SimMechanics Model of a NanoCMM

R.Cau

Student ID 0536104

Dr. Ir. P. C. J. N. Rosielle

Eindhoven, 01 June 2007
# Table of contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>5</td>
</tr>
<tr>
<td><strong>Chapter 1</strong></td>
<td></td>
</tr>
<tr>
<td>Design (J. K. van Seggelen)</td>
<td>7</td>
</tr>
<tr>
<td><strong>Chapter 2</strong></td>
<td></td>
</tr>
<tr>
<td>SimMechanics Model</td>
<td>9</td>
</tr>
<tr>
<td><strong>Chapter 3</strong></td>
<td></td>
</tr>
<tr>
<td>Output Analysis</td>
<td>21</td>
</tr>
<tr>
<td><strong>Chapter 4</strong></td>
<td></td>
</tr>
<tr>
<td>Model Results</td>
<td>25</td>
</tr>
<tr>
<td><strong>Conclusion</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>35</td>
</tr>
<tr>
<td><strong>Literature</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>37</td>
</tr>
<tr>
<td><strong>Appendix A</strong></td>
<td></td>
</tr>
<tr>
<td>M-File</td>
<td>39</td>
</tr>
<tr>
<td><strong>Appendix B</strong></td>
<td></td>
</tr>
<tr>
<td>SimMechanics Model</td>
<td>45</td>
</tr>
<tr>
<td><strong>Appendix C</strong></td>
<td></td>
</tr>
<tr>
<td>Signal Outputs</td>
<td>47</td>
</tr>
</tbody>
</table>
Introduction

A coordinate measuring machine has been developed to operate measurement procedures on objects that require high uncertainty. The nanoCMM, as it is called, has a volumetric uncertainty of 25 nm in a 50 x 50 x 4 mm measuring volume. This makes it one of the top notch appliances in high precision measuring.

In this report, the design of the nanoCMM is reviewed to investigate whether further improvements in terms of dynamics are possible. For this, a computer model was made using SimMechanics, a signal processing toolbox. This model can predict the dynamic behavior of the measuring machine, also when certain design parameters are changed.

The first chapter of this report, introduction by J. K. van Seggelen, will deal with the actual design of the nanoCMM, describing in short its construction, dynamic properties and its capabilities.

Secondly, the SimMechanics model is described, in which block representations of bearings, actuators and measuring means are made clear. Also, the configuration set up of the model is brought out, and the visualization option is explained.

Next, three analysis methods that can be applied after simulating are discussed. These analysis methods can all help processing the results of model simulations in a different way, and each can therefore be chosen for a specific purpose.

Then, some simulation results will be discussed and while changing some of the machine properties the dynamic behavior of the model is investigated. The results will explain the shape of the first couple of eigenmodes, and why they occur. These results will also give an idea about further capabilities of the design.
To measure dimensions and shape of complex three dimensional products with low uncertainty, Coordinate Measuring Machines (CMMs) are adequate instruments due to their universal applicability, easy measurement set-up and measuring flexibility.

To keep on track with the trend towards miniaturization, the industry requires a fast 3D CMM for measuring small products in array with nanometer uncertainty. Although there are commercial CMMs available with sub micrometer and even claimed nanometer measurement uncertainty, none of them allows measuring with high speed in 3D as a result of their huge moving mass in the horizontal plane and/or vertical direction.

To meet the above industry requirements, a 3D CMM with a low moving mass in x-, y- and z-direction and a volumetric uncertainty of 25 nm in a 50 x 50 x 4 mm measuring volume was developed. The design is a completely aluminum CMM, based on a horizontal air bearing system without Abbe errors, an elastically guided vertical axis, measurement by 1 nm resolution optical linear encoders and single phase Lorentz actuator drives.

The horizontal air bearing system consists of two scale beams (each carrying a reflective scale) and two intermediate bodies (each carrying an optical measuring head), all equipped with separate stress frames which prevent distortion of the optical linear encoders by preload forces. The optical encoders are in line with the lengths being measured, which eliminates Abbe errors in the horizontal plane (see figure 1.1). Further, it eliminates straightness errors of the scale beams according to the Bryan principle.

The vertical stroke of the CMM is provided by an elastically straight-guiding mechanism.

Figure 1.1: Schematic view of the nanoCMM
The CMM has a size of 450 x 450 x 200 mm, a moving mass of 8.5 kg in the horizontal plane and a moving mass of 100 g in vertical direction. Compared to today’s high accuracy CMMs, this CMM has a 4.5 to 7 times lower moving mass in the horizontal plane and an up to 300 times lower moving mass in vertical direction, which reduces power required for measuring with high speed in 3D and thereby avoids thermo-mechanical effects.

The CMM is furthermore equipped with a PID-controller, and after tuning, the positioning error in z-direction was ± 2 nm and a bandwidth of 200 Hz was realized for the elastically guided vertical axis. The positioning errors in x- and y-direction were ± 4 nm respectively ± 3 nm. The open-loop frequency response functions of the horizontal air bearing system showed a bandwidth of 35 Hz in x-direction and of 50 Hz in y-direction. This difference can be explained by the asymmetric configuration of the horizontal air bearing system for reasons of kinematic design.

The standard uncertainty of a volumetric length measurement, due to geometric errors, was estimated to be about 13 nm in best case and 38 nm in worst case. These values apply to 3D objects with dimensions of the measuring volume. For the intended small products, the expanded uncertainty in nanometers is about 11 + 0.3L in best case and 19 + 1.2L in worst case, with L the measured length in mm.

This machine will be translated to a computer model using SimMechanics. The model can give further insight in the dynamic behavior of the machine, and might show several points of interest, for instance where improvements are possible.
Chapter 2 – SimMechanics Model

A SimMechanics model has been made to predict the behavior of the machine when certain parameters of the system are changed. Though the model should represent the true behavior of the machine, it is a simplification in such a way that it can be implemented in the Simulink toolbox.

SimMechanics is a toolbox used by Simulink that is run by the MATLAB engine. SimMechanics uses a number of blocks that can be linked to each other. Each block can be given its own parameters and number of input/output ports. In fact, SimMechanics is a signal processing toolbox that can calculate the behavior of a system when an input signal is put into it. Since these calculations are based upon the common laws of linear dynamics, the model can only describe the linearized behavior of the system. Outcomes of the model should be interpreted qualitatively, rather than in a quantitative way.

