}^{6} F_j \vec{n}_j \cdot \vec{e}_2^0 \tag{7.5}$$

$$m\ddot{z} = \sum_{j=1}^{6} F_j \vec{n}_j \cdot \vec{e}_3^0 - mg \tag{7.6}$$

$$\sum_{j=1}^{6} \vec{a}_j \times \vec{F}_j = \underline{\vec{e}}^{1^T} \begin{bmatrix} J_1\dot{\omega_x} - (J_2 - J_3)\omega_y\omega_z \\ J_2\dot{\omega_y} - (J_3 - J_1)\omega_z\omega_x \\ J_3\dot{\omega_z} - (J_1 - J_2)\omega_x\omega_y \end{bmatrix} \tag{7.7}$$

With the kinematics from the above section, a setpoint in x, y, z, rx, ry and rz can be translated into six setpoints for the gaps. Either of the controllers from chapter 5 can be implemented to control these gaps, or computed torque control (7.12) can be used to control the entire system. To do so, first the equations of motion are rewritten to the form of 7.8. $\underline{M}(\underline{q})$, $\underline{H}(\underline{q})$ and $\underline{S}(\underline{q})$ are given in equations 7.9, 7.10 and 7.11. Finally in equation 7.12 a stabilizing control law is defined.

$$\underline{M}(\underline{q})\ddot{\underline{q}} + \underline{H}(\underline{q}, \dot{\underline{q}}) = \underline{S}(\underline{q})\underline{\tau} \tag{7.8}$$

With:

$$\underline{q} = [x\ y\ z\ \theta_1\ \theta_2\ \theta_3]^T \qquad\qquad \underline{\tau} = [F_1\ F_2\ F_3\ F_4\ F_5\ F_6]^T$$

$$\underline{M}(\underline{q}) = \begin{bmatrix} m\underline{I}_3 & \underline{0}_3 \\ \underline{0}_3 & \begin{bmatrix} J_1 & 0 & -J_1 \sin\theta_2 \\ J_2 \cos\theta_1 & J_2 \sin\theta_1 \cos\theta_2 & \\ -J_3 \sin\theta_1 & J_3 \cos\theta_1 \cos\theta_2 & \end{bmatrix} \end{bmatrix} \tag{7.9}$$

$$\underline{H}(\underline{q}, \dot{\underline{q}}) = \begin{bmatrix} 0 \\ 0 \\ mg \\ -J_1\dot{\theta}_2\dot{\theta}_3 - (J_2 - J_3)\omega_y\omega_z \\ -J_2(\dot{\theta}_1\dot{\theta}_2 \sin\theta_1 + \dot{\theta}_2\dot{\theta}_3 \sin\theta_1 \cos\theta_2 - \dot{\theta}_1\dot{\theta}_3 \cos\theta_1 \cos\theta_2) - (J_3 - J_1)\omega_z\omega_x \\ -J_3(\dot{\theta}_1\dot{\theta}_2 \cos\theta_1 - \dot{\theta}_2\dot{\theta}_3 \cos\theta_1 \sin\theta_2 + \dot{\theta}_1\dot{\theta}_3 \sin\theta_1 \cos\theta_2) - (J_1 - J_2)\omega_x\omega_y \end{bmatrix} \tag{7.10}$$

$$\underline{S}(\underline{q}) = \begin{bmatrix} \vec{n}_1 \cdot \underline{\vec{e}}^0 & \vec{n}_2 \cdot \underline{\vec{e}}^0 & \vec{n}_3 \cdot \underline{\vec{e}}^0 & \vec{n}_4 \cdot \underline{\vec{e}}^0 & \vec{n}_5 \cdot \underline{\vec{e}}^0 & \vec{n}_6 \cdot \underline{\vec{e}}^0 \\ (\vec{a}_1 \times \vec{n}_1) \cdot \underline{\vec{e}}^1 & (\vec{a}_2 \times \vec{n}_2) \cdot \underline{\vec{e}}^1 & (\vec{a}_3 \times \vec{n}_3) \cdot \underline{\vec{e}}^1 & (\vec{a}_4 \times \vec{n}_4) \cdot \underline{\vec{e}}^1 & (\vec{a}_5 \times \vec{n}_5) \cdot \underline{\vec{e}}^1 & (\vec{a}_6 \times \vec{n}_6) \cdot \underline{\vec{e}}^1 \end{bmatrix} \tag{7.11}$$

Computed torque control law:

$$\underline{u} = \underline{S}(\underline{q})\Big(\underline{M}(\underline{q})\Big(\ddot{\underline{q}}_d(t) - \underline{K}_d(\dot{\underline{q}} - \dot{\underline{q}}_d(t)) - \underline{K}_p(\underline{q} - \underline{q}_d(t))\Big) + \underline{H}(\dot{\underline{q}}, \underline{q})\Big) \tag{7.12}$$

7.2 Measurement

Several possibilities arise to measure the position of the tray; The absolute position can be measured with respect to a fixed point in space, or the relative position between the tray and the optics or tool can be measured. Also, both the coarse stoke stage and the maglev hexapod stage can be equipped with their own sensors. Next, both measurements need to be added up to yield a total position. Alternatively, the total position of the tray can be measured directly; but this involves the use of expensive L-mirrors. Thirdly, the position feedback can be done on either the 6DOF position or on the six airgaps. Depending on the measured position (either in 6DOF or six airgaps), the decoupling matrix needs to be used either once or twice, which may lead to the amplification of the uncertainties in this matrix. Considering the three issues mentioned above and depending on the application of the stage, a proper measurement scheme can be derived.

Chapter 8

Preloaded actuators

The concept that was chosen in the previous chapters, relies on actuators that can exert both positive and negative forces. For sake of compactness and axial symmetry, the designs are based on the round actuators. Placing one ore more sets of repelling magnets near (or within) each actuator is one way to do this. First, the placement of these opposing magnets must be determined. Then, depending on the weight of the tray that has to be supported, the range of force for each actuator can be determined, which will lead to the required strength of each permanent magnet. The placement of a lot of magnets in and near the tray will result in flux leakage. All this magnetic energy may affect the process taking place on the tray and must therefore be shielded or contained. Moreover some FEM simulations have been done to test the designs. Finally the results from the FEM simulations must be verified with physical experiments. After the results have been verified and a good characterization has been made, the new actuator is controlled with the sliding mode controller.

8.1 Configuration

To prevent introducing moments on the magnetic targets, the net attractive force and the net repulsive force need to act on the same spot. This can be achieved either by placing one pair of repelling magnets concentric within the round actuator, or by placing more pairs symmetrically around the centerline of the actuator. In the first option (figure 8.1a), the actuator can be used to 'reduce' the permanent magnets force. The advantage of this is that the magnetic insulation is already built into the design (see figure 8.3). In case of the second option (figure 8.1b), four pairs of magnets are placed symmetrically around the core; this is easy to manufacture and the large number of magnets (24 pairs) available to support the weight of tray, allows smaller, less powerful magnets. Analysis of the fluxlines in both situations, have lead to the choice for the concentric placement. Rare-earth magnets are sufficiently strong to support the weight, even with only six pairs.

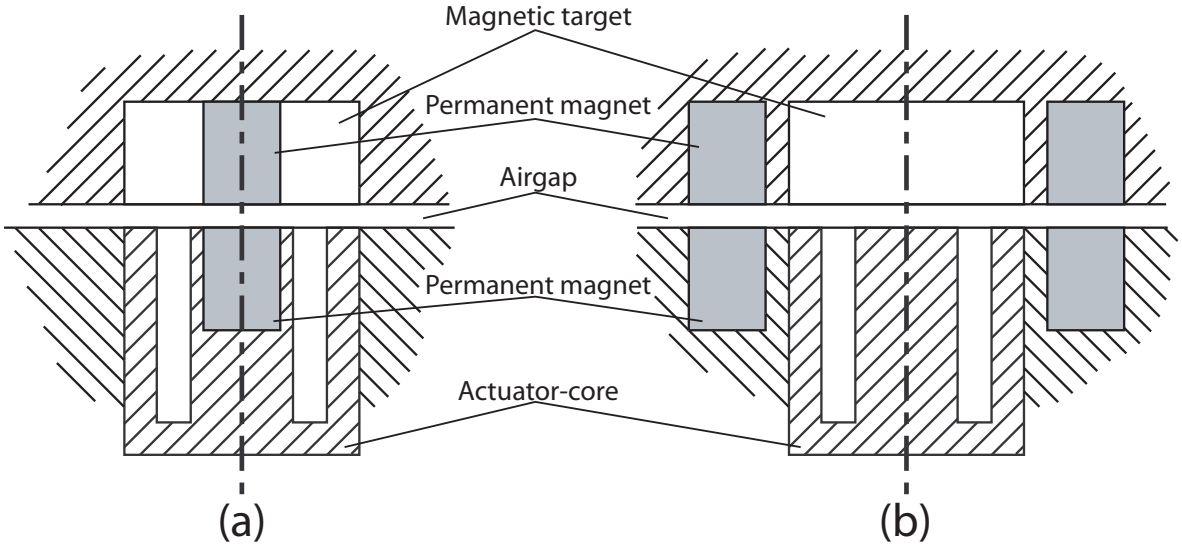


Figure 8.1: Placement of permanent magnets, within or around an actuator

8.2 Dimensioning

The forces acting on the tray are built up out of three components: gravity, attractive force from the actuators and repelling force from the magnets. The summed z-component of all magnet pairs, must equal the gravity-load due to the weight of the tray to support gravity. Moreover, the summed z-component of all magnet pairs must also provide preloading against the actuators. The tray travel in combination with the placement-angle of the actuators, determine the range of each gap (e.g. 0,1-2,5 [mm]). Using the actuatormodels from subsection 4.2.4, this gaprange leads to a forcerange. The force exerted by each actuator, must also be compensated by the permanent magnets to create equilibrium.

8.2.1 Gravity-compensation

The weight of the tray (12" diameter), the magnetic targets and their fixtures, the permanent magnets on the tray and optics for measurement, is estimated to be about 3 [kg]. Any parts added to the tray must be accounted for later. The angle at which the actuators are placed, is a trade-off between travel and in-plane force (needed for x, y and rz actuation). The more vertical the actuators are placed, the larger the travel will be without one of the actuators hitting its target. The more horizontal the placement, the larger the in-plane force component but, the gravity-component will be reduced; therefore bigger magnets would be required. Since the vertical forces required are much larger then the in-plane forces, the angle for the actuators was set at 20°. The force now decomposes to about 25% in-plane and 75% vertical. The force between one pair of magnets can be modeled as equation 8.1, the total vertical force can be calculated from equation 8.2:

$$F = K_2 \cdot \frac{P^2}{g^2} \tag{8.1}$$

$$F = 6 \cdot \sin 20^\circ \cdot K_2 \cdot \frac{P^2}{g^2} \tag{8.2}$$

with:

K_2	= Constant for permanent magnet	$[N \cdot m^2/Wb^2]$
P	= NI-equivalent for permanent magnet	[Wb]
g	= Airgap	$[m]$

8.2.2 Actuatorforce

A model of the maglev hexapod has been drawn in Unigraphics. In this model, X,Y and Z-travel are parameters that can be changed. The airgaps between each actuator and its target can be evaluated at each XYZ-position. In this manner, the extreme values for the airgaps are determined. These values are between 400 and 2200 [μm]. At 400 [μm], the actuator can exert forces between 0 and 70 [N]. At 2200 [μm], the force ranges from 0 to
8 [N]. Experiments have shown that the force exerted by the permanent magnets approaches a linear relation (probably due to the metal surrounding the magnets, see figure 8.5). Had this relation been as the theory prescribes, equilibrium between repelling and attractive force could exist without changing the current. Because the relations differ, each current dictates a unique airgap at which equilibrium exists.

8.3 Magnetic shielding

Since a lot of manufacturing processes are influenced by the presence of EM-field (e.g. e-beam machining), it is desirable to keep flux-losses to a minimum. An easy way to do this, is to equip all permanent magnets with a ferrous sleeve. By doing so, the magnetic flux tends to choose the path of least resistance (high relative permeability, μ_r) and run through the sleeve. The effect of this is illustrated in figure 8.2. The only disadvantage to this is that the effective forces between two insulated permanent magnets are reduced[1]. These theories need to be tested with both FEM-models and experiments.

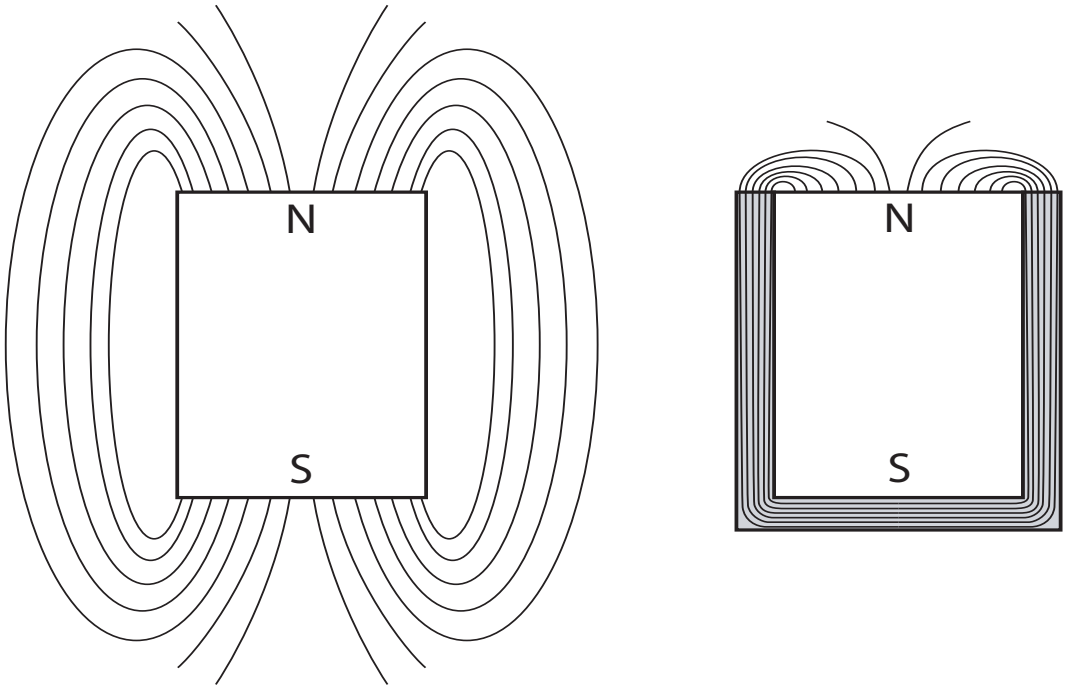


Figure 8.2: Uninsulated and insulated fluxlines

[1]Repelling forces between two magnets are caused by the compression of the field; since fluxlines cannot cross, they are pressed closer to each other when two equal poles approach. The ferrous sleeve will 'precompress' the magnetic fields, giving it weaker interaction with other fields.

8.4 FEM-simulations

The finite element package FEMLAB was used to analyse the behavior of the actuator in combination with the permanent magnets. The following matters were evaluated:

Permanent magnet strength
Because permanent magnets cannot be ordered by their NI-equivalent[2], the material, its magnetization-curve (BH-curve) and geometry determine the forces exerted between them. This means that in order to dimension a permanent magnet for an application, the NI-equivalent should be determined; either with experiments or FEM-simulations. If experiments can verify FEM-simulations, these could be used in stand alone in future. To attempt this, two opposing magnets were modeled in FEMLAB with several different parameters as they can be ordered. The simulations calculated the repelling force, which led to the choice of magnets to order. Three types were ordered, all $\frac{1}{4}$" in diameter and length: Samarium-Cobalt and Neodymium in grades 18, 28 and 38 [1].

Misalignment
When the hexapod design moves away from its zeropoint, misalignment occurs between the pairs of permanent magnets. This may lead to significant drop in repelling force. This could be improved by using one small magnet and a bigger one so that the smaller magnet's surface will always be aimed at the bigger one. Simulation has shown that the maximal misalignment that can occur in the hexapod design will only lead to 8% drop in repelling force; this can easily be overcome with the controlcurrent through the actuator.

Standalone actuator and preloaded actuator
Also, the actuator was modeled in FEMLAB to check the results from the FEM-simulations with the measured values from the experiments. The round actuator core was modeled as well as coil windings and the magnetic target. At first, the simulated values were much higher than the physical results, improvements to the model resulted in better numbers, but were still unsatisfactory. This model is then expanded with the placement of the permanent magnets in the centers. Analysis of this model has shown that equilibrium between repelling and attracting force can be created at any airgap by adjusting the control current. This means that the preloading principle works.
In figure 8.3 the left half is the model from Unigraphics, the right side is the axisymmetrical FEM-model. The fluxlines generated by the permanent magnets and the current can be distinguished. It is also visible that almost no fluxlines are leaking from the actuator due to the magnetic insulation. In reality, a holding plate was placed behing the permanent magnets in the magnetic target, isolating even more fluxlines (also visible in figure 8.4).

[2]The number of windings N with applied current I needed to create an electromagnet with the same field

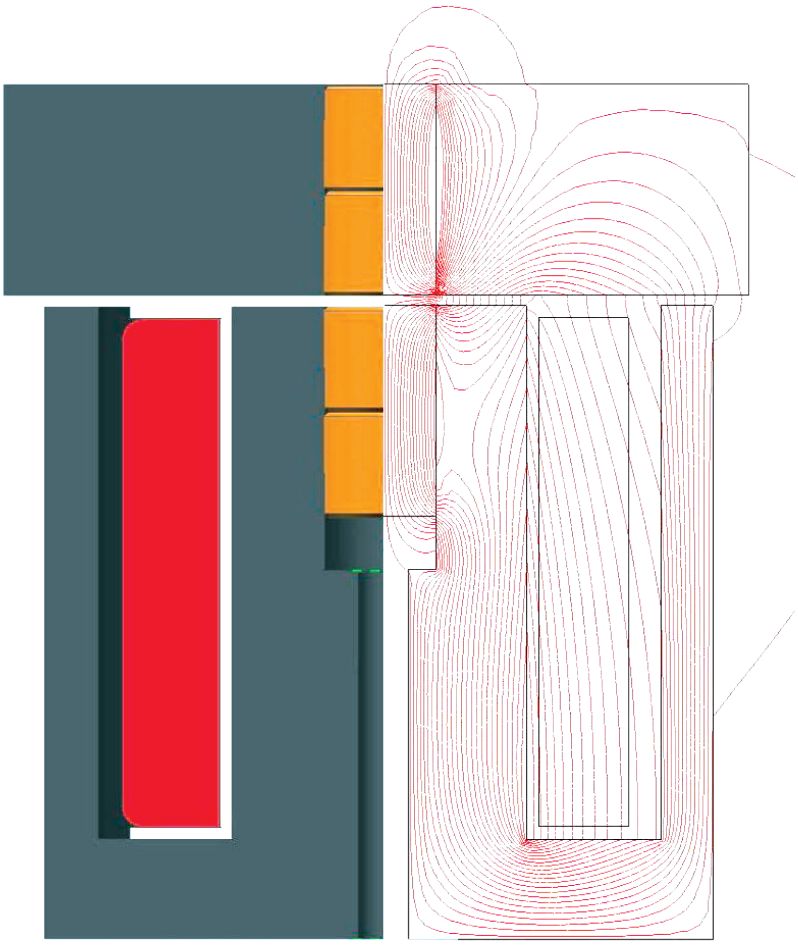


Figure 8.3: Unigraphics and FEM models for the round actuator with magnets

8.5 Experiments

8.5.1 Modified experimental setup

In order to verify the results from FEM-analysis, experiments need to be conducted. The goals of such experiments are to determine the NI-equivalent for the tested magnets. In order to do these experiment, a setup is required that can perform an accurate frictionless linear motion, that can measure the gap between the two magnets and measure the force between the two. The 1DOF setup discussed in chapter 3 is perfectly suited for these tests. The following adjustments are made to the setup to accommodate the experiments:

- A new magnetic target is machined with a hole in its center to allow placement of either one or two permanent magnets
- A small plate is machined to hold the magnets in place on the rear side of the magnetic target
- One of the round actuators is equipped with a hole in the center to place the magnets; also a hole is threaded all the way through the core to allow placement of an adjustment screw.

