MFG226311

Introduction to PowerMill Robot in Subtractive Manufacturing

Tatenda Mushonga
Autodesk

Learning Objectives

- Setting up a robot cell and positioning a part in the cell
- Creating basic toolpaths in PowerMill
- Simulating toolpaths and checking for collisions in the robot cell
- Optimizing robot joint movements and external axes by applying constraints
- Posting code

Description

We typically see robotic arms being used in welding or pick and place applications on production lines. This class will cover some lesser known uses which are becoming more relevant in manufacturing with a focus on subtractive machining. Though robots are not as precise as traditional CNC machine tools, they offer versatility by allowing attachment of various end effectors such as CNC tool spindles, additive extruders and even grinders or polishers. They can thus be used as subtractive, additive and post processing machines. The PowerMill Robot plugin takes advantage of PowerMill’s industry leading 3- and 5-axis toolpaths and the toolpath editing features to apply them to a wide range of supported robots. We will primarily cover subtractive manufacturing in both industrial and artistic settings. You will learn the workflow from setting up a robot, calculating toolpaths and optimizing robot motion for efficiency and safety.

Speaker

Tatenda has been a Technical Consultant with Autodesk for the past two years after graduating with a Master’s degree in Robotics from the University of Pennsylvania. A manufacturing and automation enthusiast, he has experience designing and manufacturing parts for robotic systems both for fun and for research purposes. As part of Autodesk, he has trained and supported customers to manufacture parts better and faster using CAM packages such as FeatureCAM and PowerMIll. In addition to his role at Autodesk, he enjoys using his technical knowledge for social and environmental good such as developing prosthetics and building robots that assist with physical rehabilitation.

Introduction

PowerMill Robot is a plugin in PowerMill that generates robot joint movements from toolpaths calculated using PowerMill's various strategies and toolpath editing tools. It also enables you to optimize the moves for both efficiency and safety. In addition, it allows you to create custom robot cells with multiple configurations depending on your application.

Requirements

The Robot plugin is available in the premium and ultimate versions of PowerMill. Recommended system requirements are a Windows (7 or higher) 64-bit computer with a multi-core CPU and 16GB of RAM. A discrete Nvidia graphics card with 2GB RAM is also highly recommended.

Why use Robots in manufacturing?

Industrial robots are not yet capable of replacing traditional machine tools in most subtractive manufacturing settings. However, they offer several advantages over machine tools such as:

1. Machining Workspace:

Robots are generally not constrained within a machine housing and therefore can reach as far as their arms can extend. This allows them to machine large parts such as clay models of automobiles which would otherwise not fit on a machine tool's table.

2. Cost:

Most robotic arms have six joints. With these degrees of freedom, they can tackle 5-axis simultaneous toolpaths out of the box. Compared to 5-axis machine tools, their cost is very low for the kinematics they can achieve.

3. Versatility:

You can attach a variety of tools to a robot's end effector such as endmills, grinders and polishers. This allows you integrate multiple stages in the manufacturing process in one cell and control them with the same software.

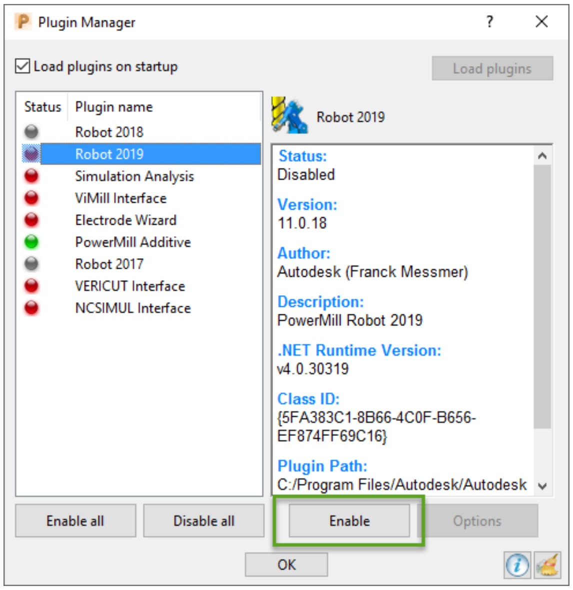
4. Collaboration

Multiple robots can be used in the same cell to collaborate on tasks. For example, one robot can be used for an additive process to extrude material while the other follows to apply a subtractive finishing pass on the deposited material.

Setting up the PowerMill Robot Plugin

To enable the plugin:

1. Click **File** from the main menu.
2. Select **Options**.
3. Click **Manage Installed Plugins**.
4. Select the latest Robot plugin and click **Enable.**



Plugin Manager

☑ Load plugins on startup

Status	Plugin name
●	Robot 2018
●	Robot 2019
●	Simulation Analysis
●	ViMill Interface
●	Electrode Wizard
●	PowerMill Additive
●	Robot 2017
●	VERICUT Interface
●	NCSIMUL Interface

Robot 2019

Status:
Disabled

Version:
11.0.18

Author:
Autodesk (Franck Messmer)

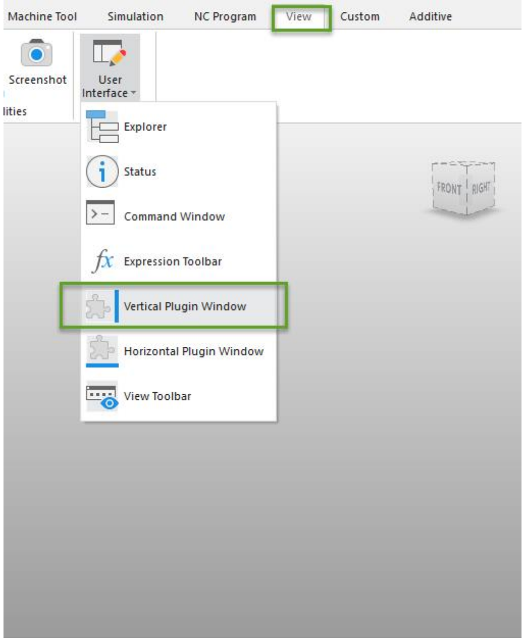
Description:
PowerMill Robot 2019

.NET Runtime Version:
v4.0.30319

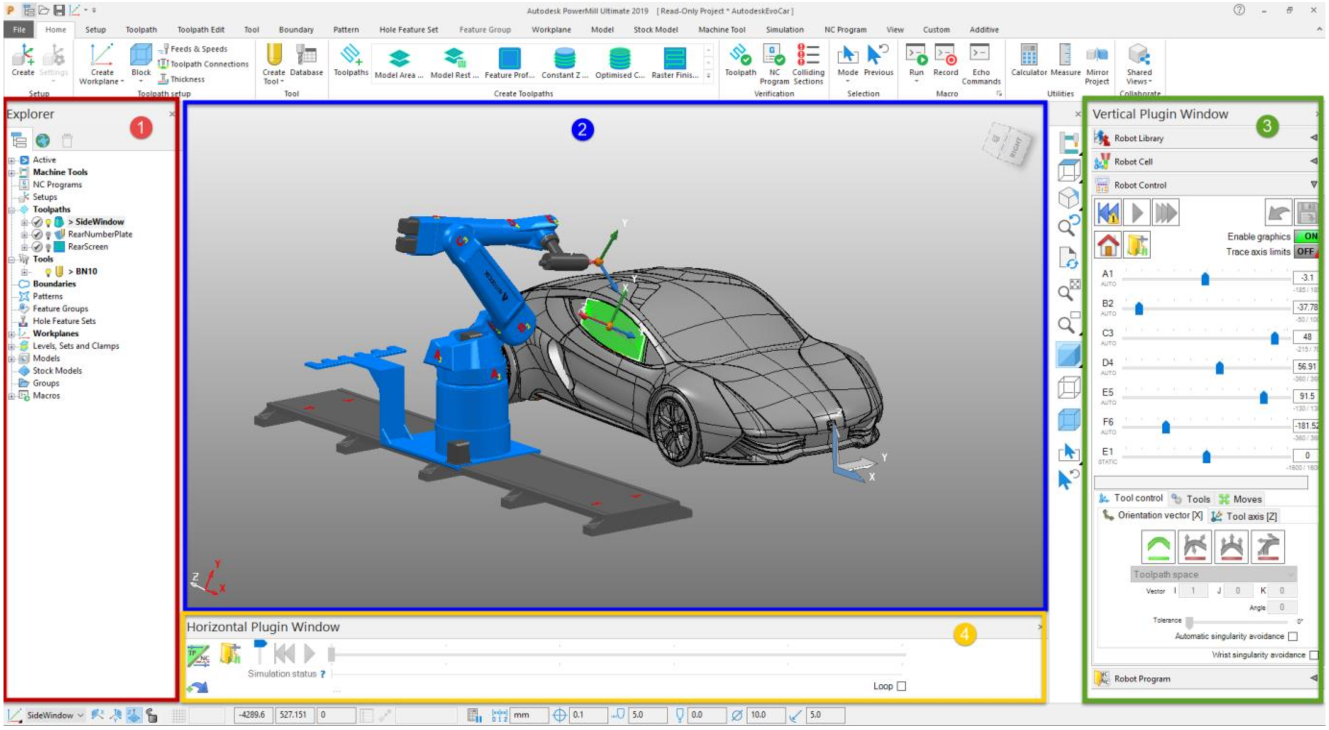
Class ID:
{5FA383C1-8B66-4C0F-B656-EF874FF69C16}