This chapter will discuss the set-up of the model, along with its parameters used.

Below, some of the most important SimMechanics blocks are listed.

Ground
A “Ground” block represents the fixed world. Each body block should be connected to at least one “Ground” block. One of the ground blocks can be used to set the machine environment.

Machine Environment
A “Machine Environment” block can be connected to one “Ground” block to define the mechanical simulation environment for the machine to which the block is connected: gravity, dimensionality, analysis mode, constraint solver type, tolerances, linearization, and visualization.

Bodies
All parts of the nanoCMM are represented in the model by only a small number of simplified bodies. A body can therefore be built up out of various parts that in reality have their own stiffness and internal degrees of freedom. Instead this model describes the system as a connection of lumped masses, each body therefore only having a prescribed mass and an inertia tensor. A body can have various output ports that represent their connection to another body, or to the ground. Stiffnesses and degrees of freedom of a body can only be modeled by placing a linear spring or a joint in between two “Body” blocks or in between a “Body” block and a “Ground” block.

Joints
In SimMechanics, there are many different kinds of “Joint” blocks, differing in degrees of freedom that can be granted. Any kind of “Joint” block has one input port (base) and one output port (follower) and must be connected to a “Body” block or a “Ground” block. A joint can prescribe the degrees of freedom between the base and the follower, or can be actuated to perform any rotation or translation of the follower relative to the base. Also, any movement originated by a “Joint” block can be registered by connecting a “Joint Sensor” block, followed by a scope or “To
Workspace” block. In order to do so, ports can be added to connect “Joint Actuator” or “Joint Sensor” blocks.

**Body Spring and Damper**
This block represents a linear spring-damper combination, for which the spring and damper constant, as well as the spring natural length can be prescribed.

**The Model**

The system is modeled mainly using the previously described blocks. The complete model can be found in Appendix B. Because the model may seem very abstract to an outsider, some details will be explained.

**Parameters**
All system values that are necessary to run the model are made parametric. Parameters that are used in the model are stored in an m-file, so that changing any variable of the model only requires changing a part of the m-file. The m-file containing the desired parameter values must be loaded into MATLAB workspace, so that when running a simulation, the model can find the right parameter values.

Intermediate Bodies A and B are indicated by A and B respectively, Scale Beam 1 and 2 are indicated by S1 and S2 respectively. Bearings that are connected to the body are indicated by an index x, y or z in front of the body name, depending on whether the bearing fixes the body in x-, y- or z-direction. For instance, a z-bearing connected to Intermediate Body A will be indicated by zA. Since there are four z-bearings connected to Intermediate Body A, each similar bearing will be given a numerical index behind the name, for instance zA1, zA2, zA3, zA4. Positions of bodies and bearings are according to a Cartesian coordinate system. Distances in x, y and z are indicated by placing the index x, y or z in front of the name. Distances are always measured from the origin of measurement to the centre of gravity of the object. For instance, the y-distance from the origin of measurement to zA1 will be indicated by yzA1, see figure 2.2.

Parameters have also been used for other system values, like spring and damper constants, spring natural lengths, body masses and inertia tensors. These can all be found in the m-file “nanoCMMdata.m” (Appendix A), through which the model can be changed for any desired set up at any time.

**Intermediate Bodies**
In the model, Intermediate Body A and B are represented by “Body” blocks, each having a mass of 4 kg with centre of gravity CGA and CGB, and an inertia tensor $I_{A,B} = \begin{bmatrix} \frac{1}{12}m_{A,B}(w_{A,B}^2 + h_{A,B}^2) & 0 & 0 \\ 0 & \frac{1}{12}m_{A,B}(h_{A,B}^2 + l_{A,B}^2) & 0 \\ 0 & 0 & \frac{1}{12}m_{A,B}(l_{A,B}^2 + w_{A,B}^2) \end{bmatrix}$.
In which $m$ is the mass of the body in kg, $l$ is the length of the body in x-direction, $w$ is the width of the body in y-direction and $h$ is the height of the body in z-direction. The indices A and B are used for intermediate body A and B respectively.

Each of the intermediate bodies is supported by four air bearings in z-direction. For Intermediate Body A these are called $zA1$, $zA2$, $zA3$ and $zA4$. For Intermediate Body B these are called $zB1$, $zB2$, $zB3$ and $zB4$. Furthermore, Intermediate Body A is guided along the base in x-direction by two air bearings, called $yA1$ and $yA2$. Intermediate Body B is guided along the base in y-direction, also by two air bearings, $xB1$ and $xB2$.

Figure 2.1 shows the block representation of Intermediate Body A and its bearings.

![Figure 2.1: Block representation of Intermediate Body A and its bearings](image)

From figure 2.1 can be seen that several other subsystems are connected to Intermediate Body A, apart from the bearings. These involve the y-actuation and y-measurement of Scale Beam 1, which will be discussed later in this chapter. Further, the z-rotation of Intermediate Body A relative to the base is measured by the subsystem “Rotation Measurement”, and another subsystem makes it possible to fix Intermediate Body A rigidly, or for any degree of freedom, to the ground. This is done for analytical reasons.
Intermediate Body B is built up similarly.

Figure 2.2 shows a more graphical representation of both intermediate bodies and their bearings, as they are positioned relative to the origin of measurement. Parametric names of the bearings and centers of gravity, along with their positional values are also shown in this figure.

Mind that, a set of two opposing bearings in the actual machine is modeled as one linear spring and damper system, positioned on their line of symmetry, and having only one spring and damper constant and one spring natural length.

Scale beams
The scale beams are represented by two “Body” blocks that are elastically connected to each other by means of “Linear Spring & Damper” blocks. On the probe side, this connection represents the two plates with elastic hinges. On the back side, the connection represents the aluminum tube and its suspension, see figure 2.3.

The scale beams each have a mass of 1.5 kg with centers of gravity CGS1 and CGS2, and an inertia tensor.
\[
I_{S1,S2} = \begin{bmatrix}
\frac{1}{12} m_{S1,S2} (w_{S1,S2}^2 + h_{S1,S2}^2) & 0 & 0 \\
0 & \frac{1}{12} m_{S1,S2} (h_{S1,S2}^2 + l_{S1,S2}^2) & 0 \\
0 & 0 & \frac{1}{12} m_{S1,S2} (l_{S1,S2}^2 + w_{S1,S2}^2)
\end{bmatrix}
\]

Where \( m \) is the mass of a scale beam, \( l \) is the length of a scale beam in x-direction, \( w \) is the width of a scale beam in y-direction and \( h \) is the height of a scale beam in z-direction. The indices S1 and S2 are used for scale beam 1 and scale beam 2 respectively.

