PowerMILL 2016

MTD User Guide

PowerMILL

Acknowledgements

This documentation references a number of registered trademarks and these are the property of their respective owners. For example, Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States.

Patents

The Raceline smoothing functionality is subject to patent applications.

Patent granted: GB 2374562 Improvements Relating to Machine Tools

Patent granted: US 6,832,876 Machine Tools

Some of the functionality of the ViewMILL and Simulation modules of PowerMILL is subject to patent applications.

Patent granted: GB 2 423 592 Surface Finish Prediction

The Vortex machining functionality is subject to patent applications.

Patent application: 1121277.6 Adaptive Clearance

The MachineDNA functionality is subject to patent applications.

Patent application: 1204908.6 Machine Testing

Licenses

Intelligent cursor licensed under U.S. patent numbers 5,123,087 and 5,371,845 (Ashlar Inc.)

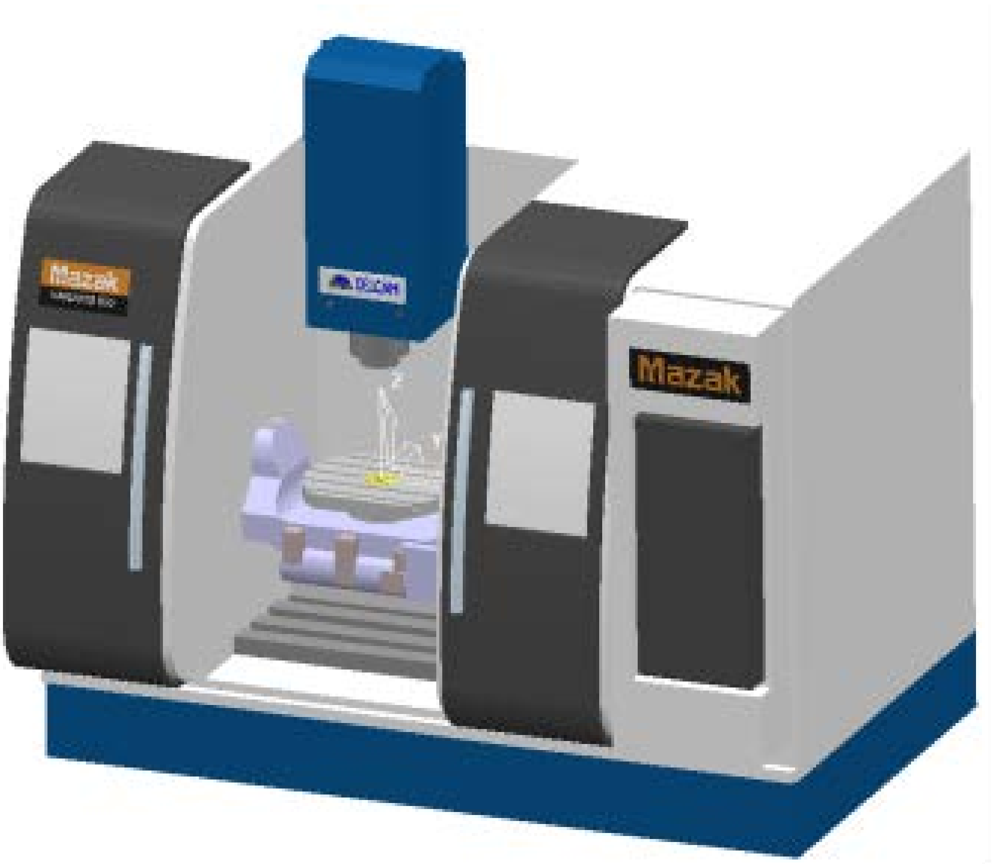
Contents

Introduction

Use this guide to create an MTD file from scratch or modify an existing MTD file to suit a particular application. It includes descriptions of the makeup of MTD files and machine tool simulation.

To use this guide, you must have a:

1. CAD model for the machine tool. This includes the spindle, machine head, table and any other aspects of the machine you wish to simulate. For example, the bed and the housing.
2. Technical data sheet, with information about:
   a. Axis limits.
   b. Axis lengths.
   c. Dimensions.
   d. Home positions (distance between table and spindle).
   e. Coordinates of the tool changes.
3. CAD model for the external components. This includes items which are not linked to the machine tool, for example, fixtures and controllers.
4. Text editor and source code editor for use with Microsoft Windows, for example, Notepad++.



Coordinates, workplanes and distances

World Workplane Position

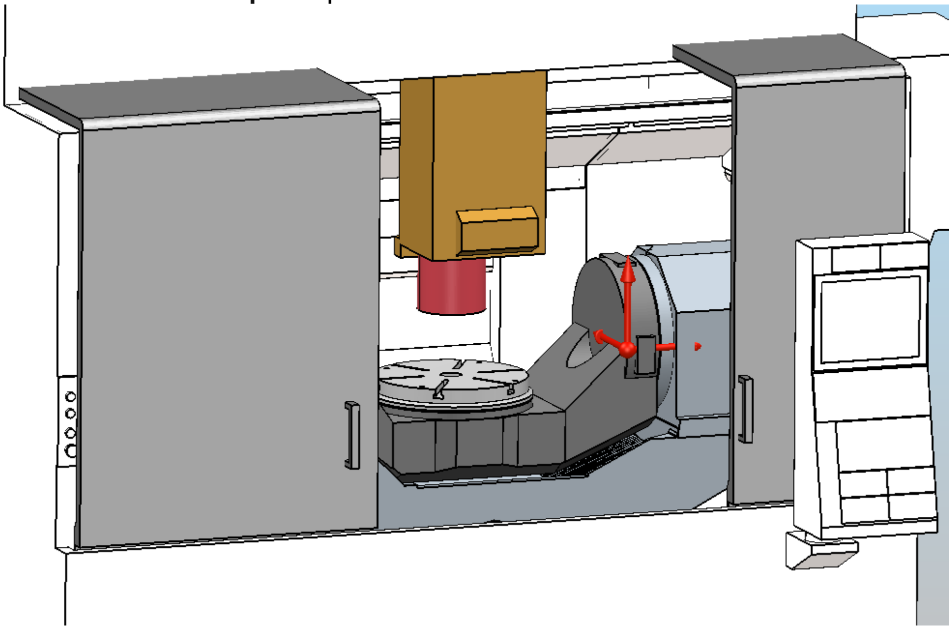
When you assemble the machine tool CAD data in PowerSHAPE, the centre of the table is located at the world coordinate (0, 0, 0). All of the remaining parts of the machine tool are located with reference to this position. The spindle should be located at **Home** or **Z-Maximum** position.

Zero Position

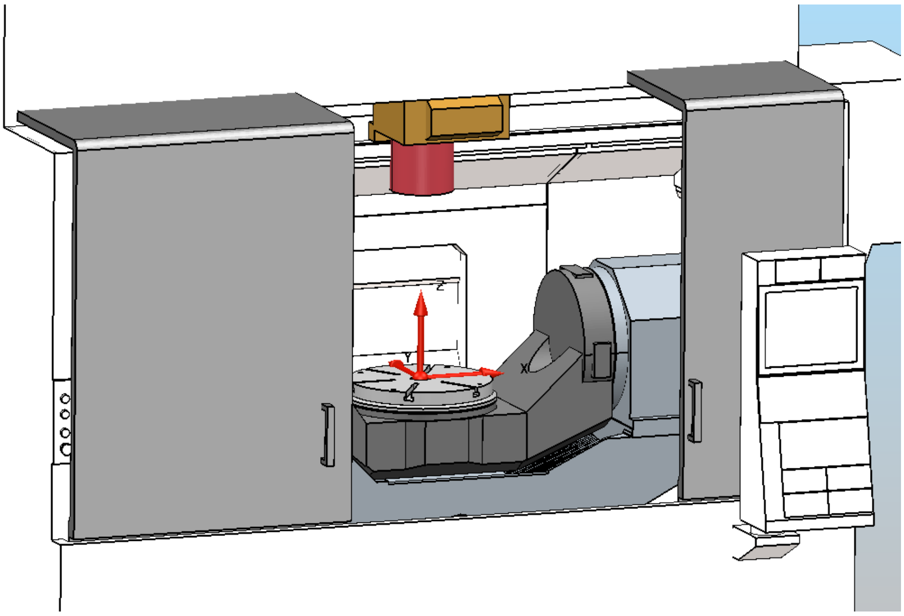
This is the position that you want the machine tool to return to when you click **Home** in PowerMILL. It is the position of the model that makes up the machine tool as it appears in PowerSHAPE.

If the **World Workplane** position or the **Zero** position are incorrect, then save the parts of the machine in the correct position. This also applies to PowerSHAPE: you can move or rotate parts of machine to the correct position.

Incorrect **World Workplane** position:



Correct **World Workplane** position:



Axes Directions

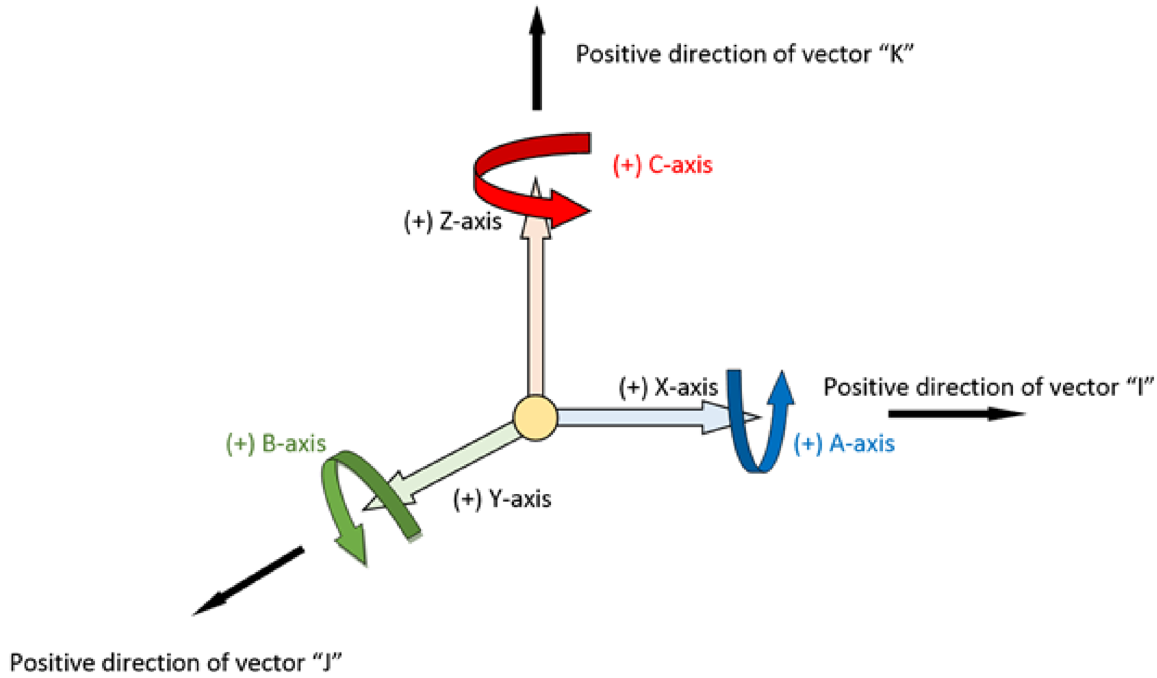
The movement of a machine tools axis, whether linear or rotational, is defined in the *.mtd file as a vector value (i, j and k). It is important that you check which direction you want an axis to move.

1 Axis Limits — Verify the minimum and maximum allowable movement value for each axis of the machine tool. This can be found from the machine tool manufacturer's technical data.

2 Head Attach Point — This is the point where the tool attaches to the spindle. It should be given as XYZ value and also include the orientation of the tool axis vector as an IJK value. It is best practice to measure this coordinate within PowerSHAPE (use tools to create geometry at the centre of the spindle tip).

The following graphic shows linear and rotary vectors of movement:

- around linear X-axis is the **A** rotary axis.
- around linear Y-axis is the **B** rotary axis.
- around linear Z-axis is the **C** rotary axis.



In some cases, there are multi-axis machines with more than 6 axes that are not kinematic. In this case, use another capital letters, such as **D**, **F**, and **G**.

Preparing CAD Data in PowerSHAPE

Loading and organising the CAD data

You can obtain the data from the machine tool manufacturer or you can model it yourself. The first step is to load and arrange the data you have in PowerSHAPE. You can remove unnecessary surfaces or solids, and then assign each axis model onto separate PowerSHAPE levels.

It advisable to label axes by their type, for example, X, Y, Z, A, B, C, Head, Bed and Table.

Moving the Machine into the origin position

To create a link between the world coordinates of PowerSHAPE and PowerMILL, it is advisable to position the machine so that the **World Workplane** position is located at the centre of the table of the machine.

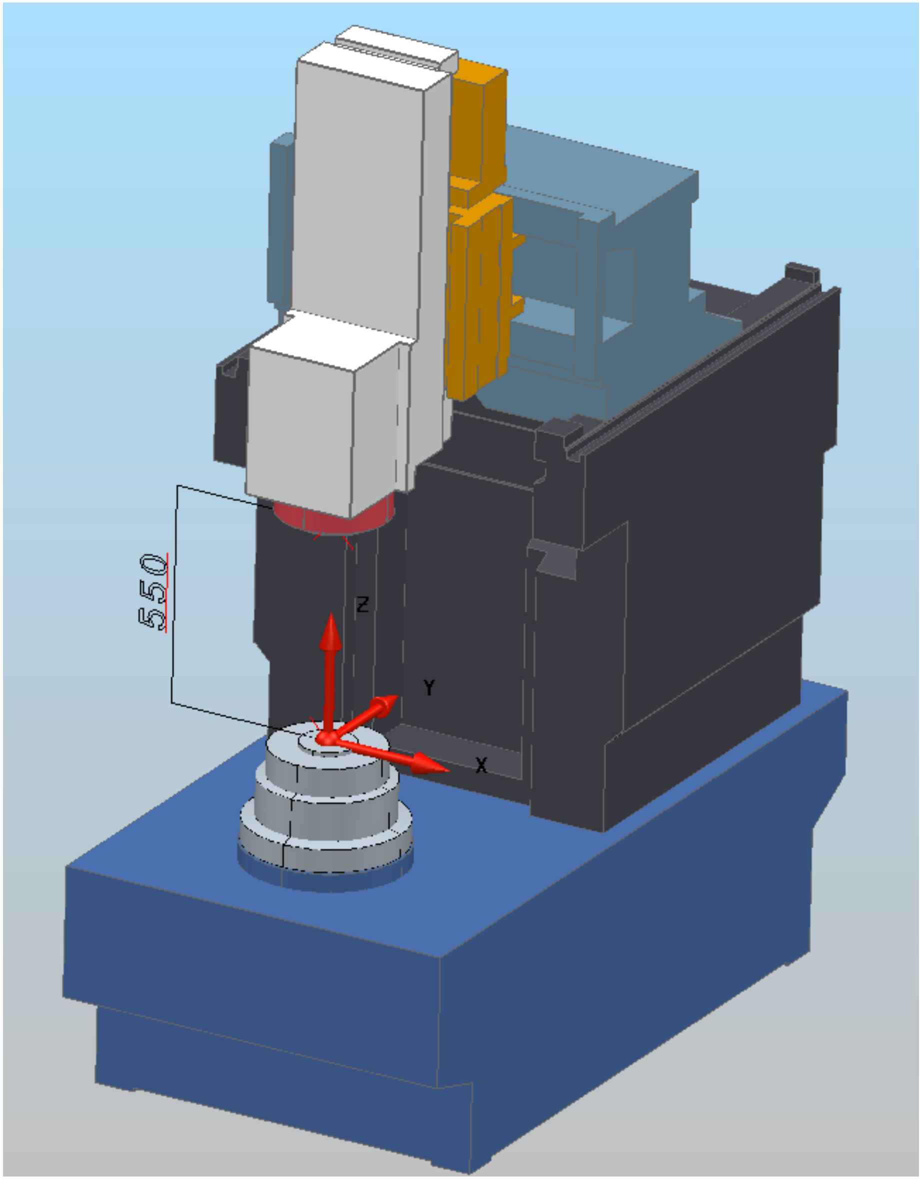
The next step is to move the axes of the machine tool into the correct position, if necessary.

The machine tool model data may set the **Home** position of an axis to an inappropriate position. For example the Z axis may be set to a position either touching or just above the machine table. In this instance you may want to change the position of the Z axis so that when it imported into PowerMILL it is clear of the model.

The following graphic shows:

- Centre of table:
  - **X**=0
  - **Y**=0
  - **Z**=0
- Centre of nose spindle:
  - **X**=0
  - **Y**=0
  - **Z**=550

The position of the **Z** axis component from the **World Workplane** must be measured and recorded in PowerSHAPE if it is positioned from its zero point, as this is used in the *.mtd file. This is the value of the gauge face and is crucial in attaching the tool to the machine tool. The rotation centre must be measured and recorded for later use.



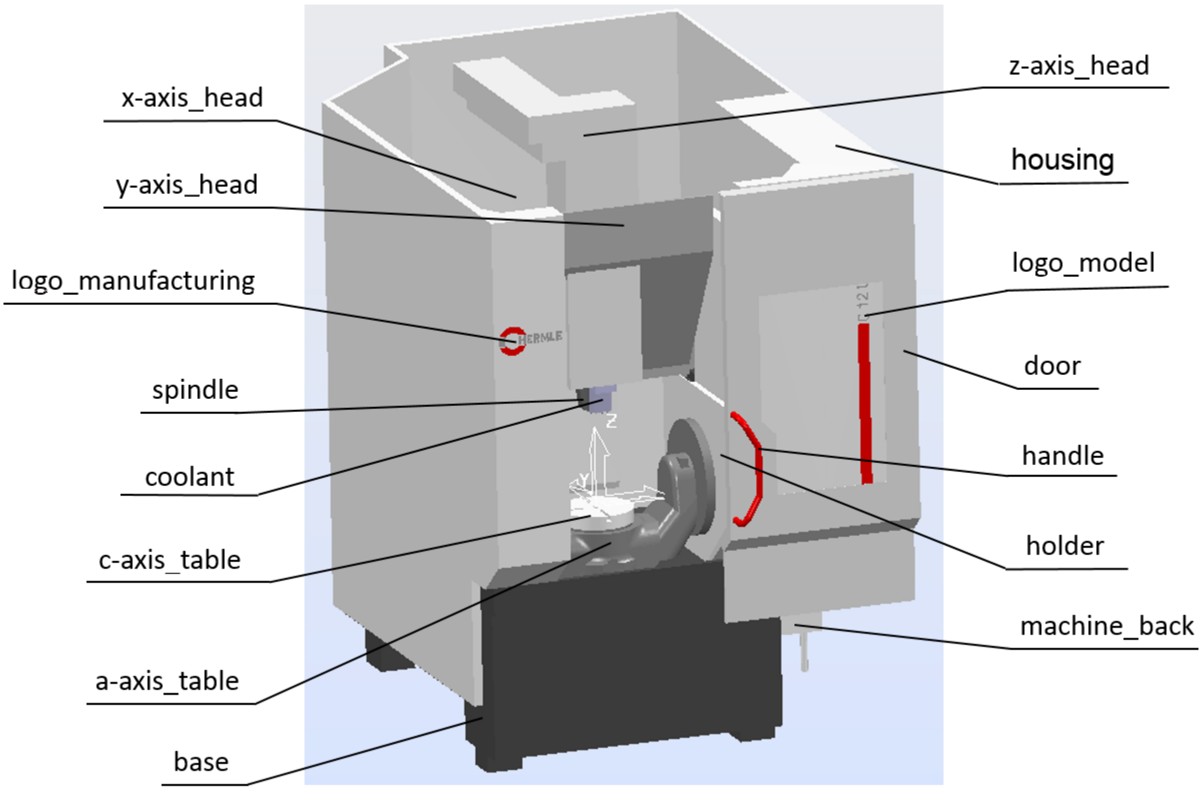
Importing External Components

If the machine tool contains a housing, tool-changer, controller or any other static entities, they must be imported or created now and positioned correctly. The machine tool model must represent the real machine that is being simulated.

Exporting Machine Tool Entities

Each axis of the machine tool (for example, the spindle, table, housing and the logos) must be exported into separate *.dmt files to be able to simulate them and edit their colour.

See the example below machine tool **Hermle_C12_SRT320**.



