
Autodesk® PowerMill® 2017

Robot

How to build a Robot simulation



Autodesk® PowerMill® 2017

© 2016 Delcam Limited. All Rights Reserved. Except where otherwise permitted by Delcam Limited, this publication, or parts thereof, may not be reproduced in any form, by any method, for any purpose.

Certain materials included in this publication are reprinted with the permission of the copyright holder.

Trademarks

The following are registered trademarks or trademarks of Autodesk, Inc., and/or its subsidiaries and/or affiliates in the USA and other countries: 123D, 3ds Max, Alias, ArtCAM, ATC, AutoCAD LT, AutoCAD, Autodesk, the Autodesk logo, Autodesk 123D, Autodesk Homestyler, Autodesk Inventor, Autodesk MapGuide, Autodesk Streamline, AutoLISP, AutoSketch, AutoSnap, AutoTrack, Backburner, Backdraft, Beast, BIM 360, Burn, Buzzsaw, CADmep, CAiCE, CAMduct, Civil 3D, Combustion, Communication Specification, Configurator 360, Constructware, Content Explorer, Creative Bridge, Dancing Baby (image), DesignCenter, DesignKids, DesignStudio, Discreet, DWF, DWG, DWG (design/logo), DWG Extreme, DWG TrueConvert, DWG TrueView, DWGX, DXF, Ecotect, Ember, ESTmep, FABmep, Face Robot, FBX, FeatureCAM, Fempro, Fire, Flame, Flare, Flint, ForceEffect, FormIt 360, Freewheel, Fusion 360, Glue, Green Building Studio, Heidi, Homestyler, HumanIK, i-drop, ImageModeler, Incinerator, Inferno, InfraWorks, Instructables, Instructables (stylized robot design/logo), Inventor, Inventor HSM, Inventor LT, Lustre, Maya, Maya LT, MIMI, Mockup 360, Moldflow Plastics Advisers, Moldflow Plastics Insight, Moldflow, Moondust, MotionBuilder, Movimento, MPA (design/logo), MPA, MPI (design/logo), MPX (design/logo), MPX, Mudbox, Navisworks, ObjectARX, ObjectDBX, Opticore, P9, PartMaker, Pier 9, Pixlr, Pixlr-o-matic, PowerInspect, PowerMill, PowerShape, Productstream, Publisher 360, RasterDWG, RealDWG, ReCap, ReCap 360, Remote, Revit LT, Revit, RiverCAD, Robot, Scaleform, Showcase, Showcase 360, SketchBook, Smoke, Socialcam, Softimage, Spark & Design, Spark Logo, Sparks, SteeringWheels, Stitcher, Stone, StormNET, TinkerBox, Tinkercad, Tinkerplay, ToolClip, Topobase, Toxik, TrustedDWG, T-Splines, ViewCube, Visual LISP, Visual, VRED, Wire, Wiretap, WiretapCentral, XSI

All other brand names, product names or trademarks belong to their respective holders.

Disclaimer

THIS PUBLICATION AND THE INFORMATION CONTAINED HEREIN IS MADE AVAILABLE BY AUTODESK, INC. "AS IS." AUTODESK, INC. DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE REGARDING THESE MATERIALS.

How to build a Robot simulation

This document explains how to build/modify an existing *.mtd file of a basic 6 axis robot simulation for PowerMill Robot based on the data available from the Robot manufacturer web site (Robot CAD data and data sheet).

This document outlines standard procedures and naming conventions that will simplify the process of creating the .mtd file. It is recommended that you follow these accurately.

The aim is to build a simulation which represents the robot cell. The position of each axis is important. So, before creating a simulation in PowerMill, you must create the robot cell (robot + spindle + external entities) in PowerShape in the right position. All must match the physical cell. If in doubt, remeasure it using the robot.

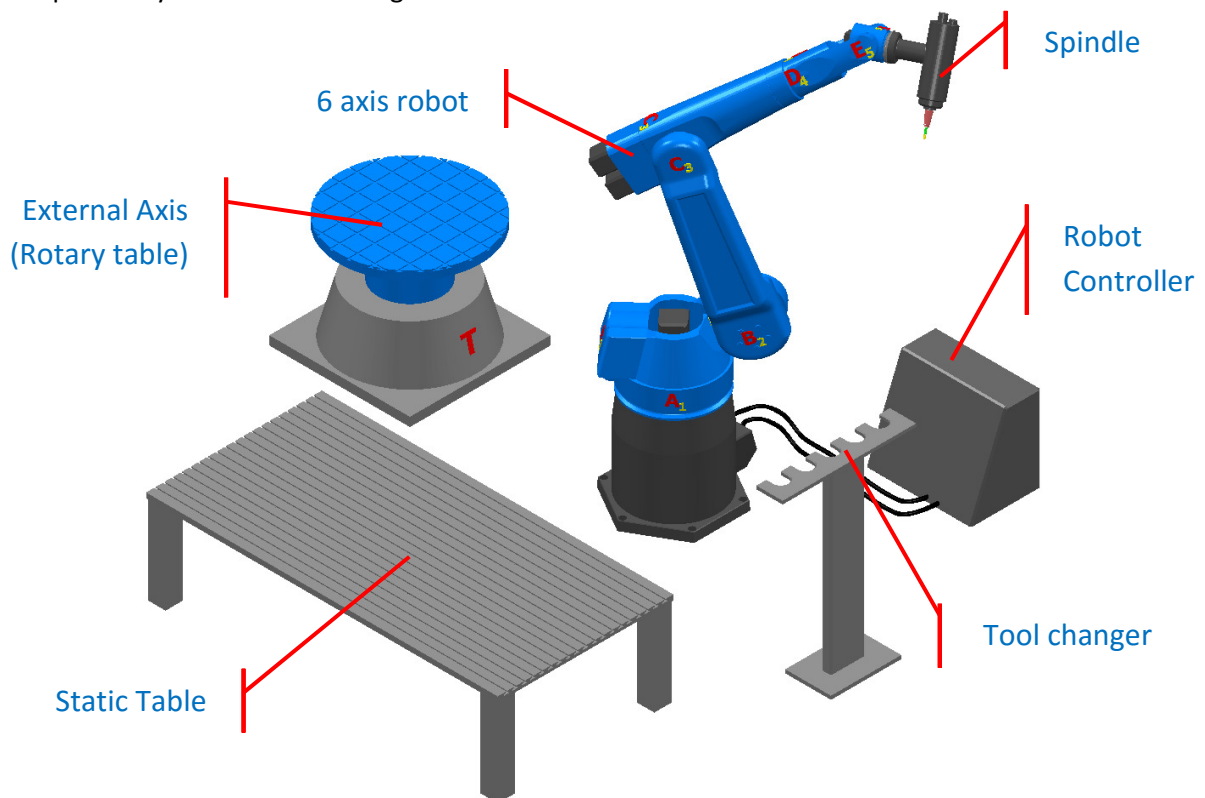
1. Requirements

1.1 Geometries (CAD models)

It's easier to work with surfaces or solids to build the robot simulation instead of triangles files.

Often, the customer or the robot manufacturer can provide data in this format.

An STL model should be used only if it is impossible to get a surface/solid model. This can be exported by the customer using the robot manufacturer simulation software.



1.1.1 Robot CAD model and data sheet

1.1.1.1 Robot CAD model

CAD data of the robot can usually be found in a surface or solid format. Do not use triangle file format (*.stl) when solid/surface are available.

The majority of the robot manufacturers allow downloading the CAD of the robot from their web site.

Where possible, download a STEP file, as often, each axis is a separated solid so there aren't a lot of surfaces to sort out.

Example where robot CAD data can be downloaded from (non-exhaustive list):

KUKA: <http://www.kuka-robotics.com/en/downloads/>

ABB: <http://new.abb.com/products/robotics/industrial-robots>

MOTOMAN: <http://www.motoman.eu/>

STAUBLI: <http://www.staubli.com/en/robotics/customer-support/cad-library/>

FANUC: *Not available from web site.*

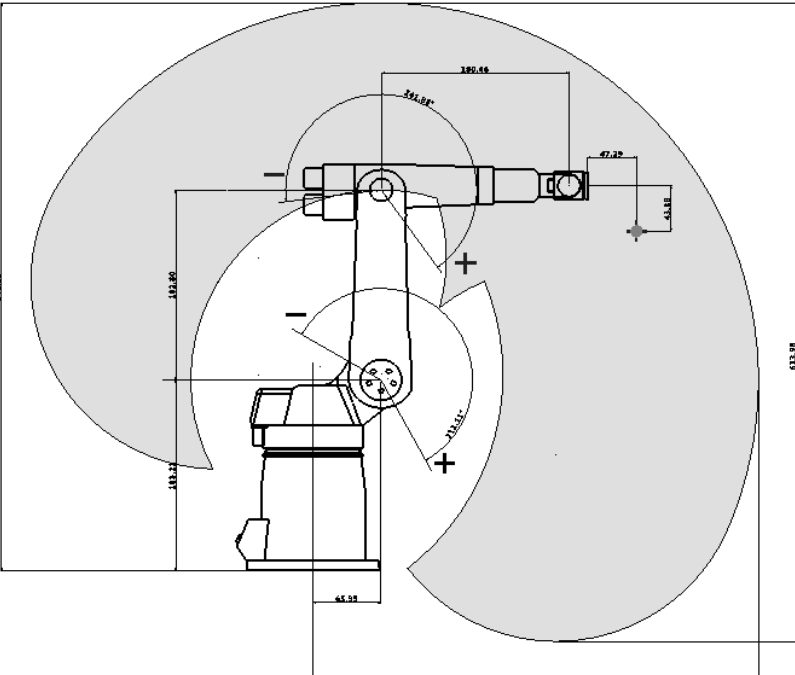
The website should also have the data sheet of the robot you are looking for, giving you more information about the axes limits, axes length, and other specifications.

1.1.1.2 Robot data sheet

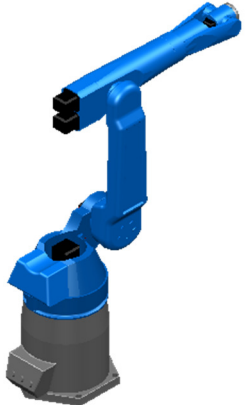

Some of the data required for the mtd file, such as the 3D model of the robot arm and the data sheet (containing basic information about the axis rotation directions and the axis limits), may be available from the manufacturer website.