The scale beams are both supported in z-direction by two air bearings. Scale Beam 1 is supported by \( z_{S1} \) and \( z_{S2} \), Scale Beam 2 is supported by \( z_{S3} \) and \( z_{S4} \). Furthermore, Scale Beam 1 is guided along Intermediate Body A in y-direction by one air bearing, \( x_{S1} \). Similarly, Scale Beam 2 is guided along intermediate body B by two air bearings, \( y_{S1} \) and \( y_{S2} \).

For Scale Beam 1 an option has been created to place an extra damper at \( x_{Sb1} \) to simulate the behavior of the system when this would be implemented in the machine.
Figure 2.4 shows the set up of the scale beams in a more graphical way. Parametric names of bearings and body centers of gravity are also shown in this figure, along with their position indications.

Some position values are not shown in this figure, this is because they lie on the axis of measurement and are equal to zero. This is a logical consequence of the design of the machine, in which Abbe errors are eliminated.

Since the figure is only 2D, position values in z-direction are also not shown. However, these values can be found in the m-file “nanoCMMdata” in Appendix A, along with all other values.

**Mounting face and Z-drive**

The mounting face and z-drive are modeled together as one mass of 1.0 kg having an inertia tensor

\[
I_Z = \begin{bmatrix}
\frac{1}{12} m_Z (w_Z^2 + h_Z^2) & 0 & 0 \\
0 & \frac{1}{12} m_Z (h_Z^2 + l_Z^2) & 0 \\
0 & 0 & \frac{1}{12} m_Z (l_Z^2 + w_Z^2)
\end{bmatrix}
\]
Where $m$ is the mass of the mounting face and z-drive, $l$ is the length in x-direction, $w$ is the width in y-direction and $h$ is the height, in z-direction. The index Z is used to indicate the application of this tensor to the mounting face and z-drive.

Figure 2.5: Block representation of the mounting face and z-drive

In the model, the mounting face and z-drive are rigidly connected on top of the scale beams (figure 2.5), with its centre of gravity CGZ directly right above the elastic hinge that connects the scale beams at the probe side. Any rotation of the mounting face can therefore only be originated by a common rotation of both scale beams.

Further, the “Body” block that represents the mounting face and z-drive has been equipped with an extra port, on which a custom joint is connected. This joint serves as an attachment point for three joint sensors that measure the displacement in x-, y- and z-direction as well as the z-rotation of the mounting face and z-drive, relative to the ground. These data are then registered by means of scopes and “To Workspace” blocks.
**Bearings**

All moving parts of the nanoCMM are guided by air bearings with respect to the fixed world, or to another body. In the model, an air bearing is described as follows. A “Ground” or “Body” block serves as the guiding surface. Connected to this guiding surface is a custom joint, which allows all degrees of freedom except the translation perpendicular to the guiding surface. A “Joint Spring & Damper” block is connected to the custom joint to account for the rotational stiffness in $\varphi$ and $\psi$ direction caused by the aligning block. On the other side, the custom joint is connected to a massless body that is necessary to build the guiding. In turn, the massless body is connected to a “Linear Spring & Damper” block that accounts for the stiffness of the bearing, and to a “Prismatic Joint” block that prescribes the direction of the stiffness. Finally, these are connected to the guided body, see figure 2.6. The stiffness of the bearing is determined by the stiffness of the air layer, the stiffness of the aligning block and in some cases, the stiffness of the suspension of the body.

The stiffness of the guiding surface is assumed to be infinitely high compared to that of the air bearing system, so that it can be left out of the model.

**Actuation**

The scale beams of the nanoCMM are driven by two Lorentz coils, for x and y actuation. In the model these actuators are represented by “Joint Actuator” blocks. These blocks can be connected to any kind of joint. In turn, this joint is connected on one side to the body that is to be driven (follower), and on the other side to the base with respect to which the body is moving to or from (base).

A “Joint Actuator” block sends a signal to a joint to do a certain translation or rotation as a function of time. In order to do this, the joint actuator needs to be connected to a “From Workspace” block from the Simulink toolbox “Sources”. This block needs to deliver a 3-dimensional input signal, being an (n x 3) matrix for position, velocity and acceleration, or a 1-dimensional vector for force or torque actuation.

For actuation of the scale beams, special configurations have been modeled that simulate the behavior of the actuation process in reality, see figure 2.7. A scale beam and an intermediate body are connected to each other by a custom joint that allows all six degrees of freedom between the two.
Next, a “Joint Actuator” block is connected to the custom joint. The “Joint Actuator” block is then set to actuate the custom joint in y-direction for Scale Beam 1, or in x-direction for Scale Beam 2. Since in the real machine, the Lorentz actuators are positioned on top of the scale beams, directly above the z-bearings away from the probe, this is also the case in the model. The input signal for the “Joint Actuator” block can be obtained from any matrix or array that is loaded in MATLAB, for instance from an m-file. The input signal can also be any other “Source” block from the Simulink toolbox.

Measurements
On both intermediate bodies a measurement system is connected that represents the measuring scales and recorders that can be found on the nanoCMM. A measurement system consists of a joint sensor that can read any angular or translational displacement, velocity and acceleration from a desired primitive, or any resulting force or torque on the joint, see figure 2.8. This signal can then be plotted as a function of time by connecting a scope, or routed to a “To Workspace” block, through which it is stored in MATLAB.

Of course, any movement of a particular body relative to another body or to the absolute world can be registered by placing a similar configuration between the two. Also movements of joints that are already part of the model, like bearings and hinges, can be registered by simply connecting a set of scopes and “To Workspace” blocks to the corresponding “Joint” blocks. The result is a graph or an array of data that can be processed in MATLAB.