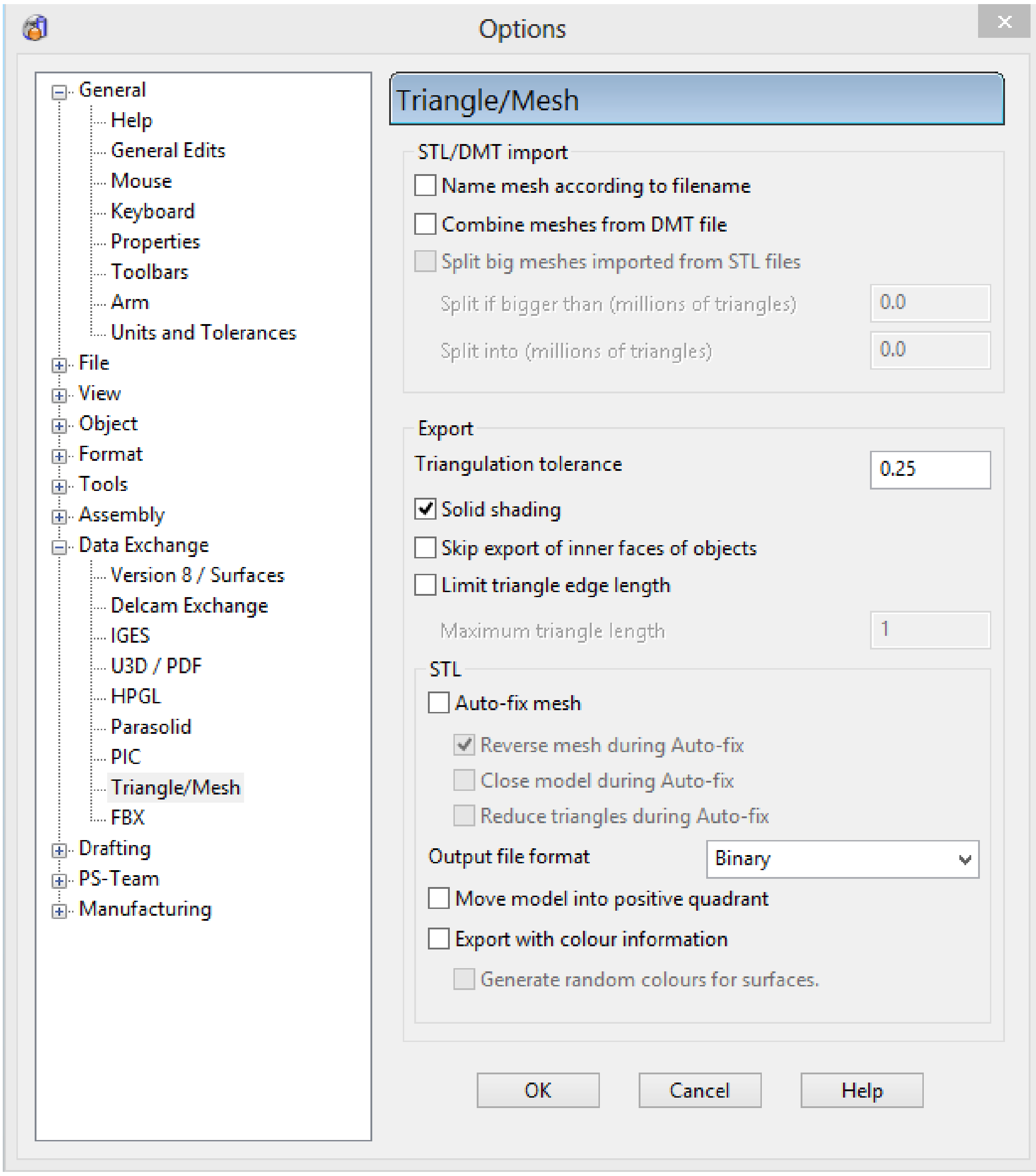
The *.dmt file contains triangles: the more triangles, the better the simulation appears. However, with more triangles, the simulation and collision-checking run more slowly.

Choose a tolerance value:

- A tolerance value of 0.025 mm is generally recommended as a good balance between quality and file size for such parts as: upper surfaces of table, spindle and some other important parts in simulations (the movable parts in working area).
- A tolerance value of 0.05 mm is generally recommended as a good balance between quality and file size for such parts as: parts of linear and rotary movements (X-axis; Y-axis; Z-axis and additional equipment).
- A tolerance value of 0.25mm is generally recommended as a good balance between quality and file size for static parts, logo, base, CNC controller and other stationary elements.

If the simulation runs too slowly, then reduce the triangulation tolerance value. All parts included on *.mtd should occupy memory not more than 3-5 Mb.

The tolerance can be set on the PowerSHAPE options page. From the **Main** toolbar select **Tools > Options > Data Exchange >Triangle / Mesh**.



Folder Structure

You must have:

- A folder containing all of the *.dmt files for the machine tool.
- The *.mtd file which describes the kinematics of the machine tool.

*.mtd File Structure

An *.mtd file has details of the solver needed to drive the machine tool. It also defines the attach points for the head and table of the machine tool. The remainder of the file is made up of objects called `machine_part`. There are two `machine_part` objects which must exist. They are called `head` and `table`. Their names must be written in lower case.

A `machine_part` can have:

- Other `machine_part` objects associated to it

An axial component can be a child of another component, which is important when dealing with the various multi axis configurations.

- Axis controls associated with it.
- *.dmt files associated with it.
- Axis controls:
  - Linear or rotational movement.
  - Rotary controls must have an address letter associated to them (typically designated as A, B or C).
  - Travel and angular limits can be defined.
  - X, Y and Z coordinates. X, Y and Z can be orientated anywhere as long as they are orthogonal.

  Only right handed axis systems are supported.

The types of axis that can be configured are:

- `simple_linear` which requires direction.
- `simple_rotary` which requires a position and a direction.

  To reverse a rotation direction, reverse the direction of vectors. Remember to reverse linear direction vector, if the table controls the axis.

When you want to create an *.mtd file, you can install the text editor on your computer, for example Notepad++

*To download Notepad++, click http://notepad-plus-plus.org/download. In the **Main** toolbar, select **Language > XML**.*

An *.mtd can consist of 6 blocks:

- Machine description (header).
- Description of PostProcessor Solver.
- Description attach points (`table_attach_point` and `head_attach_point`).
- Description of static elements.
- Description of the movable elements. The `head`.
- Description of the movable elements. The `table`.

The following is an example *.mtd:

```
<?xml version="1.0" ?>
```

```
<!-- ****************************************************** -->
<!-- Machine Tool Manufacturer  : ____________________    -->
<!-- Machine Tool Model         : ____________________    -->
<!-- Machine Tool Type          : ____________________    -->

<!--                 When       : ____________________    -->
<!--                 Checked By : ____________________    -->
<!-- ****************************************************** -->
```

1

```
<machine xmlns="x-schema:PowermillMachineTool" POST="pmpMultiaxis.*">
```

2

```
  <table_attach_point PART="table" X="0" Y="0" Z="0"/>
  <head_attach_point PART="head" X="0" Y="220" Z="430" I="0" J="0" K="1"/>
```

3

```
<!-- ===== The static parts ===== -->
    <machine_part>

    </machine_part>
```

4

```
<!-- ===== The head ===== -->
    <machine_part>

        <machine_part NAME="head" />
    </machine_part>
```

5

```
<!-- ===== The table ===== -->
     <machine_part>

        <machine_part NAME="table" />
    </machine_part>
</machine>
```

6

Editing the .mtd file for tool change simulation

Attaching a tool to a machine part

Each <machine_part> has a list of tools connected to it. Use the TOOLPASS "PART_NAME" command to pass a tool to a part. This removes the tool from the machine-tool or part and attach it to the machine part. It is treated as a part of the model for that part and is moved with the part

Defining moves before and after a tool change

Use <tool_change_pre_moves> to define moves before a tool change and <tool_change_post_moves> to define moves after a tool change. This makes it easier to implement multiple tool changers. For example:

```
<machine>
<tool_change>
<tool_change_pre_moves/>
<tool_change_moves NUMBER="1"/>
<tool_change_moves NUMBER="2"/>
<tool_change_post_moves/>
</tool_change>
<tool_change>
<tool_change_pre_moves/>
<tool_change_moves NUMBER="3"/>
<tool_change_moves NUMBER="4"/>
<tool_change_post_moves/>
</tool_change>
</machine>
```

Limiting commands to pickup or putdown

You can limit PowerMILL to use commands only when a tool is being picked up or put down:

- Pick up — Use a prefix of GETONLY.

- Put down — Use a prefix if PUTONLY.

Prefixes are followed by a colon, for example:

```
<tool_change_moves NUMBER="1">
MOVETO A0
# Put the tool down and pass it to the tool changer
PUTONLY: TOOL PASS "TOOLCHANGER"
# Pick the tool put and pass it to the machine
GETONLY: TOOLCHANGE
MOVETO A90
</ tool_change_moves>
```

Hiding static components

There is a HIDDEN axis priority that behaves like the STATIC priority but does not show in the **Machine Tool Position** dialog. For example:

```
<machine_part>
<axis>
<control_info ADDRESS="T" HOME="0" VALUE="0" PRIORITY="HIDDEN" />
<simple_rotary X="0" Y="2550" Z="0" I="0" J="0" K="1" />
</axis>
</machine_part>
```

Specifying a tool change time

You can specify the time (in milliseconds) it takes for a tool to move. Add a TIME argument after the MOVETO command, for example:

```
<tool_change_moves NUMBER="1">
MOVETO A0 TIME=50
TOOLCHANGE
MOVETO A90 TIME=50
</tool_change_moves>
```

The time you specify is a target, as collision checking takes precedence over the competition of a move within a specified time.

Machine description

The machine description can include the following information:

- The Machine Tool Manufacturer.
- The Machine Tool Model.
- The Machine Tool Type.
- The Machine Tool Configuration.
- The Linear Axes Limits.
- The Rotary Axes Limits (if these exist).
- The Component Tolerance (`0.025`, `0.05`, `0.25`).
- The name of the person that built the .mtd file.
- The date the .mtd file was built.
- The name of the person that tested the .mtd file (if necessary).
- Any problems that were fixed in the .mtd file.

*This information should be presented as **User Comments**.*

```xml
<?xml version="1.0" ?>
<!-- **************************************************** -->
<!-- Machine Tool Manufacturer     : Hermle                 -->
<!-- Machine Tool Model            : Hermle C12 SRT320      -->
<!-- Machine Tool Type             : 5-Axis Mill            -->
<!-- Machine Tool Configuration : 5-Axis Table-Table     -->
<!-- Linear Axis Limits            : Xmin -175 Xmax +175    -->
<!--                               : Ymin -220 Ymax +220    -->
<!--                               : Zmin +100 Zmax +430    -->
<!-- Rotary Axial Limits           : Amin -115 Amax +115    -->
<!--                               : Bmin -    Bmax -       -->
<!--                               : Cmin -inf Cmax -inf    -->
<!-- Component Tolerance           :   0.025,0.05,0.25      -->
<!-- **************************************************** -->
<!--               Who             : xpolk                  -->
<!--               When            : 23.10.2014             -->
<!--               Checked By      : Name of Colleague      -->
<!--               Checked         : Date Checked           -->
<!--               Revision        : 1.001                  -->
<!--                Why            : _____________________  -->
<!--               Fixed           : _____________________  -->
<!-- **************************************************** -->
<!-- Hermle_C12_SRT320 -->
```

Where:

<!-- opens line of commentary.

--> closes the line of commentary.

If you use the XML then all user comments are be highlighted in green.

See an example of the text below:

```
<!-- **************************************************** -->
<!-- Machine Tool Manufacturer      : Hermle                -->
<!-- Machine Tool Model             : Hermle C12 SRT320     -->
<!-- Machine Tool Type              : 5-Axis Mill           -->
<!-- Machine Tool Configuration     : 5-Axis Table-Table    -->
<!-- Linear Axis Limits             : Xmin -175   Xmax +175 -->
<!--                                : Ymin -220   Ymax +220 -->
<!--                                : Zmin +100   Zmax +430 -->
<!-- Rotary Axial Limits            : Amin -115   Amax +115 -->
<!--                                : Bmin -      Bmax -    -->
<!--                                : Cmin -inf   Cmax -inf -->
```

```
<!-- Component Tolerance                :        0.025, 0.05, 0.25        -->
<!-- ********************************************-->
<!--                          Who            : xpolk                    -->
<!--                          When           : 23.10.2014               -->
<!--                          Checked By     : Name of Colleague        -->
<!--                          Checked        : Date Checked             -->
<!--                          Revision       : 1.002                    -->
<!--                          Why            : Fixed wrong position-->
<!--  12.01.2015               Fixed         : fixed hierarchy          -->
<!-- ********************************************-->
<!-- Hermle_C12_SRT320 -->
```

Description of PostProcessor Solver

Below is a screenshot of the **PostProcessor Solver**:

```
<!-- Hermle_C12_SRT320 -->

<machine xmlns="x-schema:PowermillMachineTool" POST="pmpMultiaxis.*">

```

`POST` is the most important part here as it tells PowerMILL how or where to drive the machine tool simulation. You must have the correct string in the `MTD` file as **Delcam Post Processor (DPP)** is used to drive the `MTD` file. What you see in PowerMILL is also what the post outputs to the machine. Without that string you cannot guarantee that the `MTD` and the posted code match.

Below the text example of PostProcessor Solver:

```
<machine xmlns="x-schema:PowermillMachineTool" POST="pmpMultiaxis.*">
```

Also, without the DPP solver you cannot use the orientation vector functionality.

Description of the attach points

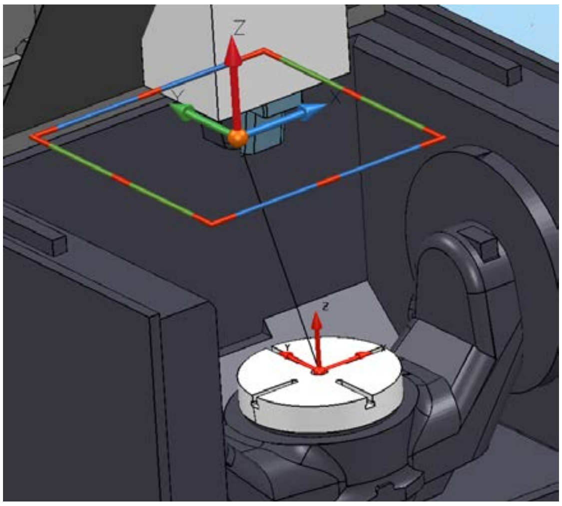
Table Attach Point as written in *.mtd

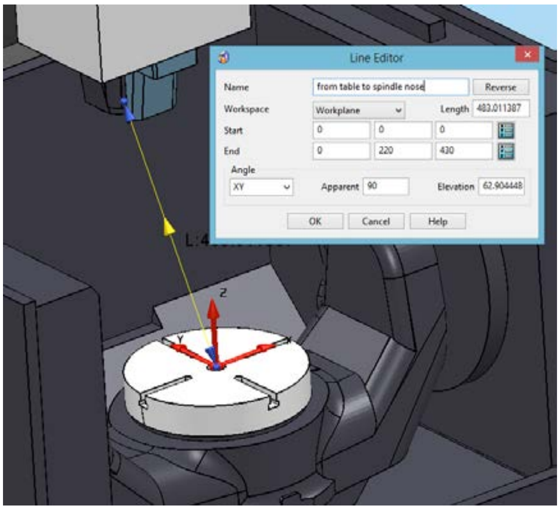
This line defines the centre point of the machine table. The machine assembly done in PowerSHAPE should be set up so that the table centre is placed at the world coordinate, which is 0, 0, 0.

```
<table_attach_point PART="table" X="0" Y="0" Z="0"/>
```

All other parts of the machine tool are then attached with reference to its position.

```
<table_attach_point PART="table" X="0" Y="0" Z="0"/>
<head_attach_point PART="head" X="0" Y="220" Z="430" I="0" J="0" K="1"/>
```





Head Attach Point as written in *.mtd

This line defines the tool attach point. The centre point of the machine spindle gauge face should be given here as the X, Y, Z value and the spindle axis vector should also be given as an I, J, K value.

```
<table_attach_point PART="table" X="0" Y="0" Z="0"/>
<head_attach_point PART="head" X="0" Y="220" Z="430" I="0" J="0" K="1"/>
```

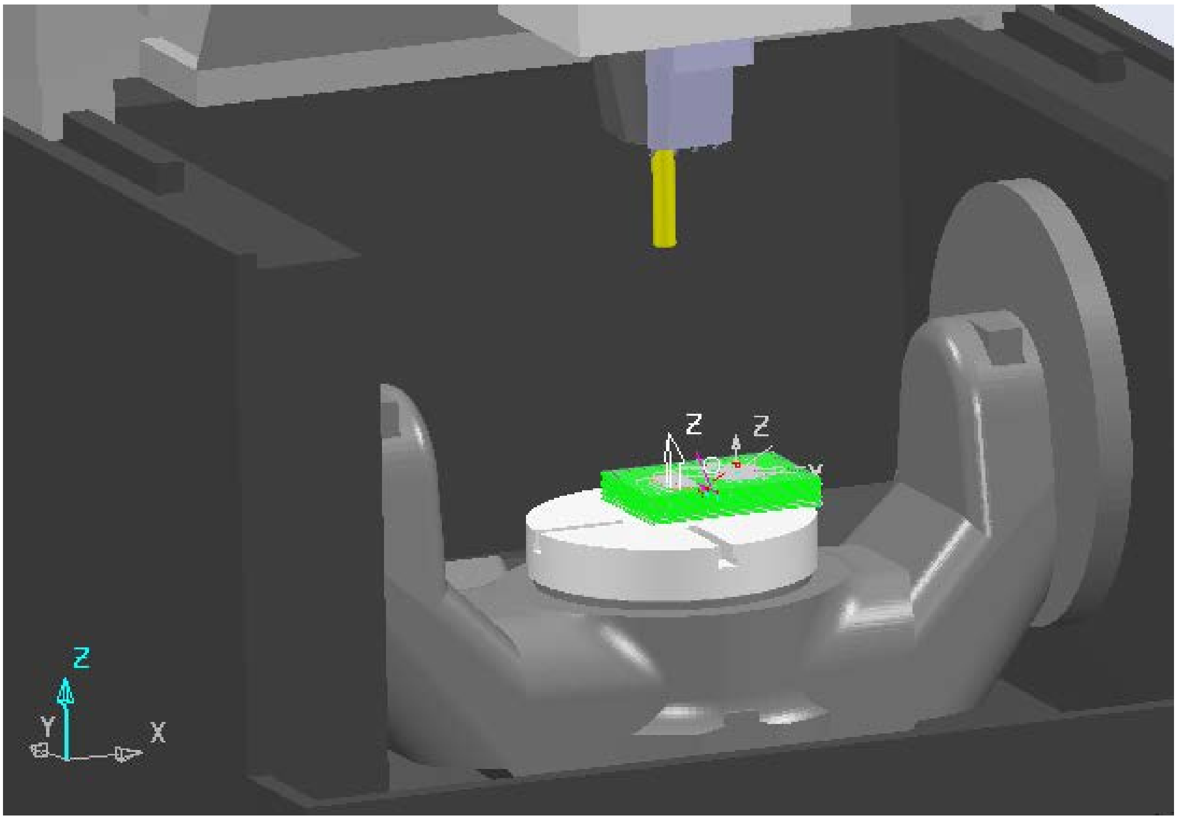
The vectors I, J, and K define the tool direction:

- To achieve a tool axis direction along the positive Z axis:
    - I=0
    - J=0
    - K=1
- To achieve a tool axis direction along the positive Y axis:

- I=0
- J=1
- K=0

- To achieve a tool axis direction along the positive X axis:
  - I=1
  - J=0
  - K=0