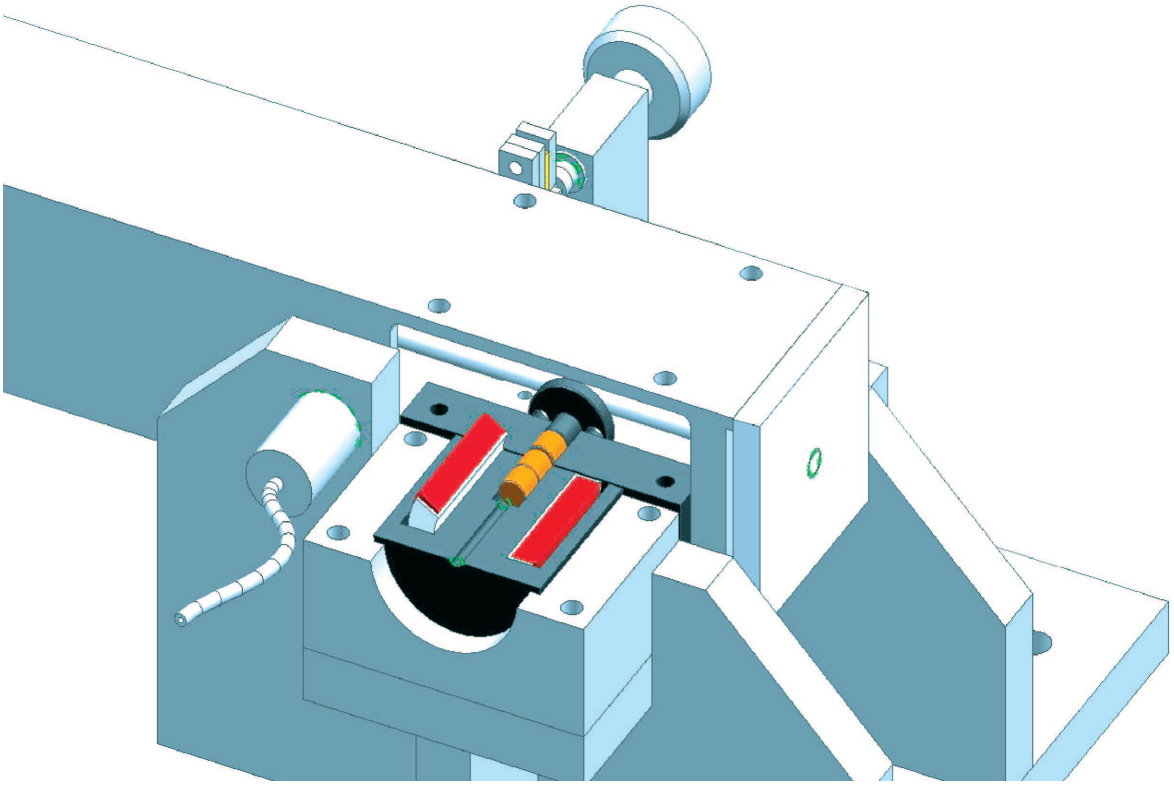


Figure 8.4: The 1DOF setup, accommodated for permanent magnets, some parts sectioned

8.5.2 Results

Three different permanent magnets are used, all of them are rare earth magnets but the strengt varies. The magnets are grade 18, 28 and 38. Grade 18 is the weakest and 38 the strongest. First of all the permanent magnets are characterized without the use of electro magnetic force. The measurements with two grade 38 magnets can be found in figure 8.5. The other measurements can be found in appendix C.1. The relationship is linear in contradiction to the theoretical model (equation 8.2). This is caused by the metal of the actuator surrounding the permanent magnets. The characterization of the permanent magnets is only done while they are placed within the actuators because this represents future applications. The behavior of the permanent magnets outside the actuators is not relevant for this design.

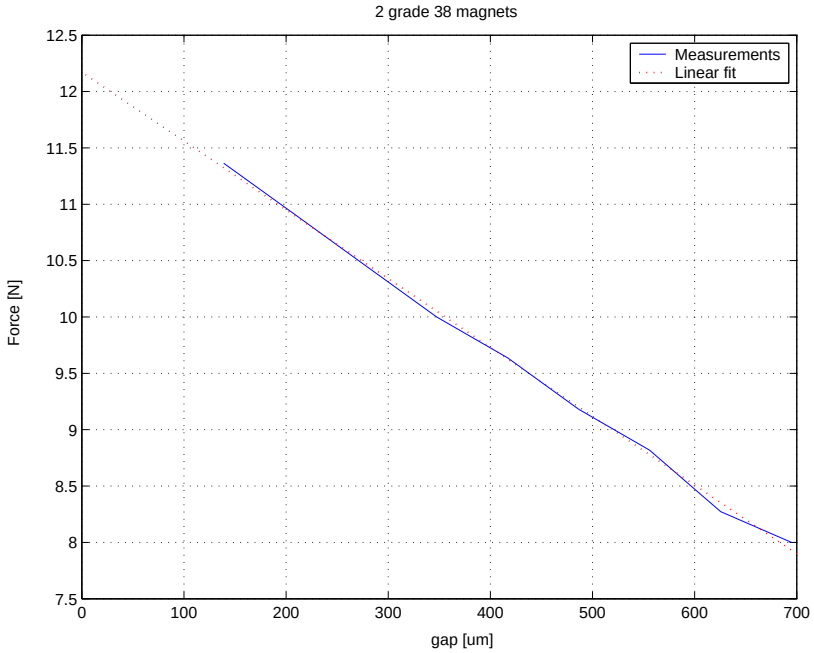


Figure 8.5: 2 grade 38 magnets with linear fit

Measurements are also done with the permanent magnets in combination with the electromagnetic actuators. The repelling forces of the permanent magnets dominate when the current through the coil is low. The attractive force slowly starts to win when the current is increased, and the beam is pulled towards the actuator. In figure 8.6 the measurements can be seen.

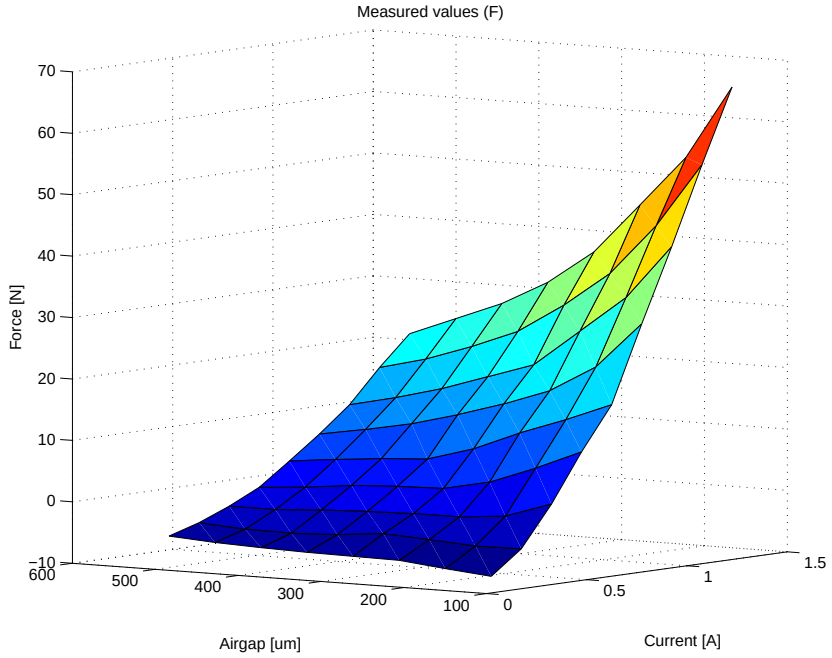


Figure 8.6: Preloaded actuator (measured with 2 grade 38 magnets)

In the following section the control for the preloaded actuator is proposed. The force constants for both the round actuator and the permanent magnets are needed. The force constants for the round actuator are already known. For the permanent magnets the linear fit is used (figure 8.5). The total force is the difference between these two:

$$m \cdot \ddot{x} = K_1 \cdot \frac{I^2}{g^2} - (K_2 \cdot g_p - K_3) - K_f \cdot x \tag{8.3}$$

With:

x	= translation beam	$[m]$
K_1	= Electromagnetic force constant	$[N \cdot m^2/A^2]$
K_2	= Permanent magnet force constant	$[N/m]$
K_3	= Permanent magnet force constant	$[N]$
K_f	= Flexure stiffness	$[N/m]$
g	= gap length $(x_0 + x_{ad} + x)$	$[m]$
g_p	= gap length $(x_0 + x)$	$[m]$
I	= current through coil	$[A]$

8.6 Control of the preloaded actuator

8.6.1 Theory

In section 5.4.5 the theory of sliding mode control is explained. The control strategy for the preloaded actuator is actually the same as the control strategy for the round actuator without the permanent magnets. Although, right now only one preloaded actuator is needed. One old actuator is replaced by the permanent magnets in the preloaded actuator. The target is repelled due to the permanent magnets and the electro magnetic force attracts the arm towards the actuator. But, in this situation a initialization is needed. The beam rests to its end stop in initial position due to the repelling permanent magnets. In case of the 6DOF design; the tray is floating at maximum height. In this position no movement is possible further away from the permanent magnets. Therefore an initialization current is applied to the coil(s). The beam or tray is attracted and right now control is possible in both directions.

8.6.2 Simulations and real-time results

In figures 8.7 and 8.8 the system performs steps from 10 up to 350 [μm]. There is stability in the whole range. In real time the tracking errors (with 100 [μm] steps) are around 20 [μm] and the steady state error is $\pm$ 3 [μm]. In figure 8.9 the system first rests to it's end stops. The system is pulled away from its end stop with the use of the initialization current. In simulation the initialization current is around the 0,72 [A] in real time this is 0,58 [A]. These two currents should be the same, this is not the case. A better characterization of the permanent magnets in combination with the electromagnetic force is needed. The capacitive sensor is again set to fine measurement. The system is attracted to its initial position (with the initialization current) and then performs 20, 40 and 50 [μm] steps. With tracking errors of successively 2, 4 and 5 [μm]. The steady state error is $\pm$ 500 [nm].

8.6.3 Conclusion

The round actuator in combination with permanent magnets controlled with a sliding mode controller offers stability and good performance in the entire operating range. The permanent magnets could offer gravity compensation in the 6DOF design. This proves the feasibility of the 6DOF design with the use of the preloaded actuator. The hybrid actuator matches the performance of the two opposing actuators. Sub micron accuracy could be achieved with the use of a laser position measurement and a better characterization of the permanent magnets in combination with the soft iron actuator.

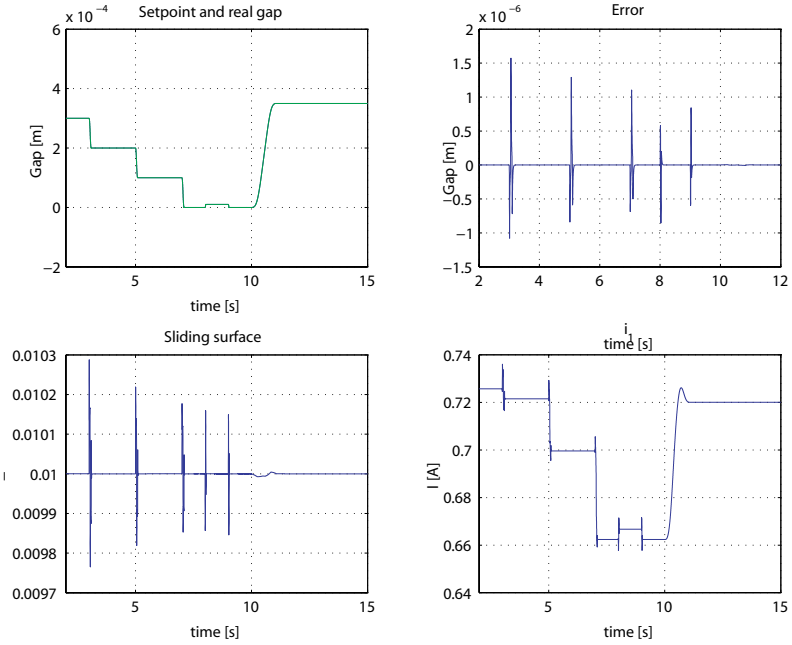


Figure 8.7: Tracking big steps (simulation)

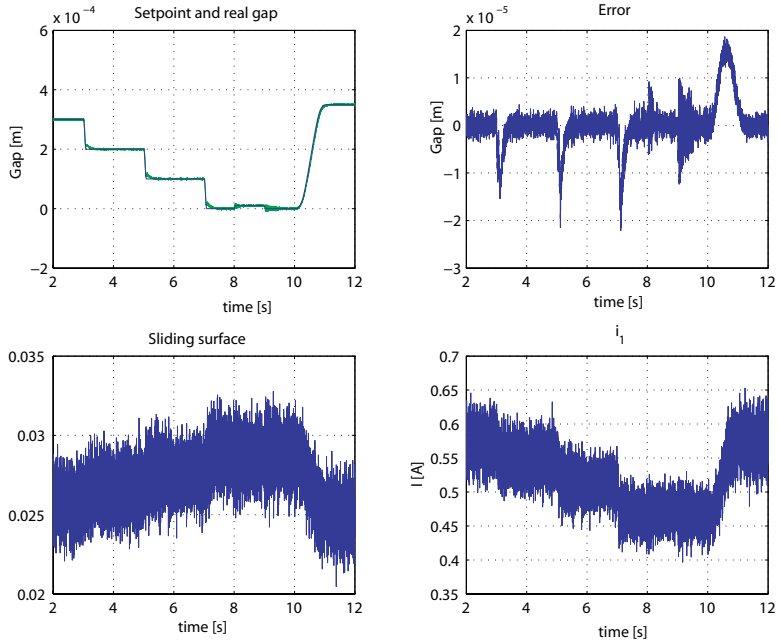


Figure 8.8: Tracking big steps (real time)

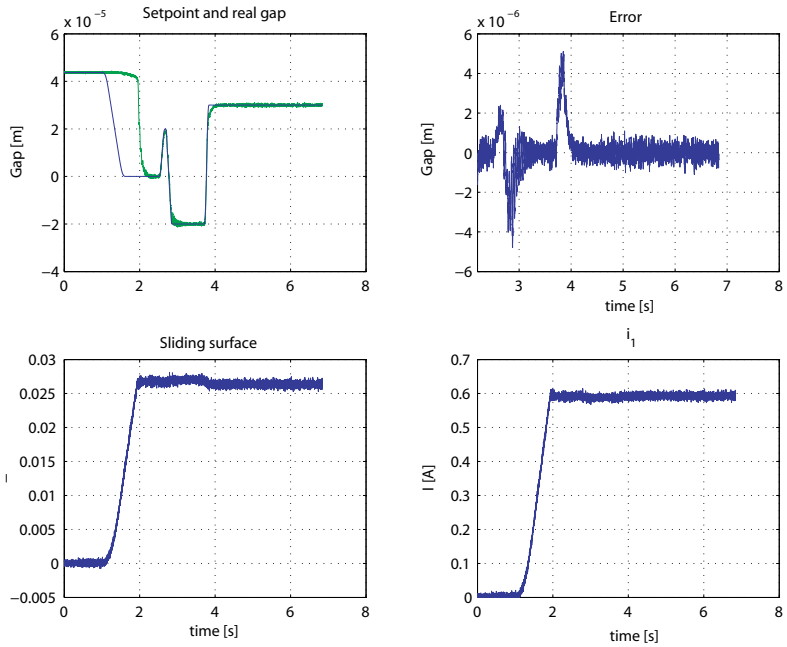


Figure 8.9: Tracking small steps (real time with fine measurement)

Chapter 9

Conclusions and recommendations

9.1 Conclusions

A new six degree of freedom motion stage has been designed. In this design the gravity is compensated with the use of permanent magnets. The permanent magnets are placed in an hybrid actuator, which combines both repelling forces and attractive, electromagnetic, forces. Laminations are no longer used in this actuator. The fabrication of this soft iron actuator is cheaper and easier. With the use of this new "preloaded" actuator the 6DOF design is feasible. A stabilizing control law is found and sub micron accuracy is achievable. To characterize the preloaded actuator and other actuator configurations a one degree of freedom setup has been designed and realized. Play, friction and hysteresis are no longer a problem due to the elastic hinge point. This setup made it possible to derive force models of the different actuator configurations.

9.2 Recommendations

- Laser measurement
  With the use of laser position measurement, it is possible to measure the position of the beam more accurately. The capacitive sensor could be set on fine measurement which improves the accuracy but limits the range. With a laser position measurement the same or even better performance can be reached in the whole operating range.

- Amplifiers
  The PWM modulation of the amplifiers introduces high frequency noise in the control currents and force- and position-measurements. Linear voltage to current amplifiers do not have this problem but were not available because of the high price. When a dedicated setup will be built, using these more expensive amplifiers is recommended.

- Preloaded actuator and FEM
  The preloaded actuator as discussed in this report, performs as expected. However, when a similar actuator needs to be designed for a different application, no good simulations are available. Therefore, it is recommended to improve the FEM-models so that they predict the actual behavior of the hybrid actuators much better.