Plugin Path:
C:/Program Files/Autodesk/Autodesk

The PowerMill Robot Interface will be displayed in a vertical window to the right of the graphical area. If the vertical window is turned off, you can display it by clicking **View** > **User Interface** > **Vertical Plugin Window**:



User Interface Layout



1. Explorer

The explorer lists entities of the project such as toolpaths, boundaries, tools and workplanes.

2. Graphical Window

This is where everything within the project such as the robot, model, toolpaths, etc can be visualized.

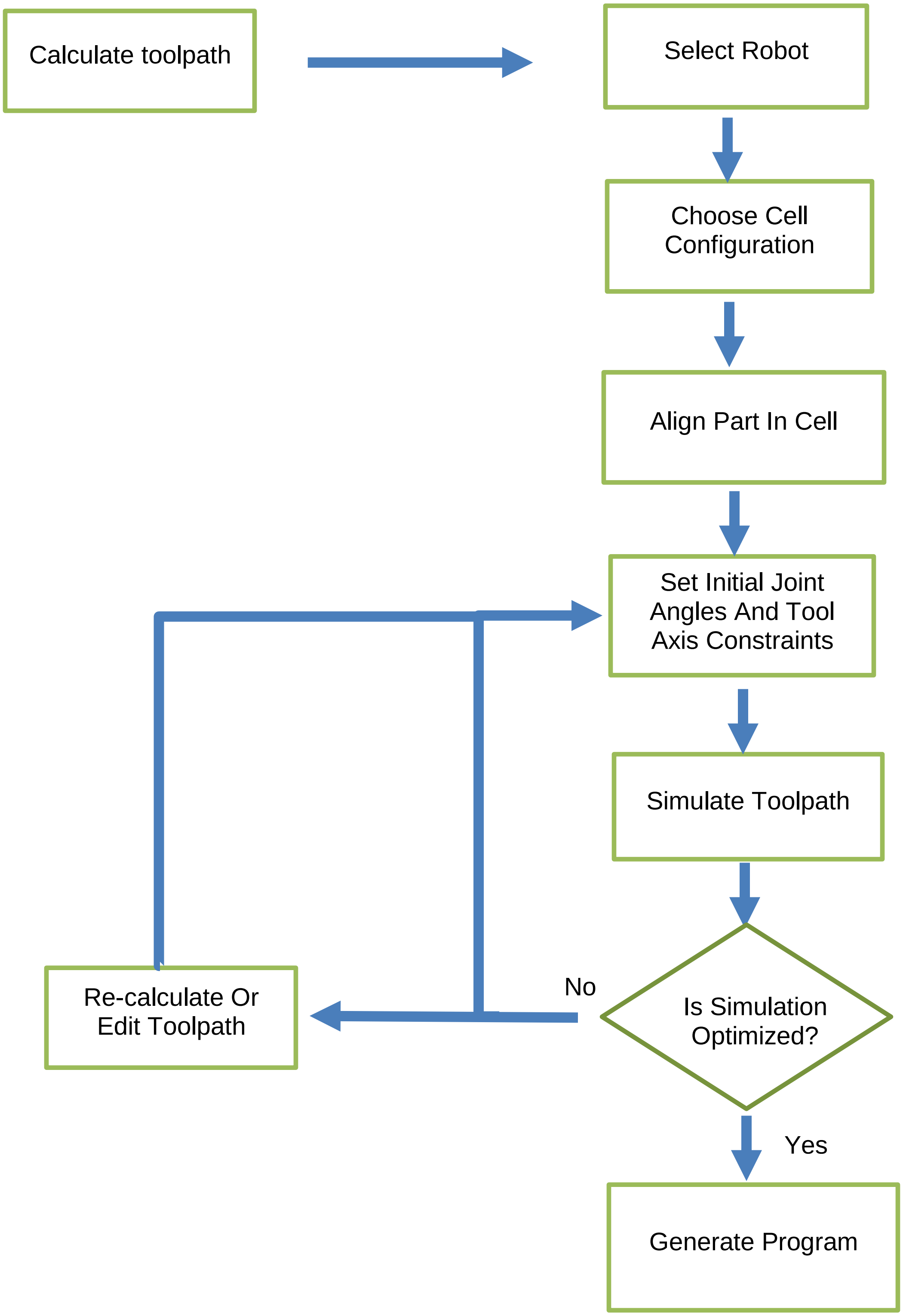
3. PowerMill Robot Plugin

The plugin area has four tabs arranged in the typical order you would follow in the workflow. Each tab will be explained in greater detail at each stage of the process.

4. PowerMill Robot Simulation Toolbar

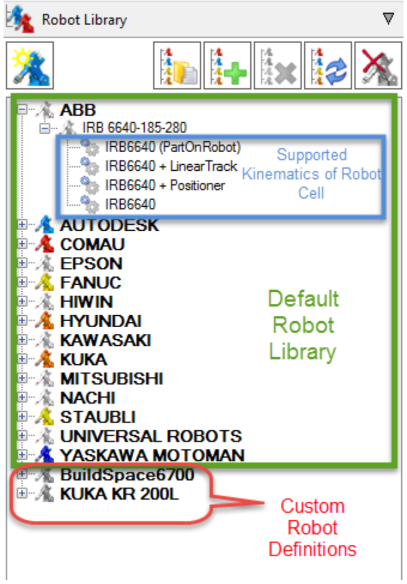
The simulation toolbar provides controls for playing back saved simulation files. It also displays issues with the toolpath along the timeline such as collisions, axis limits and singularities.

Workflow


Calculate toolpath
Select Robot
Choose Cell Configuration
Align Part In Cell
Set Initial Joint Angles And Tool Axis Constraints
Simulate Toolpath
Is Simulation Optimized?
No
Re-calculate Or Edit Toolpath
Yes
Generate Program

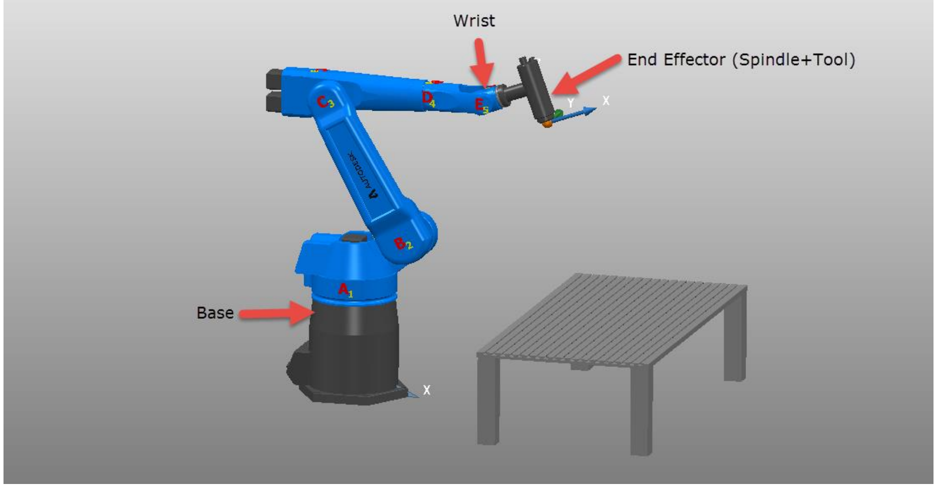
Loading a Robot Cell

You can choose a robot cell from the ones available in the default library. These can be found under the Robot Library tab. Each robot model will have different configurations and kinematics it supports. For example, the same robot can machine a part on a table or it can hold the part on the end effector. You can also create and customize your own cell if your robot is not among the default models.



