KTH Industrial Engineering and Management

Feed Drive System Testbed for Monitoring and Control of Linear Axes Degradation Using Sensor-based Metrology

Pablo Albiol Graullera
Mattias Dahlqvist
Martin Favre
Jonathan Hytönen
Oden Lobell
Filip Stenbeck
Martin Zakardissnehf

Supervisor: Bengt Eriksson
Examiner: Jan Wikander

Mechatronics Advanced Course, Fall 2016
Royal Institute of Technology
Stockholm, Sweden

Abstract

Machine tool linear axes are widely used in the industry for positioning the workpiece or tool. During usage the parts in the mechanical structure will degrade and this will lead to inaccuracies in positioning. The current procedures to diagnose the magnitude of these inaccuracies and calibrate the systems are expensive and time consuming. This report presents the construction of a two-axis feed-drive system with the ability to artificially induce errors, thus simulating errors that occurs in real world machines. The test bench is intended to be used in further research on error detection of stacked errors through an inertial measurement unit. It has the capability to replicate a low frequency error of 20 μm. A control system was implemented which allows trajectory movements from a user-friendly GUI. The stability, robustness and accuracy of the mechanical system is not verified due to limited time. A tri-axial gyroscope and accelerometer are placed on the working surface and the ability to detect the presence of an induced error have been verified.

Acknowledgements

We would like to take this opportunity to thank Anton Kviberg and Jan Stamer for their help with manufacturing and assembling the testbed, we would also like to thank Payman Tehrani for his advice regarding the NI system, lastly we would like to thank Andreas Archenti, Károly Szipka and Bengt Eriksson for their invaluable help and advice throughout the project.

Team IMU

2016-12-20

Contents

Nomenclature

AC Alternating current

CAD Computer-Aided Design

CMM Coordinate Measurement Machine

CNC Computer Numerical Control

$CRio$ CompactRio

DC Direct current

DMA Direct Memory Access

DOF Degrees of Freedom

FEM Finite Element Method

$FIFO$ First In First Out

$FPGA$ Field-Programmable Gate Array

GUI Graphical User Interface

HIL Hardware-in-the-Loop

HMI Human Machine Interface

IMU Inertial Measurement Unit

ISO International Organization for Standardization

KTH Kungliga Tekniska Högskolan

NC Numeric Control

NI National Instruments

OS Operating System

PID Proportional–Integral–Derivative

RPM Revolutions Per Minute

RT Real-Time

$RTOS$ Real-Time Operating System

TUR Test Uncertainty Ratio

UI User Interface

VI Virtual Instrument

1 Introduction

1.1 Background and objectives

Machine tools that depend on multi-axial feed drive systems are often used with high loads and frequent usage during their functional lifespan. Material fatigue, abrasion or adhesion will eventually cause degradation and this will affect the precision of the tool. To counteract the degradation of the CNC-machines, visual inspection and preventive maintenance like calibration are done [Vogl et al., 2015].

In order to recalibrate feed drive systems today; ball bars, cross grid encoders,and laser interferometers are used to measure the geometric errors. This is often not cost effective since the machine must be taken off the production line during the calibration. If these errors could be measured and corrected autonomously it would lower production costs and prolong the life of the machine.

This report is done as part of the Mechatronics Advanced Course, a project course which begun in the spring of 2016 and finished in December 2016. The project is done in collaboration with Royal Institute of Technology (KTH) and National Institute of Standards and Technology (NIST) and is a continuation of a study by Vogl et al.. The aforementioned study investigates the error detection for one-axial feed drive system using an IMU as measurement instrument.

To be able to continue existing research a more controllable working environment with more than one-axis is needed. This report will cover the construction of a two-axis testbed with the ability to induce errors and the identification of these errors with a measurement instrument.

1.2 Pre-study conclusion

The preliminary research (see Appendix E for the full pre-study) in the field of error inducement and error detection through the use of a sensor-based method concludes that the project is reasonable and does not need any large alteration. All requirements stated are possible to fulfill.
Discussions with the Swedish supplier Acumo have been held, and the company will provide the guideways, ball screws and motor. By choosing one supplier for the test bench components, the quality can be assured, compatibility guaranteed and the costs minimized. The IMU components have been recommended by G. Vogl. A complete parts list for both the test rig and the IMU can be seen in Appendix D.

1.3 Project goal of the stakeholder

The final goal of the project is to construct a two-axis testbed which meets the requirements mentioned in 1.5 in order to closely emulate a real high-precision CNC-machine in the industry. The testbed should have the ability to induce artificially different types of repeatable geometric errors (low, middle and high frequency errors) with the purpose to emulate the deterioration in the guideways of real machines.

The testbed shall also have the possibility to place an IMU and other sensor instruments on the top plate to allow for the detection and identification of the previous induced errors.

1.4 Scope and limitations

The long term vision of the project is to develop a self-diagnostic and calibration system for machine tools. This methodology could be used later to evaluate the state of a machine before using it for production and to recalibrate the machine if needed.

This report is aimed towards the construction of a two-axis feed drive system, inducing artificial errors to the feed drive system and early measurements of them using an IMU. The actual correction and recalibration of the specific hardware will be excluded from this report.

1.5 Requirements

The following requirements are set requirements from the stakeholder in order to validate the functionality of the test bench and the IMU.

Test Bench and Electrical System Requirements

- The minimal stiffness of each axis will be more or equal to 400 $\frac{Nm}{\mu m}$ in the axis direction.
- The axial position accuracy shall be $\pm 5\mu m$.
- The test bench shall be able to induce controlled errors in position and straightness on the controlled axis of a magnitude of at least $20\mu m$.

IMU Requirements

- The IMU shall measure the position error with an accuracy of $\pm 5\mu m$.
- The IMU shall detect a straightness error of at least $20\mu m$ with a test uncertainty level more than 4:1.
- The IMU should not average more than 10 measurements to be within $\pm 5\mu m$ of the inferometer measurement.

Software Requirements

- The positioning accuracy shall be $\pm 5\mu m$.
- The software should have a user friendly GUI for advanced users. The GUI should have modifiable control parameters in the interface.
- The system should have trajectory controlled tool paths.

Overall System Requirements

- The system shall have a safety distance from where no human can be harmed from the workplace.

2 Method and Design Process

2.1 Project process

At the start of the fall semester the overall development process was changed from an agile development process to a concurrent engineering approach since the success of the project was highly dependent on the arrival of the hardware components. A concurrent engineering methodology allowed the subgroups in the team to work with the electronic hardware and software in parallel with the mechanical hardware development.

This has been proven to give good results for the overall project management. The last mechanical parts were manufactured and assembled the last week before the final presentation. However, as the electronic hardware and software group had been working in parallel, it was possible to integrate all the different components and have a well functional prototype in time.

During the software intensive development phase the team responsible for the electronics and software leaned towards using more agile methods suitable for software development. The focus has been on developing incremental functionality and testing each part independently to integrate it later into the main program. Thus the whole team has put effort into a specific component each week in order to allow for rapid development, continuous improvement, waste reduction and flexible response to change. Another important aspect has been working under requirements and working through the collaborative efforts of self-organizing cross-functional teams.

2.2 Design and Development Method

2.2.1 Mechanical Hardware

The majority of the mechanical hardware components were designed the project team and then manufactured with the help of KTH's production team. The various parts which were not designed by the team were acquired from different suppliers.

The mechanical design and development has been done with the support of different CAD-models and FEM-analysis. CAD-models have been used to design the manufactured parts, CAD-models of the acquired parts have also been employed. All of them have been combined to create a complete model of the testbed. This model has been very helpful to visualize the whole testbed, understand the connections between the different mechanical components and provide help during the assembly.

FEM-analysis have been realized to make sure that the designed parts meet the desired strength and deformation requirements. In addition they have also been useful to optimize different parts and reduce weight.

Furthermore, multiple discussions have been held with different suppliers and technicians at the production department of KTH leading to multiple CAD-model iterations and improvements.

2.2.2 Electrical Hardware and Software

The programming and the setup of the electrical system was done in parallel and started when the electrical hardware arrived from NI. The various software parts have been programmed incrementally to enable debugging and verification of each part separately.

LabVIEW video-tutorials have been used to build a solid knowledge of how to program the different subsystems and learn about NI best practices and architectural design patterns. Special effort has been made into structuring and organizing the code in a modular and reusable way.

An important advantage during the software development phase has been

the possibility of using simulated axis. This has been very convenient as it is very easy to switch from the real drives to the simulated axis, working like a Hardware-in-the-loop (HIL) simulation. This feature has allowed the group to develop and test the software without the need of having the mechanical parts assembled.

In addition, several discussions have been held with an expert from National Instruments to evaluate the quality of the different programs and to distribute and organize the code in a better way.

2.3 Verification and Validation

To ensure that the requirements of the testbed are fulfilled, verification tests need to be made. Early in the project it was said that the final verification test of the whole testbed would be to use an interferometer and compare it with sensor data from inertial measurement units. This would test all the elements of the setup, what errors could be introduced and measured as well as the requirements regarding mechanical accuracy of the testbed. The laser interferometer was not used due to limited time. However, verification tests were made during the project, these were smaller subsystem tests to verify the functionality of motor control, measuring of sensor data and inducement of low frequency errors.

The control system for the motors were validated through trajectory control test in both axis. Selected control parameters were chosen in software, like velocity, acceleration and jerk for a position controlled loop. This was used to detect delays and stability of the control system.

Sensor data was verified by comparing sensor data from standardized tests with and without induced low frequency errors.

2.4 Proposed Methodology

The proposed methodology to identify artificially induced geometric errors consist of performing experiments. This means using a measurement instrument (IMU) formed by a triaxial accelerometer and a gyroscope to measure

low, middle and high frequencies errors.

The proposed experiments are described as follows: A set of straight line moves are done without the error induced and the data from the sensors is averaged. After this, the error is induced and the same number of experiments are repeated. Finally the average results from both tests are compared to identify the induced errors.

Accelerometer data is used to identify straightness errors and gyroscope data to measure angular errors. A Matlab script is provided to analyze and display the results.

3 System Design

3.1 Mechanical Hardware

This section summarizes the hardware design and theory behind the error induction.

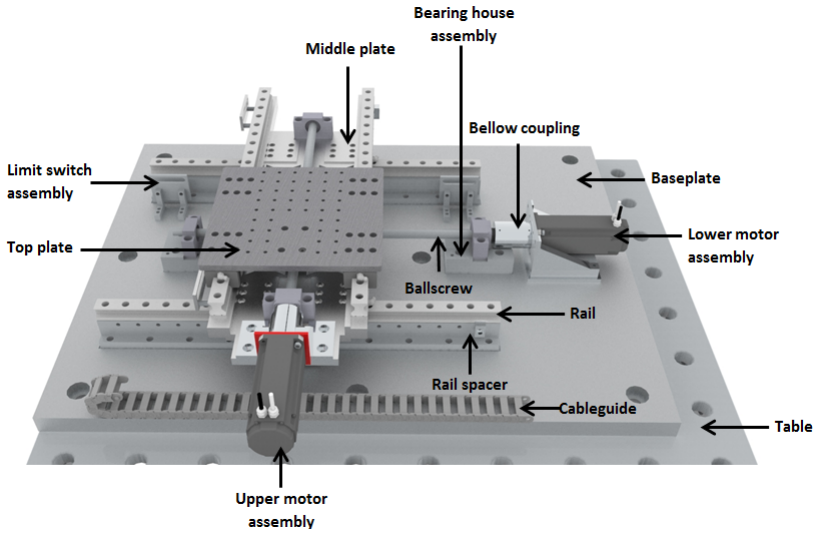


Figure 3.1: A summary of hardware components.

The mechanical components of the testbed, Figure 3.1, consist of both pre-manufactured and custom made components. The pre-manufactured components are used in applications on the testbed where due to tolerance and time restrictions they cannot be produced in-house. Some areas of the testbed required to be machined in-house due to the complexity of the parts. A summary of the components can be seen in Appendix A.

3.1.1 Theory

Error induction analysis

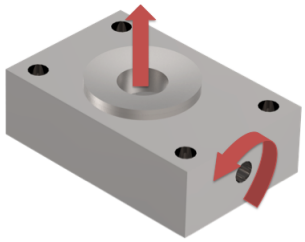


Figure 3.2: A force inducer.

In the pre-study in Appendix E it was stated that the simplest way to induce an error was to bend the rail using a displacer seen in Figure 3.2 donated by Modig Machine Tool AB. This device transforms rotational displacement into a linear one.

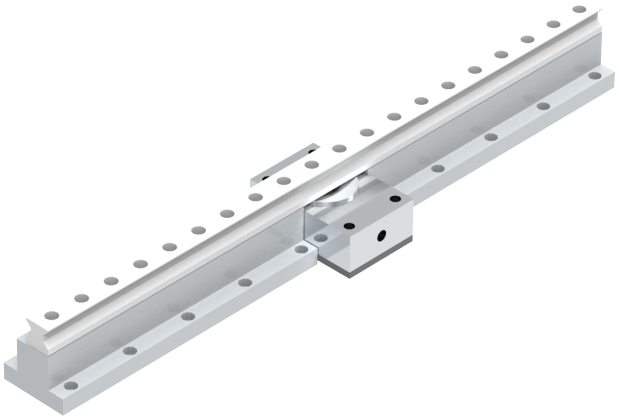


Figure 3.3: The choice of placement for the pusher.

This is placed under the middle of the rail as seen in Figure 3.3. The linear displacement can be measured with a dial indicator and accurately set. A finite element analysis can be seen in Appendix B.4 where a force is applied to deform the rail approximately 20 µm as per the requirements in Section 1.5. This study shows that the required force is approximately $8kN$ and as the bolt is M10 this force is very attainable. However the stress of approximately $700MPa$ implies that this error induction should be done with care to not to exceed the yield strength of the steel rail.

Accuracy

Because of the accuracy needed when manufacturing and assembling the components it is important to have a clear definition of it so that it can be measured correctly. Accuracy as defined by ISO 5725 is described by the terms trueness and precision. Trueness refers to the difference to the average value from a large series of tests compared to a reference value. The parallelism error between two axes of linear motion is defined as the angle between a reference straight line and another linear axis [ISO 5725:1994(E), 1994]. Several other relevant and useful definitions can be found in the ISO standard.

3.1.2 Pre-manufactured components

This section describes the pre-manufactured and purchased mechanical hardware used in the design.

Rails

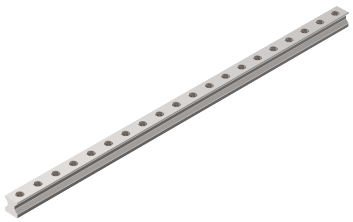


Figure 3.4: One of the four rails.

The rail seen in Figure 3.4 is manufactured by NSK with their model number R1A25. Each rail is $600mm$ long and made of carbon steel. The rail is fastened with M6 bolts. Four of these rails were procured, one set of two for the X-axis and another set of two for the Y-axis [NSK, 2009].

Sliders

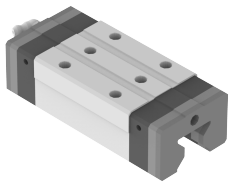


Figure 3.5: One of the eight slides.

The slider seen in Figure 3.5 is manufactured by NSK and has the model number RAA25ALK6Z. The accuracy grade of this component is NSK standard K6 which will provide a running parallelism of 9 µm. It goes along the rail and has six threaded M6 holes to connect items on top. A total of eight sliders were procured. Each axis of movement has a set of four sliders. The main material is carbon steel with plastic details. Attached to each slider is a lubrication unit from NSK [NSK, 2009].

Ball screw

Figure 3.6: One of the two ball screws.

The ball screw seen in Figure 3.6 is manufactured by NSK and has the model number PSS1505N1D0561. The accuracy grade is C5 which together with the stroke length of $300mm$ responds to a end positional error of 23 µm which a fluctuation of 18 µm [THK, 2016]. They are held in place by two bearing houses each and connected to a motor through a bellow coupling. They have an effective stroke of $300mm$ and have a lead of $5mm$. The material is steel. The ball screw arrived preloaded [NSK, 2006].

Couplings

Figure 3.7: One of the two couplings.

The coupling seen in Figure 3.7 is produced by the company JAKOB and and modified by Acumo to match the motor and ball screw. The coupling is a bellow coupling that uses friction to transfer momentum. The middle part

of the coupling that looks like a spring allows the ball screw and motor to be misaligned and still function properly [JAKOB, 2016].

Table

Figure 3.8: The table.

The testbed is mounted to the table seen in Figure 3.8 with 8 M20 bolts. The table has a weight of $430kg$, the weight makes it possible to use the table as a stable reference surface as it will not move considerably when affected by external forces in comparison to the rest of the system. The table is made by the company Siegmund.

Cableguides

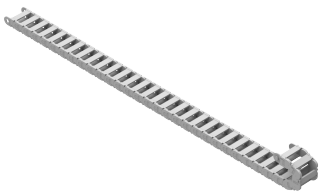


Figure 3.9: One of the cableguides.

To secure the movement of the different cables of the two moving levels two cableguides were procured, seen in Figure 3.9. The model is IGUS E2.10.30.018.0. The inner dimensions of 10x30mm are sufficient to fit all cables. The minimum bending radius of 18mm allows it to fit in the test bench. One meter is given for each level. [IGUS, 2009]

3.1.3 Manufactured parts

The following chapter will describe the process of the design of the custom made parts.

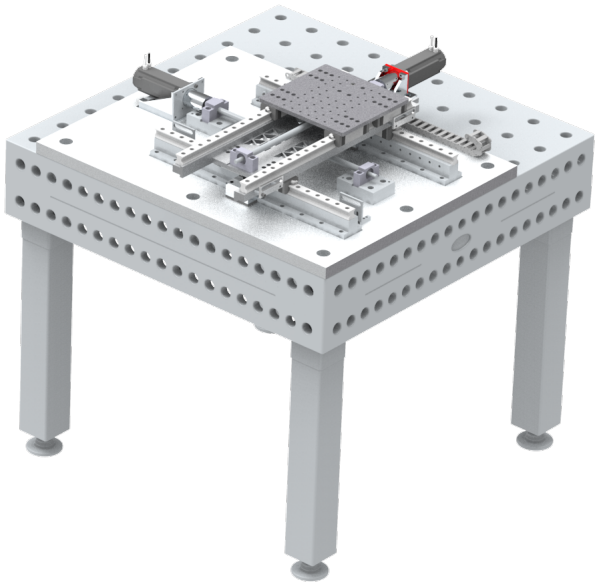


Figure 3.10: The full design.

All these components were initially iteratively designed in CAD and then manufactured at an in-house workshop. As mentioned in the requirements (1.5) the aim for the positioning accuracy of the test bench is 5 µm. There is also the requirement to minimize static errors and eliminate dynamic errors. This places two requirements on the test bench. Not to flex or move dynamically under forces to avoid unrepeatable errors and to allow for simple

alignment to avoid static errors. There is also a secondary requirement to minimize manufacturing time and cost.

To assist the design process, finite element analysis was used to determine the maximum deviation with a given force.

Base plate

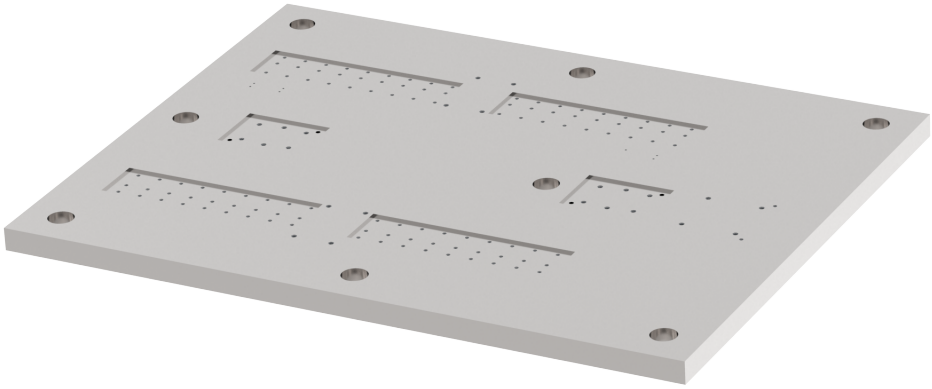


Figure 3.11: Baseplate.

The plate is a 70x90x30cm steel plate weighing approximately 150kg. The main purpose of the base plate is to create a stable reference upon which all other parts can be mounted. The surface of the plate had not been machined and did not have a known surface smoothness. A solution to this was to manufacture slots with clearance fit in which the different parts can be placed. This allows for all the parts to be placed on the same plane. To connect the plate to the table eight holes were added to allow for M20 bolts. For all other holes it was deemed to be too difficult to use through holes due

to the table underneath so all items on top of the base plate are connected through threaded holes.

Rail spacer

Figure 3.12: Rail spacer.

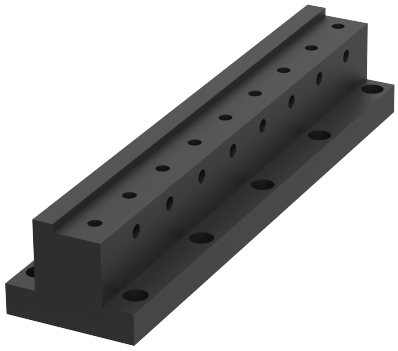


Figure 3.13: Rail spacer with the reference surface.

Four rail spacers are used of two different designs, Figure 3.12 and 3.13. Two of each are used. The rail spacer holds a twofold purpose in lifting the rails up so the ball screw can fit between the base plate and the middle plate. The rail spacer seen in Figure 3.13 has a reference wall which the rail will be pushed against using the clamps seen in Figure 3.18 to make it straight.

To be able to use the wall as a reference surface and to ensure that the spacers have the same height they were machined while mounted to the base plate.

It is fastened to the base plate with ten M6 bolts on the sides as well as nine M6 bolts in the middle which also is connected to the rail above. To simplify manufacturing the spacer does not have any protruding parts and can all be cut from one bar.

Bearing House spacers

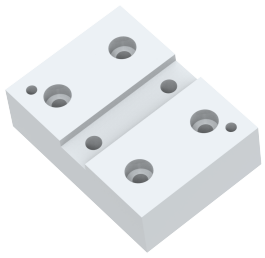


Figure 3.14: The spacer between the base plate and the bearing house.

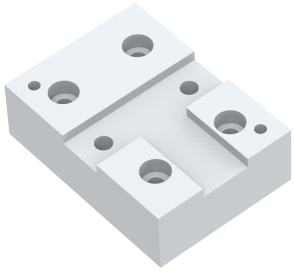


Figure 3.15: The spacer between the base plate and the bearing house on the motor side.

Two spacers, seen in Figure 3.14 and Figure 3.15, are manufactured to lift the bearing houses up to a more favourable position. A socket is cut out in the middle to get a flat surface. The bearing house is placed in the slot with a clearance fit. The spacer is fastened in the base plate with four M8 bolts. To minimize the movement of the spacer during dynamic loads a set of $6mm$ conical alignment pins are used. In the spacer on the motor side as seen in Figure 3.15 an additional socket has been cut out to give space for the motor coupling.

Motor holder

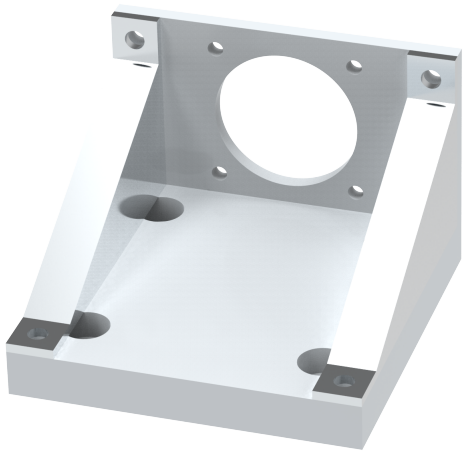


Figure 3.16: Lower motor assembly.

The construction seen in Figure 3.16 lifts the motor up so it is aligned with the ball screw, it also makes it possible to mount the motor. The assembly is fastened to the base plate with four M8 bolts. The frontal plate fastens the motor with four M5 bolts and is stabilized by the two supports that are fastened with two M6 bolts each.

Limit switch holder

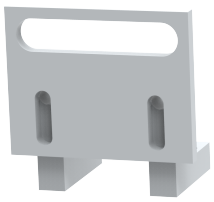


Figure 3.17: Limit Switch Holder

To limit the movements of the lower level in cases of malfunction two limit switches are placed along the rail. These switches have an M10 thread on them so two M10 nuts are used to lock their position. This makes it possible to move the switch in the xy-plane to adjust how close to the sliders the limit switch are and how far from the end of the rail the limit switch are positioned. To adjust the height of the limit switch two M6 bolts and nuts are used. The limit switch assembly is connected to the base plate with four M3 bolts.

Alignment clamp

Figure 3.18: Clamp to push the rail towards its reference.

Due to the accuracy requirements the rail needs to be aligned towards a reference surface to ensure straightness. To simplify the manual process of this setup the clamps in Figure 3.18 will push the rail onto the reference surface in the rail spacer. This setup will only be used on one side on each level, see Figure 3.13, to not create conflicts caused by having two separate references.

It is important for the rail to endure the force of its bolt without any major deformations. A finite element analysis which emulates the clamp with a M6 bolt tightened to $24kN$, an exaggerated scenario, was done to see if it could withstand the force. The report can be seen in Appendix B.3. Here the maximum displacement is approximately 60 µm. This was deemed acceptable as it does not protrude past the clamps edges.

Middle plate

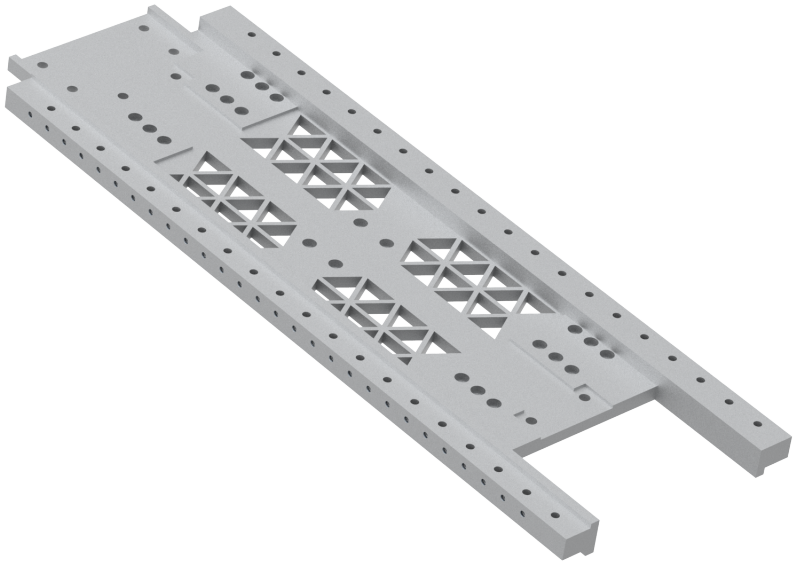


Figure 3.19: Middleplate.

The middle plate is the base for the upper feed drive. This means that like the base plate it needs to be stiff but since it is also a part of the moving system it needs to be light to reduce the dynamic forces during acceleration and retardation. To fulfill these contradicting requirements the middle plate has a complex design with parts overlapping to save space and triangular cut-outs to keep the weight down while still being strong enough to withstand the forces during acceleration. With these requirements in mind the material chosen is Alumec which is an aluminum alloy that is stronger than ordinary aluminium while still having the same density [Uddeholm, 2016].

The holes for the rails have counter bores on the bottom for the nuts which makes it possible to use the same alignment system as on the lower rail while

still avoiding the nuts from colliding with equipment on the base plate.

To help in the optimization a finite element analysis was done to confirm the robustness. The analysis showed that the first of two main sources of deflection was during maximum acceleration of the middle plate in the connection to the lower ball screw.

This first analysis can be seen in Appendix B.1 where $500N$ was applied in the connection to the lower ball screw. This was to emulate $50kg$ in 1g acceleration. The maximum deviation here was approximately 3 µm which was deemed acceptable.

The second scenario was when the middle plate would flex when the top plate was in edge positions. This analysis can be seen in Appendix B.2 where 2x30N was applied on one edge. This emulated a $6kg$ plate on the edge of the construction. The maximum deviation here was approximately 3 µm which was deemed acceptable.

Motor bracket upper

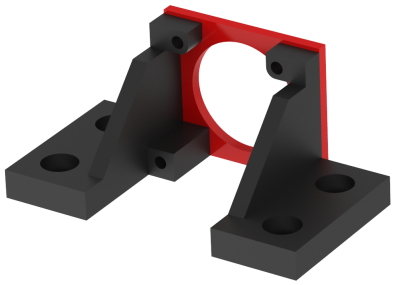


Figure 3.20: The assembly to hold the motor in place on the middle plate.

The assembly seen in Figure 3.20 connects the motor to the middle plate. The plate in red is made out of rubber to reduce vibrations from the motor while the black parts are made out of steel.

Limit switch holder for the middle Plate

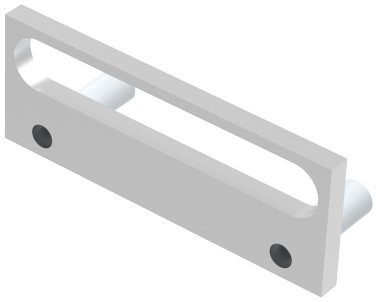


Figure 3.21: Limit switch holder.

Much like in Section 3.1.3, two limit switches will be placed at each end of a rail to hinder out-of-bounds movement. The construction as seen in Figure 3.21 is a plate with an M10 hole for the limit switch. It is then fastened to the middle plate through two $20mm$ spacers with M5 bolts. This assembly allows for the limit switch to be moved in two axes.

Ball screw connector

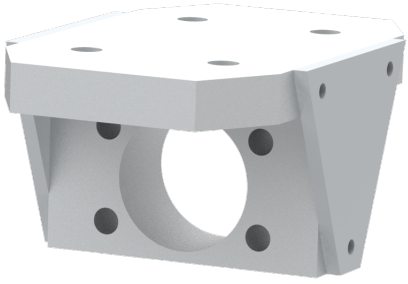


Figure 3.22: Fastener to connect the ball screw to the upper and middle plate.

The ball screw fastener, seen in Figure 3.22, allows the movement of the ball screw to be transferred to the middle and upper plate of the testbed. The

ball screw connects to the fastener via the ballscrew-nut seen in Figure 3.6, that fits in the center hole of the fastener with four additional M5 bolts. The ball screw fastener connects to the upper and middle plate with four M5 bolts whose holes have counter-bores to ensure a collision free connection for the bolt and nut. The fastener is reinforced with two triangle formed side plates on each side to prevent axial displacement.

Top plate

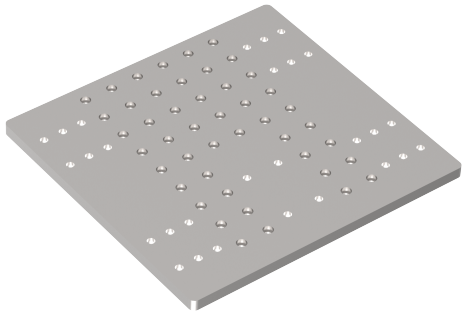


Figure 3.23: The working surface to place sensors on.

The plate seen in Figure 3.23 will be the top and working surface. It will be movable in two axes and carry the different sensors. To allow for different sensor setups the whole plate is covered in holes for M5 bolts with counter bore on the lower side to avoid collisions. The plate is fastened in the guides with four sets of six M6 bolts as well to the ball screw fastener with four M6 bolts.

3.1.4 Assembly

Table

The legs where attached to the table. To make sure that the table were straight a machine spirit level were used. The spirit level measured along the diagonals of the table and the legs were lowered or extended depending on what the spirit level showed. As components are mounted on the table the height of the legs are adjusted again due to the extra weight being added.

Base plate

The first step in the assembly is to bolt the base plate down to the table, which causes the plate to bend. To measure the flatness of the base plate a height measuring tool is used to measure the height at different points along the x and y-axis, since it is lowered towards the surface by hand there is a big uncertainty in the actual height difference but the profile of the plate should be accurate. Figure 3.24 shows the profile of the base plate after the first tightening. To fix this shims can be placed between the base plate and the table where the plate is bending downward.

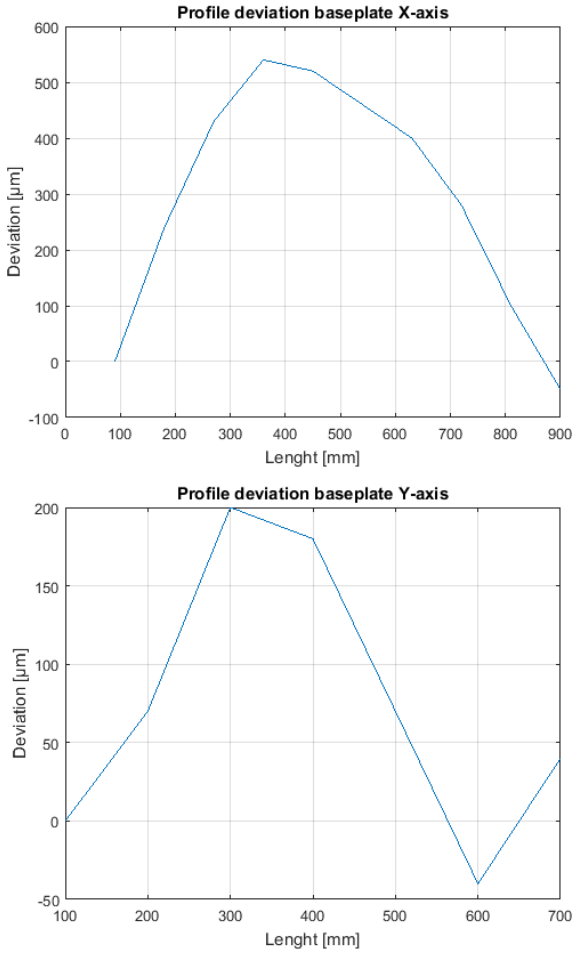


Figure 3.24: A profile of the base plate in X and Y

Rail measurement

Before the rails is installed on the rail spacers, see Section 3.1.3, a measurement is made to get a better understanding of the initial imperfections of the rails. This measurement could also be used to better assemble the construction as to tighten screws in a way that counters the initial imperfections. The rails were measured in a Coordinate Measuring Machine (CMM) which has a low enough error of margin to successfully evaluate the straightness of the rails.