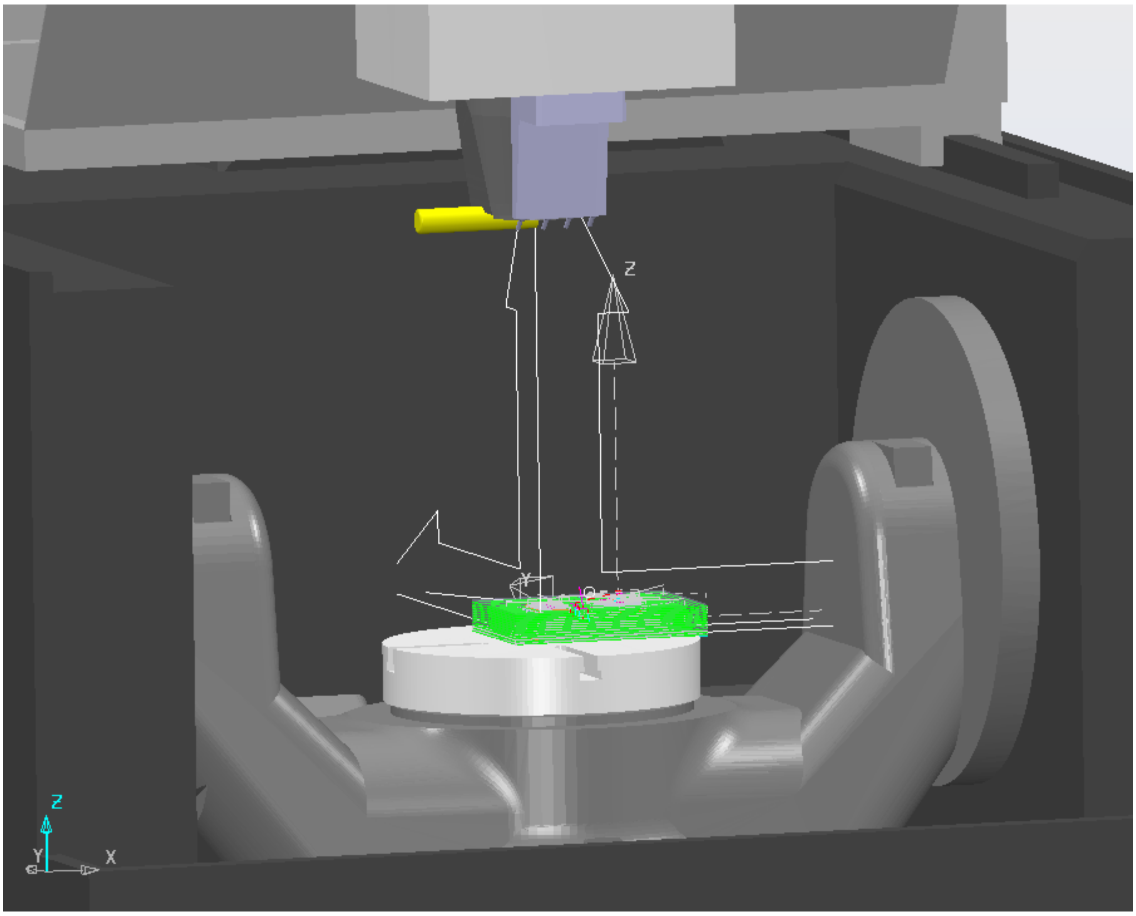
In this case:
<head_attach_point PART="head" X="0" Y="220" Z="430" I="0" J="0" K="1"/>



If you change the direction of the vectors in line head_attach_point for this machine, I="0" J="0" K="1" on I="1" J="0" K="0" then this gives:

```
<table_attach_point PART="table" X="0" Y="0" Z="0"/>
<head_attach_point PART="head" X="0" Y="220" Z="430" I="1" J="0" K="0"/>
```

This gives the following result:

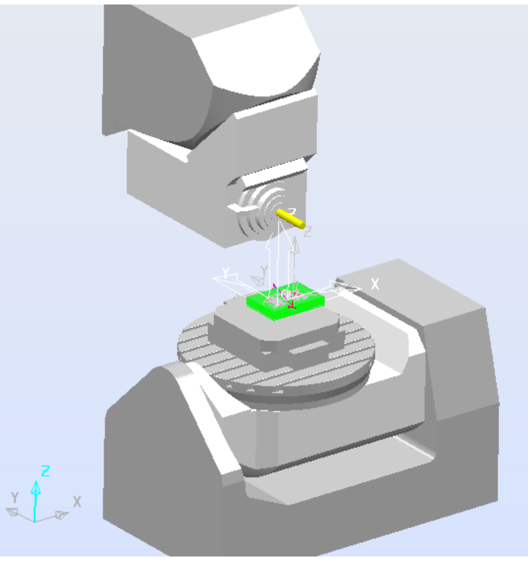


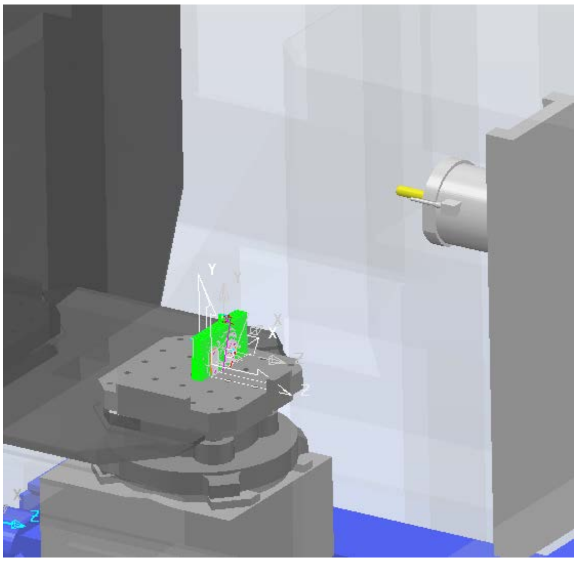
The tool (yellow cylinder) is in a horizontal position.

This is the wrong tool position for this machine. It is very important to determine correct position and direction for `table_attach_point` and `head_attach_point`.

You can change the value of the `I`, `J` and `K` vectors to create *.mtd machines with non-standard kinematics.

The following examples show DMG and Okuma machine tools:





Do not use capital letters in the `attach_point` description.

```
<table_attach_point PART="Table" X="0" Y="0" Z="0"/>
<head_attach_point PART="Head" X="0" Y="220" Z="430" I="1" J="0" K="0"/>
```

```
<table_attach_point PART="table" X="0" Y="0" Z="0"/>
<head_attach_point PART="head" X="0" Y="220" Z="430" I="1" J="0" K="0"/>
```

Tool change simulation is available from PowerMILL 2015 R2. This is an important function for difficult projects.

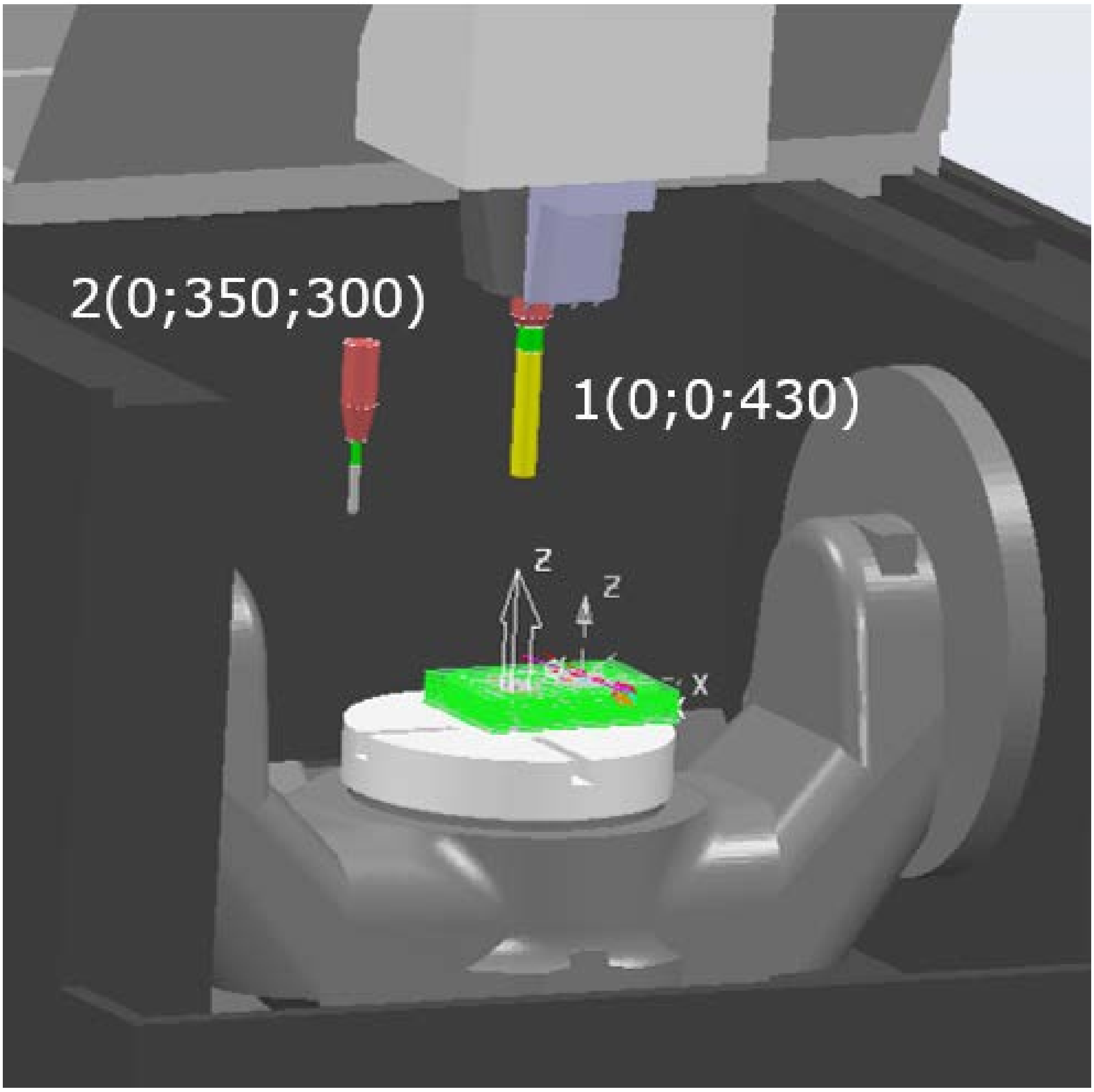
Creating the tool-change point.

```
<tool_change>
   <tool_change_pre_moves>
         MOVETO X0 Y350 Z300 A0 C0
   </tool_change_pre_moves>
    <tool_change_post_moves>
         WAIT 200
         MOVETO X0 Y350
         WAIT 200
         MOVETO Z300
    </tool_change_post_moves>
    <tool_change_moves>
         MOVETO X0 Y350
         WAIT 200
         TOOLCHANGE
         MOVETO X0
         WAIT 200
    </tool_change_moves>
</tool_change>
```

Where:

- Command `MOVETO X0 Y350 Z300 A0 C0` describes the position and direction of the point of tool changing.

- Command `WAIT 200` indicates the delay in milliseconds



Point number one indicates the position of the `head_attach_point`.

Point number two indicates the position of the `tool_change`.

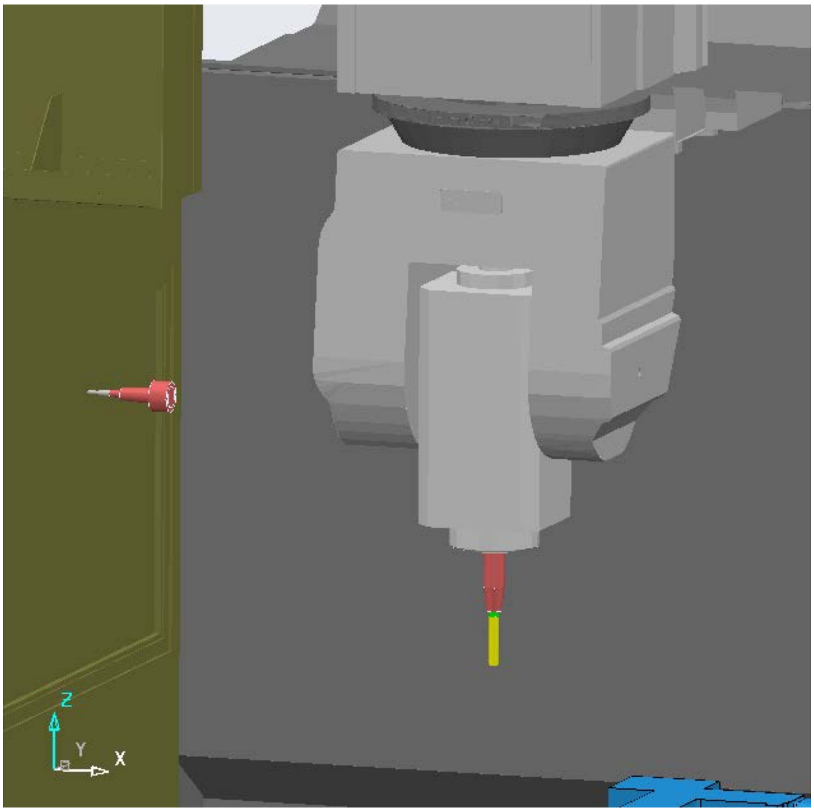
The Tool_change on the next picture (in XML language) is:

```xml

<table_attach_point PART="table" X="0" Y="0" Z="0"/>
<head_attach_point PART="head" X="0" Y="220" Z="430" I="0" J="0" K="1"/>

<tool_change>
    <tool_change_pre_moves>
        MOVETO X0 Y350 Z300 A0 C0
    </tool_change_pre_moves>
    <tool_change_post_moves>
        WAIT 200
        MOVETO X0 Y350
        WAIT 200
        MOVETO Z300
    </tool_change_post_moves>
    <tool_change_moves>
        MOVETO X0 Y350
        WAIT 200
        TOOLCHANGE
        MOVETO X0
        WAIT 200
    </tool_change_moves>
</tool_change>
```

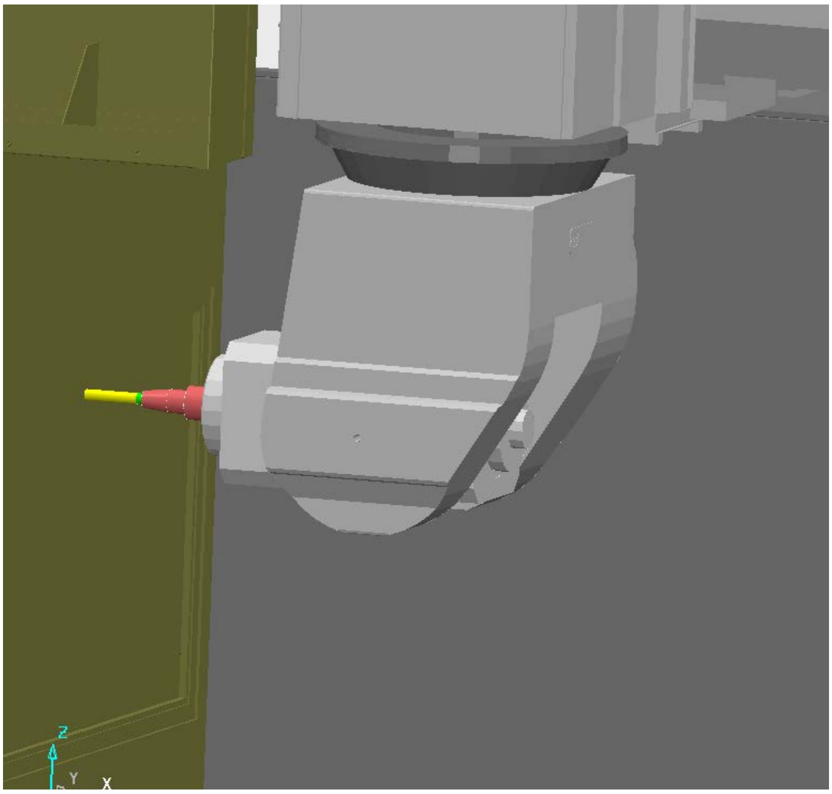
The tool change command is: MOVETO X735 Y0 Z500 B-90 C270.

```xml
<tool_change>
    <tool_change_pre_moves>
        MOVETO X735 Y0 Z500 B-90 C270
    </tool_change_pre_moves>
    <tool_change_post_moves>
        WAIT 200
        MOVETO X735 Y0
        WAIT 200
        MOVETO Z500
    </tool_change_post_moves>
    <tool_change_moves>
        MOVETO X750 Y0
        WAIT 200
        TOOLCHANGE
        MOVETO X745
        WAIT 200
    </tool_change_moves>
</tool_change>
```

Before changing the tool:



At the time of changing the tool:



Description of static elements

The next step is to define the machine parts. First, define which parts of the machine tool are static, such as the **Base**, **Housing** and the **Controller**. They need to be defined in order to visually represent the machine tool.

All static parts of the machine should be described in the block `The static bits`.

```
<!-- ===== The static bits ===== -->
        <machine_part>
            <model_list OPACITY="10">
                <dmt_file>
                    <path FILE="Hermle_C12_SRT320/housing.dmt" />
                    <rgb R="250" G="250" B="250"/>
                </dmt_file>
                <dmt_file>
                    <path FILE="Hermle_C12_SRT320/door.dmt" />
                    <rgb R="240" G="240" B="240"/>
                </dmt_file>
                <dmt_file>
                    <path FILE="Hermle_C12_SRT320/handle.dmt" />
                    <rgb R="255" G="0" B="0"/>
                </dmt_file>
            </model_list>
        </machine_part>
        <machine_part>
            <model_list>
                <dmt_file>
                    <path FILE="Hermle_C12_SRT320/logo_1.dmt" />
                    <rgb R="255" G="0" B="0"/>
                </dmt_file>
            </model_list>
        </machine_part>
        <machine_part>
            <model_list>
                <dmt_file>
                    <path FILE="Hermle_C12_SRT320/logo_2.dmt" />
                    <rgb R="200" G="200" B="200"/>
                </dmt_file>
            </model_list>
        </machine_part>
    <machine_part>
        <model_list>
            <dmt_file>
                <path FILE="Hermle_C12_SRT320/base.dmt" />
                <rgb R="80" G="80" B="80"/>
            </dmt_file>
        </model_list>
    </machine_part>
```

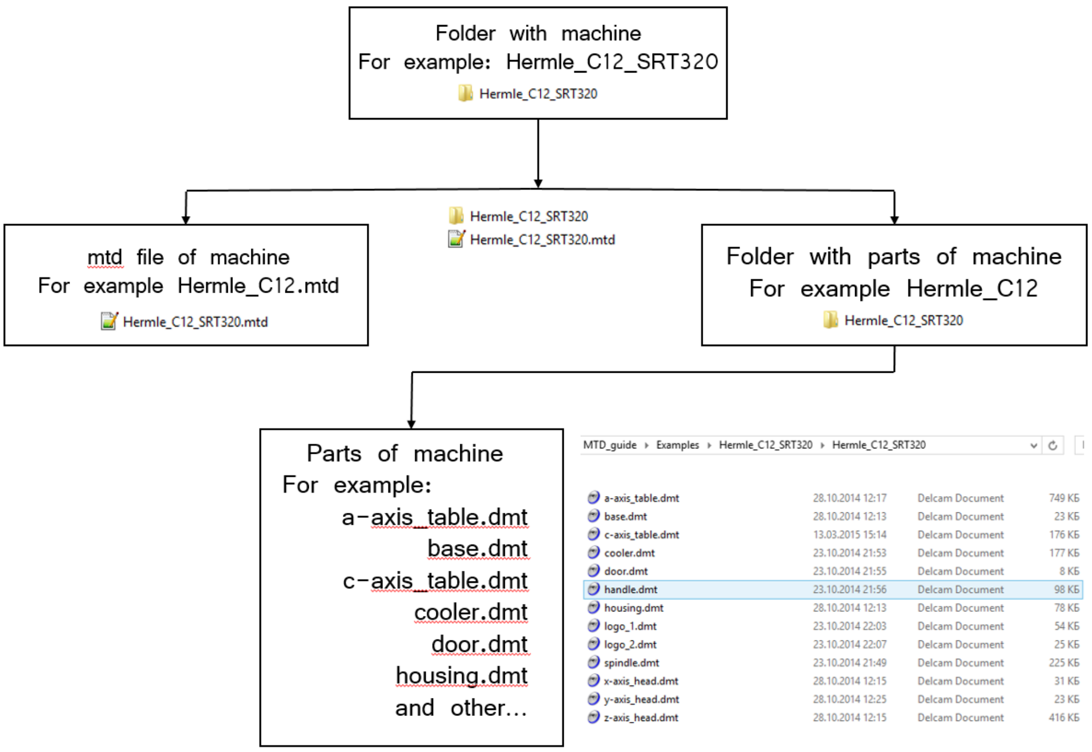
Any part of the machine is loaded into .mtd * using the following commands:

```
<machine_part>
<model_list>
<dmt_file>
<path FILE="machine/part_1.dmt"/>
<rgb R="0" G="140" B="255"/>
</dmt_file>
</model_list>
</machine_part>
```

Where:

- <path FILE="machine/part_1.dmt"/> . This line gives the location of the machine part.
- <rgb R="0" G="140" B="255"/> . This line gives the colour of machine part.

A colour in the R (red) G (green) B (blue) colour model is described by indicating how much of each of the red, green, and blue is included. The colour is expressed as an RGB triplet (r, g, b), each component of which can vary from zero to a defined maximum value. If all the components are at zero the result is black; if all are at maximum (255), the result is the brightest representable white.