Loading a Default Robot Cell

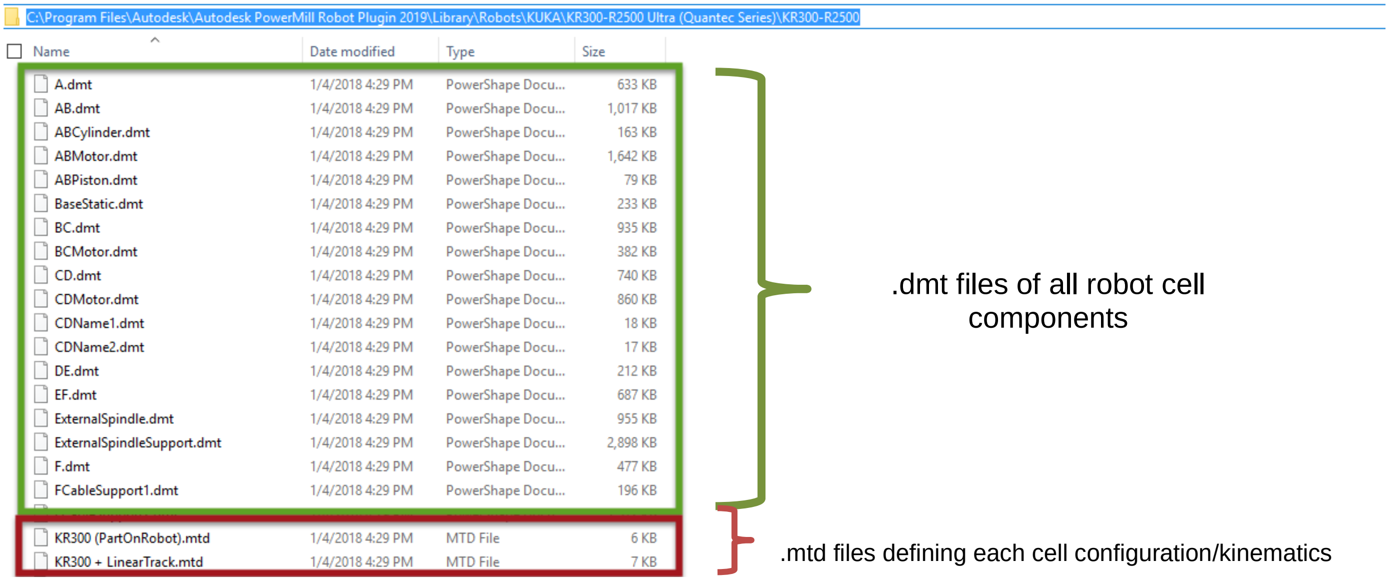
To load a robot from the default list, simply double click on it from the list in the Robot Library tab. For this demo, choose “**R2-6X-Spindle0+StaticTable**” under Autodesk. The setup of this cell is the simplest kinematic in PowerMill Robot where there is one robot and no external axes being controlled. There are six joints: A1, B2, C3, F4, E5 and F6 from the robot’s base to the end effector respectively.



Creating and Customizing Your Own Robot Cell

Not all models from each manufacturer are present in the default library; therefore, there will be situations where you need to add your robot to the library and customize the cell.

The robot can be defined in a similar way to a machine tool using a Machine Tool Definition (.mtd) file. The .mtd file is an XML style format with definitions of each robot/machine component, position, axis limits, file location etc. CAD models of the robot can be obtained from the manufacturer's website or from your reseller. The model has to be split into its individual components and each component saved in a Delcam Machining Triangles (.dmt) format. Tools such as **Autodesk Powershape** or the **Autodesk Manufacturing Data Exchange Utility** can be used to convert the models to this format. The .mtd file will then define how each of the .dmt files behaves and specify its axis limits.



C:\Program Files\Autodesk\Autodesk PowerMill Robot Plugin 2019\Library\Robots\KUKA\KR300-R2500 Ultra (Quantec Series)\KR300-R2500

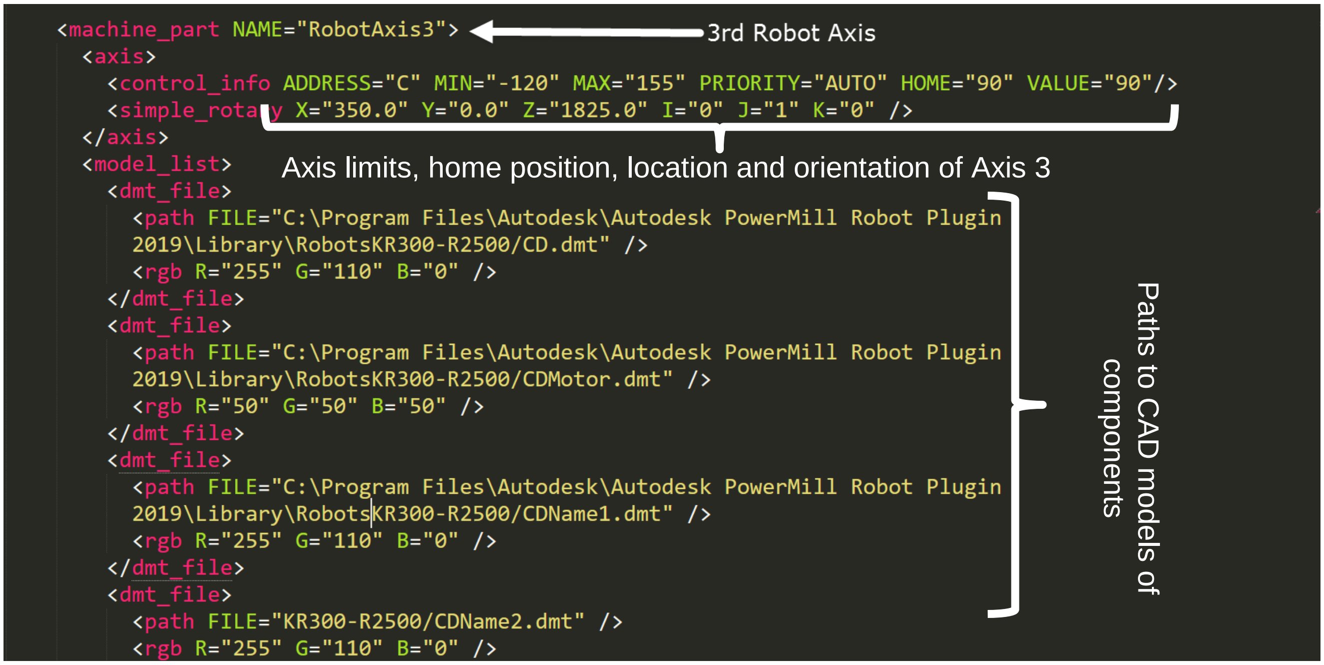
Name	Date modified	Type	Size
A.dmt	1/4/2018 4:29 PM	PowerShape Docu...	633 KB
AB.dmt	1/4/2018 4:29 PM	PowerShape Docu...	1,017 KB
ABCylinder.dmt	1/4/2018 4:29 PM	PowerShape Docu...	163 KB
ABMotor.dmt	1/4/2018 4:29 PM	PowerShape Docu...	1,642 KB
ABPiston.dmt	1/4/2018 4:29 PM	PowerShape Docu...	79 KB
BaseStatic.dmt	1/4/2018 4:29 PM	PowerShape Docu...	233 KB
BC.dmt	1/4/2018 4:29 PM	PowerShape Docu...	935 KB
BCMotor.dmt	1/4/2018 4:29 PM	PowerShape Docu...	382 KB
CD.dmt	1/4/2018 4:29 PM	PowerShape Docu...	740 KB
CDMotor.dmt	1/4/2018 4:29 PM	PowerShape Docu...	860 KB
CDName1.dmt	1/4/2018 4:29 PM	PowerShape Docu...	18 KB
CDName2.dmt	1/4/2018 4:29 PM	PowerShape Docu...	17 KB
DE.dmt	1/4/2018 4:29 PM	PowerShape Docu...	212 KB
EF.dmt	1/4/2018 4:29 PM	PowerShape Docu...	687 KB
ExternalSpindle.dmt	1/4/2018 4:29 PM	PowerShape Docu...	955 KB
ExternalSpindleSupport.dmt	1/4/2018 4:29 PM	PowerShape Docu...	2,898 KB
F.dmt	1/4/2018 4:29 PM	PowerShape Docu...	477 KB
FCableSupport1.dmt	1/4/2018 4:29 PM	PowerShape Docu...	196 KB
KR300 (PartOnRobot).mtd	1/4/2018 4:29 PM	MTD File	6 KB
KR300 + LinearTrack.mtd	1/4/2018 4:29 PM	MTD File	7 KB

.dmt files of all robot cell components

.mtd files defining each cell configuration/kinematics

Sample file arrangement of robot components and .mtd file.