Below is an example of a robot data sheet that can be found on a robot manufacturer web site.

AUTODESK R2 - Model 2017



Technical drawing of the Autodesk R2 robot arm showing dimensions and axis rotation directions. Dimensions include 189.48, 54.88°, 47.32°, 43.88, 113.95, 113.20, 113.20, 45.58, and 128.81. Axis rotation directions are indicated with '+' and '-' signs.

Robot Type	Autodesk R2 - Model 2017
Maximum load (center of gravity)	185 kg
Number of axes	6
Controller model	Autodesk controller 2017/R2
Position	Floor / Wall
Repeatability	± 0.04 mm
Robot weight	880 kg

Axes	Range	Speed
Axis 1 (A)	± 185°	112°/s
Axis 2 (B)	-50° / +100°	89°/s
Axis 3 (C)	- 215° / +70°	118°/s
Axis 4 (D)	± 360°	169°/s
Axis 5 (E)	± 130°	212°/s
Axis 6 (F)	± 720°	110°/s

1.1.2 CAD of the spindle (or laser head, welding head, trimming tool...)

This CAD is the object which is attached to the 6th axis of the robot. In our case, it's often a spindle, but it could also be a welding or cladding head, a trimming tool, etc.

An STL model should be used only if it is impossible to get a surface/solid model. This can be exported by the customer using the robot manufacturer simulation software.

1.1.3 CAD of the External axes

This can be rails or rotary tables, attached or not to the robot.

An STL model should be used only if it is impossible to get a surface/solid model. This can be exported by the customer using the robot manufacturer simulation software.

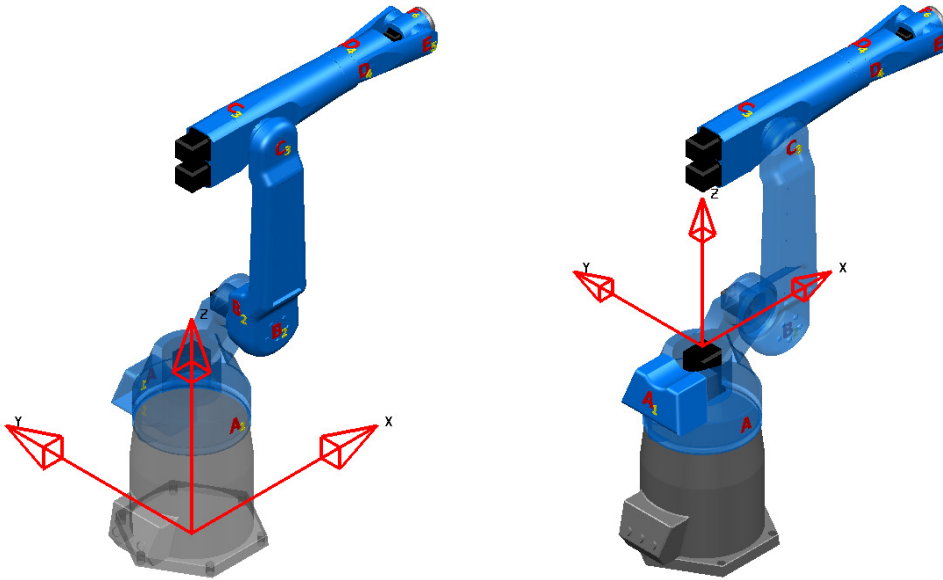
1.1.4 CAD of the robot Cell

This is everything not linked to the robot and modeling is optional. It contains static objects, such as the cell where the robot is located, the fixtures, and the robot controller. An STL model should be used only if it is impossible to get a surface/solid model. This can be exported using the robot manufacturer simulation software.

1.2 Coordinates, Workplane and Positions

1.2.1 Robot World workplane position & orientation

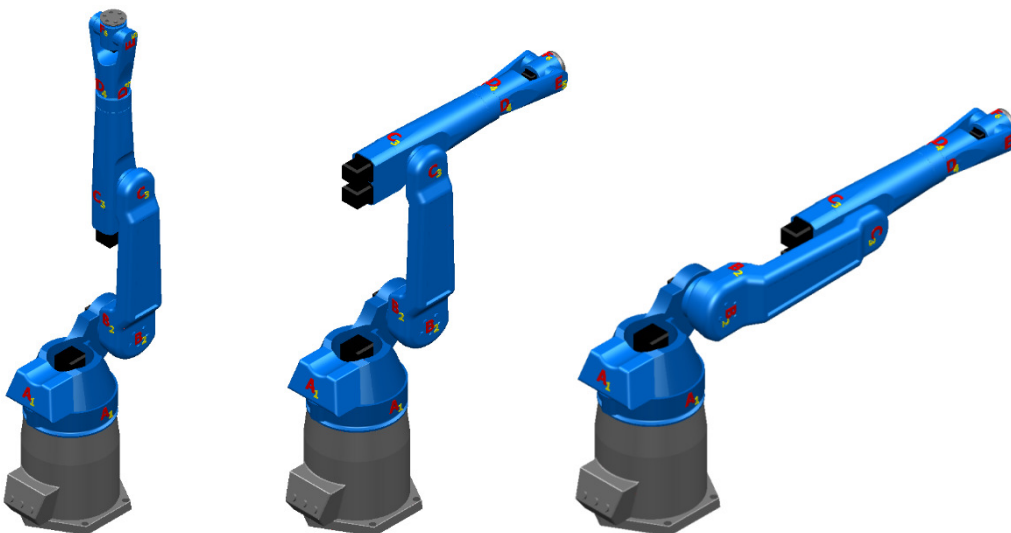
Check where the robot world workplane is.



1.2.2 Robot “Zero position”

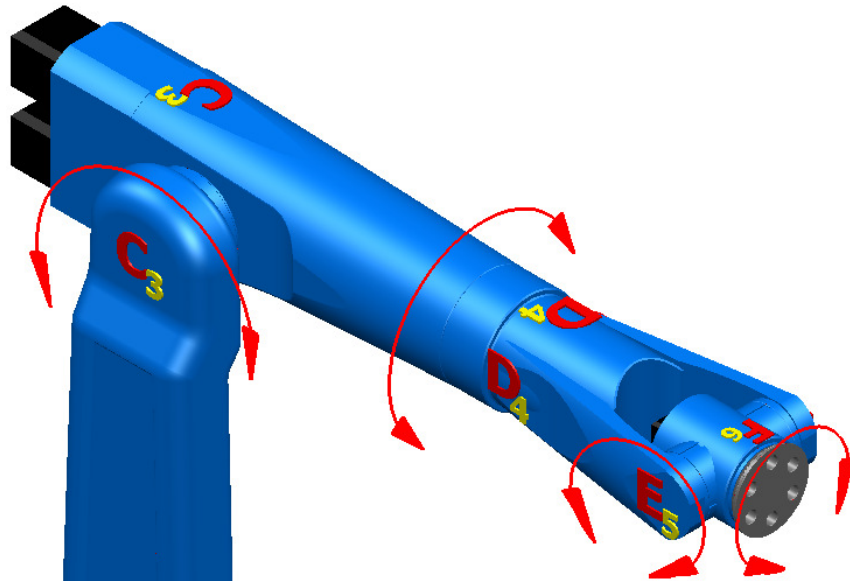
“Zero position” is the robot’s position when all joint angles and/or axes are moved to their 0 value. This is also valid for external axes (rails, rotary axes).

The robot zero position is different for each robot manufacturer. When you build the simulation, you will have to move the robot to this position.



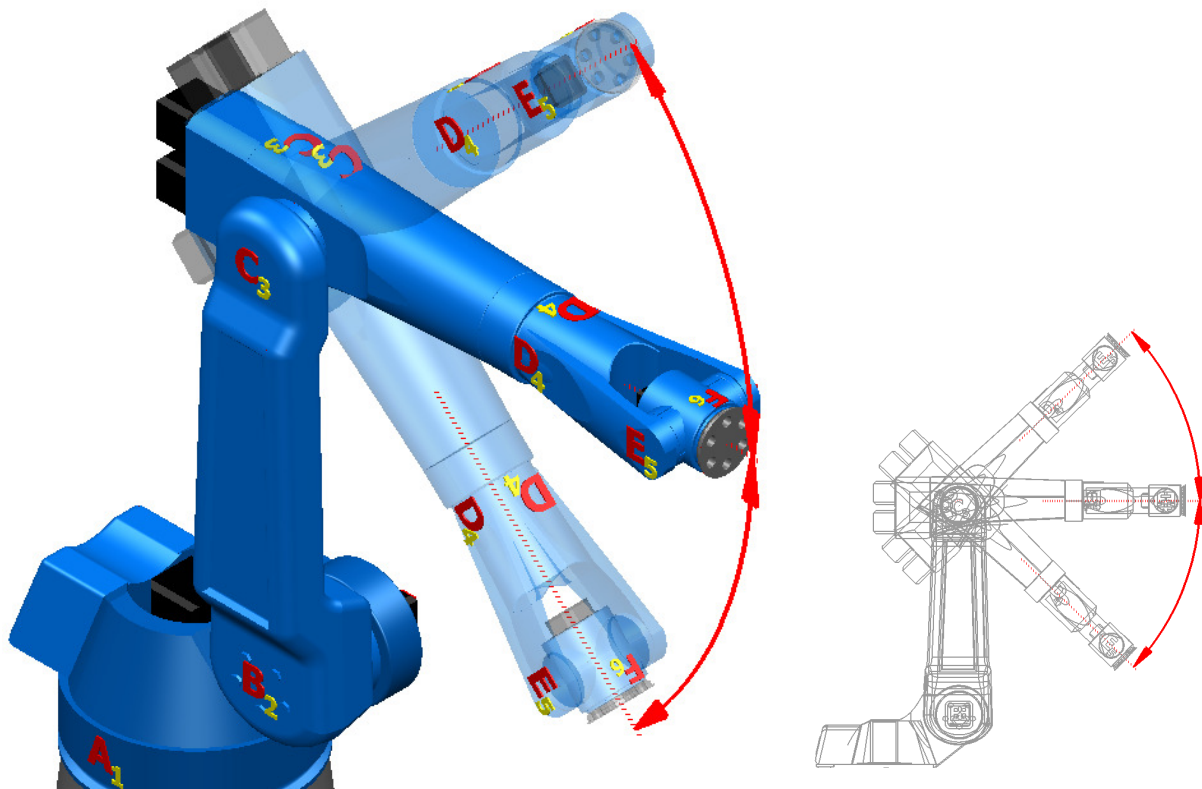
1.2.3 Axes directions

Check the direction of each axis of the robot (external axes too).