In this model, scopes and “To Workspace” blocks have already been added for any significant translation or rotation of bodies and joints. A list of these is given in Appendix C, along with the corresponding data that is recorded. In case other information is required for special purposes, the model can easily be changed by following the procedure described above.
**Configuration Parameters**

Once the model is completely constructed, the desired parameters must be set for calculation of the dynamic behavior of the model. This is done in the Configuration Parameters dialog box (figure 2.9), which is a vital feature of the Simulink toolbox. Here the simulation time can be set, and an appropriate solver for the model can be chosen. Since this model is constructed out of several bodies with different masses and stiffnesses, the model may be considered as a dynamically stiff system, which makes the “ODE23s” solver most suitable. The minimum and maximum calculation step-size are set to 1e-3 respectively 5e-3, which account for calculation of the greatest part of the dynamic behavior.

Faster phenomena that occur in the system will not be made clear by this model. However, in this case this will not be of any significant value, since the slower dynamic properties occurring in the range between 0 and 500 Hz are the main focus in this case. Choosing the right solver and step size has a great influence on results as well as CPU time and should therefore be chosen well.

![Figure 2.9: The Configuration Parameters dialog](image)

In the “Machine Environment” dialog box (figure 2.10), an option exists to set the gravity vector, which would be appropriate in this case. The gravity vector is set to \[ 0 \ 0 \ -9.81 \].

Correspondingly, the machine environment is set to 3D analysis, since not only in-plane dynamics, but also in behavior in z-direction can be of importance. Choosing 2D for this option would save a great deal in simulation time, but this can only be enabled if the machine actually moves only in 2D, otherwise the simulation will stop with an error.

In this dialog box the type of analysis mode and the linear and angular assembly tolerance that apply to the model also need to be configured. The analysis mode can be set to either Forward Dynamics, applicable to systems where a force or torque
signal serves as an input variable, or Inverse Dynamics, when the input is a motion signal.

The linear and angular assembly tolerances are set to 1e-3 m and 1e-3 rad respectively, which is the standard value for any SimMechanics model. Decreasing the tolerance would result in increased CPU times, but will deliver more accurate results.

Constraint solver type is set to “Stabilizing” which adds a self-correcting term to the state equations to be solved that stabilizes the numerical solution, so that it evolves toward, rather than drift away from, the actual solution.

In the Linearization pane, the state perturbation type is set to “fixed”, with a perturbation size of 1e-5.

Finally, to enable visualization and animation of the connected machine, the “Visualize Machine” option is selected, in order to enable the “Machine for Model” feature.

Visualization
The machine can be visualized using the “Machine for Model” option in the Configuration Parameters dialog box. This enables the feature to see the model in an abstract form, showing the bodies and joints and how they are connected. The “Machine for Model” program builds up an image of the model by using the parameter input provided, such as relative positions of joints and bodies to draw a map of the machine, and inertia tensors to draw body surfaces or equivalent body ellipsoids. Since the visualization is only an interpretation of the model through its input parameters, it will never show the actual appearance of the machine. However, this option can be used to check whether the model is built correctly already at an early stage, and is therefore a valuable tool.
“Machine for Model” can even show an animation of the model when an input signal is put into it, so that an impression of the dynamic behavior of the model under different circumstances can be obtained. This animation can eventually be recorded to an AVI-format, to be reviewed later. The speed of animation, however, is dependent on the size of simulation time intervals and the speed of the processor. Since enabling the “Machine for Model” option requires some CPU time, this adds to the work required from the processor and therefore slows down the calculation speed.

In figure 2.11, an example of using the “Machine for Model” option is given.

![Figure 2.11: Visualization window](image)

In the “Machine for Model” dialog window, the appearance of the visualization can be adjusted by using the options present. The model can be rotated to the preferred view, or the user can zoom in or out to watch certain aspects of the model in more detail.

The appearance of the model can be changed by switching on or off the indication of joints or body centers of gravity, to get a better view of the model. Also body surfaces can be turned on or off, or convex hulls or equivalent body ellipsoids can be displayed, which are a representation of the inertia tensors, as mentioned before.

All in all, the “Machine for Model” option is a powerful tool to get an idea of the model and its behavior, and can be used to check for correctness already at an early stage.

The model is now ready to perform simulations. First, some analysis methods are discussed in the next chapter.
Chapter 3 – Output Analysis

Once the model has been built and configured, it can be used to predict the behavior of the nanoCMM machine. To do so, the behavior of the model can be analyzed and then translated to the real system.

The model needs an input signal to simulate the behavior of the machine. The type of input signal is dependent on the analysis type that has to be done.

In all cases, the output signal can be obtained from any part of the model, specifically bodies and joints. “Joint Sensor” blocks that are connected to Scopes or “To Workspace” blocks can register the resulting motion or forces that occur at any point of the system as a result of the input signal.

In this chapter, some different types of available input signals are discussed.

Time domain analysis
An input signal to a joint needs to be a 1-dimensional force or torque signal, or a 3-dimensional motion signal matrix containing a position, a velocity, and an acceleration column. All of these are regarded as being in the time domain, all be it simulation time. Input signals that are i.e. constant, linear or sinusoidal can therefore be made, as long as they are a function of time. An input signal can be made in MATLAB and loaded into SimMechanics by using a “From Workspace” block.

An example of a motion input signal is given below.

\[
\begin{align*}
A &= 5e^{-3}; \\
f &= 65; \\
w &= 2\times\pi\times f; \\
t &= 0:0.01:1.99; \\
displacement &= A\times\sin(w\times t); \\
velocity &= w\times A\times\cos(w\times t); \\
acceleration &= -w^2\times w\times A\times\sin(w\times t); \\
input &= [t' \ \text{displacement}' \ \text{velocity}' \ \text{acceleration}'];
\end{align*}
\]

Where \(A\) equals the amplitude of the signal, \(f\) is the frequency and \(t\) is the time.

An example of such a motion input signal can be seen in figure 3.1.

![Figure 3.1: Motion input signal with a frequency of 5 Hz](image-url)
An example of a force input signal is given below.

\[
\text{inputf} = \text{zeros(size(t))}'; \\
\text{inputf}(2,1) = 100; \\
\text{inputf} = [t' \text{ inputf}];
\]

Which represents a force pulse signal at time t=0.01 s as can be seen in figure 3.2.

![Figure 3.2: 100 N Force pulse input signal at time t=0.01 s](image)

Mind that a motion input signal should always be 3-dimensional, of which the velocity input is exactly the derivative of the position or rotation, and the acceleration input is exactly the derivative of the velocity.