Bibliography

[1] www.dextermag.com

[2] Fevre, L.J.P. "Design and construction of a 6DOF positioning system with nanometer accuracy", M.Sc. Thesis, Florida Institute of Technology, December 2003

[3] Gutierrez, H.M. and Fevre, L.J.P. "Design and construction of a 6DOF positioning system with long range motion in XY and nanometer resolution usign magnetic servo-levitation", presented at MIT, April 2004

[4] Gutierrez, H.M. and Ro, P.I. "Magnetic servo levitation by sliding mode control of non-affine systems with algebraic input invertibility"

[5] Gutierrez, H.M. and Ro, P.I. "Sliding mode control of a nonlinear-input system: application to a magnetically levitated fast-tool servo", IEEE transactions on Industrial Electronics, vol 45 no. 6, December 1998

[6] Lion Precision "Technical Brief LT02-0012", March 2004

[7] Rosielle, P.C.J.N. and Reker, E.A.G. "Constructieprincipes 1, bedoeld voor het nauwkeurig bewegen en positioneren", maart 2000

[8] Slotine, J. and Li, W. "Applied nonlinear control", Prentice Hall, London, 1991

[9] Smeets, R. and Nijmweegen, F.C. "User Manual TUeDACS/1 Quadratuur/Analog/Digital interface", Eindhoven University of Technology, Department of Physics, Physical and Technical Labratory Automation Group, BLN 2000-18 UM, March 2001

[10] Trumper, David L., Holmes, Michael L., Ludwick, Stephen J. "Modeling and Control of a 6DOF magnetic/fluidic motion control stage", 1996

[11] Trumper, D.L., Olson, S.M. and Subrahmanyan, P.K. "Linearizing control of magnetic suspension systems", IEEE transactions on CST, vol 5 no. 4, July 97

[12] "Uitreikbladen, Nauwkeurigheid van machines (4U700)", Eindhoven University of Technology, lecture notes 2003

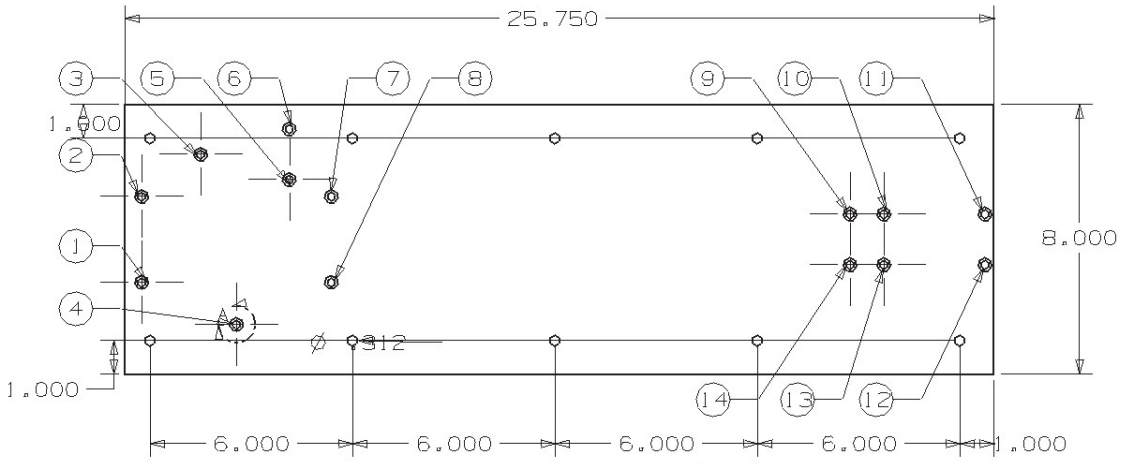
[13] Visser, E.E. "Modelling, instrumentation and control of a 6DOF positioning stage" Masters thesis, Eindhoven University of Technology, Mechanical Engineering, DCT 2004.112

[14] Wouw, Nathan van de "Multibody Dynamics", Eindhoven University of Technology, lecture notes 2003

Appendix A

Technical drawings

Part#	Name	Material	Amount
1	Actuator Base	Aluminum	1
2	Baseplate	Aluminum	1
3	Beam	Aluminum	1
4	E-actuator clamp1	Aluminum	2
5	E-actuator clamp2	Aluminum	2
6	E-actuator clamp3	Aluminum	2
7	Round actuator clamp	Aluminum	4
8	Round core	Soft Iron	2
9	Stack mount	Aluminum	1
10	Flexures	Aluminum	1
11	Mount Plate	Aluminum	1
12	Stack mount	Aluminum	1
13	TBS nut	Stainless steel	1
14	TBS Thread	Stainless steel	1
15	Support1	Aluminum	2
16	Support2	Aluminum	1
17	Support3	Aluminum	1
18	Support4	Aluminum	1
19	Support5	Aluminum	1
20	Support6	Aluminum	1
21	Endblock	Aluminum	1



#	X	Y
1	0.500	2.727
2	0.500	5.273
3	2.250	6.523
4	3.313	0.443
5	4.875	5.773
6	4.875	7.273
7	6.125	5.273
8	6.125	2.727
9	21.500	4.750
10	22.500	4.750
11	25.500	4.750
12	25.500	3.250
13	22.500	3.250
14	21.500	3.250

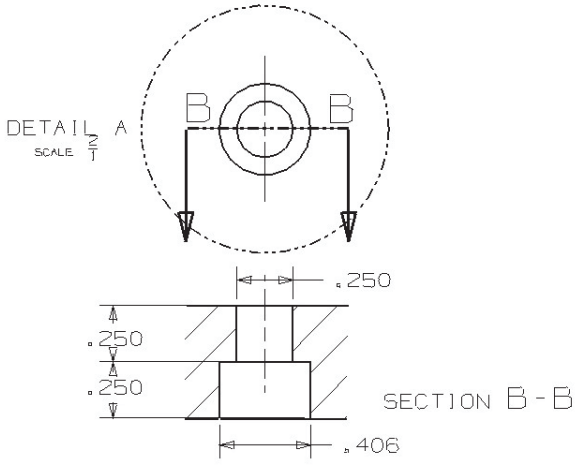


Figure A.1: Part# 2, Baseplate

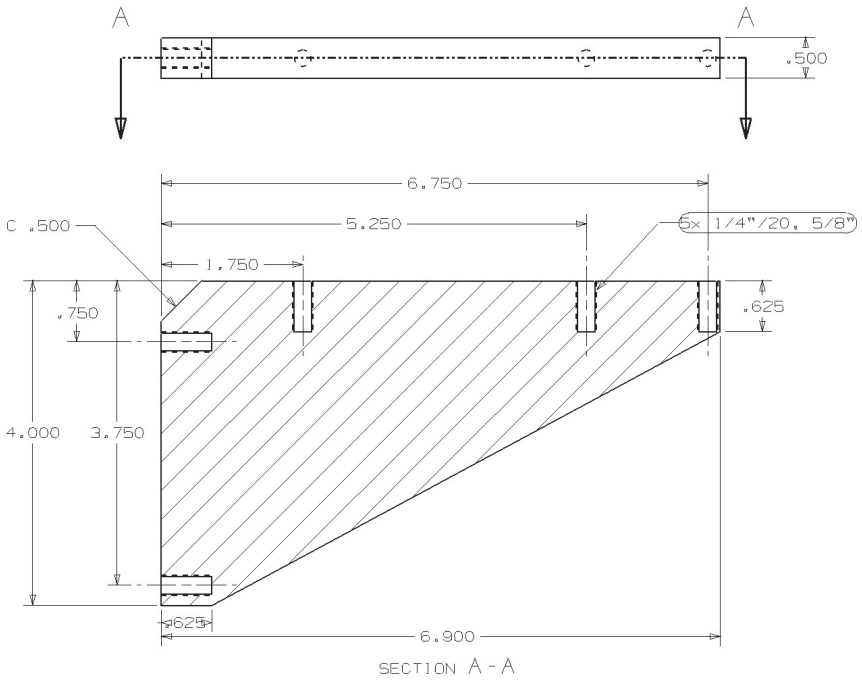


Figure A.2: Part# 15, Support1

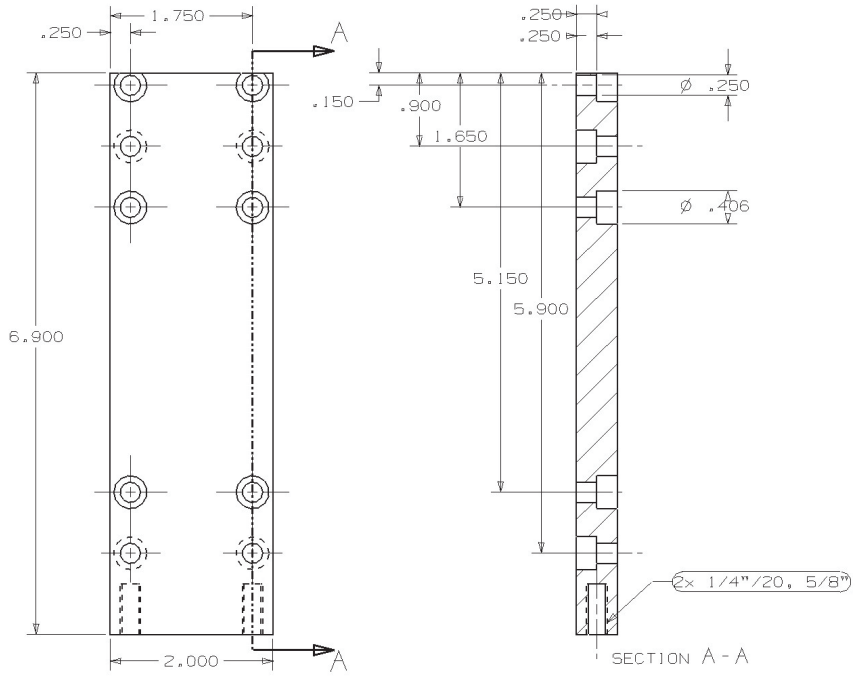


Figure A.3: Part# 11, Mount plate

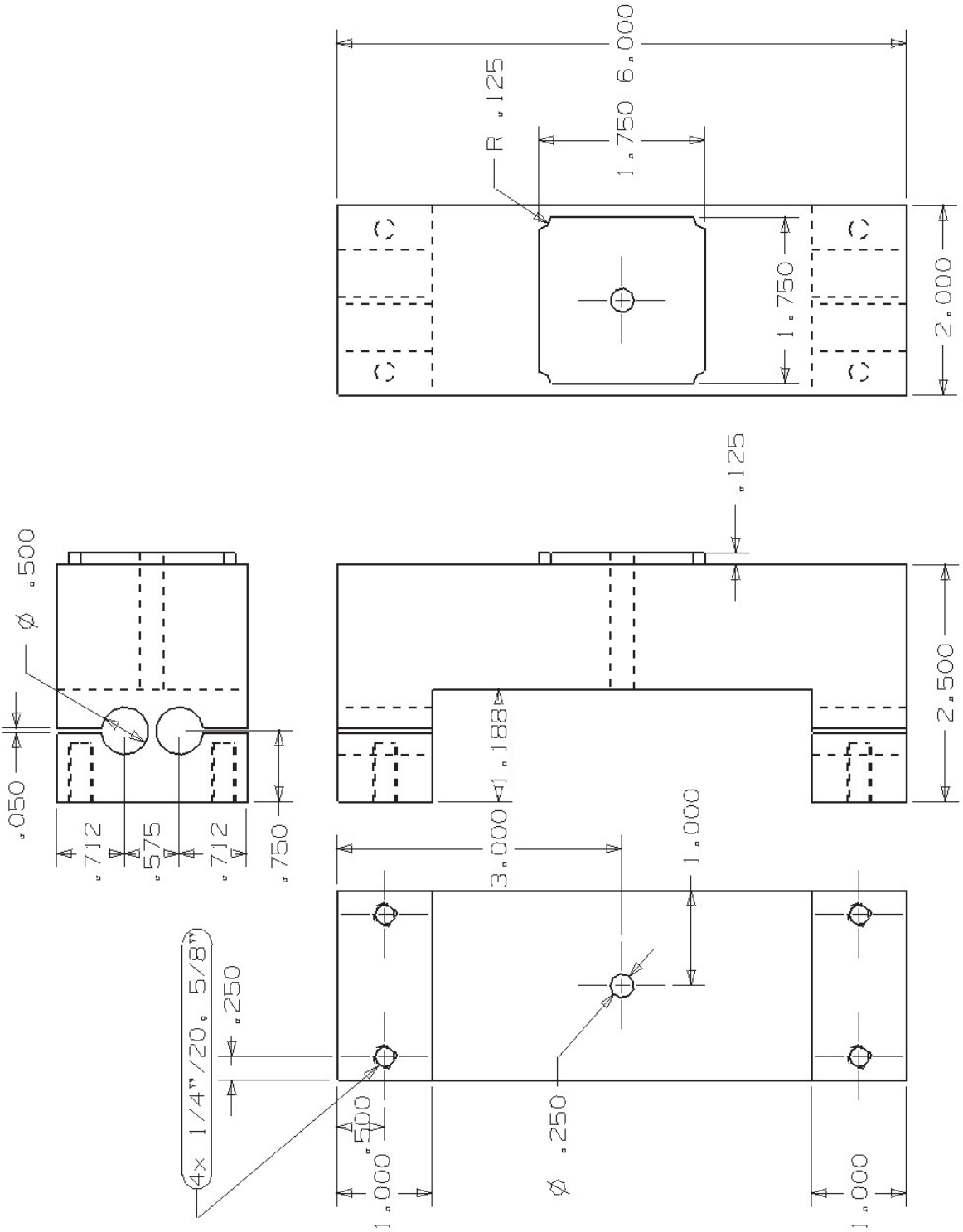


Figure A.4: Part# 10, Flexures

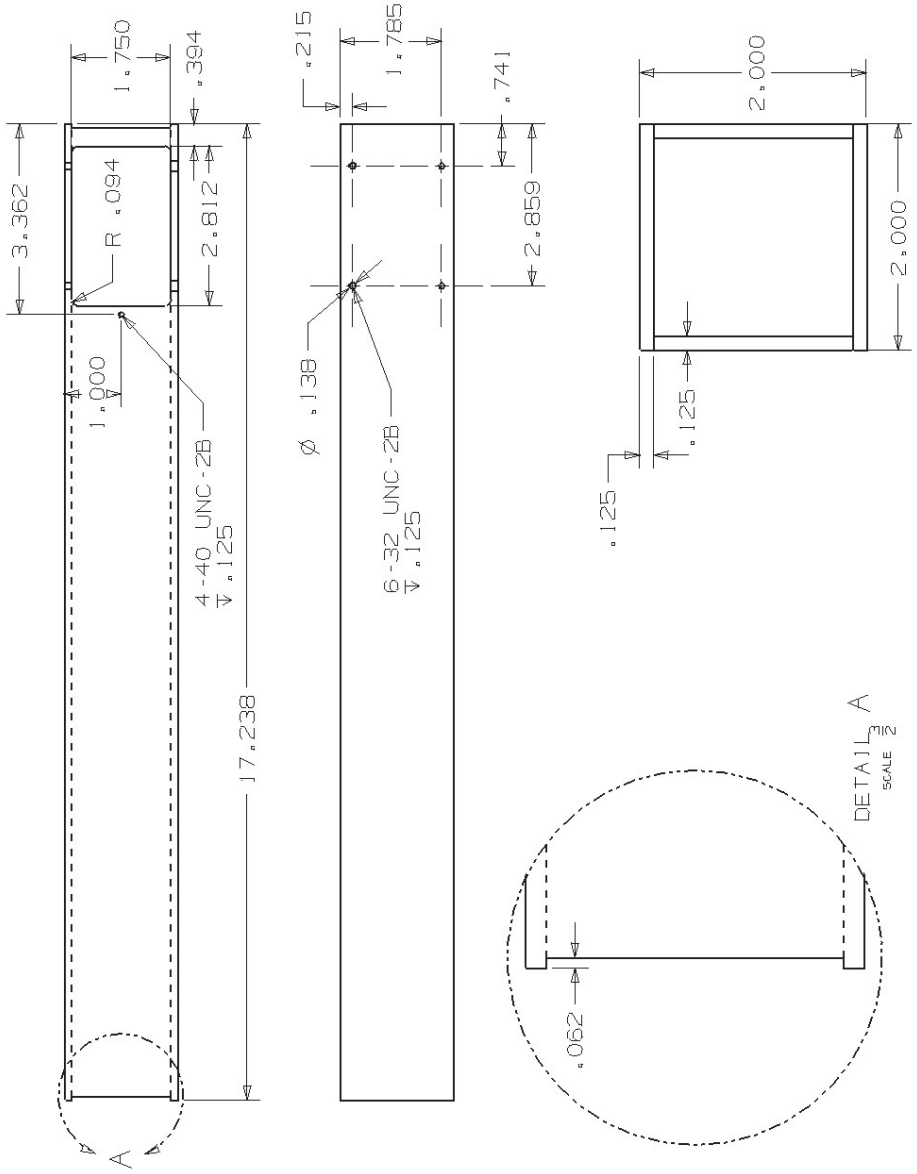


Figure A.5: Part# 3, Beam

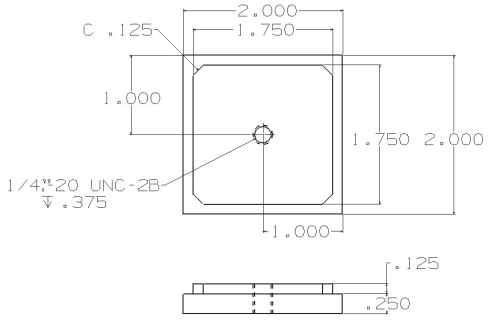


Figure A.6: Part# 21, Endblock

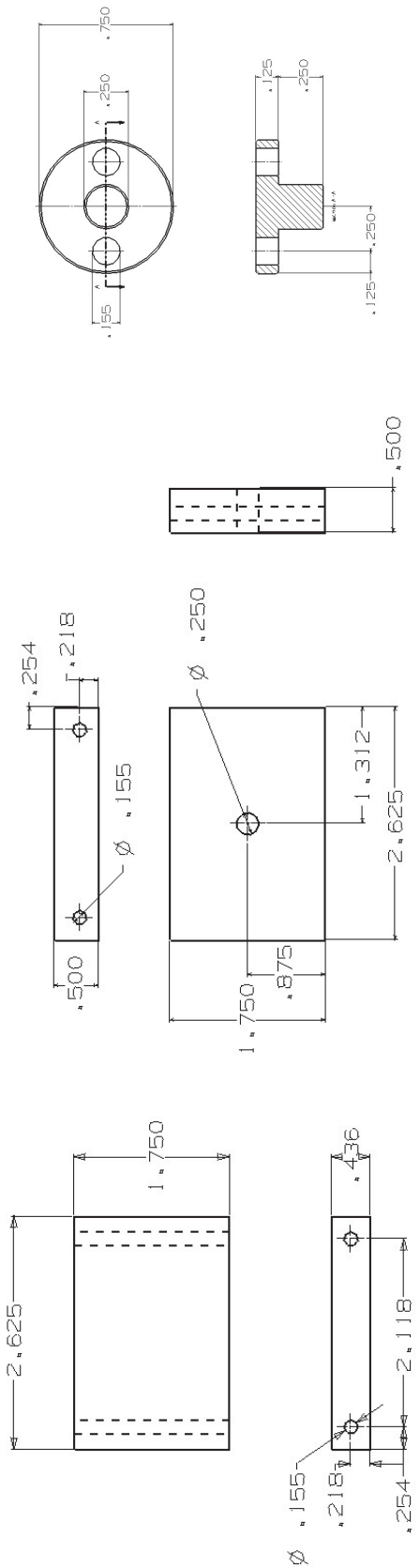
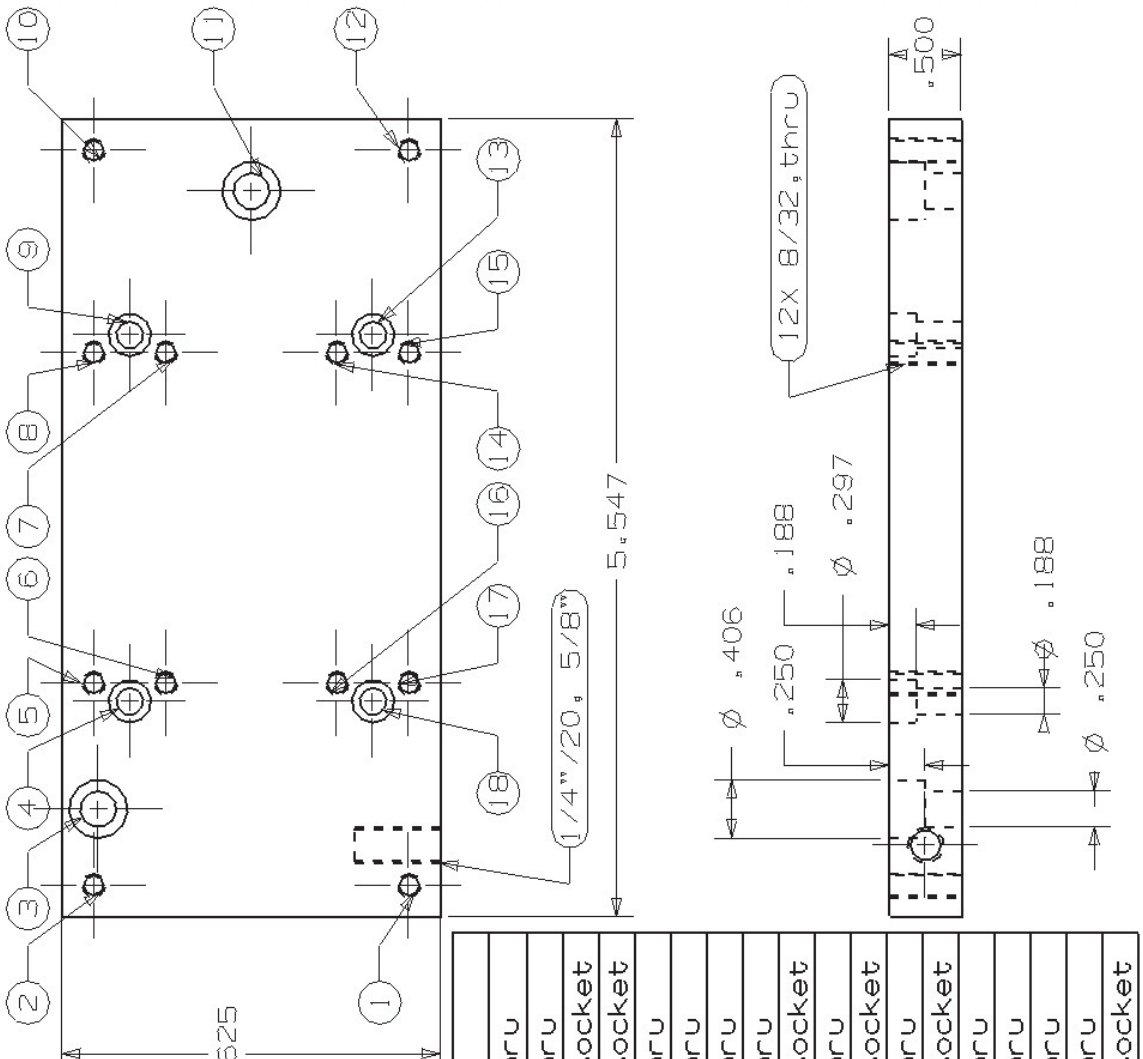