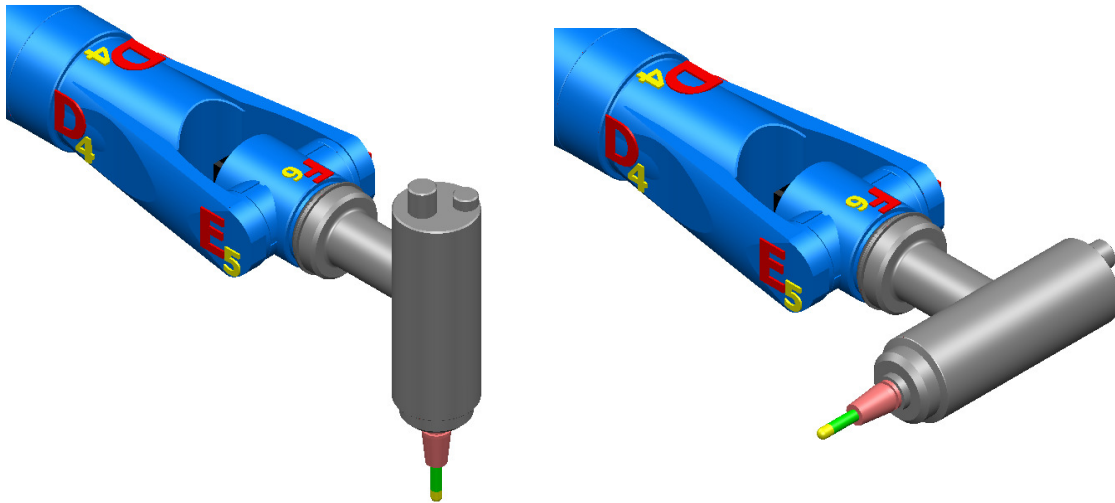
1.2.4 Axes min/max range

Verify the minimum and maximum value of each axis.



1.2.5 Spindle orientation

Check the position of the spindle when the 6th axis is set at its zero position.

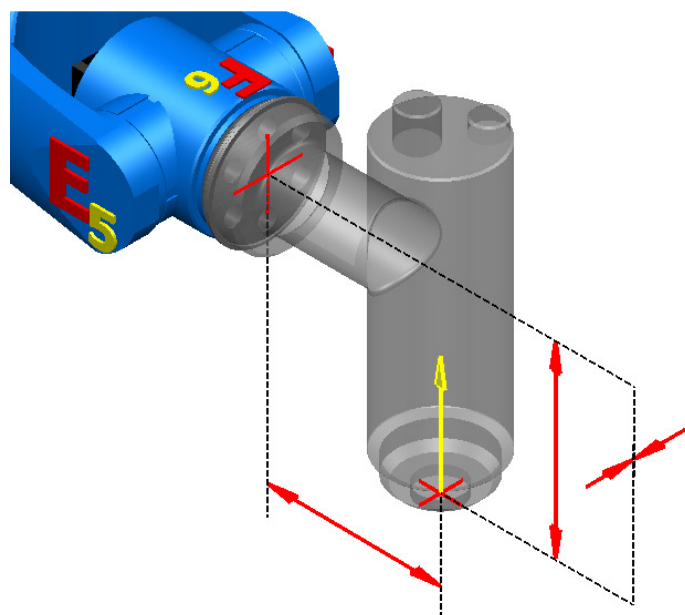


1.2.6 “Tool attach point” position

Because the CAD may be inaccurate, the best way to ascertain the tool attach point is to measure the “real” coordinates (X, Y, Z) of the tool attach point at the tip of the spindle (in red). One way to get these values is to use a tool and spindle calibration, described in the document:

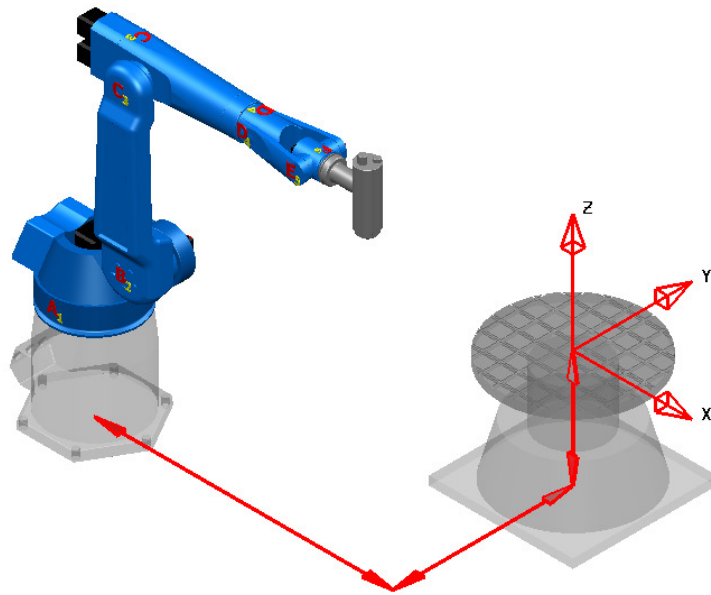
PowerMill Robot - Tool and Spindle Calibration.pptx

It is also important to know the orientation of the tool axis vector (in yellow); this is calculated by the tool and spindle calibration.



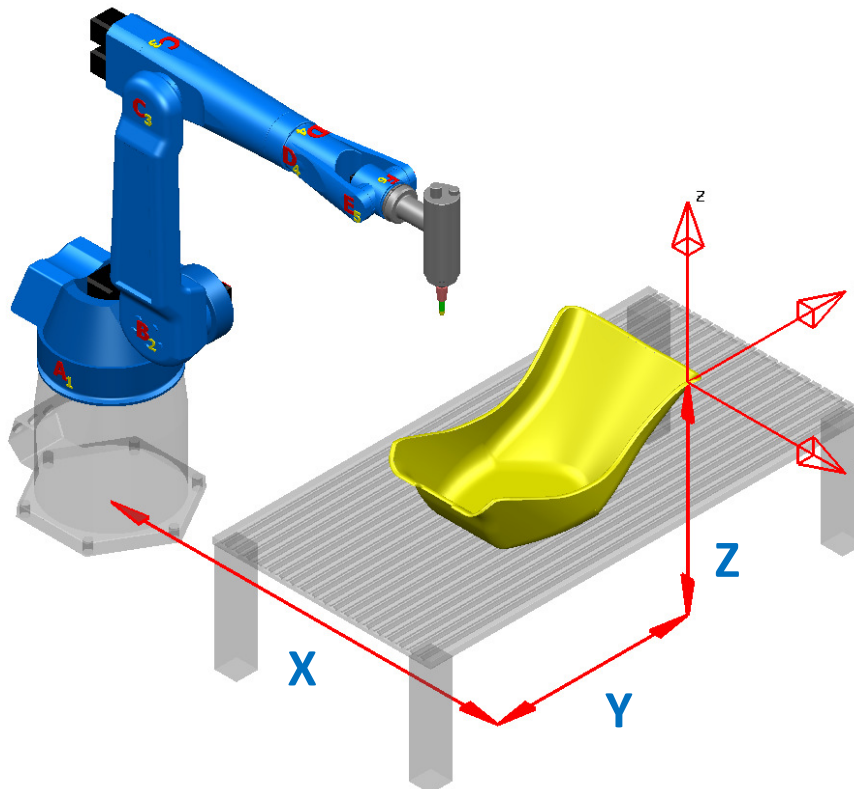
1.2.7 External axis position

Check the position of the table from the robot origin workplane (X, Y, Z coordinates).



1.2.8 Position of the part from the robot

You must accurately know the position of the part from the robot world workplane.



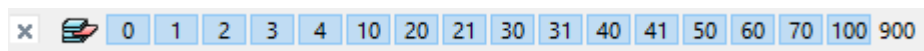
2. Preparation of the CAD Model using Autodesk PowerShape

2.1 Loading and organizing the robot CAD data

The first step is to load the files downloaded from the web site into PowerShape.

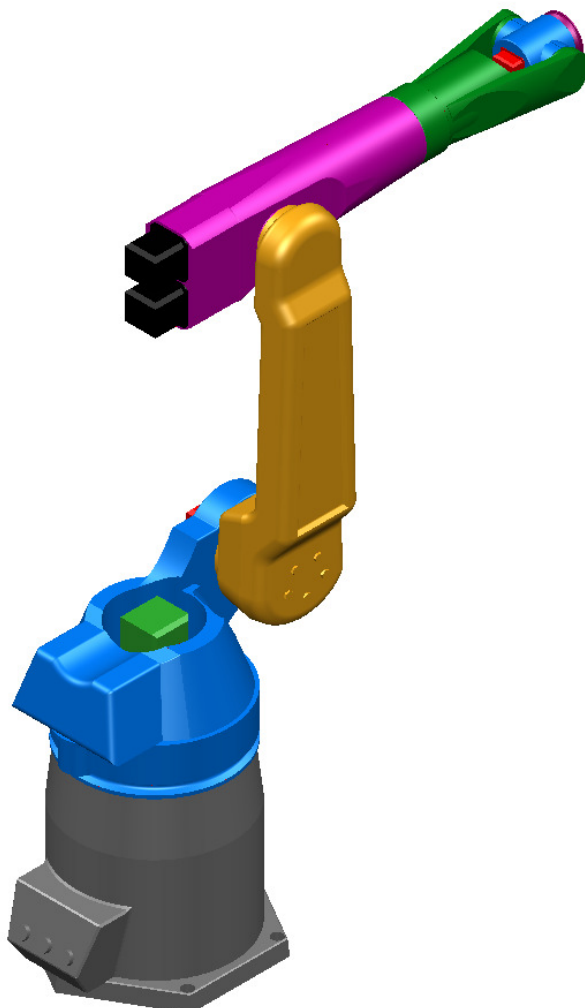
If you download a STEP file (if available from web site), often, each robot axis, labels and logo are into a separate solid. In that case, you do not need to organize the CAD model.

Some robot manufacturers only provide IGES files that contain only surfaces in a single layer. In that case you will need to spend some time in PowerShape, removing unnecessary surfaces/solids and sorting each axis surfaces/solids into separate PowerShape layers.



Note that only the external surfaces of the robot are needed for the simulation and too many surfaces can have a negative impact on simulation speed.