On the other hand, a force or torque signal should always be 1-dimensional.

For both type of signals, a time column should be placed in front of the input matrix for it to be a valid input signal.

Time domain analysis can be used to simulate the behavior of the system at a specific input frequency, for instance to look at the different eigenmodes that occur at their eigenfrequencies. Furthermore, this method can be used to quantitatively predict the error of the machine, once the model is tuned so that it perfectly resembles the actual machine. Since this requires a lot of experimental data, which was not available for this research, this is not discussed in this report. However, time domain analysis can give a qualitative comparison for the system when certain design parameters are changed.

**Frequency analysis**

In case it is desirable to do a frequency analysis on the model, a “Band-Limited White Noise” block can be used. This block generates normally distributed random numbers, at a specific sample rate which is related to the correlation time of the noise.

The Band-Limited White Noise can be connected to a “Joint Actuator” block and serves as a 1D force input signal to a joint. This causes dynamic behavior in the model that contains all frequencies. The response of all parts of the model can be registered by connecting scopes and “To Workspace” blocks, as was described in the previous chapter. These response functions can then serve as an output signal.
The input and output signal can be used to form a transfer function which can then be used to make a frequency response figure.

A bode plot can be made by using the command “tfestimate( )” in MATLAB. This command needs an input and output signal of equal length, a window function, an overlap constant, an NFFT constant, and a sample frequency.

\[ [P, Hz] = \text{tfestimate}(\text{input, output, window, overlap, nfft, sample\_frequency}) \]

The window-input can be any window function i.e. Hanning or Hamming. The sample frequency is dependent on the time step that was used in the Configuration Parameters dialog, which is discussed in the previous chapter. Sample frequency can never be higher than the inverse of the time step. NFFT is also dependent on the chosen time-step, as well as the total simulation time. The overlap function can help averaging the outcome, and must be chosen smaller than NFFT. In order to get a good estimate of the transfer function, NFFT and sample frequency should be chosen as large as possible. However, this increases CPU load and corresponding CPU times. Consider therefore how much information is necessary to produce good results.

The magnitude and the frequency scale of the estimated transfer function are stored in a vector, which can then be plotted.

Frequency analysis can be used to find the resonance and anti-resonance peaks of the system, and give an idea of its bandwidth.

An example is given below (figure 3.3), where the input signal is a Band-Limited White Noise, while the output is the absolute x-displacement of the mounting face and z-drive.

![x-driven frequency response function](image)

**Figure 3.3: Example of a frequency response function, from a Band-Limited White Noise input signal**
**Impulse response analysis**

By connecting a “Pulse Generator” block, one can do an impulse response analysis of the model.

![Pulse generator dialog](image)

Figure 3.4: Pulse generator dialog

This block can simulate a single pulse by setting the period time higher than the total simulation time, and making sure the pulse width resembles a Dirac function. The amplitude of the pulse is a measure for the force that is put into the system.

![0.1 N x-actuated impulse response](image)

Figure 3.5: Result of a 0.1 N force pulse input signal
Chapter 4 - Model Results

Now that the model has been built and verified, and all configuration parameters have been determined, the model can be used to do simulations of the behavior of the real nanoCMM machine.

This chapter shows the results of a number of simulations that were performed, plotted in figures 4.1 to 4.11. Note that the values that are shown in the graphs can deviate from the actual values that would be obtained by doing experimental tests. This model has not yet been compared to the actual machine, and its results can therefore only be interpreted qualitatively, rather than quantitatively.

x-Actuation
The following results are based on an actuation in x-direction of the scale beams, as would be the case for the real machine. Intermediate bodies A and B are supported on the base by their bearings, and the scale beams are guided sideways along the intermediate bodies as well as the base, in z-direction.

Figure 4.1 below shows the frequency response function measured at the z-drive, after a Band-Limited White Noise has been used as an input signal.

![x-driven frequency response function](image)

Figure 4.1: Frequency response function when the scale beams are driven in x-direction
Two different outputs have been measured to form the plot. The green line originates from an x-measurement of the z-drive, while the blue line results from the rotation of the z-drive about the z-axis. Note that the model shows two different transfer functions that are correlated to each other at several points. The resonance peaks at approximately 120 Hz and 300 Hz and the anti-resonance peaks at 220 Hz and 500 Hz occur in both response functions.

Also note that the green line shows a resonance peak at 185 Hz, and an anti-resonance peak around 50 Hz. This anti-resonance peak is caused when Scale Beam 1 passes a force through x-bearing xS1, which causes Intermediate Body A to rotate about the z-axis. However, this rotation is suppressed by y-bearings yA1 and yA2. The same holds for Intermediate Body B. These rotations happen in anti-phase, so that the rotations of both bodies are converted to a translation of these two bodies together.

This can also be seen in figure 4.2, where the translations in x-direction of the z-drive and Intermediate Body A are plotted against time.

Note that Intermediate Body A moves further than the Z-drive. The difference in distance is 0.5e-3 m, which is the prescribed natural length of the air bearing xS1.

![Figure 4.2: Eigenmode at 50 Hz](image-url)
The resonance peak at 185 Hz is caused when Intermediate Body A moves in its rotational eigenfrequency. This rotation takes the scale beams along with it and eventually the Z-drive.

This is also shown in figure 4.3, where the rotation of Intermediate Body A and the translation in x-direction of the z-drive are plotted against time.
The anti-resonance peak that occurs at around 240 Hz is caused by an opposing rotation of Intermediate Body A and B. Because they rotate in an opposite direction, the scale beams stay at rest and any input signal is not followed through by the system.

![Figure 4.4: Eigenmode at 240 Hz](image)

This can also be seen in figure 4.4, where the rotations of Intermediate Body A and B are plotted against time.
The last resonance peak below 500 Hz occurs at around 300 Hz. The corresponding eigenmode is a rotation of Intermediate Body B which is in anti-phase with the rotation of the scale beams. As can be seen in figure 4.6, Intermediate Body A also shows a small resonance peak at this frequency, but this can barely be seen when we look at the eigenmode in time domain.

The eigenmode that occurs around 300 Hz is therefore mainly caused by bearings yS1 and yS2, along which the scale beams are guided through Intermediate Body B.