You will also need the datasheet of the robot from the manufacturer or reseller. The datasheet will have specifications of dimensions between the joints, their limits, the home position, etc. These parameters will be needed for the .mtd file. It can be created/edited using a text editor of your choice. It is recommended to use an editor with syntax highlighting like Notepad++ or Sublime Text to visualize the tags easily.



```
<machine_part NAME="RobotAxis3">
  <axis>
    <control_info ADDRESS="C" MIN="-120" MAX="155" PRIORITY="AUTO" HOME="90" VALUE="90"/>
    <simple_rotary X="350.0" Y="0.0" Z="1825.0" I="0" J="1" K="0" />
  </axis>
  <model_list>
    <dmt_file>
      <path FILE="C:\Program Files\Autodesk\Autodesk PowerMill Robot Plugin
      2019\Library\RobotsKR300-R2500/CD.dmt" />
      <rgb R="255" G="110" B="0" />
    </dmt_file>
    <dmt_file>
      <path FILE="C:\Program Files\Autodesk\Autodesk PowerMill Robot Plugin
      2019\Library\RobotsKR300-R2500/CDMotor.dmt" />
      <rgb R="50" G="50" B="50" />
    </dmt_file>
    <dmt_file>
      <path FILE="C:\Program Files\Autodesk\Autodesk PowerMill Robot Plugin
      2019\Library\RobotsKR300-R2500/CDName1.dmt" />
      <rgb R="255" G="110" B="0" />
    </dmt_file>
    <dmt_file>
      <path FILE="KR300-R2500/CDName2.dmt" />
      <rgb R="255" G="110" B="0" />
```

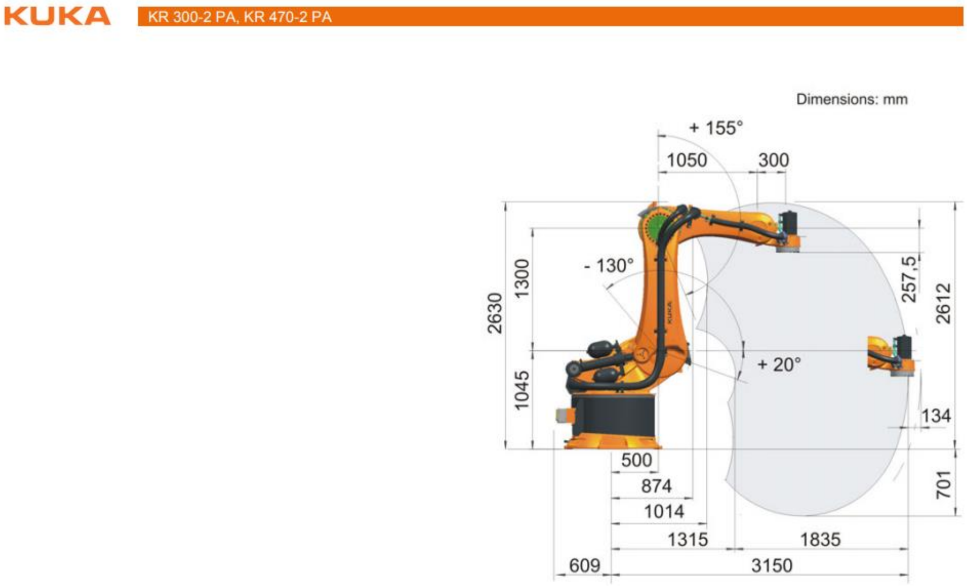
3rd Robot Axis

Axis limits, home position, location and orientation of Axis 3

Paths to CAD models of components

Example section of .mtd file showing definition of a robot axis

As shown in the section of the .mtd above, the robot can be defined simply by specifying each component's position from the robot base and pointing to the file location of its CAD models. You can use any of the .mtd files of the default robots as a template to define your own robot. They can be found in the help folder: `C:\Program Files\Autodesk\Autodesk PowerMill Robot Plugin 2019\Library\Robots`. The datasheet will include information such as dimensions shown below to fully define your robot.

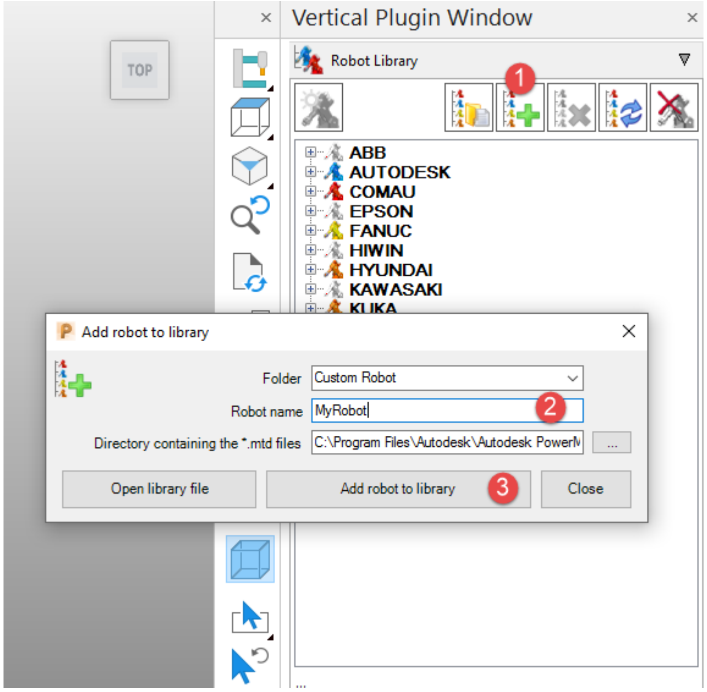


In addition to defining the robot axes and their limits, you will also need to define the head attach point and the table attach point. The head attach point is where the endmill or whichever tool being used attaches to the spindle. The table attach point defines where the base of the robot attaches to the table/floor.