Make sure to use the naming convention as below (see also section 4)



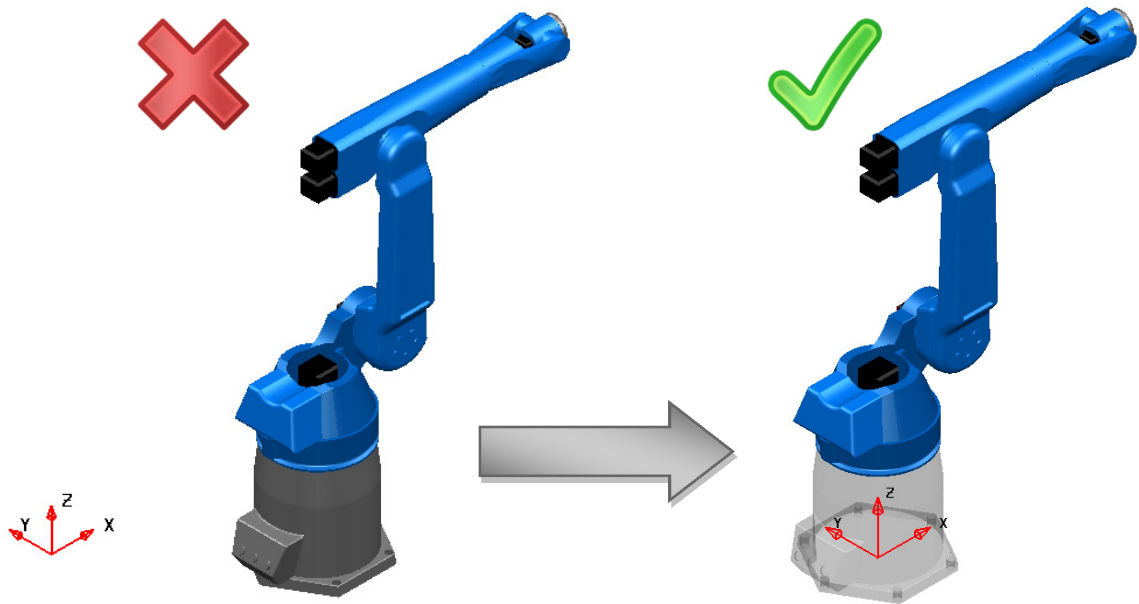
<input checked="" type="checkbox"/>	10	A
<input checked="" type="checkbox"/>	20	AB
<input checked="" type="checkbox"/>	21	AB Motor
<input checked="" type="checkbox"/>	30	BC
<input checked="" type="checkbox"/>	31	BC Logo
<input checked="" type="checkbox"/>	40	CD
<input checked="" type="checkbox"/>	41	CD Motor
<input checked="" type="checkbox"/>	50	DE
<input checked="" type="checkbox"/>	60	EF
<input checked="" type="checkbox"/>	70	F
<input checked="" type="checkbox"/>	100	Spindle
<input checked="" type="checkbox"/>	900	Not Used

2.2 Moving the robot into position

To ensure the compatibility of the robot with PowerMill Robot, building a simulation must follow certain rules.

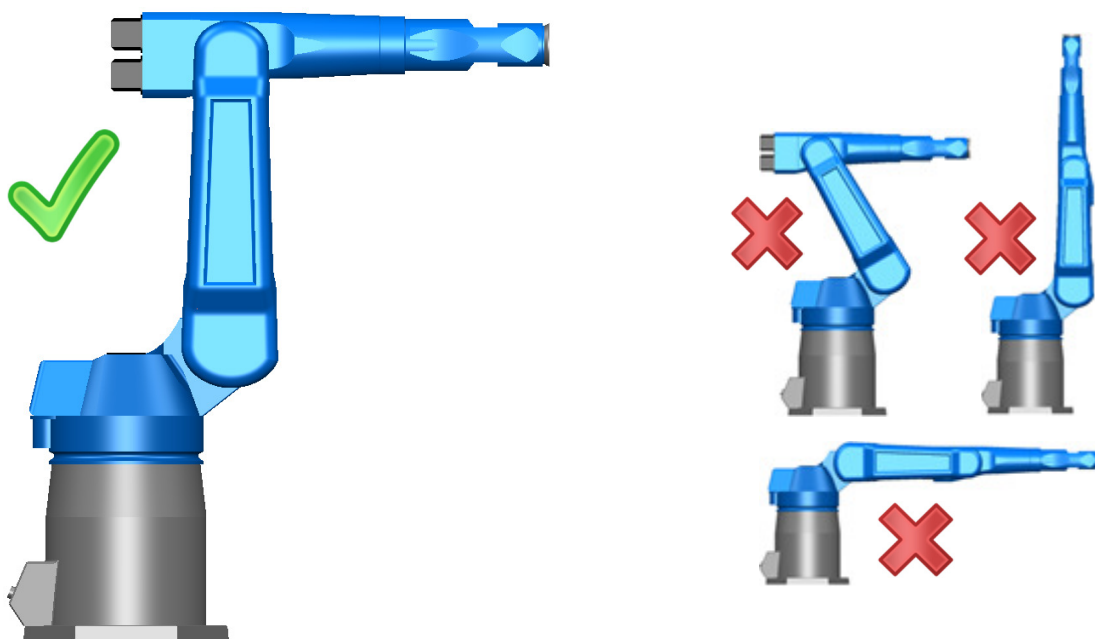
First, ensure that the robot world workplane is identical to the PowerShape world workplane as described below. If not, move the robot to the right position.

Note that the robot world workplane can be different depending on the robot manufacturer.



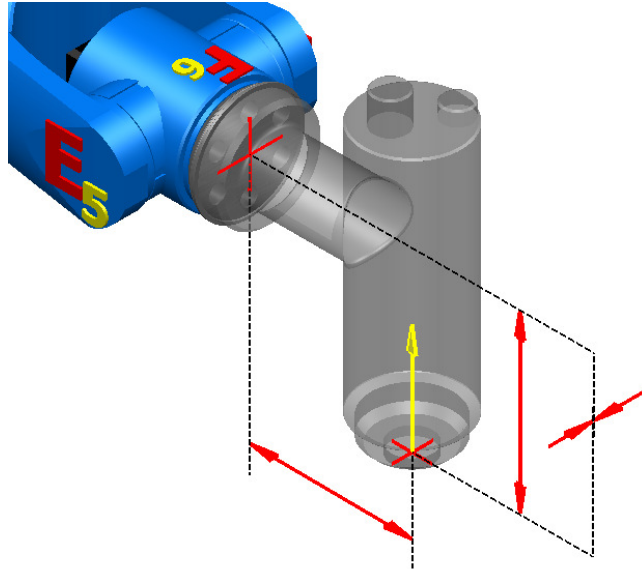
The next step is to move the robot in the position below, even if it is not the zero position of the robot axes.

In the majority of the cases you will not have to do this as the robot manufacturer provides the robot in this position already.



2.3 Importing and positioning of the spindle

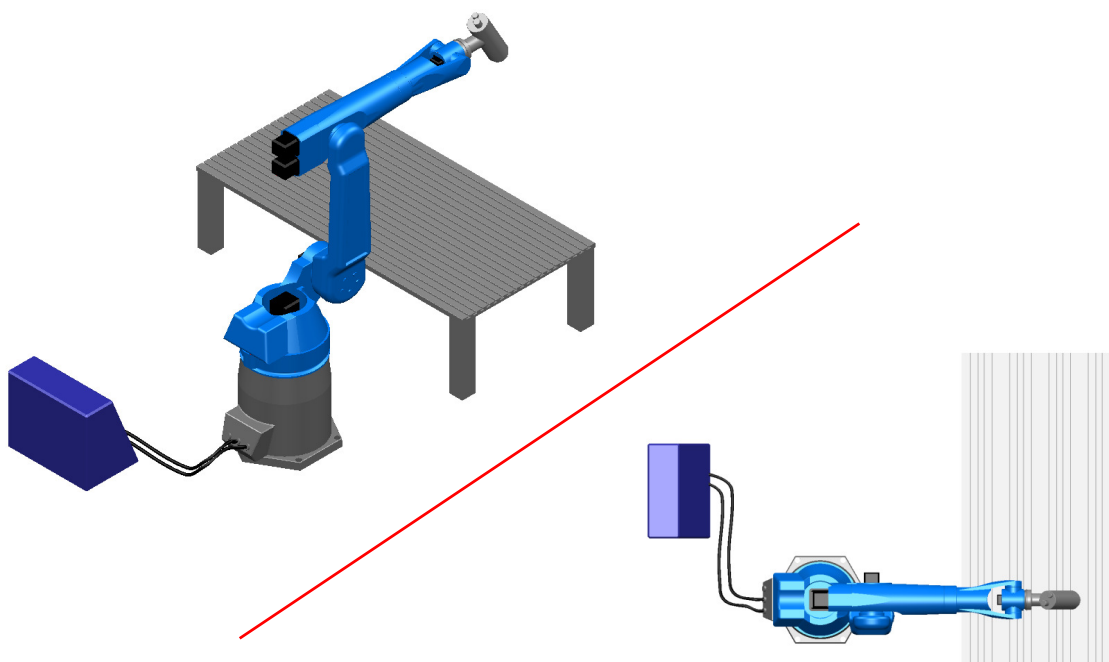
Import the spindle and move it to the correct position, using the coordinates measured on the “real” robot (or from a spindle calibration), as shown on the picture below:



2.4 Importing cell entities

If the robot cell contains a table, a controller or some other static entities that you want to see in the robot simulation, you can import or draw them into PowerShape now. Of course, the position must be accurate, as measured on the real robot cell.

At this stage, the PowerShape project must match the actual robot and everything must be there (except the tools).