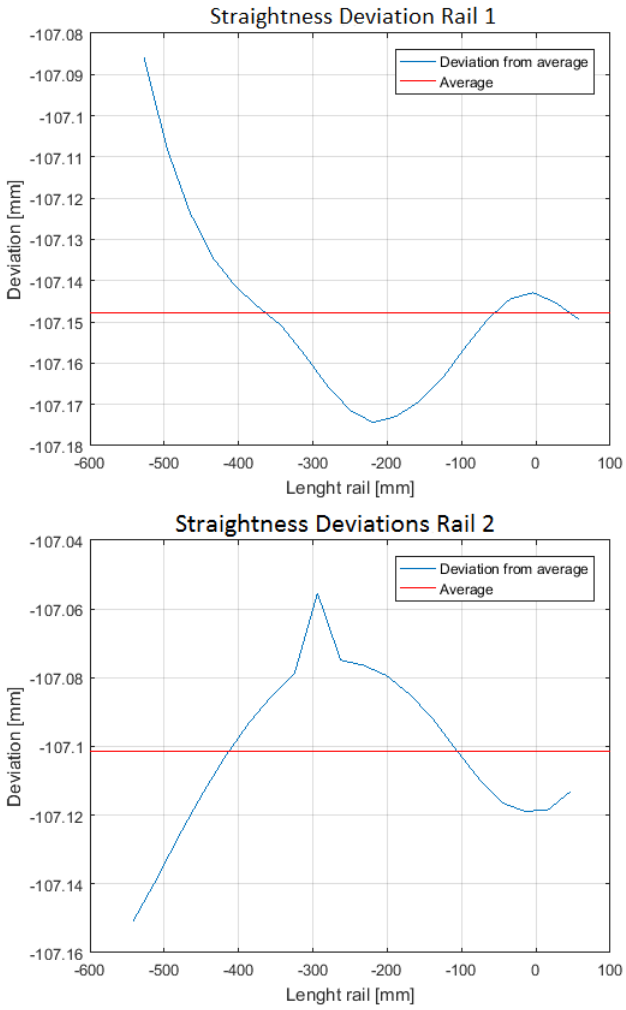


Figure 3.25: Straightness deviation of bottom rails

Figure 3.25 shows the straightness of the rails for the bottom level of the testbed. The measurement on rail 1 resulted in a total straightness deviation of 68.7 µm which is the distance between the highest and lowest point of the measured surface. The measurement on rail 2 resulted in a straightness deviation of 90.4 µm.

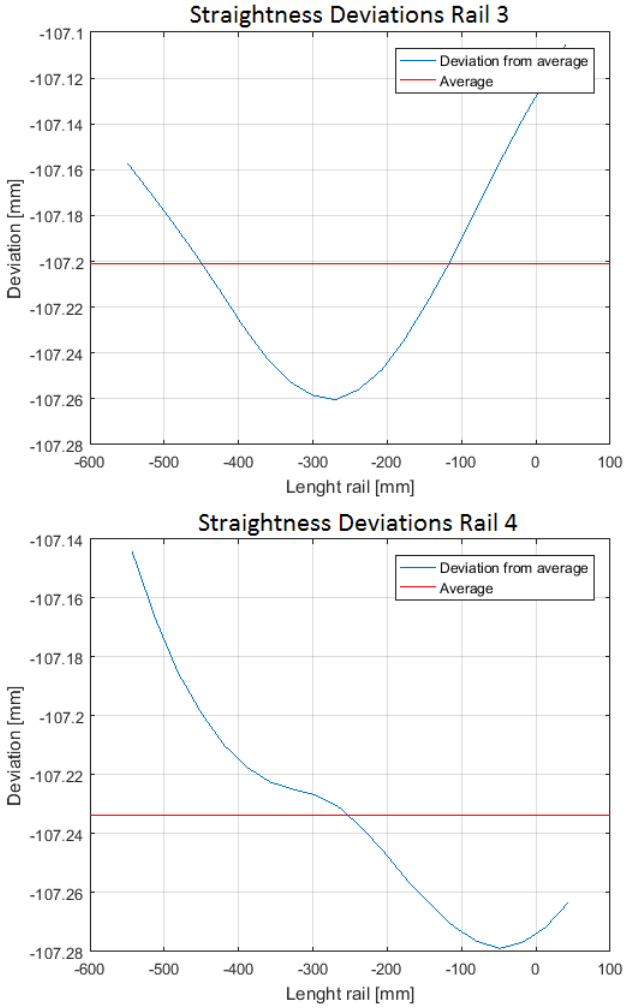


Figure 3.26: Straightness deviation of top rails

The straightness deviation on the top rails is represented in Figure 3.26. The measured straightness of rail 3 and rail 4 are 129 µm and 42 µm respectively.

Lower feed drive

The rails are mounted following the NSK catalogue [NSK, 2016]. First the master rail's datum face is pressed towards the reference surface in Figure 3.13 using the clamps in Figure 3.18. The clamps and the master rail's bolts are tightened to recommended values. This ensures that the rail is in line of the reference surface. The slave rail is mounted with its datum face inwards and lightly tightened to still allow movement. The two sliders are added onto the rails with their datum sides on the same side as their respective rail.

The ball screw connector seen in Figure 3.22 is mounted and tightened onto a ball screw and then tightened onto the middle plate in Figure 3.19. The two bearing houses for the lower plane were mounted and the prepared ball screw with middle plate inserted. The middle plate is then tightened onto the guides.

To align everything the middle plate was moved back and forth multiple times and the bolts on the slave rail and the bearing houses were procedurally tightened to the recommended value.

Upper feed drive

For the upper feed drive the same procedure used for the lower feed drive is used except that the bearing house further away from the motor is tightened first. The reason for this is that there is so little play between the bearing house and the middle plate so the ball screw could not align itself properly.

Limit switches

The limit switches are mounted to the holders and their position adjusted so they trigger when a slider passes by and with enough distance to the edge of the rail so the platform can stop in time.

3.2 Electrical Hardware

3.2.1 Architecture

The electrical hardware consists of everything that is needed to control the motors and read the sensors as well as create a human machine interface. The high-end requirements of this project made it necessary to use high quality hardware.

The setup is chosen in collaboration with National Instruments to ensure compatibility across the components and it contains three different main parts. The PC, which provides the human machine interface as well as the data logging, the CompactRIO and drives, which are programmable real-time systems with control loops, and the motors and sensors, which are the parts that interact with the environment.

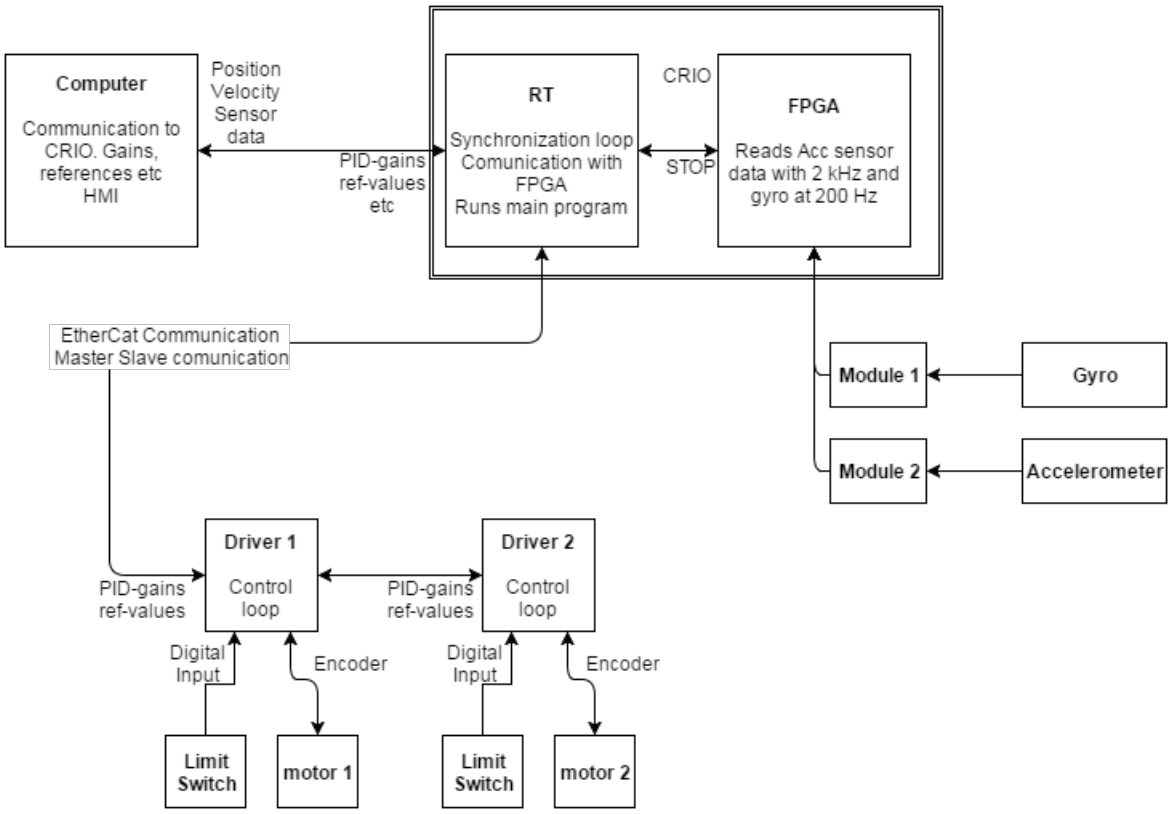


Figure 3.27: Electrical Hardware Architecture

An overview of the electrical hardware can be seen in Figure 3.27. The computer is running National Instrument's LabVIEW which communicates with the NI CompactRIO unit via an USB interface. The CompactRio is running code in real-time on its RT and FPGA units which are communicating with the drives over the EtherCAT interface. The CompactRio sends position requests to the drives and built-in control loops in the drives perform the necessary movement of the motors, to synchronize the movement of the two different axis the CompactRio receives position information from the drives. The drives alternates the frequency over the three-phases to control the speed and direction of the motor and a built-in encoder in the motor sends position information back to the drive. The limit switches are directly connected to the drives and commands a stop of all movement if triggered.

3.2.2 Host PC

The host PC could be any computer that meets the requirements for installing NI's LabVIEW suite and has an USB port for communication with the CompactRio. In this setup a Lenovo laptop with Windows 7 is used. The host PC connects to the CompactRio via USB and communicates using protocols built in to LabVIEW.

3.2.3 CompactRio

Figure 3.28: CompactRio 9030

The CompactRio is a real-time embedded industrial controller made by National Instruments. It contains a real-time controller, reconfigurable I/O modules, a FPGA module as well as an Ethernet expansion chassi. The CompactRio has USB ports for communication with a PC and EtherCat ports which could either be used for communication with a PC or additional hardware modules. The system also comes with serial ports as well as slots to connect I/O modules. In this project the NI-cRio 9030 was used. It uses a 1.33GHz dual core Intel Atom processor, 1GB RAM, 4GB system memory and comes with four slots to connect I/O modules [Instruments, 2015b].

The CompactRio acts as the main processing unit in the setup. Code written in LabVIEW is deployed to the CompactRio and its built in RT and FPGA units. It coordinates the data logging to the host PC and communicates with the motor drives over EtherCAT. Two I/O modules, one analog and one

digital, collects data from the accelerometer and gyroscope which is processed by the CompactRio.

RT System

The RT system is a part of the CompactRio and is one of two processing targets to which LabVIEW code can be deployed. The real-time processor is used for signal processing and communication software.

FPGA

The FPGA, or the Field-Programmable Gate Array, is the second processing target of the CompactRio. It consist of arrays of programmable logic blocks that can be wired together in different configurations using software. The logic blocks are programmed with flip-flops which stores a single bit of information each. This means it can have excellent performance for simple tasks with demanding sample rate, but there is a significant trade off in terms of the amount and type of programs that can be deployed to it, it is time consuming to compile to and it cannot handle floating point numbers. The FPGA unit used in this setup is the Xilinx Kintex-7 7K70T with 82000 programmable flip flops and 4860 kbits of block RAM [Instruments, 2015b].

3.2.4 Drives

To control the motors two Kollmorgen AKD servo drives were used, one for each motor. The drives include a closed-loop feedback system for controlling the motors with the option to change the control loop parameters. Apart from controlling and driving the motor the driver also contains a number of pins for connecting limit and home switches. By connecting the limit switches directly to the drive a fast response to a triggered limit switch can be ensured. The drives also have advanced functions for autotuning the gains of the control loops as well as for checking the performance.
The drive communicates with the CRio using EtherCat and through a software interface built into LabVIEW the control parameters can be accessed and changed. While running the CRio will send position or velocity requests and the drive will execute the command and deal with all the necessary control, this means that after an initial tuning of the control parameters, the drive can be seen as a black box in the system design.

For full technical specifications of the AKD servo drives refer to the Kollmorgen AKD servo drive specification sheet.

3.2.5 Electrical Motors and Encoders

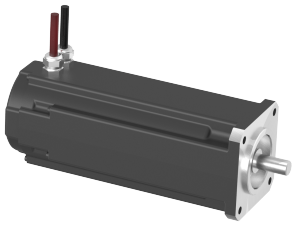


Figure 3.29: Kollmorgen AKM-24D.

Two three-phase brushless AKM-24D servomotors made by Kollmorgen (Figure 3.29), capable of providing a maximum speed of 8000rpm and a peak torque of 4.76Nm, are used as actuators. They come equipped with an analog resolver which reads the motor shaft position by measuring the induced current in windings placed around the rotor. Kollmorgen's proprietary SFD3 (Smart Feedback Device) feedback system converts the analog resolver signal to 24bit position data and communicates this digitally, alongside motor temperature, to the drive using a RS-485 serial communication interface, see Table 3.1 for more specifications.

Motor Specifications	
Max DC bus voltage	640V
Stall Torque	1.41Nm
Maximum speed	8000RPM
Peak Torque	4.75Nm
Feedback Data Specifications	
Resolution	16,777,216cpr = 0.0000216°
Accuracy	< +/- 0.1625°
Sample frequency	19.5kHz

Table 3.1: AKM-24D servomotor specifications

3.2.6 Limit Switches

To ensure that the motors does not operate out of bounds and risk damaging the setup, hardware limit switches are used. Installing one limit switch in both ends of each axis brought the total to 4 limit switches. The type used are M10 Inductive Proximity Sensor PNP, seen in Figure 3.30 which will detect the presence of metal in close proximity in front of it and toggle a high signal to low on its data pin (active high) [Lamb, 2016]. This is not optimal in terms of safety and it is recommended to invert this signal to prevent false negatives when power failure is inflicted on the limit switches. In addition to the four limit switches, two identical switches were intended to be used as home switches but were never installed because of lack of cables.

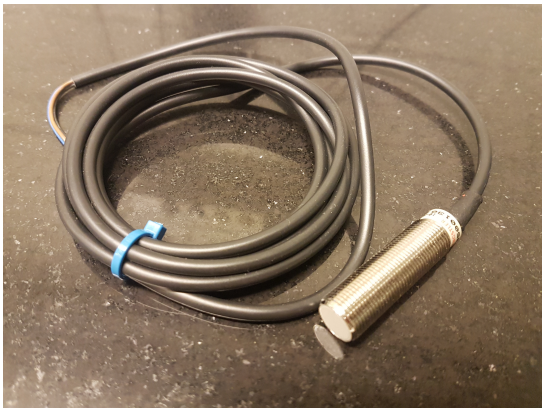


Figure 3.30: One of the limit switches.

No datasheet was acquired so all information had to be estimated. The sensor trigger distance is approximately 3mm. The supply voltage is 10-35 VDC.

3.2.7 IMU

The IMU, or inertial measurement unit, is a device that measures a body's angular rate and specific force using accelerometers and gyroscopes. In this setup they are placed on the working surface and the cables led through cable guides connecting them to the stationary CompactRIO unit where data can be processed and logged. To connect them to the CompactRIO, two IO

modules were used, one NI 9230 Analog Input module to read analog signals and one NI 9871 RS485/RS422 Serial Interface for digital communication.

Accelerometer

Figure 3.31: Dytran 7503D1 Accelerometer

The accelerometer used in the setup is a Dytran Instruments model 7503D1. It measures the acceleration in three axes in a range of +/- 2G. It connects using a 9pin connector, two pins for powering the unit and two pins for each axis for transferring analog data, the ninth pin is not used. The output data is an analog signal where 2 volts corresponds to one positive G. Using the NI 9230 Analog input module each of the three axis are read by the compactRIO separately. Some important specifications can be found in the following table.

Specifications	
Input Range	+/-19.6 m/s^2
Frequency Response	0-400 Hz
Sensitivity Differential	204 $mV/m/s^2$
Output Noise, Differential	69 $\mu m/s^2/\sqrt{Hz}$

Table 3.2: Specifications of Dytran Instruments 7503D1 [Instruments, 2015a]

Gyroscope

Figure 3.32: G200D-100-300 digital triaxial gyro

The gyroscope used is a G200D-100-300 digital triaxial gyro supplied by Gladiator Technologies. It measures the rate of change of the angle in three axes thus providing information about the working surface's yaw, pitch and roll. The gyroscope uses a digital interface and some internal processing and filtering is done in the unit. Parameters such as sample rate, filter bandwidth and baud-rate can be accessed and modified by connecting the unit to a PC using a proprietary USB dongle and the bundled Glamr software. This software also allows for testing the performance of the sensor and to edit communication protocol details. The gyro is individually tested before delivery to verify that it meets the specifications and the results are delivered together with the device. The communication with the compactRIO is done via a NI9871 module which provides support for RS-485 serial communications, in this setup a 1kHz sampling rate was used. The gyro requires external power to operate, this is currently provided by a battery pack and the unit has to be manually switched off with a switch after use to stop the batteries from draining.

Specifications	
Range	+/-100°/s
Angle Random Walk	$0.002°/s/\sqrt{Hz}$
Bias	0.0011°/s
Sensor Resolution	0.001°/s
Max update rate	2.5kHz
Bandwidth	200Hz

Table 3.3: Specifications of G200-100-300 digital triaxial gyroscope [Technologies, 2014]

3.3 Software

As this testbed platform is intended to be constructed for future research, one important requirement of this project was to have a setup which was easy to use and to modify with a functional graphical user interface (GUI). Additionally, real-time constraints exist to control the testbed and sampling of the different sensors. Another relevant consideration is that this platform must be flexible in the sense of having the possibility to incorporate new hardware or use new sensors with minimum effort.

Given these requirements LabVIEW was chosen as the framework to work with. LabVIEW (Laboratory Virtual Instrument Engineering Workbench) is a system-design platform and development environment for a visual programming language from National Instruments. LabVIEW programs are called Virtual Instruments (VIs). Graphical blocks substitute the type of code seen in traditional programming language, which provide an enormous abstraction and ease-of-use for the user. The programming is done at a very high level allowing for great reusability and scalability. For example, some programming structures, such as threads, that would take hours to code in general purpose programming languages like C/C++ take minutes to implement in LabVIEW with parallel loops. Other advantages concerns the ease of implementing better memory management or control access to common resources.

This together with the NI hardware provides a real-time platform, consisting of a real-time system, a real-time operating system and a FPGA for higher

performance, and makes the integration of the different subsystems easy and seamless, reducing development times and facilitating rapid prototyping and debugging. On the other hand it is needed to have a good planning when programming, otherwise the graphical code will be disorganized making it difficult to maintain and upgrade. It is also a challenge to work in parallel using LabVIEW and for experienced text based programmers it can be hard to get used to this new programming paradigm at the beginning.

LabVIEW video-tutorials have been used to build a solid knowledge base in order to program the different subsystems following NI best practices and architectural design patterns. It is imperative to understand how a real-time system works and how this is linked to the functions when coding. For instance, one must be aware of what functions or structures are deterministic or not, how to send information from different parts of the system in real-time, how to synchronize different parts of the code or how to sample data without losing a single value.

The software of the system is divided into three main parts. It consists of a first VI running on a Windows host computer, a second VI running on the real-time system, which is a CompactRio, and a third VI that runs on the FPGA of the CompactRio. This architecture is depicted in Figure 3.33.

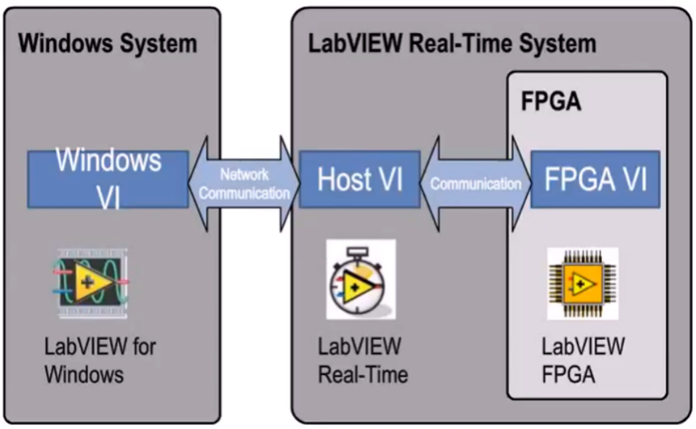


Figure 3.33: CompactRio common architecture.

3.3.1 Host PC

The program/VI running on the host PC is the GUI and it acts as the human machine interface (HMI) sending control commands to the testbed computer (i.e. CompactRio) and also monitors its status as well as of the FPGA. The final interface of this program can be seen in Figure 3.34.

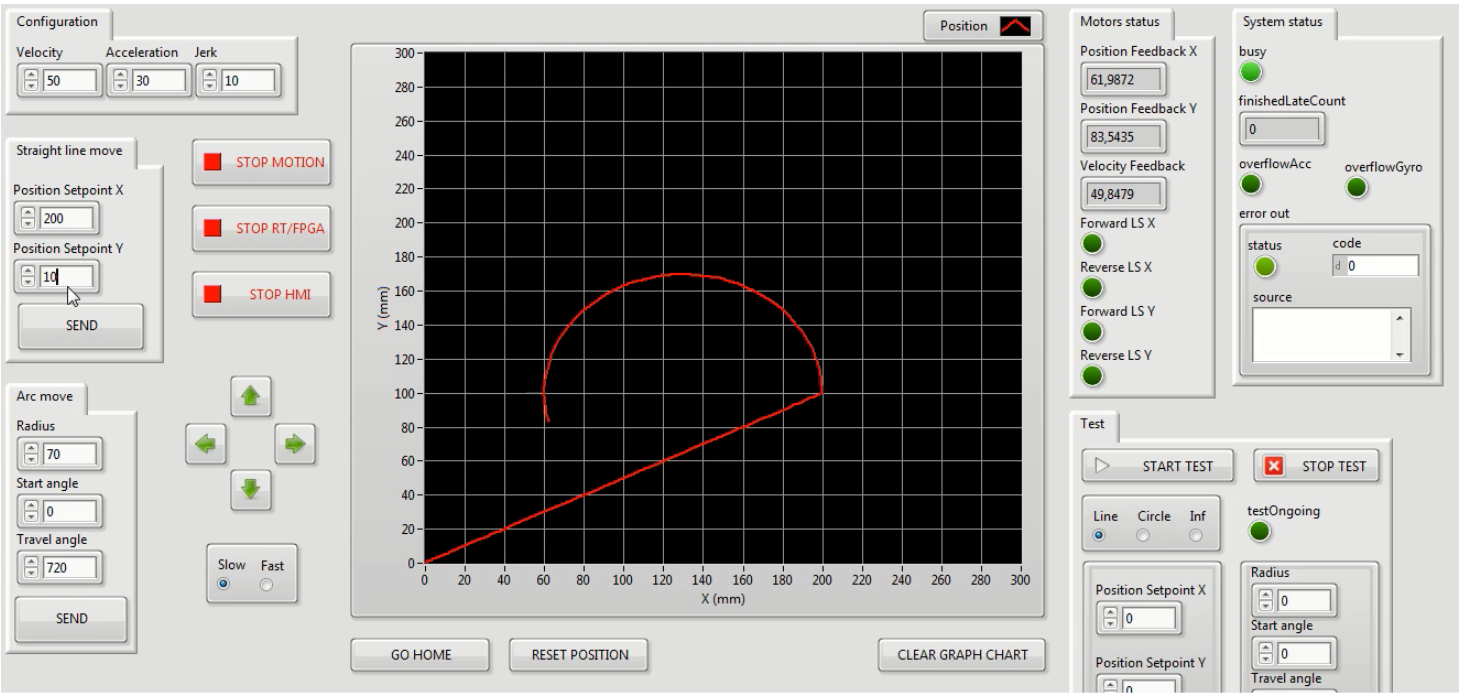


Figure 3.34: Graphic User Interface (GUI) in the host PC.

This VI is divided into two main parts, an infinite while loop that reads and display information from the CompactRio (communication loop) and an event loop structure used to actuate and send commands to the testbed computer.

The communication loop can be seen in Figure 3.35 and it is quite simple. It is executed every $50ms$ and reads and display the following information: Data from the motors/drivers (positions, velocities, limit switches etc.), indicators regarding the status of the CompactRio (commands going on, tests being executed and number of times that the deterministic control loop fin-

ished late) and indicators concerning the status of the FPGA (overflows in the data acquired from the accelerometers or gyros and consequently loss of data). The data is shared using network-public shared variables with the option RT FIFO enabled when the data is being sent from or to the deterministic control loop.

The loop finishes when the stop_HMI button is pressed or the program is stopped but this does not imply that the program running on the CompactRio will finish.

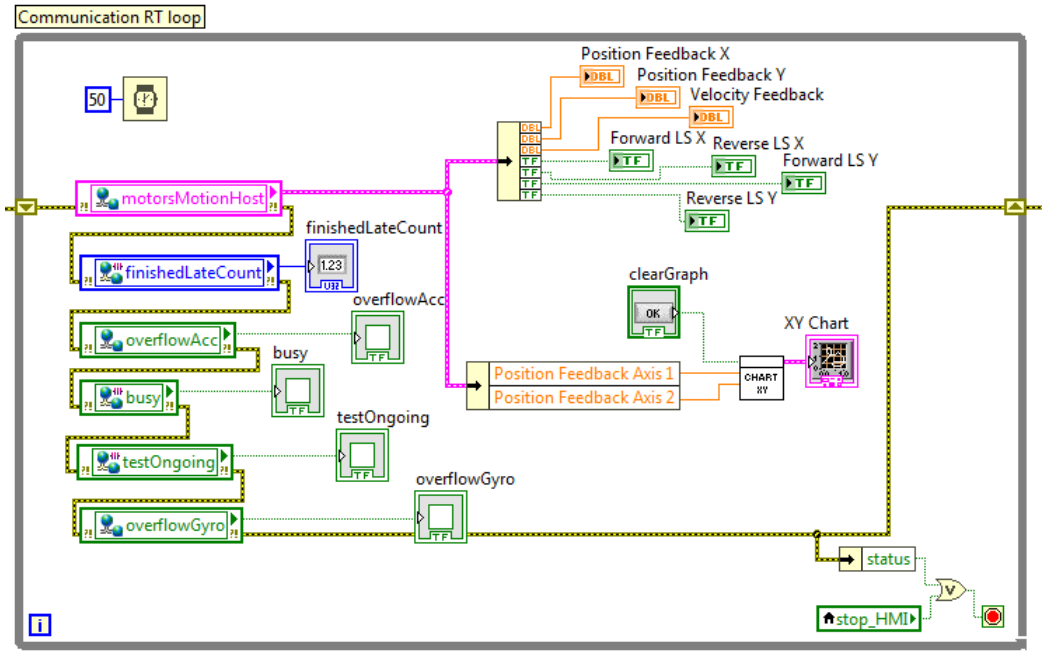


Figure 3.35: Communication loop with the RT host.

The event loop consists of a case structure inside a while-loop with a specific case for every button in the GUI. The events are triggered when the buttons are pressed and depending on the button a different network-public shared variable is sent to the CRio. Examples of the possible events are: set straight line, set arc, stop CRio and FPGA, stop HMI, reset position, perform test and move forward. In Figure 3.36 an example is shown for the set_straight_line button event. As it can be seen some parameters are read from the controls in the GUI and sent to the testbed using the shared variable motorsSetLine.

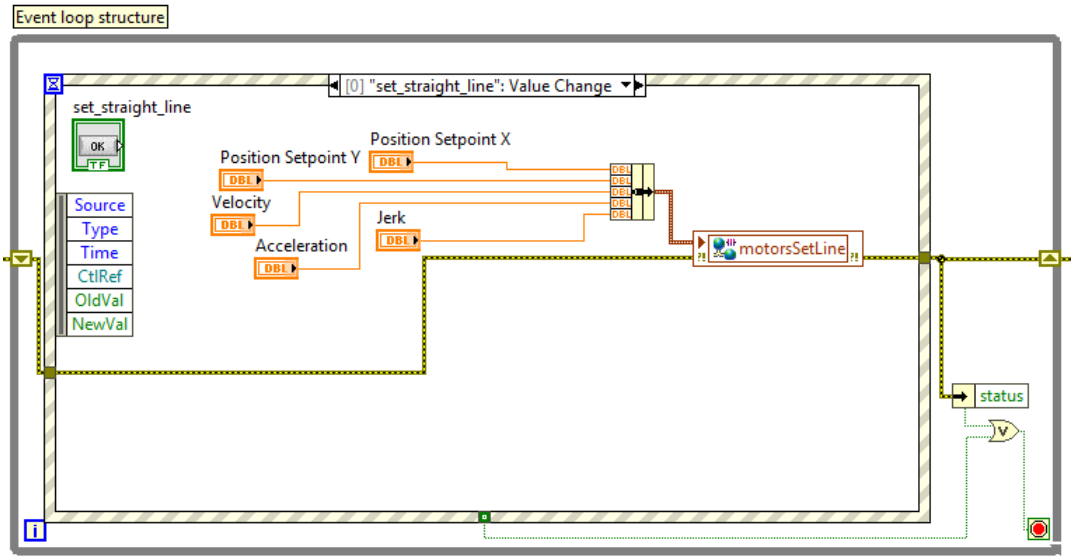


Figure 3.36: Event loop structure to send commands to the RT host.

3.3.2 CompactRio / RT host

The program/VI running on the CompactRio is the most important one, it is the logic of the testbed and it is the responsible for its control, logging and monitoring. The structure of this VI is depicted in Figure 3.37. At the beginning there is an initialization section where the different network-public shared variables and inter-process variables are initialized, in this part FIFO buffers are cleared and a reference to the FPGA is opened as well. Then follows the main part of the program, which is composed of three main loops (the deterministic control loop, the FPGA communication loop and the data-logging loop) and finally in the shutdown section, the FPGA is reset to its initial state and the reference to it is closed.

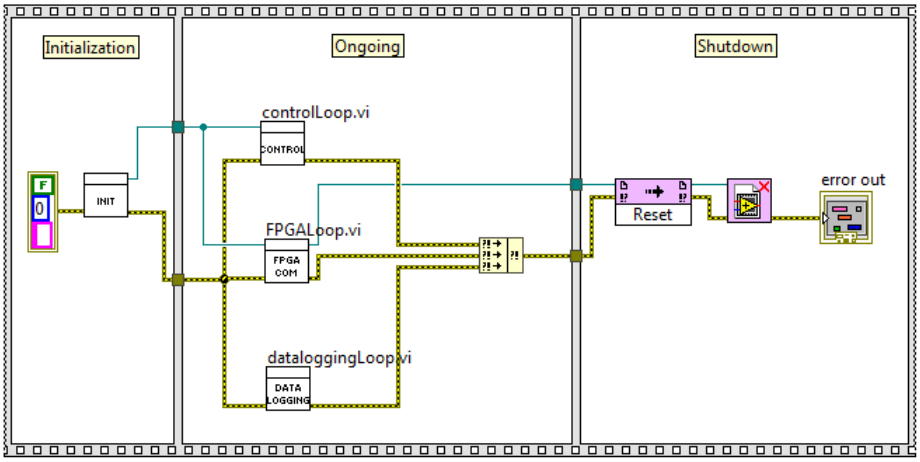


Figure 3.37: Structure of the main program in the RT host.

The deterministic control loop, which can be seen in Figure 3.38, is running at 1kHz and is responsible for reading some of the commands sent from the host PC and sending them to the drivers so that the testbed can move in lines, arcs, etc. At the beginning the motors are enabled and then the timed-loop starts. In every iteration the program checks for updated values received from the host PC and if a new value has been received it performs the corresponding action. To make the code look better and increase the modularity this is implemented in multiple subVIs. In addition information about the drivers/motors is read and sent to the data-logging loop to save this data into a file.

Other important feature of this loop is the option to do tests in order to identify the induced geometric errors. This functionality is implemented inside the controlTests subVI. When a test is going to be started this subVI asserts an interrupt in the FPGA so that both the CompactRio and the FPGA start logging data to the files at the same time in a synchronized way and meanwhile the FPGA is waiting for this interrupt to be asserted. It is important to underline that the CompactRio and the FPGA are only synchronized at the beginning of the tests and then it is assumed that its period will remain constant so that knowing the rates of every loop it is possible to map the data acquired in the CompactRio and the FPGA. This is a critical task since one of the most important requirements of this testbed platform is to perform repeatable tests so it is a strong requisite for the

software to be able to repeat these tests in position and time.

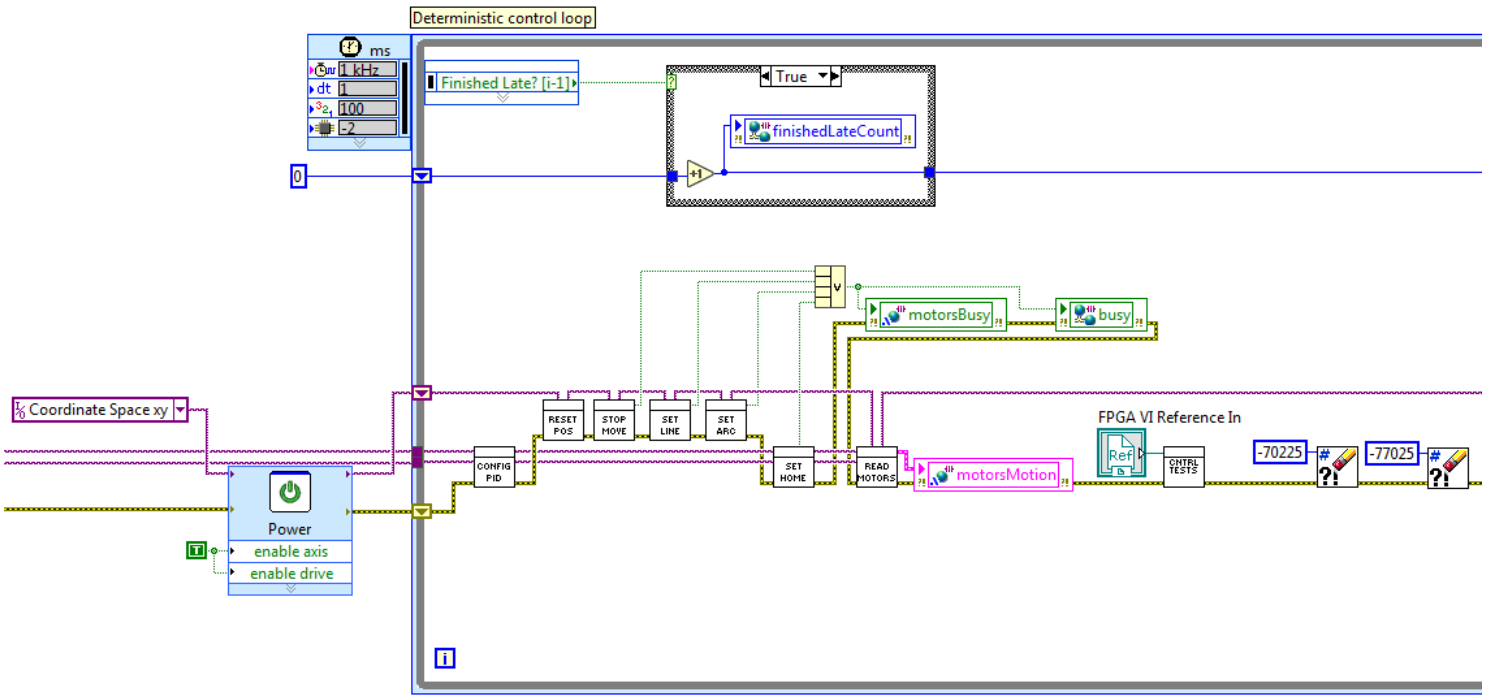


Figure 3.38: Control loop in the RT host.