You can include all static parts on one <machine_part>:

```
<machine_part>
<model_list>
<dmt_file>
<path FILE="machine/part_1.dmt"/>
<rgb R="0" G="140" B="255"/>
</dmt_file>
<dmt_file>
<path FILE="machine/part_2.dmt"/>
<rgb R="0" G="0" B="0"/>
</dmt_file>
<dmt_file>
<path FILE="machine/part_3.dmt"/>
<rgb R="0" G="140" B="255"/>
```

```
</dmt_file>
<dmt_file>
<path FILE="machine/part_4.dmt"/>
<rgb R="0" G="140" B="255"/>
</dmt_file>
<dmt_file>
<path FILE="machine/part_5.dmt"/>
<rgb R="0" G="140" B="255"/>
</dmt_file>
</model_list>
</machine_part>
```

Alternatively, you can use for each static element a personal <machine_part>:

```
<machine_part>
<model_list>
<dmt_file>
<path FILE="machine/part_1.dmt"/>
<rgb R="0" G="140" B="255"/>
</dmt_file>
</model_list>
</machine_part>
<machine_part>
<model_list>
<dmt_file>
<path FILE="machine/part_2.dmt"/>
<rgb R="0" G="0" B="0"/>
</dmt_file>
</model_list>
</machine_part>
<machine_part>
<model_list>
<dmt_file>
<path FILE="machine/base.dmt"/>
<rgb R="60" G="80" B="255"/>
</dmt_file>
</model_list>
</machine_part>
```

Use the most important static part at the end of static block. For example: base, table, pallet etc.

Below is an example of the full text of the static block:

```
<!-- ===== The static bits ===== -->
<machine_part>
<model_list OPACITY="10">
<dmt_file>
<path FILE="Hermle_C12_SRT320/housing.dmt"/>
<rgb R="250" G="250" B="250"/>
</dmt_file>
<dmt_file>
<path FILE="Hermle_C12_SRT320/door.dmt"/>
<rgb R="240" G="240" B="240"/>
</dmt_file>
<dmt_file>
<path FILE="Hermle_C12_SRT320/handle.dmt"/>
<rgb R="255" G="0" B="0"/>
</dmt_file>
</model_list>
</machine_part>
<machine_part>
<model_list>
<dmt_file>
<path FILE="Hermle_C12_SRT320/logo_1.dmt"    />
<rgb R="255" G="0" B="0"/>
```

```
</dmt_file>
</model_list>
</machine_part>
<machine_part>
<model_list>
<dmt_file>
<path FILE="Hermle_C12_SRT320/logo_2.dmt" />
<rgb R="200" G="200" B="200"/>
</dmt_file>
</model_list>
</machine_part>
<machine_part>
<model_list>
<dmt_file>
<path FILE="Hermle_C12_SRT320/base.dmt" />
<rgb R="80" G="80" B="80"/>
</dmt_file>
</model_list>
</machine_part>
```

Additional commands for static block:

- Transparency or opacity of machine parts (if necessary).
- Collision detection.

Transparency or opacity of the machine parts (if necessary)

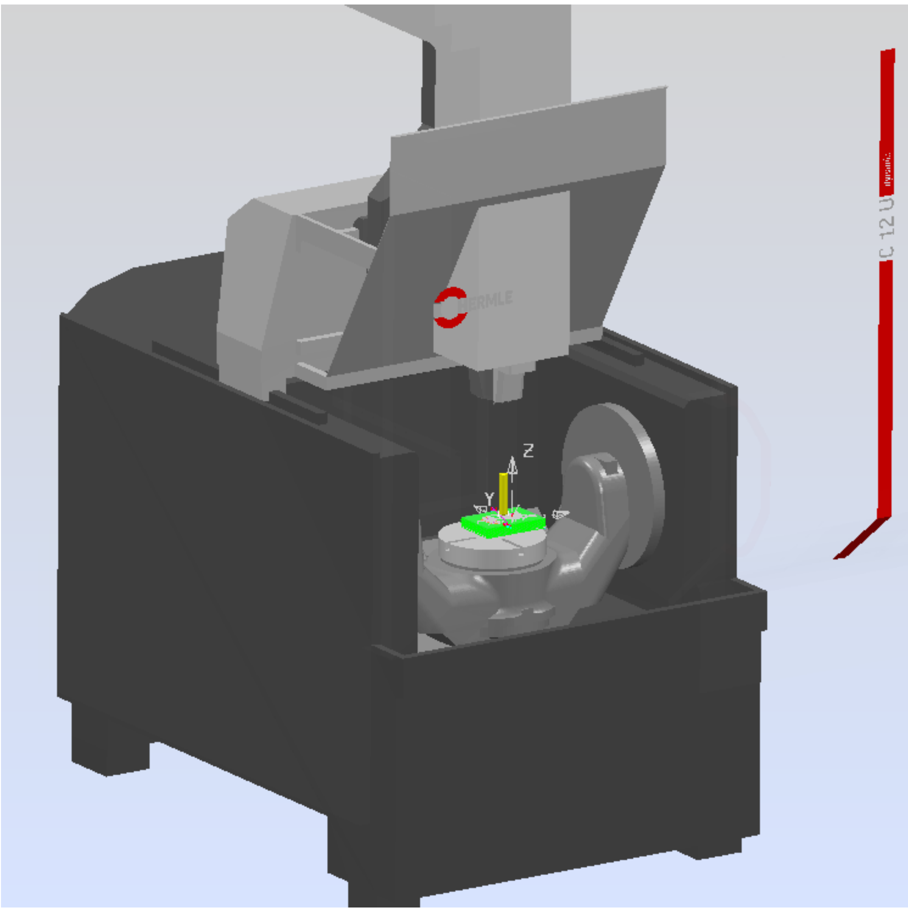
Use the next command:

`<model_list OPACITY="XX">` where XX can be from 0 up to 100.

This command is used for obtaining a partly-transparency:

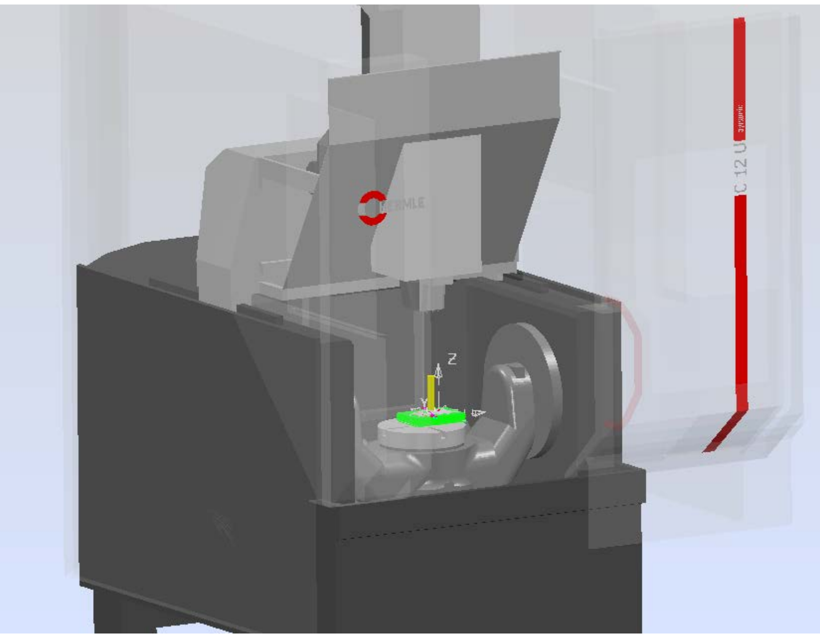
- `0` — 100% transparent.
- `10` — 90% transparent.
- `100` — 0% transparent

For example, enter `<model_list OPACITY="0">` (giving 100% transparency):



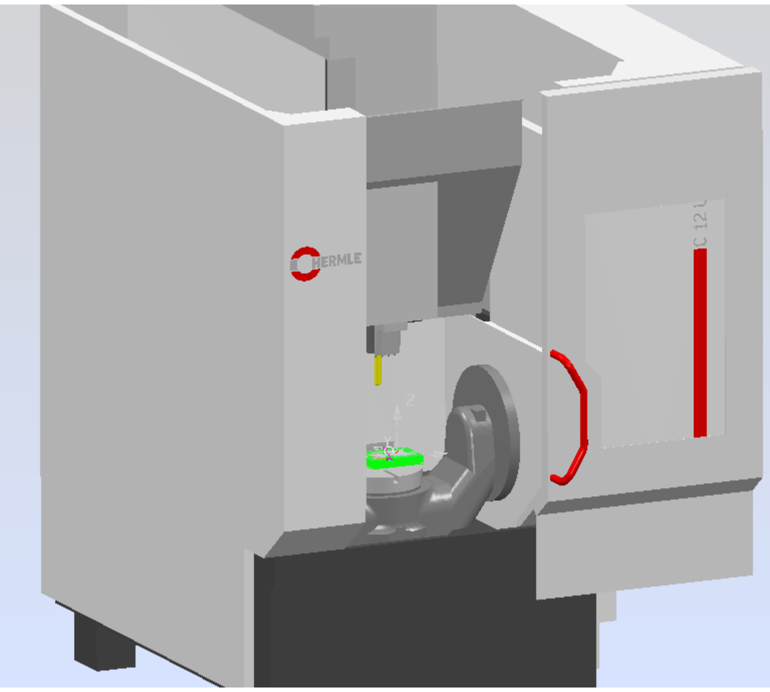
```
<!-- ===== The static bits ===== -->
    <machine_part>
        <model_list OPACITY="0">
            <dmt_file>
                <path FILE="Hermle_C12_SRT320/housing.dmt" />
                <rgb R="250" G="250" B="250"/>
            </dmt_file>
```

`<model_list OPACITY="10">` (giving 90% transparency):




```
===== The static bits ===== -->
    <machine_part>
        <model_list  OPACITY="10">
            <dmt_file>
                <path FILE="Hermle_C12_SRT320/housing.dmt" />
                <rgb R="250" G="250" B="250"/>
```

`<model_list OPACITY="100">` (giving 0% transparency):



```
<!-- ===== The static bits ===== -->
    <machine_part>
        <model_list OPACITY="100">
            <dmt_file>
                <path FILE="Hermle_C12_SRT320/housing.dmt" />
                <rgb R="250" G="250" B="250"/>
            </dmt_file>
```

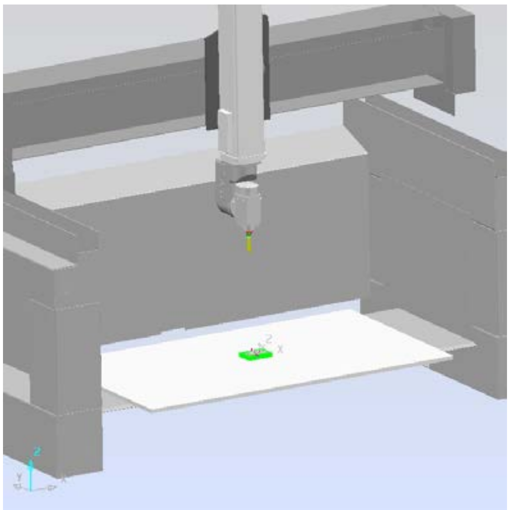
All machine parts, that are included in such `<model_list OPACITY="10">` have the same transparency. If you want use different transparency then use different `<model_list OPACITY="XX">`.

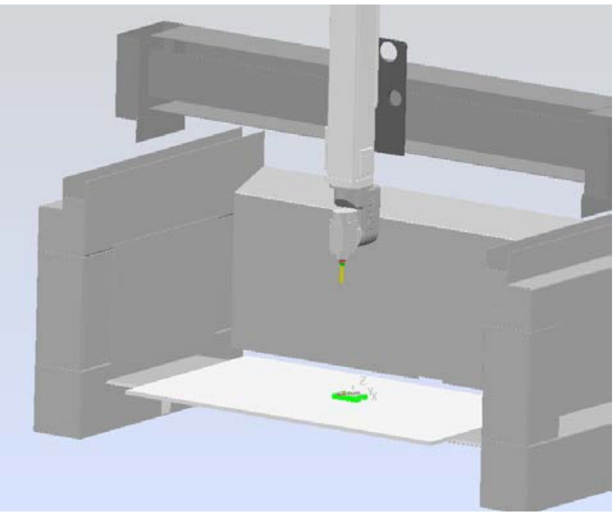
Collision detection

The static parts are used for more detailed visualization of the machine and determining the correct workspace for simulation. Simulation of movements of the machine is very important for detection of collision for all types of machines tools (3-axis, 4-axis, 5-axis and others).

PowerMILL does not detect collisions with static elements (it is dangerous, especially for the five-axis machine tools).

See the next example with **CMS_Antares Head-head** machine (the table part is not moveable).

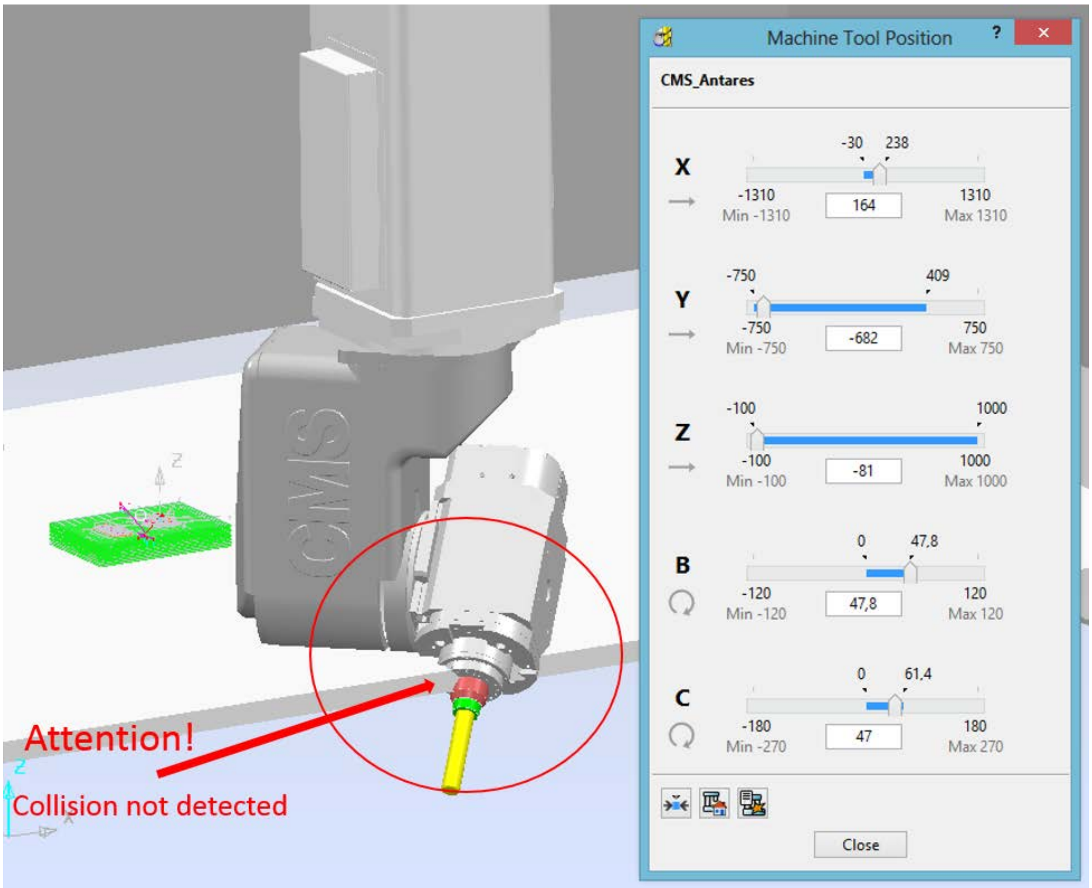




The following shows the static block text, where pallet.dmt is the upper surface of the table.

```
<!-- Static bits -->
    <machine_part>
        <model_list>
            <dmt_file>
                <path FILE="CMS_Antares/Carter.dmt" />
                <rgb R="200" G="200" B="200" />
            </dmt_file>
            <dmt_file>
                <path FILE="CMS_Antares/pallet.dmt" />
                <rgb R="255" G="255" B="255" />
            </dmt_file>
        </model_list>
    </machine_part>
<machine_part NAME="table" />
<!-- The head -->
```

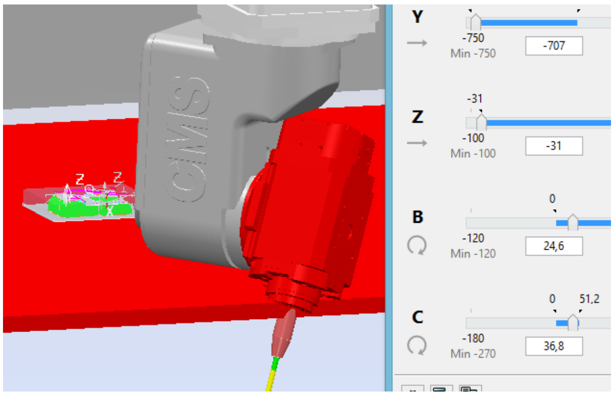
Using **Machine Tool Position** in PowerMILL enables you to change the Z axis position. If you move the Z axis down (simulation of collision Z-axis_head with a table), the collision is not detected in PowerMILL.

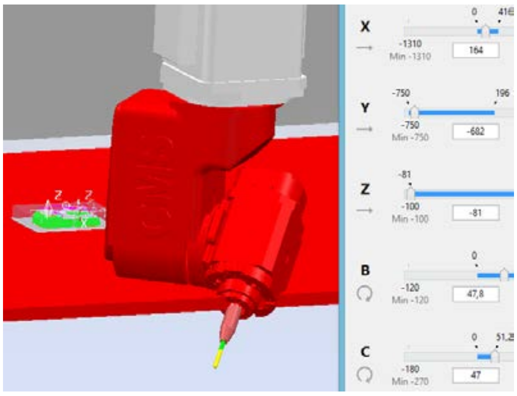


You must change the description of pallet.dmt.

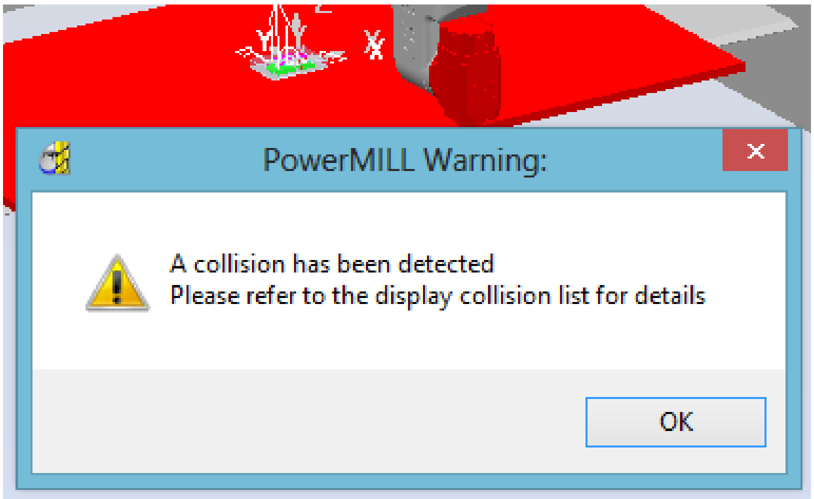
```
<machine_part NAME="pallet">
<model_list>
<dmt_file>
<path FILE="CMS_Antares/pallet.dmt" />
<rgb R="255" G="255" B="255" />
</dmt_file>
</model_list>
</machine_part>
```

When the NAME parameter is added to the <machine_part> description, PowerMILL detects the collision correctly:





When a collision occurs, the colliding parts of machine are coloured and a warning message is displayed:



All static parts of machine, which are located at working area, must be described as above.