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

  <!-- Where tool is attached -->
  <head_attach_point PART="head" X="1720" Y="0" Z="1510" I="0" J="0" K="1" U="1" V="0" W="0" />
  <!-- Offset of the world workplane -->
  <table_attach_point PART="table" X="0" Y="0" Z="0" I="0" J="0" K="1" U="1" V="0" W="0" />
```

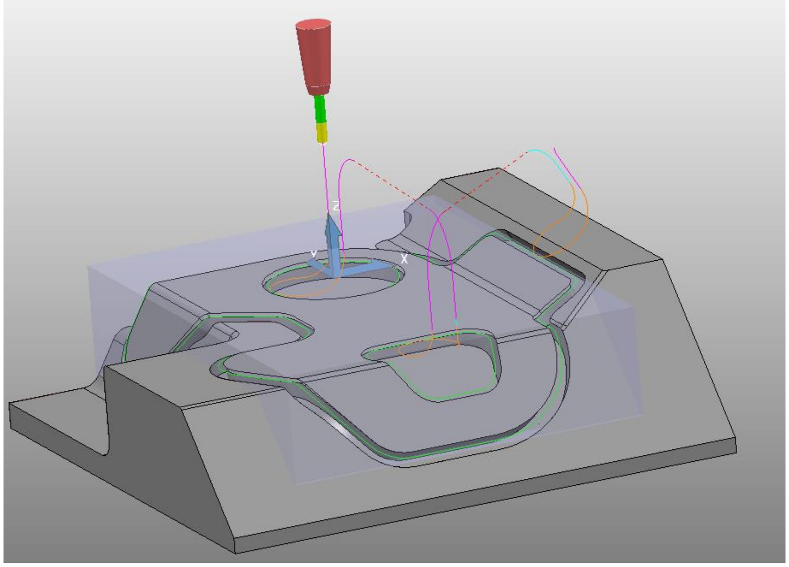
Once you have defined the .mtd file, you can then add the robot to the library by clicking the **Add a robot to library** icon and specifying the path where the .mtd file resides:



Folder: Custom Robot
Robot name: MyRobot
Directory containing the *.mtd files: C:\Program Files\Autodesk\Autodesk PowerM

Creating Toolpaths

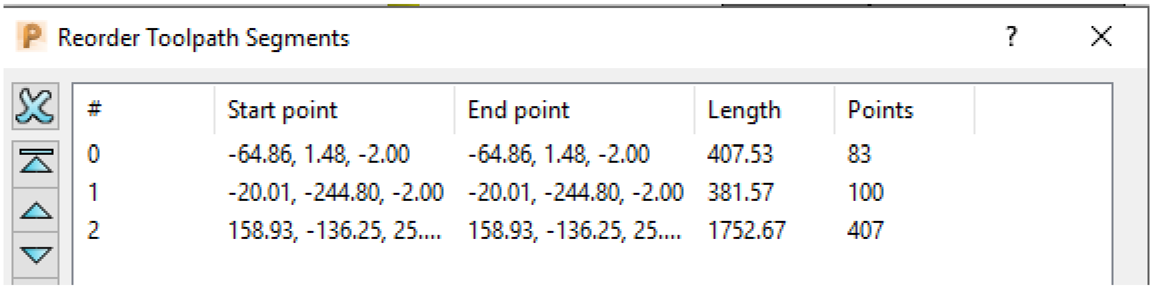
We will use the projects from the PowerMill Robot help folder for the demo. Open the **DemoToolControl6DWithFixture** project from `C:\Program Files\Autodesk\Autodesk PowerMill Robot Plugin 2019\Library\Projects.` You will notice that the project has three toolpaths already calculated. Simulate these toolpaths and observe their behavior. We will re-create them and then apply one of them to a robot. We will emphasize the simulation and optimization on the outside profile toolpath.



To calculate the toolpath:

1. Expand the **Levels, Sets and Clamps** in the explorer.
2. Right-click the "**Surfaces**" level and click **Select Surfaces** to highlight the part surfaces.
3. Click on the **Toolpaths** icon from the Home tab.
4. Choose **Profile Finishing.**
5. In the toolpath form, set the tolerance to 0.25mm, Axial offset to -2mm, Drive curve to **Outside edge** and Tool Axis to **Lead/Lean**.
6. Click **Calculate** and you will get a toolpath called "1" that includes the outer profile and the pockets.

We will now separate the toolpath into individual segments. First, we must visualize how long each segment is by clicking on **Reorder** from the **Toolpath Edit** tab. The segments are 407.53mm, 381.57mm and 1752.67mm respectively.



#	Start point	End point	Length	Points
0	-64.86, 1.48, -2.00	-64.86, 1.48, -2.00	407.53	83
1	-20.01, -244.80, -2.00	-20.01, -244.80, -2.00	381.57	100
2	158.93, -136.25, 25....	158.93, -136.25, 25....	1752.67	407

1. From the **Toolpath Edit** tab click **Divide.**
2. Under the **Length** tab we can enter any value less than the smallest segment length (381.57) to keep all segments and split them.
3. Click **Apply** and then close the dialog.
4. 3 separate toolpaths will be created. Activate the one that machines the outer profile and then rename it to "Outer Profile".

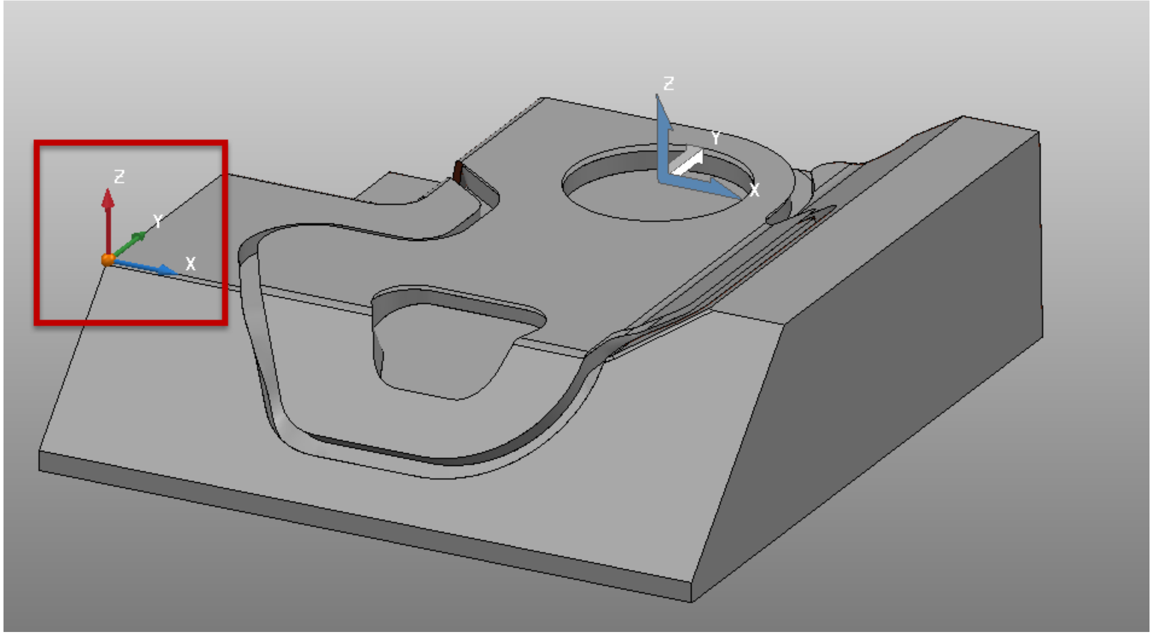
Next, we will position the part in a robot cell and then simulate the toolpath on a robot.

Positioning The Part In A Cell

Once you have the toolpath, it can be applied to any robot in your library.
Load the Autodesk **R2-6X-Spindle0+StaticTable** from the Robot Library tab. Once loaded, the robot base will be positioned at the same location as the part, triggering a collision. The part must be moved to a location on the table to be within the robot's accessible workspace.

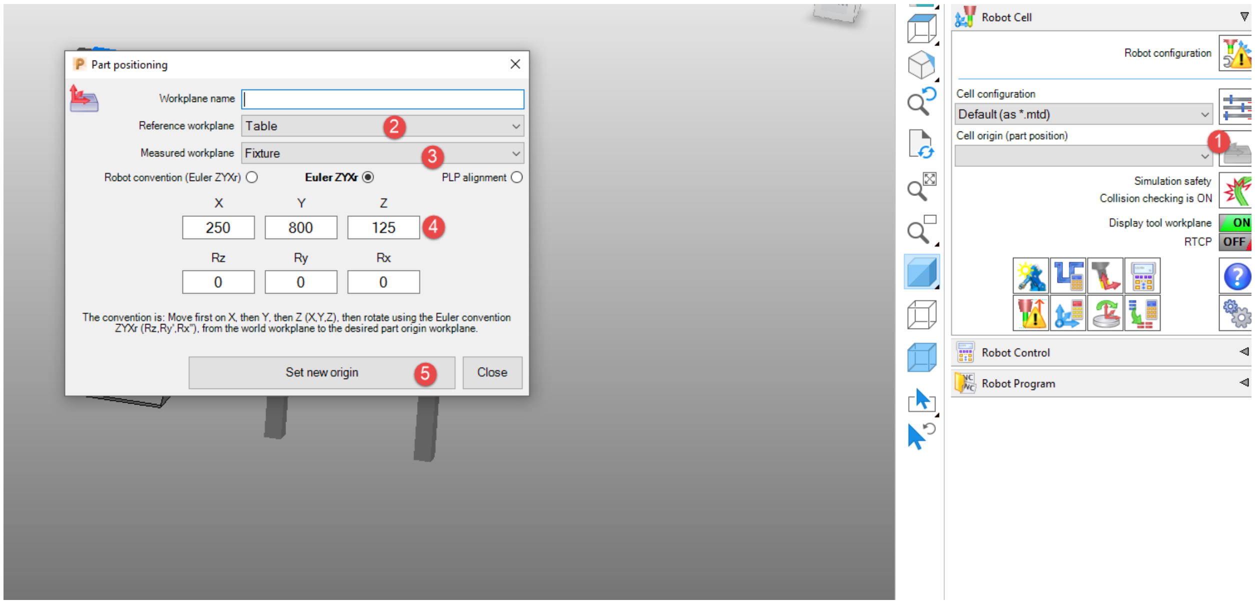
Create a workplane at the corner of the table by choosing **Workplane at Point** from the Home tab. Use the **Workplane Editor** to change the location in the global transform to X 1000, Y -1000, Z 530. Change the name to "Table".