In Figure 3.39 the communication loop with the FPGA is shown. The purpose of this loop is to read the data sent from the FPGA, consisting of FIFOs, and log it into different files. As it can be seen the data is only read and logged if a test is being executed. Firstly the accelerometer FIFO is read and logged and then it is the gyroscope FIFO. Overflow information from the FPGA is also sent to the host PC. This loop is executed every 25 milliseconds. The most interesting to comment about this loop is that the FIFOs need to be read faster than they are written to avoid having overflow problems and consequently losing data. To manage this it is needed to take into account the writing frequency to the FIFOs in the FPGA as well as the FIFOs sizes.

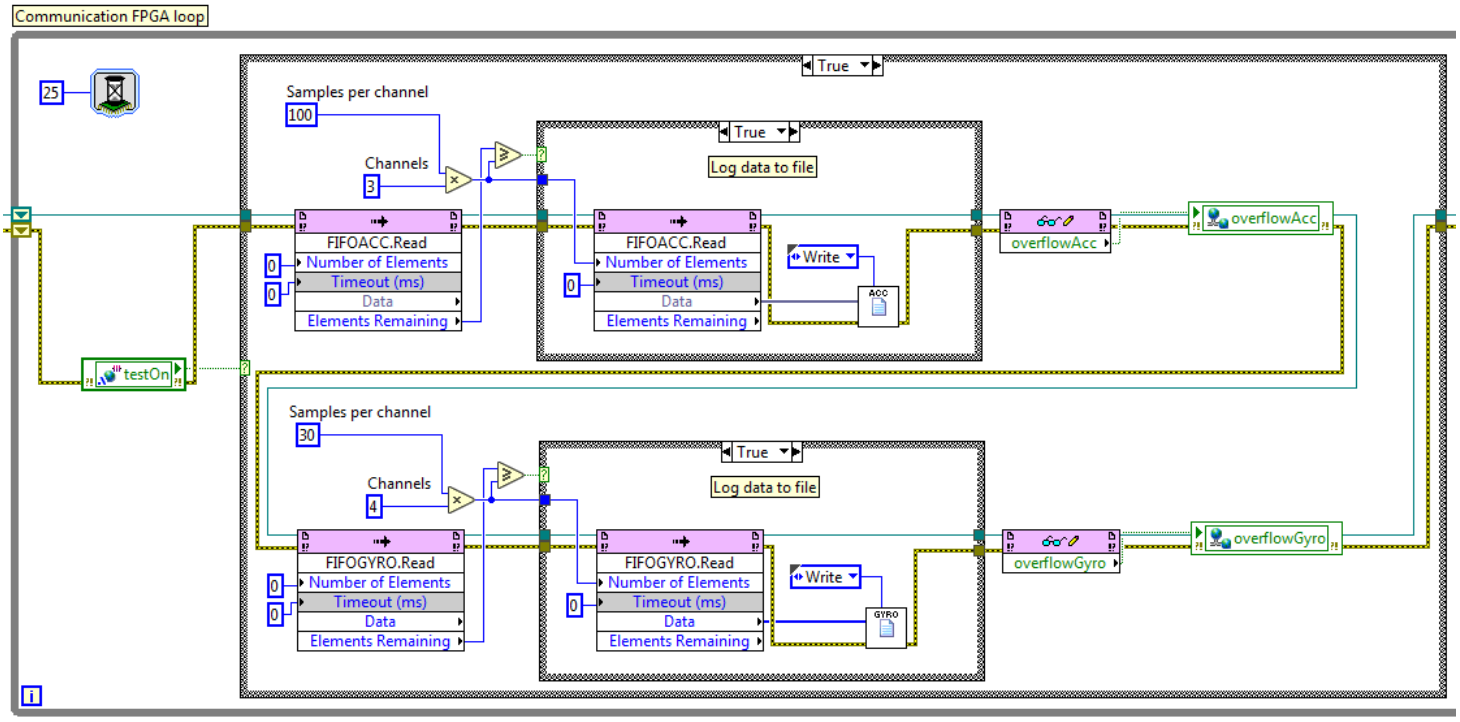


Figure 3.39: Communication loop with the FPGA in the RT host.

The data-logging loop (Figure 3.40) runs at a frequency of 10 Hz and its main function is to log data from the motors and send some of this data to the host PC for monitoring the testbed. Furthermore, this loop is also in charge of creating new files when a test is executed and closing the files when the test is finished.

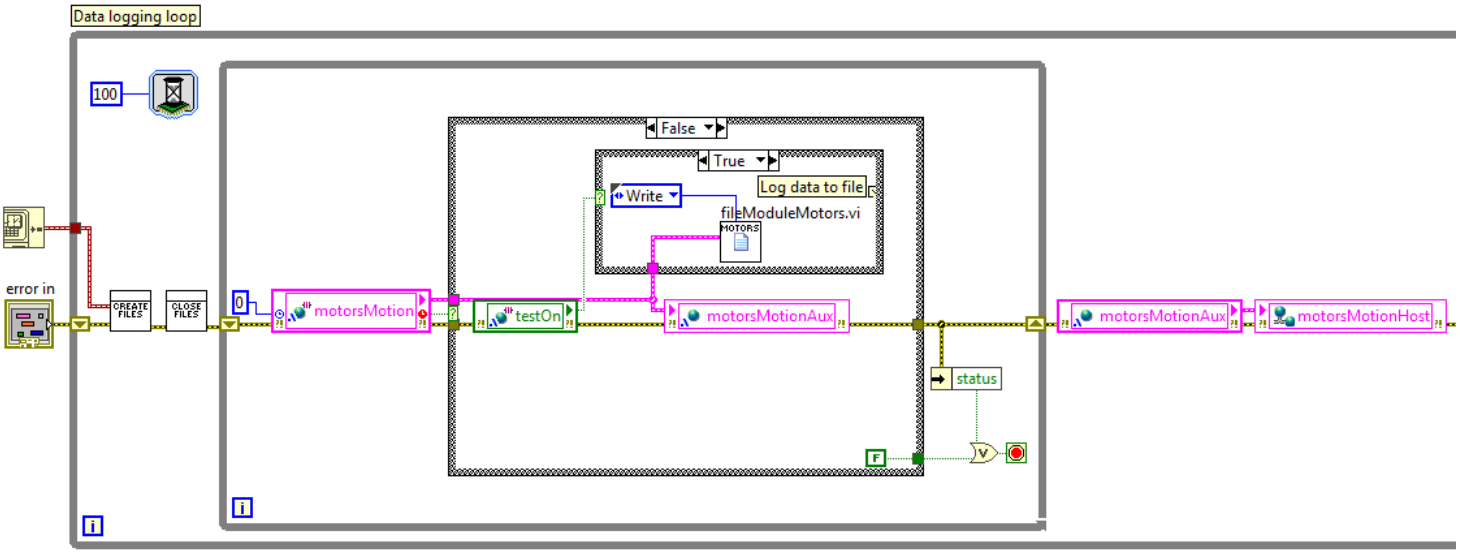


Figure 3.40: Data logging and PC communication loop in the RT host.

3.3.3 FPGA

Although the CompactRio is using the Scan Interface feature for the communication through Ethercat with the motors drivers, it is also using the FPGA interface mode to handle the communication with the C series modules to read the accelerometers (NI 9239) and the gyro (NI 9871). This is, the CompactRio is configured in a hybrid mode taking advantage of both options.
This choice was made because the FPGA interface mode is more flexible when it comes to reading the sensors at the desired frequency. If the scan engine is used, the scan period is the same for all the I/O channels. The behaviour of the scan engine can be seen in Figure 3.41.

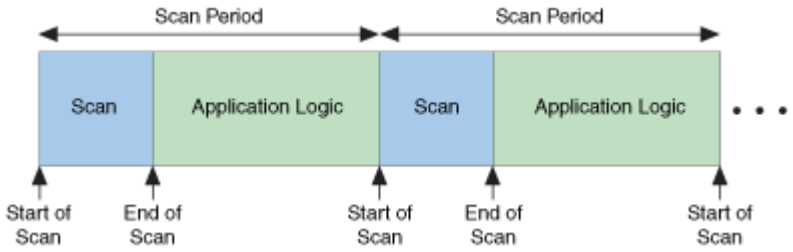


Figure 3.41: NI Scan Engine timing.

In addition, the maximum NI scan engine rate supported on CompactRIO controllers is 1kHz and some sensors might need to be sampled at a higher frequency. Besides, the scan mode adds considerable overhead in the RT processor and hence it reduces the fastest loop rates (Figure 3.42).

Number of Channels	Scan Mode (Hz)[1]	FPGA Mode (Hz) [2]	Compact FieldPoint (Hz) [3]	Machine Control Architecture (Hz) [4]
1	1000[5]	7100	605	435
16	1000	3500	313	not tested
80	399	974	n/a	not tested

Figure 3.42: Fastest loop rate for T2 test with different HW/SW setup ([NI, 2015]).

The code in the FPGA is divided into two parts; a loop responsible for reading the accelerometers and another loop used to read the gyroscope. In Figure 3.43 the loop that is reading the accelerometers is illustrated. The code is quite simple, the program is reading three analog channels of a NI 9239 C Series module and then is copying this data into a FIFO which will be read later in the CompactRio. The communication between the FPGA and the CompactRio is made using a Direct Memory Access (DMA) FIFO channel (the CompactRio used in this project provides 16 of these channels). It is also interesting to remark that the frequency of this loop is set by the NI 9239 C series module as it is internally timed. This module is capable of sampling at frequencies from 50 KHz to 1,613 KHz. A choice of 2 KHz was made considering the state of the art made and the capabilities of the other sensors present on the system.

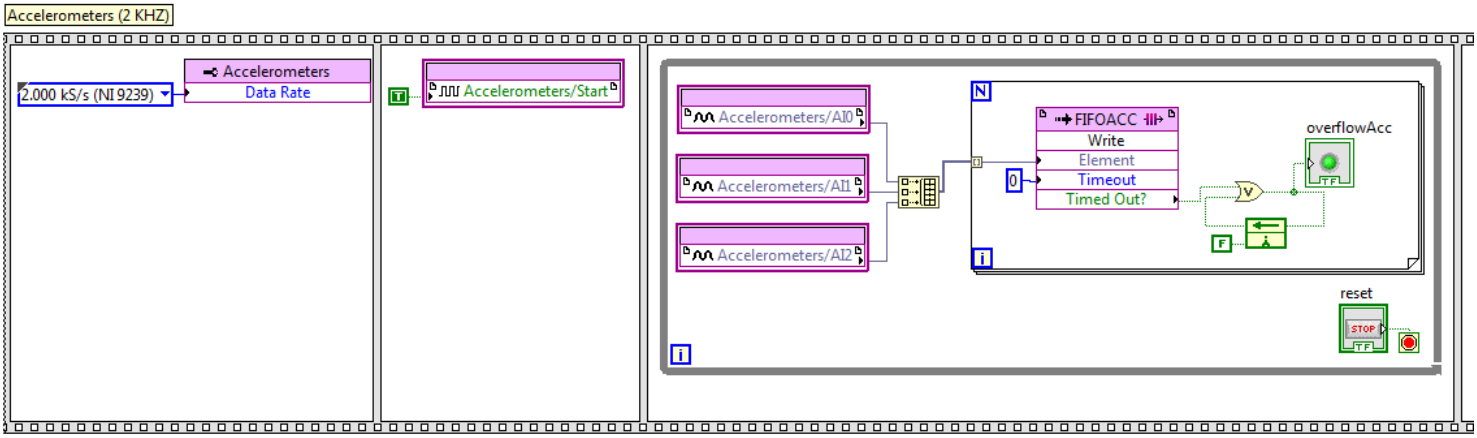


Figure 3.43: Accelerometers reading loop in the FPGA.

On the other hand, the gyroscope used in this project sends a digital signal so a digital C series module is needed, like the NI 9871 used in this project. This digital signal consists of a stream of 12 bytes starting with a header ("0x2f") followed by a message counter to identify lost messages, the actual angular speeds for X, Y and Z, the temperature, a byte indicating some properties of the sensor and a checksum (two's complement sum). The implementation is depicted in Figure 3.44, as it can be seen when there are more than 12 bytes available in the serial port the first byte is read and if the value of this byte equals the header of the message, then the 11 left bytes are read. Finally the checksum is calculated and if it is correct, the angular speeds are decoded and the values are copied into a FIFO which will be read again later in the CompactRio.

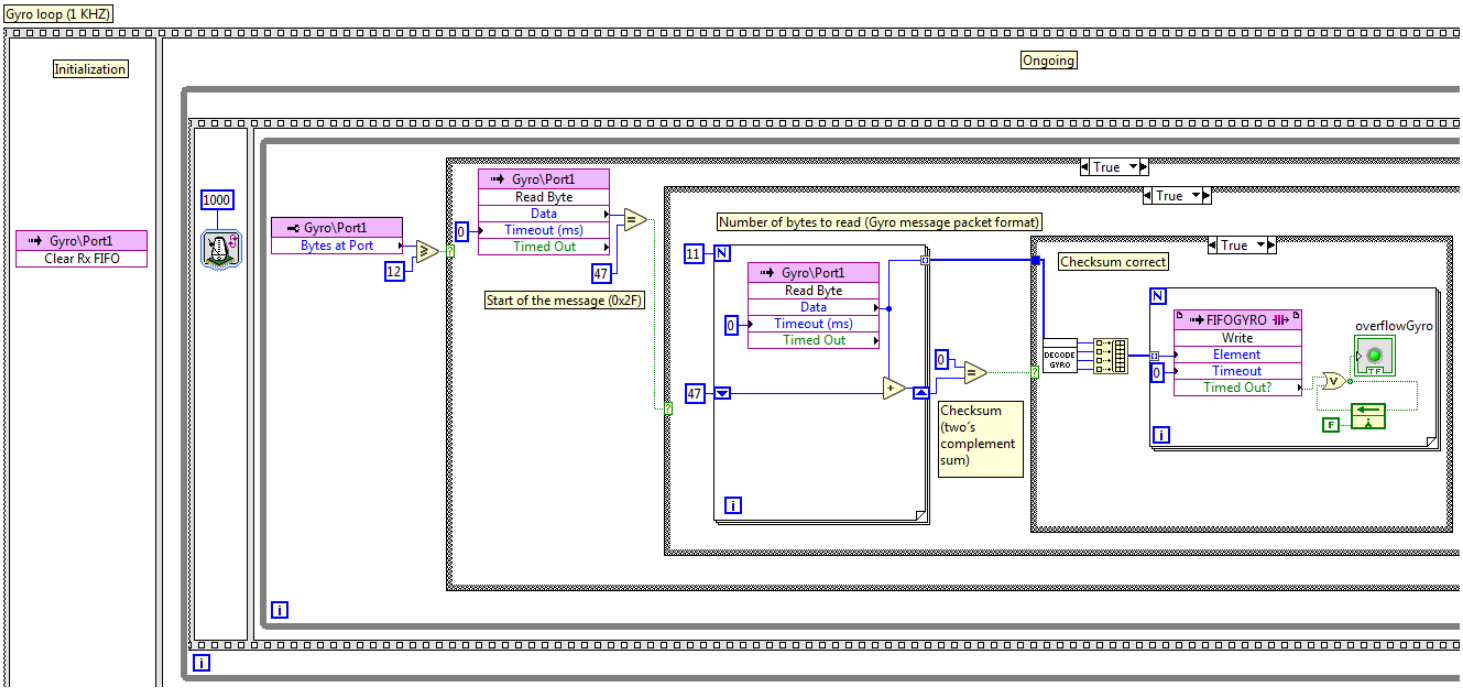


Figure 3.44: Gyroscope reading loop in the FPGA.

3.4 Control System

As it has been explained before, the position control of the motors is done internally in the motor-drivers. The deterministic control loop in the CompactRio acts as an interface between the software and the hardware (drivers) and it is used mainly to send references to the drivers (straight lines, arcs, etc), update PID gains and read data from the motors and the state of the limit switches. In order to manage all this, LabVIEW provides a library for motion control called SoftMotion. SoftMotion library provides an easy way to implement functions such as straight lines of both single axis and coordinate tables, arcs etc.. Although the functions are done on a high level is possible to to change and configure a lot of parameters.

The control loop inside the drive consists of a cascade control architecture to control the position (although it is possible to choose to control the velocity instead).

The motion controller contains supervisory control, trajectory generation, a spline engine, a control loop, and I/O. In addition, the controller also monitors the system for errors, certain events and faults in the system.

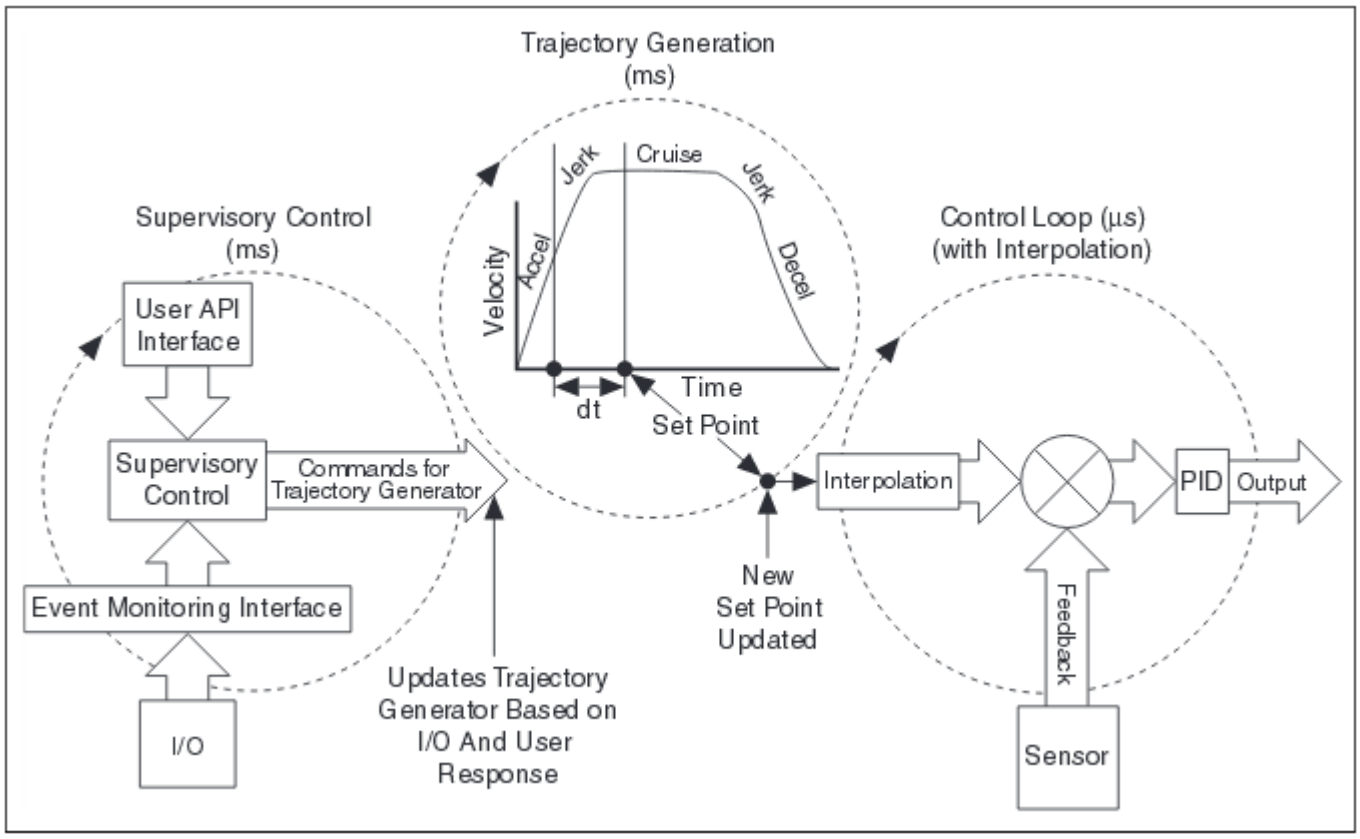


Figure 3.45: Three main loops of a typical motion controller.

In Figure 3.45 the main components of a typical motion controller system are shown. The picture also describes how the different parts interact to make the motion control possible.
The axis configuration windows provided with the NI SoftMotion library can be seen in Figure 3.46. Along with this comes an interactive test panel which is also quite useful to test the motors.

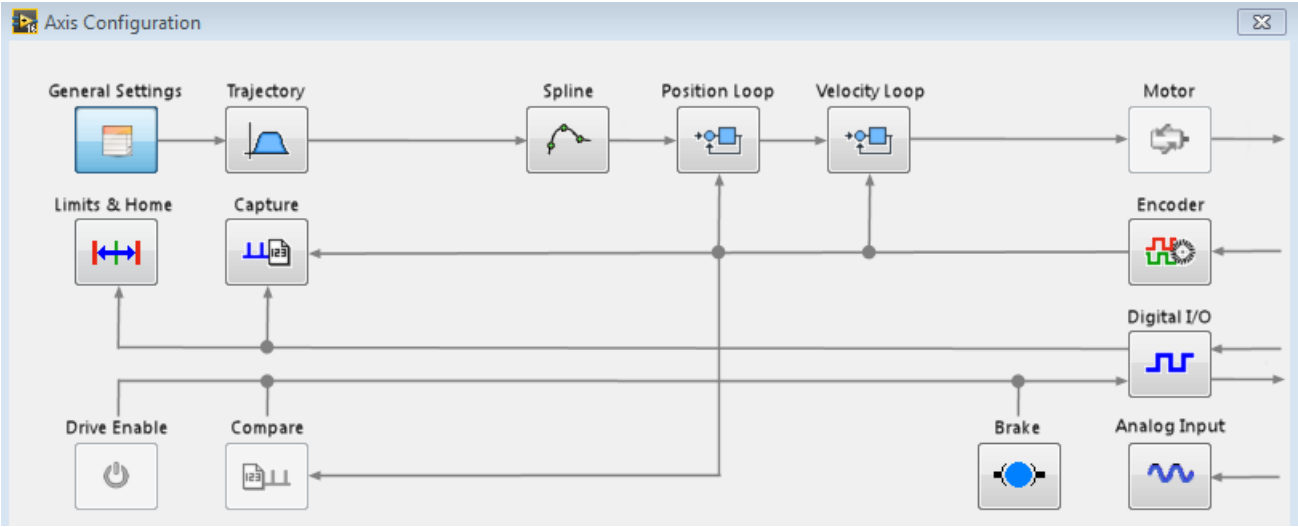


Figure 3.46: Axis configuration window (SoftMotion library).

NI SoftMotion provides properties and methods that provide both low-level as well as high-level access to NI SoftMotion axis data, including configuration information, reading and writing data, and status information.

4 Results and Evaluation

4.1 Mechanical Performance

4.1.1 Robustness

The testbed was designed so that the accuracy should not be affected by external sources. The testbed has a high material thickness on a majority of its components to decrease material flex and thus stochastic deviations in the system. The bolts used for assembly purposes of the testbed were chosen with a high safety factor in mind, which resulted in using bolts with a marginal larger diameter than normal as an effort to decrease the stochastic deviations. As stated in Section 1.5, the testbed should achieve a lateral stiffness of 400 $\frac{N}{\mu m}$ during translational movement which it is estimated to achieve. However this is not a verified result.

4.1.2 Error Induction

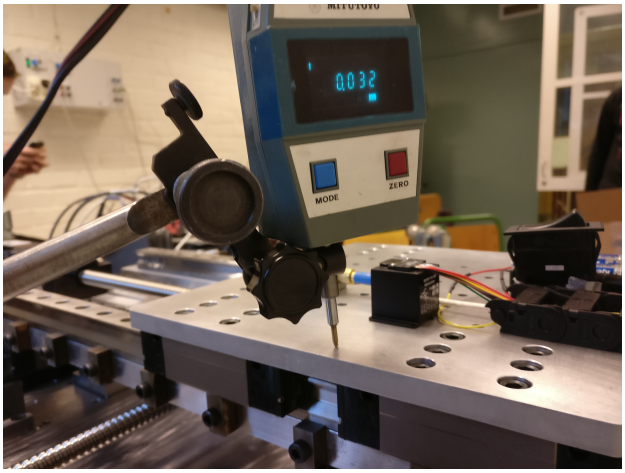


Figure 4.1: Measured geometric error of 32 μm.

As stated in Section 1.5, the induced displacement errors should be greater than 20μm. To verify that this requirement was met an error was induced by deforming one of the rails with the displacer (3.2), which translates an applied torque to a vertical force directed at the rail. The error was measured by using a digital dial indicator capable of micrometer precision placed on the top plate, confirming that a geometric error was successfully applied to the testbed, Figure 4.1.

4.1.3 Accuracy

Without proper measurement equipment or time it's not possible to verify the positioning accuracy nor the dynamic errors to confirm the requirements in Section 1.5.

4.2 Safety

The system is limited in movement by four hardware limit switches (two per axis), which keep users and observers safe in case of unexpected behavior. When a limit switch is triggered, the motors stop immediately making it

impossible to restart the movement in the direction of the triggered limit switch. The testbed has to go to a safe position within the boundaries before a new movement can be executed. This feature is built inside the drivers.

The table also acts as a natural border around the testbed itself and keeps the testbed firmly in place (Section 1.5).

4.3 Software

4.3.1 Host PC

The testbed has an easy-to-use user interface which allows the user to generate tests and move in different trajectories, it also shows position and the status of the system (software requirements in Section 1.5). Making a test requires only to press a button, this automatically makes the move, plots some data for monitoring purposes and logs the data in the real-time computer (CompactRio).

4.3.2 CompactRio

In a real time system it is necessary to verify that the different deterministic loops finishes in time and that the non-deterministic loops are scheduled by the real-time operating system to have enough time to be executed and complete their tasks in order to avoid the starvation phenomena.

It is also essential to check the memory used by the program is below the limits of the hardware used and to prevent memory leaks.

Figure 4.2 shows the CPU usage in the CompactRio when the main program is running. This picture was taken in the middle of a test, when there is more overhead in the CPU due to the data logging of the different sensors.

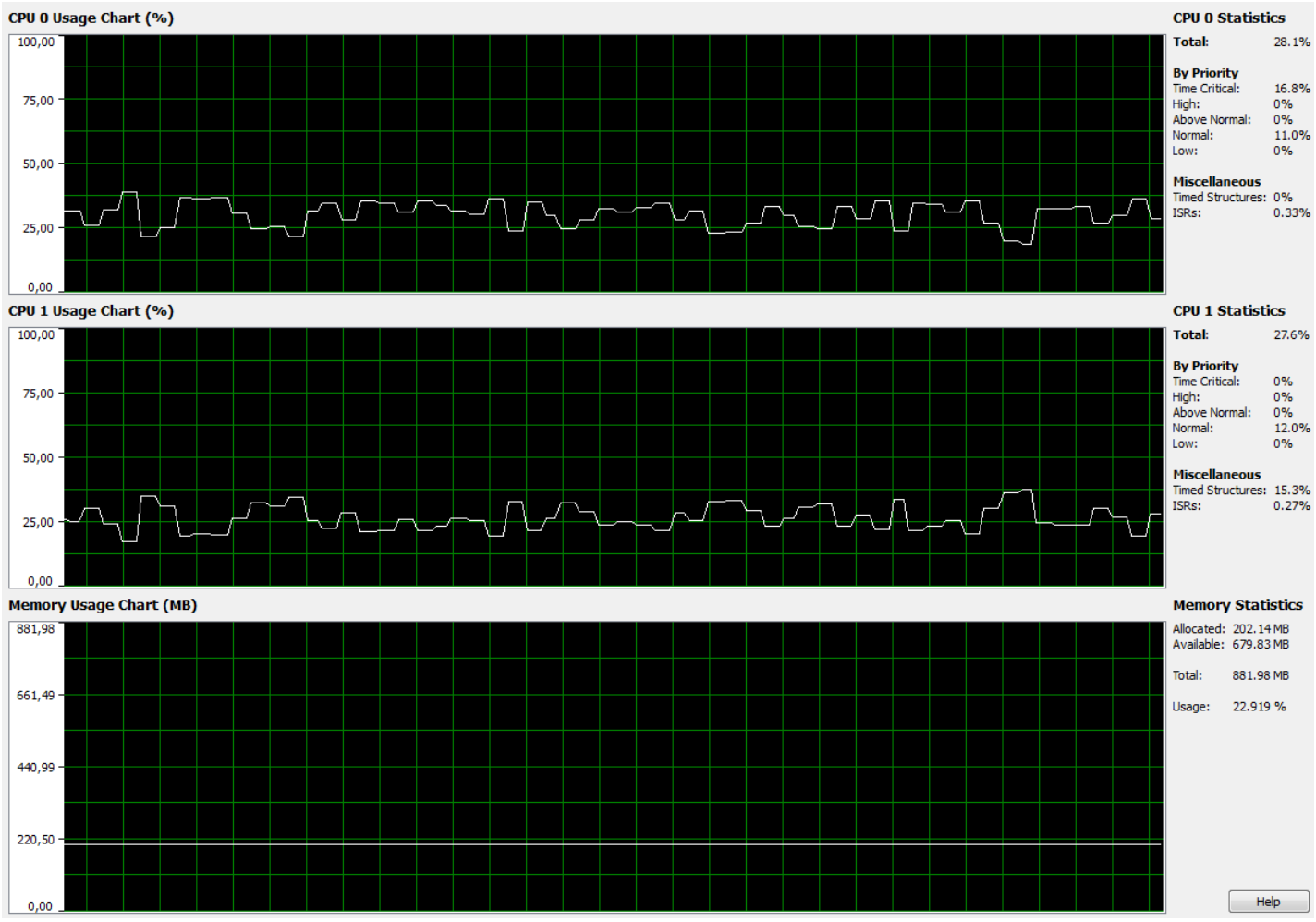


Figure 4.2: CPU Usage in the CompactRio with the main program running.

As it can be seen the CPU usage in both CPUs is below 40 %. This ensures that the non-deterministic loops have time enough to be executed and that the whole program has a good health. It is recommended to keep the CPU usage below 75 % in general to be able to manage eventual CPU usage peaks. This upper-bound leaves still room for the program for future expansions, new functionality or decreasing the rates of the different loops.

When no program is executed the CPU usage in the CompactRio is 15 % because some tasks of the real-time operating system are always running in

the background, so this is the lower limit in the CompactRio with the current configuration. When the main program is running but data is not being logged the CPU usage is below 25 % in both CPUs.

It is also interesting to notice that the deterministic loop (time critical priority) has been allocated to the CPU 0, so it would be possible to have another deterministic loop in the CPU 1.

Regarding the memory usage, it represents a 23 % and this value remains constant. This means that there are not any memory leaks. When no program is executed the memory usage in the CompactRio is about 22 %, so there is still a lot of space available for the main program.

4.3.3 FPGA

The compilation process in the FPGA takes about 10 minutes and when it is done it is possible to see a log and some reports showing the main results.

In Figure 4.3 the final utilization report is shown. The number of slice registers, slice LUTs, block RAMs and DSP48s are displayed. These are resources needed for the FPGA in order to perform logic functions, store bit patterns in registers or store data in memory.

The program running on the FPGA consists only of two loops to read the accelerometers and the gyroscope. As seen in Figure 4.3 the total resources of the FPGA are far from being used.

Device Utilization	Used	Total	Percent
Slice Registers	5986	82000	7,3
Slice LUTs	6313	41000	15,4
Block RAMs	6	135	4,4
DSP48s	1	240	0,4

Figure 4.3: Final device utilization in the FPGA (placement).

The 40MHz clock of the FPGA is also used in this program and this requirement is met as it can be checked in the final timing report of the FPGA (Figure 4.4).

Clocks	Requested (MHz)	Maximum (MHz)
40 MHz Onboard Clock	40,00	Met

Figure 4.4: Final timing in the FPGA (routing).

4.4 Control Performance

To make reproducible measurements for test with fixed spatial frequencies it is important that the target velocity is reached quickly and is kept stable until the deceleration phase. To test how accurate the controller was for each axis a series of tests were made. The conducted tests were linear movements from 0 to 250mm in X-Y direction first each axis separate and later combined. This was done for three different set of speeds 20, 80 and 150mm/s, these were the same as the speeds later used to evaluate the accelerometer and gyroscope data.