Figure A.7: PM-holder, magnetic targets, solid and laminated



#	X	Y	Type
1	0.219	0.219	8/32, thru
2	0.219	2.406	8/32, thru
3	0.750	2.375	0.250" socket
4	1.500	2.156	0.164" socket
5	1.625	2.406	8/32, thru
6	1.625	1.906	8/32, thru
7	3.922	1.906	8/32, thru
8	3.922	2.406	8/32, thru
9	4.047	2.156	0.164" socket
10	5.328	2.406	8/32, thru
11	5.047	1.312	0.250" socket
12	5.328	0.219	8/32, thru
13	4.047	0.469	0.164" socket
14	3.922	0.719	8/32, thru
15	3.922	0.219	8/32, thru
16	1.625	0.719	8/32, thru
17	1.625	0.219	8/32, thru
18	1.500	0.469	0.164" socket

Figure A.8: Part# 1, Actuator base

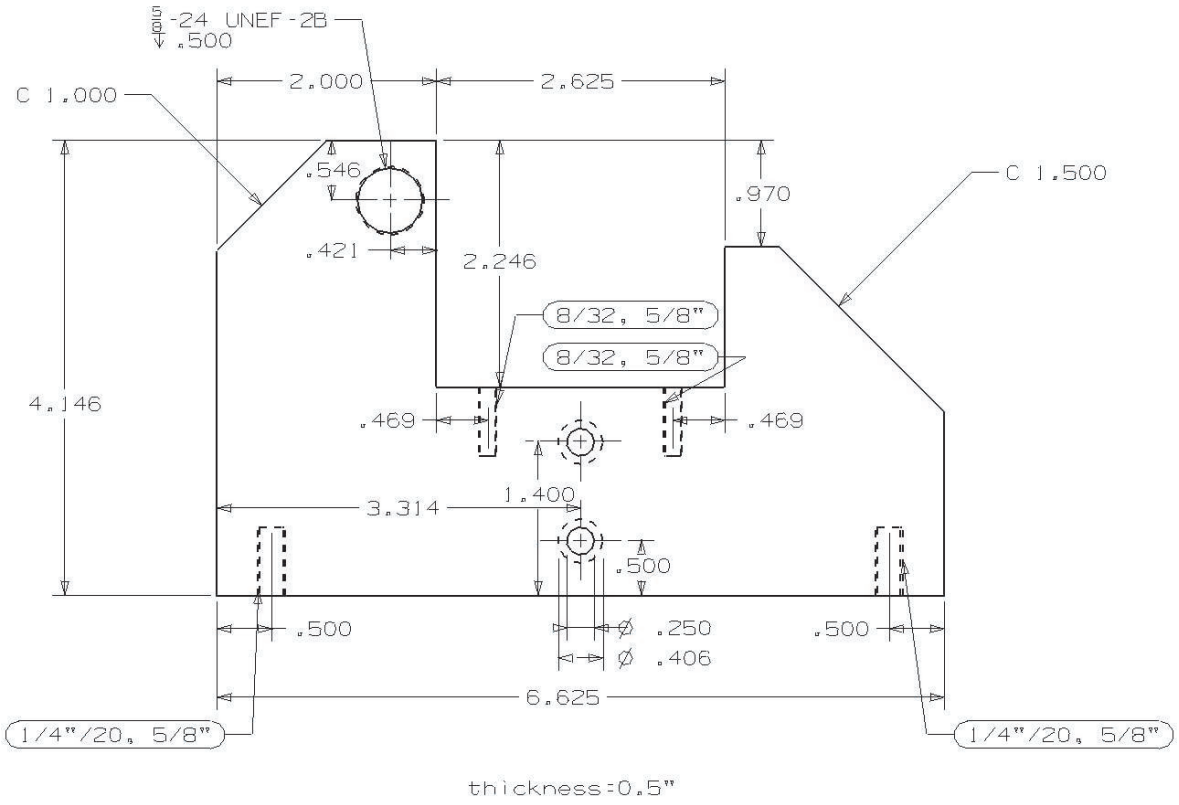


Figure A.9: Part# 16, Support2

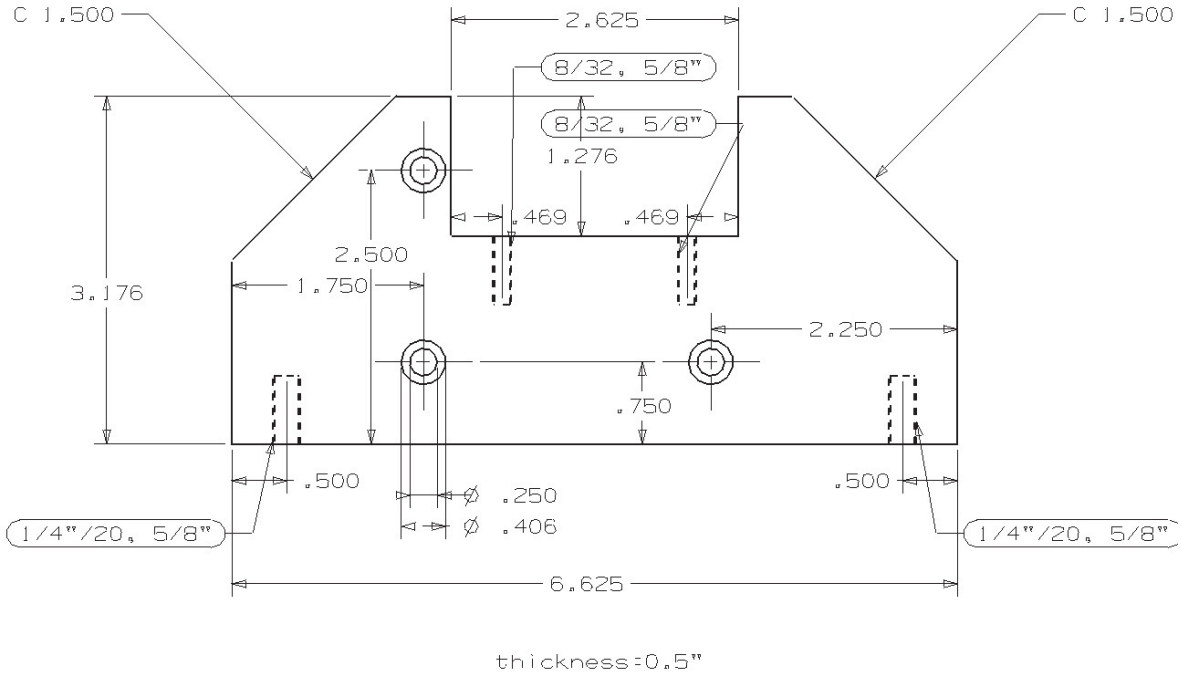


Figure A.10: Part# 17, Support4

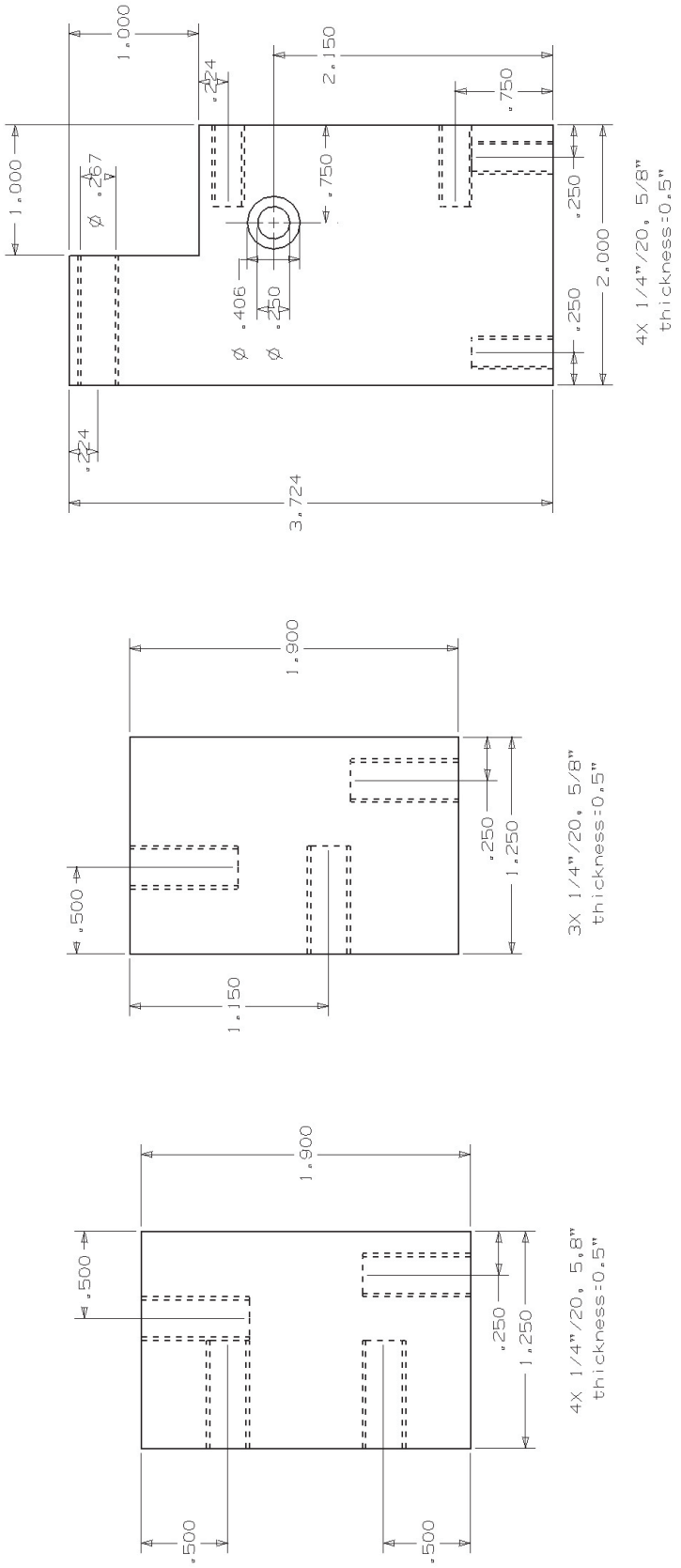


Figure A.11: Part# 18, 19, 20, Support 3, 5, 6

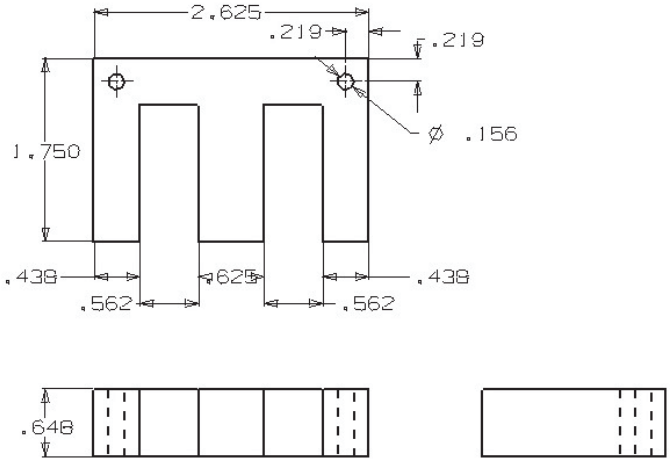


Figure A.12: E-shaped laminated core

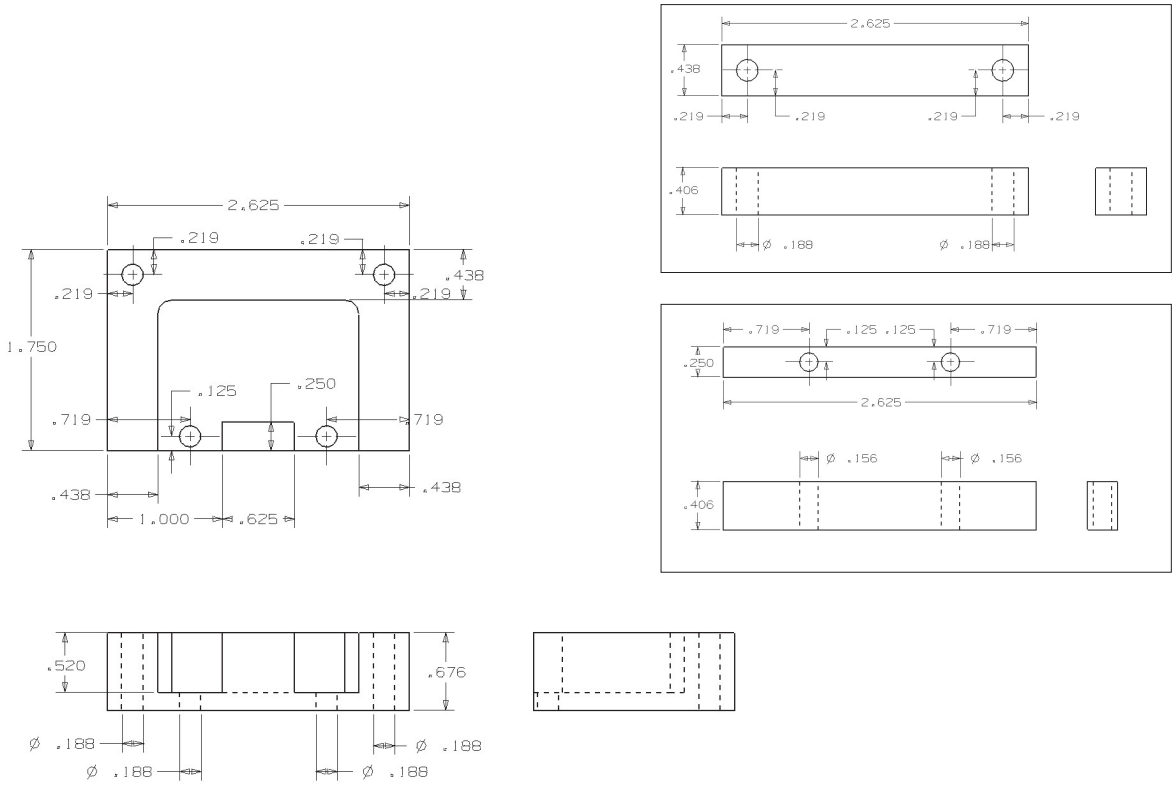


Figure A.13: Part# 4, 5, 6, E-actuator clamps 1, 2, 3

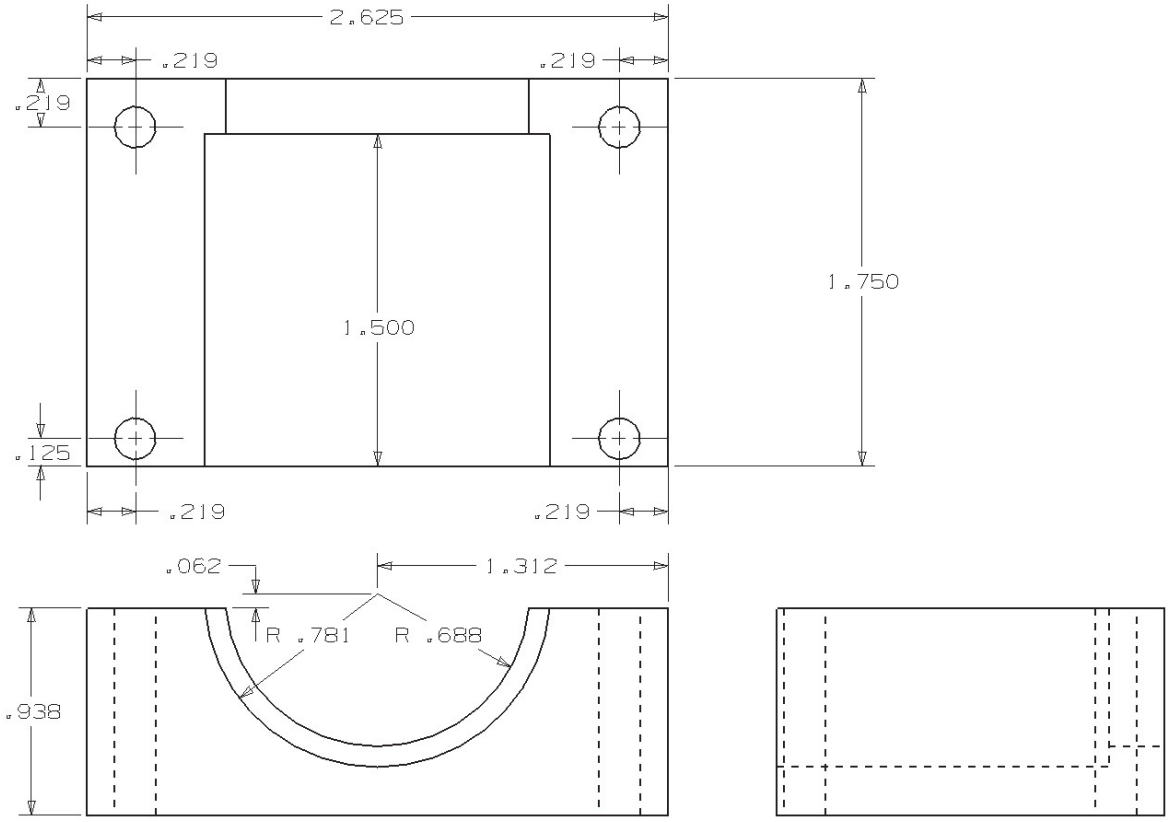


Figure A.14: Part# 7, Round actuator clamp

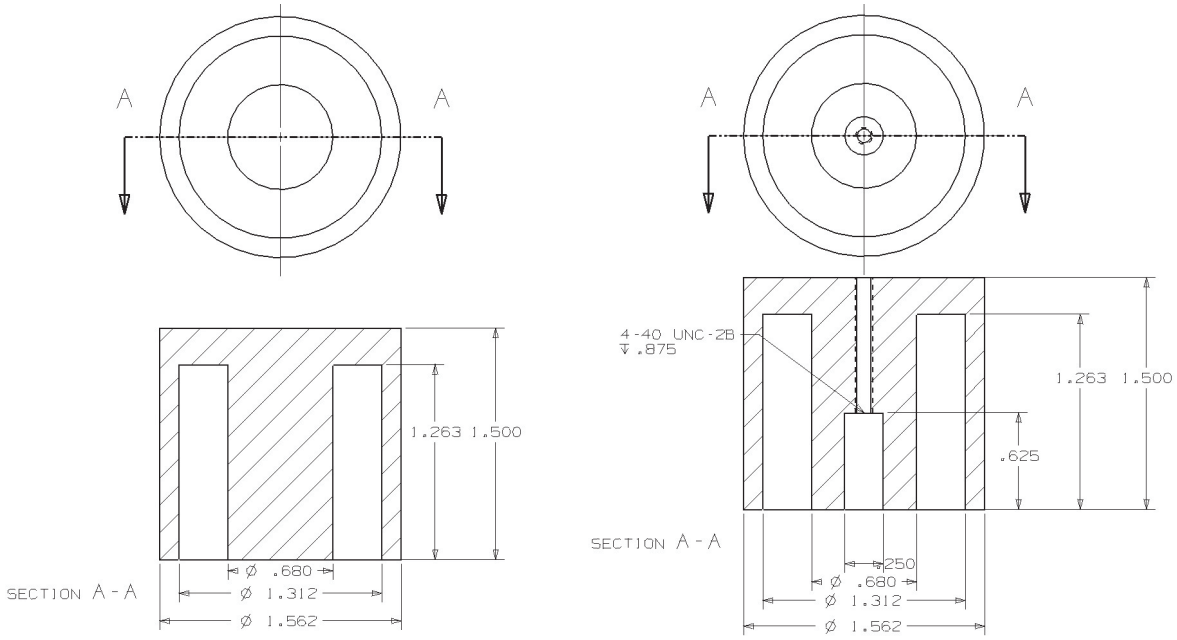


Figure A.15: Part# 8, Round actuator core

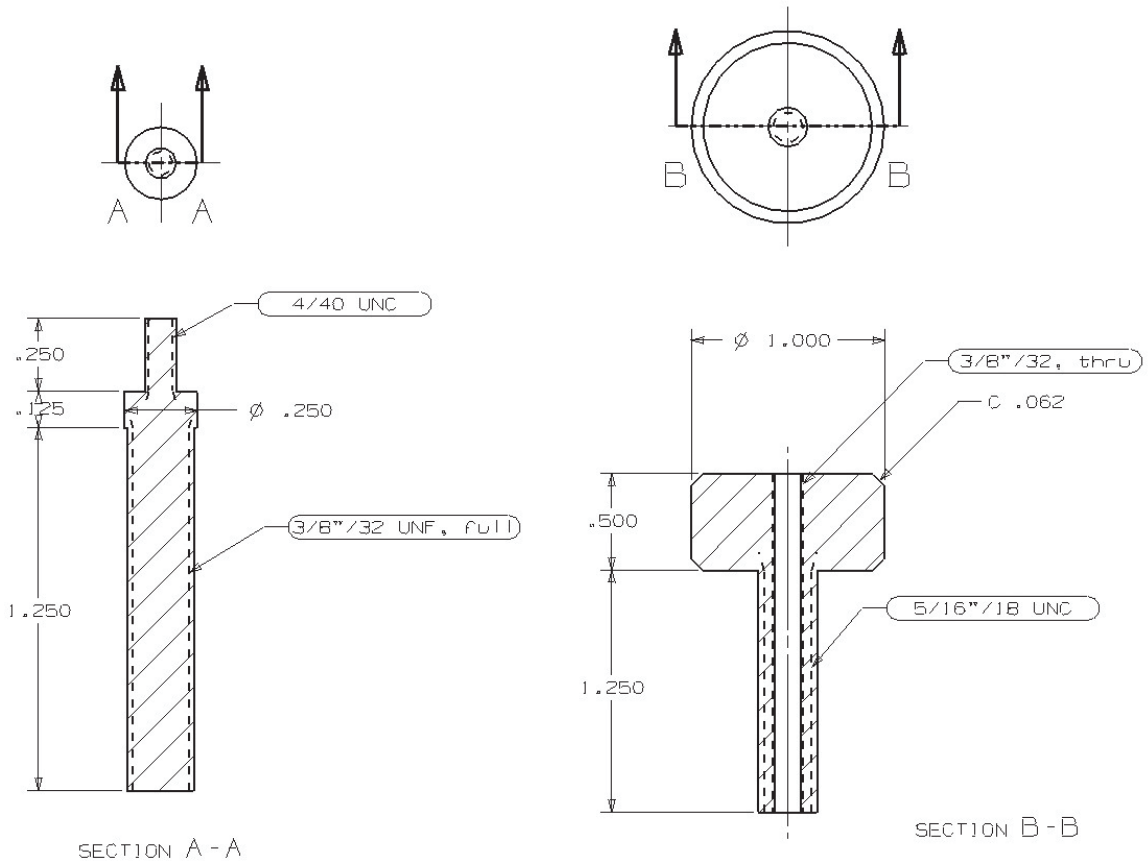


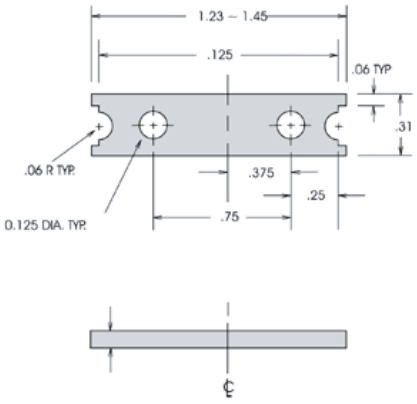
Figure A.16: Part# 13, 14, TBS differential thread

Appendix B

Hardware used

B.1 Transducer Techniques TBS-40 with TBS-MK2

Transducer Techniques®



SPECIFICATIONS

Rated Output (R.O.):	2 mV/V nominal
Combined Error:	0.25% of full scale
Zero Balance:	± 0.3 mV/V
Compensated Temp. Range:	20° to 120°F
Temp Effects:	Zero Balance 0.02% of FS/°F, Output 0.02%/°
Resistance:	(Input and Output) 1200 ohms ± 300 ohms
Insulation Resistance:	1000@ 50 VDC
Ecitation Voltage:	10 VDC
Safe Overload:	150% FS
Full Safe Deflection:	0.010 to 0.0500
Lead Wire:	9' sheilded PVC four conductor 30 AWG
Materials:	301 SS(berryllium copper 1/4 and 1/2 lb. units)
Deflection Inches:	.025

B.2 Connections

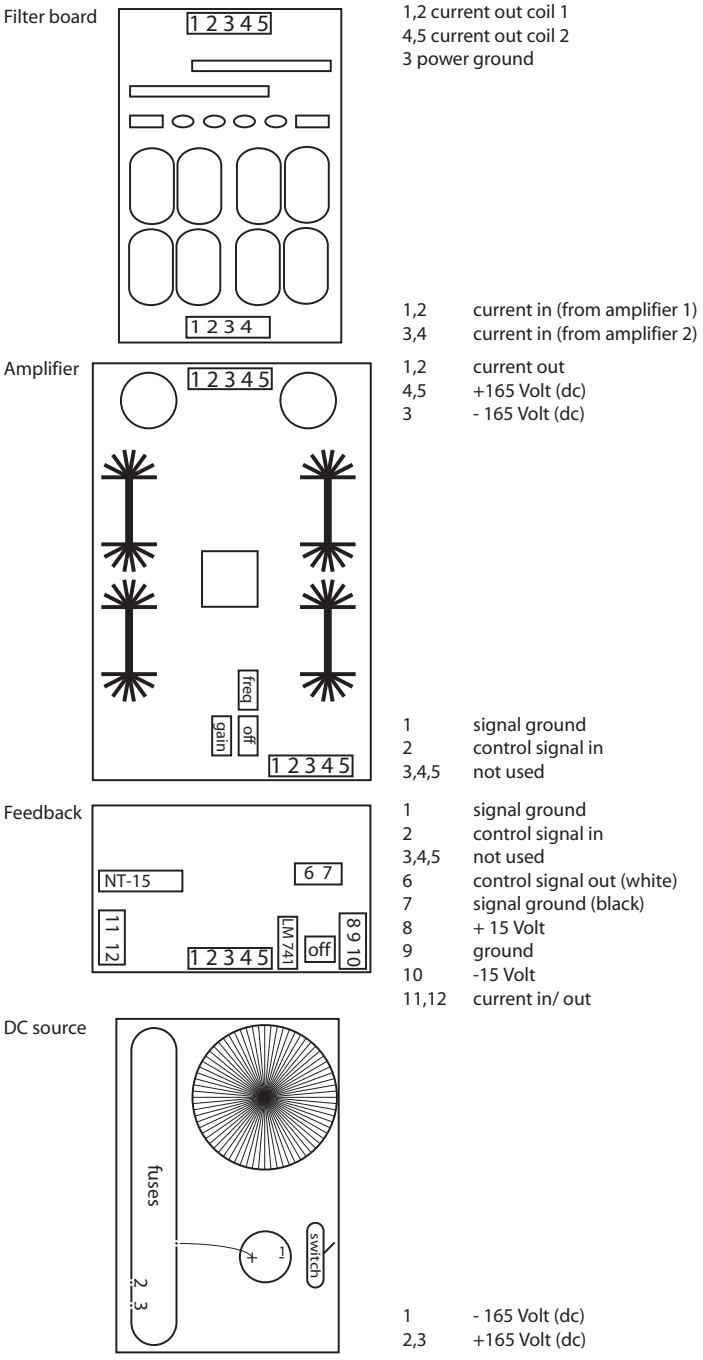


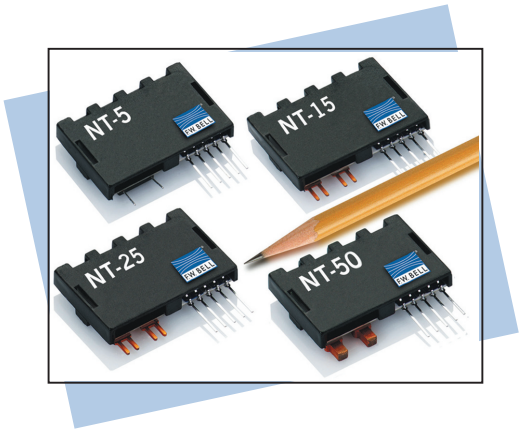
Figure B.1: Connections between the different components

B.3 Current sensors

NT Series

Current Sensors

Magneto-Resistive Current Sensors for Peak Currents up to 150 A



- Excellent Accuracy
- No Field Concentration
- Small and Compact Design
- Same Shape for all Current Ranges

F.W. BELL®

SYPRIS TEST & MEASUREMENT
6120 Hanging Moss Road • Orlando, Florida 32807
Phone (407) 678-9748 • Fax (407) 677-5765 • www.fwbell.com

Rev. date 04/2003

Magneto-Resistive Current Sensors

*The global miniaturization of electric systems in modern industrial applications is setting new challenges regarding the cost and size of electronic components and sensing devices. With NT Series current sensors from **F.W. Bell**, a surface of only 2.6 cm² on a printed circuit board is necessary for a potential free and galvanically isolated measurement of DC, AC and impulse currents up to 150 Apk. Due to the high sensitivity of the anisotropic magneto-resistive (AMR) effect, there is no need for a magnetic field concentrator around the primary conductor. Therefore, no remanence occurs. The small weight of the NT Series is another advantage compared to conventional current measurement methods. The high overall accuracy is partially given by the measurement resistor that is integrated in the system. The output voltage, measured across that resistor, is directly proportional to the primary current.*

The MR Effect

In thin films of permalloy (Fe-Ni) material, the electrical resistance changes when an external magnetic field is applied in the plane of the film. This change is due to the rotation of the film magnetization. The variation of the resistance due to an external field is called the anisotropic magneto-resistive (AMR) effect. Due to a special design of the chip, the resistance change is proportional over a wide range of measured field.

Current Measurement with MR

The operating principle of F.W. Bell NT current sensors is based on a differential magnetic field measurement with compensation (Fig. 1). The primary current is fed through a U-shaped conductor, creating a field gradient H_{prim} between the two sides of the conductor. The thin film magnetoresistors are placed on a silicon chip and connected in a Wheatstone bridge. The chip is mounted together with the analog interface electronics on a single in-line hybrid circuit. In order to obtain a high linearity (0.1%) and a low temperature sensitivity, a current I_S is fed back to the sensor chip through a compensation conductor located above the magnetoresistors. The resulting field H_{comp} exactly compensates H_{prim}, so that the sensors always work around a single operating point. At the output of the sensors, the compensation current flows through a measurement resistor R_M. The output voltage, measured across that resistor, is $V_{OUT} = \pm 2.5V$ at $\pm I_{PN}$. The nominal current is only determined by the geometry of the primary current conductor.

Wide Measurement Range