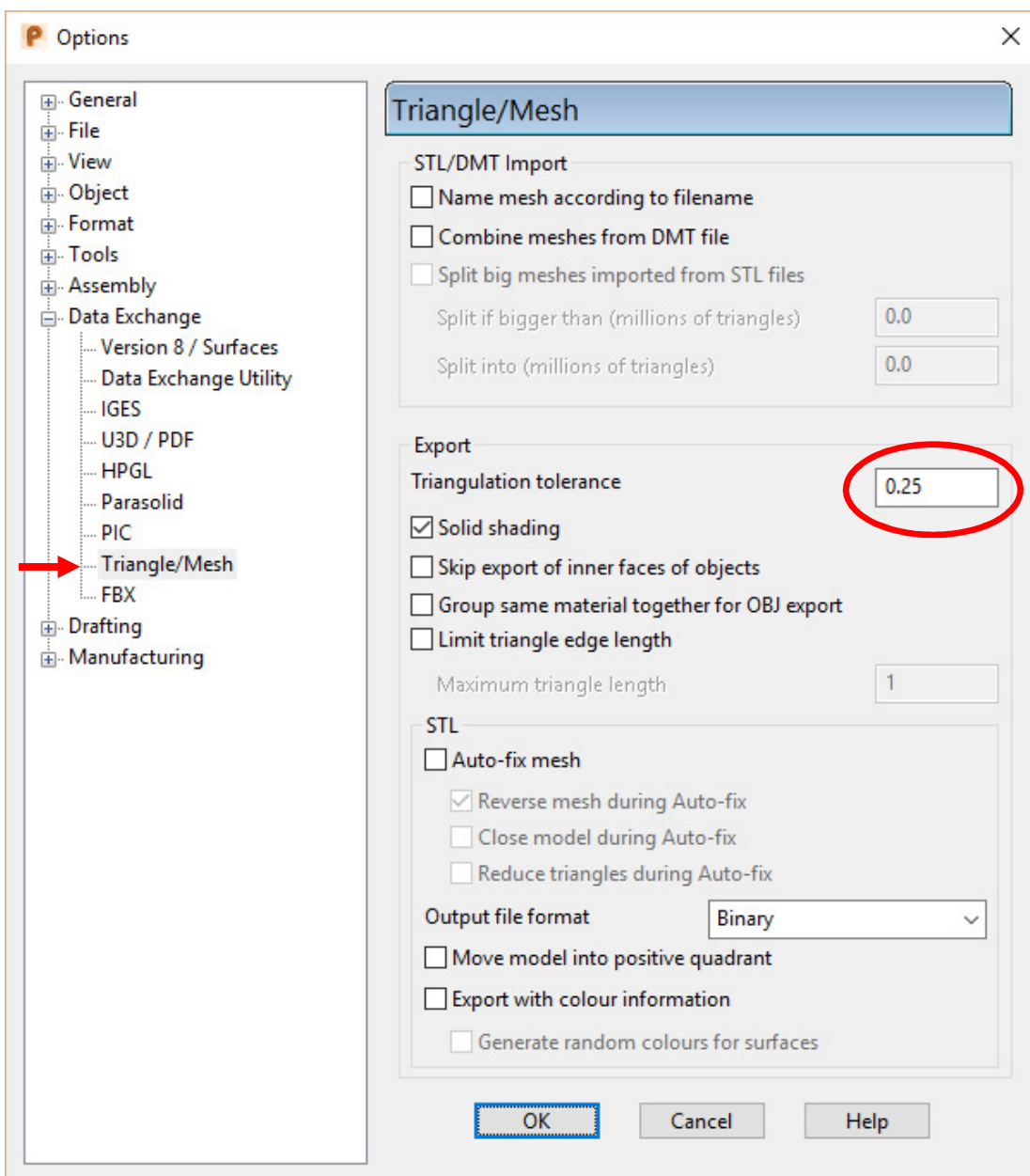
3. Export robot entities and its cell

Each robot axis, the engines, logos and all entities from the cell, must be exported into separate *.mtd files to be able to simulate them separately and to give them different colours.

The *.mtd file contains triangles. The more triangles, the better will the robot simulation look, but the slower will the simulation and collision checking run.

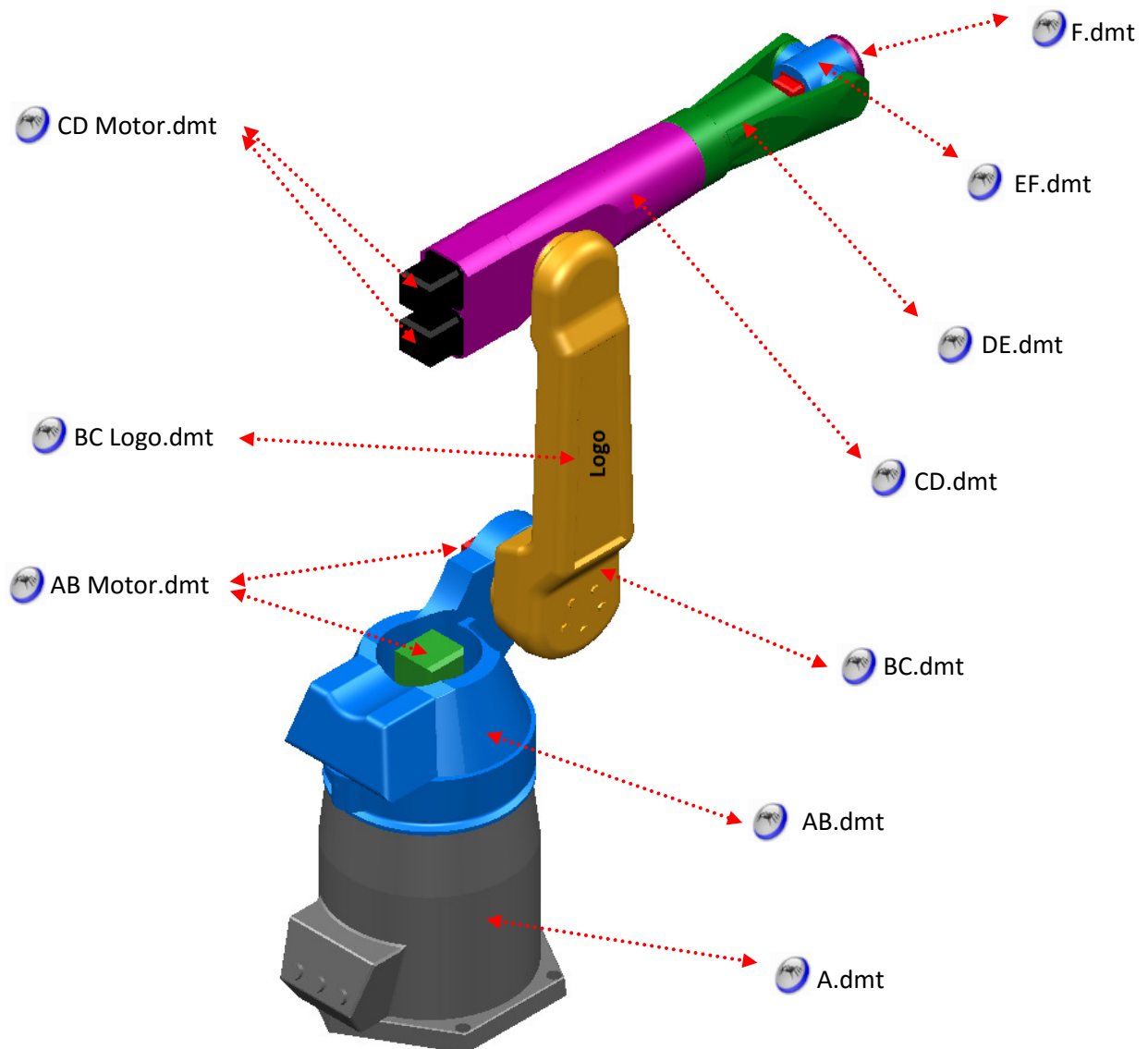
A tolerance value of 0.25mm is a good balance between quality and file size when exporting the data to *.dmt files. (If the file sizes are still too large or the computer too slow, increase this value).

The Triangulation tolerance can be set in Triangle/Mesh page of the PowerShape Options dialog.



Now export each entity of the robot and its cell into separate *.dmt files.

Make sure to use the same naming convention as below:



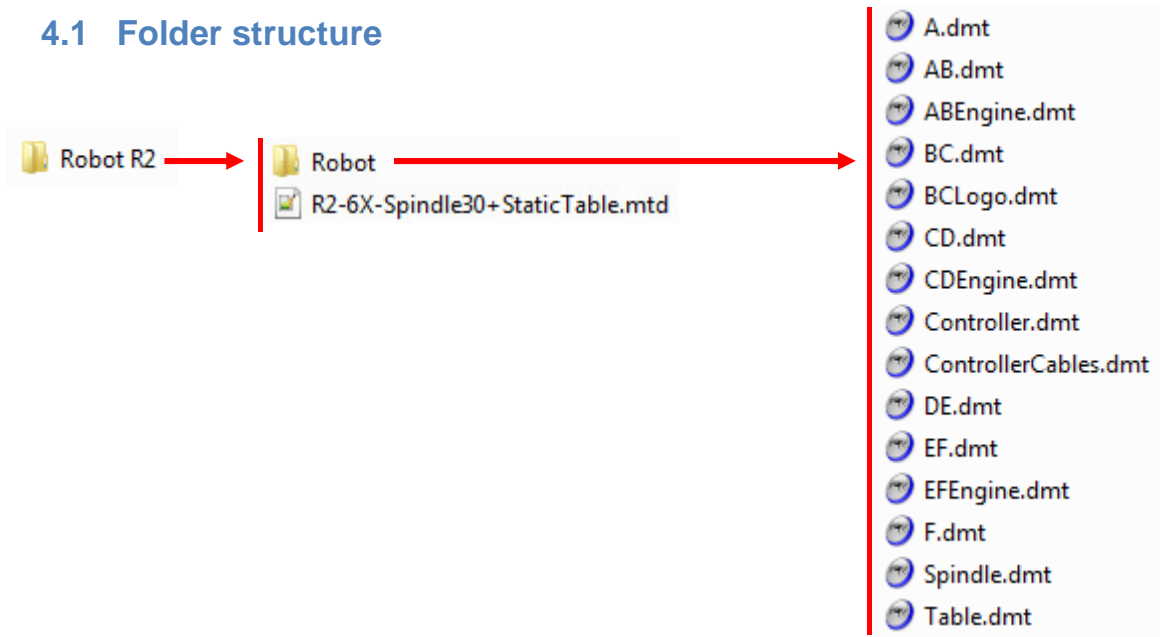
“A.dmt” can also be called “Base.dmt”

The spindle can be exported as “Spindle.dmt”, the table as “Table.dmt”...

If you are producing .mtd files regularly then PowerShape macros can be used to automate this part of the process.

4. PowerMill MTD file

4.1 Folder structure



4.2 *.mtd file structure

The *.mtd file contains all the information required to control the robot simulation and is specific to the robot it was created for. The most effective way to create a new *.mtd file is to copy an existing one and modify it. This process is described on the next page.