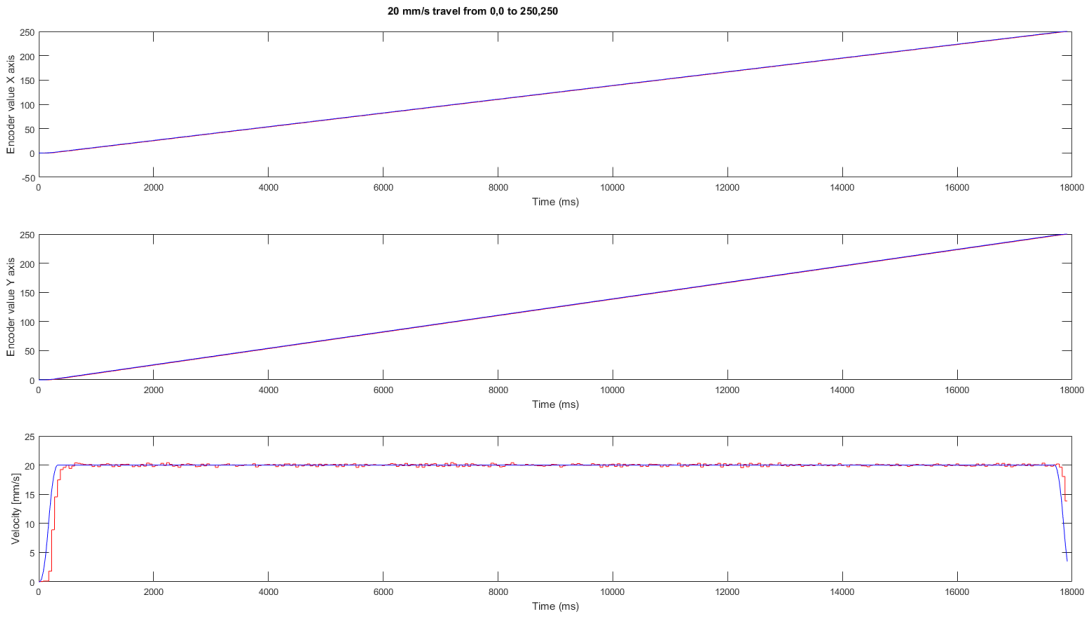


Figure 4.5: Position control plot for constant velocity test at a speed of 20mm/s

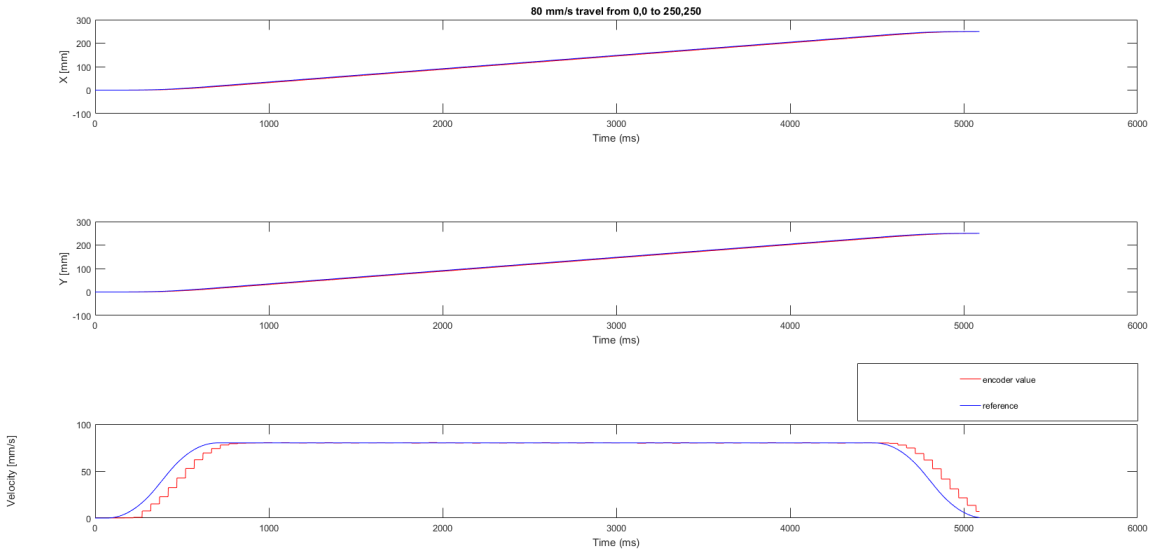


Figure 4.6: Position control plot for constant velocity test at a speed of $80mm/s$

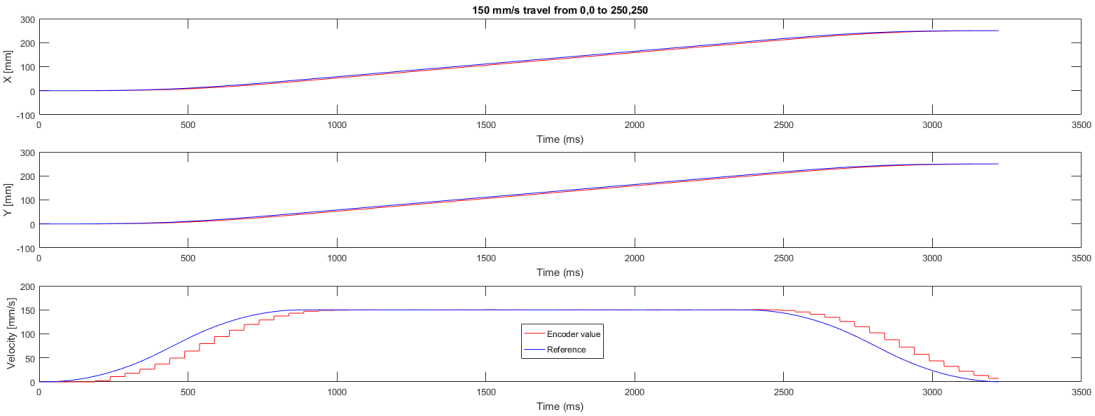


Figure 4.7: Position control plot for constant velocity test at a speed of $150mm/s$

For the measurements in Figure 4.5, 4.6 and 4.4 the only changed variable is the maximum speed. The settings for the PID gains in position, velocity and current loops were unchanged from the factory settings.

4.5 IMU Measurements

The purpose of the testbed is to enable the collection of sensor data and by analysing this evaluate to what degree an induced error in the rails can be detected. This section shows the results from the initial testing of the sensors.

4.5.1 Accelerometer Tests

To test the performance of the accelerometer movements of $250mm$ in the x-axis were made. This was done for slow, normal and high speeds (20, 80, 150 mm/s). The measured voltage was converted to acceleration and then integrated twice to position. The integrated acceleration drifts from the actual value after some time during the measurements, to calibrate for the drift a calibration test was first conducted. Noise was measured for 10 seconds when the accelerometer was in a static position. The mean of the acceleration for this measurement was then subtracted from the other tests.

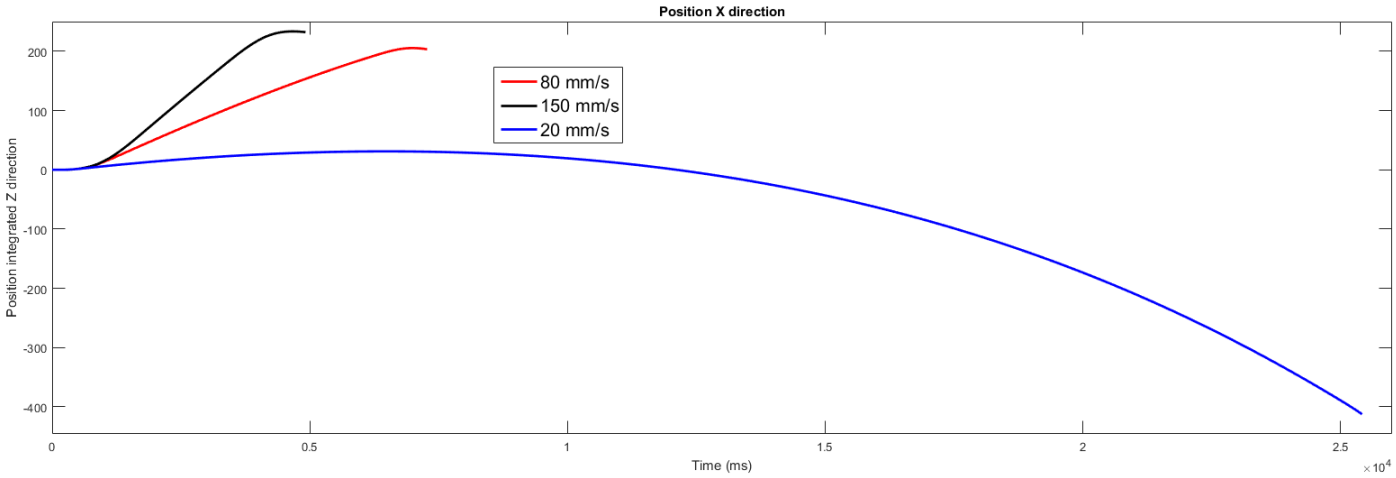


Figure 4.8: Position in x-axis from integrated accelerometer data for three separate tests, 150, 80, 20mm/s respectively.

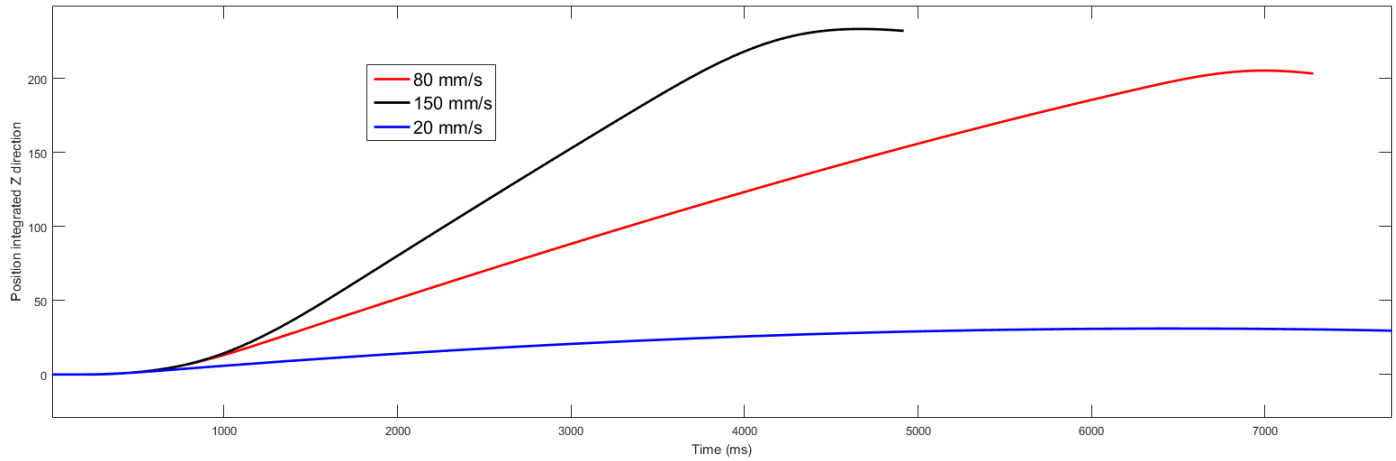


Figure 4.9: Previous figure zoomed

4.5.2 Detection of Induced Errors

As a final test the sensors were tested if they could detect a low frequency error of $20\mu m$ and $40\mu m$ in amplitude. This was done by measuring the difference between sensor data collected from applied error inducement to the same measurement with no applied deformation. The top plate was placed so that with a stroke length of $250mm$ in x direction the sensor would travel over the force inducer. The tests where made with three different speeds (20, 80, 150 mm/s). The theory is that by taking the difference of the sensor data from the test with and without induced error the profile should appear in the signal. 10 measurements for each test were made and then averaged to decrease stochastic errors.

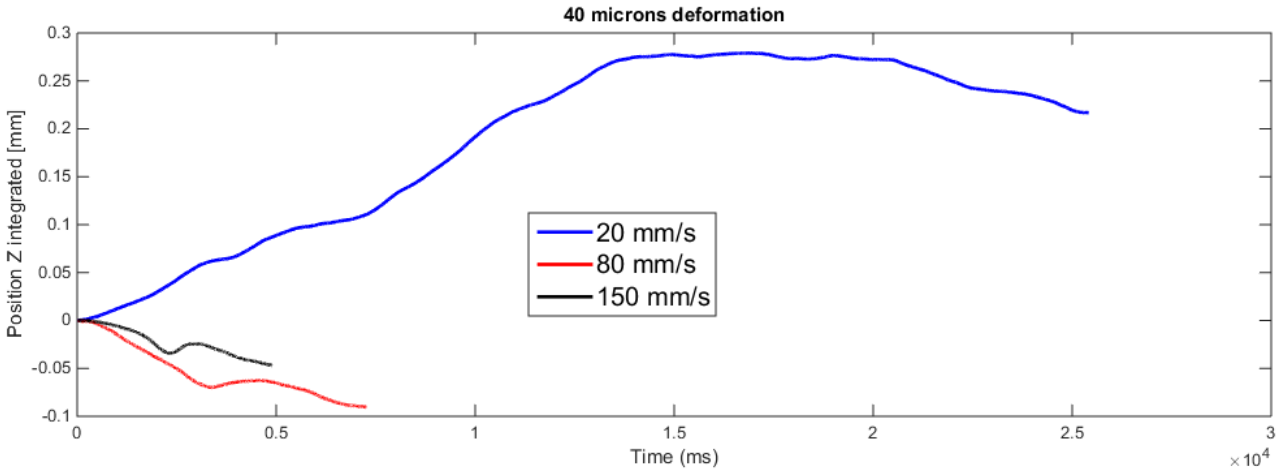


Figure 4.10: Signal difference between no error induced and $40\mu m$ deformation for speeds 20, 80 and 150 mm/s. Data is taken from accelerometer data in z-direction and integrated twice for position. Random walk is apparent for longer measurements.

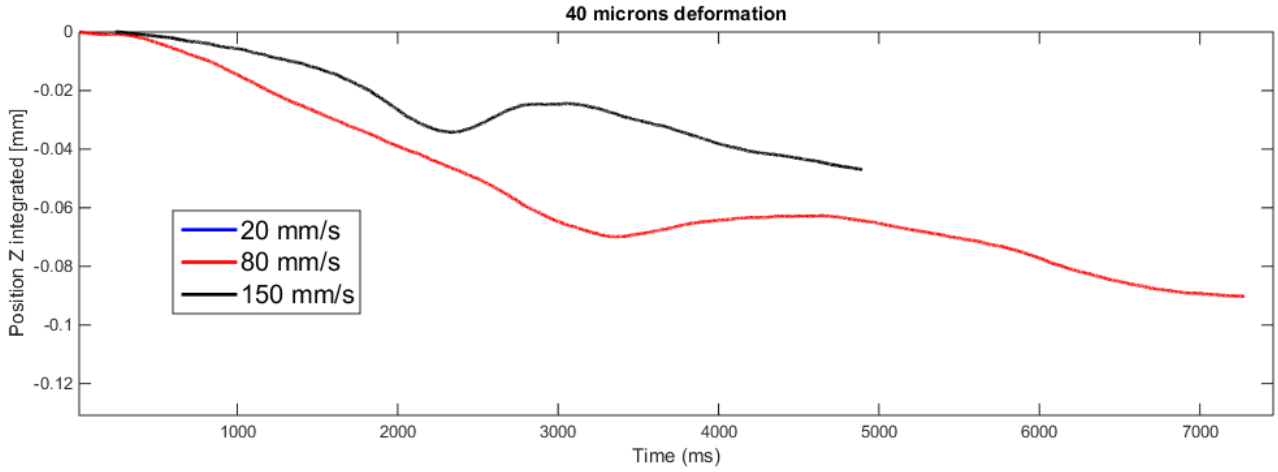


Figure 4.11: Zoomed picture. Some drift is visible, but some signal from the deformation seems to be detected.

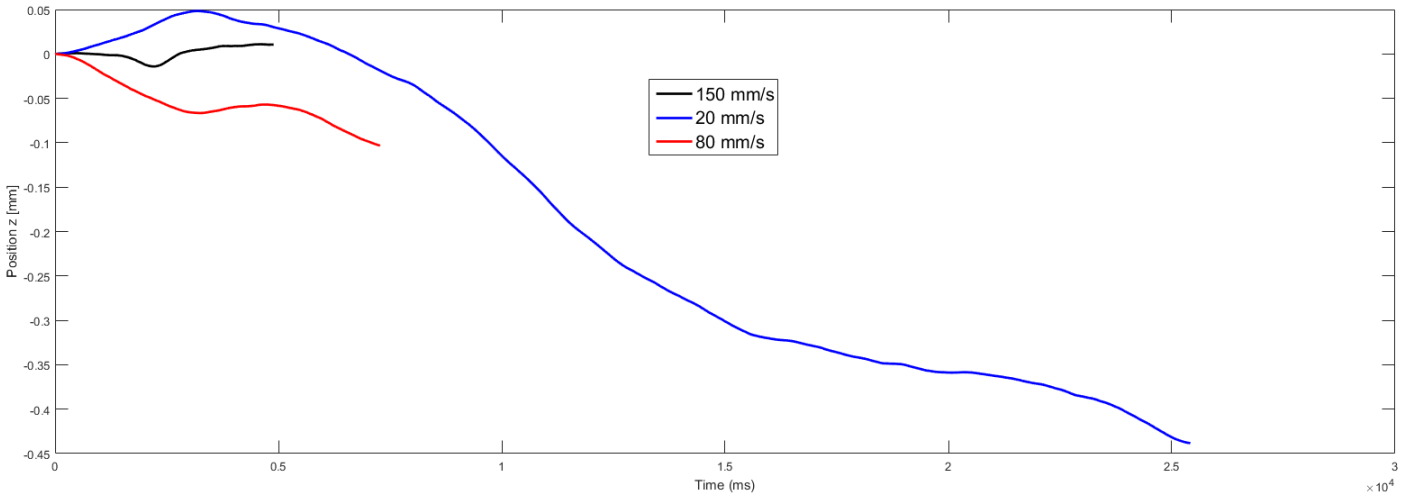


Figure 4.12: Signal difference between no error induced and 20μm deformation for speeds 20, 80, 150 mm/s.

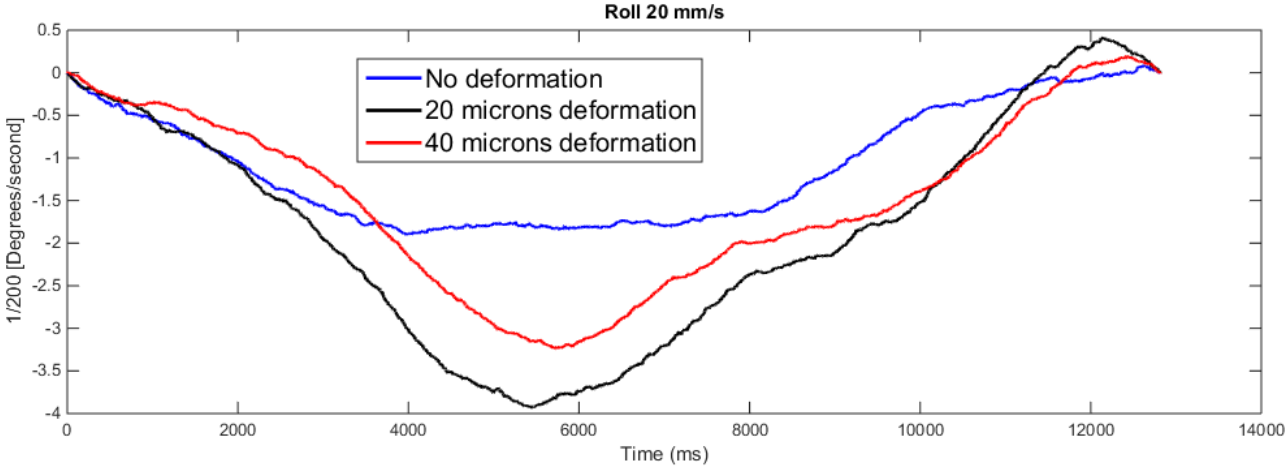


Figure 4.13: Data collected by the gyroscope and integrated once to angle. Measurements at $20mm/s$, with 0, 20 and $40\mu m$ applied deformation.

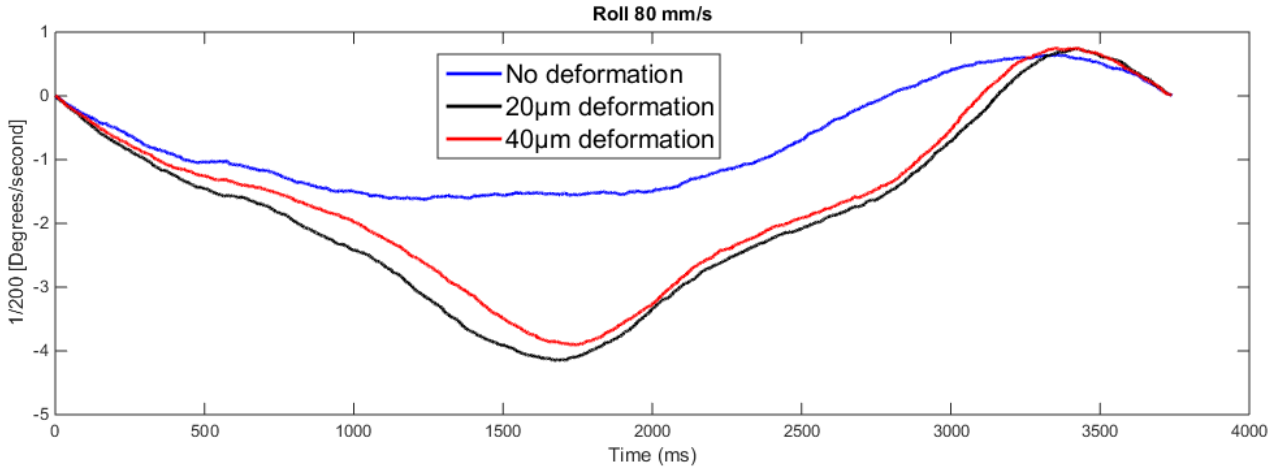


Figure 4.14: Gyroscope data from measurements at $40mm/s$, with 0, 20 and $40\mu m$ applied deformation.

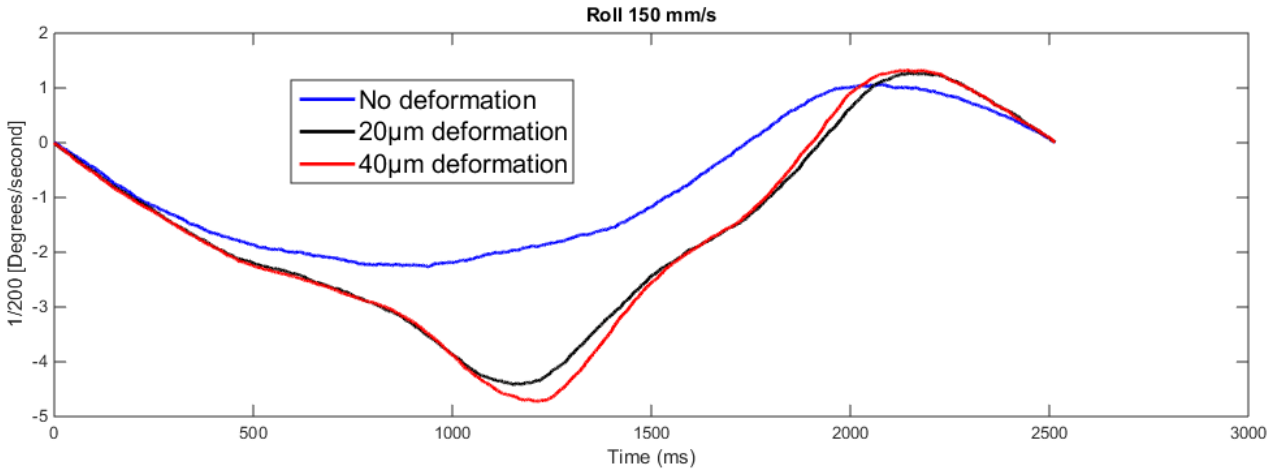


Figure 4.15: Gyroscope data from measurements at $150mm/s$, with 0, 20 and $40\mu m$ applied deformation.

The data extracted from the experiment it shows that it is possible to detect the difference between no deformation and deformation. It is however not possible to discern the difference between $20\mu m$ and $40\mu m$ deformation.

5 Discussion and Conclusion

5.1 Hardware

Whilst is was initially planned to confirm the hardware construction using a laser interferometer but due to the time frame this was not realistic. Therefore the construction remains unverified. However our assumptions would be that the construction does not fulfill the error requirements in its current state. This being as a $5\mu m$ positioning accuracy is incredibly difficult to do and requires hardware, measuring instruments and experience that we did not have access to during the construction.

The errors induced to the test bed in, Section 4.1.2, were measured by a digital dial indicator to confirm that a geometric error was introduced to the system. However due to the uncertainty of the measurement there is not possible to successfully verify the amplitude of the error. To achieve this a laser interferometer is required to receive a complete error profile of the rails to act as a reference to the measurements concluded in, 4.5.2. Regarding the mechanical performance of the test bed, it is estimated to achieve the stiffness requirement in 1.5 but this is not verified.

5.2 Control Performance

When observing the data shown in Figures 4.5, 4.6 and 4.4 it was concluded that it was not necessary to tune the controller in order to test the error inducement with the sensor data. With speeds higher than $150mm/s$ it is recommended to further investigate performance enhancement through tuning the controller as well as increase the maximum acceleration and jerk through software. From Figure 4.4 it can be seen that the reference speed

correspond to the target velocity roughly 50% of the time during the measurement, and the actual velocity is approximately $80ms$ delayed from the reference. The delay in the system is mainly what can be improved with better tuning. No steady state error is seen in the measurement and the controller positions of the workpiece.

The velocity plots have a low sample time of $50ms$. This is due to the fact that the motor driver are set to position control which reduces the sample rate of current velocity messages to the CompactRio. The values in between the data points can be disregarded (the motor velocity increases continuously).

5.3 Accelerometer Data

In the experiment shown in Figure 4.8 the accelerometer was calibrated before the measurements. The calibration data was measured over a 10 second period with the accelerometer kept at a static position. The average of the calibration measurement was then subtracted to each axis to reduce bias in the signal. Due to random walk the bias changes over time so that the calibrated compensation is no longer accurate for data collected for sequential measurements. In Figure 4.8 and 4.5.1 it can be seen that the accelerometer perceive that it is constantly accelerating in -x direction. Since the signal is integrated twice a small bias gives large impact for long measurements.

5.4 Error Detection Tests

In Figure 4.8 it can be seen that the signal from the accelerometer drifts with time, this impacts the results more the longer the measurements take. This drift is due to the calibration made from the calibration measurements was no longer valid. Not enough tests on the accelerometer were made to conclude how often the sensor need to be re-calibrated. The different tests seen in Figure 4.8 where conducted within minutes from the calibration data was collected, for long measurements with small signal amplitude some posterior work is required or the measurements will disappear in sensor noise. For shorter measurements the calibration is of less importance as the error propagates with time, making the resulting signal more clear.

Figure 4.5.2 and 4.5.2 shows the results from 10 tests that were made for each speed and deformation and averaged to cancel out noise. When calibrating the acceleration in z direction it was concluded that the calibration data was not valid due to the random walk making the calibration measurements invalid, rendering the actual sensor input from the test undetectable. To counter this, the z-axis was calibrated with the data from the same measurements. In theory it is correct to assume that the mean of the acceleration in z direction of an entire run from start to stop should be 0. Based on this theory 10 measurements were averaged and the mean acceleration of these were subtracted from the subsequent test results to reduce the influence of sensor bias.

Four sets of data were taken during the error inducement, one with no error, one with $20\mu m$ deformation and two with $40\mu m$ deformation. The first data set with $40\mu m$ deformation did not show any deviation from the data gathered without the induced error, neither in the accelerometer nor the gyroscope measurements. This data set was excluded from the results since it was concluded that the applied deformation was not accurate.

Since the laser interferometer was not used to measure the actual profile, the presence of a deformation can only be seen by measuring the difference in sensor output between tests with a non deformed and deformed rail, and by repeating this to see that the results are consistent. Little can be said about how well the data corresponds to the actual deformation as the profile is unknown. However, the results suggests that the gyroscope can detect low frequency error even at low speeds, whilst the accelerometer has too much drift to reliably detect the low frequency errors needed to fulfill the requirements. The laser interferometer would conclude with certainty if the actual profile corresponds to the read value, during this experiment the only reference that is obtained is the measured peak value of the deformation, this with an unknown certainty and tolerance. When reading the deformation applied before the tests with the digital dial indicator it is not sure that the actual peak value of the deformation is measured. This could be partly explain why the difference between the $20\mu m$ and $40\mu m$ deformations cannot be detected; it is possible that the deformations applied were in fact very similar. The only thing that can be concluded with certainty is that there is an error detectable by the sensors when it is induced.

Looking through the data plots from the gyroscope it can be observed that the profile between different speeds are rather consistent. This would decrease the probability of false positives when detecting a change due to the deformation applied. The data shows promise that the low frequency deformation can be detected with the gyro and the accelerometer, more measurements are required to further investigate the use of IMU applications in detection of degradation in CNC machines.

6 Future Work

6.1 Mechanical Hardware

- Induce high frequency errors in the rails. In this project only a very low spatial frequency error was induced but it was always the intention to make repeatable high frequency errors, to detect other forms of deterioration like the surface deterioration in the contact area between rails and slider etc.
- Verifying the positioning accuracy of the system using a laser interferometer or equally effective methods.
- If positioning accuracy does not fulfill requirements then:
  - Analyze play in ball screws and sliders. They may have to be exchanged for higher precision and preloaded setups.
  - Analyze the rails' alignment. If they are off alignment then the reference surfaces may have to be cut and the alignment redone with proper equipment.
  - Analyze the planarity of the different surfaces with proper equipment and compensate errors.
- Design a mount for the cable guide so that it is possible to attach it to the base plate.
- Apply rust-preventative agent to the unprotected metal surfaces.
- In the construction described the test bed cannot utilize its full travel length and does therefore not fulfill the requirements of 300mm travel range. There does however exist possibilities to shave off material from

the ball screw connector to allow for a travel range that does fulfill the requirement.

- Due to a mismatch between the CAD model of the ball screw and the actual ball screw the lower level suffers from a subpar match between one of the bearing and its bearing house. The solution is to re-order a ball screw that fits or accommodate the longer ball screw with a different construction.
- Multiple limit switch holders are made from plastic and are both very unstable and fragile. A solution to this is to remake them in a more durable and less flexible material.
- Add threads to the top plate's holes to simplify the addition and removal of sensors and other tools.
- Design a container for the sensors to robustly fasten them to minimize vibrations and other disturbances.

6.2 Electrical Hardware and Control System

- Tune the control parameters. It is recommended to use Kollmorgen WorkBench which is the software provided by the motor and motor drive manufacturer. This is installed but not prepared due to lack of time. In order to use the auto tuning (Performance Servo Tuner) safety precautions must be taken, mainly configuring the limit switches, maximum speeds, positioning, etc.
- The limit switches have a high output on triggering. This means that if the limit switches are disconnected there is no way to notice this as the system will only indicate a lack of collision. Inverting this signal to have low signal on trigger will allow the system to notice disconnections.
- The software has an implemented homing functionality that is intended to be used for repeatable measurements. It has been tested using the reverse limit switches as temporary homing switches but this is not recommended, the limit switches should not be used in any other case than system failure. The homing function is that the system travels

backwards until the home switch triggers (or a limit switch, in that case it changes direction) then moves back to make a controlled positioning.

- Analysis of filters in hardware and software. The gyro has a built in low pass filter, the performance of this has not been verified. Suitable filters in software have neither been investigated but should be implemented to increase the performance of the collected data.

- The gyroscope is currently powered by batteries, a better solution would be to run a cable along the cable guides and get power from the mains instead. The gyroscope requires 3.3-5.1V DC power [Technologies, 2014], which is not found anywhere else in the setup, why either a separate power adapter or a voltage divider needs to be purchased and installed.

- During the construction the switch to the main power supply have been used as a make shift emergency stop button. This has been sufficient during the development but it should be replaced with a proper easily accessible emergency stop button before it is used by people outside of the development team.

- At the moment the drives are connected to the CompactRio through EtherCAT since the drives have a SoftMotion interface for its operation but this does not allow to modify all the features inside the drives, like the control loop structure (in order to test other control strategies) or other low-level details. In addition, using the EtherCAT communication standard requires to use the Scan Engine in the CompactRio, which has a maximum period of 1 ms. It would be interesting to investigate the possibilities given by the NI 9502, NI 9503 and NI 9505 C series drive modules to operate and control different motors. Another option would be to use the NI 951X drive interface module hardware. The advantage of using these C series modules is that the control loop could be implemented in the FPGA, so the FPGA interface mode could be used instead of the Scan Engine. This solution would increase the flexibility of the system since it would be possible to access to almost all the signals inside the drives and it would allow for the internal logic to be modified. Besides, the FPGA has plenty of resources which are not fully used at the moment.

6.3 Software

- User commands outside the boundary of the testbed should not be run. This should be implemented right away, all data is in variables in LabVIEW, only simple arithmetic's are needed.

- The system relies as of now for the user to make each command through the user interface and moves can not be queued. In the future it is recommended that the system should be able to run moves through a text file with G-code or similar syntax to make more complicated tests with sequential moves repeatable.

- The GUI displays all the monitored values every 50 ms but it would be better to check which of those values have been updated since last time to avoid displaying old data.

- Some messages can be lost in the gyroscope since it is a digital sensor. It is recommended to implement an indicator that shows the number of lost messages. This is not implemented at the moment but the message count of every gyro message is being logged so that the number of lost messages could be checked later.

- The CompactRio works in two different modes; configuration mode and active mode. The configuration mode is used to deploy VIs, variables and settings to the RT computer whereas the active mode is necessary to run the Scan Engine and so on to manage the EtherCAT communication with the drives. At the moment the configuration mode is switched to the active mode using the NI Distributed System Manager but it would be better to perform this programmatically at the start of the program.

- When the program is run the first time after the CompactRio has been initialized there is a problem with the network-public shared variables because their values do not propagate properly around the system. This is just a warning and the program works without problems when the run button is pressed again but it would be good to find a fix.

- Although CPU and memory usage are not a problem, some software optimization that could be investigated are:

- Disable publishing options for network-public shared variables.
- Write data to file at once instead of line to line. Writing data to files is thought to take a considerable amount of CPU usage.
- Uninstall not used features in the CompactRio.
- Compare the performance of the synchronous and asynchronous API of the SoftMotion library to see if it is more efficient to change the control loop architecture of the main VI in the CompactRio to use the API in a synchronous way.

Bibliography

[IGUS, 2009] IGUS (2009). *E2 micro Series E2.10*. IGUS.

[Instruments, 2015a] Instruments, D. (2015a). *OUTLI NE/I NSTALLATION DWG, TRIAXIAL DC ACCELEROMETER, 7503D SERIES*. Dytran Instruments.

[Instruments, 2015b] Instruments, N. (2015b). *Specifications NI cRIO-9030 - Embedded CompactRIO Controller with Real-Time Processor and Reconfigurable FPGA*. National Instruments.

[ISO 5725:1994(E), 1994] ISO 5725:1994(E) (1994). Accuracy (trueness and precision) of measurement methods and results. Standard, International Organization for Standardization, Geneva, CH.

[JAKOB, 2016] JAKOB (2016). *Servo couplings catalogue*. JAKOB.

[Lamb, 2016] Lamb, F. (2016). *Industrial Automation: Hands-On*. McGraw-Hill Education.

[NI, 2015] NI (2015). Ni scan engine performance benchmarks. [Online; accessed 20-December-2016].

[NSK, 2006] NSK (2006). *NSK Ball Screws for Standard Stock*. NSK.

[NSK, 2009] NSK (2009). *NSK Linear Guides*. NSK.

[NSK, 2016] NSK (2016). *NSK Precision Machine Components*. NSK.

[Technologies, 2014] Technologies, G. (2014). *G200D Digital Triaxial Gyro Datasheet*. Gladiator Technologies.

[THK, 2016] THK (2016). *Ball Screw: General Catalog*. THK.

[Uddeholm, 2016] Uddeholm (2016). *Alumec 89 catalogue*. Uddeholm.

[Vogl et al., 2015] Vogl, G. W., Weiss, B. A., and Donmez, M. A. (2015). A sensor-based method for diagnostics of machine tool linear axes. *NA*.