The NT Series is made of four different sensors with nominal currents of 5, 15, 25 and 50 A, respectively. Each sensor can measure up to 3 times the nominal current for a period of 3 seconds. The NT Series has been designed for the measurement of DC, AC and impulse currents from a few milliampere up to 150 A. The isolation voltage between the primary current conductor and the interface electronics is 3.5kV $_{eff}$ (50/60 Hz for 1 min.).

Small and Compact Design

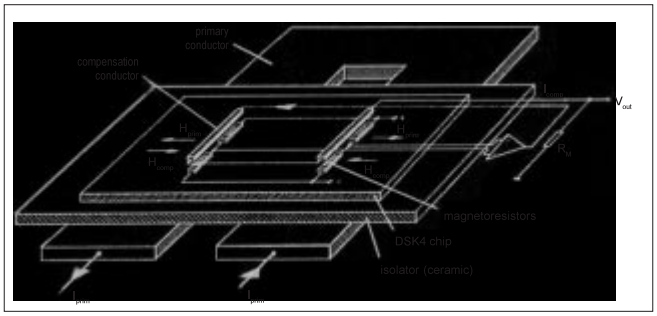
The sensitivity of MR sensors is almost 50 times higher then that of Hall sensors. A concentration of the measured magnetic field around the primary conductor with a ferrite or iron core is therefore not necessary. This results in very small and light weight (4g...6.5g) current sensor microsystems. The surface used by the NT on the PCB board is only 35 x 7.3 mm². This corresponds to approximately 1/3 of the surface required by conventional current sensors available on the market.

High Accuracy

The basic accuracy of the NT is 0.3% of I_{PN}. Since no field concentration is used, F.W. Bell current sensors have no remanence. This positively affects the overall accuracy. Since the output of the NT product family is a voltage, the error of the measurement resistor R_M is already included in the overall error of 0.8% (basic accuracy X + offset voltage V_O + error of the measurement resistor R_M) at I_{PN} and room temperature.

Large Spectrum of Applications

The NT Series offers a cost effective solution for current sensing in a variety of standard as well as custom specific industrial applications.



The easy through-hole mounting in a printed circuit board and the small footprint are significant advantages in applications like servo or 3-phase current drives as well as frequency inverters for DC drives. Other applications include mains adapters, uninterrupted power supplies, battery powered applications, building control and automation. Whenever DC, AC and impulse currents have to be measured, F.W. Bell current sensors are there.

Rev. date 04/2003

Magneto-Resistive Current Sensors

For the potential free measurement of electric currents (DC, AC, impulse…) with the magneto-resistive (AMR) technology. The nominal currents are 5, 15, 25 and 50 A, with a galvanic isolation between the primary current and the output signal.

Applications
- 3 phase current drives and servo drives
- Frequency inverters for DC drives
- Mains adapters
- Uninterruptable power supplies
- Battery powered applications
- Solar technology
- Building control and automation
- Welding equipment

Advantages
- Excellent accuracy, linearity and dynamics
- Small and compact design
- Light weight (0.14….0.23 oz)
- Small sensitivity to interferences
- Internal measurement resistor R_M
- No field concentration, therefore no remanence

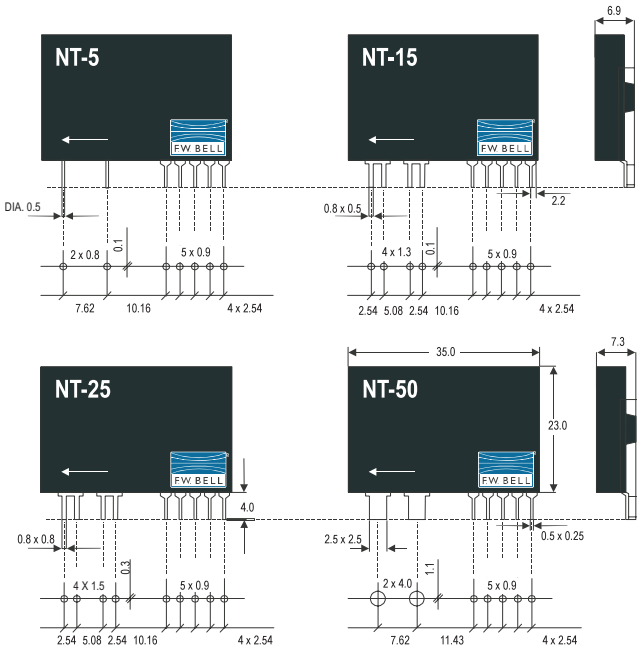
		Unit	NT-5	NT-15	NT-25	NT-50
Electrical Data	Primary nominal current, I_{PN}	A	5	15	25	50
	Primary current measurement range[1]	A	0... ± 15	0... ± 45	0... ± 75	0... ± 150
	Overload[2]	A	10 X I_{PN}			
	Output voltage at ± I_{PN}	V	± 2.5			
	Internal resistor of the NT	Ω	< 150			
	Supply voltage ± 5%[3]	V	± 12... ± 15			
	Power consumption @ I_{PN}	mA	< 40			
	Resistance of the primary conductor	mΩ	< 12	< 1	< 0.5	< 0.15
	Isolation test voltage, effective	kV	3.5			
	Measurement tension[4]	V	600			
Accuracy	Accuracy[5] at I_{PN} and room temp.	%	< ± 0.3			
	Overall accuracy at I_{PN} and room temp.	%	< ± 0.8			
	Linearity	%	< ± 0.1			
	Typical offset voltage at room temp.	mV	± 7.5			
	Sensitivity drift[6]	%/°C	± 0.01			
	Max. offset over temp. (- 25°C... + 85°C)	mV	± 35			
Dynamic Data	Reaction time (10% of I_{PN})	µs	< 0.15			
	Rise time (10%... 90% of I_{PN})[7]	µs	< 1.7	< 1.7	< 1.2	< 1.0
	Frequency range (deviated amplitude)	kHz	DC... 100			
General Data	Temperature range	°C	- 25°C... + 85°C			
	Storage temperature	°C	- 25°C... + 100°C			
	Mass	g (oz)	4.0 (0.14)	4.2 (0.15)	4.5 (0.16)	6.5 (0.23)
	Dimensions	mm (in)	35 x 23.5 x 7.3 (1.4 x 0.92 x 0.29)			
	Surface on PCB board	mm^2 (in^2)	256 (0.4)			
	Isolated, self-extinguishing housing material		UL 94-VO			
	Standards		EN50178 • EN61010 • CE-sign			

NOTES:
[1] For 3 s; I_P = 2 X I_{PN} for 10 s
[2] For 20 ms, then 20 s max. I_{PN}
[3] At V_c = ±12 V: I_P = 2 X I_{PN}. Restrictions on accuracy and dynamic range
[4] Pollution degree 2, cat. II
[5] Without offset V_O and tolerance error of the measurement resistor R_M
[6] Only dependent of the TC of the measurement resistor R_M
[7] With di/dt = 100A/µs

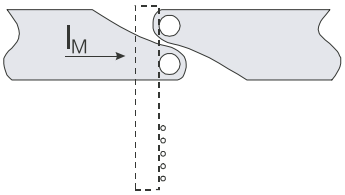
Rev. date 04/2003

Mechanical **Dimensions**

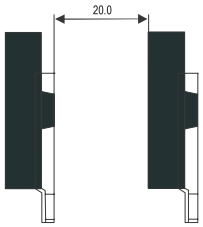
Dimensions with drilling plans



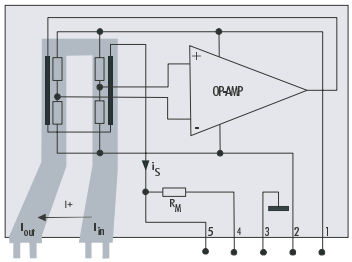
Recommended current path layout



Recommended minimal distance



Pinning



Pin 1 = $+V_S$
2 = $-V_S$
3 = SUPPLY GND (PINS 3,4 SHOULD BE CONNECTED TOGETHER)
4 = OUTPUT GND
5 = V_{OUT}

I_{in} = current input
I_{out} = current output
← = positive current direction

All dimensions in mm

Note: Due to continuous process improvement, specifications subject to change without notice.