Figure 4.5 shows the rotation of the z-drive and Intermediate Body B, plotted as a function of time.
When we look at the z-rotation frequency response functions in more detail, making use of figure 4.6, we can confirm that the first resonance peak is actually caused by rotation of the scale beams and Intermediate Body B only. Intermediate Body A does not take part in this eigenmode.

![Frequency response function](image)

**Figure 4.6: Frequency response function of the rotations of the z-drive, Intermediate Body A and Intermediate Body B, when the scale beams are actuated in x-direction**

At the second resonance peak and corresponding eigenmode, Intermediate Body A slightly rotates as well. The second eigenmode is therefore a common rotation of all bodies in the system.

Note that Intermediate Body A also has a resonance peak around 185 Hz, which was already shown in figure 4.3.
The model is equipped with an extra option to add a damper at position $xSb1$, which would make the design more symmetrical. Figure 4.7 is a result of this option, and shows the frequency response function after the damper has been added.

Adding a damper at position $xSb1$ will have the following effect.

![x-driven frequency response function](image)

**Figure 4.7: Frequency response function of the system after adding a damper at $xSb1$**

The $z$-rotation line has barely changed; the extra damper only diminishes the magnitude of the resonance peaks slightly. However, the green line has changed a lot. The damper at $xSb1$ takes out the energy that would otherwise be used to rotate the intermediate bodies, which stabilizes the $x$-translation of the $z$-drive.

This happens at least until the first resonance peak in the blue line, after which a small anti-resonance peak occurs, followed up by a resonance peak. These are caused by the rotational eigenmodes of the intermediate bodies.
**y-Actuation**

When the scale beams are driven in y-direction, the frequency response function has the following shape (figure 4.8).

![y-driven frequency response function](image)

**Figure 4.8: Frequency response function when the scale beams are driven in y-direction**

This looks a lot like the x-actuated system, when an extra damper is added at xSb1. The only difference is that below the first resonance peak in the z-rotation line, the y-translation line shows a resonance peak, whereas in the x-actuated system this would be an anti-resonance peak.

Results of adding a damper at xSb1 are shown in figures 4.9 and 4.10. In fact, adding a damper has no significant influence on the y-translation frequency response function. The difference between the system with and without damping can only be seen when we zoom in close to the resonance peaks, where the damper slightly decreases the magnitude of the peaks.
Figure 4.9: Frequency response function with and without damping. The highlighted area is shown in more detail in figure 4.10.

Figure 4.10: Detailed figure of the highlighted area in figure 4.9
When we look at the frequency response functions of the intermediate bodies when the system is actuated in y-direction, the results are the same as for the x-actuated system, as can be seen in figure 4.11.

The first and second eigenmode, which are formed by rotations of the scale beams and the intermediate bodies are therefore not dependent on whether the system is actuated in x- or y-direction.

Figure 4.11: Frequency response function of the rotations of the z-drive, Intermediate Body A and Intermediate Body B, when the scale beams are actuated in y-direction
Conclusion

To investigate the dynamic behavior of a nanoCMM machine, a SimMechanics model has been made. This model is a simplified representation of the actual machine, various parts of the machine are replaced by single body blocks that have only mass and inertia properties. Internal stiffness and degrees of freedom of the various parts are modeled by special configurations which approximate the behavior of the real machine.

The model is supported by an m-file, from which all system parameters are loaded. Before any simulation, these can be changed so that the effect of these parameters on the system can be registered. The model can also be used to investigate the consequences of adding certain features to the machine, like additional dampers. In this way, the model may serve as a tool to improve the actual machine.

Results of any simulation can be registered and transported to MATLAB workspace, where it can be processed to numerical or graphical form, which is useful for comparison with earlier results. The machine can so be tuned on a qualitative basis. Results can be processed in time domain, or in frequency domain, whichever is more suitable.

Frequency analysis can be used to locate resonance and anti-resonance peaks, while time domain analysis can point out shapes of eigenmodes that occur under various frequencies. This method has been used to investigate the first three eigenmodes that occur below 500 Hz in the frequency domain. Also, the two anti-resonance peaks are explained.

However, in the absence of experimental results, the model has not been compared to the nanoCMM. This means that the model can only be verified partly, through means of analytical insight. Any results that come from model simulations can therefore only be interpreted qualitatively, rather than quantitatively. Nevertheless, the model is an excellent means to analyze and possibly improve the dynamic properties of the machine.
**Literature**


- www.nanoCMM.eu
clear all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DATA FILE FOR USING nanoCMM.mdl
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% INPUT SIGNAL PARAMETERS
Ay = 5e-3; % amplitude of the sinusoidal signal for y-actuation
Ax = 5e-3; % amplitude of the sinusoidal signal for x-actuation
fy = 5; % frequency for y-actuation
fx = 220; % frequency for x-actuation
wy = 2*pi*fy;
wx = 2*pi*fx;

% time vector for y-actuation
ty = 0:0.01:1.99;
% time vector for x-actuation
tx = 0:0.01:1.99;

% INPUT SIGNALS
displacementy = Ay*sin(wy*ty);
velocityy = wy*Ay*cos(wy*ty);
accelerationy = -wy*wy*Ay*sin(wy*ty);

displacementx = Ax*sin(wx*tx);
velocityx = wx*Ax*cos(wx*tx);
accelerationx = -wx*wx*Ax*sin(wx*tx);

% MOTION INPUT SIGNALS
inputy = [ty' displacementy' velocityy' accelerationy'];
inputx = [tx' displacementx' velocityx' accelerationx'];

% FORCE PULSE INPUT SIGNAL
inputf = zeros(size(ty))';
inputf(2,1) = 100;
inputg = [ty' inputf];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%% INTERMEDIATE BODY A %%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% CENTRE OF GRAVITY
xCGA = 0;
yCGA = 190e-3;
zCGA = 50e-3;

% Z-BEARINGS
xzA1 = -68.5e-3;
yzA1 = 255e-3;

xzA2 = 68.5e-3;
yzA2 = 255e-3;

xzA3 = -68.5e-3;
yzA3 = 125e-3;

xzA4 = 68.5e-3;
yzA4 = 125e-3;

% Y-BEARINGS
xyA1 = -68.5e-3;
yyA1 = 150e-3;
zyA1 = -25e-3;

xyA2 = 68.5e-3;
yyA2 = 150e-3;
zyA2 = -25e-3;