A Bill of Materials

Bill of Materials Test bed

Type	Part definition	quantity	Comment
Mechanical components	Table	1	Siegmund
	Baseplate	1	Custom made
	Rail spacer	2	Custom made
	Rail spacer alignment	2	Custom made
	Rail-R1A25-LCN/M	4	Acumo
	Slider-RAA25ALK6Z	8	Acumo
	Ballscrew-PSS1505N1D0561	2	Acumo
	Bearinghouse-WBK12-01A	2	Acumo
	Bearinghouse-WBK12S-01	2	Acumo
	Bellow coupling MKM12 14G6/12G6	2	Acumo
	Lower motor assembly	1	Custom made
	Lower limit switch assembly	2	Custom made
	Spacer WBK12-01A	2	Custom made
	Spacer WBK12S-01	2	Custom made
	Alignment clamp	32	Custom made
	Lower ballscrew fastener	1	Custom made
	Middleplate	1	Custom made
	Upper motor assembly	1	Custom made
	Upper limit switch assembly	2	Custom made
	Upper ballscew fastener	1	Custom made
	Topplate	1	Custom made
	Bolts	304	Sifvert Skruv
Electrical components	CompactRIO cRIO-9035	1	National Instruments Part number: 783848-01
	Serial Interface Module NI 9871	1	National Instruments Part number: 779892-02
	AC/DC Analog Input module NI 9230	1	National Instruments Part number: 783824-01
	AKMD24D Brushless Servo Motors	2	National Instruments Part number: 781542-01
	EtherCAT AKD Servo Drives	2	National Instruments Part number: 781521-01
	NI PS-10 Desktop DC Power Supply	1	National Instruments Part number:782698-01
	Inductive Limit switch	4	Ordered part
	Accelerometer 7503D1	1	Dytran Instruments
	G200D Digital Triaxial Gyro	1	Gladiator Technologies
Software components	LabVIEW 2016	1	National Instruments

B Finite Element Analysis

B.1 Middle Plate 500N X-axis

Middle plate 500N side

Company
KTH

Author
I am µ

Date
19 October 2016

Software Used
Solid Edge ST(108.00.00.091 x64)
Femap (11.1.2)

Solver Used
NX Nastran (9)

Table of Contents

---

1. Introduction

2. Model Information

Document	Middle model.asm

3. Study Properties

Study Property	Value
Study name	Static Study 1
Study Type	Linear Static
Mesh Type	Tetrahedral
Iterative Solver	On
NX Nastran Geometry Check	On
NX Nastran command line	
NX Nastran study options	
NX Nastran generated options	
NX Nastran default options	
Surface results only option	On

4. Study Geometry

4.1 Solids

Solid Name	Material	Mass	Volume	Weight
sideplate_fastener.par:1	Stainless Steel, 420 (Occurrence)	0.030 kg	3887.618 mm^3	295.265 mN
subplate.par:1	Stainless Steel, 420 (Occurrence)	10.720 kg	1383204.931 mm^3	105054.414 mN
Rail-R1A250600LCNP6Z-15-15-NSK.par:1	Stainless Steel, 420 (Occurrence)	1.835 kg	236838.041 mm^3	17987.849 mN
Rail-R1A250600LCNP6Z-15-15-NSK.par:2	Stainless Steel, 420 (Occurrence)	1.835 kg	236838.041 mm^3	17987.849 mN
Nut-PSS1505-.par:1	Stainless Steel, 420 (Occurrence)	0.161 kg	20766.447 mm^3	1577.212 mN

ballscrew_fastener_ver2.par:1	Stainless Steel, 420 (Occurrence)	0.590 kg	76127.677 mm^3	5781.897 mN
sideplate_fastener.par:2	Stainless Steel, 420 (Occurrence)	0.030 kg	3887.618 mm^3	295.265 mN

5. Material Properties

5.1 Stainless Steel, 420

Property	Value
Density	7750.000 kg/m^3
Coef. of Thermal Exp.	0.0000 /C
Thermal Conductivity	0.025 kW/m-C
Specific Heat	502.000 J/kg-C
Modulus of Elasticity	199947.953 MegaPa
Poisson's Ratio	0.290
Yield Stress	344.738 MegaPa
Ultimate Stress	655.002 MegaPa
Elongation %	0.000

6. Override Properties

7. Loads

Load Name	Load Type	Load Value	Load Distribution	Load Direction	Load Direction Option
Force 1	Force	5e+005 mN	Per Entity	(-1.00, 0.00, 0.00)	Along a vector

8. Constraints

Constraint Name	Constraint Type	Degrees of Freedom
Fixed 1	Fixed	FREE DOF: None

Fixed 2	Fixed	FREE DOF: None

9. Connector

Connector Name	Connector Type	Search Distance	Minimum Search Distance	Coefficient Of Static Friction	Penalty Value
Connector 1	Glue	0.36 mm			100.00
Connector 2	Glue	0.36 mm			100.00
Connector 3	Glue	0.36 mm			100.00
Connector 4	Glue	0.36 mm			100.00
Connector 5	Glue	0.36 mm			100.00
Connector 6	Glue	0.36 mm			100.00
Connector 7	Glue	0.36 mm			100.00
Connector 8	Glue	0.36 mm			100.00

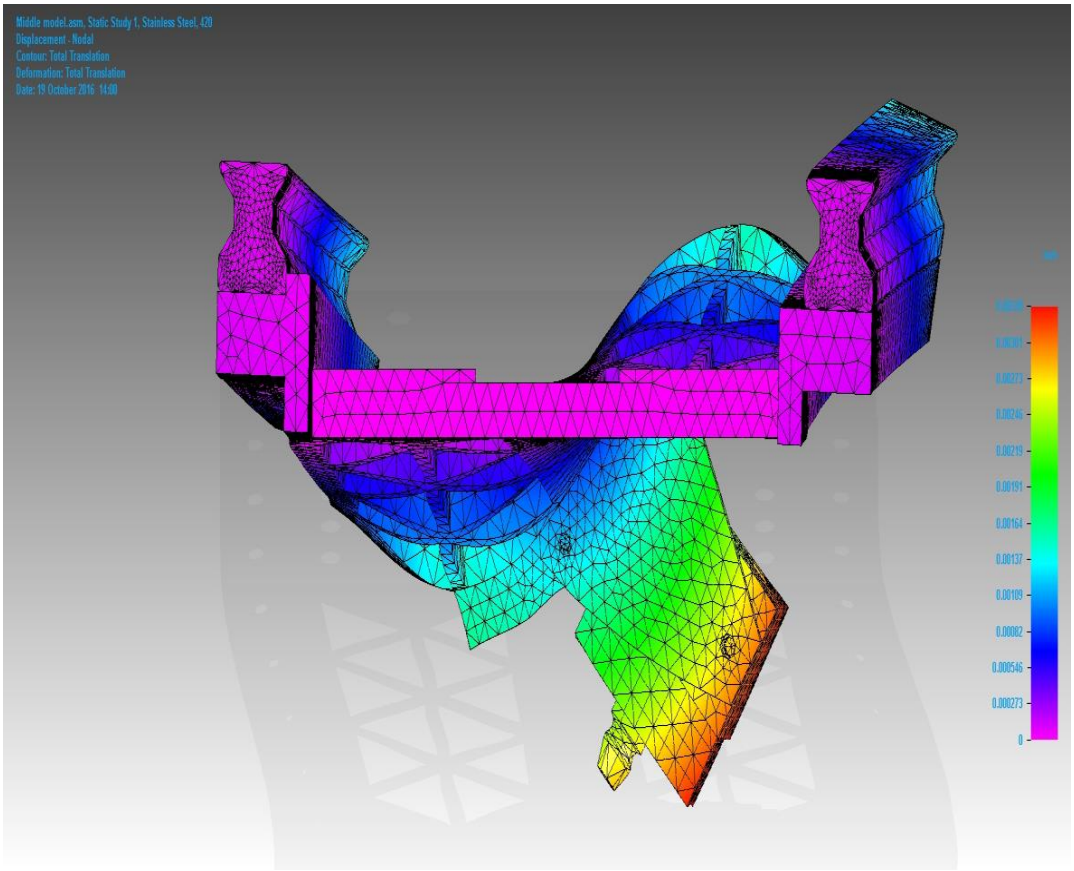
10. Mesh Information

Mesh type	Tetrahedral
Total number of bodies meshed	7
Total number of elements	202,950
Total number of nodes	333,240
Subjective mesh size (1-10)	2

11.Results

11.1 Displacement Results

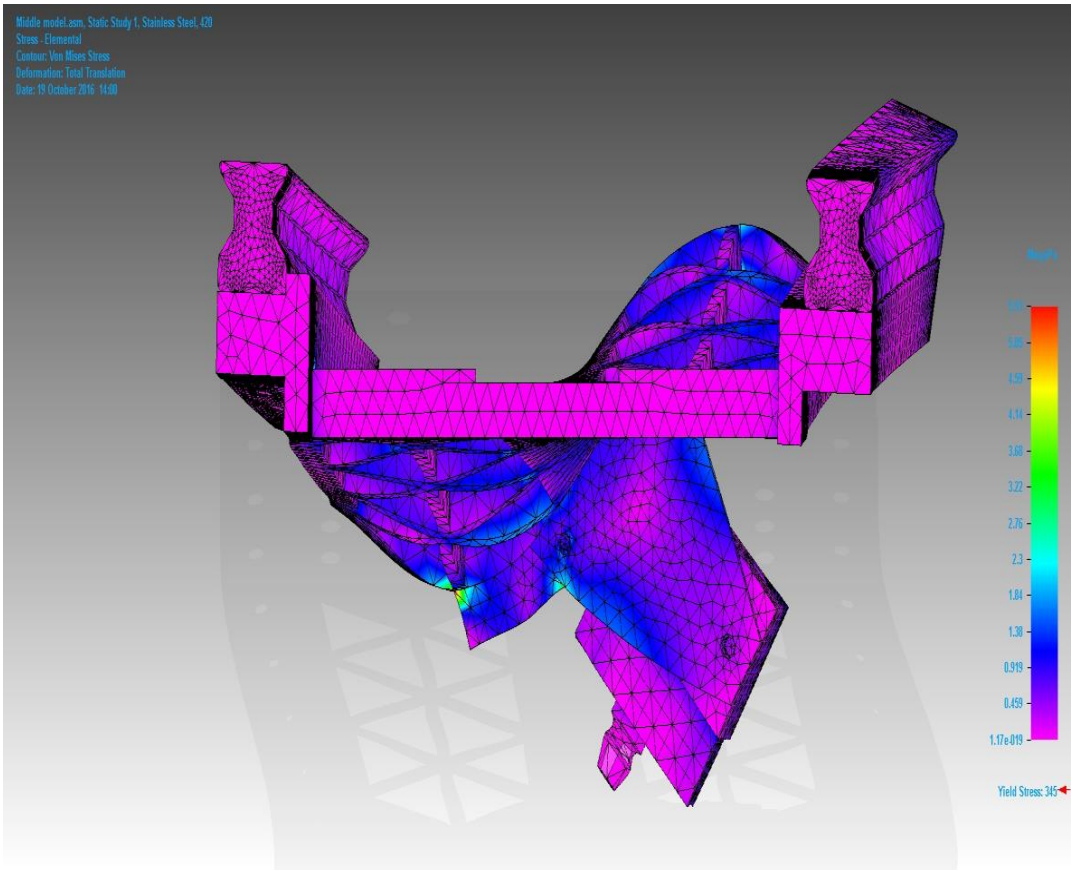
Result component: Total Translation				
Extent	**Value**	**X**	**Y**	**Z**
Minimum	0 mm	-40.475 mm	548.475 mm	-0.000 mm
Maximum	0.00328 mm	21.000 mm	339.998 mm	-43.000 mm



Total Translation

11.2 Stress Results

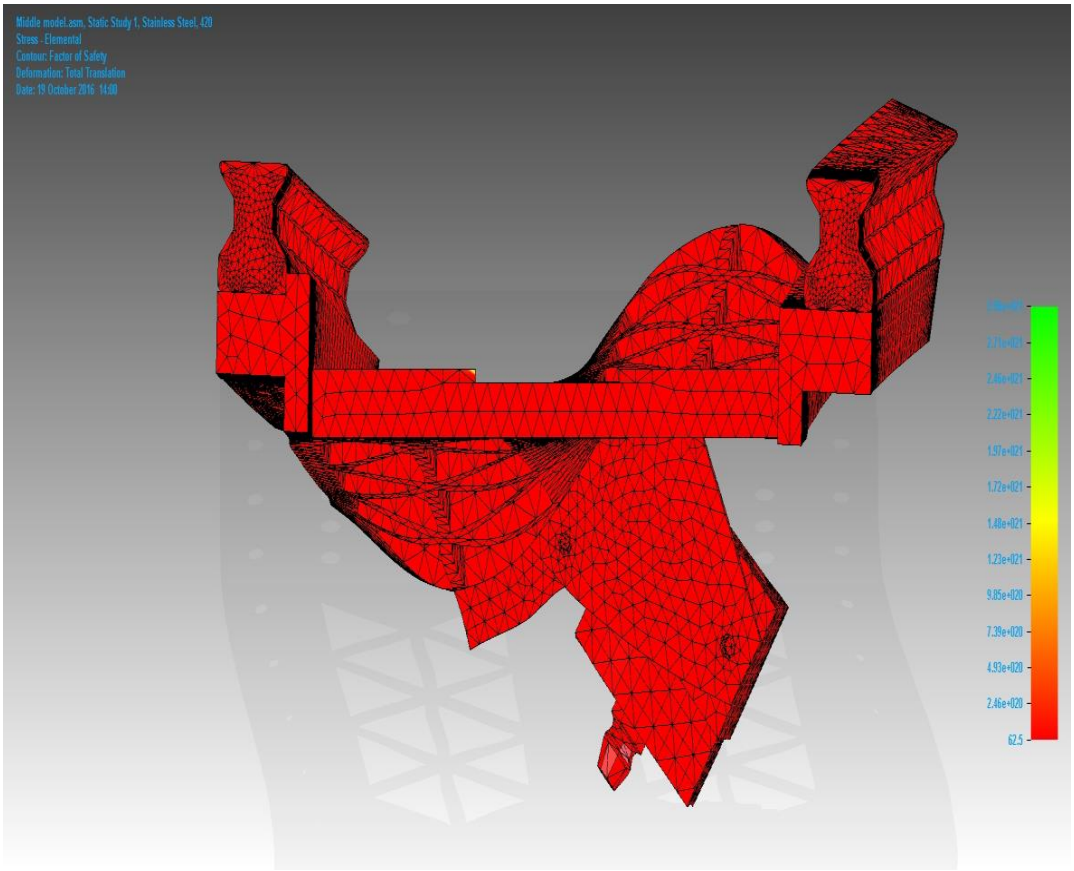
Result component: Von Mises				
Extent	**Value**	**X**	**Y**	**Z**
Minimum	1.17e-019 MegaPa	24.200 mm	563.500 mm	12.500 mm
Maximum	5.51 MegaPa	48.500 mm	360.000 mm	0.000 mm



Von Mises

11.3 Factor of Safety Results

Result Component: Factor of Safety				
Extent	**Value**	**X**	**Y**	**Z**
Minimum	62.5	48.500 mm	360.000 mm	0.000 mm
Maximum	2.96e+021	24.200 mm	563.500 mm	12.500 mm



Factor of Safety

---

12. Optimizations

---

13. Conclusion

---

14. Disclaimer

Important Information

This report should not be used to solely judge a design idea's suitability to a given set of environmental conditions. Siemens makes every effort to ensure that its products provide as much guidance and help as possible. However this does not replace good engineering judgment, which is always the responsibility of our users. A qualitative approach to engineering should ensure that the results of this evaluation are evaluated in conjunction with the practical experience of design engineers and analysts, and ultimately experimental test data. The results contained within this report are believed to be reliable but should not be construed as providing any sort of warranty for fitness of purpose.

B.2 Middle Plate 2x30 Z-axis

Middle plate 2x30N side

Company
KTH

Author
I am µ

Date
19 October 2016

Software Used
Solid Edge ST(108.00.00.091 x64)
Femap (11.1.2)

Solver Used
NX Nastran (9)

Table of Contents

---

1. Introduction

2. Model Information

Document	Middle model.asm

3. Study Properties

Study Property	Value
Study name	Static Study 2
Study Type	Linear Static
Mesh Type	Tetrahedral
Iterative Solver	On
NX Nastran Geometry Check	On
NX Nastran command line	
NX Nastran study options	
NX Nastran generated options	
NX Nastran default options	
Surface results only option	On

4. Study Geometry

4.1 Solids

Solid Name	Material	Mass	Volume	Weight
Slider-RAA25AL3-K-NSK.par:1	Stainless Steel, 420 (Occurrence)	0.561 kg	72363.239 mm^3	5495.988 mN
Rail-R1A250600LCNP6Z-15-15-NSK.par:1	Stainless Steel, 420 (Occurrence)	1.835 kg	236838.041 mm^3	17987.849 mN
subplate.par:1	Stainless Steel, 420 (Occurrence)	10.720 kg	1383204.931 mm^3	105054.414 mN
Rail-R1A250600LCNP6Z-15-15-NSK.par:2	Stainless Steel, 420 (Occurrence)	1.835 kg	236838.041 mm^3	17987.849 mN
Slider-RAA25AL3-K-NSK.par:1	Stainless Steel, 420 (Occurrence)	0.561 kg	72363.239 mm^3	5495.988 mN
Slider-RAA25AL3-K-NSK.par:1	Stainless Steel, 420 (Occurrence)	0.561 kg	72363.239 mm^3	5495.988 mN
Slider-RAA25AL3-K-NSK.par:1	Stainless Steel, 420 (Occurrence)	0.561 kg	72363.239 mm^3	5495.988 mN

5. Material Properties

5.1 Stainless Steel, 420

Property	Value
Density	7750.000 kg/m^3
Coef. of Thermal Exp.	0.0000 /C
Thermal Conductivity	0.025 kW/m-C
Specific Heat	502.000 J/kg-C
Modulus of Elasticity	199947.953 MegaPa
Poisson's Ratio	0.290
Yield Stress	344.738 MegaPa
Ultimate Stress	655.002 MegaPa
Elongation %	0.000

6. Override Properties

7. Loads

Load Name	Load Type	Load Value	Load Distribution	Load Direction	Load Direction Option
Force 1	Force	30 mN	Per Entity	(0.00, 0.00, -1.00)	Along a vector
Force 2	Force	30 mN	Per Entity	(0.00, 0.00, -1.00)	Along a vector

8. Constraints

Constraint Name	Constraint Type	Degrees of Freedom
Fixed 1	Fixed	FREE DOF: None

9. Connector

Connector Name	Connector Type	Search Distance	Minimum Search Distance	Coefficient Of Static Friction	Penalty Value
Connector 1	Glue	0.36 mm			100.00
Connector 2	Glue	0.36 mm			100.00
Connector 3	Glue	0.36 mm			100.00
Connector 4	Glue	0.36 mm			100.00
Connector 5	Glue	0.36 mm			100.00
Connector 6	Glue	0.36 mm			100.00

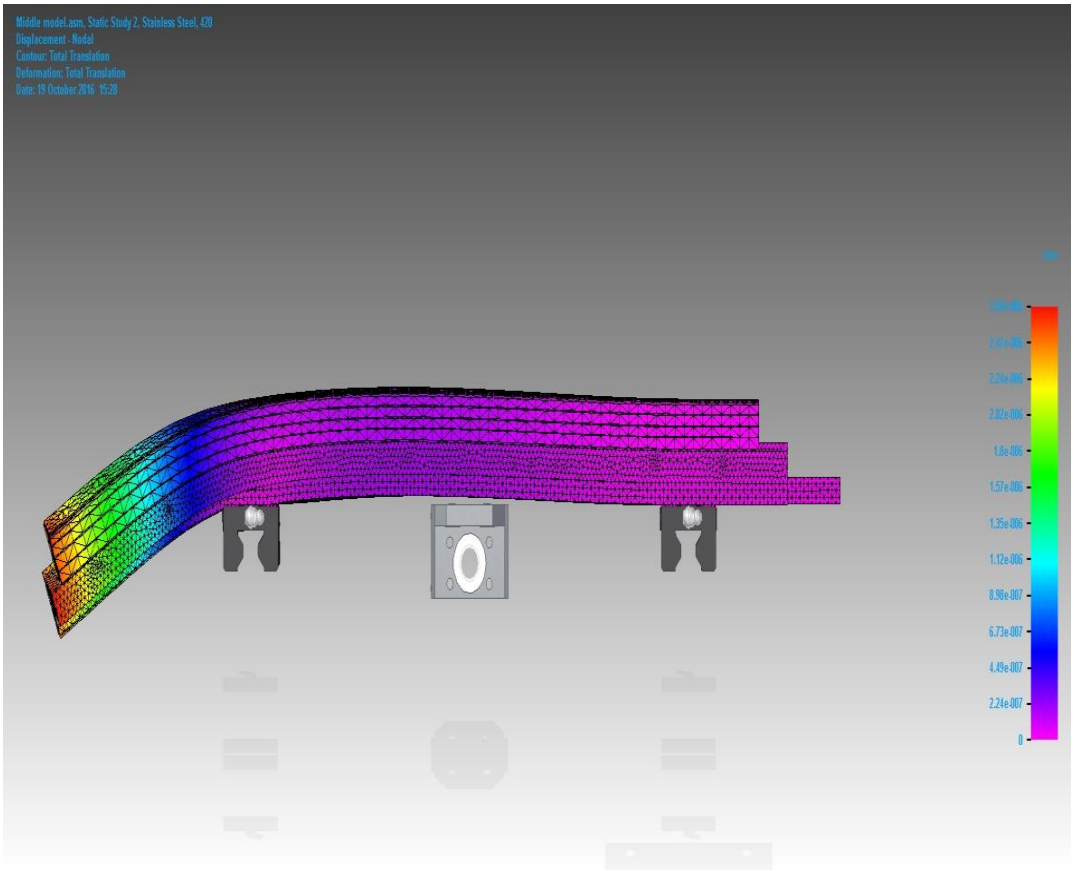
10. Mesh Information

Mesh type	Tetrahedral
Total number of bodies meshed	7
Total number of elements	440,394
Total number of nodes	696,780
Subjective mesh size (1-10)	3

11.Results

11.1 Displacement Results

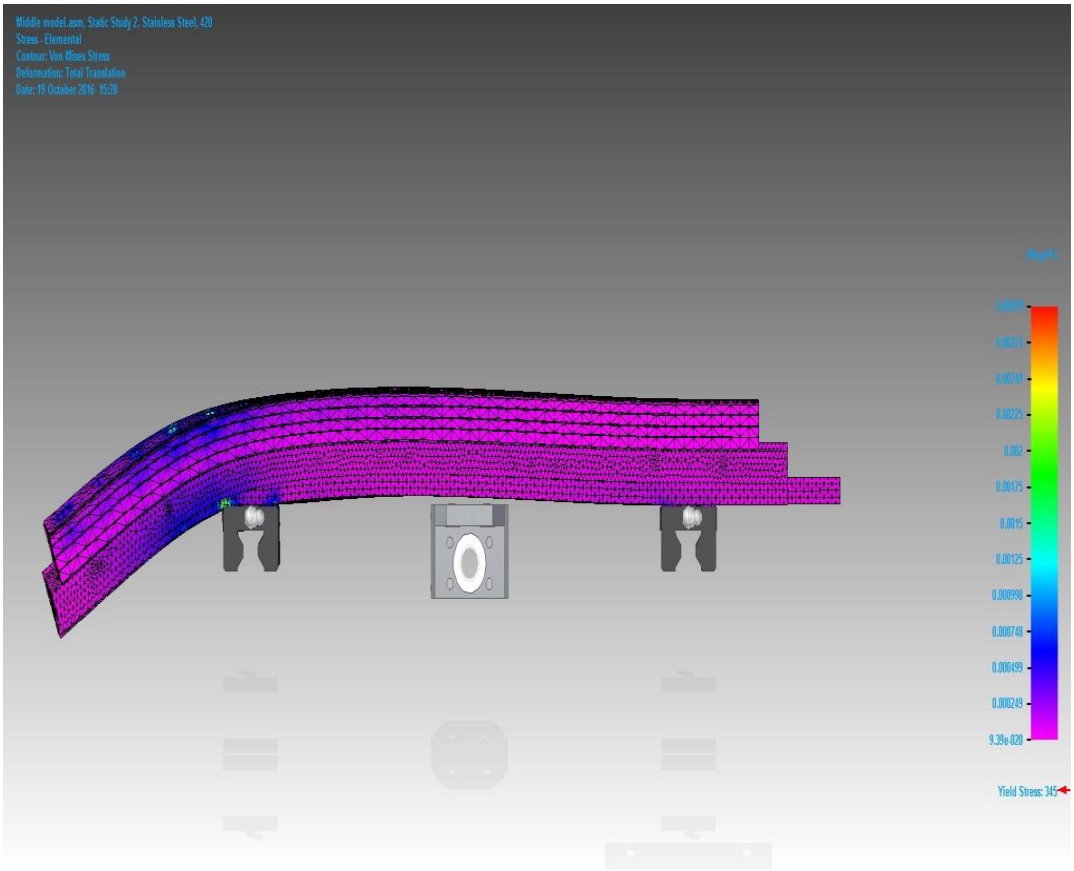
Result component: Total Translation				
Extent	**Value**	**X**	**Y**	**Z**
Minimum	0 mm	69.536 mm	547.000 mm	-31.000 mm
Maximum	2.69e-006 mm	110.000 mm	700.000 mm	25.000 mm



Total Translation

11.2 Stress Results

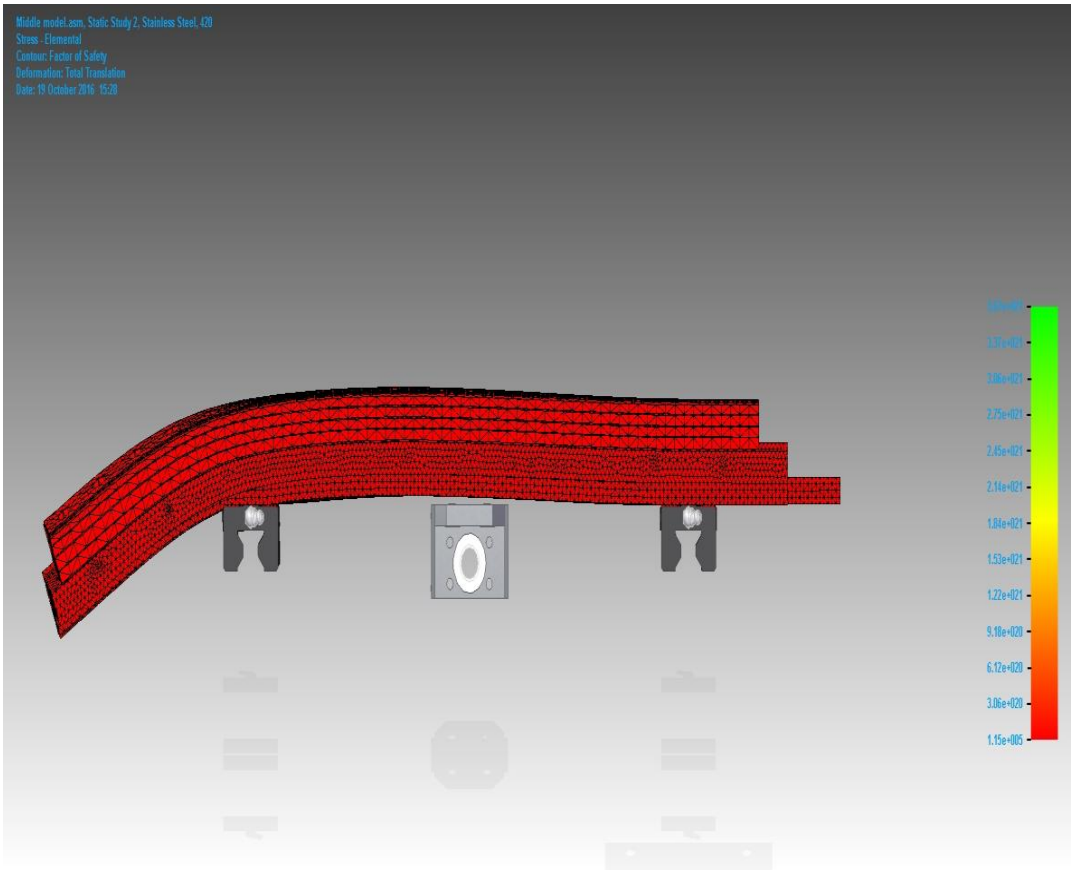
Result component: Von Mises				
Extent	**Value**	**X**	**Y**	**Z**
Minimum	9.39e-020 MegaPa	22.750 mm	552.000 mm	-21.000 mm
Maximum	0.00299 MegaPa	-87.000 mm	551.837 mm	-0.000 mm



Von Mises

11.3 Factor of Safety Results

Result Component: Factor of Safety				
Extent	**Value**	**X**	**Y**	**Z**
Minimum	1.15e+005	-87.000 mm	551.837 mm	-0.000 mm
Maximum	3.67e+021	22.750 mm	552.000 mm	-21.000 mm



Factor of Safety

---

12. Optimizations

---

13. Conclusion

---

14. Disclaimer

Important Information

This report should not be used to solely judge a design idea's suitability to a given set of environmental conditions. Siemens makes every effort to ensure that its products provide as much guidance and help as possible. However this does not replace good engineering judgment, which is always the responsibility of our users. A qualitative approach to engineering should ensure that the results of this evaluation are evaluated in conjunction with the practical experience of design engineers and analysts, and ultimately experimental test data. The results contained within this report are believed to be reliable but should not be construed as providing any sort of warranty for fitness of purpose.

B.3 Alignment Clamp

Clamp 24kN

Company
KTH

Author
I am µ

Date
den 24 oktober 2016

Software Used
Solid Edge ST(108.00.00.091 x64)
Femap (11.1.2)

Solver Used
NX Nastran (9)

Table of Contents

---

1. Introduction

---

2. Model Information

Document	clamp.par

3. Study Properties

Study Property	Value
Study name	Static Study 3
Study Type	Linear Static
Mesh Type	Tetrahedral
Iterative Solver	On
NX Nastran Geometry Check	On
NX Nastran command line	
NX Nastran study options	
NX Nastran generated options	
NX Nastran default options	
Surface results only option	On

4. Study Geometry

4.1 Solids

Solid Name	Material	Mass	Volume	Weight
clamp.par	Stainless Steel, 420	0,020 kg	2589,416 mm^3	196,666 mN

5. Material Properties

5.1 Stainless Steel, 420

Property	Value
Density	7750,000 kg/m^3
Coef. of Thermal Exp.	0,0000 /C
Thermal Conductivity	0,025 kW/m-C
Specific Heat	502,000 J/kg-C
Modulus of Elasticity	199947,953 MegaPa
Poisson's Ratio	0,290
Yield Stress	344,738 MegaPa
Ultimate Stress	655,002 MegaPa
Elongation %	0,000

6. Override Properties

---

7. Loads

Load Name	Load Type	Load Value	Load Distribution	Load Direction	Load Direction Option
Force 1	Force	2,4e+007 mN	Per Entity	(-1,00, 0,00, 0,00)	Along a vector

---

8. Constraints

Constraint Name	Constraint Type	Degrees of Freedom
Fixed 1	Fixed	FREE DOF: None

---

9. Mesh Information

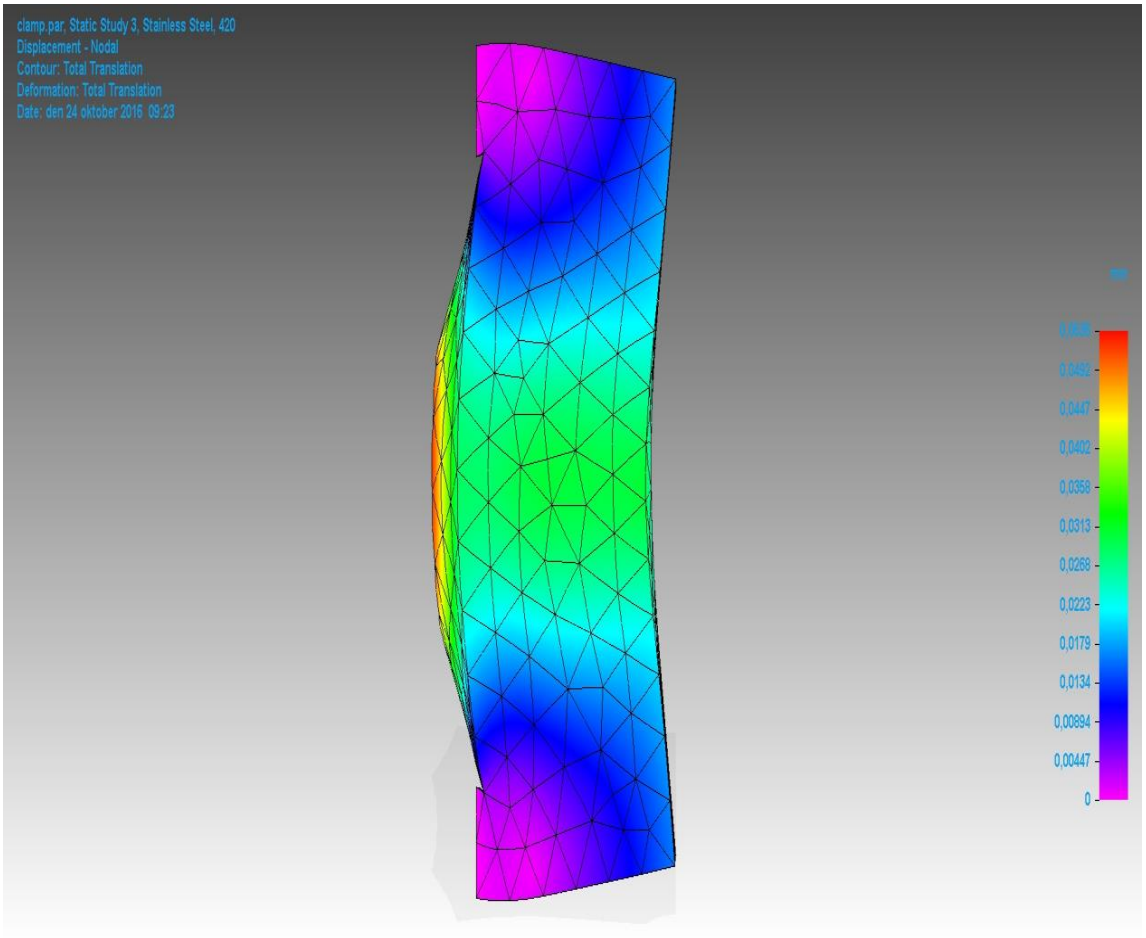
Mesh type	Tetrahedral
Total number of bodies meshed	1
Total number of elements	4 166
Total number of nodes	6 852
Subjective mesh size (1-10)	3