SYPRIS TEST & MEASUREMENT
6120 Hanging Moss Road • Orlando, Florida 32807
Phone (407) 678-9748 • Fax (407) 677-5765 • **www.fwbell.com**

F.W. BELL®

Rev. date 04/2003

B.4 Electrical scheme feedback amplifiers

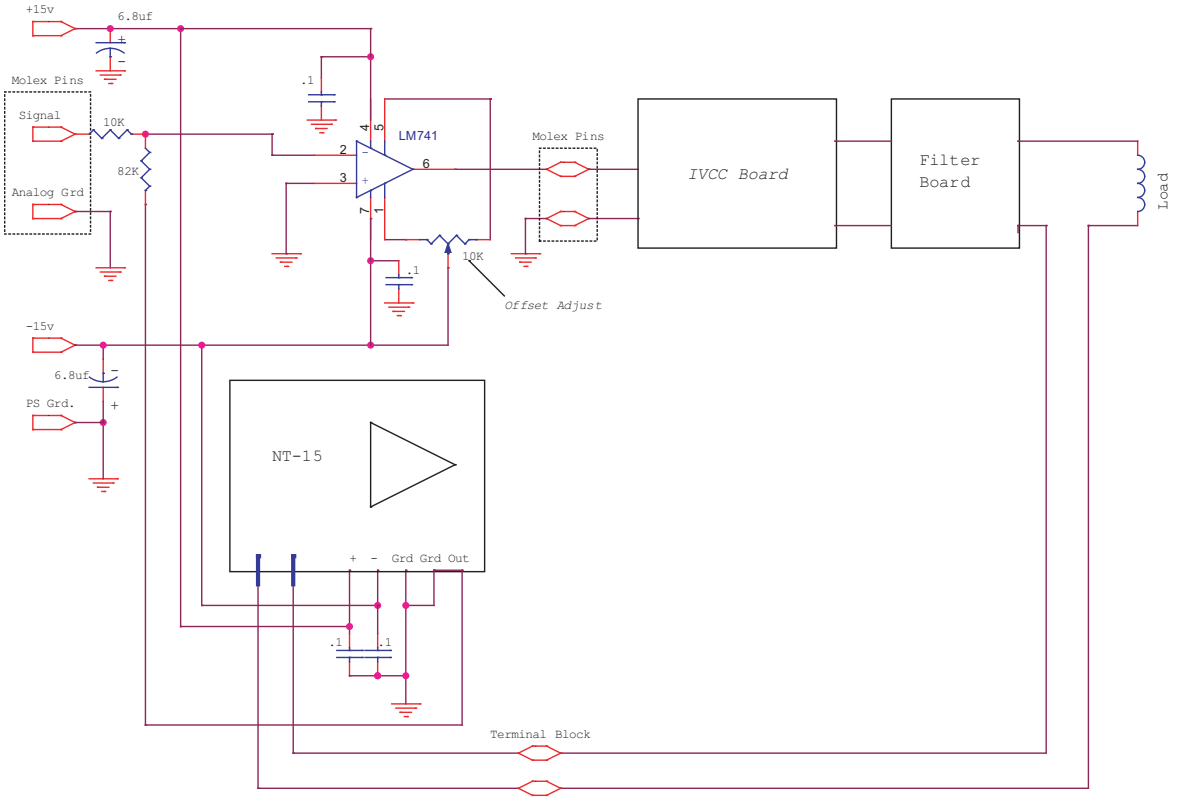


Figure B.2: Electrical scheme feedback amplifiers

Appendix C

Measurements

C.1 Permanent magnets

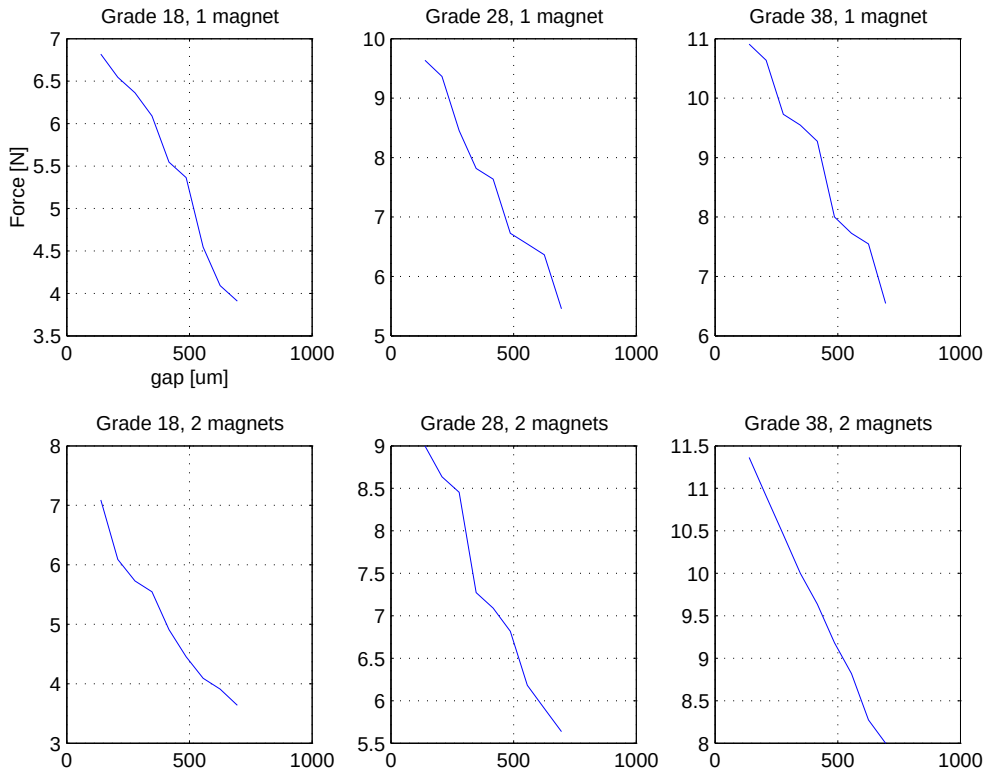


Figure C.1: Measurements all different permanent magnet configurations

C.2 Actuators with solid target

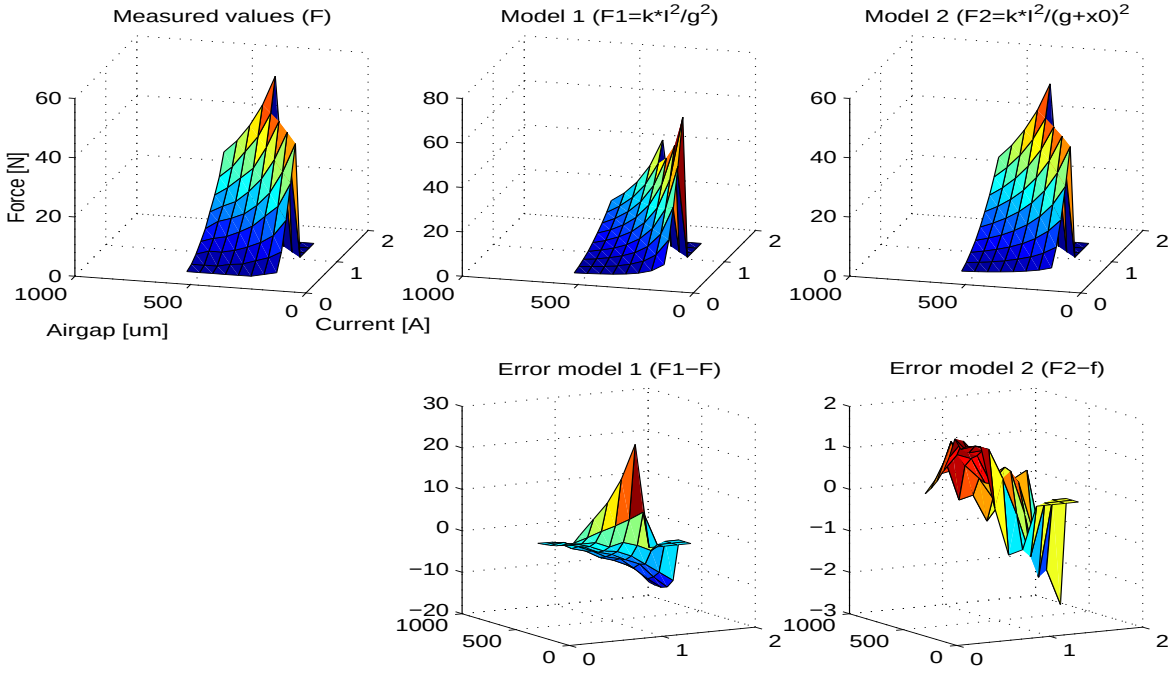


Figure C.2: E-type actuator with solid target

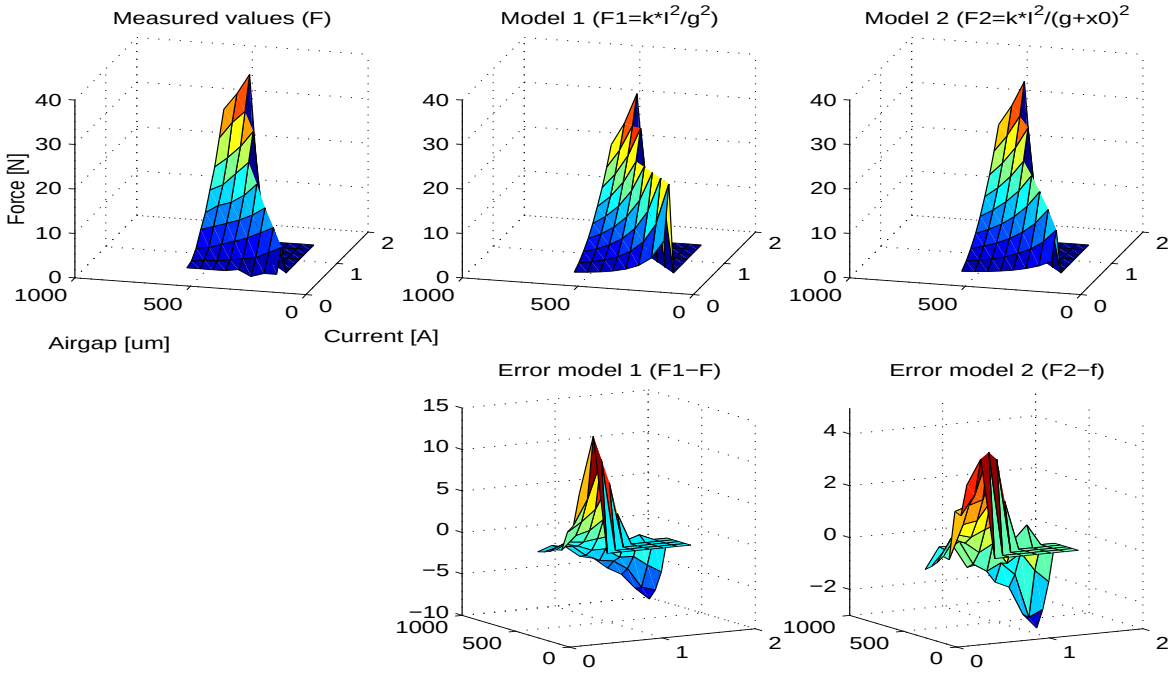


Figure C.3: Round actuator with solid target

Appendix D

M-files

D.1 M-file theoretical force model

D.1.1 E shaped actuator

```
% ========================================
% Theoretical model of the force of one E-shaped coil
% ========================================
close all;
clear all;
clc;

N   = 258;

%      S3                      S4
%              a               h
%     ================       ====
%     ================ d S2  ====
%     ================       ====
%     <-b-->                 ====
%     ======  -----          ====
%     ========-------       ====
%   e ==<-k->=------=g= c S1  ====  A
%     ==A=====-------       ====  |
%     ==|===  -----          ====  |
%     ==|i==                 ====  | i
%     ==|=============       ====  |
%     ==|============= d S2  ====  |
%     ==V=============       ====  V
%              j        <--->             Thickness = f
%                         x
%      Actuator Dimensions
```

```
%    ========R1=========Ra1===
%    ==                     ==
%    ==                     R2
%    ==   |                 ==
%    == | |                 ==
%    ===|=|==R3=========Ra2===
%    == | |                 ==
%    ==   |                 ==
%    ==  F                  R2
%    ==                     ==
%    ========R1=========Ra1===
%
%      Equivalent circuit

a   = 1.753*25.4e-3;            % [m]
b   = 0.455*25.4e-3;            % [m]
c   = 0.631*25.4e-3;            % [m]
d   = 0.433*25.4e-3;            % [m]
e   = 2.625*25.4e-3;            % [m]
f   = 0.648*25.4e-3;            % [m]
g   = 0.325*25.4e-3;            % [m]
h   = 0.500*25.4e-3;            % [m]
i   = (0.5*e-0.5*d)*25.4e-3;    % [m]
j   = (a-0.5*b)*25.4e-3;        % [m]
k   = 0.272*25.4e-3;            % [m]

Volume = (2*((a-b)*d)+c*(a-b)+b*e)*f;
Mass = 0.2275;

Density =           Mass/Volume;  % lower then orthosil due to glue
Density_orthosil =  7700;
Density_percent =   Density/Density_orthosil;

f = Density_percent*f;

S1  = c*f;  %[m^2]
S2  = d*f;  %[m^2]
S3  = b*f;  %[m^2]
S4  = h*f;  %[m^2]

% Constants
mu0 = 4e-7*pi;
mur = 47000;         % From Thesis L.Fevre p.17
mu  = mu0*mur;

R1  = k/(mu*S1) + i/(mu*S3) + j/(mu*S2);
R2  = i/(mu*S4);
```

```
R3  = g/(mu*S1);

A   = (S1+2*S2)/(2*S1*S2);
Psum = k/(mu*S1) + i/(mu*S3) + j/(mu*S2) + i/(mu*S4) + 2*g/(mu*S1);

x   = [400e-6:10e-6:500e-6];
I   = [0:0.1:1.5];

for ii = 1:length(I);
for ix = 1:length(x);
    Ra1 = x(ix)/(mu0*S2);
    Ra2 = x(ix)/(mu0*S1);

%     Rt1 = R1 + R2 + Ra1;
%     Rt2 = R3 + Ra2;
%     Rt3 = 0.5*Rt1 + Rt2;


Ft(ii,ix)= (N^2*mu0/(2*A)) * (I(ii)^2/(Psum/(2*A*mur)+x(ix))^2);

end
end

figure(1)
surf(x,I,abs(Ft))
%axis([5e-4 15e-4 0 2 0 40])
hold on
grid on
xlabel('Gap [m]')
ylabel('Current [A]')
zlabel('Force [N]')
```

D.1.2 Round actuator

```
% ========================================
% Theoretical model of the force of one round shaped coil
% ========================================
close all; clear all; clc;

N   = 270;

%      S3                      S4
%              a               h
%     ================       ====
%     ================ d S2   ====
%     ================       ====
%     <-b-->                 ====
%     ======----------       ====
%     =======---------       ====
%   e ==<-k->--------g c S1   ====   A
%     ==A===----------       ====   |
%     ==|===----------       ====   |
%     ==|i==                 ====   | i
%     ==|=============       ====   |
%     ==|============= d S2   ====   |
%     ==V=============       ====   V
%              j        <--->              Thickness = f
%                         x
%      Actuator Dimensions (round!!!)

%    ========R1=========Ra1===
%    ==                     ==
%    ==                     R2
%    ==   |                 ==
%    == | |                 ==
%    ===|=|==R3=========Ra2===
%    == | |                 ==
%    ==   |                 ==
%    ==  F                  R2
%    ==                     ==
%    ========R1=========Ra1===
%
%      Equivalent circuit

a   = 1.500*25.4e-3;            % [m]
b   = 0.237*25.4e-3;            % [m]
c   = 0.680*25.4e-3; % diameter % [m]
d   = 0.125*25.4e-3;            % [m]
e   = 1.562*25.4e-3; % diameter % [m]
```

```
f   = 0.648*25.4e-3;            % [m]
g   = 0.029*25.4e-3;            % [m]
h   = 0.500*25.4e-3;            % [m]
i   = e/2*25.4e-3;              % [m]
j   = (a-0.5*b)*25.4e-3;        % [m]
k   = (1/2*b+g)*25.4e-3;        % [m]

S1  = 1/4*pi*c^2;               %[m^2]
S2  = 1/4*pi*(e^2-(e-2*d)^2);   %[m^2]
S3  = S1;                       %[m^2]
S4  = h*f;                      %[m^2]

% Constants
mu0 = 4e-7*pi;
mur = 4000;    % From Thesis L.Fevre p.17
mu  = mu0*mur;

R1  = k/(mu*S1) + i/(mu*S3) + j/(mu*S2);
R2  = i/(mu*S4);
R3  = g/(mu*S1);

A   = (S1+S2)/(S1*S2);
Psum = k/(mu*S1) + i/(mu*S3) + j/(mu*S2) + i/(mu*S4) + g/(mu*S1);

x   = [400e-6:10e-6:500e-6];
I   = [0:0.1:1.5];

for ii = 1:length(I);
for ix = 1:length(x);
    Ra1 = x(ix)/(mu0*S2);
    Ra2 = x(ix)/(mu0*S1);

%     Rt1 = R1 + R2 + Ra1;
%     Rt2 = R3 + Ra2;
%     Rt3 = Rt1 + Rt2;

Ft(ii,ix)= (N^2*mu0/(2*A)) * (I(ii)^2/(Psum/(2*A*mur)+x(ix))^2);
end
end

figure(1)
mesh(x,I,abs(Ft))
hold on
grid on
xlabel('Gap [m]')
ylabel('Current [A]')
zlabel('Force [N]'
```

D.2 M-file used for fitting force model

```
%=========================================================
% Round-actuator
% tbs 40 sensor
% 7 March 2005
%=========================================================

i=[0.3     0.45    0.6     0.75    0.9     1.05    1.2     1.35    1.5];
g=[150     200     250     300     350     400     450     500];
F=[0       0       0       0       0       0       0       0       0;
   12      0       0       0       0       0       0       0       0;
   7       18      0       0       0       0       0       0       0;
   5       13      23      37.5    0       0       0       0       0;
   4       10      18.5    28      44.5    0       0       0       0;
   3.5     8       14      23      35      46      62      0       0;
   3.5     7       12      19      27      37      49      64      75;
   3       6       11      17.5    24      31      41      52      63];

% Due to position of force sensor the force acting on actuator is 10 percent lower.
F=F/1.1;
g=g*1.112;

%=========================================================
% Model Fit for Round_actuator
% 8 March 2005
% F=K* I^2/g^2
%=========================================================

syms K;
ni=0;
errsum=0;

for I=i;
    ni=ni+1;
    ng=0;
    for G=g;
        ng=ng+1;
        Fmodel1(ng,ni)=K*(I^2/(G*1e-6)^2);          % Model
        Fmodel2(ng,ni)=Fmodel1(ng,ni);              % Copy for not removing zeros
        if F(ng,ni) == 0
           Fmodel1(ng,ni)=0;
       end
       errsum=errsum+(Fmodel1(ng,ni)-F(ng,ni))^2;
    end
end
```

```
K_opt=double(solve(diff(sqrt(errsum)),K));          % Least squares-fit
Fmodel1=subs(Fmodel1,K,K_opt);
Fmodel2=subs(Fmodel2,K,K_opt);

%=========================================================
% Model Fit for Round_actuator
% 8 March 2005
% F=K* I^2/(g+x0)^2
%=========================================================
syms K x0;
ni=0;
errsum=0;

for I=i;
    ni=ni+1;
    ng=0;
    for G=g;
        ng=ng+1;
        Fmodel1_plus(ng,ni)=K*(I^2/(G*1e-6+x0)^2);      % Model
        Fmodel2_plus(ng,ni)=Fmodel1_plus(ng,ni);        % Copy for not removing zeros
        if F(ng,ni) == 0
           Fmodel1_plus(ng,ni)=0;
       end
       errsum=errsum+sqrt((Fmodel1_plus(ng,ni)-F(ng,ni))^2);
    end
end

% =========================================================
% Calculation and plots for Gain and Additional Gaplength Round_actuator
% 8 March 2005
% F=K* I^2/(g+x0)^2
% =========================================================

nKK=0;
Gain = 0.0e-6:2e-6:30e-6;
GapL = 10e-6:20e-6:900e-6;

for KK = Gain;
    KK
    nKK = nKK+1;
    nX0 = 0;
    for XX0 = GapL;
        nX0 = nX0+1;
        ERRsum(nKK,nX0)=subs(errsum,[K,x0],[KK,XX0]);
    end
end
```

```
figure(3)
contour(GapL,Gain,ERRsum,100);
title('Error function K and x0')

Gain_opt= 9.07e-6;                          % Resulting from manual readout from plots
x0_opt = 155e-6;                            % Resulting from manual readout from plots

Fmodel1_plus=subs(Fmodel1_plus,[K,x0],[Gain_opt,x0_opt]);
Fmodel2_plus=subs(Fmodel2_plus,[K,x0],[Gain_opt,x0_opt]);

figure
subplot(2,3,1)
surf(i,g,F);
title('Measured values (F)')
xlabel('Current [A]')
ylabel('Airgap [um]')
zlabel('Force [N]')
view(-60,20)
subplot(2,3,2)
surf(i,g,Fmodel1);
title('Model 1 (F1=k*I^2/g^2)')
xlabel('Current [A]')
ylabel('Airgap [um]')
zlabel('Force [N]')
view(-60,20)
subplot(2,3,3)
surf(i,g,Fmodel1_plus);
title('Model 2 (F2=k*I^2/(g+x0)^2')
xlabel('Current [A]')
ylabel('Airgap [um]')
zlabel('Force [N]')
view(-60,20)
subplot(2,3,5)
surf(i,g,Fmodel1-F);
title('Error model 1 (F1-F)')
xlabel('Current [A]')
ylabel('Airgap [um]')
zlabel('Force [N]')
view(-30,10)
subplot(2,3,6)
surf(i,g,Fmodel1_plus-F);
title('Error model 2 (F2-f)')
xlabel('Current [A]')
ylabel('Airgap [um]')
zlabel('Force [N]')
view(-30,10)
shg
```