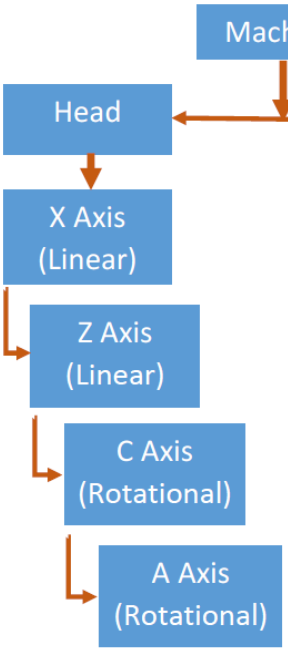
Description of the movable elements

This topic looks at the main types of machine tools (kinematics and structure). Every *.mtd that you create differs, depending on the configuration of the machine tool. This is important for 5 axis machine tools, where the axes are defined in the *.mtd as head or table entities. For 5 axis machines there will be 3 different configurations:

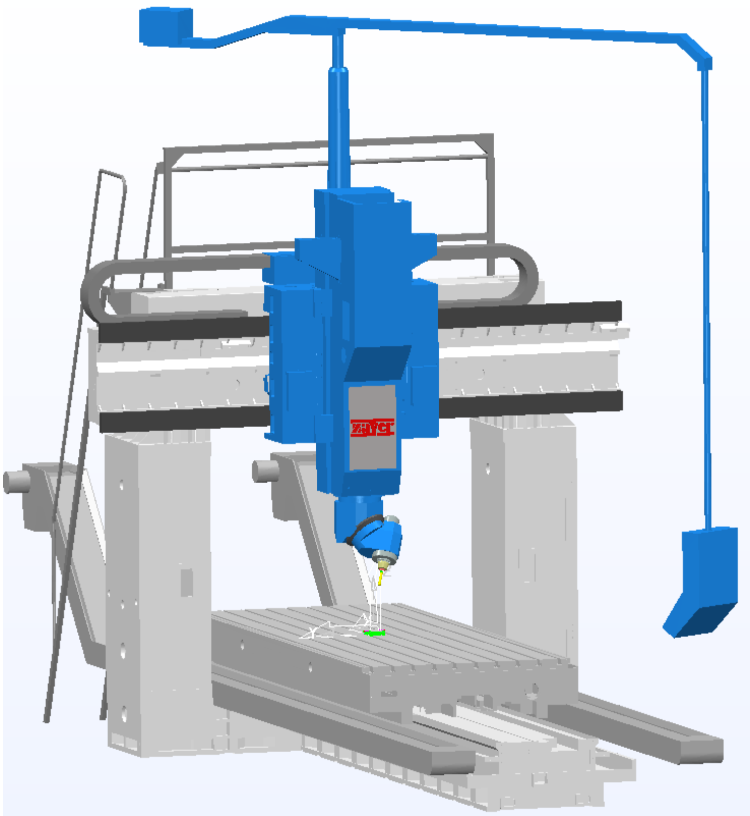
- Head-Head.
- Head-Table.
- Table-Table (Trunion).

Below are three graphs showing how the different configurations can be split into head and table components. All machine tools are not limited to just these configurations; they are just given as a guide:

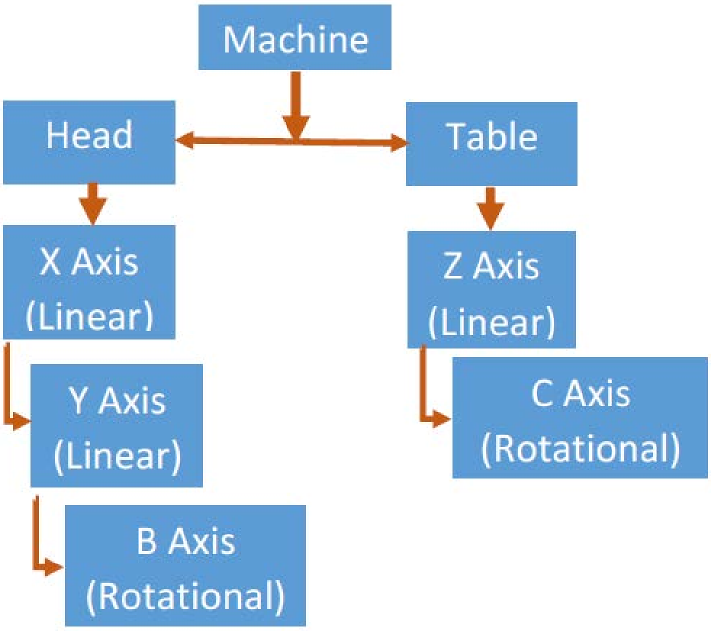
Head-Head



Example of **Head-Head – Zayer FPC-AR**:



Head-Table



Example of **Head-Table – DMG DMU80 monoBLOCK**:

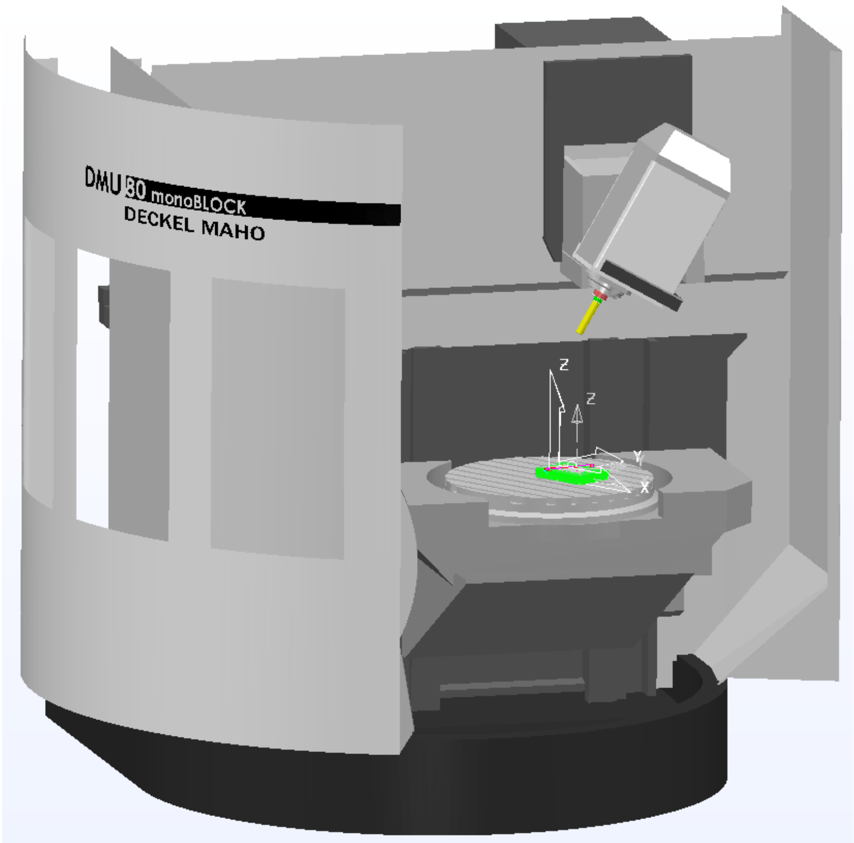
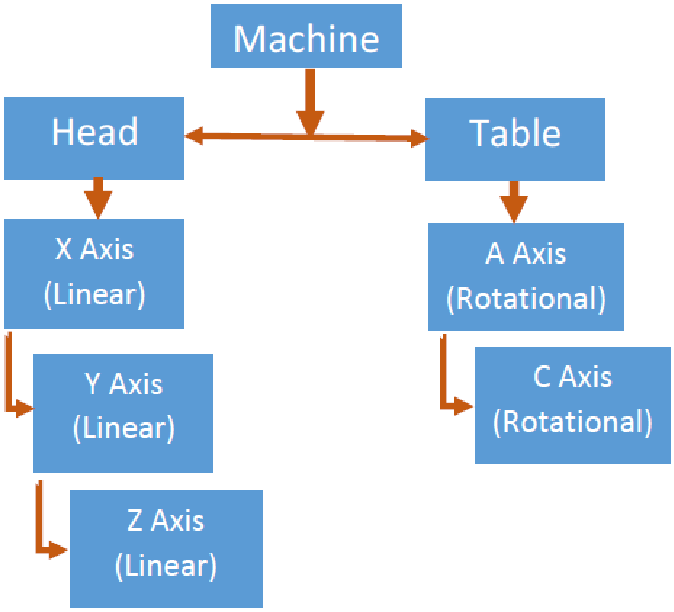
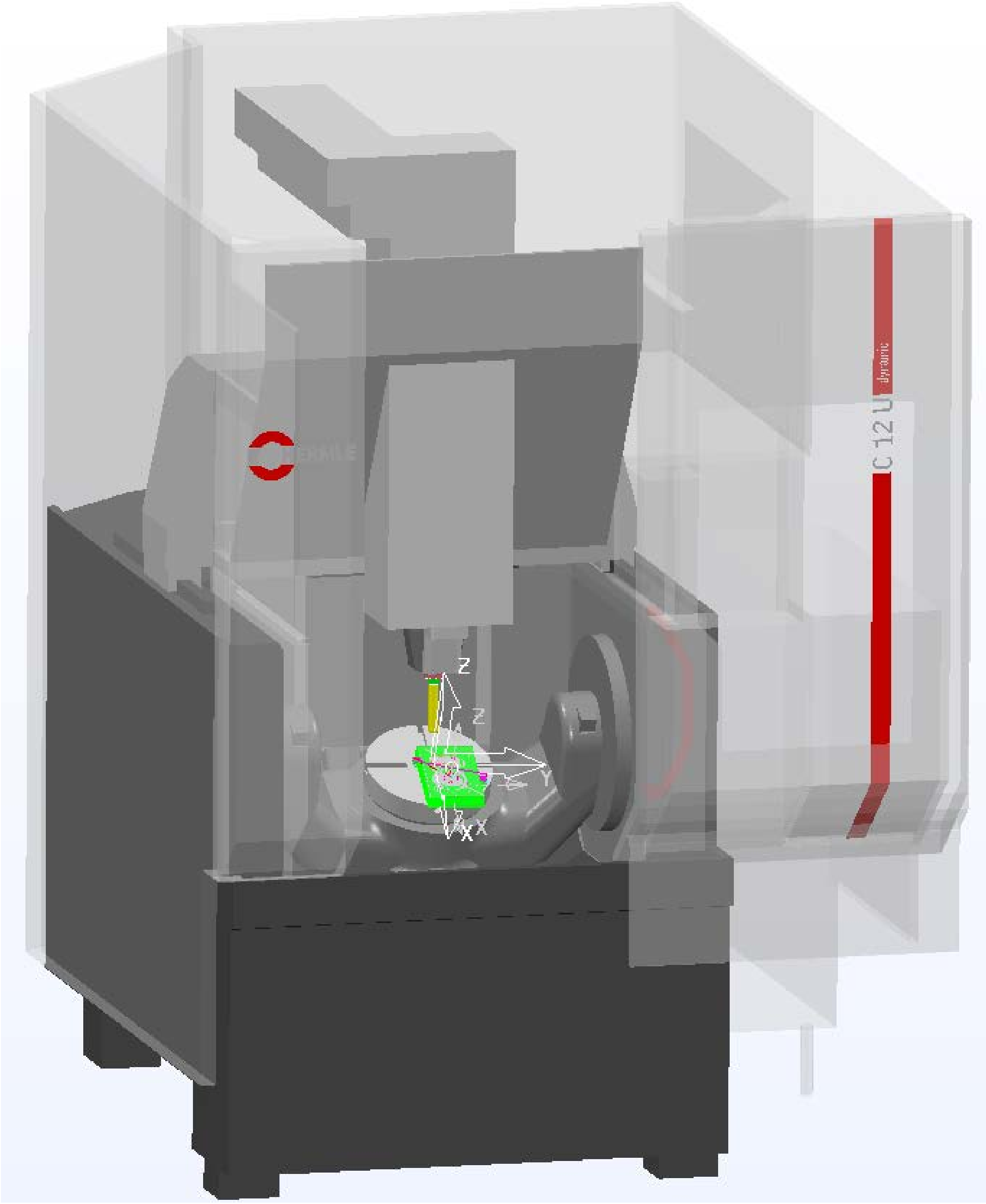


Table-Table (Trunion)



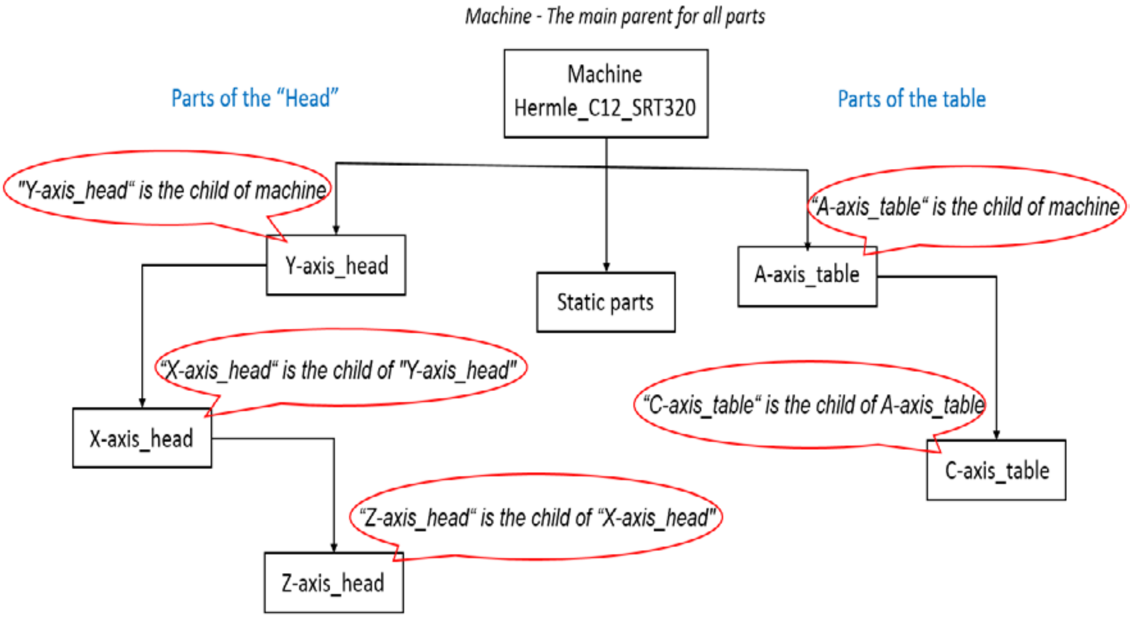
Example of **Table-Table Hermle_C12_SRT320** (Table 320):



Now you can move on to creating the <machine_part> definitions. As mentioned before, there are two <machine_part> definitions: one for the head of the machine and one for the table. This example defines a machine tool that has a Table-Table configuration.

As a result of this configuration, the 'Head' machine part consists of elements making up the X, Y and Z axial components. The 'Table' machine part consists of elements making up the A and C axial components. See previous picture (**Hermle_C12_SRT320**).

Let us consider hierarchy of machine movements:



This example begins with the Y axis. The component model that makes up the Y axis acts as the parent component upon which the remaining axial components for X and Z are attached. You must give each axial component a definition:

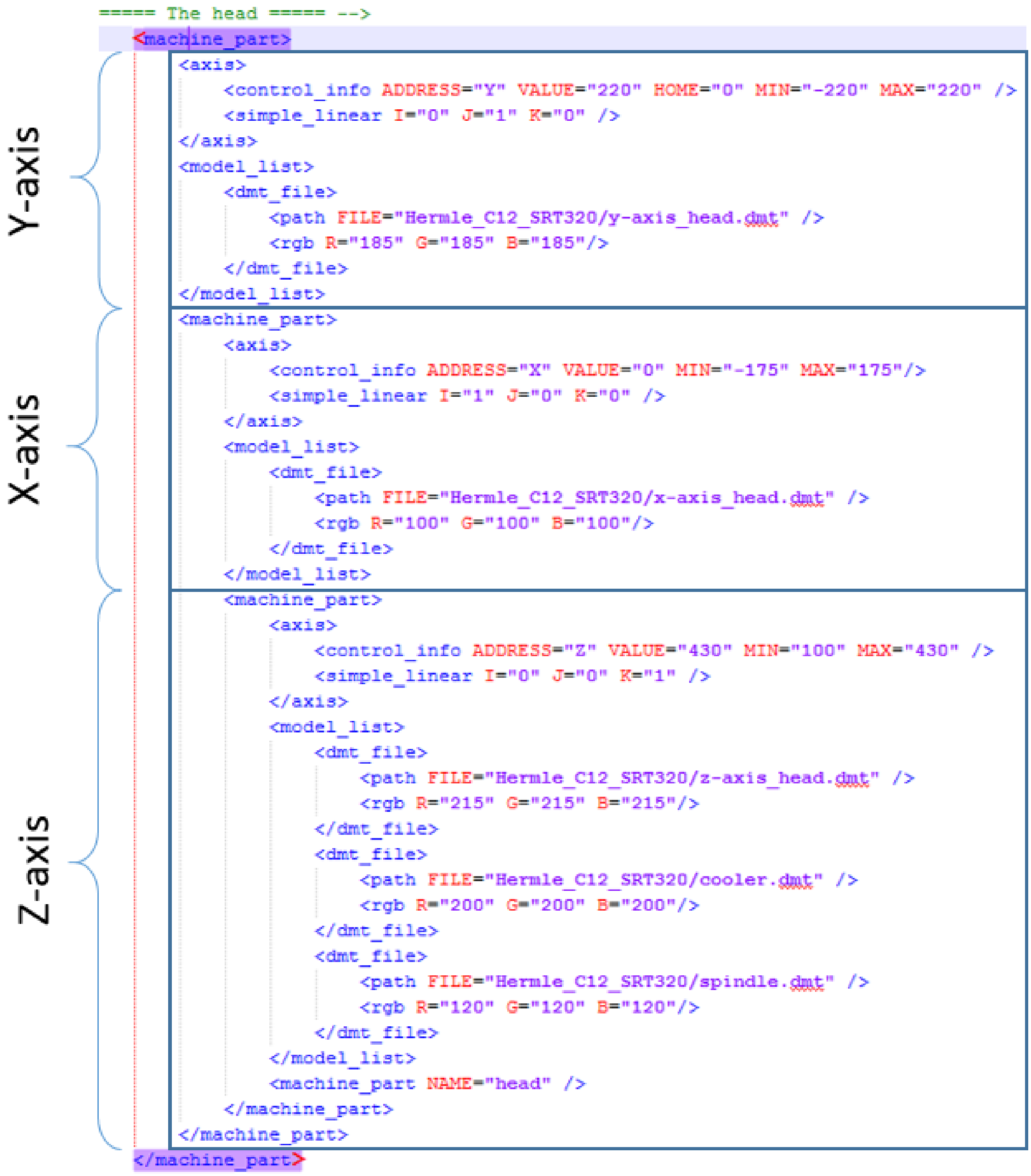
<axis>: Used to define an axis. Within this we need to give an axis a name, a value, limits and a movement vector.

```
<axis>
    <control_info ADDRESS="Y" VALUE="220" HOME="0" MIN="-220" MAX="220" />
    <simple_linear I="0" J="1" K="0" />
</axis>
```

Use the following commands to define the axis:

- control_info ADDRESS — This simply defines the axis name as X, Y, Z, A, B, C or any other names.
- VALUE — This is the value that is shown in the Machine Information Dialog when the machine is in its home position.
- MIN and MAX — These are the limits that the axis can travel to.
- PRIORITY — Defines the priority of movements, respectively "HIGH", "MEDIUM" and "LOW".
- HOME — Assigns a new correct position of "home position" (if VALUE is wrong).
- simple_linear — Defines how an axis moves if it is only for linear movement using vectors I, J or K.
- simple_rotary — Defines how an axis moves only for rotational movement. You must define the centre of rotation and a vector to define the direction of rotation.

How they all fit together in our Hermle machine tool example:




```
===== The head ===== -->
<machine_part>
<axis>
    <control_info ADDRESS="Y" VALUE="220" HOME="0" MIN="-220" MAX="220" />
    <simple_linear I="0" J="1" K="0" />
</axis>
<model_list>
    <dmt_file>
        <path FILE="Hermle_C12_SRT320/y-axis_head.dmt" />
        <rgb R="185" G="185" B="185"/>
    </dmt_file>
</model_list>
<machine_part>
    <axis>
        <control_info ADDRESS="X" VALUE="0" MIN="-175" MAX="175"/>
        <simple_linear I="1" J="0" K="0" />
    </axis>
    <model_list>
        <dmt_file>
            <path FILE="Hermle_C12_SRT320/x-axis_head.dmt" />
            <rgb R="100" G="100" B="100"/>
        </dmt_file>
    </model_list>
    <machine_part>
        <axis>
            <control_info ADDRESS="Z" VALUE="430" MIN="100" MAX="430" />
            <simple_linear I="0" J="0" K="1" />
        </axis>
        <model_list>
            <dmt_file>
                <path FILE="Hermle_C12_SRT320/z-axis_head.dmt" />
                <rgb R="215" G="215" B="215"/>
            </dmt_file>
            <dmt_file>
                <path FILE="Hermle_C12_SRT320/cooler.dmt" />
                <rgb R="200" G="200" B="200"/>
            </dmt_file>
            <dmt_file>
                <path FILE="Hermle_C12_SRT320/spindle.dmt" />
                <rgb R="120" G="120" B="120"/>
            </dmt_file>
        </model_list>
        <machine_part NAME="head" />
    </machine_part>
</machine_part>
</machine_part>
```

This Hermle machine tool has:

Centre of table: X=0; Y=0; Z=0.

Centre of spindle: X=0; Y=220; Z=430.