---

10. Results

10.1 Displacement Results

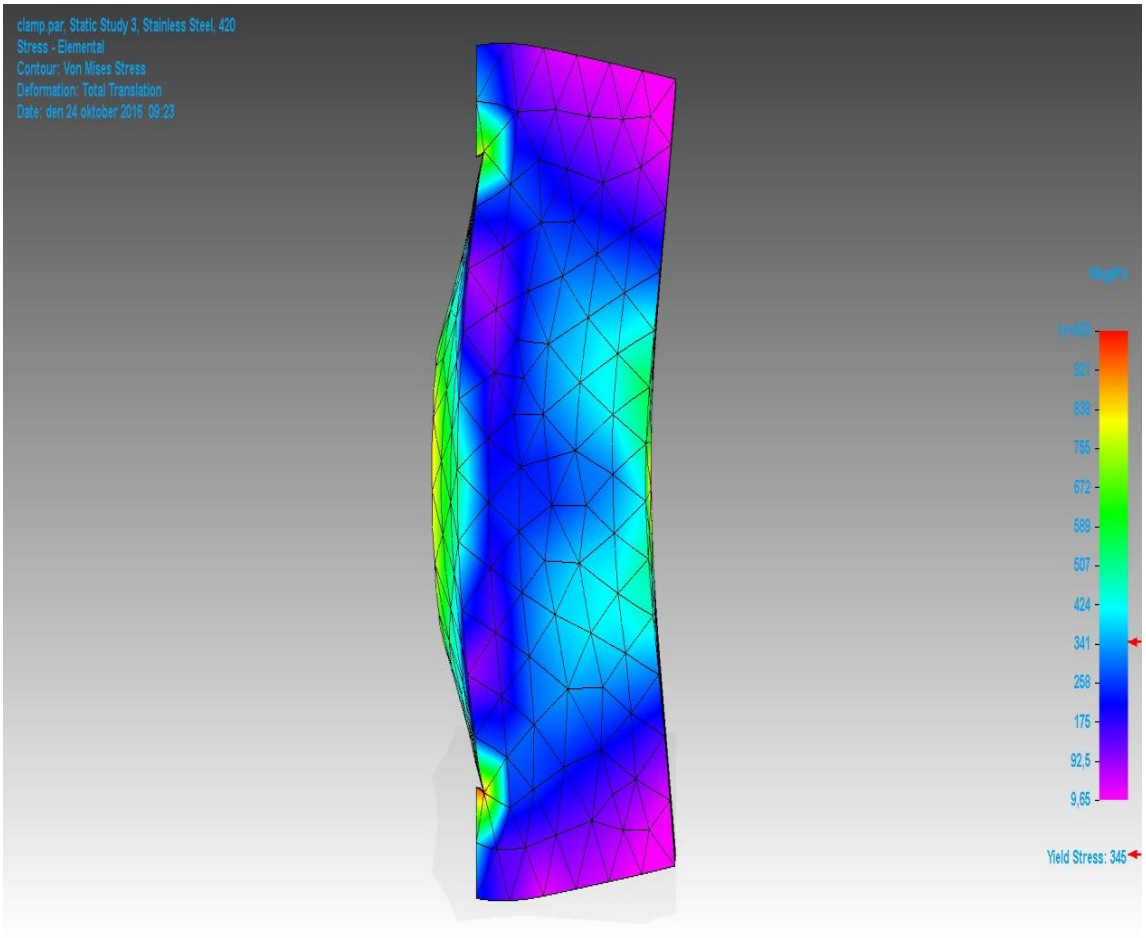
	Result component: Total Translation			
Extent	**Value**	**X**	**Y**	**Z**
Minimum	0 mm	-0,500 mm	9,375 mm	-11,500 mm
Maximum	0,0536 mm	2,700 mm	4,514 mm	0,300 mm



Total Translation

10.2 Stress Results

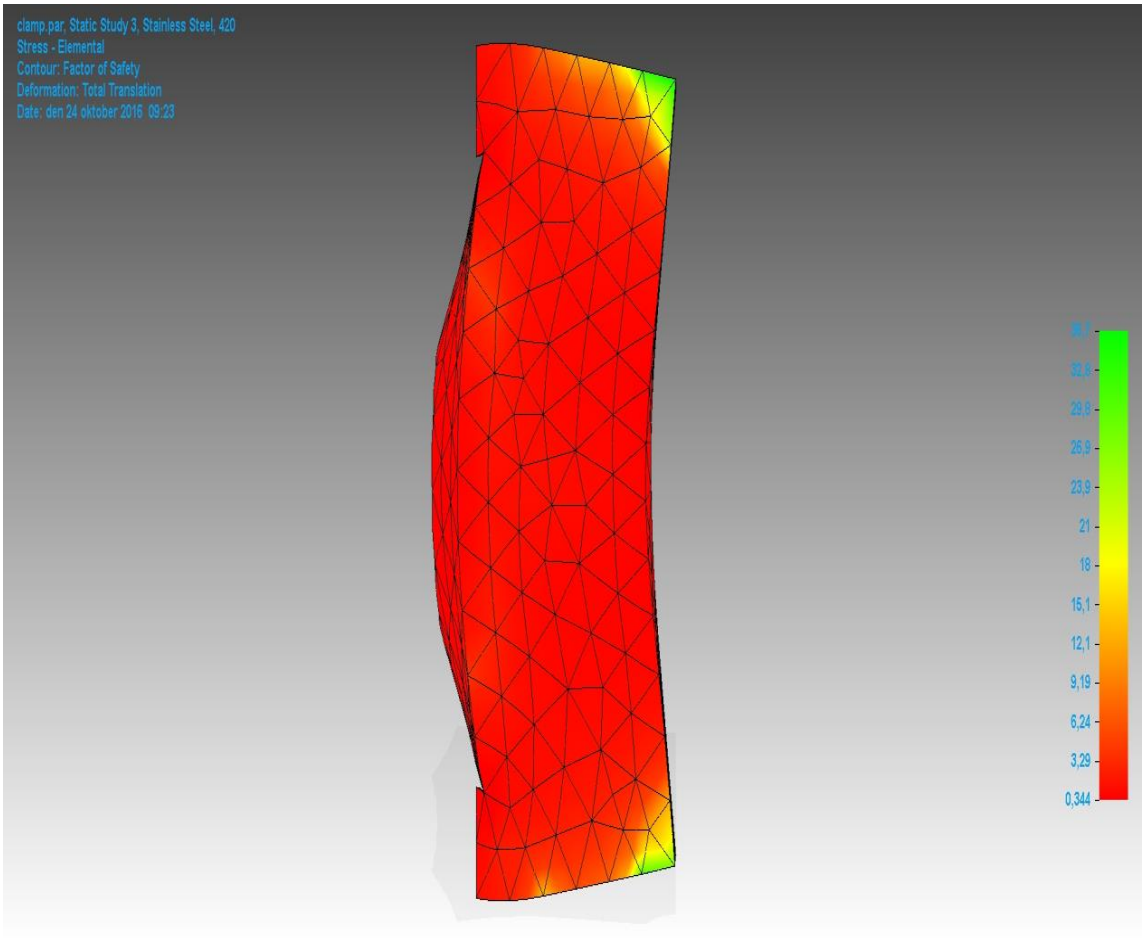
Result component: Von Mises				
Extent	**Value**	**X**	**Y**	**Z**
Minimum	9,65 MegaPa	9,500 mm	0,000 mm	11,500 mm
Maximum	1e+003 MegaPa	9,500 mm	13,000 mm	-0,500 mm



Von Mises

10.3 Factor of Safety Results

Result Component: Factor of Safety				
Extent	**Value**	**X**	**Y**	**Z**
Minimum	0,344	9,500 mm	13,000 mm	-0,500 mm
Maximum	35,7	9,500 mm	0,000 mm	11,500 mm



Factor of Safety

---

11. Optimizations

---

12. Conclusion

---

13. Disclaimer

Important Information

This report should not be used to solely judge a design idea's suitability to a given set of environmental conditions. Siemens makes every effort to ensure that its products provide as much

guidance and help as possible. However this does not replace good engineering judgment, which is always the responsibility of our users. A qualitative approach to engineering should ensure that the results of this evaluation are evaluated in conjunction with the practical experience of design engineers and analysts, and ultimately experimental test data. The results contained within this report are believed to be reliable but should not be construed as providing any sort of warranty for fitness of purpose.

B.4 Pusher on Rail

Rail pusher

Company
KTH

Author
I am µ

Date
19 December 2016

Software Used
Solid Edge ST(108.00.00.091 x64)
Femap (11.1.2)

Solver Used
NX Nastran (9)

Table of Contents

---

1. Introduction

2. Model Information

Document	Lower model.asm

3. Study Properties

Study Property	Value
Study name	Static Study 1
Study Type	Linear Static
Mesh Type	Tetrahedral
Iterative Solver	On
NX Nastran Geometry Check	On
NX Nastran command line	
NX Nastran study options	
NX Nastran generated options	
NX Nastran default options	
Surface results only option	On

4. Study Geometry

4.1 Solids

Solid Name	Material	Mass	Volume	Weight
SpacerV2Clean_inv.par:1	Steel (Default)	0.000 kg	0.000 mm^3	0.000 mN
SpacerV2Clean.par:1	Steel (Default)	0.000 kg	0.000 mm^3	0.000 mN
block.par:1	Steel (Default)	0.000 kg	0.000 mm^3	0.000 mN
Rail-R1A250600LCNP6Z-15-15-NSK.par:1	Steel (Default)	0.000 kg	0.000 mm^3	0.000 mN

5. Material Properties

5.1 Steel

Property	Value
Density	7833.000 kg/m^3
Coef. of Thermal Exp.	0.0000 /C
Thermal Conductivity	0.032 kW/m-C
Specific Heat	481.000 J/kg-C
Modulus of Elasticity	199947.953 MegaPa
Poisson's Ratio	0.290
Yield Stress	262.001 MegaPa
Ultimate Stress	358.527 MegaPa
Elongation %	0.000

---

6. Override Properties

---

7. Loads

Load Name	Load Type	Load Value	Load Distribution	Load Direction	Load Direction Option
Force 1	Force	8e+006 mN	Per Entity	(0.00, 0.00, 1.00)	Along a vector

---

8. Constraints

Constraint Name	Constraint Type	Degrees of Freedom
Fixed 1	Fixed	FREE DOF: None
Fixed 2	Fixed	FREE DOF: None

---

9. Connector

Connector Name	Connector Type	Search Distance	Minimum Search Distance	Coefficient Of Static Friction	Penalty Value
Connector 1	Glue	0.31 mm			100.00

Connector 2	Glue	0.31 mm	100.00

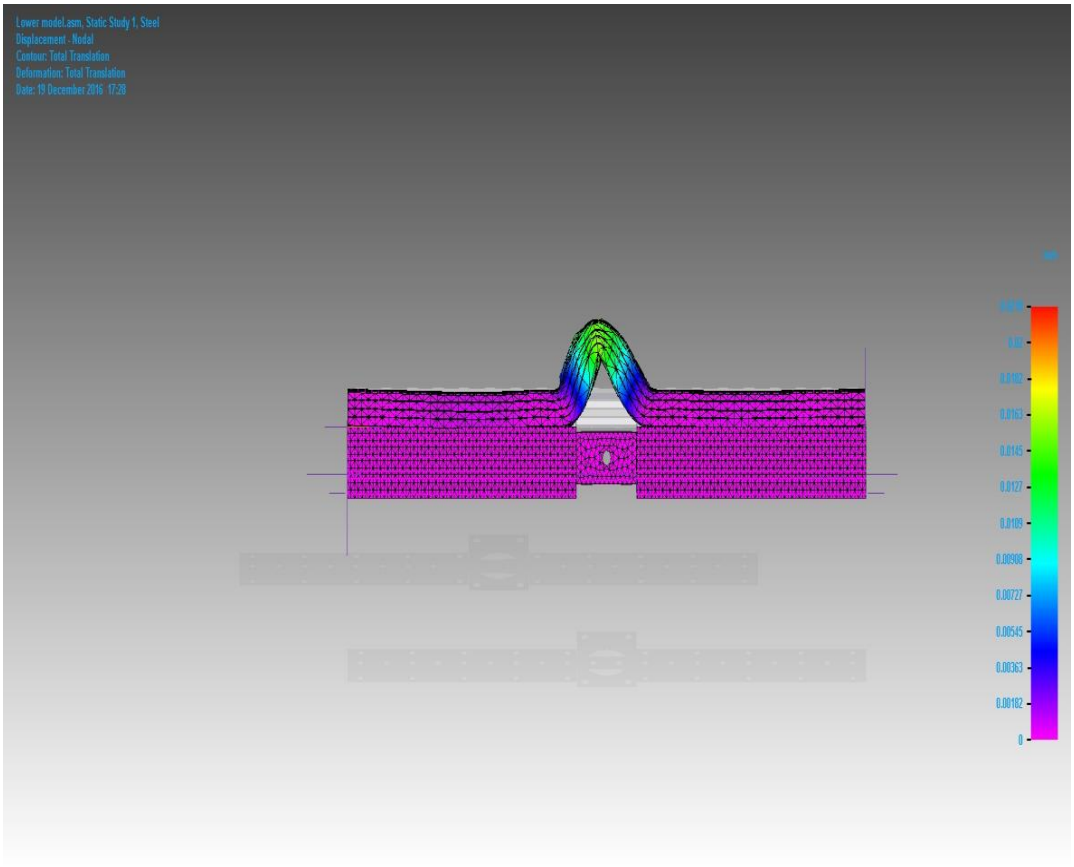
10. Mesh Information

Mesh type	Tetrahedral
Total number of bodies meshed	4
Total number of elements	154,436
Total number of nodes	238,478
Subjective mesh size (1-10)	3

11.Results

11.1 Displacement Results

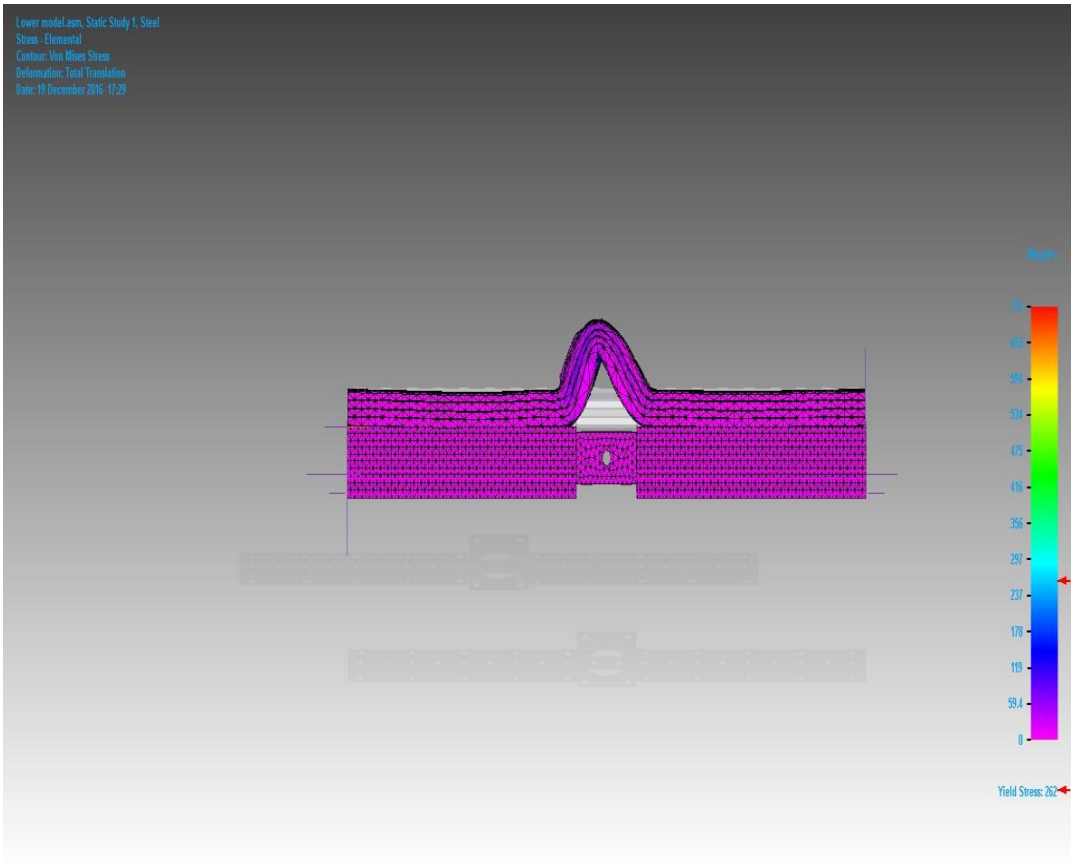
Result component: Total Translation				
Extent	**Value**	**X**	**Y**	**Z**
Minimum	0 mm	242.800 mm	124.500 mm	58.000 mm
Maximum	0.0218 mm	187.180 mm	78.187 mm	72.000 mm



Total Translation

11.2 Stress Results

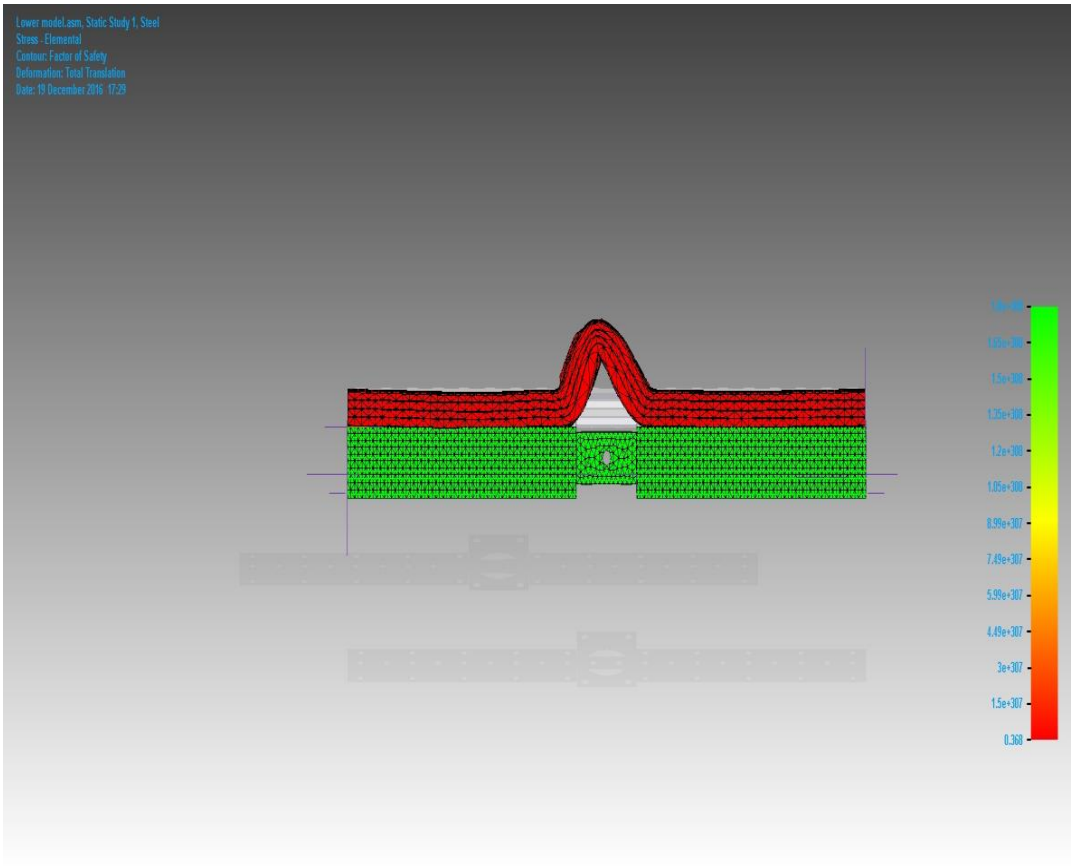
Result component: Von Mises				
Extent	**Value**	**X**	**Y**	**Z**
Minimum	0 MegaPa	242.800 mm	124.500 mm	58.000 mm
Maximum	712 MegaPa	192.233 mm	75.000 mm	72.000 mm



Von Mises

11.3 Factor of Safety Results

Result Component: Factor of Safety				
Extent	**Value**	**X**	**Y**	**Z**
Minimum	0.368	192.233 mm	75.000 mm	72.000 mm
Maximum	1.8e+308	242.800 mm	124.500 mm	58.000 mm



Factor of Safety

---

12. Optimizations

---

13. Conclusion

---

14. Disclaimer

Important Information

This report should not be used to solely judge a design idea's suitability to a given set of environmental conditions. Siemens makes every effort to ensure that its products provide as much guidance and help as possible. However this does not replace good engineering judgment, which is always the responsibility of our users. A qualitative approach to engineering should ensure that the results of this evaluation are evaluated in conjunction with the practical experience of design engineers and analysts, and ultimately experimental test data. The results contained within this report are believed to be reliable but should not be construed as providing any sort of warranty for fitness of purpose.

C Bill of Material of Used Bolts

Bill of Materials Bolts

Supplier: Sifvert Skruv AB Allen type: MC6S 12.9

Model	Part of entry	Part of exit	Size	Nut	Counterbore clear of socket he	Quantity	Checked	Checked with mod
Column1	Column2	Column22	Column	Column	Column62	Column7	Column8	Column82
Lower	SpacerV2	Baseplate	M6X14	No		10	2016-10-04	2016-10-13
Lower	SpacerV2_inv	Baseplate	M6X14	No		10	2016-10-04	2016-10-13
Lower	SpacerV2Clean	Baseplate	M6X14	No		10	2016-10-04	2016-10-13
Lower	SpacerV2Clean_inv	Baseplate	M6X14	No		10	2016-10-04	2016-10-13
Lower	Pushing block	Baseplate	M8X45	No	No counterbore	8	2016-10-04	2016-10-13
Lower	Rail-R1A250600LCNP6Z-15-15-NSK	Baseplate	M6X65	No		40	2016-10-04	2016-10-13
Lower	WBK12S-01	Baseplate	M8X55	No		2	2016-10-04	2016-10-13
Lower	Spacer_WBK12S-01_ver2	Baseplate	M8X30	No		4	2016-10-04	2016-10-13
Lower	Spacer_WBK12-01A_ver2	Baseplate	M8X30	No		4	2016-10-04	2016-10-13
Lower	WBK12-01A	Baseplate	M8X55	No		2	2016-10-04	2016-10-13
Lower	Motorbracket-stabilizer_ver2	Baseplate	M6X30	No	No counterbore	2	2016-10-04	2016-10-13
Lower	Motorbracker_lower_top_ver2	Motorbracker_lower_bottom_v	M6X8	No	No counterbore	2	2016-10-04	2016-10-13
Lower	Motorbracker_lower_top_ver2	Motorbracket-stabilizer_ver2	M6X14	M6	No counterbore	2	2016-10-04	2016-10-13
Lower	Motorbracker_lower_top_ver2	Danaher-AKM24X-ANCNC-00	M5X18	M5	No counterbore	4	2016-10-06	2016-10-17
Lower	Motorbracker_lower_bottom_ver2	Baseplate	M8X18	No		4	2016-10-06	2016-10-13
Lower	limit_switch_holder	Baseplate	M3X12	No		8	2016-10-06	2016-10-13
Lower	limit_switch_holder	limit_switch_SLIDER	M6X22	M6	No counterbore	4	2016-10-06	2016-10-13
Lower	Clamp	SpacerV2	M6X45	M6		8	2016-10-24	2016-10-24
Lower	Clamp	SpacerV2_inv	M6X45	M6		8	2016-10-24	2016-10-24
Lower	Baseplate	table	M20X60	M20		8	2016-10-24	2016-10-24
Middle	WBK12S-01	subplate	M8X45	M8		2	2016-10-06	2016-10-14
Middle	WBK12-01A	subplate	M8X45	M8		2	2016-10-06	2016-10-14
Middle	Nut-PSS1505-	ballscrew_fastener_ver2	M5X35	M5		4	2016-10-06	2016-10-17
Middle	ballscrew_fastener_ver2	subplate	M6X12	M6		4	2016-10-06	2016-10-17
Middle	sideplate_fastener	ballscrew_fastener_ver2	M3X6	No	No counterbore	6	2016-10-06	2016-10-14
Middle	Rail-R1A250600LCNP6Z-15-15-NSK	subplate	M6X35	M6		40	2016-10-24	2016-10-24
Middle	subplate	Slider-RAA25AL3-K-NSK_0	M6X12	No		24	2016-10-06	2016-10-14
Middle	Motorbracket_upper_hook_right	subplate	M8X18	M8		2	2016-10-06	2016-10-14
Middle	Motorbracket_upper_hook	subplate	M8X18	M8		2	2016-10-06	2016-10-14
Middle	Danaher-AKM24X-ANCNC-00	Motorbracket_upper_hook	M5X30	M5	No counterbore	1	2016-10-06	2016-10-17
Middle	Danaher-AKM24X-ANCNC-00	Motorbracket_upper_hook_righ	M5X30	M5	No counterbore	1	2016-10-06	2016-10-17
Middle	Danaher-AKM24X-ANCNC-00	Motorbracket_upper_hook	M5X45	M5	No counterbore	1	2016-10-06	2016-10-31
Middle	Danaher-AKM24X-ANCNC-00	Motorbracket_upper_hook_righ	M5X45	M5	No counterbore	1	2016-10-06	2016-10-31
Middle	limit_switch_holder_top	subplate	M5X30	No	No counterbore	4	2016-10-06	2016-10-17
Middle	Clamp	subplate	M6X65	No		28	2016-10-24	2016-10-24
Upper	subplate_top	Slider-RAA25AL3-K-NSK_0	M6X8	No		24	2016-10-06	2016-10-14
Upper	subplate_top	ballscrew_fastener_upper_ver2	M6X12	M6		4	2016-10-06	2016-10-17
Upper	Nut-PSS1505-	ballscrew_fastener_upper_ver2	M5X20	M5		4	2016-10-06	2016-10-14

Total with alignment: 304

D Matlab Script

Matlab Code

```
clear all
close all
clc

i = 0;
Position1_Old = [];
Velocity_Old = [];
Position2_Old = [];

PositionSetpoint1_Old = [];
PositionSetpoint2_Old = [];
VelocitySetpoint_Old = [];

Position1 = [];
Velocity = [];
Position2 = [];

PositionSetpoint1 = [];
PositionSetpoint2 = [];
VelocitySetpoint = [];

I = 0;


prompt = 'Enter number of runs: ';
```

```
26 j = input(prompt);
27 j=j-1;
28 for i = 0:j
29     FileNameEnc = ['20161216_121931_Test_', num2str(i),
       '_MotorsMotion.txt'];
30     Enc_Matrix = dlmread(FileNameEnc,'\t',1,0);
31     [x,y] = size(Enc_Matrix);
32
33     FileNameIMU = ['20161216_121931_Test_', num2str(i),
       '_Acc.txt'];
34     IMU_Matrix = dlmread(FileNameIMU,'\t',1,0);
35     [X,Y] = size(IMU_Matrix);
36
37     FileNameIMU_gyro = ['20161216_121931_Test_',
       num2str(i),'_Gyro.txt']
38     IMU_Matrix_gyro = dlmread(FileNameIMU_gyro,'\t'
       ,1,0);
39     [X_gyro,Y_gyro] = size(IMU_Matrix_gyro);
40
41
42     %After loading the data the number of datapoints is
       compared against
43     %previous runs and the shortest dataset is
       identified
44     datapoints_Enc = x;
45     datapoints_IMU = X;
46     datapoints_gyro = X_gyro;
47     if I == 0
48         min_data_Enc = datapoints_Enc;
49         min_data_IMU = datapoints_IMU;
50         min_data_gyro = datapoints_gyro;
51     end
52     if I > 0
53         if min_data_Enc > datapoints_Enc
54             min_data_Enc = datapoints_Enc;
55         end
56         if min_data_IMU > datapoints_IMU
57             min_data_IMU = datapoints_IMU;
```

```
58          end
59          if min_data_gyro > datapoints_gyro
60              min_data_gyro = datapoints_gyro;
61          end
62      end
63
64
65      % The vectors containing the data is picked and
            added to the previous
66      % data
67      Position1 = Enc_Matrix(1:1:min_data_Enc,1);
68      Position2 = Enc_Matrix(1:1:min_data_Enc,2);
69      Velocity = Enc_Matrix(1:1:min_data_Enc,3);
70
71      PositionSetpoint1 = Enc_Matrix(1:1:min_data_Enc,4);
72      PositionSetpoint2 = Enc_Matrix(1:1:min_data_Enc,5);
73      VelocitySetpoint = Enc_Matrix(1:1:min_data_Enc,6);
74
75
76      AccX = IMU_Matrix(1:1:min_data_IMU,1);
77      AccY = IMU_Matrix(1:1:min_data_IMU,2);
78      AccZ = IMU_Matrix(1:1:min_data_IMU,3);
79
80      GyroX = IMU_Matrix_gyro(1:1:min_data_gyro,2);
81      GyroY = IMU_Matrix_gyro(1:1:min_data_gyro,3);
82      GyroZ = IMU_Matrix_gyro(1:1:min_data_gyro,4);
83
84
85      if I == 0
86          Position1_Old = Position1;
87          Velocity_Old = Velocity;
88          Position2_Old = Position2;
89          PositionSetpoint1_Old = PositionSetpoint1;
90          PositionSetpoint2_Old = PositionSetpoint2;
91          VelocitySetpoint_Old = VelocitySetpoint;
92
93          AccX_Old = AccX;
94          AccY_Old = AccY;
```

```
95          AccZ_Old = AccZ-mean(AccZ);
96
97          GyroX_Old = GyroX;
98          GyroY_Old = GyroY;
99          GyroZ_Old = GyroZ;
100     else
101         Position1_Old = Position1_Old(1:min_data_Enc) +
                Position1;
102         Velocity_Old = Velocity_Old(1:min_data_Enc) +
               Velocity;
103         Position2_Old = Position2_Old(1:min_data_Enc) +
                Position2;
104
105
106
107         AccX_Old = AccX_Old(1:min_data_IMU) + AccX;
108         AccY_Old = AccY_Old(1:min_data_IMU) + AccY;
109         AccZ_Old = AccZ_Old(1:min_data_IMU) + AccZ-mean
               (AccZ);
110
111         GyroX_Old = GyroX_Old(1:min_data_gyro) + GyroX;
112         GyroY_Old = GyroY_Old(1:min_data_gyro) + GyroY;
113         GyroZ_Old = GyroZ_Old(1:min_data_gyro) + GyroZ;
114     end
115
116     I = 1;
117
118 end
119 i=i+1;
120 % The mean values are calculated by dividing the added
       data with the number of runs
121 Position1Mean = Position1_Old/i;
122 VelocityMean = Velocity_Old/i;
123 Position2Mean = Position2_Old/i;
124
125 AccXMean = AccX_Old/(i*1.95234)
       *9.82*1000-1.193506766048837e+02;              %hard
       coded compensation
```

```
126 AccYMean = AccY_Old/(i*1.97413)
        *9.82*1000-1.812515110894969e+02;
127 AccZMean = AccZ_Old/(i*1.95552)*9.82*1000;
128 format long;
129
130 GyroXMean = GyroX_Old/i;
131 GyroYMean = GyroY_Old/i;
132 GyroZMean = GyroZ_Old/i;
133
134 GyroXMeanOld = GyroXMean;
135 GyroYMeanOld = GyroYMean;
136 GyroZMeanOld = GyroZMean;
137
138
139 %Seems like the accelerometer drifts from the
        calibrated values within
140 %minutes of calibration, became worse and worse for
        each test, therefore we
141 %are approximating that the accelerometer does not
        travel in z direction
142 %and recalibrating the accelerometer for each dataset.
        (this is
143 %theoretically not correct but the signal should still
        be in there, only offset)
144 AccZMean = AccZMean-mean(AccZMean);
145 AccYMean = -AccYMean;
146
147 VelIntegrX = [];
148 VelIntegrX(1) = 0;
149 AccIntegrX = [];
150 AccIntegrX(1) = 0;
151
152 for i=2:length(AccXMean)
153     VelIntegrX(i) = (VelIntegrX(i-1) + AccXMean(i)
            *0.0005);
154     AccIntegrX(i) = (AccIntegrX(i-1) + VelIntegrX(i)
            *0.0005 + 0.5*AccXMean(i)*0.0005^2);
155 end
```

```
156
157 VelIntegrY = [];
158 VelIntegrY(1) = 0;
159 AccIntegrY = [];
160 AccIntegrY(1) = 0;
161
162 for i=2:length(AccYMean)
163     VelIntegrY(i) = (VelIntegrY(i-1) + AccYMean(i)
        *0.0005);
164     AccIntegrY(i) = (AccIntegrY(i-1) + VelIntegrY(i)
        *0.0005 + 0.5*AccYMean(i)*0.0005^2);
165 end
166
167 VelIntegrZ = [];
168 VelIntegrZ(1) = 0;
169 AccIntegrZ = [];
170 AccIntegrZ(1) = 0;
171
172 for i=2:length(AccZMean)
173     VelIntegrZ(i) = (VelIntegrZ(i-1) + AccZMean(i)
        *0.0005);
174     AccIntegrZ(i) = (AccIntegrZ(i-1) + VelIntegrZ(i)
        *0.0005 + 0.5*AccZMean(i)*0.0005^2);
175 end
176 AccIntegrZold=AccIntegrZ;
177 clearvars VelIntegrZ AccIntegrZ AccZMean AccZ AccZ_Old;
178
179 t_IMU = [0:1:min_data_IMU-1];
180 t_gyro = [0:1:min_data_gyro-1];
181
182
183 %
184 % figure;
185 %
186 % subplot(3,1,1)
187 % plot(t_IMU, GyroXMean(1:min_data_IMU))
188 % title('Roll')
189 % ylabel('Degrees/time [Unit]')
```

```
190 % xlabel('Time (ms)')
191 %
192 % subplot(3,1,2)
193 % plot(t_IMU, GyroYMean(1:min_data_IMU))
194 % title('Pitch')
195 % ylabel('Degrees/Time [Unit]')
196 % xlabel('Time (ms)')
197 %
198 % subplot(3,1,3)
199 % plot(t_IMU, GyroZMean(1:min_data_IMU))
200 % title('Yaw')
201 % ylabel('Degrees/Time [Unit]')
202 % xlabel('Time (ms)')
203 i = 0;
204 Position1_Old = [];
205 Velocity_Old = [];
206 Position2_Old = [];
207
208 PositionSetpoint1_Old = [];
209 PositionSetpoint2_Old = [];
210 VelocitySetpoint_Old = [];
211
212 Position1 = [];
213 Velocity = [];
214 Position2 = [];
215
216 PositionSetpoint1 = [];
217 PositionSetpoint2 = [];
218 VelocitySetpoint = [];
219
220 I = 0;
221
222
223 prompt = 'Enter number of runs: ';
224 j = input(prompt)
225 j=j-1;
226 for i = 0:j
227     FileNameEnc = ['20161216_123508_Test_', num2str(i),
```

```
        '_MotorsMotion.txt'];
228     Enc_Matrix = dlmread(FileNameEnc,'\t',1,0);
229     [x,y] = size(Enc_Matrix);
230
231     FileNameIMU = ['20161216_123508_Test_', num2str(i),
            '_Acc.txt'];
232     IMU_Matrix = dlmread(FileNameIMU,'\t',1,0);
233     [X,Y] = size(IMU_Matrix);
234
235     FileNameIMU_gyro = ['20161216_123508_Test_',
            num2str(i),'_Gyro.txt']
236     IMU_Matrix_gyro = dlmread(FileNameIMU_gyro,'\t'
            ,1,0);
237     [X_gyro,Y_gyro] = size(IMU_Matrix_gyro);
238
239
240     %After loading the data the number of datapoints is
            compared against
241     %previous runs and the shortest dataset is
            identified
242     datapoints_Enc = x;
243     datapoints_IMU = X;
244     datapoints_gyro = X_gyro;
245 %     if I == 0
246 %         min_data_Enc = datapoints_Enc;
247 %         min_data_IMU = datapoints_IMU;
248 %     end
249         if min_data_Enc > datapoints_Enc
250             min_data_Enc = datapoints_Enc;
251         end
252         if min_data_IMU > datapoints_IMU
253             min_data_IMU = datapoints_IMU;
254         end
255         if min_data_gyro > datapoints_gyro
256             min_data_gyro = datapoints_gyro;
257         end
258
259
```

```
260     % The vectors containing the data is picked and
           added to the previous
261     % data
262     Position1 = Enc_Matrix(1:1:min_data_Enc,1);
263     Position2 = Enc_Matrix(1:1:min_data_Enc,2);
264     Velocity = Enc_Matrix(1:1:min_data_Enc,3);
265
266     PositionSetpoint1 = Enc_Matrix(1:1:min_data_Enc,4);
267     PositionSetpoint2 = Enc_Matrix(1:1:min_data_Enc,5);
268     VelocitySetpoint = Enc_Matrix(1:1:min_data_Enc,6);
269
270
271     AccX = IMU_Matrix(1:1:min_data_IMU,1);
272     AccY = IMU_Matrix(1:1:min_data_IMU,2);
273     AccZ = IMU_Matrix(1:1:min_data_IMU,3);
274
275     GyroX = IMU_Matrix_gyro(1:1:min_data_gyro,2);
276     GyroY = IMU_Matrix_gyro(1:1:min_data_gyro,3);
277     GyroZ = IMU_Matrix_gyro(1:1:min_data_gyro,4);
278
279
280     if I == 0
281         Position1_Old = Position1;
282         Velocity_Old = Velocity;
283         Position2_Old = Position2;
284         PositionSetpoint1_Old = PositionSetpoint1;
285         PositionSetpoint2_Old = PositionSetpoint2;
286         VelocitySetpoint_Old = VelocitySetpoint;
287
288         AccX_Old = AccX;
289         AccY_Old = AccY;
290         AccZ_Old = AccZ-mean(AccZ);
291
292         GyroX_Old = GyroX;
293         GyroY_Old = GyroY;
294         GyroZ_Old = GyroZ;
295     else
296         Position1_Old = Position1_Old(1:min_data_Enc) +
```

```
            Position1;
297         Velocity_Old = Velocity_Old(1:min_data_Enc) +
                Velocity;
298         Position2_Old = Position2_Old(1:min_data_Enc) +
                Position2;
299
300
301
302         AccX_Old = AccX_Old(1:min_data_IMU) + AccX;
303         AccY_Old = AccY_Old(1:min_data_IMU) + AccY;
304         AccZ_Old = AccZ_Old(1:min_data_IMU) + AccZ-mean
                (AccZ);
305
306         GyroX_Old = GyroX_Old(1:min_data_gyro) + GyroX;
307         GyroY_Old = GyroY_Old(1:min_data_gyro) + GyroY;
308         GyroZ_Old = GyroZ_Old(1:min_data_gyro) + GyroZ;
309     end
310
311     I = 1;
312
313 end
314 i=i+1;
315 % The mean values are calculated by dividing the added
      data with the number of runs
316 Position1Mean = Position1_Old/i;
317 VelocityMean = Velocity_Old/i;
318 Position2Mean = Position2_Old/i;
319
320 AccXMean = AccX_Old/(i*1.95234)
      *9.82*1000-1.193506766048837e+02;              %hard
      coded compensation
321 AccYMean = AccY_Old/(i*1.97413)
      *9.82*1000-1.812515110894969e+02;
322 AccZMean = AccZ_Old/(i*1.95552)*9.82*1000;
323 AccZMean=AccZMean-mean(AccZMean);
324
325
326 GyroXMean = GyroX_Old/i;
```

```
327 GyroYMean = GyroY_Old/i;
328 GyroZMean = GyroZ_Old/i;
329
330
331 VelIntegrX = [];
332 VelIntegrX(1) = 0;
333 AccIntegrX = [];
334 AccIntegrX(1) = 0;
335
336 for i=2:length(AccXMean)
337     VelIntegrX(i) = (VelIntegrX(i-1) + AccXMean(i)
        *0.0005);
338     AccIntegrX(i) = (AccIntegrX(i-1) + VelIntegrX(i)
        *0.0005 + 0.5*AccXMean(i)*0.0005^2);
339 end
340
341 VelIntegrY = [];
342 VelIntegrY(1) = 0;
343 AccIntegrY = [];
344 AccIntegrY(1) = 0;
345
346 for i=2:length(AccYMean)
347     VelIntegrY(i) = (VelIntegrY(i-1) + AccYMean(i)
        *0.0005);
348     AccIntegrY(i) = (AccIntegrY(i-1) + VelIntegrY(i)
        *0.0005 + 0.5*AccYMean(i)*0.0005^2);
349 end
350
351 VelIntegrZ = [];
352 VelIntegrZ(1) = 0;
353 AccIntegrZ = [];
354 AccIntegrZ(1) = 0;
355
356 for i=2:length(AccZMean)
357     VelIntegrZ(i) = (VelIntegrZ(i-1) + AccZMean(i)
        *0.0005);
358     AccIntegrZ(i) = (AccIntegrZ(i-1) + VelIntegrZ(i)
        *0.0005 + 0.5*AccZMean(i)*0.0005^2);
```

```
359 end
360 %%%%%%%%%%%%%%%%%%%%%%%%intergrating gyro
361
362 GyroIntegrX = [];
363 GyroIntegrY = [];
364 GyroIntegrZ = [];
365 GyroIntegrX(1) = 0;
366 GyroIntegrY(1) = 0;
367 GyroIntegrZ(1) = 0;
368 GyroIntegrXOld = [];
369 GyroIntegrYOld = [];
370 GyroIntegrZOld = [];
371 GyroIntegrXOld(1) = 0;
372 GyroIntegrYOld(1) = 0;
373 GyroIntegrZOld(1) = 0;
374 %%%%Calibration
375
376 GyroXMean = GyroXMean -mean(GyroXMean);
377 GyroYMean = GyroYMean -mean(GyroYMean);
378 GyroZMean = GyroZMean -mean(GyroZMean);
379 GyroXMeanOld = GyroXMeanOld - mean(GyroXMeanOld);
380 GyroYMeanOld = GyroYMeanOld - mean(GyroYMeanOld);
381 GyroZMeanOld = GyroZMeanOld - mean(GyroZMeanOld);
382
383
384 for i=2:length(GyroXMean)
385     GyroIntegrX(i) = (GyroIntegrX(i-1) + GyroXMean(i)
        *0.001);
386     GyroIntegrY(i) = (GyroIntegrY(i-1) + GyroYMean(i)
        *0.001);
387     GyroIntegrZ(i) = (GyroIntegrZ(i-1) + GyroZMean(i)
        *0.001);
388     GyroIntegrXOld(i) = (GyroIntegrXOld(i-1) +
        GyroXMeanOld(i)*0.001);
389     GyroIntegrYOld(i) = (GyroIntegrYOld(i-1) +
        GyroYMeanOld(i)*0.001);
390     GyroIntegrZOld(i) = (GyroIntegrZOld(i-1) +
        GyroZMeanOld(i)*0.001);
```

```
391 end
392
393
394
395 if length(t_IMU)>min_data_IMU
396     t_IMU = [0:1:min_data_IMU-1];
397 else
398     min_data_IMU=length(t_IMU);
399 end
400
401 if length(t_gyro)>min_data_gyro
402     t_gyro = [0:1:min_data_gyro-1];
403 else
404     min_data_gyro=length(t_gyro);
405 end
406 % figure;
407 % plot(t_IMU, AccIntegrZold(1:min_data_IMU),'r',t_IMU,
      AccIntegrZ(1:min_data_IMU),'b');
408 % title('Acceleration Integrated Z-axis')
409 % ylabel('Accelerometer value')
410 % xlabel('Time (ms)')
411 % figure;
412 % datest= AccIntegrZold(1:min_data_IMU)-AccIntegrZ(1:
      min_data_IMU);
413 % plot(t_IMU, datest(1:min_data_IMU));
414 % title('Acceleration Integrated Z-axis')
415 % ylabel('Accelerometer value')
416 % xlabel('Time (ms)')
417 %
418 % figure;
419 % plot(t_IMU, AccZMean(1:min_data_IMU));
420 % title('Acceleration Integrated Z-axis')
421 % ylabel('Accelerometer value')
422 % xlabel('Time (ms)')
423
424
425
426
```

```
427 % subplot(3,1,1)
428 % plot(t_gyro, GyroIntegrX(1:min_data_gyro),'r',t_gyro,
      GyroIntegrXOld(1:min_data_gyro),'b')
429 % title('Roll')
430 % ylabel('Degrees/time [Unit]')
431 % xlabel('Time (ms)')
432 %
433 % subplot(3,1,2)
434 % plot(t_gyro, GyroIntegrY(1:min_data_gyro),'r',t_gyro,
      GyroIntegrYOld(1:min_data_gyro),'b')
435 % title('Pitch')
436 % ylabel('Degrees/Time [Unit]')
437 % xlabel('Time (ms)')
438 %
439 % subplot(3,1,3)
440 % plot(t_gyro, GyroIntegrZ(1:min_data_gyro),'r',t_gyro,
      GyroIntegrZOld(1:min_data_gyro),'b')
441 % title('Yaw')
442 % ylabel('Degrees/Time [Unit]')
443
444 figure;
445 plot(t_gyro, GyroIntegrX(1:min_data_gyro),'r',t_gyro,
     GyroIntegrXOld(1:min_data_gyro),'b')
446 title('Roll')
447 ylabel('Degrees/time [Unit]')
448 xlabel('Time (ms)')
449
450 %%%%
451 %difference
452 DiffGyro(1:min_data_gyro)=GyroIntegrX(1:min_data_gyro)-
     GyroIntegrXOld(1:min_data_gyro);
453
454 figure
455 plot(t_gyro,DiffGyro(1:min_data_gyro))
456 title('Roll')
457 ylabel('Degrees/time [Unit]')
458 xlabel('Time (ms)')
459 xlabel('Time (ms)')
```