Create a workplane called "Fixture" at the top of the fixture by snapping onto the vertex shown below:
Note: It might be useful to hide the robot in order to see the part easily by clicking the ***Display/Hide Robot*** *icon from the Robot Library tab:*



To set the part on the table using the created workplanes, we will use the **Robot Cell Tab.**

1. Click the **Part Positioning** icon from the Robot Cell tab.
2. In the Part Positioning form, set the **Reference Workplane** to “Table”.
3. Set the **Measured Workplane** to “Fixture.”
4. Set the position coordinates to X 250, Y 800, Z 125.
5. Click **Set New Origin.**



Part positioning

Workplane name:

Reference workplane: Table

Measured workplane: Fixture

Robot convention (Euler ZYXr) ○ Euler ZYXr ● PLP alignment ○

X	Y	Z
250	800	125

Rz	Ry	Rx
0	0	0

The convention is: Move first on X, then Y, then Z (X,Y,Z), then rotate using the Euler convention ZYXr (Rz,Ry',Rx''), from the world workplane to the desired part origin workplane.

Robot Cell

Robot configuration

Cell configuration: Default (as *.mtd)

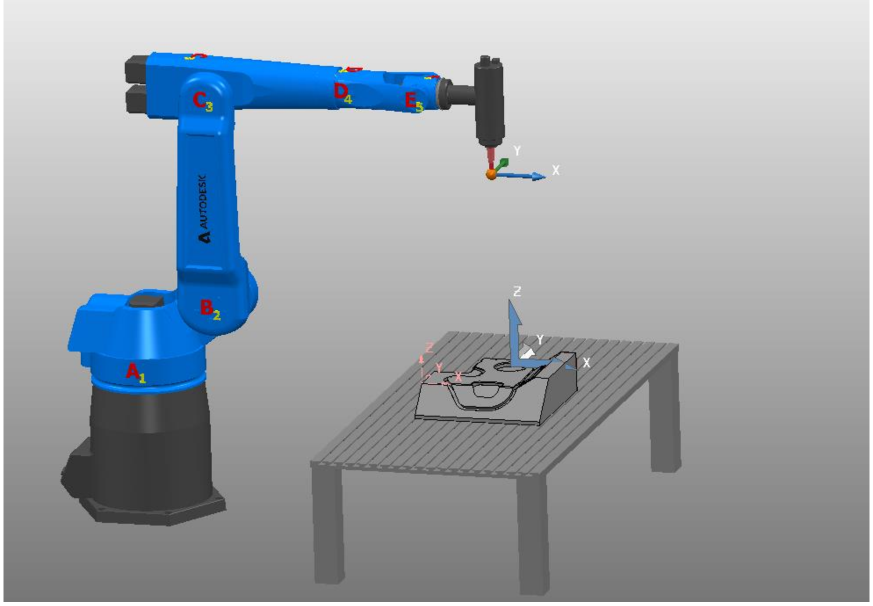
Cell origin (part position)

Simulation safety: Collision checking is ON

Display tool workplane: ON

RTCP: OFF

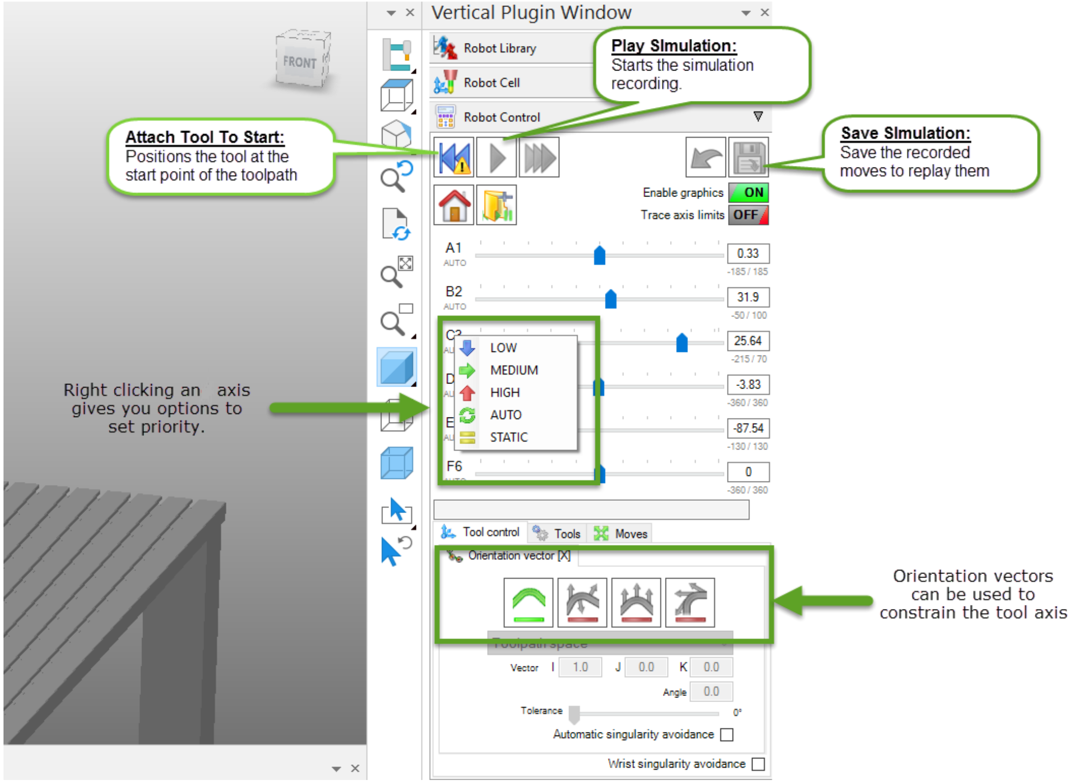
The part is now positioned, and we are ready to simulate the toolpath. The part and the robot should be aligned as shown below:



Simulating the Toolpath and Checking for Collisions

Unlike machine tools that require cartesian coordinates for NC code, robots require angular values for each robot joint through every move of the toolpath. In order to get these values, inverse kinematics have to be applied for each single point of the toolpath. Therefore, it is required to simulate the entire toolpath from start to finish to get a solution to post to the robot.

In PowerMill Robot, the initial position of the robot in simulation is crucial because it will affect all subsequent orientations of the robot. As a result, this will also affect the safety and efficiency of the toolpath. In addition to setting the initial position, the **Robot Control** tab has several tools that can be used to restrict both the robot axes and the tool axis to make the toolpath safe and efficient. The key functions are highligted below:



Play SImulation: Starts the simulation recording.

Attach Tool To Start: Positions the tool at the start point of the toolpath

Save SImulation: Save the recorded moves to replay them

Right clicking an axis gives you options to set priority.

Orientation vectors can be used to constrain the tool axis

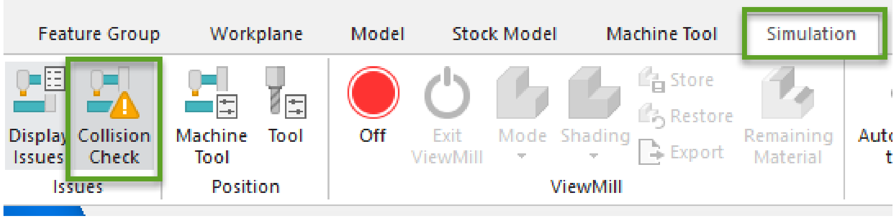
Enable graphics ON
Trace axis limits OFF

A1 AUTO 0.33 -185 / 185
B2 AUTO 31.9 -50 / 100
C3 25.64 -215 / 70
D -3.83 -360 / 360
E -87.54 -130 / 130
F6 0 -360 / 360

Orientation vector [X]

Vector I 1.0 J 0.0 K 0.0
Angle 0.0
Tolerance 0°
Automatic singularity avoidance □
Wrist singularity avoidance □

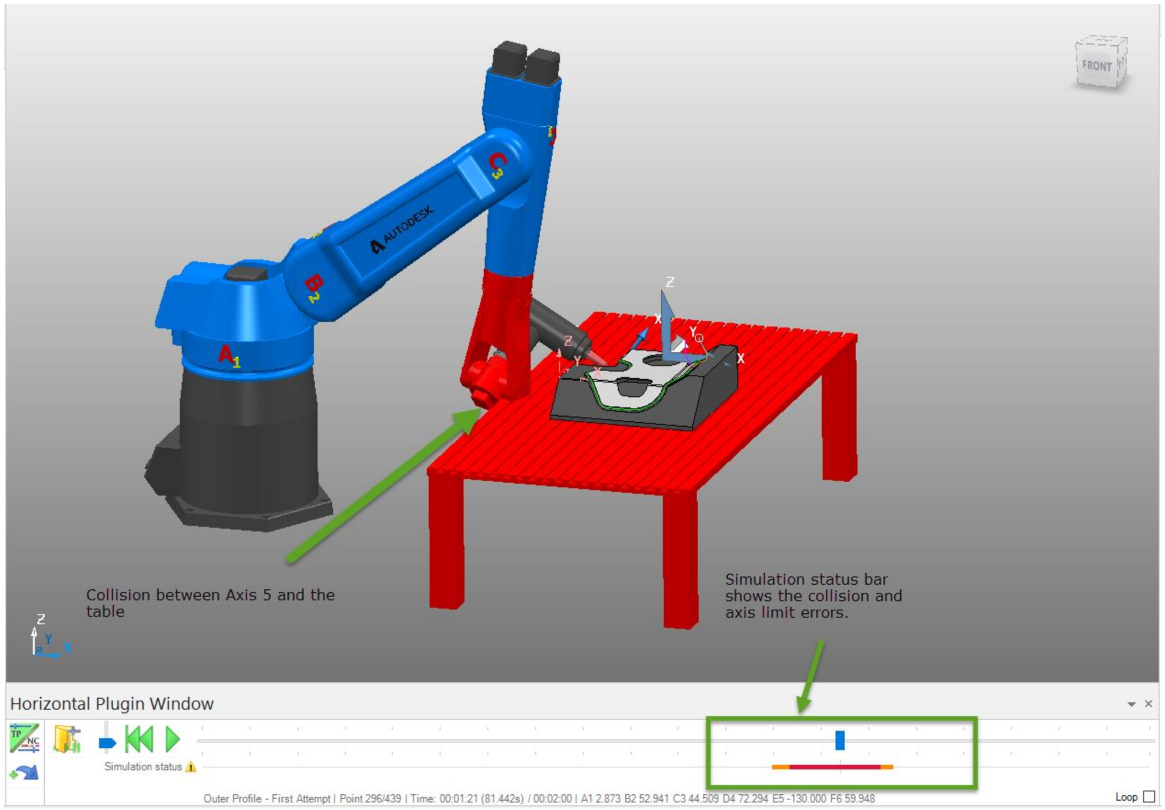
Before starting the simulation, turn collision checking on from PowerMill's Simulation tab to detect gouges and collisions.



Recording the simulation:

1. Activate the “Outer Profile” toolpath.
2. Click the **Attach Tool To Start** icon.