% X-BEARINGS
xxS1 = 0;
yxS1 = 125e-3;
zxS1 = 40e-3;

xxSb1 = 0;
yxSb1 = 255e-3;
zxSb1 = 40e-3;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%% INTERMEDIATE BODY B %%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% CENTRE OF GRAVITY
xCGB = 190e-3;
yCGB = 0;
zCGB = 50e-3;

% Z-BEARINGS
xzB1 = 125e-3;
yzB1 = 68.5e-3;

xzB2 = 255e-3;
yzB2 = 68.5e-3;

xzB3 = 125e-3;
yzB3 = -68.5e-3;

xzB4 = 255e-3;
yzB4 = -68.5e-3;

% Y-BEARINGS
xyS1 = 125e-3;
yyS1 = 0;
zyS1 = 40e-3;

xyS2 = 255e-3;
yyS2 = 0;
zyS2 = 40e-3;

% X-BEARINGS
xxB1 = 150e-3;
yxB1 = 68.5e-3;
zxB1 = -25e-3;
\[ xx_2 = 150e^{-3}; \]
\[ yy_2 = -68.5e^{-3}; \]
\[ zz_2 = -25e^{-3}; \]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SCALE BEAM
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Z-BEARINGS
\[ xz_1 = 0; \]
\[ yz_1 = 307.5e^{-3}; \]
\[ xz_2 = 0; \]
\[ yz_2 = 72.5e^{-3}; \]
\[ xz_3 = 72.5e^{-3}; \]
\[ yz_3 = 0; \]
\[ xz_4 = 307.5e^{-3}; \]
\[ yz_4 = 0; \]

% CENTRES OF GRAVITY
\[ x_{CGS_y} = 0; \]
\[ y_{CGS_y} = 0.5(yz_1-yz_2) + yz_2; \]
\[ z_{CGS_y} = 40e^{-3}; \]
\[ x_{CGS_x} = 0.5(xz_4-xz_3) + xz_3; \]
\[ y_{CGS_x} = 0; \]
\[ z_{CGS_x} = 40e^{-3}; \]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Z-MOUNTING FACE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\[ x_{Z1} = xz_3 - 0.02; \]
\[ y_{Z1} = yz_2 - 0.02; \]
\[ z_{Z1} = z_{CGS_y} + 0.02; \]
\[ x_{Z2} = xz_3 - 0.02; \]
\[ y_{Z2} = yz_2 - 0.02; \]
\[ z_{Z2} = z_{CGS_x} + 0.02; \]

% CENTRE OF GRAVITY
\[ x_{CGZ} = x_{Z1}; \]
\[ y_{CGZ} = y_{Z1}; \]
\[ z_{CGZ} = 100e^{-3}; \]
% % MASSES % %
\begin{verbatim}
m_A = 4.0; \quad \% mass of intermediate body A
m_B = 4.0; \quad \% mass of intermediate body B
m_scale = 1.5; \quad \% mass of a scale beam
m_Z = 1.0; \quad \% mass of the mounting face and z-drive
\end{verbatim}

% % INERTIAS % %
\begin{verbatim}
% INTERMEDIATE BODY A
zA = 100e-3;
yA = 0.5*(yzA1 - yzA3) + yzA3;
xA = -xzA3 + xzA4;
I_A = \begin{bmatrix}
(1/12)*m_A*(zA^2+yA^2) & 0 & 0 \\
0 & (1/12)*m_A*(xA^2+zA^2) & 0 \\
0 & 0 & (1/12)*m_A*(xA^2+yA^2)
\end{bmatrix};

% INTERMEDIATE BODY B
zB = 100e-3;
yB = yzB1 - yzB3;
xB = 0.5*(xzB4 - xzB3) + xzB3;
I_B = \begin{bmatrix}
(1/12)*m_B*(zB^2+yB^2) & 0 & 0 \\
0 & (1/12)*m_B*(xB^2+zB^2) & 0 \\
0 & 0 & (1/12)*m_B*(xB^2+yB^2)
\end{bmatrix};

% SCALE BEAMS
zSy = 40e-3;
ySy = 0.5*(yzS1 - yzS2) + yzS2;
xSy = 40e-3;
I_Sy = \begin{bmatrix}
(1/12)*m_scale*(zSy^2+ySy^2) & 0 & 0 \\
0 & (1/12)*m_scale*(xSy^2+zSy^2) & 0 \\
0 & 0 & (1/12)*m_scale*(xSy^2+ySy^2)
\end{bmatrix};

zSx = 40e-3;
ySx = 40e-3;
xSx = 0.5*(xzS4 - xzS3) + xzS3;
I_Sx = \begin{bmatrix}
(1/12)*m_scale*(zSx^2+ySx^2) & 0 & 0 \\
0 & (1/12)*m_scale*(xSx^2+zSx^2) & 0 \\
0 & 0 & (1/12)*m_scale*(xSx^2+ySx^2)
\end{bmatrix};

% MOUNTING FACE AND Z-DRIVE
zZ = 100e-3;
yZ = 70e-3;
xZ = 70e-3;
\end{verbatim}
\[
I_Z = [(1/12)*m_Z*(zZ^2+yZ^2) \ 0 \ 0; \\
0 (1/12)*m_Z*(xZ^2+zZ^2) \ 0; \\
0 0 (1/12)*m_Z*(zZ^2+yZ^2)];
\]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% AIR BEARING PARAMETERS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

kI40b = 1/(1/3.1e7 + 1/7.5e7); % 40mm Air bearing z-stiffness at the probe side
kI40a = 1/(1/3.1e7 + 1/7.5e7); % 40mm Air bearing z-stiffness away from the probe
k50 = 3.9e7; % 50mm Air bearing x- or y-stiffness
kSa = 6.8e6; % 40mm Air bearing z-stiffness of the scale
kSb = 1.0e7; % 40mm Air bearing z-stiffness of the scale beam away from the probe
kR = 16; % rotation stiffness of air bearing
d40 = 0; % damping for air bearing 40mm
d50 = 0; % damping for air bearing 50mm
dSb1 = 1000; % extra damping for scale beam y
l40 = 5e-6; % air gap for 40mm air bearing
l50 = 5e-6; % air gap for 50mm air bearing