E Pre-study

Test Bench for Monitoring and Control of Linear Axis Degradation Using Sensor-based Metrology

PABLO ALBIOL GRAULLERA
MATTIAS DAHLQVIST
MARTIN FAVRE
JONATHAN HYTÖNEN
SHIZUN LI
ODEN LOBELL
FILIP STENBECK
MARTIN ZAKARDISSNEHF

Mechatronics Advanced Course 2016

Supervisor: Bengt Eriksson
Examiner: Martin Edin Grimheden

Abstract

Machine tool linear axes are widely used in the industry for positioning the workpiece or tool. During usage the guideways will degrade and this will lead to inaccuracies in positioning. The current procedures to diagnose the magnitude of these inaccuracies and calibrate the systems are expensive and time consuming. This report presents a preliminary study of geometric error inducement in feed drive systems and presents concepts of how to induce these errors, it also investigates the possibility of error detection for stacked errors through an inertial measurement unit.

Preface

We want to thank Gregory Vogl and Andreas Archenti for enlightening discussions and for laying the foundation for this project.

We would also like to thank Bengt Eriksson for his feedback and guidance.

Finally we would like to thank Károly Szipka for invaluable help throughout the project.

Team I am μ

Contents

Nomenclature

Abbreviations

Abbreviation	Description
AC	Alternating Current
CCC	Cross Coupling Controller
CNC	Computer Numerical Control
CMM	Coordinate Measuring Machine
DC	Direct Current
DOF	Degrees of Freedom
EDM	Electrical Discharge Machining
FLC	Fuzzy Logic Controller
IDS	Inertial Displacement Sensors
IMU	Inertial Measurement Unit
ISO	International Organization for Standardization
MEMS	Micro Electronic Mechanical Systems
NC	Numerical Control
RPM	Revolutions Per Minute
TCM	Tool Condition Monitoring
TUR	Test Uncertainty Ratio
ZPETC	Zero Phase Error Tracking Controller

1 Introduction

Machine tools that depend on multi-axial feed drive systems are often used with high loads and frequent usage during their functional lifespan. Material fatigue, abrasion or adhesion will eventually cause degradation and this will affect the precision of the tool [Li et al., 2014]. To counteract the degradation of the CNC-machines visual inspection and preventive maintenance like calibration are done [Uhlmann et al., 2008].

In order to recalibrate the feed drive system today ball bars, cross grid encoders, and laser interferometers are used to measure the geometric errors [Xu et al., 2016]. This is often not cost effective since the machine must be taken of the production line during the calibration. Instead, if these errors would be measured and corrected autonomously it would lower production costs and prolong the life of the machine.

1.1 Background and Objective

This report is done as part of the Mechatronics Advanced Course, a project course given in the spring of 2016 and continuing in autumn. The project is done in collaboration with Royal Institute of Technology (KTH) and National Institute of Standards and Technology (NIST) and is a continuation of a study by Vogl et al.. The study investigates the error detection for one axial feed drive system using an IMU [Vogl et al., 2015].

The project is divided into three phases: (1) Building a two axis test bench with the possibility to introduce errors, (2) Development of a sensor-based measurement instrument, (3) Conducting uncertainty analysis and construct error budgets of measurements. The test bench should meet the requirements in Appendix A.

The project's primarily focus is building the test bench, i.e. the first two phases. The problem can be condensed to, (1), how to introduce geometric and repeatable errors in a two axis feed drive system? And (2), how to measure these errors?

The objective of this report is to do preliminary research of how to induce repeatable

stacked errors in the biaxial feed drive system and if they are possible to detect using an IMU and present the error inducing concepts.

1.2 Scope

The long term vision of the project is to develop a self-diagnostic and calibrating system for machine tools. This report is aimed to make the preliminary research on the subject and is limited to the construction of a two axis feed drive system, construction of an IMU, inducing artificial errors to the feed drive system and detecting these errors through an IMU. The actual correction and recalibration of the specific hardware will be excluded in this report.

2 Theory

In building a test rig a lot of components are needed, these components are often used in multiaxial machines such as CNC machines, 3D printers or CMM. This section introduces the necessary theory and components needed to build a test bench, as well as the basic concepts behind machine tool errors and how they affect the machined parts geometric and dimensional accuracy. The choice of components will affect the options of error inducement.

2.1 Test Bench

2.1.1 Guideways

Linear Rolling Guideways

To achieve a linear moving motion between the carriage and the guideway, rolling elements such as balls or rollers can be placed in either the carriage or the guideway. With this setup, high positional accuracy can be achieved with minimum driving force required since the contact between carriage and guideway is a rolling contact which reduces friction [HIWIN, 2013]. By applying an appropriate preload to the system the space between the rolling elements can be reduced which increases the stiffness and rigidity of the system [Shaw and Su, 2011].

Ball bearing elements are more applicable in high velocity applications due to their lower frictional resistance. However the load distribution on roller bearings is a line rather than a point as with ball bearings, which means that the roller bearing elements have a higher load capacity [NTN Corporation, 2001].

Different configurations of the rolling elements can be used to further enhance the characteristics of the rolling elements in linear motion guideways. For example, recirculating units can be used to increase the stroke length of the carriage and achieve further travel lengths. The bearing elements are placed in the carriage

rather on the guideway, so a limited number of bearing elements can achieve a longer travel length of the machine tool.

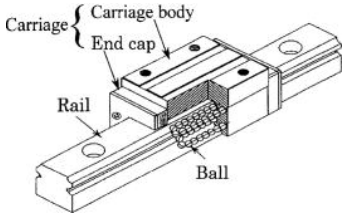


Figure 2.1. Recirculating ball bearings in a carriage.

The recirculating motion is achieved when the bearing elements that temporarily rolls on guideway come in contact with an endcap placed in the carriage, were the bearing elements circulating direction is forced to reversed and fed back into the carriage (see Fig. 2.1). With recirculation the rolling elements have constant lubrication as they are fed back into the carriage and thus increase the service life of the feed drive system. The rolling elements are contained and sealed within the carriage which reduces the chance of dust entering the rolling elements which eventually lowers the systems accuracy [Schaeffler Group, 2007].

With crossed roller bearings in the guideway, the rollers are placed in a crisscross pattern in the guideway to ensure a long contact length between the rollers and the carriage which gives a high rigidity in the system and less elastic deformation in the bearing element. Since the rollers are placed in the guideways itself, the stroke lengths of the carriage is limited by the lengths of the guideway [THK, 2007].

Hydrostatic Guideways

In the hydrostatic guideways the surface of the platform and rails are separated by cells that are pressurized with thin oil. Since there is no direct contact between the platform and rail the stick slip and static friction is eliminated. The guides have several cells that usually have different pressure in them to be able to handle varying operating conditions. To pressurize the cells one pump is used in combination with restrictors to give the cells different pressure. The hydrostatic guideways used to be expensive and hard to install but recently manufacturers have developed a new kind that is based on the standardized dimensions of the rolling guideways. This new version offer high precision, better damping for high dynamic rigidity and is more cost-effective than its predecessor [Altintas et al., 2011]. This new kind has a chamber system inside the platform that is charged with hydraulic oil. The oil is fed to the pressure sider under continuous pressure filling the pressure cells. Integrated chokes are set so the pressure in the cells is uniformly distributed. The unpressurised oil is extracted from the chamber system and is fed back to the oil circuit (see Fig. 2.2, [Schaeffler Technologies, 2015]).

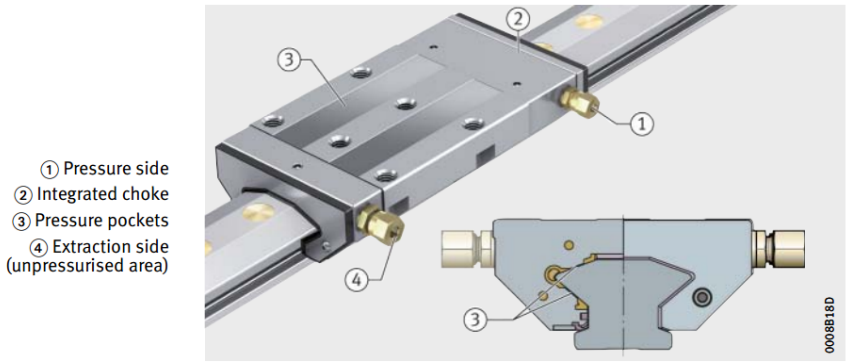


Figure 2.2. Functional parts of the compact hydrostatic guideway

2.1.2 Motors

Electrical motors can run either on AC or DC power and translates alternate or direct current into rotational or linear motion. They come in a number of various designs, to cover all would be beyond the scope of this report but there are three main categories. Brushless DC, brushed DC and induction, or asynchronous, AC motor. They all rely on forces produced by a magnetic field between the stators and a rotor, and the means of producing this magnetic field is what makes them differ from each other.

Brushed DC

The brushed DC motor is usually designed with permanent magnets in the stator and with windings in the rotor that change polarity as it turns using a mechanical commutator. The brushes of the commutator is subject to wear and require periodic replacement and the thermal efficiency is typically around 75%. The cost of a brushed DC motor is low and it is relatively easy to model for control purposes as it displays a linear relation between torque and current as well as voltage and RPM which makes the controller inexpensive as well [Gamazo-Real and Gómez-Gil, 2010]. However the windings in the rotor makes heat removal difficult and commutator arcing is a problem at high voltages, this limits the applications and it is therefore most commonly used in smaller low power and low end products [Krishnan, 2009].

Brushless DC

The brushless DC motor commonly uses permanent magnets in the rotor and change the polarity in the stator by running a current in the windings. This eliminates the

need for a mechanical commutator and the motor is thus less susceptible to wear. Changing the current in the stator to produce forces betweeen the rotor and the stator at the correct time requires more complex control. This can be done with analog hardware or digitally with software and a microcontroller in combination with encoders to read rotor position. This improves commutation efficiency as well control performance. The brushless motor have several advantages over the brushed DC, including higher torque per watt as well as torque to weight ratio leading to higher efficiency, typically around 90% [Krishnan, 2009]. Increased reliability, reduced noise, longer lifetime, less electromagnetic interference and easier heat reduction due to the heat generating parts being located on the outside of the motor. This comes at a substantial cost increase as well as a more complex control implementation.

Induction AC

An induction or asynchronous AC motor is run by alternating current. The rotor usually consists of windings in which a current is induced by the stator. The stator consists of windings in which the alternating current creates a rotating magnetic field, which creates forces to drive the rotor. They can come in both three phase as well as single phase configuration with the single phase typically being used for smaller loads and may have a fixed rotation direction. Although traditionally used in fixed speed applications, thanks to the advancement of control theory and variable frequency drives they are increasingly used in varible speed service. The efficiency is around 85% up to 97% and it has low wear as a result of the lack of mechanical commutators. The motor itself is relatively cheap but the need of inverters to handle the AC current adds some cost. The heat removal is relatively easy as most heat will be generated in the stator windings located on the outer part of the motor, but due to the induced currents in the rotor it may be required to cool it in high power applications. Modeling for control purposes is complex, and the control can consist of changing the frequency to control the speed as well as regulating the voltage or current of the supplied AC to control the torque. Varying the frequency is the most common way to control the motor and can be done to a high precision [Cheng et al., 2011].

Linear Motors

Linear motors have a lot of advantages, they have high precision, high speed and high acceleration, because of this they are becoming more prevalent in the industry, especially when high accuracy and precision is needed [Wang and Xi, 2008].

The principle of a linear motor is the same as an AC or DC motor but the motor is unfolded. It is possible to reach sub-micron precision with linear motion [Brandenburg et al., 2000]. Since there is no need to convert rotary motion to linear motion the friction is reduced, which reduces noise and increases the precision. Because of

the lack of centrifugal force the speed is in theory unlimited and greater than in rotary motors [Hu et al., 2013]. Some disadvantages is the cost associated with linear motors, compared to rotary motors and ball screws it is more expensive [Olarra et al., 2009].

2.1.3 Rotary Encoders

A rotary encoder, or shaft encoder, is an electro-mechanical device that converts the angular position or motion of a shaft or axle to an analog or digital code. The most common type are optical encoders, which consist of a light source and a sensor that are placed on opposing sides of a disc with sectors that are either transparent or opaque. These will produce pulses of light when the disc is spinning which are measurable. In incremental encoders the pulses of lights are counted and by knowing the amount of pulses per revolution the rotational velocity is derived. In absolute encoders the disc have specific pulse patters depending on which rotation the disc currently has [Wilson, 2005].

2.1.4 Software

In feed drive machines today many different types of control algorithms are used. The choice of which algorithm used are based on how fast your system should be and how accurate the modeling of the system is. Errors that will occur through software is position errors along the axis and contour errors. Where the latter is the error perpendicular to the reference trajectory toward the actual position. In order to minimize the errors in the system, feedback control is used. For low speed and low friction feed drive systems a proportional controller could be enough. Often a more advanced PID or state space controller is used in the industry. In this report the feedback control theory will not be explained further and will be assumed as previous knowledge. For feed drive systems that require more accuracy or have a higher complexity, control algorithms like different feedforward or cross-coupling controllers can be used. It should be acknowledged that there are multiple types of control algorithms that are used in this field that are not mentioned in this report, such as optimal control, adaptive control and robust control. These are more rarely used in industry and are considered to be less likely to be used in the project.

Feedforward Controllers

A feedforward controller is a controller that predicts the system or subsystem response in order to calculate the input for a desired output. This can often make the system faster but will require low model errors. There will always be model errors and to reduce them a feedback controller must also be implemented in conjunction with the feedforward controller. In general there are two different types of models for feedforward controllers [Koren and Lo, 1992].

- Trajectory control, where a feedforward part is the inverse of the plant and the control input is the sum of the output from the feedback controller and the feedforward controller. This will then make the feedforward controller only dependant on the plant thus making the design of the two controllers seperate to each other.

- Feedforward controller where the controller is the inverse of the total open loop system making the controller dependant on the feedback controller.

There are many examples of how to design the latter for feed drive systems. Zero Phase Error Tracking controller (ZPETC) is a methodology to control non-minimal phase systems [Tomizuka, 1987].

Cross Coupling Controller

Cross coupling controllers (CCC) is a method specific for biaxial systems and was first proposed by [Koren, 1980]. This method is a way to minimize the contour error for nonlinear and corner paths by taking the error for each axis to alternate the control signal. For usual feedback systems the computer predefines a path for each axis and does not take the the position error for the other axis in consideration. By using the cross coupling controller the input signal of each axis will also be affected by the contour error with a weighting factor W,

$$U_i(n) = E_i(n) + WE(n). \tag{2.1}$$

Where $U_i(n)$, in is the control signal at instance (n) for the ith axis, $E_i(n)$ is the position error for the ith axis at instance n and $E(n)$ is proportional to the contour error at instance (n). This example envolves a simple proportional controller with gain factor 1 for the position errors but can of course be combined with a more complex PID controller.

This system requires no extra hardware compared to regular feedback systems and improves the performance by controlling the system as a single unit rather then separate control loops in conjunction [Shih et al., 2002].

Fuzzy Logic Controller

FLC is a more robust way of generating the output and might be better for feed drive systems that have large friction disturbances than feed forward systems or the more traditional feedback systems. This controller have been suggested to be used in conjunction with the cross coupling controller with some success [Yeh et al., 1997].

The main advantage with fuzzy logic is that it can be used for very complex systems where a precise model cant be achieved. FLC has been used in CNC-machines

mostly to adjust the feed rate in the cutting process where there is degeneration of the cutting tool [Kim and Jeon, 2011].

The basic principle of fuzzy logic is to derive the controller input through probability analysis of the sensor data. This can than make complex systems have a less deterministic output given a specific control system.

2.2 Geometric Errors in Linear Axes

Geometric errors in machine tools lead to performance degradation, which affects the final geometric and dimensional accuracy of machined workpieces negatively. These faults account for economic losses up to tens of billions of US dollars per year [Vogl et al., 2015]. Thus, this is an area that has been widely researched.

In a three-axial CNC–machine there are 21 geometric error sources in every direction that directly affect the accuracy performance [Caballero-Ruiz et al., 2007]. Errors can be classified into three levels: volumetric errors, axis motion errors and geometric errors, where volumetric and axis errors are the source of geometric errors.

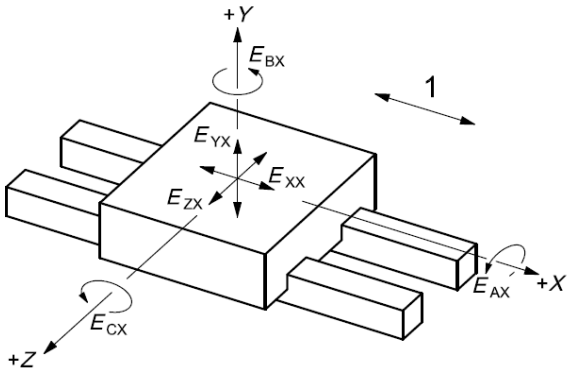


Figure 2.3. Angular and linear error motions of a component commanded to move along a (nominal) straight-line trajectory parallel to the X-axis.

The different errors shown in Fig. 2.3 are due to many error sources. Among others, the most important causes of errors are: of kinematic errors, thermo-mechanical errors, loads, dynamic forces and motion control and control software [Schwenke et al., 2008].

One of the objectives of this project is to develop a sensor-based measurement instrument in order to monitor linear axis degradation in feed drive systems. This knowledge could be very useful for the industry since more automated and efficient methods are needed to self-diagnose the condition of a machine as well as to plan

its calibration.

Axis degradation leads to translational (straightness) and angular errors (see Fig. 2.4), so the measurement instrument is expected to identify these deviations. According to the ISO 10791-2 standard the acceptable straightness error is limited to 20 μm and the angular error is limited to 60 μrad (for linear axis of vertical machining centers).

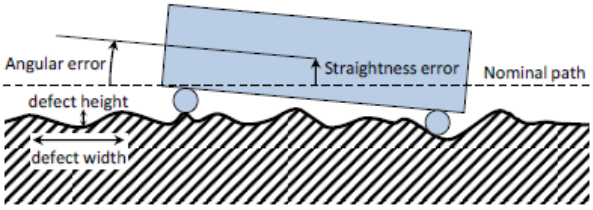


Figure 2.4. Schematic of effect of linear axis degradation on axis motion (courtesy: Gregory W. Vogl, NIST).

In order to measure these errors, the measurement uncertainties must be less than the specified tolerances. Vogl et al. recommends a TUR of at least 4:1 (5 μm of straightness error uncertainty and 15 μrad of angular error uncertainty) but a TUR of 10:1 or 100:1 would be desired (Table 2.1).

TUR	Straightness Error Uncertainty	Angular Error Uncertainty
4:1	5 μm	15 μrad
10:1	2 μm	6 μrad
100:1	0.2 μm	0.6 μrad

Table 2.1. Measurement uncertainties for straightness and angular errors based on TUR values.

3 State of the Art

This chapter discusses the state of the art of what is considered the key problems in this project. This is determined to be how to create repeatable and controllable errors in feed drive systems and how to identify them with an IMU. Then, this is followed by a short discussion on how to compensate for these errors.

3.1 Geometric Error Generation

Vogl et al. introduced errors in three different ways. The first way was to raise the rail using shims; this error will simulate the low spatial frequency degradation of a machine tool [Vogl, 2016].

To simulate the high frequency errors Vogl et al. took out the balls from one of the trucks and introduced errors to them by removing material from them [Vogl, 2016]. Vogl used the same technique to introduce the error on all of the balls so the size of the errors where roughly the same. Between the tests he took out all the balls again adding one additional error on all of the balls in a random location. The reason for this is to simulate how the errors increase over time. An other way that Vogl introduced errors was by deforming the rails using a punch to add small dents that the trucks would roll over to create high frequency errors.

It is possible to simulate degradation in the physical machine tool testing system by inducing geometric errors in the feed-drive system. This can be accomplished by changing the geometric characteristics of the system by grinding away small parts of the carriage itself. A method to analyse this is by examining the vibration and acoustic emissions while running multiple carriages with different sized artificially made defects on a linear guideway with recirculating ball bearings [Ohta et al., 2010]. Research has been conducted regarding this subject by comparing six carriages with different defects with a reference carriage with no defects. The carriages run over a guideway at constant linear velocity of 1 m/s and the vibrations and acoustic emissions were analysed by an accelerometer and acoustic emission sensor. The defects in the carriages showed an increase of vibrations and acoustic emissions.

This was not caused by the depth or the length of the defects, but the defect angle [Ohta et al., 2010].

In a study by Verl et al. a test bench with an error was needed, they introduced the errors by using at least one worn out part. To introduce an error that is similar to the real ones that occur over time in a reasonable time they accelerated the wear progress by adding abrasive material to the lubrication [Verl et al., 2009].

3.2 Geometric Error Detection

As discussed previously, a measurement instrument and a sensor based method is needed in order to study the degradations phenomena in linear axes of machine tools and to be able to help manufacturers to plan maintenance, production and improve part quality in an automated way.

Several attempts have been made using different sensors and methods but their success has been limited. One suggestion utilizes a sensorless method to monitor mechanical degradation by executing predefined excitation cycles in order to identify typical disturbances, such as pittings and backlash on the guideways [Plapper and Weck, 2001]. Another makes use of a motor torque sensor to monitor the status of the translational axes [Li et al., 2014]. Feng and Pan have studied the relation between ball screw preload variations and vibrations in the machine parts, which are identified by some accelerometers [Feng and Pan, 2012]. Another attempt has been made by Garinei and Marsili to detect defects in ball screw actuators through a Hall effect sensor [Garinei and Marsili, 2012]. And Biehl et al. have also tried to measure the preload in ball screw drives but with a piezoresistive thin film sensor [Biehl et al., 2012].

TCM systems are systems that monitor the condition of the machine tool. They do not determine the exact position, instead they work by monitoring the wear and tear in order to avoid tool breakdown [Rehorn et al., 2005]. A variety of sensors have been used in research some of which could be could be used in error detection. Acoustic emission sensors have been used to measure the contact between the work piece and the tool [Kumar et al., 2010]. In one study, capacitance sensors were used to measure geometric errors in a miniaturized machine tool [Lee and Yang, 2005]. Capacitive sensors in combination with laser interferometry have also been used to measure all the geometric errors in a three-axis machine tool [Lee et al., 2016].

The characteristics of the sensors used limited the success of these experiments. On one hand, built-in sensors usually have a high accuracy but it is difficult to place them close to the tracked mechanical components. On the other hand, external sensors, such as laser interferometers, give very good results but they are very expensive and the cost associated with does not make it worthwhile.

In conclusion, a cost-effective and flexible tool that can diagnose the state of a machine tool in a short period of time is needed. For all these reasons, an IMU is proposed as the measurement instrument. Even if it may not have seemed possible to use an IMU for this application in the past, the development of these sensors under the last years have made them viable for high accuracy and low noise purposes.

3.2.1 Inertial Measurement Unit

An IMU is a device for measuring three axis acceleration, angular turning rate and sometimes magnetic field around its body. Its construction consist of a combination of accelerometers, gyroscopes and occasionally magnetometers, in a single package. The data gather ed from these sensors can be used, after fusing the different measurements, to estimate velocity, position and orientation.

IMUs have traditionally mainly been used in inertial navigation systems of aircrafts, spacecrafts, missiles and drones. However, the development of MEMS under the last years have made it possible to use these sensors in many other different areas, such as, manufacturing, robotics, sports, medical rehabilitation or augmented reality [Ahmad et al., 2013].

For example, Godha and Cannon propose to integrate a GPS with a low cost IMU for land vehicle navigation with the aim to reduce GPS constraints and improve positional accuracy [Godha and Cannon, 2005]. Another application is suggested by Won et al. who use an IMU to keep track of a fastening tool and to identify fastened bolts [Won et al., 2009]. Ciuti et al. took advantage of different MEMS to monitor, sense and give feedback to users about their health status, as well as information during physical activities [Ciuti et al., 2015].

Regarding the objective of the project, IMUs have also been used in feed drive systems of machining tools. Vogl et al. have developed an instrument (formed by individual accelerometers, gyroscopes and inclinometers) and a sensor-method to estimate the condition of a linear axis with the aim to identify linear axis degradation [Vogl et al., 2015]. Liao and Lee (2009) have also studied machine performance degradation in a chiller system with the aid of 6 accelerometers and a fixed cycle test procedure [Liao and Lee, 2009].

As discussed before, it is essential to select an appropriate IMU to meet the high-end requirements of this project and to be able to identify geometric errors in the scale of micrometers. Due to this it is necessary to further investigate the possible IMU errors, their effects and how to minimize them.

The market of inertial measurement sensors is very wide both in terms of performance and price. Depending on similar performance classes they can be classified in

different grades: Navigation or Marine grade, Tactical grade, Industrial grade and Automotive or Consumer grade. Table 3.1 gathers typical performance parameters for different IMU grades.

IMU Grade	Accel Bias (mg)	Accel noise ($mg/\sqrt{Hz}$)	Gyro Bias ($^\circ/h$)	Gyro Noise ($^\circ/h/\sqrt{Hz}$)
Navigation	0.01-0.1	0.005-0.01	0.001-0.01	0.002-0.005
Tactical	0.1-1	0.2-0.4	0.01-1	0.2-0.5
Industrial	1-10	N/A	1-100	N/A
Automotive	>10	N/A	>100	N/A

Table 3.1. Typical Accelerometer and Gyros Bias and Noise for Different IMU Grades. Table modified from previous works [Groves, 2013, Gebre-Egziabher, 2004].