Here is a description of the *.mtd file structure for a robot holding the tool:

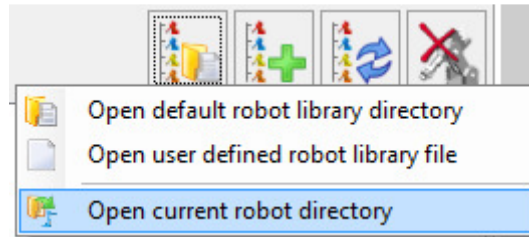
Make sure to respect this structure and format (xml format) for a better readability.

```
<machine xmlns="x-schema:PowerMillMachineTool" POST="SimPost. ">
  ..<head_attach_point PART="head" ...Coordinates... />
  ..<table_attach_point PART="table" ... Coordinates ... />
  ..<machine_part NAME="table" /> → the part is on the table (floor)
  ..<machine_part>
    ...List of static entities...
  ..</machine_part>
  ..<machine_part>
    ...Robot Axis 1...
    ....<machine_part>
      ...Robot Axis 2...
      .....<machine_part>
        ...Robot Axis 3...
        .....<machine_part>
          ...Robot Axis 4...
          .....<machine_part>
            ...Robot Axis 5...
            .....<machine_part>
              ...Robot Axis 6...
              .....<machine_part NAME="head" /> → the tool is on the head (robot)
              .....</machine_part>
            .....</machine_part>
          .....</machine_part>
        .....</machine_part>
      .....</machine_part>
    ....</machine_part>
  ..</machine_part>
</machine>
```


4.3 Modifying an Existing *.mtd File

From the PowerMill robot library, load in a robot from the same manufacturer as the robot you are creating a simulation for. Ideally load a robot with a similar configuration (that is, same number and type of external axes).

From the toolbar at the top of the Robot Library tab select Open Robot Library, and select Open Current Robot Directory.



This will take you to the directory containing the *.mtd file for that robot.

Select the appropriate *.mtd file from the folder and copy it into the location containing the folder with the exported CAD models.

Rename the *.mtd to give it a descriptive name, this is the name that will appear in the PowerMill Robot library when selecting a robot.

Open the file using a text editor such as Notepad++.

The following values will need to be updated for the new robot:

1. The file path for the CAD models. Provided the naming convention was followed when the CAD was exported only the folder name will need to be modified.
2. Head attach point.
3. Table attach point.
4. Axis minimum and maximum values and home positions.
5. Centre of rotation definitions. The rotation directions will remain unchanged in most instances.

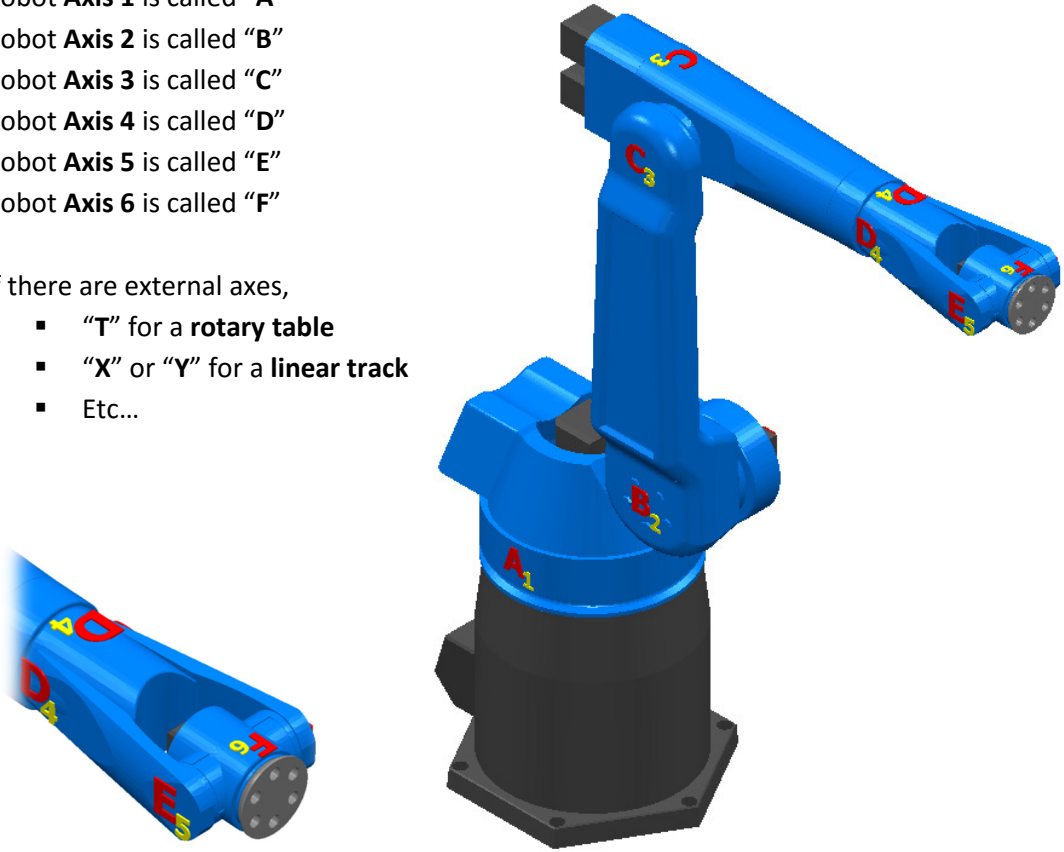
Details on how to determine values for these is included in the following pages.

4.4 Robot simulation

A standard robot has 6 axes. Make sure to use the naming convention below (A to F, one letter only):

- Robot **Axis 1** is called “**A**”
- Robot **Axis 2** is called “**B**”
- Robot **Axis 3** is called “**C**”
- Robot **Axis 4** is called “**D**”
- Robot **Axis 5** is called “**E**”
- Robot **Axis 6** is called “**F**”

- If there are external axes,
 - “**T**” for a **rotary table**
 - “**X**” or “**Y**” for a **linear track**
 - Etc...



More information about the default settings of a specific robot manufacturer is available in the help folder of PowerMill Robot. See files “*Robot Setup - xxx.pdf*”

The example below describes the properties of the second axis of the robot. The line numbers are only present to assist with identification of key components.

Any subset can be left out of the description, for example the axis definition should not be used for static parts and if no CAD model is to be used to represent movement then a model list is not required.

```
1) <machine_part>
2)   <axis>
3)     <control_info ADDRESS="B" MIN="-50" MAX="100" PRIORITY="AUTO" HOME="-30" VALUE="0"/>
4)     <simple_rotary X="327" Y="0.0" Z="916" I="0" J="1" K="0" />
5)   </axis>
6)   <model_list>
7)     <dmf_file>
8)       <path FILE="Robot/BC.dmf" />
9)       <rgb R="0" G="130" B="255" />
10)    </dmf_file>
11)    <dmf_file>
12)      <path FILE="Robot/BCLogo.dmf" />
13)      <rgb R="20" G="20" B="20" />
14)    </dmf_file>
15)  </model_list>
```

In this example:

Lines 2 to 5 contain all the information required to define the axis:

- This robot axis is the second one (starting from the floor) as the name is B.
- The minimum rotary angle is -50°.
- The maximum rotary angle is +100°.
- The priority of this axis is set to AUTO (default value).

Priorities can be set as: LOW, MEDIUM, HIGH, AUTO and STATIC.

AUTO means that linear axes will move before rotary axes.

STATIC means that the axes can be moved manually (for prepositioning) but will not be moved dynamically by the simulation.

- The HOME position of this axis is -30° (position when clicking on the HOME button).
- The VALUE of this axis when it was exported from PowerShape was 0°.
If the axis position is not 0° at its drawing position in PowerShape (paragraph 3.2), you can set the value here.
- The axis rotates about a line parallel to vector (0, 1, 0) passing through point (327, 0, 916) relative to the robot world work plane. To reverse the direction or rotation reverse the vector direction: (0,1,0) → (0,-1,0)

Lines 6 through 15 contain the information about the 3D models that move on the axis.

- The files "Robot\BC.dmf" and "Robot\BCLogo.dmf" are the CAD files representing this axis and are placed in the folder called "Robot". This folder is in the same file location as the *.mtd file.
- The colour code of the robot axis "BC.dmf" is: Red=0, Green=130, Blue=255.
- The colour code of the Logo "BCLogo.dmf" is: Red=20, Green=20, Blue=20.

Note: There is no closing statement for the machine part node (</machine_part>); this is because all subsequent axes (that is, all axes that are attached and move with this axis) must be defined within this machine part. Closing statements for all axes in the robot arm must appear after the definition of the 6th axis.

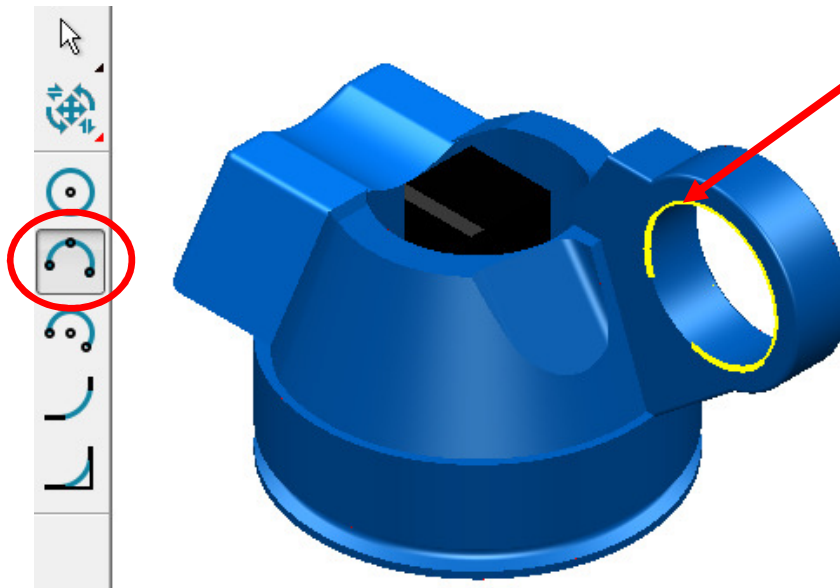
How to get the XYZ values (<simple_rotary X="327.141" Y="0.0" Z="916.1"...)?

An easy way of knowing these values is by getting the center point of each rotary axis of the robot, from PowerShape.