3. Click the **Play Simulation** icon.
    - The speed of the simulation can be adjusted with the regular PowerMill simulation speed controls.
4. Save the simulation and name it “Outer Profile Free”.

This simulation is free of any constraints as the tool and robot axes were not limited to any orientations or priorities respectively. However, some collisions and axis limits were detected as shown below:



Collision between Axis 5 and the table

Simulation status bar shows the collision and axis limit errors.

Outer Profile - First Attempt | Point 296/439 | Time: 00:01:21 (81.442s) / 00:02:00 | A1 2.873 B2 52.941 C3 44.509 D4 72.294 E5 -130.000 F6 59.948

Axis 5 collided with the table and it also reached its limit in the area highlighted in red on the simulation status bar. Why does the robot choose this orientation when there are other safer and more efficient ways?

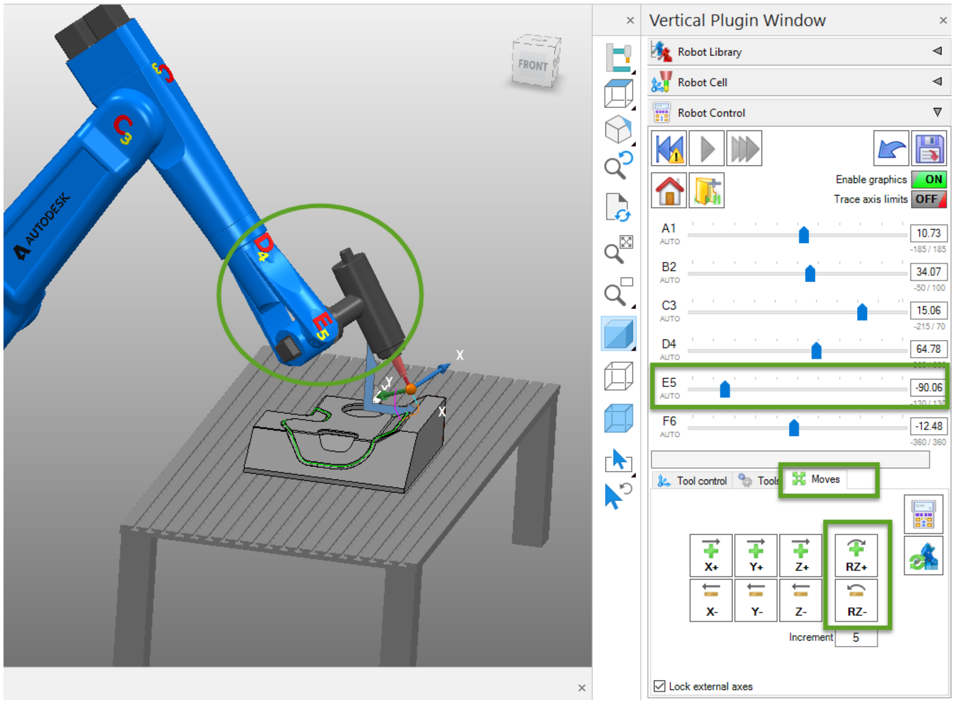
The inverse kinematics solver finds the “easiest” solution for a given toolpath point. This solution is not always the most optimal, therefore we must use various constraints to narrow down the solutions to safe and efficient ones. These constraints can be either on the robot axes or the tool axes.

Optimizing The Robot’s Motion For Safety and Efficiency

Avoiding collisions by adjusting initial position

In the previous instance, the robot tried to swing axis 5 in order to access the part but ended up overcorrecting in the process. We will try to set the initial position of the robot in such a way that the 5th axis is slightly rotated to avoid the overcorrection.

1. Attach the tool to the start of the toolpath.
2. Use the **Moves** area of the Robot Control tab to adjust the initial position of Axis 5 to something close to what’s shown below (approx. -90 degrees). Use the RZ+ and RZ- tools to jog the robot into this orientation:



Vertical Plugin Window

Robot Control

Enable graphics ON

Trace axis limits OFF

A1 AUTO 10.73 -185 / 185

B2 AUTO 34.07 -50 / 100

C3 AUTO 15.06 -215 / 70

D4 AUTO 64.78

E5 AUTO -90.06 -130 / 130

F6 AUTO -12.48 -360 / 360

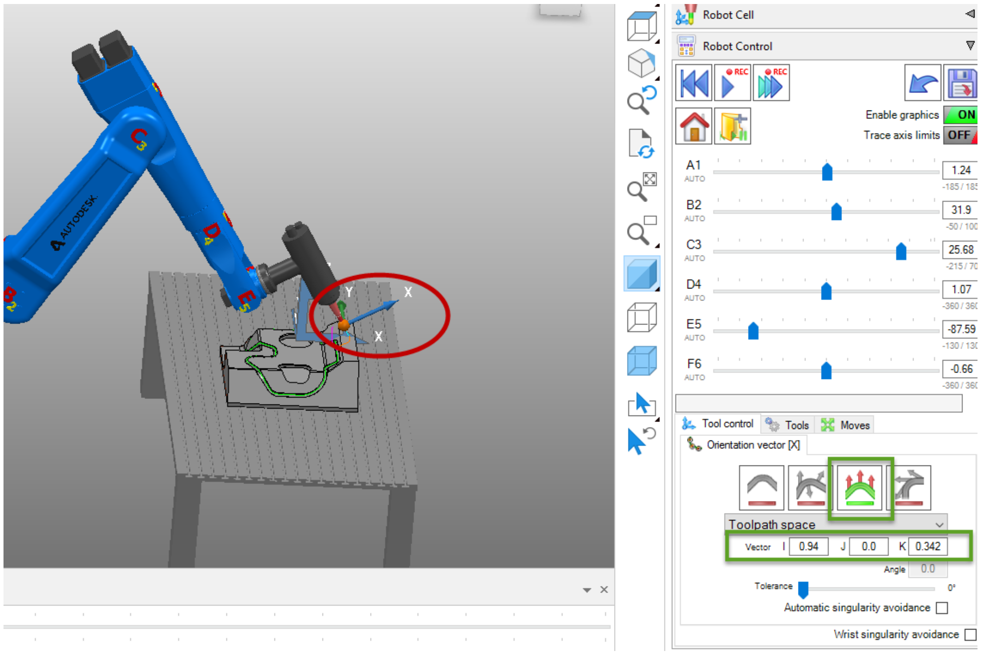
Increment 5

☑ Lock external axes

3. Attach the tool to start again.
4. Record the simulation and save it as “Outer Profile Initial Position”.

The toolpath is now free of collisions, but it is very inefficient. Notice how Axes 4 and 5 move drastically throughout the toolpath. They do this especially in the corners when the tool axis has to change direction. For example, In the section to the right(-Y) of the non-circular pocket, Axis 4 swings from over 100 degrees to less than -35 degrees in a few toolpath moves. The toolpath can be posted to the robot in this state, but it would not be ideal for the longevity of the robot’s motors and gears. A smoother transition in the corners can be achieved by locking the tool axis in a specific direction, which in turn restricts the robot axes to a narrower range. In general, the less the robot has to significantly re-adjust, the better.

Using vector constraints to lock the tool axis



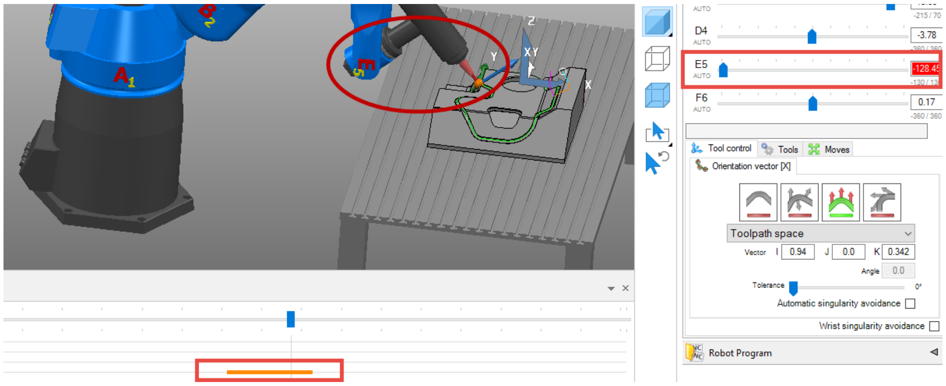
Enable graphics ON
Trace axis limits OFF

A1 AUTO	1.24	-185 / 185
B2 AUTO	31.9	-50 / 100
C3 AUTO	25.68	-215 / 70
D4 AUTO	1.07	-360 / 360
E5 AUTO	-87.59	-130 / 130