D.3 M-file used for parameter estimation

```
%=========================================================
% Parameter estimation
%=========================================================
clc;
clear all;

%load measurements
load data_NL_30march;

%Filtering noise from signals
fitdata= idfilt(data(1:(length(data)/2),:),12,0.1);
valdata= idfilt(data((length(data)/2+1):length(data),:),12,0.1);

% Make coloms of measurements (only for EVEN measurement points)
Gapf=fitdata(:,1);
i_1f=fitdata(:,2);
i_2f=fitdata(:,3);

Gapv=valdata(:,1);
i_1v=valdata(:,2);
i_2v=valdata(:,3);

clear data

% Make real value of Gap
Gapf=(Gapf.*-250).*1e-6;
Gapv=(Gapv.*-250).*1e-6;
i_1f=i_1f./5;
i_2f=i_2f./5;
i_1v=i_1v./5;
i_2v=i_2v./5;


Ts = 1/1000;
Adgap=63e-6;
x0=500e-6;

fdata  = iddata(Gapf,[i_1f.^2./(x0+Adgap+Gapf).^2, i_2f.^2./(x0+Adgap-Gapf).^2],Ts);
vdata  = iddata(Gapv,[i_1v.^2./(x0+Adgap+Gapf).^2, i_2v.^2./(x0+Adgap-Gapf).^2],Ts);

% Define state-space model with user-defined free parameters
% Initialize model :

A  = [0 1;-4600 0];
B  = [0 0;-6e-6 6e-6];
```

```
C  = [1 0];
D  = [0 0];
K  = [0;0];

x0 = [Gapf(1); (Gapf(2)-Gapf(1))/Ts];

mod = idss(A,B,C,D,K,x0,'Ts',0);

As  = [0 1;NaN NaN];
Bs  = [0 0;NaN NaN];
Cs  = [1 0];
Ds  = [0 0];
Ks  = [NaN;NaN];
%Ks  = [0;0];

x0s = [NaN; NaN];

setstruc(mod,As,Bs,Cs,Ds,Ks,x0s);

set(mod,'Ts',0);

modfit = pem(fdata,mod);

Vsim = sim(modfit,iddata([],vdata.u,Ts),[Gapv(1);(Gapv(2)-Gapv(1))/Ts]);
Fsim = sim(modfit,iddata([],fdata.u,Ts),[Gapf(1);(Gapf(2)-Gapf(1))/Ts]);

modfit
```

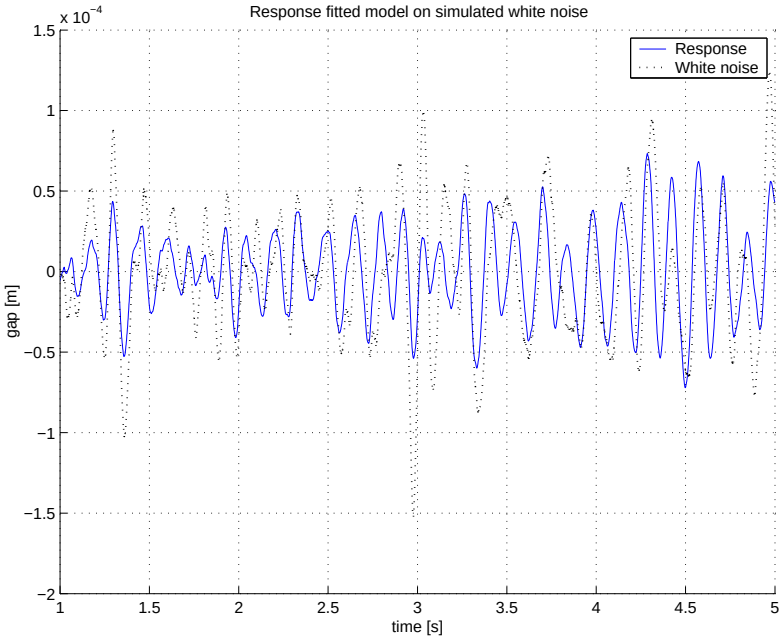


Figure D.1: Response fitted parameters on white noise.

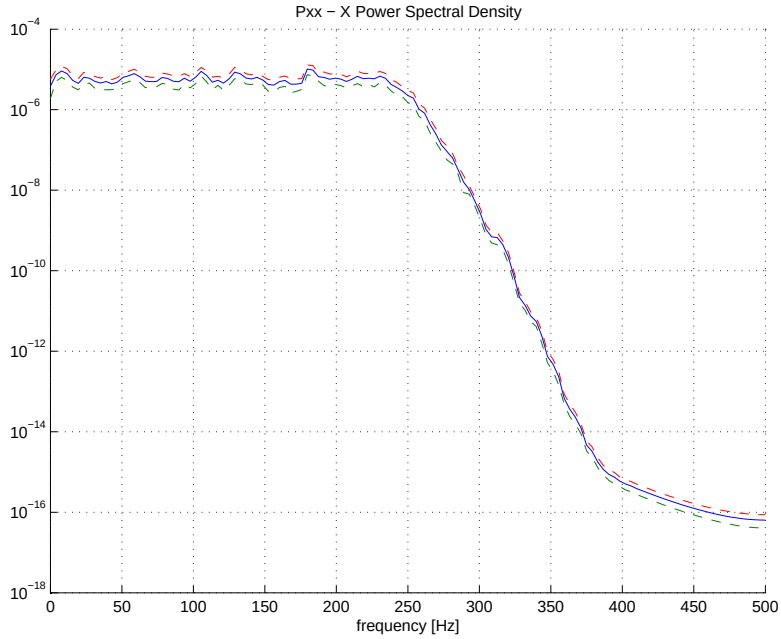


Figure D.2: Noise spectrum i_1.

D.4 M-file used for computing the LQR

```
%=============================================================================
% Calculation of gain matrix and observer
% 17 April 2005
%=============================================================================

m = 0.8;
k = 3652;
ksi =0.05;
a = [0 1; -k/m -ksi/m];
b = [0; 1/m];
c = [1 0];
d = [0];
b2 = [b zeros(2,1)];
c2 = [c; c];
d2 = [d -1];
d2 = [0 0; d2];
% now the inputs are v and ref, and outputs are x and error (x-r)
[ai,bi,ci,di] = zp2ss([],[0],[1]);     % Integrator zpk
[aa,ab,ac,ad] = append(a,b2,c2,d2,ai,bi,ci,di);     % [SYS1 0;0 SYS2]
sys1 = ss(aa,ab,ac,ad);
sys2 = ss(0,0,0,1);     % integrator will be connected as positive feedback
clsys = feedback(sys1,sys2,[3],[2],+1);     %the 3rd state is integral of error
[aa,ab,ac,ad] = ssdata(clsys);
[aa,ab,ac,ad] = ssdelete(aa,ab,ac,ad,[3],[2],[4]);
ac = eye(3);
ad = zeros(3,2);
[xa,xb,xc,xd] = ssdelete(aa,ab,ac,ad,[2],[]);
plant = ss(xa,xb,xc,xd);
% aa,ab,ac,ad has now 2 inputs [v; r], 3 states and three outputs [x; xdot; int(err)]
% xa,xb,xc,xd has 1 input [v] and three outputs [x; xdot; int(err)]
T= 0.001;
[dax,dbx,dcx,ddx] = ssdata(c2d(plant,T,'zoh'));
[da,db,dc,dd] = ssdata(c2d(ss(a,b,c,d),T,'zoh'));

q = diag([0 1e1 1e9]);     % controller penalty for each state
r  = diag([1e-1]);     % penalty for control action
k  = dlqry(dax,dbx,dcx,ddx,q,r);     % feedback gain matrix for control law:  u = -k*x

qe  = diag([1e-1 1e-1]);     % covariance of the process noise
re  = 1e-1;     % covariance of the sensor noise
L   = dlqe(da,diag([1e-3 1e-3]),dc,qe,re);     % observer matrix DLQE(A,G,C,Q,R)
[ao,bo,co,do] = destim(da,db,dc,dd,L,[1],[1]);

% % for future implementation on C: consider:
L = [L; 0];
```

```
[ua,ub,uc,ud] = dreg(dax,dbx,dcx,ddx,k,L,[1],[2],[1]);

% %Simulation
% q1 = ss(ua,ub,-uc,ud);
% q2 = ss(dax,dbx,dcx,ddx);
% clsys = feedback(series(q1,q2,[1],[1]),sys2,[1],[3],-1);
% clsys = minreal(clsys);
% u = 40e-6*[zeros(200,1) ones(200,1)]';
% t  = 0.01:0.01:2;
% [ta,tb,tc,td]=ssdata(clsys);
% [y,x] = dlsim(ta,tb,tc,td,u);
% plot(y);
% grid;
```

D.5 M-file used for kinematics maglev hexapod

```
% Kinematic modeling for maglev hexapod

clear all
close all
clc

% declaring the 6 DOFs
syms x y z rx ry rz

% declaring other parameters:
% r=radius of actuatorcenters, phi1=angle between two targets (top view),
% phi2=angle between two actuators (top view),
% dz1=z distance between target and tray surface
% dz1=z distance between antuator and tray surface
% Default values:
% r=5; phi1=(20/180)*pi; phi2=(24/180)*pi; phi3=(20/180)*pi; dz1=-.77; dz2=-.8;
syms r phi1 phi2 phi3 dz1 dz2

% Rotation tensor A10 and its approximation (sin(a)=a; cos(a)=1)
A10=[   cos(ry)*cos(rz),
        cos(ry)*sin(rz),
        -sin(ry);
        -cos(rx)*sin(rz)+sin(rx)*sin(ry)*cos(rz),
        cos(rx)*cos(rz)+sin(rx)*sin(ry)*sin(rz),
        sin(rx)*cos(ry);
        sin(rx)*sin(rz)+cos(rx)*sin(ry)*cos(rz),
        -sin(rx)*cos(rz)+cos(rx)*sin(ry)*sin(rz),
        cos(rx)*cos(ry)];

A10_approx=[     1,         rz,        -ry
```

```
               -rz+rx*ry,  1+rx*ry*rz, rx
                rx*rz+ry, -rx+ry*rz,   1];

% Body-fixed vectors to the targets (e1)
a1=[    -r*sin(phi1/2),      r*cos(phi1/2),      dz1];
a2=[     r*sin(phi1/2),      r*cos(phi1/2),      dz1];
a3=[     r*cos(pi/6-phi1/2),-r*sin(pi/6-phi1/2), dz1];
a4=[     r*cos(pi/6+phi1/2),-r*sin(pi/6+phi1/2), dz1];
a5=[    -r*cos(pi/6+phi1/2),-r*sin(pi/6+phi1/2), dz1];
a6=[    -r*cos(pi/6-phi1/2),-r*sin(pi/6-phi1/2), dz1];

% Fixed world vectors to the actuators (e0)
b1=[    -r*sin(phi2/2),      r*cos(phi2/2),      dz2];
b2=[     r*sin(phi2/2),      r*cos(phi2/2),      dz2];
b3=[     r*cos(pi/6-phi2/2),-r*sin(pi/6-phi2/2), dz2];
b4=[     r*cos(pi/6+phi2/2),-r*sin(pi/6+phi2/2), dz2];
b5=[    -r*cos(pi/6+phi2/2),-r*sin(pi/6+phi2/2), dz2];
b6=[    -r*cos(pi/6-phi2/2),-r*sin(pi/6-phi2/2), dz2];

% Plot option to check vectors
% plot(a1(1),a1(2),'*',a2(1),a2(2),'*',a3(1),a3(2),'*',a4(1),a4(2),'*',
    a5(1),a5(2),'*',a6(1),a6(2),'*')
% hold on; grid on
% plot(b1(1),b1(2),'*',b2(1),b2(2),'*',b3(1),b3(2),'*',b4(1),b4(2),'*',
    b5(1),b5(2),'*',b6(1),b6(2),'*')

% Computing all Gap-vectors
%
%      *<----------------- ai---------------------* e1
%      ^                                          ^
%      gi                                          \ Rcm
%      |                                            \
%      *<------------------ bi----------------------* e0
%
Rcm=[x,y,z];
g1=Rcm+a1*A10_approx-b1;
g2=Rcm+a2*A10_approx-b2;
g3=Rcm+a3*A10_approx-b3;
g4=Rcm+a4*A10_approx-b4;
g5=Rcm+a5*A10_approx-b5;
g6=Rcm+a6*A10_approx-b6;

% Normal vectors for each target (e0)
v1=[         1,              0,     1/tan(phi3)]*A10_approx;v1=v1/(sqrt(sum(v1.^2)));
v2=[        -1,              0,     1/tan(phi3)]*A10_approx;v2=v2/(sqrt(sum(v2.^2)));
v3=[-sin(pi/6),     -cos(pi/6),     1/tan(phi3)]*A10_approx;v3=v3/(sqrt(sum(v3.^2)));
v4=[ sin(pi/6),      cos(pi/6),     1/tan(phi3)]*A10_approx;v4=v4/(sqrt(sum(v4.^2)));
```

```
v5=[-sin(pi/6),      cos(pi/6),     1/tan(phi3)]*A10_approx;v5=v5/(sqrt(sum(v5.^2)));
v6=[ sin(pi/6),     -cos(pi/6),     1/tan(phi3)]*A10_approx;v6=v6/(sqrt(sum(v6.^2)));

% Calculating distance between actuator-center and plane of magnetic target
gap1=dot(g1,v1);
gap2=dot(g2,v2);
gap3=dot(g3,v3);
gap4=dot(g4,v4);
gap5=dot(g5,v5);
gap6=dot(g6,v6);

% Checking the Gap-lengths at zero
% subs(gap1,[x,y,z,rx,ry,rz,r,phi1,phi2,phi3,dz1,dz2],[0,0,0,0,0,0,5,(20/180)*
    pi,(24/180)*pi,(20/180)*pi,-1,-1.1])
% subs(gap2,[x,y,z,rx,ry,rz,r,phi1,phi2,phi3,dz1,dz2],[0,0,0,0,0,0,5,(20/180)*
    pi,(24/180)*pi,(20/180)*pi,-1,-1.1])
% subs(gap3,[x,y,z,rx,ry,rz,r,phi1,phi2,phi3,dz1,dz2],[0,0,0,0,0,0,5,(20/180)*
    pi,(24/180)*pi,(20/180)*pi,-1,-1.1])
% subs(gap4,[x,y,z,rx,ry,rz,r,phi1,phi2,phi3,dz1,dz2],[0,0,0,0,0,0,5,(20/180)*
    pi,(24/180)*pi,(20/180)*pi,-1,-1.1])
% subs(gap5,[x,y,z,rx,ry,rz,r,phi1,phi2,phi3,dz1,dz2],[0,0,0,0,0,0,5,(20/180)*
    pi,(24/180)*pi,(20/180)*pi,-1,-1.1])
% subs(gap6,[x,y,z,rx,ry,rz,r,phi1,phi2,phi3,dz1,dz2],[0,0,0,0,0,0,5,(20/180)*
    pi,(24/180)*pi,(20/180)*pi,-1,-1.1])

% Translational equations of Motion
syms F1 F2 F3 F4 F5 F6

f1=[         1,              0,     1/tan(phi3)];f1=f1/(sqrt(sum(f1.^2)));
f2=[        -1,              0,     1/tan(phi3)];f2=f2/(sqrt(sum(f2.^2)));
f3=[-sin(pi/6),     -cos(pi/6),     1/tan(phi3)];f3=f3/(sqrt(sum(f3.^2)));
f4=[ sin(pi/6),      cos(pi/6),     1/tan(phi3)];f4=f4/(sqrt(sum(f4.^2)));
f5=[-sin(pi/6),      cos(pi/6),     1/tan(phi3)];f5=f5/(sqrt(sum(f5.^2)));
f6=[ sin(pi/6),     -cos(pi/6),     1/tan(phi3)];f6=f6/(sqrt(sum(f6.^2)));

Fx=f1(1)*F1+f2(1)*F2+f3(1)*F3+f4(1)*F4+f5(1)*F5+f6(1)*F6;
Fy=f1(2)*F1+f2(2)*F2+f3(2)*F3+f4(2)*F4+f5(2)*F5+f6(2)*F6;
Fz=f1(3)*F1+f2(3)*F2+f3(3)*F3+f4(3)*F4+f5(3)*F5+f6(3)*F6;

% Rotational equations of Motion
syms drx dry drz ddrx ddry ddrz J1 J2 J3

M_sum=cross(a1,F1*f1) + cross(a2,F2*f2) + cross(a3,F3*f3) + cross(a4,F4*f4)
    + cross(a5,F5*f5) + cross(a6,F6*f6);
% Mx=M_sum(1);
% My=M_sum(2);
% Mz=M_sum(3);
```

```
% wx=drx-drz*sin(ry);
% wy=dry*cos(rx)+drz*sin(rx)*cos(ry);
% wz=-dry*sin(rx)+drz*cos(rx)*cos(ry);
%
% dwx= ddrx-ddrz*sin(ry)-drz*dry*cos(ry);
% dwy= ddry*cos(rx)-drx*dry*sin(rx)+ddrz*sin(rx)*cos(ry)
    -dry*drz*sin(rx)*sin(ry)+drx*drz*cos(rx)*cos(ry);
% dwz=-ddry*sin(rx)-drx*dry*cos(rx)+ddrz*cos(rx)*cos(ry)
    -drx*drz*sin(rx)*cos(ry)-dry*drz*cos(rx)*sin(ry);
%
% MMx=J1*dwx-(J2-J3)*wy*wz;
% MMy=J2*dwy-(J3-J1)*wz*wx;
% MMz=J3*dwz-(J1-J2)*wx*wy;

syms Mx My Mz
DDrx=  (Mx+(J2-J3) * (dry*cos(rx)+drz*sin(rx)*cos(ry))  *
    (dry*cos(rx)+drz*sin(rx)*cos(ry)) )/J1 + ddrz*sin(ry)+dry*drz*cos(ry);
DDry= ((My+(J3-J1) * (drx-drz*sin(ry))                  *
    (-dry*sin(rx)+drz*cos(rx)*cos(ry)) )/J2 - ddrz*sin(rx)*
    cos(ry)+dry*drz*sin(rx)*sin(ry)-drx*drz*cos(rx)*
    cos(ry)+drx*dry*sin(rx)) / cos(rx);
DDrz= ((Mz+(J1-J2) * (drx-drz*sin(ry))                  *
    (dry*cos(rx)+drz*sin(rx)*cos(ry)) )/J3 +
    ddry*sin(rx)+drx*dry*cos(rx)+drx*drz*sin(rx)*cos(ry)+
    dry*drz*cos(rx)*sin(rx)        ) / cos(rx)*cos(ry);

% Ready for copy-paste into simulink
% 6DOF-6GAP

% Translational equations of motion

% Computing Mx,My,Mz

% Rotational equations of motion (ddrx=, ddry=, ddrz=)
```