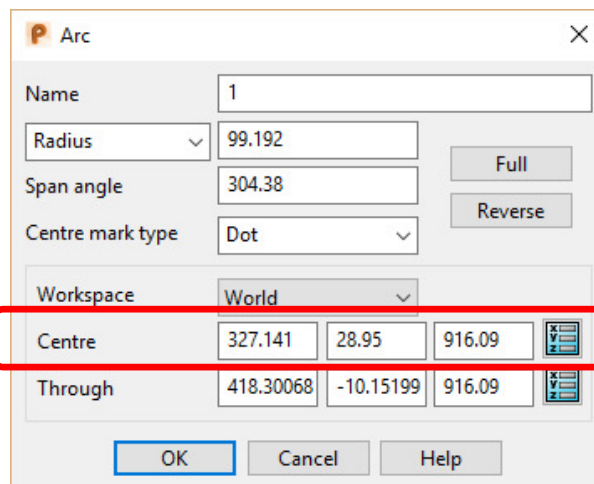
First, make sure the PowerShape world workplane is active:



Then, create a circle by three points around the rotary axis:



Then, editing this arc will give you the center point coordinate:



As this axis rotates around the vector (0,1,0) you do not need to specify the Y coordinate, so the result becomes:

```
<simple_rotary X="327.141" Y="0.0" Z="916.1" I="0" J="1" K="0" />
```

4.5 Tool and Table Attach Points

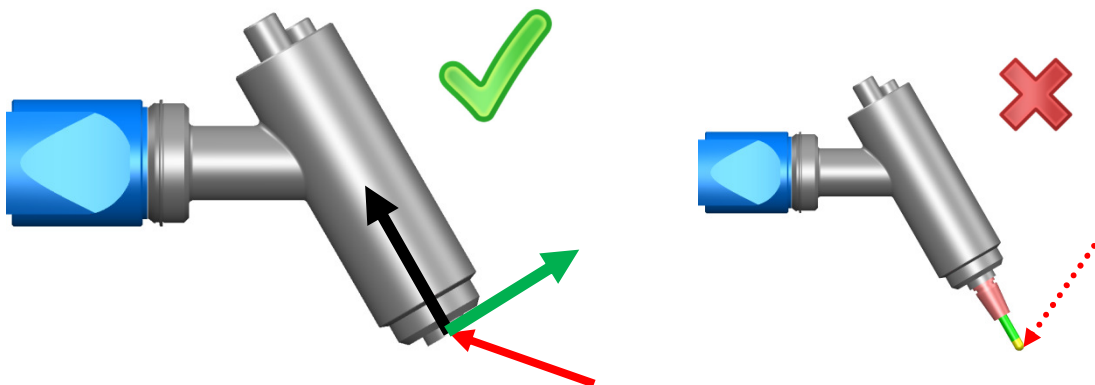
The information required to define the attach points for the tool (head) and the part (table) are identical. Only the process to describe head attach point is described in this document as the process can be more complex and often the table attach point will be left at the base of the robot.

This describes the position (and orientation) where the tool is attached to the spindle tip.
The reference of these coordinates is the robot world workplane.

```
<head_attach_point PART="head" X="1623.52" Y="-1.94" Z="1665.428" I="-0.5" J="0" K="0.866025"  
U="0.866025" V="0" W="0.5" />
```

The Part definition describes which machine part the tool is going to move with. This must match exactly the name of a machine part defined within the *.mtd file. In the case of a tool mounted on the robot arm, this machine part should be either the robot axis 6 or a machine part that moves with it.

For the table attach point the part should be a machine part that moves with any external axes controlling part positioning or a static part in the cell. If no external axes are used a static part will still be required to attach the part to.



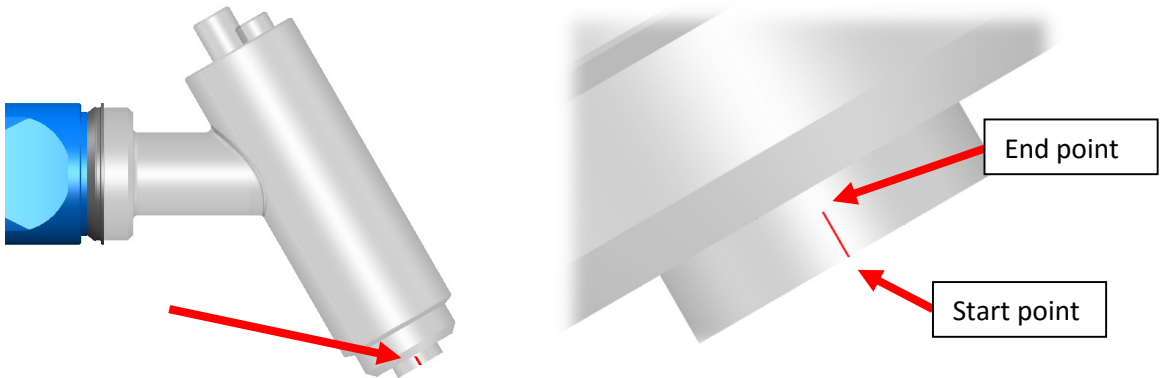
The red arrow shows the tool attach point position ($X="1623.52"$ $Y="-1.94"$ $Z="1665.428"$)

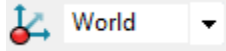
The black arrow describes the tool vector direction ($I="-0.5"$ $J="0"$ $K="0.866025"$)

The green arrow defines the direction of the tool workplane X axis ($U="0.866025"$ $V="0"$ $W="0.5"$)

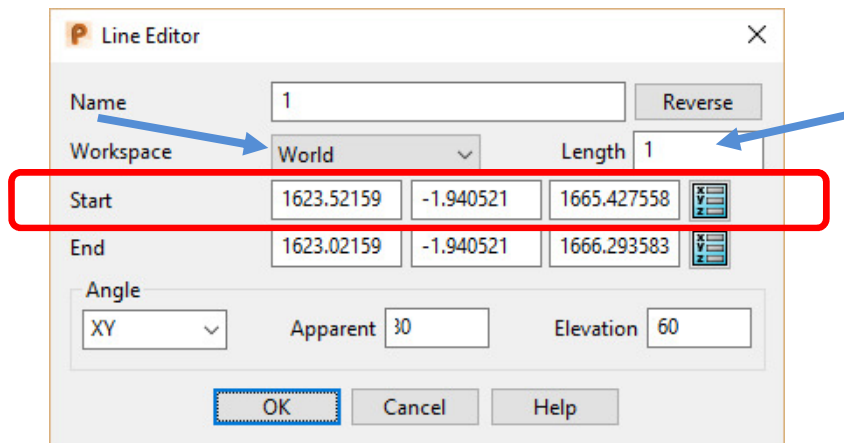
How to get the XYZ IJK UVW values (X="1623.52" Y="-1.94" Z="1665.428" I="-0.5" J="0" K="0.866025 U="1" V="0" W="0" U="0.866025" V="0" W="0.5")?

First, create a line, normal to the spindle face, with its start point on the spindle face and a length of 1mm:

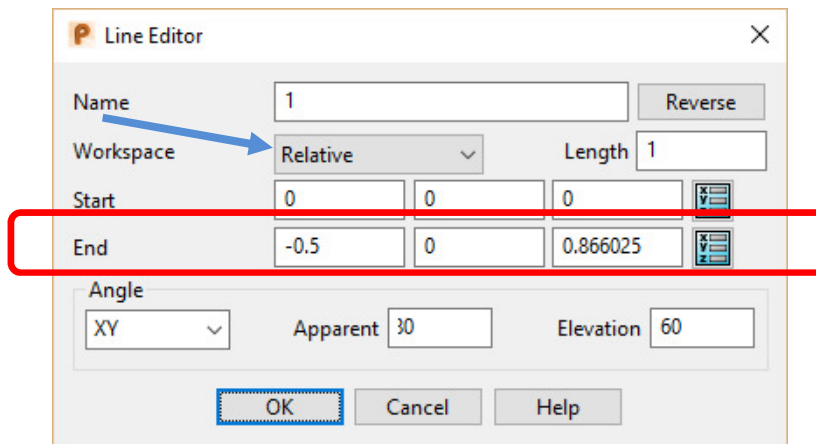


Then, make sure that PowerShape world workplane is active: 

For the XYZ coordinates, edit the line and use the start point coordinates:





To get the vector coordinates, change the "Workspace" to become "Relative":






The end point coordinates will give you the IJK values.

The UVW vector must be perpendicular to the IJK vector.

One method to calculate this vector is to select the vector defining the IJK direction, open the **Workplane** menu in PowerShape:  and select **Workplane Aligned to Geometry**:  click on either end of the line to create a new workplane.

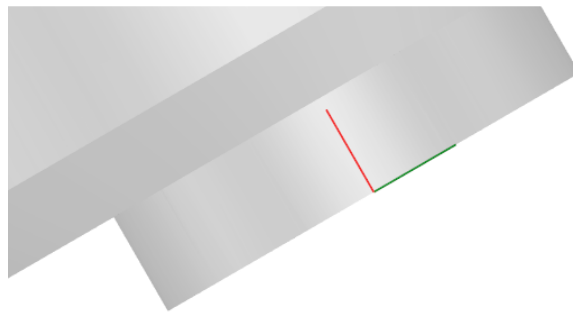
Select the line, open the **General Edits** menu:  and select **rotate**: 

Ensure either X or Y is selected as the principle axis:    and rotate the line 90°, ensuring that you are creating a single copy.

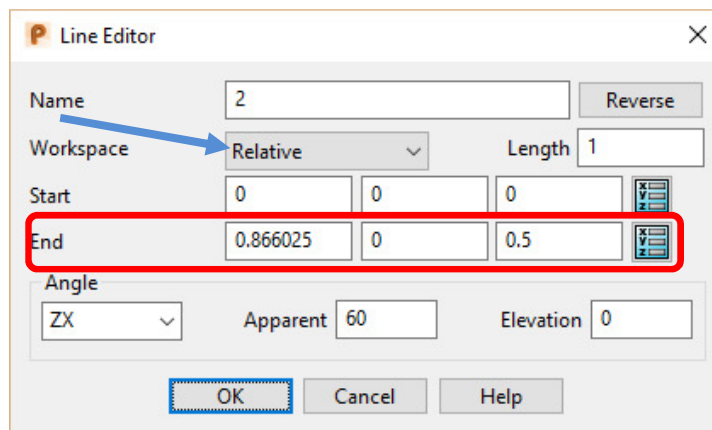


This new line can be rotated about the z axis of the created workplane to any orientation depending on the desired orientation of the tool workplane X axis.