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SCALE BEAM STIFFNESS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

kSpx = 4e6; % stiffness of the connection at the probe side in x-direction
kSpy = 4e6; % stiffness of the connection at the probe side in y-direction
k_rod = 2.5e7; % stiffness of the rod at the backside
dSp = 0; % damping in the connection at the probe side
dSr = 0; % damping in the rod at the backside
l_rod = sqrt(yzS1^2+xzS4^2); % length of the rod at the backside
## Appendix C — Signal Outputs

<table>
<thead>
<tr>
<th>Location</th>
<th>Name</th>
<th>Scopes</th>
<th>To Workspace</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intermediate Body A</td>
<td>yA1</td>
<td>Displacement x</td>
<td>Displacement x</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Displacement y</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Displacement z</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rotation y</td>
<td></td>
</tr>
<tr>
<td></td>
<td>yA2</td>
<td>Displacement x</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Displacement y</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Displacement z</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rotation y</td>
<td></td>
</tr>
<tr>
<td></td>
<td>zA1</td>
<td>Displacement x</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Displacement y</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Displacement z</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rotation z</td>
<td></td>
</tr>
<tr>
<td></td>
<td>zA2</td>
<td>Displacement x</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Displacement y</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Displacement z</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rotation z</td>
<td></td>
</tr>
<tr>
<td></td>
<td>zA3</td>
<td>Displacement x</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Displacement y</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Displacement z</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rotation z</td>
<td></td>
</tr>
<tr>
<td></td>
<td>zA4</td>
<td>Displacement x</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Displacement y</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Displacement z</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rotation z</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CGA</td>
<td>Rotation z</td>
<td>Rotation z</td>
</tr>
<tr>
<td></td>
<td>y-Actuation</td>
<td>Position y</td>
<td>Input Signal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Velocity y</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Acceleration y</td>
<td></td>
</tr>
<tr>
<td>Intermediate Body B</td>
<td>xB1</td>
<td>Displacement x</td>
<td>Displacement y</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Displacement y</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Displacement z</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rotation x</td>
<td></td>
</tr>
<tr>
<td></td>
<td>xB2</td>
<td>Displacement x</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Displacement y</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Displacement z</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rotation x</td>
<td></td>
</tr>
<tr>
<td></td>
<td>zB1</td>
<td>Displacement x</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Displacement y</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Displacement z</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rotation z</td>
<td></td>
</tr>
<tr>
<td></td>
<td>zB2</td>
<td>Displacement x</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Displacement y</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Displacement z</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rotation z</td>
<td></td>
</tr>
<tr>
<td></td>
<td>zB3</td>
<td>Displacement x</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Displacement y</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Displacement z</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rotation z</td>
<td></td>
</tr>
<tr>
<td></td>
<td>zB4</td>
<td>Displacement x</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Displacement y</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Displacement z</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rotation z</td>
<td></td>
</tr>
<tr>
<td>CGB</td>
<td>Rotation z</td>
<td>Rotation z</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
<td>------------</td>
<td></td>
</tr>
<tr>
<td>x-Actuation</td>
<td>Input Signal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x-Measurement</td>
<td>Position x</td>
<td>Velocity x</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acceleration x</td>
<td>Position x</td>
<td></td>
</tr>
</tbody>
</table>

**Scale Beams**

<table>
<thead>
<tr>
<th>xS1</th>
<th>Displacement x</th>
<th>Displacement x</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Displacement y</td>
<td>Displacement y</td>
</tr>
<tr>
<td></td>
<td>Displacement z</td>
<td>Displacement z</td>
</tr>
<tr>
<td></td>
<td>Rotation x</td>
<td>Rotation x</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>xSb1</th>
<th>Displacement x</th>
<th>Displacement x</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Displacement y</td>
<td>Displacement y</td>
</tr>
<tr>
<td></td>
<td>Displacement z</td>
<td>Displacement z</td>
</tr>
<tr>
<td></td>
<td>Rotation x</td>
<td>Rotation x</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>yS1</th>
<th>Displacement x</th>
<th>Displacement y</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Displacement y</td>
<td>Displacement y</td>
</tr>
<tr>
<td></td>
<td>Displacement z</td>
<td>Displacement z</td>
</tr>
<tr>
<td></td>
<td>Rotation y</td>
<td>Rotation y</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>yS2</th>
<th>Displacement x</th>
<th>Displacement y</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Displacement y</td>
<td>Displacement y</td>
</tr>
<tr>
<td></td>
<td>Displacement z</td>
<td>Displacement z</td>
</tr>
<tr>
<td></td>
<td>Rotation y</td>
<td>Rotation y</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>zS1</th>
<th>Displacement x</th>
<th>Displacement y</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Displacement y</td>
<td>Displacement y</td>
</tr>
<tr>
<td></td>
<td>Displacement z</td>
<td>Displacement z</td>
</tr>
<tr>
<td></td>
<td>Rotation z</td>
<td>Rotation z</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>zS2</th>
<th>Displacement x</th>
<th>Displacement y</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Displacement y</td>
<td>Displacement y</td>
</tr>
<tr>
<td></td>
<td>Displacement z</td>
<td>Displacement z</td>
</tr>
<tr>
<td></td>
<td>Rotation z</td>
<td>Rotation z</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>zS3</th>
<th>Displacement x</th>
<th>Displacement y</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Displacement y</td>
<td>Displacement y</td>
</tr>
<tr>
<td></td>
<td>Displacement z</td>
<td>Displacement z</td>
</tr>
<tr>
<td></td>
<td>Rotation z</td>
<td>Rotation z</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>zS4</th>
<th>Displacement x</th>
<th>Displacement y</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Displacement y</td>
<td>Displacement y</td>
</tr>
<tr>
<td></td>
<td>Displacement z</td>
<td>Displacement z</td>
</tr>
<tr>
<td></td>
<td>Rotation z</td>
<td>Rotation z</td>
</tr>
</tbody>
</table>

**Z-Drive**

<table>
<thead>
<tr>
<th>Z</th>
<th>Translation x</th>
<th>Translation x</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Translation y</td>
<td>Translation y</td>
</tr>
<tr>
<td></td>
<td>Translation z</td>
<td>Translation z</td>
</tr>
<tr>
<td></td>
<td>Rotation z</td>
<td>Rotation z</td>
</tr>
</tbody>
</table>