Below is a text of description of the Y-axis:

```
<machine_part>
<axis>
   <control_info ADDRESS="Y" VALUE="220" HOME="0" MIN="-220" MAX="220"
/>
   <simple_linear I="0" J="1" K="0" />
</axis>
   <model_list>
 <dmt_file>
```

```
                <path FILE="Hermle_C12_SRT320/y-axis_head.dmt" />
<rgb R="185" G="185" B="185"/>
            </dmt_file>
    </model_list>
```

Use the following commands:

- <machine_part> — Create (open) the part of the .mtd.
- <axis> and </axis> — Open and close the description of the axis.
- <control_info ADDRESS="Y" — Create a Y axis. "Y" is the name of the axis.
- VALUE="220" — A command that states that the current position of the Y axis is 220 mm (data from PoweSHAPE).
- HOME="0" — A command that returns the Y-axis to the "0" position (centre of table); If you use HOME="-100", then the position of Y-axis is Y=-100. The command HOME is used to give the correct position for all axial parts of machine (origin position).
- MIN="-220" and MAX="220" — Minimum and maximum limits of axis movement.
- <simple_linear … /> — Create the linear axis.
- I="0" J="1" K="0" /> — Direction of movement; if I="1" J="0" K="0" then motion occurs along the X axis, if I="0" J="1" K="0" then motion takes along Y axis, if I="0" J="0" K="1" then motion occurs along the Z-axis.
- <model_list> and </model_list> — Open and close the line of model.
- <dmt_file> and </dmt_file> — Open and close the file.
- <path FILE="Hermle_C12_SRT320/y-axis_head.dmt" /> — The address where the file is located.
- <rgb R="185" G="185" B="185"/> — The colour scheme of the machine part: R-red, G-green, B-blue.
- </machine_part> close the part of mtd.

Additional commands, that describe the rotary parts of the machine.

```
<axis>
    <control_info ADDRESS="A" MIN="-115" MAX="115" />
    <simple_rotary X="0" Y="0" Z="80" I="-1" J="0" K="0" />
</axis>
```

Such commands as:

- <simple_rotary /> — Create the rotary axis.
- X="0" Y="0" Z="80" — The coordinates of the centre of rotation relative to the global coordinate system (table_attach_point).
- I="-1" J="0" K="0" — Direction of movement; if I="1" J="0" K="0" then rotation occurs around the X axis, if I="0" J="1" K="0" then rotation occurs around the Y axis, if I="0" J="0" K="1" then rotation occurs around the Z axis.

Minus indicates at the direction of rotation, if I="-1" J="0" K="0" then rotation occurs around X axis in negative direction.

Full details about a coordinates and direction of rotation are given in a later section.

Each machine axis can contain several parts of the machine, for example Z-axis_head. Traditionally Z-axis included such parts as: Z-axis, the spindle, cooling equipment and the logo:

```
<machine_part>
    <axis>
        <control_info ADDRESS="Z" VALUE="430" MIN="100" MAX="430" />
        <simple_linear I="0" J="0" K="1" />
    </axis>
    <model_list>
        <dmt_file>
            <path FILE="Hermle_C12_SRT320/z-axis_head.dmt" />
            <rgb R="215" G="215" B="215"/>
        </dmt_file>
        <dmt_file>
            <path FILE="Hermle_C12_SRT320/cooler.dmt" />
            <rgb R="200" G="200" B="200"/>
        </dmt_file>
        <dmt_file>
            <path FILE="Hermle_C12_SRT320/spindle.dmt" />
            <rgb R="120" G="120" B="120"/>
        </dmt_file>
    </model_list>
    <machine_part NAME="head" />
</machine_part>
```

You can use different colours for each part, but can't use different OPACITY. OPACITY are applied for all parts in one <model_list>.

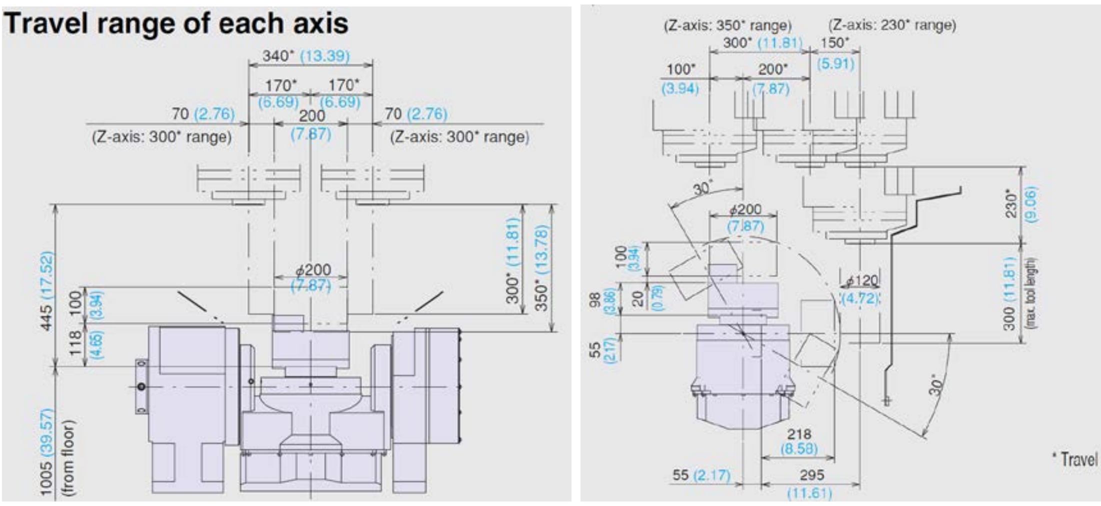
VALUE does not always need to be presented in origin position. You can use the command HOME to define this position.

The most important parameter for each machine is distance between the upper surface table and the spindle nose. This distance is called Spindle Nose to Table. These values define the limits of the Z-axis: Zmin and **Zmax**, relative the centre of the table (X=0; Y=0; Z=0).

For example, specification **Haas VF-2**:

TRAVELS	S.A.E.	METRIC
X Axis	30 "	762 mm
Y Axis	16 "	406 mm
Z Axis	20 "	508 mm
Spindle Nose to Table (~ max)	24 "	610 mm
Spindle Nose to Table (~ min)	4 "	102 mm

Moreover, the information about the Spindle Nose to Table can be given by drawings stating the travel range of each axis. Below is an example of **Okuma_Millac_33TU**.

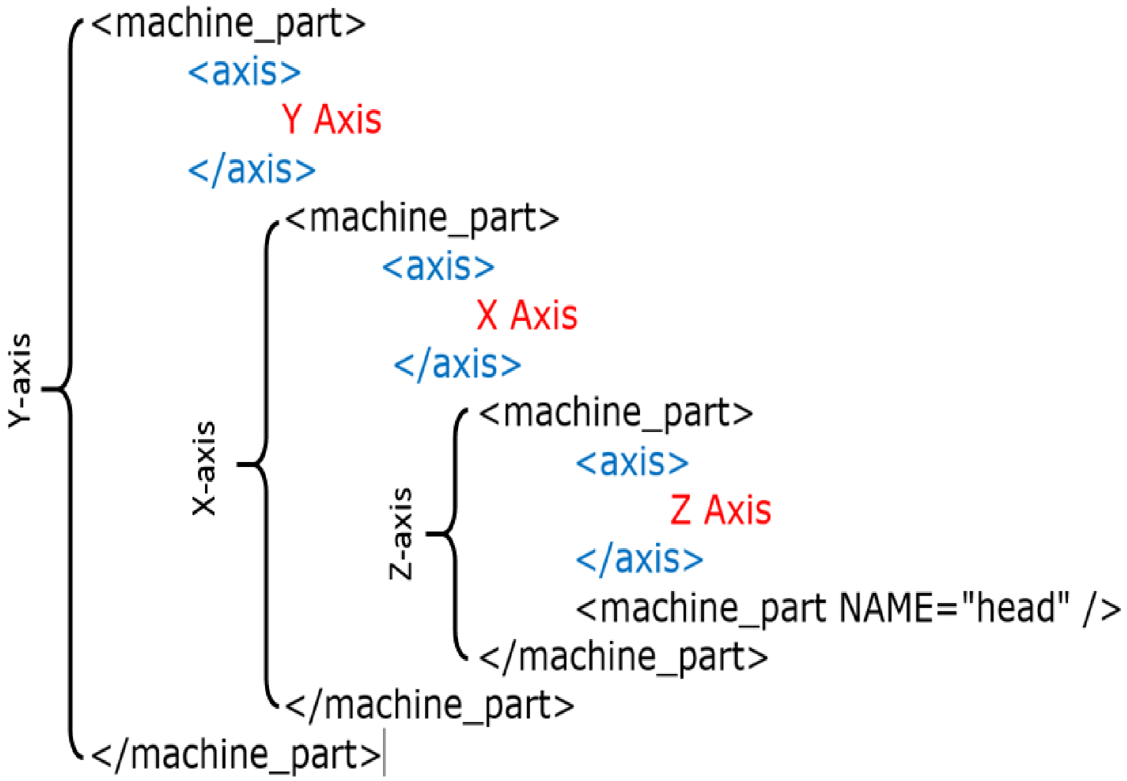


The example above (`Z-axis_head` on the previous page) shows a list of three models that are used to making up the Z-axis of the machine tool: `z-axis_head`, `cooler`, `spindle`. All of the remaining components are axial.

You must associate axis components and controls to the primary `machine_part` *object.*

Following is an example of the machine tool **Hermle_C12_SRT320**. The Y-axis is the primary (or parent) component for the head assembly. Therefore, before you close the Y-axis `machine_part`, you must open a new `machine_part` for each new axis component and continue until all head components making up the head of the machine are defined, excluding specific details for each axis.

The layout for the Hermle_C12_SRT320*.mtd is as follows:



Full text of the components of “head” is shown below:

```
<!-- ===== The head ===== -->
<machine_part>
     <axis>
   <control_info ADDRESS="Y" VALUE="220" HOME="0" MIN="-220" MAX="220"
/>
          <simple_linear I="0" J="1" K="0" />
     </axis>
     <model_list>
          <dmt_file>
               <path FILE="Hermle_C12_SRT320/y-axis_head.dmt" />
               <rgb R="185" G="185" B="185"/>
          </dmt_file>
     </model_list>
          <machine_part>
               <axis>
                    <control_info ADDRESS="X" VALUE="0" MIN="-175"
MAX="175"/>
                    <simple_linear I="1" J="0" K="0" />
               </axis>
                    <model_list>
                         <dmt_file>
                              <path FILE="Hermle_C12_SRT320/x-
axis_head.dmt" />
```

```
                              <rgb R="100" G="100" B="100"/>
                         </dmt_file>
                    </model_list>
                         <machine_part>
                              <axis>
                    <control_info ADDRESS="Z" VALUE="430" MIN="100"
MAX="430" />
                                   <simple_linear I="0" J="0" K="1" />
                              </axis>
                                   <model_list>
                                        <dmt_file>
                                       <path FILE="Hermle_C12_SRT320/z-
axis_head.dmt" />
                                             <rgb R="215" G="215"
B="215"/>
                                        </dmt_file>
                                        <dmt_file>
                                             <path
FILE="Hermle_C12_SRT320/cooler.dmt" />
                                             <rgb R="200" G="200"
B="200"/>
                                        </dmt_file>
                                        <dmt_file>
                                             <path
FILE="Hermle_C12_SRT320/spindle.dmt" />
                                             <rgb R="120" G="120"
B="120"/>
                                        </dmt_file>
                                   </model_list>
                              <machine_part NAME="head" />
                         </machine_part>
          </machine_part>
</machine_part>
```

At the end of the block `"The head"` you can use: `<machine_part NAME="head" />` This line defines the parts of the machine (all parts previously described: `y-axis_head.dmt; x-axis_head.dmt; z-axis_head.dmt; cooler.dmt; spindle.dmt)` as "`The head`" and connects them with the solver of PowerMILL and Delcam Post Processor.

Direction of movement in “The Head” block

The positive movement of head-axes of machine tool must match with the positive directions of the axes of the global coordinate system.

According to this, the direction of vectors of movement (`I`, `J` and `K`) must have positive values. See below:

```
<axis>
    <control_info ADDRESS="Y" VALUE="220" HOME="0" MIN="-220" MAX="220" />
    <simple_linear I="0" J="1" K="0" />
</axis>
```

```
<axis>
    <control_info ADDRESS="X" VALUE="0" MIN="-175" MAX="175"/>
    <simple_linear I="1" J="0" K="0" />
</axis>
```

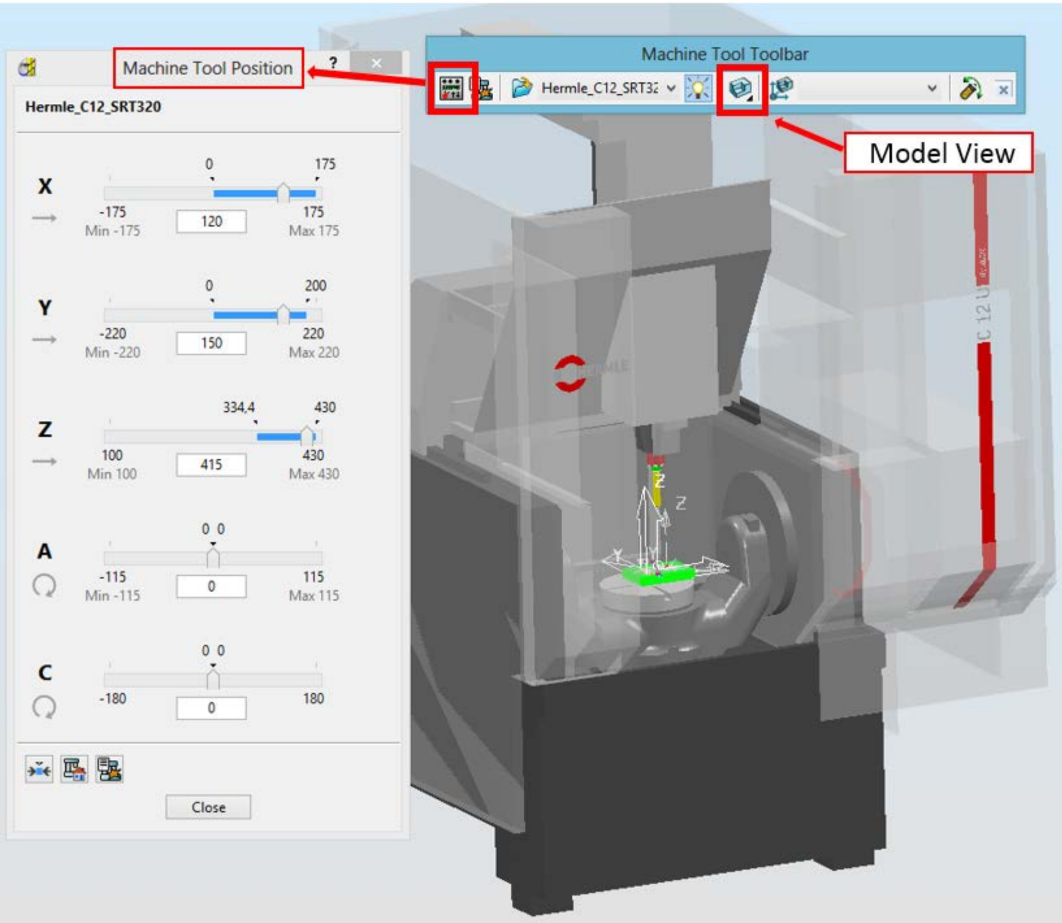
```
<axis>
    <control_info ADDRESS="Z" VALUE="430" MIN="100" MAX="430" />
    <simple_linear I="0" J="0" K="1" />
</axis>
```

Check the direction of movement.

1 Open the machine tool (**Hermle**) in PowerMILL.

2 Select **View> Toolbar> Machine Tool**.

3 In the **Machine Tool** toolbar, select **Model View**.



Use the **Machine Tool Position** dialog to check the direction of movement is correct.

Description of the movable elements. The "table"

Consider the movable elements of the table using an example **Hermle_C12_SRT320**. The rules of description are the same as for the head of the machine. In our **Hermle** machine tool example there are two axes components making up the table of the machine: `a-axis_table` and `c-axis_table`.

Below is a screenshot from the *.mtd file:

```
<!-- ===== The table ===== -->
    <machine_part>
        <axis>
            <control_info ADDRESS="A" MIN="-115" MAX="115" />
            <simple_rotary X="0" Y="0" Z="80" I="-1" J="0" K="0" />
        </axis>
        <model_list>
            <dmt_file>
                <path FILE="Hermle_C12_SRT320/a-axis_table.dmt" />
                <rgb R="140" G="140" B="140"/>
            </dmt_file>
        </model_list>
        <machine_part>
            <axis>
                <control_info ADDRESS="C" />
                <simple_rotary X="0" Y="0" Z="0" I="0" J="0" K="-1" />
            </axis>
            <model_list>
                <dmt_file>
                    <path FILE="Hermle_C12_SRT320/c-axis_table.dmt" />
                    <rgb R="200" G="200" B="200"/>
                </dmt_file>
            </model_list>
            <machine_part NAME="table" />
        </machine_part>
    </machine_part>
</machine>
```

The `a-axis_table` is the primary (or parent) component for the table assembly. Therefore, before you close the a-axis `machine_part`, open a new `machine_part` for the `c-axis_table` component.

At the end of the block, "The table" you must use `<machine_part NAME="table"/>` this line defines the parts of the machine as "The Table" (all parts previously described: `a-axis_table.dmt` and `c-axis_table.dmt`) and connects them with the solver of PowerMILL and Delcam Post Processor.

This example uses the same definitions that were used with the machine head components. However, one difference highlighted is when you use a negative vector for the linear and rotary movement. In our case, `a-axis_head` and `c-axis_head` has negative direction.

As the table controls the a-axis, in order to simulate a positive a-axis movement from a toolpath, the table needs to move in the opposite direction. When you view the movement in the machine tool position dialog, all movements appear normal. However, the table moves in the opposite direction to the Workplane.

```
<axis>
    <control_info ADDRESS="A" MIN="-115" MAX="115" />
    <simple_rotary X="0" Y="0" Z="80" I="-1" J="0" K="0" />
</axis>
```

```
<axis>
    <control_info ADDRESS="C" />
    <simple_rotary X="0" Y="0" Z="0" I="0" J="0" K="-1" />
</axis>
```

All axes (linear and rotary) included in "The Table" should move in opposite direction therefore need use minus for I, J or K vectors.

Full text of the `"Table"` components are shown below:

```
<!-- ===== The table ===== -->
<machine_part>
     <axis>
          <control_info ADDRESS="A" MIN="-115" MAX="115" />
          <simple_rotary X="0" Y="0" Z="80" I="-1" J="0" K="0" />
     </axis>
          <model_list>
               <dmt_file>
                    <path FILE="Hermle_C12_SRT320/a-axis_table.dmt" />
                    <rgb R="140" G="140" B="140"/>
               </dmt_file>
          </model_list>
               <machine_part>
                    <axis>
                         <control_info ADDRESS="C" />
                         <simple_rotary X="0" Y="0" Z="0" I="0" J="0"
K="-1" />
                    </axis>
                         <model_list>
                              <dmt_file>
                                   <path FILE="Hermle_C12_SRT320/c-
axis_table.dmt" />
                                   <rgb R="200" G="200" B="200"/>
                              </dmt_file>
                         </model_list>
                    <machine_part NAME="table" />
               </machine_part>
</machine_part>
```

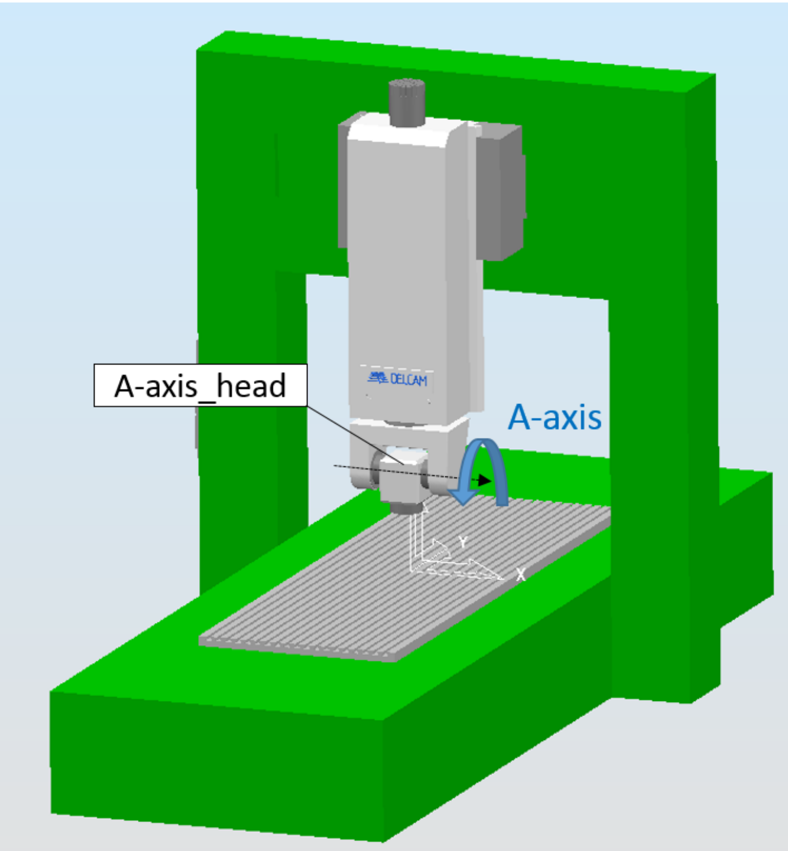
How to get XYZ values in PowerSHAPE

When you create an *.mtd file you need to know the exact location of the head attach point, rotation centre and axis values. We can do this in PowerSHAPE by creating some simple geometry at any points of interest.