Delete the created workplane.



Double click on the line and change the “Workspace” to “Relative”:



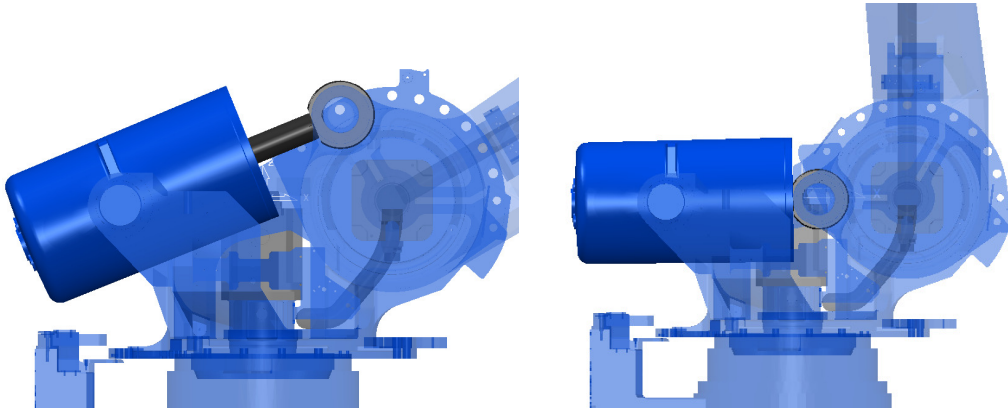
The end point coordinates will give you the UVW values. The resulting line in the *.mtd file will take the form:

```
<head_attach_point PART="head" X="1623.52" Y="-1.925" Z="1665.427"  
I="-0.5" J="0" K="0.866025" U="0.866025" V="0" W="0.5" />
```

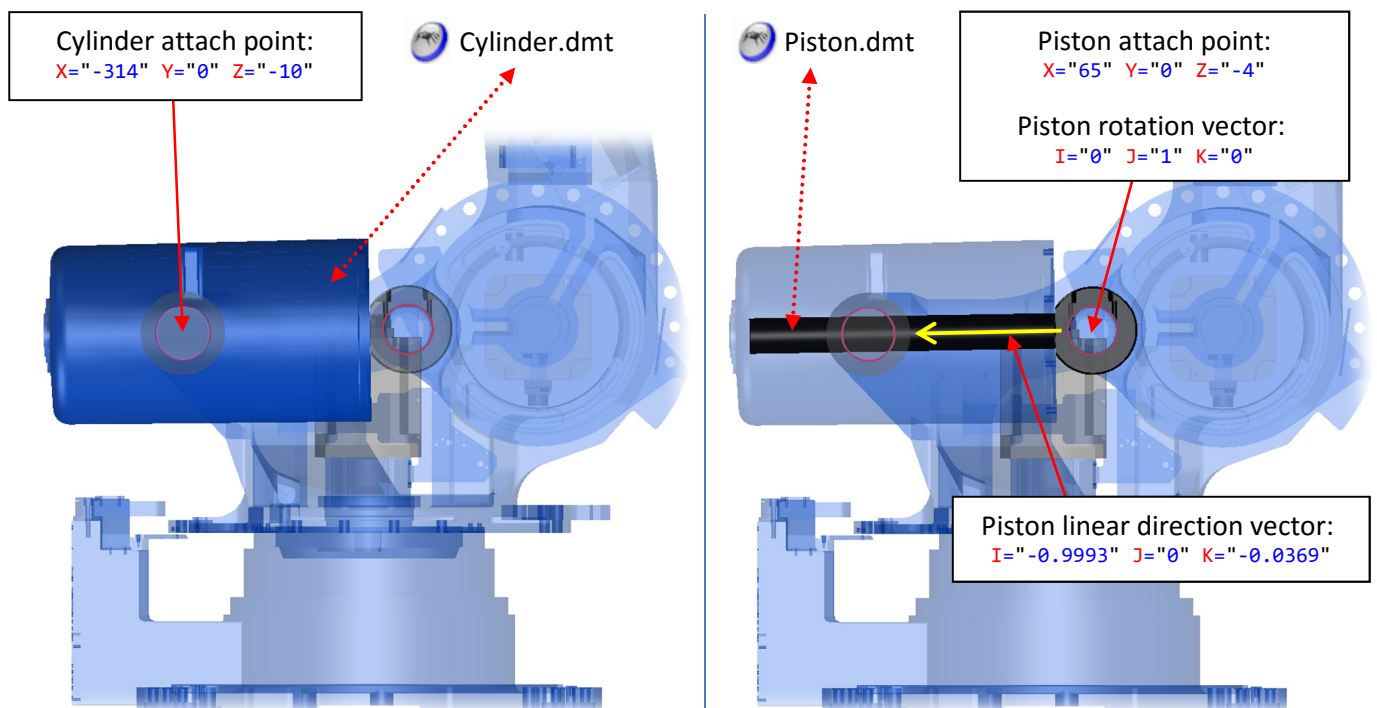
4.6 Creating Linked Conditions

In simulations it can be useful to be able to simulate axes movements that obey dependent relationships (that is 2 components that move relative to each other as a result of the movement of other axis components).

For example one end of the piston shown below moves with the second axis of the robot while the other remains attached to the first axis of the robot.



To achieve this a few values need to be determined; the axis and vectors of the movement and the CAD that will be constrained by this motion these are outlined in the image below.



To add this information to the mtd file there are a few key concepts to understand:

- The fixed points will move with whichever axes they are defined on in the mtd file.
- The machine parts for the dependent relationship are defined on the final axis they are attached to (that is, the axis moving with the highest number of other axes).
- The machine movements and order are determined from the final attach point and worked backwards.

In the above example the piston is connected between axes 1 and 2, the mtd file for these axes is listed below and a description of the functionality is included below.

```

<machine_part NAME="RobotAxis1">
  <axis>
    <control_info ADDRESS="A" MIN="-180" MAX="180" PRIORITY="AUTO" HOME="0"/>
    <simple_rotary X="0" Y="0" Z="0" I="0" J="0" K="1" />
  </axis>
  <fixed_point NAME="CylinderAttachPoint" X="-314" Y="0.0" Z="-10"/>
  ...
</machine_part NAME="RobotAxis1">
<machine_part NAME="RobotAxis2">
  <axis>
    <control_info ADDRESS="B" MIN="-60" MAX="76" PRIORITY="AUTO" HOME="0"/>
    <simple_rotary X="285" Y="0" Z="0" I="0" J="1" K="0" />
  </axis>
  ...
  <machine_part NAME="Piston">
    <link>
      <rotary X="65" Y="0.0" Z="-4" I="0" J="1" K="0" />
    </link>
    <model_list>
      <dmtd_file>
        <path FILE="Robot/Piston.dmt" />
        <rgb R="50" G="50" B="50" />
      </dmtd_file>
    </model_list>

    <machine_part NAME="Cylinder">
      <link>
        <linear I="-0.9993" J="0" K="-0.0369" />
        <fixed_point NAME="CylinderAttachPoint" />
      </link>
      <model_list>
        <dmtd_file>
          <path FILE="Robot/Cylinder.dmt" />
          <rgb R="0" G="73" B="219" />
        </dmtd_file>
      </model_list>

    </machine_part>
  </machine_part>

  <machine_part NAME="RobotAxis3">
    ...
  </machine_part>

```

Cylinder attach point declaration

Piston attach point and rotation vector

Piston *.dmt file

Cylinder attach point reference and piston linear direction vector

Cylinder *.dmt file

The cylinder attach point defined above is the point about which the piston assembly rotates on the first axis. This is defined in the machine part for robot axis 1 as the pivot point will rotate about axis A, but will not move when robot axis B is jogged.

On the second axis we have a new machine part defined called "Piston". This is separate to the main machine part containers and subsequent axes will not be contained within the definition of this part.

"Piston" has a link, not an axis, the rotary link allows the piston assembly to rotate freely on robot axis 2 about the defined rotation centre point and vector to maintain a set of constraints.

These constraints are set in the machine part "Cylinder". The link on this machine part limits the movement of "Cylinder" to a linear motion, relative to "Piston", to allow the point at the defined fixed point to remain stationary. The linear direction rotates with the "Piston" machine part.

Both "Piston" and "Cylinder" machine parts are closed once the definition is complete.

4.7 Robot cell

All entities attached to the floor and which are not moving must be declared at the beginning of the file and are not linked to any axis.

```
<machine_part NAME="StaticParts">
  <model_list>
    <dmt_file>
      <path FILE="Robot/A.dmt" />
      <rgb R="80" G="80" B="80" />
    </dmt_file>
    <dmt_file>
      <path FILE="Robot/Table.dmt" />
      <rgb R="140" G="140" B="140" />
    </dmt_file>
    <dmt_file>
      <path FILE="Robot/Controller.dmt" />
      <rgb R="100" G="100" B="100" />
    </dmt_file>
    <dmt_file>
      <path FILE="Robot/ControllerCables.dmt" />
      <rgb R="0" G="0" B="0" />
    </dmt_file>
  </model_list>
</machine_part>
```

External axes, such as rotary tables and positioners, are defined using the same text descriptions and structure as axes on the robot.

```
<machine_part NAME="PositionerAxis1">
  <axis>
    <control_info ADDRESS="R" MIN="-90" MAX="90" PRIORITY="STATIC" HOME="0"/>
    <simple_rotary X="1660" Y="0.0" Z="550.0" I="1" J="0" K="0" />
  </axis>
  <model_list>
    <dmt_file>
      <path FILE="Robot/PositionerAxis1.dmt" />
      <rgb R="80" G="80" B="80" />
    </dmt_file>
  </model_list>

  <machine_part NAME="PositionerAxis2">
    <axis>
      <control_info ADDRESS="T" MIN="-720" MAX="720" PRIORITY="STATIC" HOME="0"/>
      <simple_rotary X="1660" Y="0.0" Z="0.0" I="0" J="0" K="1" />
    </axis>
    <model_list>
      <dmt_file>
        <path FILE="Robot/PositionerAxis2.dmt" />
        <rgb R="0" G="130" B="255" />
      </dmt_file>
    </model_list>

    <machine_part NAME="table" />

  </machine_part>
</machine_part>
```