Appendix E

C-code

E.1 Sliding Mode Controller

```
m_model ============================================================
//      E.Visser, Feb 2004. Modified 06/23/2004
//      1DOF model of arm
//
//      Modified by Niels de Kleijn 02/15/2005
//      Horizontal SMC_NA 1DOF controller
//=============================================================================

#define S_FUNCTION_NAME Controller_SMC_NA
#define S_FUNCTION_LEVEL 2

#include "simstruc.h"
#include "math.h"


#define     CONT_ST         0               //  No. of continous states:
#define     DISC_ST         0               //  No. of discrete states
#define     GLOBALINTS      0               //  No. of global integers
#define     GLOBALREALS     0               //  No. of global reals


#define     NIP             16               //  No. of input ports

#define     IPW0            1                //  Input port 0 width     x
#define     IPW1            1                //  Input port 1 width     xdot
#define     IPW2            1                //  Input port 2 width     xddot
#define     IPW3            1                //  Input port 3 width     r
#define     IPW4            1                //  Input port 4 width     rdot
#define     IPW5            1                //  Input port 5 width     rddot
#define     IPW6            1                //  Input port 6 width     err
#define     IPW7            1                //  Input port 7 width     edot
```

```
#define     IPW8            1                   //  Input port 8 width     eddot
#define     IPW9            1                   //  Input port 9 width     eint
#define     IPW10           1                   //  Input port 10 width    lambda1
#define     IPW11           1                   //  Input port 10 width    lambda2
#define     IPW12           1                   //  Input port 11 width    phi
#define     IPW13           1                   //  Input port 12 width    minK1
#define     IPW14           1                   //  Input port 14 width    minK2
#define     IPW15           1                   //  Input port 15 width    gk

#define     NOP             3                  //  No. of output ports
                                                //  0       i_1
                                                //  1       i_2
                                                //  2       control signal

#define     OPW0            1                  //  Output port 0 width
                                                //  0       i_1

#define     OPW1            1                  //  Output port 1 width
                                                //  1       i_2

#define     OPW2            1                  //  Output port 2 width
                                                //  2       debug

#define     U0(element)     (*uPtrs0[element]) //  Pointer to Input Port0
#define     U1(element)     (*uPtrs1[element]) //  Pointer to Input Port1
#define     U2(element)     (*uPtrs2[element]) //  Pointer to Input Port2
#define     U3(element)     (*uPtrs3[element]) //  Pointer to Input Port3
#define     U4(element)     (*uPtrs4[element]) //  Pointer to Input Port4
#define     U5(element)     (*uPtrs5[element]) //  Pointer to Input Port5
#define     U6(element)     (*uPtrs6[element]) //  Pointer to Input Port6
#define     U7(element)     (*uPtrs7[element]) //  Pointer to Input Port7
#define     U8(element)     (*uPtrs8[element]) //  Pointer to Input Port8
#define     U9(element)     (*uPtrs9[element]) //  Pointer to Input Port9
#define     U10(element)    (*uPtrs10[element])//  Pointer to Input Port10
#define     U11(element)    (*uPtrs11[element])//  Pointer to Input Port11
#define     U12(element)    (*uPtrs12[element])//  Pointer to Input Port12
#define     U13(element)    (*uPtrs13[element])//  Pointer to Input Port13
#define     U14(element)    (*uPtrs14[element])//  Pointer to Input Port14
#define     U15(element)    (*uPtrs15[element])//  Pointer to Input Port15

//===========================================================================
// System parameters
//===========================================================================

#define mass            0.8                         // [kg]
#define ksi             0.05                        // [-]
#define Kf              3652                        // [N/m]
```

```
#define x01             497e-6                          // [m]
#define x02             523e-6                          // [m]

#define xnom            0                               // [m]

#define pi              3.141592654

// ============================================================================
// E actuator
//=============================================================================
#define adGap           67e-006     // Resulting from manual readout model fit

#define K1              6.36e-006   // Resulting from manual fit or PEM
#define K2              6.36e-006   // Resulting from manual fit or PEM
// ============================================================================
// Round actuator
//=============================================================================
//#define adGap         xxe-006     // Resulting from manual readout model fit

//#define K1            xxxe-006   // Resulting from manual fit or PEM
//#define K2            xxxe-006   // Resulting from manual fit or PEM

//=============================================================================
// Initialize
//=============================================================================

static void mdlInitializeSizes(SimStruct *S)
{
    ssSetNumSFcnParams(S, 0);
    if (ssGetNumSFcnParams(S) != ssGetSFcnParamsCount(S)) {
        return;
    }

    ssSetNumContStates(S, CONT_ST);
    ssSetNumDiscStates(S, DISC_ST);

    if (!ssSetNumInputPorts(S, NIP)) return;
     ssSetInputPortWidth(S, 0, IPW0);
    ssSetInputPortDirectFeedThrough(S, 0, 1);
     ssSetInputPortWidth(S, 1, IPW0);
    ssSetInputPortDirectFeedThrough(S, 1, 1);
     ssSetInputPortWidth(S, 2, IPW0);
    ssSetInputPortDirectFeedThrough(S, 2, 1);
     ssSetInputPortWidth(S, 3, IPW0);
    ssSetInputPortDirectFeedThrough(S, 3, 1);
     ssSetInputPortWidth(S, 4, IPW0);
```

```
    ssSetInputPortDirectFeedThrough(S, 4, 1);
     ssSetInputPortWidth(S, 5, IPW0);
    ssSetInputPortDirectFeedThrough(S, 5, 1);
     ssSetInputPortWidth(S, 6, IPW0);
    ssSetInputPortDirectFeedThrough(S, 6, 1);
     ssSetInputPortWidth(S, 7, IPW0);
    ssSetInputPortDirectFeedThrough(S, 7, 1);
     ssSetInputPortWidth(S, 8, IPW0);
    ssSetInputPortDirectFeedThrough(S, 8, 1);
     ssSetInputPortWidth(S, 9, IPW0);
    ssSetInputPortDirectFeedThrough(S, 9, 1);
     ssSetInputPortWidth(S, 10, IPW0);
    ssSetInputPortDirectFeedThrough(S, 10, 1);
     ssSetInputPortWidth(S, 11, IPW0);
    ssSetInputPortDirectFeedThrough(S, 11, 1);
     ssSetInputPortWidth(S, 12, IPW0);
    ssSetInputPortDirectFeedThrough(S, 12, 1);
     ssSetInputPortWidth(S, 13, IPW0);
    ssSetInputPortDirectFeedThrough(S, 13, 1);
      ssSetInputPortWidth(S, 14, IPW0);
    ssSetInputPortDirectFeedThrough(S, 14, 1);
     ssSetInputPortWidth(S, 15, IPW0);
    ssSetInputPortDirectFeedThrough(S, 15, 1);

    if (!ssSetNumOutputPorts(S, NOP)) return;
    ssSetOutputPortWidth(S, 0, OPW0);
    ssSetOutputPortWidth(S, 1, OPW1);
    ssSetOutputPortWidth(S, 2, OPW2);

    ssSetNumSampleTimes(S, 1);
    ssSetNumRWork(S, GLOBALREALS);
    ssSetNumIWork(S, GLOBALINTS);
    ssSetNumPWork(S, 0);
    ssSetNumModes(S, 0);
    ssSetNumNonsampledZCs(S, 0);

    // Take care when specifying exception free code - see sfuntmpl_doc.c
    ssSetOptions(S, SS_OPTION_EXCEPTION_FREE_CODE);
}

static void mdlInitializeSampleTimes(SimStruct *S)
{
    ssSetSampleTime(S, 0, CONTINUOUS_SAMPLE_TIME);
    ssSetOffsetTime(S, 0, 0.0);
}
```

```
//=============================================================================
// Control computation and outputs (i_1 & i_2)
//=============================================================================

static void mdlOutputs(SimStruct *S, int_T tid)
{
    InputRealPtrsType   uPtrs0  = ssGetInputPortRealSignalPtrs(S,0);
    InputRealPtrsType   uPtrs1  = ssGetInputPortRealSignalPtrs(S,1);
    InputRealPtrsType   uPtrs2  = ssGetInputPortRealSignalPtrs(S,2);
    InputRealPtrsType   uPtrs3  = ssGetInputPortRealSignalPtrs(S,3);
    InputRealPtrsType   uPtrs4  = ssGetInputPortRealSignalPtrs(S,4);
    InputRealPtrsType   uPtrs5  = ssGetInputPortRealSignalPtrs(S,5);
    InputRealPtrsType   uPtrs6  = ssGetInputPortRealSignalPtrs(S,6);
    InputRealPtrsType   uPtrs7  = ssGetInputPortRealSignalPtrs(S,7);
    InputRealPtrsType   uPtrs8  = ssGetInputPortRealSignalPtrs(S,8);
    InputRealPtrsType   uPtrs9  = ssGetInputPortRealSignalPtrs(S,9);
    InputRealPtrsType   uPtrs10 = ssGetInputPortRealSignalPtrs(S,10);
    InputRealPtrsType   uPtrs11 = ssGetInputPortRealSignalPtrs(S,11);
    InputRealPtrsType   uPtrs12 = ssGetInputPortRealSignalPtrs(S,12);
    InputRealPtrsType   uPtrs13 = ssGetInputPortRealSignalPtrs(S,13);
    InputRealPtrsType   uPtrs14 = ssGetInputPortRealSignalPtrs(S,14);
    InputRealPtrsType   uPtrs15 = ssGetInputPortRealSignalPtrs(S,15);
    real_T              *i_1    = ssGetOutputPortRealSignal(S,0);
    real_T              *i_2    = ssGetOutputPortRealSignal(S,1);
    real_T              *DB     = ssGetOutputPortRealSignal(S,2);
    real_T               x,xdot,xddot,r,rdot,rddot,err,edot,eddot,eint,n,rho,
                         Ss,dSdt,g1,g2,minK1,minK2,lambda1,lambda2,phi,gk;
    real_T               i1sw,i1eq,i2sw,i2eq;


// define inputs
x       = U0(0);
xdot    = U1(0);
xddot   = U2(0);
r       = U3(0);
rdot    = U4(0);
rddot   = U5(0);
err     = U6(0);
edot    = U7(0);
eddot   = U8(0);
eint    = U9(0);
lambda1 = U10(0);
lambda2 = U11(0);
phi     = U12(0);
minK1   = U13(0);
minK2   = U14(0);
gk      = U15(0);
```

```
// calculate Gaps
g1 = x01+adGap+x;
g2 = x02+adGap-x;

// Sliding surface
Ss      = edot + lambda1*err + lambda2*eint;

DB[0] = Ss;

dSdt    = -rddot - lambda1*rdot + lambda2*err;
rho     = dSdt + lambda1*xdot - Kf*x;
n       = fabs(rho);

if  (Ss >= 0.0)
    {
    i1sw    = gk*(Ss/phi)*sqrt((n*(g1*g1))/K1);
    i2sw    = 0;
    }
if  (Ss < 0.0)
    {
    i1sw    = 0;
    i2sw    = -gk*(Ss/phi)*sqrt((n*(g2*g2))/K2);
    }

if  (rho >= 0.0)
    {
    i1eq    = g1*sqrt(rho/K1);
    i2eq    = 0;
    }
if  (rho < 0.0)
    {
    i1eq    = 0;
    i2eq    = g2*sqrt(-rho/K2);
    }

i_1[0] = i1sw+i1eq;
i_2[0] = i2sw+i2eq;

}
//============================================================================
// Terminate
//============================================================================

static void mdlTerminate(SimStruct *S)
{
  UNUSED_ARG(S);            /* unused input argument */
}
```

```
#ifdef  MATLAB_MEX_FILE    /* Is this file being compiled as a MEX-file? */
#include "simulink.c"      /* MEX-file interface mechanism */
#else
#include "cg_sfun.h"       /* Code generation registration function */
#endif
```

E.2 Model of the beam

```
//===== Arm_model ========================================================================
//      E.Visser, Feb 2004. Modified 06/23/2004
//      1DOF model of arm
//
//      Modified by Niels de Kleijn 02/15/2005
//      Horizontal SMC_NA 1DOF model with flexures and 2 actuators
//========================================================================================

#define S_FUNCTION_NAME Arm_model_SMC_NA
#define S_FUNCTION_LEVEL 2

#include "simstruc.h"
#include "math.h"


#define     CONT_ST         2               //  No. of continous states:
                                            //  0       Gap
                                            //  1       Gap_dot

#define     DISC_ST         0               //  No. of discrete states
#define     GLOBALINTS      0               //  No. of global integers
#define     GLOBALREALS     1               //  No. of global reals
                                            //  0       Gap 1 [m]

#define     NIP             2               //  No. of input ports
                                            //  0       v

#define     IPW0            1               //  Input port 0 width
                                            //  0       i_1

#define     IPW1            1               //  Input port 0 width
                                            //  0       i_2

#define     NOP             1               //  No. of output ports
                                            //  0       Gap

#define     OPW0            2               //  Output port 0 width
```

```
                                                //  0       Gap
                                                //  1       dGap

#define     U1(element)      (*uPtrs0[element]) //  Pointer to Input Port0 (i_1)
#define     U2(element)      (*uPtrs1[element]) //  Pointer to Input Port1 (i_2)

//=============================================================================
// System parameters
//=============================================================================

#define mass            0.8                         // [kg]
#define ksi             0.05                        // [-]
#define Kf              3652                        // [N/m]

#define x01             497e-6                      // [m]
#define x02             523e-6                      // [m]
#define xnom            0                           // [m]
#define adGap           67e-006                     //

#define K1              6.36e-006                   //
#define K2              6.36e-006                   //

//=============================================================================
// Initialize
//=============================================================================

static void mdlInitializeSizes(SimStruct *S)
{
    ssSetNumSFcnParams(S, 0);
    if (ssGetNumSFcnParams(S) != ssGetSFcnParamsCount(S)) {
        return;
    }

    ssSetNumContStates(S, CONT_ST);
    ssSetNumDiscStates(S, DISC_ST);

    if (!ssSetNumInputPorts(S, NIP)) return;
    ssSetInputPortWidth(S, 0, IPW0);
    ssSetInputPortDirectFeedThrough(S, 0, 1);
    ssSetInputPortWidth(S, 1, IPW1);
    ssSetInputPortDirectFeedThrough(S, 1, 1);

    if (!ssSetNumOutputPorts(S, NOP)) return;
    ssSetOutputPortWidth(S, 0, OPW0);

    ssSetNumSampleTimes(S, 1);
    ssSetNumRWork(S, GLOBALREALS);
```

```
    ssSetNumIWork(S, GLOBALINTS);
    ssSetNumPWork(S, 0);
    ssSetNumModes(S, 0);
    ssSetNumNonsampledZCs(S, 0);

    // Take care when specifying exception free code - see sfuntmpl_doc.c
    ssSetOptions(S, SS_OPTION_EXCEPTION_FREE_CODE);
}

static void mdlInitializeSampleTimes(SimStruct *S)
{
    ssSetSampleTime(S, 0, CONTINUOUS_SAMPLE_TIME);
    ssSetOffsetTime(S, 0, 0.0);
}


//=============================================================================
// Initialize continuous states & global real variables
//=============================================================================

#define MDL_INITIALIZE_CONDITIONS

static void mdlInitializeConditions(SimStruct *S)
{
    real_T *x       = ssGetContStates(S);

    x[0] =  xnom;                       // Initial gap when beam is in middle position
    x[1] =  0.0;                        // Initial gap_dot

}

//=============================================================================
// Define outputs
//=============================================================================

static void mdlOutputs(SimStruct *S, int_T tid)
{

    real_T            *Gap      = ssGetOutputPortRealSignal(S,0);
    real_T            *x        = ssGetContStates(S);
    real_T            *dx       = ssGetdX(S);
```

```
    // Endblocks

    if (x[0] > x02)
        {
        x[0] = x02;
        x[1] = 0;
        }
    if (x[0] < -x01)
        {
        x[0] = -x01;
        x[1] = 0;
        }


    Gap[0]      = x[0];
    Gap[1]      = x[1];

}

//=============================================================================
// Compute derivatives of continuous states
//=============================================================================

#define MDL_DERIVATIVES

static void mdlDerivatives(SimStruct *S)
{
    InputRealPtrsType   uPtrs0  = ssGetInputPortRealSignalPtrs(S,0);
    InputRealPtrsType   uPtrs1  = ssGetInputPortRealSignalPtrs(S,1);
    real_T              *dx     = ssGetdX(S);
    real_T              *x      = ssGetContStates(S);
    real_T              F,Fflex,Fd,Ft,wn;

    // U1(0) = i_1
    // U2(0) = i_2


    F = -(K1*(U1(0)*U1(0))/((x01+adGap+x[0])*(x01+adGap+x[0])))
+(K2*(U2(0)*U2(0))/((x02+adGap-x[0])*(x02+adGap-x[0])));

    Fflex = Kf*x[0];

    wn = sqrt(Kf/mass);

    Fd = 2*ksi*wn*x[1];

    Ft = F-Fflex-Fd;
```

```
    dx[0]   = x[1];
    dx[1]   = Ft/mass;


}

//=============================================================================
// Terminate
//=============================================================================

static void mdlTerminate(SimStruct *S)
{
  UNUSED_ARG(S);            /* unused input argument */
}

#ifdef  MATLAB_MEX_FILE    /* Is this file being compiled as a MEX-file? */
#include "simulink.c"      /* MEX-file interface mechanism */
#else
#include "cg_sfun.h"       /* Code generation registration function */
#endif
```

E.3 Controller with permanent magnets

```
//=========================================================================
// Control computation and outputs (i_1 & i_2)
//=========================================================================

// define inputs
x       = U0(0);
xdot    = U1(0);
xddot   = U2(0);
r       = U3(0);
rdot    = U4(0);
rddot   = U5(0);
err     = U6(0);
edot    = U7(0);
eddot   = U8(0);
eint    = U9(0);
lambda1 = U10(0);
lambda2 = U11(0);
phi     = U12(0);
// Permanent magnet
//Kp        = U13(0);
//P         = U14(0);
```

```
// calculate Gaps
g1 = x01+adGap+x;

// calculate force permanent magnet
Fp = -0.0056e6*(x01+x)+7.8;

// Sliding surface
Ss      = edot + lambda1*err + lambda2*eint;

DB[0] = Ss;

dSdt    = -rddot - lambda1*rdot + lambda2*err;
rho     = dSdt + lambda1*xdot - Fp; // - Kf*x
n       = fabs(rho);

if  (Ss >= 0.0)
    {
    isw = (Ss/phi)*sqrt((n*(g1*g1))/K1);
    }
if  (Ss < 0.0)
    {
    isw = 0;
    }

if  (rho >= 0.0)
    {
    ieq = g1*sqrt(rho/K1);
    }
if  (rho < 0.0)
    {
    ieq = 0;
    }

i[0] = isw+ieq;

if  (i[0] >1.50)
    {
    i[0] = 1.50;
    }
}
```