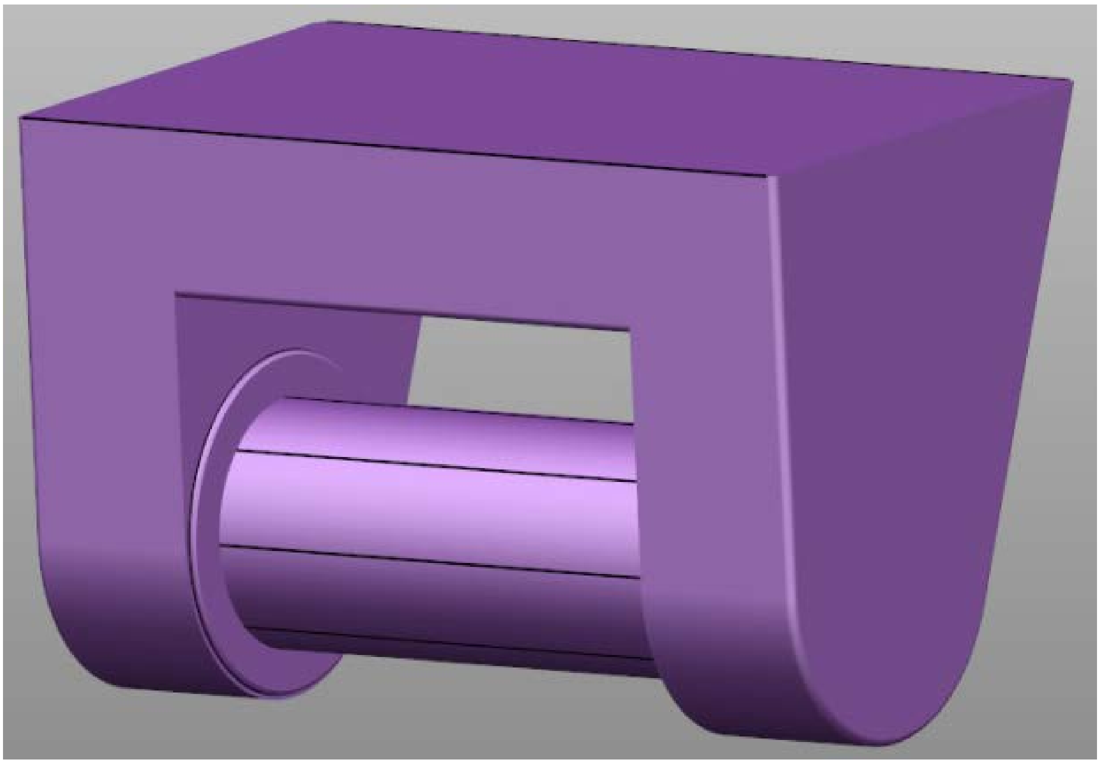
Consider a few examples:

- **Head-Head** machine tool with collinear vectors of rotation (the axes of rotation are parallel to the axes of the global coordinate system).

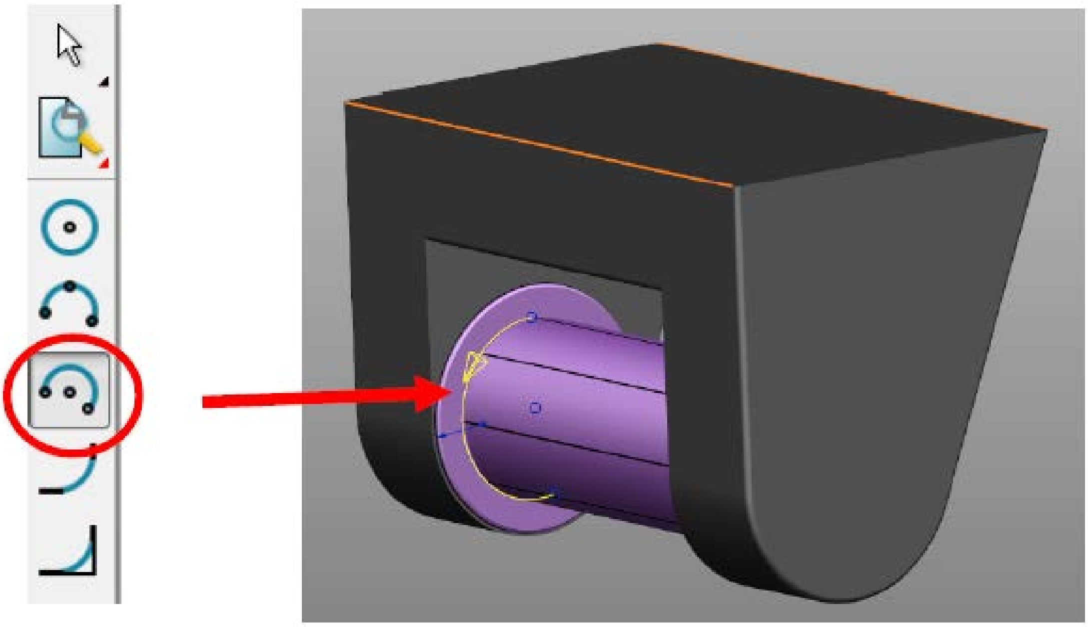
  For example, you need the centre of rotation for the A axis of the Rye machine tool. Below is an image of the machine tool where the A axis rotates around the X axis.



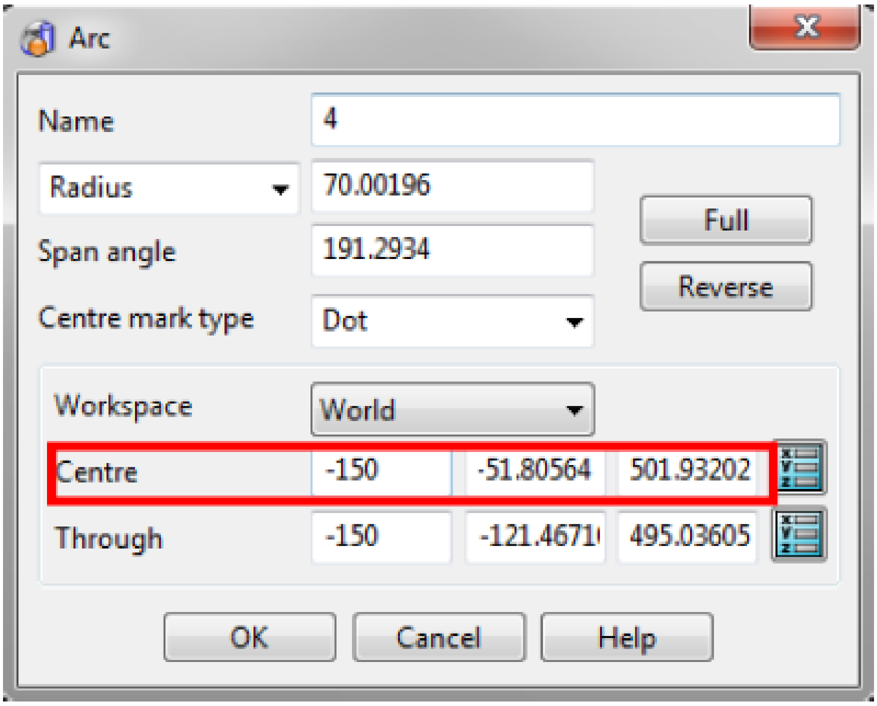
The arrow shows the part of the machine where the A axis component is attached. The A axis component rotates around a centre point of this part so in PowerSHAPE we isolate this part.



We then create some simple geometry (a line or an arc for example) where we can easily note the exact location for the centre of rotation:



Editing the arc gives the centre point coordinates:



*The global coordinate system (GCS) must be activated and be located in the centre of the table (X=0 Y=0 Z=0). We can either change the X value to zero in order to move the arc to the centre of the cylinder or just ignore it and keep it zero in the *.mtd*

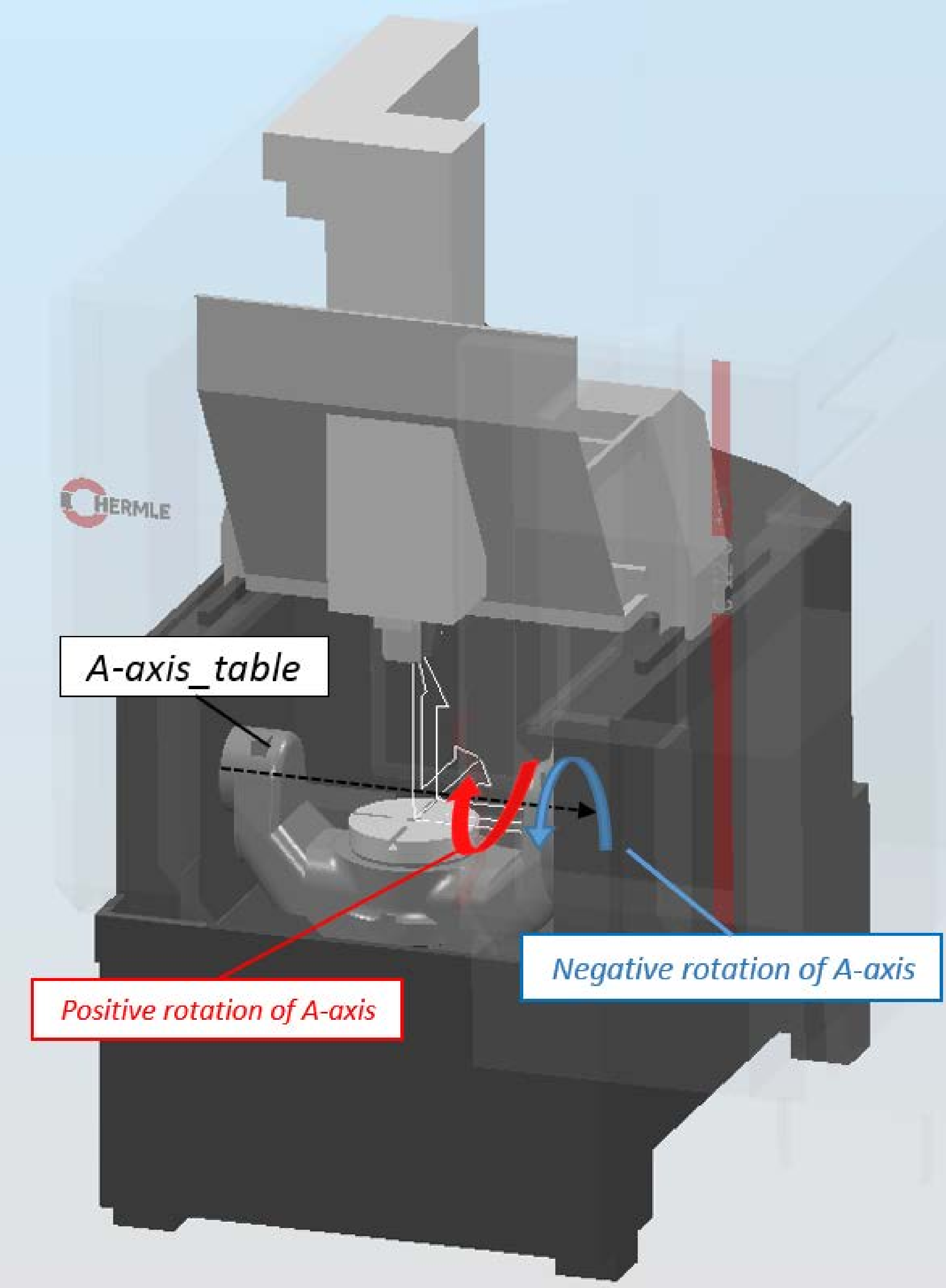
This results in:

```
<axis>
  <control_info ADDRESS="A" MIN="-110" MAX="110" />
  <simple_rotary X="0" Y="-51.80564" Z="501.93202" I="1" J="0" K="0" />
</axis>
<model_list>
  <dmt_file>
    <path FILE="rye/barrel_1.dmt" />
    <rgb R="245" G="245" B="245" />
  </dmt_file>
```

The rotation of A-axis_head occurs around the X axis therefore direction of vectors are: I="1" J="0" K="0".

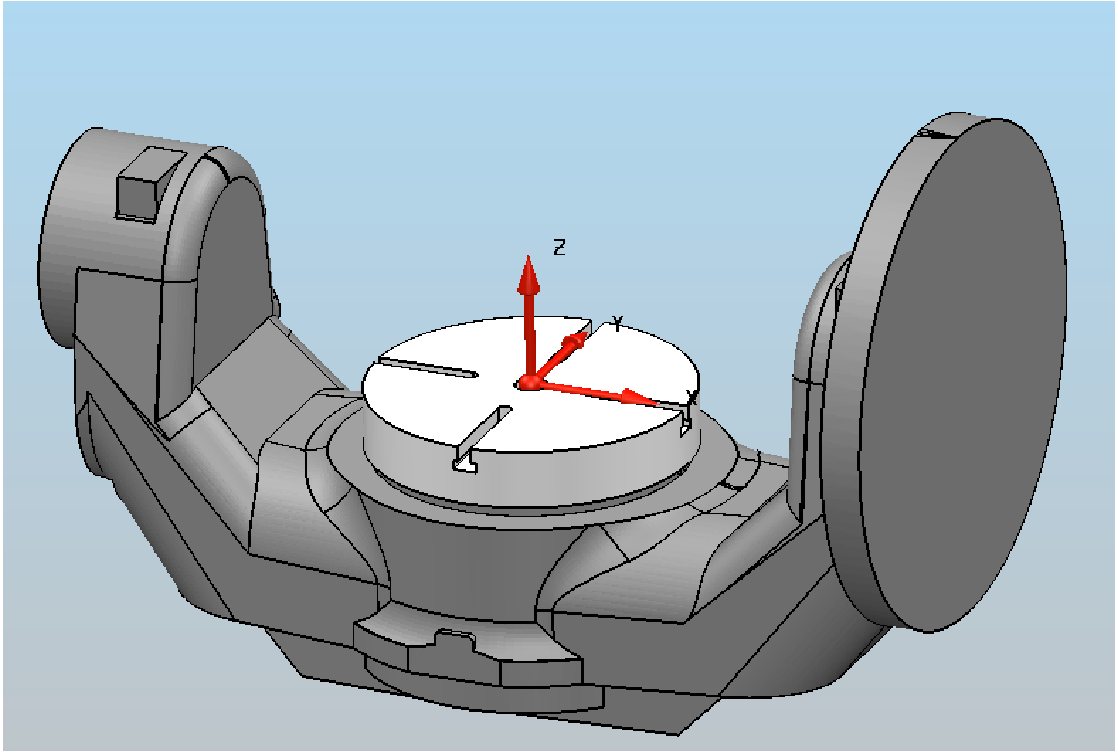
Use positive value of vectors for movement's parts of head. Use negative value of vectors for movement's parts of table.

Table-Table machine tool (**Hermle_C12_SRT320**) with collinear vectors of rotation (the axes of rotation are parallel to the axes of the global coordinate system).

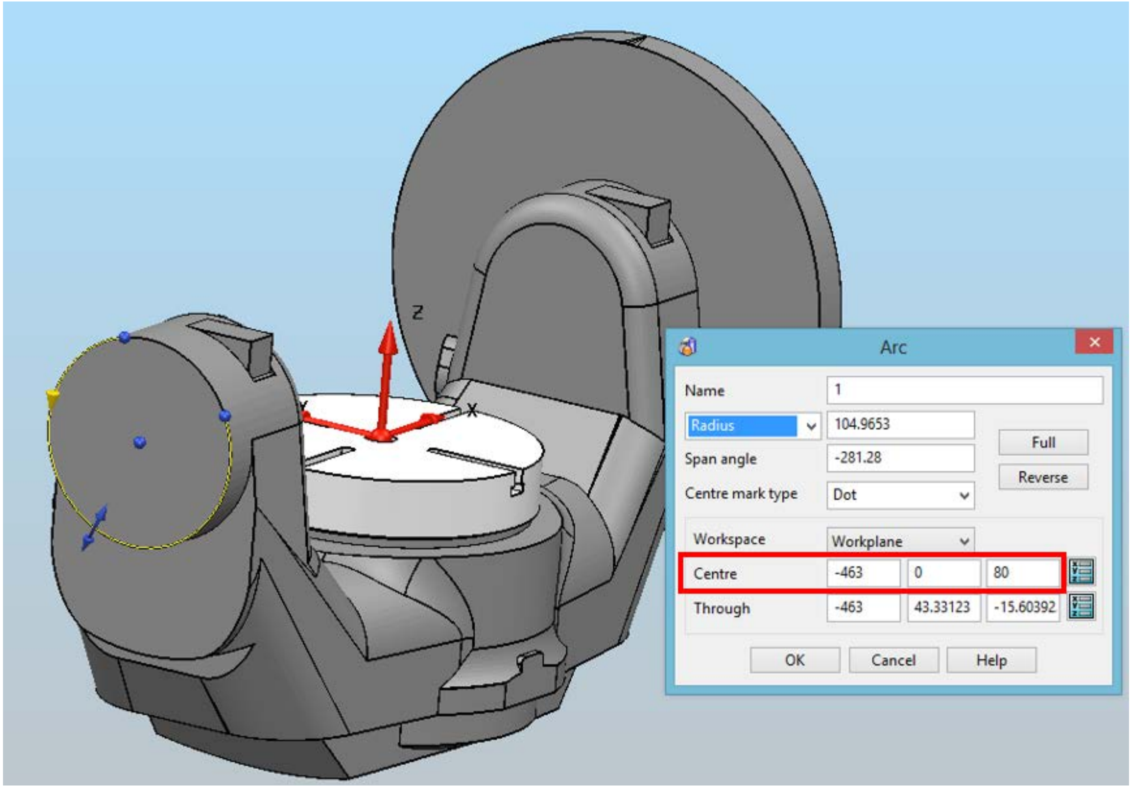

HERMLE
A-axis_table
Negative rotation of A-axis
Positive rotation of A-axis

Let us open A-axis_table and C-axis_table in PowerSHAPE. Then you can create new Local Coordinate System (LCS number 1) at X=0 Y=0 Z=0. This coordinate system duplicates the Global Coordinate System (GCS). See below.

In PowerSHAPE is very important to have the correct position of the activated coordinate system.