In most of the low cost IMUs, the accelerometers and gyroscopes are manufactured as MEMS. These are made of relatively small structures, which are manufactured using silicon or quartz. They have many advantages, such as; small size, low power consumption, low weight, robust construction, short start-up time, high reliability and are appropriate to operate in hostile environments.

Liu et al. discuss that it is complicated to make MEMS-based accelerometers based on common technologies (piezoresistive, piezoelectric and capacitive) with enough sensitivity and resolution for applications that require accuracies below the micro-g. Instead they suggest a micro-machined accelerometer which takes advantage of a tunneling transducer to feature micro-g resolution [Liu et al., 1998].

However, MEMS sensors can still be used in applications where accuracy in the range of micrometers is needed for short periods of time, as it can be seen in the works of Spiewak et al., who have conducted several experiments to characterize, identify and attenuate errors for different accelerometers [Spiewak, 2005, Spiewak et al., 2013].

3.2.2 Methodology

The biggest problem associated with accelerometers and gyros is their big drift, this becomes even more significant if the position is estimated by integrating the acceleration twice or estimating the angular position by integrating the angular speed due to a non zero mean getting amplified, this might result in an increasingly bad measurement with time. Spiewak (2005), argues that they are not suitable for static or quasi-static measurements, so it results complicated to use them as the only measuring system [Spiewak, 2005].

There are two main approaches to deal with this problem, one consist of using one or several secondary sensors to overcome the disadvantages of the IMU by fusioning the different sensors data with the aim of having a combined final measurement that is better than each individual sensor reading. The second approach consist of using the IMU for short periods of time in order to limit the sensor drift.

A lot of research has been done regarding the first method and it has been proven to give satisfactory results for many applications. The idea behind it is to collect data from multiple sensors and to execute an algorithm, such as a Kalman filter or an extended Kalman filter, to fuse the data. For example, Godha and Cannon, (2005) propose the use of a Kalman filter to fuse the data of an IMU with a GPS [Godha and Cannon, 2005]. This method allows to improve the GPS accuracy while the IMU drift problem is reduced because the Kalman filter is used to estimate its bias. So it boosts the advantages and lowers the disadvantages of both sensors allowing the final setup to achieve an accuracy in the range of centimeters.

However, when it is not possible to utilize a secondary sensor or a higher accuracy is desired, others methods might be used. As discussed before, Vogl et al. developed a sensor based method to estimate the state of linear axes machine tool degradation by collecting data from three accelerometers, a tri-axial gyroscope and two inclinometers [Vogl et al., 2015]. Vogl et al. use the gyro signal to get angular errors by integrating them once and accelerometers to get translational errors by integrating them twice. In addition, inclinometers were used to get low frequency angular errors.

Vogl et al. performed several fixed-cycle tests where one axis is moved its entire travel length at different constant speeds (0.02 m/s for slow speed, 0.1 m/s for moderate speed and 1 m/s for fast speed). This is done to compensate for the different bandwith and noise properties of the sensors used and to be able to identify different spatial frequencies. In addition, this method takes advantage of the enhanced signal-to-noise and lower sensor drift at faster speeds, while taking advantage of the detection of higher spatial frequencies at slower speeds. After this data is collected, it is filtered, combined together and averaged to get the original geometric errors. The whole procedure can be seen in Fig. 3.1.

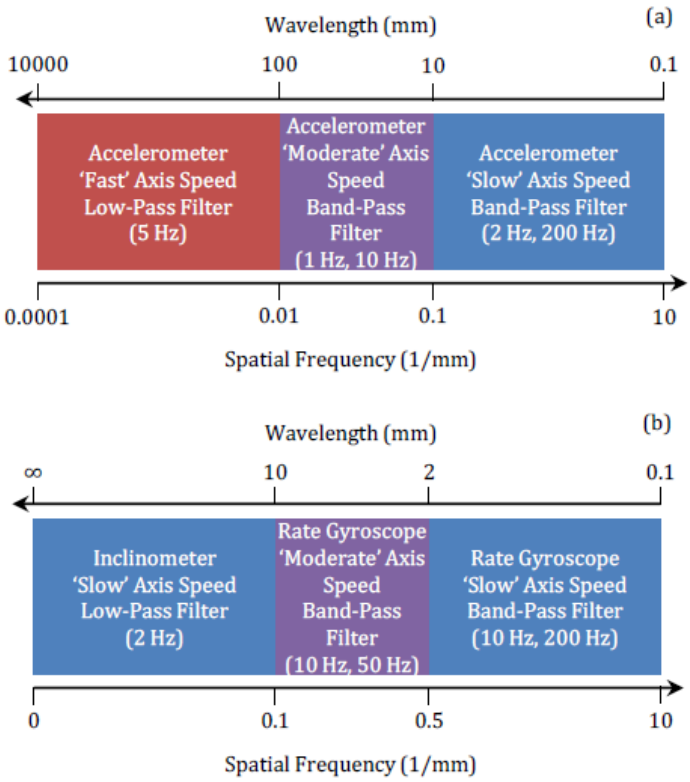


Figure 3.1. Fixed-cycle test data analysis for (a) straightness errors and (b) angular errors ([Vogl et al., 2015]).

The results from Vogl et al. show that the method is acceptable as it can achieve TURs of the given constraint of 4:1. However, the methodology is limited, the uncertainties of the errors are both due to the methodology proposed and the sensors noise. It is concluded that the limited bandwidth of the accelerometers (at low frequencies) lead to uncertainties for the straightness errors whereas the gyros noise is responsible of the angular error uncertainties.

Another approach is discussed by Spiewak, who argues that micrometer accuracy of IDS is possible during short periods of time (less than 1 second) employing commercially available accelerometers [Spiewak, 2005]. Spiewak, studied the impact of static phenomena and he considers that because of measuring by IMUs involves integration, the accuracy depends heavily on the low frequency distortions. In order to compensate for it, the static sensor noise error has been approximated by the following equation:

$$\epsilon = a_0 + a_1 t^2 + a_2 sin(2\pi t/T_m + a_3) \tag{3.1}$$

where $a_i, i = 1, 2, 3$ and Tm are empirical constants and the noise is almost sinusoidal.

In addition to this, the dynamic phenomena constitutes another important source of errors, which is mainly caused by non-linearities of the transducer and imperfections in the manufacturing. To deal with this, Spiewak, (2005) suggest to use multiple sensors and to average their output signals, attenuate distortions through different models, isolate sensors from non-essential vibrations and to use an optimal sensor design procedure. This paper also proposes a method that involves minimization of a target function which is the difference between the signals sampled and predicted by an appropriate model.

Finally, other research show that introducing sensor redundancy in the system can improve the overall accuracy since sensor failure can be identified and multiple data collection can allow to provide enough valuable information. The results obtained from Clausen et al., (2015) show that multiple low-grade IMUs can achieve a better performance than a single IMU [Clausen et al., 2015].

3.3 Geometric Error Compensation

In a three-axial CNC–machine there are 21 geometric error sources in every direction that directly affect the accuracy performance [Caballero-Ruiz et al., 2007]. The geometric errors are mostly derived from wear in the machine tools but also from built in inaccuracies and kinematic errors [Ramesh et al., 2000]. From interferometer measurements a mathematical model of the errors can be designed to act as a compensation model for these geometrical errors. The resulting error estimation can be achieved by back-propagation [Rumelhart et al., 1988]. Research regarding this subject has been conducted and an error compensation model has been successfully been implemented into a CNC-machine. The model modified the output position command of the CNC-machine by calculating the difference between the nominal position and the predicted error. The CNC-machine was run in a backward and forward direction with a error compensation model, which was later compared to an uncompensated model. The results showed a decrease in positional geometric error by 40 µm [Raksiri and Parnichkun, 2004].

The time delay introduced to the system by using error compensation has proved to be an issue. The overall volumetric accuracy is reduced if the correction signal is not updated fast enough. To work around this problem a real time computation of the compensation algorithms with a neglectible time delay is required. [Longstaff et al., 2005] used a Siemens 840D controller [Siemens, 2015] to basically run the compensation algorithms in real time by applying the compensated output during the spare cycle time from the NC processor. This compensated controller was compared to an uncompensated in a CNC-machine where the input the system was

the geometric errors collected by an laser interferometer, the results showed a 97% reduction in geometric errors [Longstaff et al., 2005].

4 Proposed Solutions

This chapter describes the planned solution to the discussed problem.

4.1 Construction of Test Bench

A two axis feed drive system with multiple ways of inducing different artificial repeatable errors will be constructed. The requirements for the test bench can be seen in Appendix A. The parts used for the construction can be seen in Table B. Motors will be controlled by a PID controller with a trajectory planner. The controller shall receive a two-dimensional position command, calculate an optimal trajectory path and execute it.

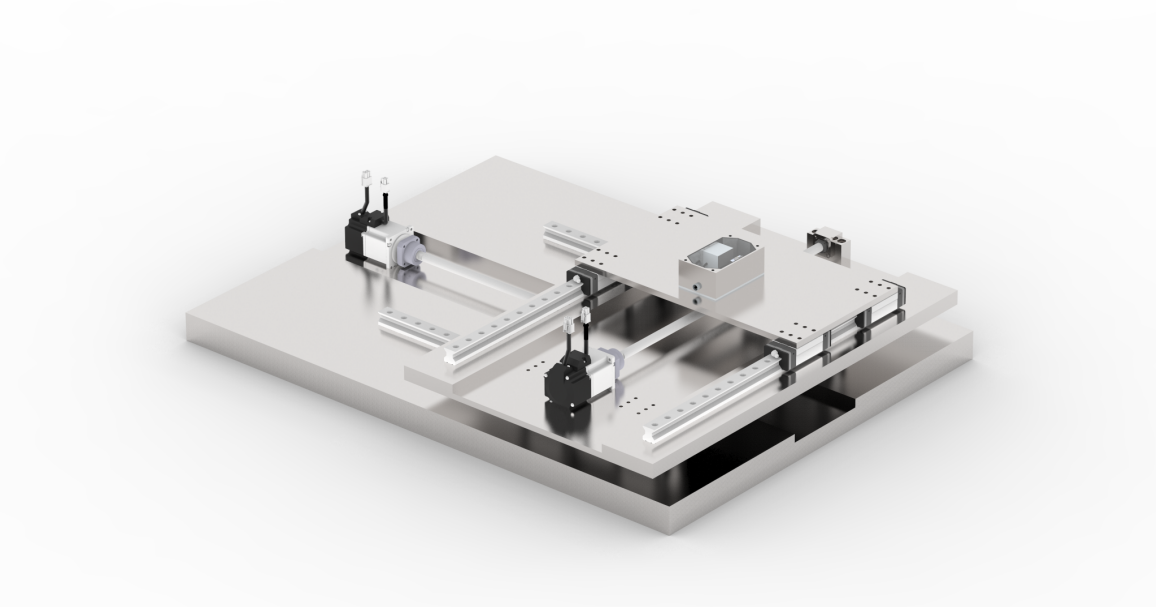


Figure 4.1. Test bench

4.1.1 Verification Tests

The test bench must be constructed with minimal misalignment. A laser measurement system will be used to verify this. A reference will be placed upon the test bench carriage, one axis will be locked and the other will be moved with small incremental steps from one end of its path to the other. In every stop the reference position will be measured. The optical laser can determine position for three axis as well as the pitch and jaw during the measurement. Feedback regarding the straightness of the measurement will be used to calibrate and rearrange the guide rails to be within specifications from Appendix A. The locked axis will be moved 0.5 cm and the test will be repeated until the whole 2D-plane has been covered. The test bench must fulfill the requirement of accuracy to be able to validate the measured results. This test will be done one axis at a time. The controller is given a known position command and shall execute it. The test bench carriage reference shall position itself with the accuracy specified in the requirements. The actual position will be measured with a laser system.

4.2 Construction of IMU

An IMU will be constructed from a triaxial gyroscope, a triaxial accelerometer and a diaxial inclinometer as specified in Table B. The components will be mounted in a custom made box that in turn will be mounted onto the test bench carriage (see Fig. 4.2). The signals will be sent to the controller and a monitoring computer and interpreted there.

4.2.1 Specifications

The IMU's requires very low bias noise but only a small operating range. Recommended components [Vogl, 2016]:

Gyroscope

- Range of $\pm 100°/sec$.
- Bias stability $4°/hour$ 1σ.
- Bias due to temperature $< 0.05°/sec$ 1σ.

Accelerometer

- Range of $\pm 2g$.
- Sensor resolution $0.001°/sec$.
- Frequency response 0-400 Hz $\pm 1dB$.

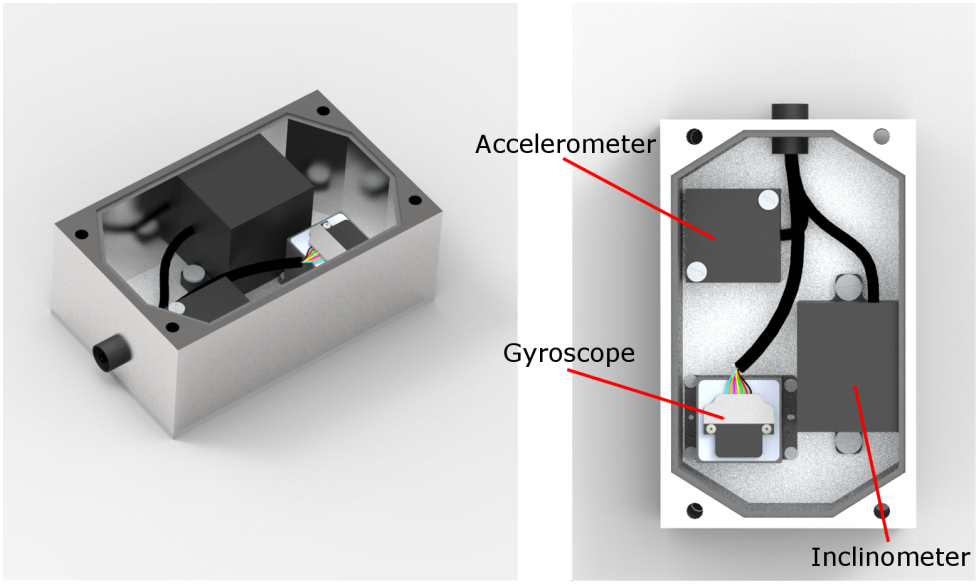


Figure 4.2. The IMU

4.2.2 Verification Tests

Accuracy

The IMU must be able to measure deviations as specified by the requirements. This test assumed the existence of a reference measurement done by with a laser measurement system which will be done prior. The mounted IMU will perform the same movement as the reference measurement and shall report a similar enough result.

4.3 Error Inducement Concepts

Real world errors can be anything from microscopic chipping creating high frequency errors to the entire guideway being bent from for example overload creating low frequency errors. It will be necessary to attempt to simulate the full range of these errors as realistically as possible. A number of possible concepts have been developed for the chosen test bed configuration. Some of these concepts could be combined with each other whereas others will require a separate test bed set up, the important part is to have both high, medium and low frequencies represented.

4.3.1 Concept #1 - Indenting the Rail

By indenting or scratching the surface of the guideway where the rollers travel it's possible to simulate errors. Depending on the size of these indentations, medium-high frequency to very high frequency errors could be generated. Since the guideways would be physically damaged, the errors would be similar to the real errors that occur in machine tools due to wear on the guideways. The indentations would be made by creating around ten slits with the same width in an array with an EDM. By dividing the guideway in to multiple sections and have a different error array in every section it is possible to test multiple errors in one run.

Figure 4.3. Guideway with a slit array

Advantages:

- Realistic simulation errors.
- Can simulate a range of errors in a single run.
- Can simulate high frequency errors.
- The position and the size of the errors can be specifically chosen.

Disadvantages:

- Would require an extra rail with the slit arrays.
- Changing between the modified and an original rail is necessary to try different errors and could lead to non-repeatable behaviours.

4.3.2 Concept #2 - Deforming the Rollers

In order for the trucks to easily travel along the guideway they have rollers to lower the friction. By deforming these rollers it's possible to induce errors, this would be done by either removing material from the rollers in an abrasive way or use and EDM to cut slits in rollers. The abrasive method would result in an errors that's close to real wear but it would be hard to know exactly the size of the errors. By choosing to deform combinations of several rollers and alternating the size of the deformation different amount of wear could be simulated. The downside is that it would be very difficult to exchange the rollers in a manner that still provides repeatability.

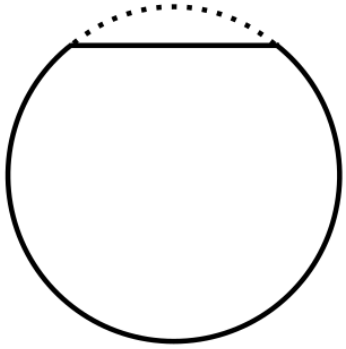


Figure 4.4. How the rollers could be deformed

Advantages:

- Able to induce high frequency errors.
- Can be used on the same guideway as several other error inducement concepts.

Disadvantages:

- May be a tedious and time consuming process of removing and deforming rollers as well as difficult to replace the rollers.
- Errors might appear seemingly at random and won't be the same from run to run.
- Removing the ball slide to replace rollers will change the preload of the entire system, making comparisons to previous runs harder.

4.3.3 Concept #3 - Linear Push

By replacing some of the screws that are securing the guideway to the base plate with piezo actuators, or find other ways to enable a linear force to push various parts of the guideway, small deformations will appear in the rail. By varying the pressure and the position of the actuators different low frequency errors can be induced. As long as the rail does not plastically deform the errors could quickly be changed and the set up could be used in combination with other error inducement concepts. Securing the actuators to the guideway and base plate will however introduce some design problems.

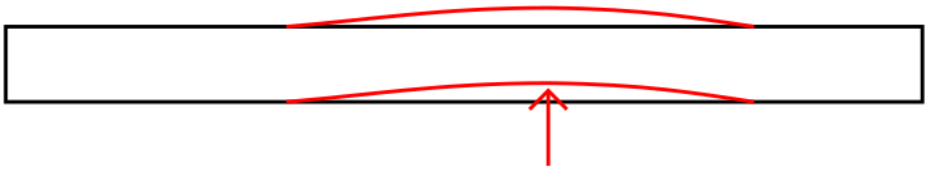


Figure 4.5. Bending of the guideway

Advantages:

- Very controllable and errors are easily repeated.
- The induced errors can be manipulated between runs with little effort.

Disadvantages:

- Extra steps must be taken during the design phase to allow for these linear forces.
- Constructing this will be time consuming.
- The forces needed to bend the rail may be too large to realistically handle.

4.3.4 Concept #4 - Software

By making small changes to the signals that is sent from the controller to the motors it's possible to generate errors by moving the platform in the Y-axis while measuring the straightens along X-axis. The errors simulated could be in any frequency region, even the high frequency region depending on the control and motor performance. This method would require fast and precise control in order to simulate the exact same errors over several runs.

Advantages:

- No additional hardware required, easy and cheap to implement.
- Could induce both high frequency and low frequency errors.
- Quite controllable and can easily be adjusted between runs without affecting preloads or other parts of the system.

Disadvantages:

- Requires a precise and fast control system.
- It may be hard to validate it as a realistic error.

4.3.5 Concept #5 - Linear Force Pulses

If the actuators from concept #3 is used it is possible to induce low frequency errors by bending the rail. However if the actuators are activated in pulses it is possible to induce higher frequency errors due to the vibrations in the guideway.

Advantages:

- Able to induce high frequency errors without having to damage any parts.
- Is easy to alter the frequency to get other types of error
- Can be combined with other types of error inducements.
- Comes as a bonus to the concept #3.

Disadvantages:

- Hard to measure a reference signal without the aid of dynamic measuring methods.
- It will be difficult to sync the phase of the pulses with the trajectory of the ball slide between repeated runs.

5 Conclusions and Status Report

The preliminary research in the field of error inducement and error detection through the use of a sensor-based method concludes that the project description is reasonable and does not need any large alterations. All requirements stated in Appendix A are possible to fulfill. Multiple realistic concepts on how to induce the errors have been derived in Chapter 4.

Discussions with the Swedish supplier Acumo have been held, and the company will provide the guideways, ball screws and motor. By choosing one supplier for the test bench components, the quality can be assured, compatibility guaranteed and the costs kept down. The IMU components have been recommended by G. Vogl. A more complete parts list for both the test bed and the IMU that will fulfill the requirements can be seen in Appendix B. Most of the components will be ordered before summer, and assembly will start in beginning of September, the installation of the IMU will be done in parallel.

6 Future Research

The future usage of the test bench will be in research. Some points may be:

- Developing and calibrating new sensors in a controlled environment.
- Developing and calibrating self-diagnosis techniques.
- Measuring the component and location errors of each axis with different types of instruments and building a geometric model from the results.
- Thorough study of the interpolation errors when using different methodologies.
- Building a model which connects direct and indirect measurements.
- Study the errors in position control loops.
- Perform uncertainty analysis of muli-axes feed drive systems.
- Analyze the energy consumption of feed drive systems.
- Study the effect of different system preloads.
- Study the thermal behaviour of the feed drive system.

Bibliography

[Ahmad et al., 2013] Ahmad, N., Ghazilla, R. A. R., Khairi, N. M., and Kasi, V. (2013). Reviews on various inertial measurement unit (imu) sensor applications. *International Journal of Signal Processing Systems*, 1(2):256–262.

[Altintas et al., 2011] Altintas, Y., Verl, A., Brecher, C., Uriarte, L., and Pritschow, G. (2011). Machine tool feed drives. *CIRP Annals-Manufacturing Technology*, 60(2):779–796.

[Biehl et al., 2012] Biehl, S., Staufenbiel, S., Recknagel, S., Denkena, B., and Bertram, O. (2012). Thin film sensors for condition monitoring in ball screw drives. In *1st Joint International Symposium on System-Integrated Intelligence 2012: New Challenges for Product and Production Engineering*, pages 59–61.

[Brandenburg et al., 2000] Brandenburg, G., Bruckl, S., Dormann, J., Heinzl, J., and Schmidt, C. (2000). Comparative investigation of rotary and linear motor feed drive systems for high precision machine tools. In *Advanced Motion Control, 2000. Proceedings. 6th International Workshop on*, pages 384–389.

[Caballero-Ruiz et al., 2007] Caballero-Ruiz, A., Ruiz-Huerta, L., Baidyk, T., and Kussul, E. (2007). Geometrical error analysis of a cnc micro-machine tool. *Mechatronics*, 17(4):231–243.

[Cheng et al., 2011] Cheng, Q.-M., Cheng, Y.-M., Wang, Y.-F., and Wang, M.-M. (2011). Overview of control strategies for ac motor. *Power System Protection and Control*, 39(9):145–154.

[Ciuti et al., 2015] Ciuti, G., Ricotti, L., Menciassi, A., and Dario, P. (2015). Mems sensor technologies for human centred applications in healthcare, physical activities, safety and environmental sensing: A review on research activities in italy. *Sensors*, 15(3):6441–6468.

[Clausen et al., 2015] Clausen, P., Skaloud, J., Gilliéron, P.-Y., Merminod, B., Perakis, H., Gikas, V., and Spyropoulou, I. (2015). Position accuracy with redundant mems imu for road applications. In *Proceedings of the ENC-GNSS 2015*, number EPFL-CONF-207585.

[Feng and Pan, 2012] Feng, G.-H. and Pan, Y.-L. (2012). Investigation of ball screw preload variation based on dynamic modeling of a preload adjustable feed-drive system and spectrum analysis of ball-nuts sensed vibration signals. *International Journal of Machine Tools and Manufacture*, 52(1):85–96.

[Gamazo-Real and Gómez-Gil, 2010] Gamazo-Real, V.-S. and Gómez-Gil (2010). Position and speed control of brushless dc motors using sensorless techniques and application trends. *PMC Web*.

[Garinei and Marsili, 2012] Garinei, A. and Marsili, R. (2012). A new diagnostic technique for ball screw actuators. *Measurement*, 45(5):819–828.

[Gebre-Egziabher, 2004] Gebre-Egziabher, D. (2004). *Design and performance analysis of a low-cost aided dead reckoning navigator*. PhD thesis, Citeseer.

[Godha and Cannon, 2005] Godha, S. and Cannon, M. (2005). Integration of dgps with a low cost mems-based inertial measurement unit (imu) for land vehicle navigation application. In *Proceedings of the 18th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS05)*, pages 333–345.

[Groves, 2013] Groves, P. D. (2013). *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, Chichester, 2nd edition.

[HIWIN, 2013] HIWIN (2013). Linear guideway technical information.

[Hu et al., 2013] Hu, Y. B., Xie, J., and Hu, L. B. (2013). Linear motor direct drive key technology problems and the solution countermeasures. In *Applied Mechanics and Materials*, volume 411, pages 3052–3055. Trans Tech Publ.

[Kim and Jeon, 2011] Kim, D. and Jeon, D. (2011). Fuzzy-logic control of cutting forces in cnc milling processes using motor currents as indirect force sensors. *Precision Engineering*, 35(1):143–152.

[Koren, 1980] Koren, Y. (1980). Cross-coupled biaxial computer control for manufacturing systems. *Journal of Dynamic Systems, Measurement, and Control*, 102(4):265–272.

[Koren and Lo, 1992] Koren, Y. and Lo, C. (1992). Advanced controllers for feed drives. *CIRP Annals-Manufacturing Technology*, 41(2):689–698.

[Krishnan, 2009] Krishnan, R. (2009). *Permanent magnet synchronous and brushless DC motor drives*. CRC press.

[Kumar et al., 2010] Kumar, M., Dotson, K., and Melkote, S. N. (2010). An experimental technique to detect tool–workpiece contact in micromilling. *Journal of Manufacturing Processes*, 12(2):99 – 105. Available from: `http://www.sciencedirect.com/science/article/pii/S1526612510000332`.

[Lee and Yang, 2005] Lee, J. and Yang, S. (2005). Measurement of geometric errors in a miniaturized machine tool using capacitance sensors. *Journal of materials processing technology*, 164:1402–1409.

[Lee et al., 2016] Lee, J.-C., Lee, H.-H., and Yang, S.-H. (2016). Total measurement of geometric errors of a three-axis machine tool by developing a hybrid technique. *International Journal of Precision Engineering and Manufacturing*, 17(4):427–432. Available from: `http://dx.doi.org/10.1007/s12541-016-0053-5`.

[Li et al., 2014] Li, Y., Wang, X., Lin, J., and Shi, S. (2014). A wavelet bicoherence-based quadratic nonlinearity feature for translational axis condition monitoring. *Sensors*, 14(2):2071–2088.

[Liao and Lee, 2009] Liao, L. and Lee, J. (2009). A novel method for machine performance degradation assessment based on fixed cycle features test. *Journal of Sound and Vibration*, 326(3):894–908.

[Liu et al., 1998] Liu, C.-H., Barzilai, A. M., Reynolds, J. K., Partridge, A., Kenny, T. W., Grade, J. D., and Rockstad, H. K. (1998). Characterization of a high-sensitivity micromachined tunneling accelerometer with micro-g resolution. *Microelectromechanical Systems, Journal of*, 7(2):235–244.

[Longstaff et al., 2005] Longstaff, A. P., Fletcher, S., and Myers, A. (2005). Volumetric compensation for precision manufacture through a standard cnc controller. In *20th Annual Meeting of the American Society for Precision Engineering*.

[NTN Corporation, 2001] NTN Corporation (2001). Ball and roller bearings.

[Ohta et al., 2010] Ohta, H., Matsuura, K., Kato, S., and Igarashi, Y. (2010). Vibration and acoustic emission of linear-guideway type recirculating ball bearings with a millimeter-sized artificial defect in the carriage. *Journal of Tribology*, 132(1):011101.

[Olarra et al., 2009] Olarra, A., Uriarte, L., and de Argandoña, I. R. (2009). New developments in drives and tables. In *Machine Tools for High Performance Machining*, pages 129–158. Springer.

[Plapper and Weck, 2001] Plapper, V. and Weck, M. (2001). Sensorless machine tool condition monitoring based on open ncs. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 3, pages 3104–3108. IEEE.

[Raksiri and Parnichkun, 2004] Raksiri, C. and Parnichkun, M. (2004). Geometric and force errors compensation in a 3-axis cnc milling machine. *International Journal of Machine Tools and Manufacture*, 44(12):1283–1291.

[Ramesh et al., 2000] Ramesh, R., Mannan, M., and Poo, A. (2000). Error compensation in machine tools—a review: Part i: geometric, cutting-force induced

and fixture-dependent errors. *International Journal of Machine Tools and Manufacture*, 40(9):1235–1256.

[Rehorn et al., 2005] Rehorn, G. A., Jiang, J., and Orban, E. P. (2005). State-of-the-art methods and results in tool condition monitoring: a review. *The International Journal of Advanced Manufacturing Technology*, 26(7):693–710. Available from: `http://dx.doi.org/10.1007/s00170-004-2038-2`.

[Rumelhart et al., 1988] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1988). Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1.

[Schaeffler Group, 2007] Schaeffler Group (2007). Linear Recirculating Roller Bearing and Guideway Assemblies.

[Schaeffler Technologies, 2015] Schaeffler Technologies (2015). Hydrostatic Compact Guidance System.

[Schwenke et al., 2008] Schwenke, H., Knapp, W., Haitjema, H., Weckenmann, A., Schmitt, R., and Delbressine, F. (2008). Geometric error measurement and compensation of machines—an update. *CIRP Annals-Manufacturing Technology*, 57(2):660–675.

[Shaw and Su, 2011] Shaw, D. and Su, W. (2011). Study of stiffness of a linear guideway by fea and experiment. *Structural Longevity*, 5(3):129–138.

[Shih et al., 2002] Shih, Y.-T., Chen, C.-S., and Lee, A.-C. (2002). A novel cross-coupling control design for bi-axis motion. *International Journal of machine tools and manufacture*, 42(14):1539–1548.

[Siemens, 2015] Siemens (2015). SINUMERIK 840D sl .

[Spiewak, 2005] Spiewak, S. A. (2005). Versatile inertial displacement sensor for planar motion. In *MEMS, NANO and Smart Systems, 2005. Proceedings. 2005 International Conference on*, pages 463–466. IEEE.

[Spiewak et al., 2013] Spiewak, S. A., Ludwick, S. J., and Hauer, G. (2013). A test setup for evaluation of harmonic distortions in precision inertial sensors. *Journal of Manufacturing Science and Engineering*, 135(2):021015.

[THK, 2007] THK (2007). Features of the Cross Roller Guide/Ball Guide.

[Tomizuka, 1987] Tomizuka, M. (1987). Zero phase error tracking algorithm for digital control. *Journal of Dynamic Systems, Measurement, and Control*, 109(1):65–68.

[Uhlmann et al., 2008] Uhlmann, E., Geisert, C., and Hohwieler, E. (2008). Monitoring of slowly progressing deterioration of computer numerical control machine

axes. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 222(10):1213–1219.

[Verl et al., 2009] Verl, A., Heisel, U., Walther, M., and Maier, D. (2009). Sensorless automated condition monitoring for the control of the predictive maintenance of machine tools. *CIRP Annals-Manufacturing Technology*, 58(1):375–378.

[Vogl, 2016] Vogl, G. W., editor (2016). *Meeting regarding "Diagnostics for geometric performance of machine tool linear axes"*. Greg W. Vogl.

[Vogl et al., 2015] Vogl, G. W., Weiss, B. A., and Donmez, M. A. (2015). A sensor-based method for diagnostics of machine tool linear axes.

[Wang and Xi, 2008] Wang, L. and Xi, F. (2008). *Smart devices and machines for advanced manufacturing*. Springer Science & Business Media.

[Wilson, 2005] Wilson, J. S. (2005). *Sensor Technology Handbook*. Elsevier, 1st edition.

[Won et al., 2009] Won, S.-H. P., Golnaraghi, F., and Melek, W. W. (2009). A fastening tool tracking system using an imu and a position sensor with kalman filters and a fuzzy expert system. *Industrial Electronics, IEEE Transactions on*, 56(5):1782–1792.

[Xu et al., 2016] Xu, X., Tao, T., Jiang, G., Huang, X., Liu, R., and Liu, L. (2016). Monitoring and source tracing of machining error based on built-in sensor signal. *Procedia CIRP*, 41:729–734.

[Yeh et al., 1997] Yeh, Z.-M., Tarng, Y., and Lin, Y. (1997). Cross-coupled fuzzy logic control for multiaxis machine tools. *Mechatronics*, 7(8):663–681.

A Requirements

The following requirements are set requirements from the stakeholder in order to validate the functionality of the test bench and the IMU that have been investigate during the research phase of the project. The choice of components and design of the system was made from this premise.

Test Bench and Electrical System Requirements

- The minimal stiffness of each axis will be more or equal to 400 $[\frac{N}{\mu m}]$ in the axis direction.
- The axial positioning accuracy shall be $\pm$ 5 μm.
- The test bench shall be able to induce controlled errors in position and straightness on the controlled axis of a magnitude of at least 20 μm.

IMU Requirements

- The IMU shall measure the position error with an accuracy of $\pm$5 μm.
- The IMU shall detect a straightness error of at least 20 μm with a test uncertainty level more than 4:1.
- The IMU should not average more than 10 measurements to be within $\pm$ 5 μm of the interferometer measurement.

Software Requirements

- The positioning accuracy shall be $\pm$ 5 μm.
- The software should have a user friendly GUI for advanced users. The GUI should have modifiable control parameters in the interface.
- The system should have trajectory controlled tool paths.

Overall System Requirements

- The system shall have a safety distance from where no human can be harmed from the workpiece.

Soft Requirements

These are listed as "soft requirements" since they does not need to be met in order to fulfill the functionality of the system, but serve as an indicator of the construction design.

- The tabletop of the bench will be 250x250 $[mm]$.
- The maximum stroke will be 300x300 $[mm]$.
- The maximum feed force on the workpiece will be 2000 $[N]$.
- The maximum speed of the workpiece will be 30 $[\frac{m}{s}]$.
- The maximum mass of the workpiece will be 100 $[kg]$.

B Parts List

Not all parts are ordered at the time of writing and it might be subject to changes.

Component	Amount	Comments
Servodriver 400W LIQI	2	One received free as a gift from Acumo.
Servomotor 400W keyshaft LIQI	2	One received free as a gift from Acumo.
Sensor Cable Panasonic, 2m, 50-750W.	2	One received free as a gift from Acumo.
Motorcable LIQI, 2m, 50-750W	2	One received free as a gift from Acumo.
Ball screw lead 5mm length 400mm C5	2	
NSK fixed side support unit	2	
NSK fixed side support unit	2	
Carriage 25m	8	
Bellows Couplings G6	2	

Table B.1. Estimated parts list for the test bench

Component	Amount	Comments
Accelerometer 4630A-002-XXX	1	
Gyro G200D-100-300	1	
Inclinometer SX46125	1	

Table B.2. Estimated parts list for the IMU