Then you must create some simple geometry (a line or an arc, for example) where we can easily note the exact location for the centre of rotation:



*We can either change the X value to 0 in order to move the arc to the centre of the table or just ignore it and keep it zero in the *.mtd. As a result, we get:*

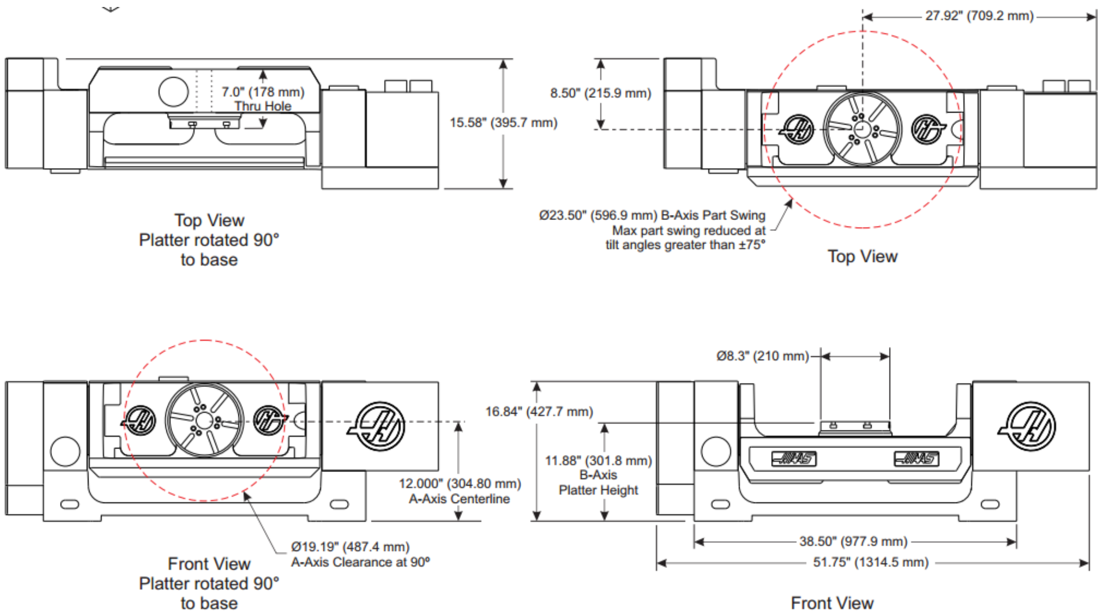
```
<axis>
    <control_info ADDRESS="A" MIN="-115" MAX="115" />
    <simple_rotary X="0" Y="0" Z="80" I="-1" J="0" K="0" />
</axis>
<model_list>
    <dmt_file>
        <path FILE="Hermle_C12_SRT320/a-axis_table.dmt" />
        <rgb R="140" G="140" B="140"/>
```

The centre of rotation of C-axis_table:

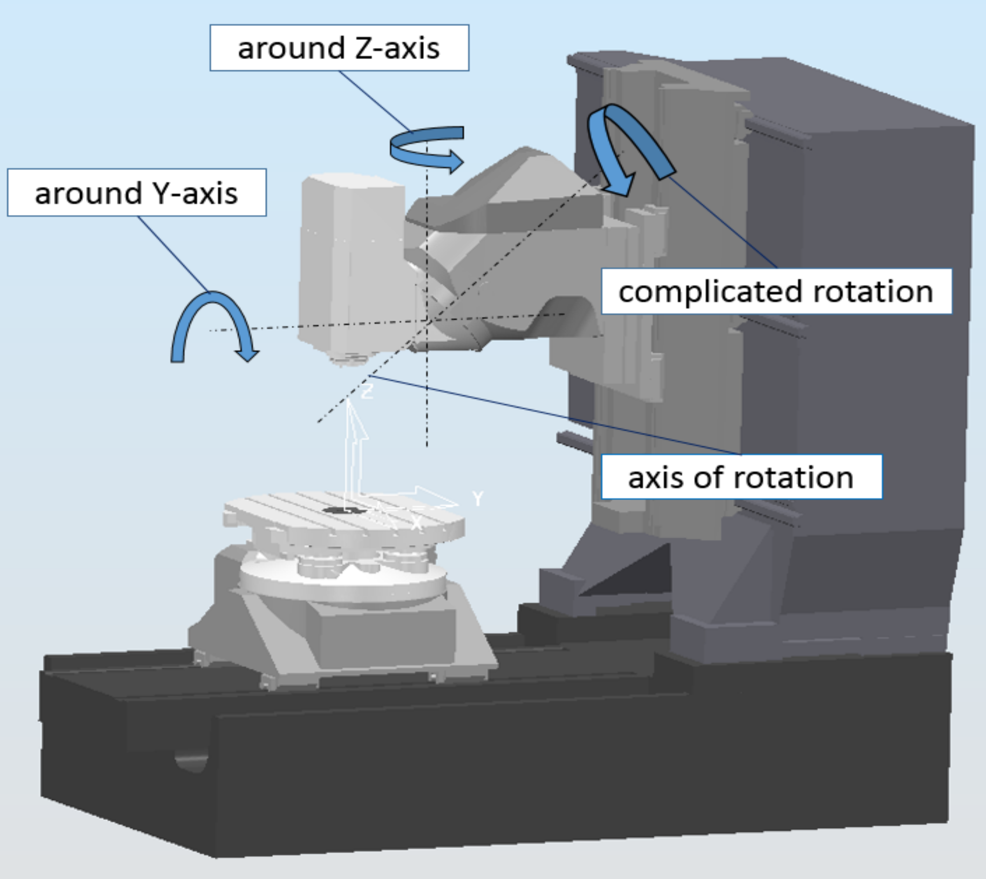
```
<axis>
    <control_info ADDRESS="C" />
    <simple_rotary X="0" Y="0" Z="0" I="0" J="0" K="-1" />
</axis>
<model_list>
    <dmt_file>
        <path FILE="Hermle_C12_SRT320/c-axis_table.dmt" />
        <rgb R="200" G="200" B="200"/>
```

In some cases difficult define correct position the part of machine (for example: the bad quality geometry machine, wrong machine assembly). Use specification of machine tool to find the right solution.

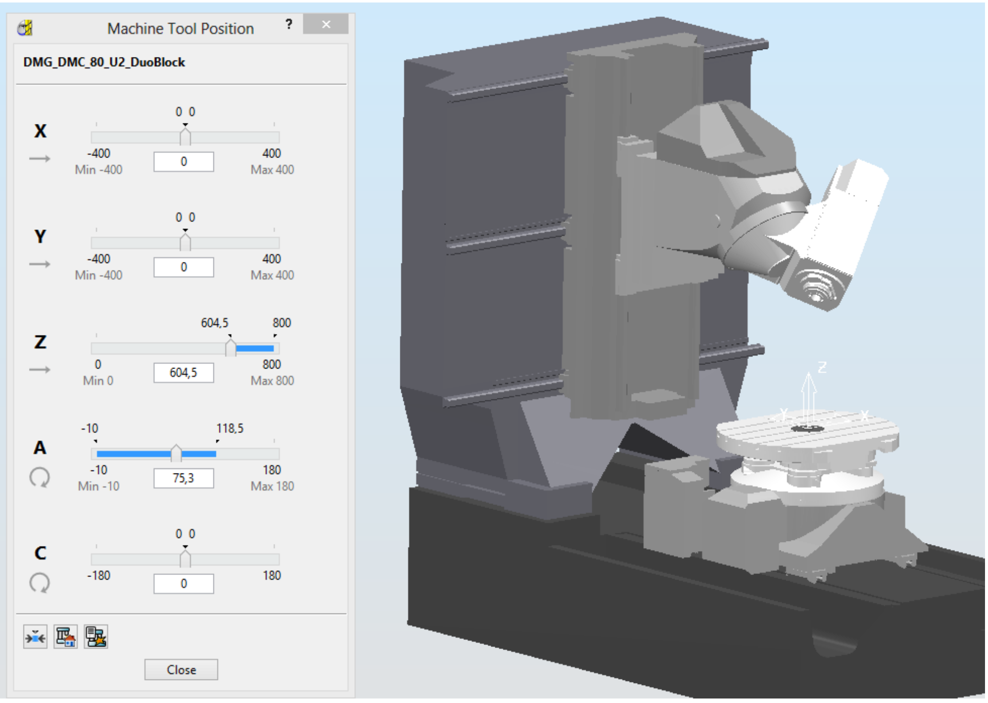
Example with **Haas rotation table TR 210**:



“Head-Table” machine tool ("DMG_DMC_80_U2_DuoBlock") with not collinear vectors of rotation (the axes of rotation are not parallel to the axes of the global coordinate system).

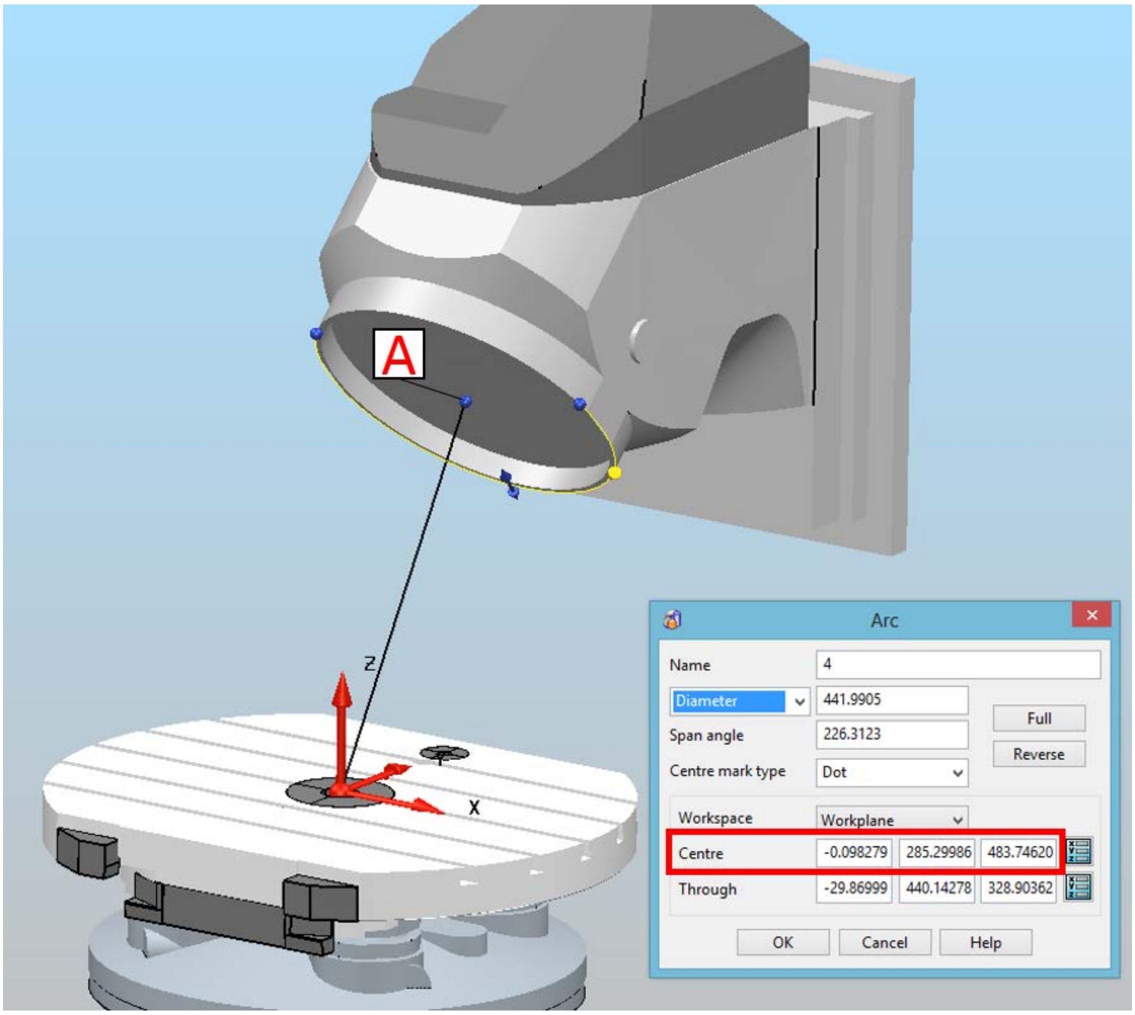


Such rotation is called the complicated, because consists of two rotational movements (around Z-axis and around Y-axis).



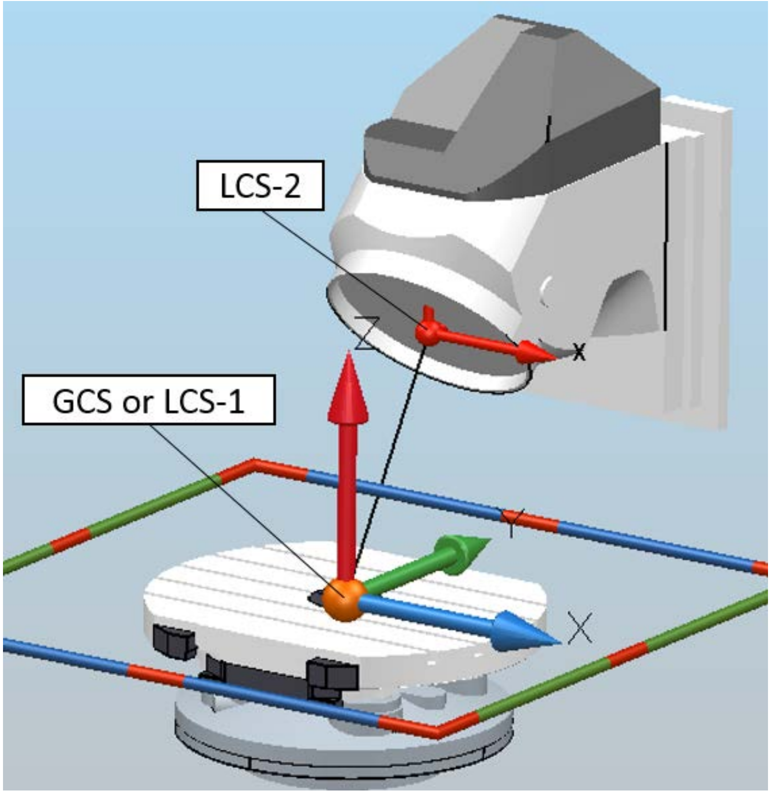
It is necessary to find the centre of rotation and the direction of motion. Consider the plane of rotation.

Let us create new local coordinate system (LCS-1) in the centre of table `X=0 Y=0 Z=0`. We then create some simple arc where we can note the exact location of the centre of rotation:

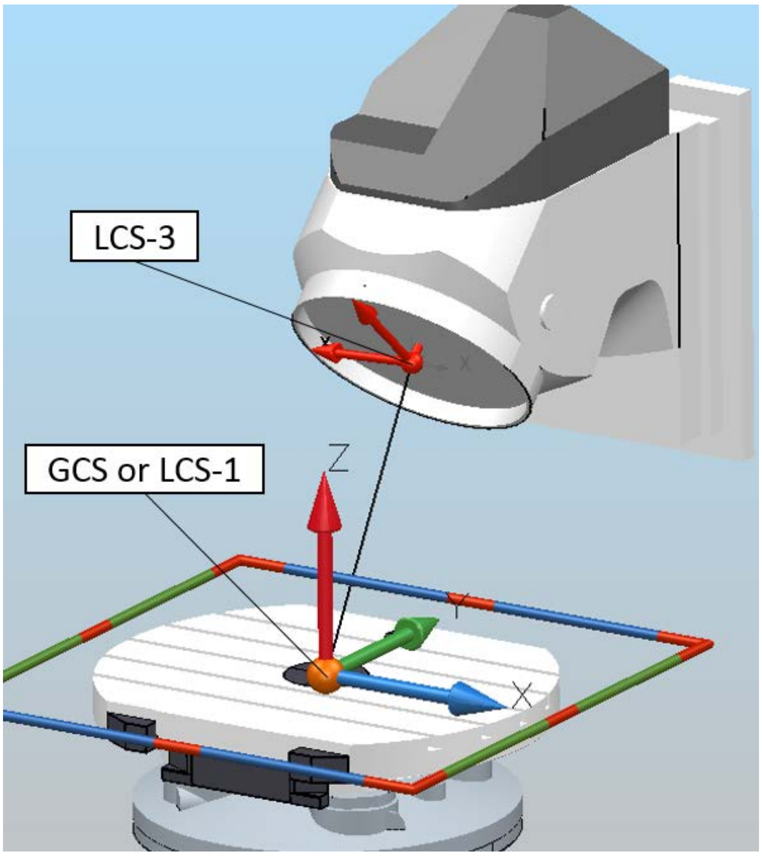


Double clicking on arc. The point **A** is centre of rotation with coordinates: X=-0.098279 Y=285.29986 Z=483.74620. This coordinates we use for define centre of rotation.

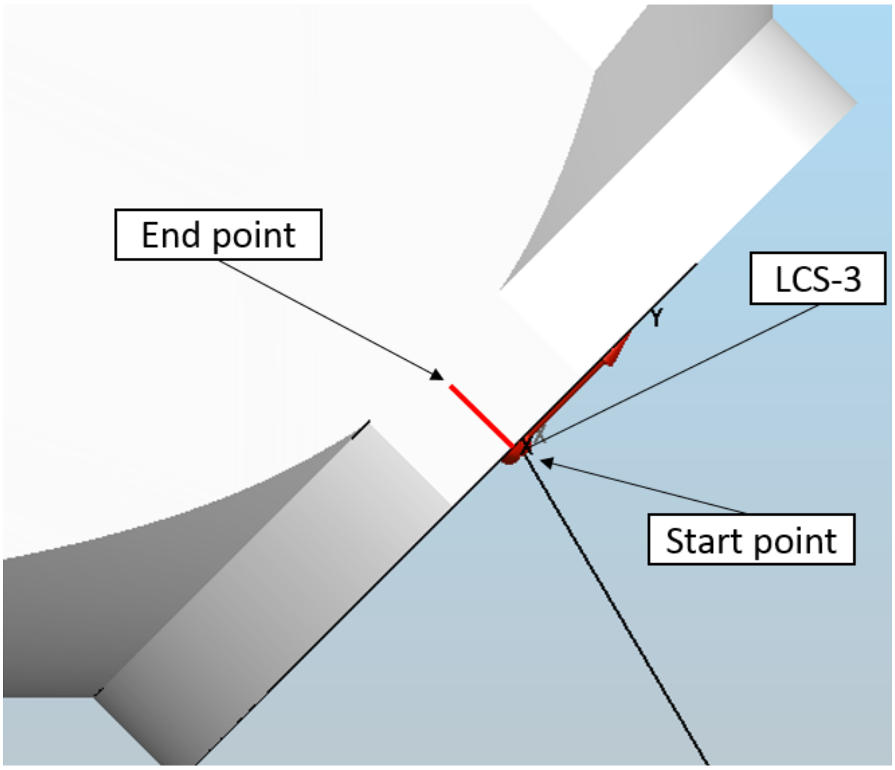
At the centre of rotation (point **A**) we create local coordinate system (LCS-2) oriented like the global coordinate system. See below:



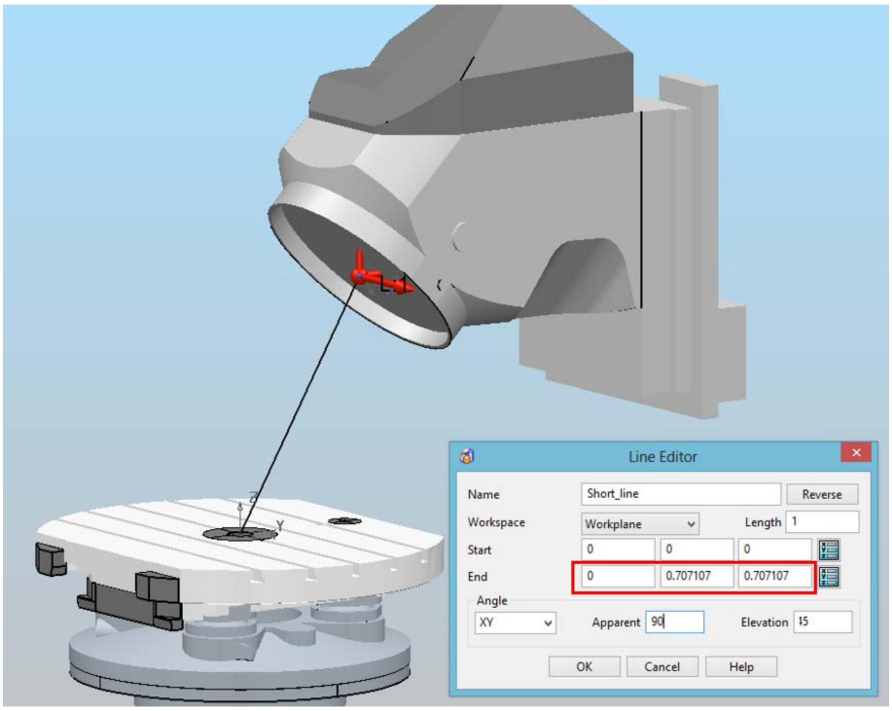
Then, additionally create local coordinate system (LCS-3) at the point **A**, oriented perpendicular created arc (Z-axis should be perpendicular created arc). See below:



By using activated LCS-3 create short line (start point x=0; y=0; z=0; and end point x=0; y=0; z=1)



Then we activate LCS-2 and double click on short line. See result below:



Where I=0; J=0.707107; K=0.707107. We find the direction of movement.

See screenshot below:

```
<machine_part>
  <axis>
    <control_info ADDRESS="A" VALUE="0" MIN="-10" MAX="180" />
    <simple_rotary X="-0.1" Y="285.3" Z="483.75" I="0" J="0.707107" K="0.707107" />
  </axis>
  <model_list>
    <dmt_file>
      <path FILE="DMG_DMC_80_U2_DuoBlock/B_Axis.dmt" />
```

If the elevation is 45 degrees then we can use `I=0; J=1; K=1`*.*

Finally, use PowerMILL and Delcam PostProcessor for testing MTD (check start and end point, correct position of all parts of machine, collision detection, direction of movements and etc.).