F6 AUTO	-0.66	-360 / 360

Vector I 0.94 J 0.0 K 0.342
Angle 0.0
Tolerance 0°
Automatic singularity avoidance □
Wrist singularity avoidance □

1. Return the robot to its home position by clicking the **Go To Home** icon in the Robot Control tab.
2. Attach the tool to start such that the X axis of the tool is pointing in the direction shown above.
3. Use the **Tool Control** panel to lock the X axis of the tool in the current direction by clicking the **Vector** tool control icon (3rd icon from the right) highlighted above.
4. Re-attach the tool to start.
5. Record and save the simulation as “Outer Profile Vector Control.”

This time the toolpath was smoother in the corner transitions. However, it reached axis limits in the same area as the first simulation with free constraints:



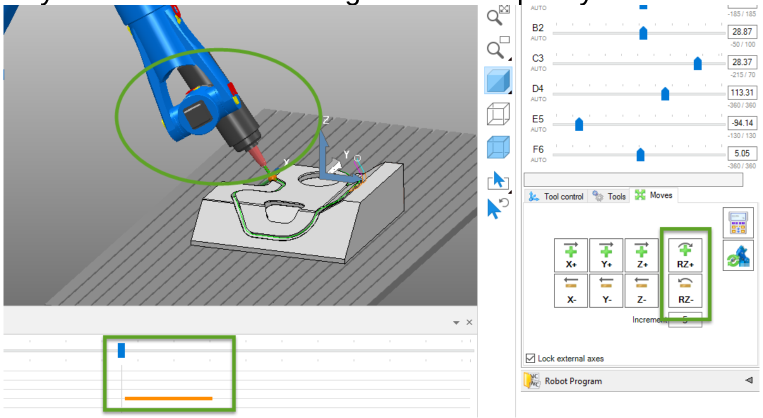
AUTO		-215 / 70
D4 AUTO	-3.78	-360 / 360
E5 AUTO	-128.45	-130 / 130
F6 AUTO	0.17	-360 / 360

Vector I 0.94 J 0.0 K 0.342
Angle 0.0
Tolerance 0°
Automatic singularity avoidance □
Wrist singularity avoidance □

The next step in controlling the tool axis is applying orientation vectors within a region of the toolpath. This allows us to isolate the area where axis limits are being reached from the constraint applied to the rest of the toolpath.

Constraining the toolpath locally with orientation vectors

1. Move the robot to its home position.
2. In the **Tools** panel, click the **Edit Orientation Vectors** icon.
3. In the graphics window, drag select the entire toolpath.
4. Click **Apply Orientation Vectors**. This will constrain the tool axis globally in the same manner as the previous method.
5. Attach the tool to start.
6. Record and save the simulation (the same axis limit error as before will be reported but ignore it).
7. Use the simulation toolbar at the bottom to move the play-head to the point just before when axis limits are reached.
8. Re-position the robot to the pose shown below by using the **Moves** panel. This will be the orientation used by the robot within the region we will specify.

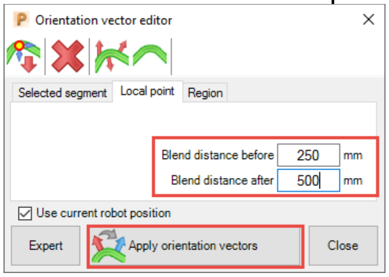


-185 / 185
B2 AUTO 28.87 -50 / 100
C3 AUTO 28.37 -215 / 70
D4 AUTO 113.31 -360 / 360
E5 AUTO -94.14 -130 / 130
F6 AUTO 5.05 -360 / 360

Increment 5

☑ Lock external axes

9. In the **Tools** panel, click the **Edit Orientation Vectors** icon:
10. In the dialog box, choose **Local Point.**
11. Set the **Blend Distance Before** to 250mm and the **Blend Distance After** to 500mm. This will allow the robot to smoothly transition into this pose 250mm prior to reaching this point and 500mm after so that the transition is not abrupt.



Orientation vector editor

Selected segment | Local point | Region

Blend distance before 250 mm
Blend distance after 500 mm

☑ Use current robot position

Expert | Apply orientation vectors | Close

12. Click **Apply Orientation Vectors** and close the dialog box.
13. Re-attach the tool to start.
14. Record the simulation and save it as “Outer Profile Local Control”.

After having applied tool axis constraints and locally applying orientation vectors, the toolpath is now both free of collisions and more efficient. The final step is posting the code.

Posting Code

Code for robots is very different from machine tools in that it doesn't output toolpath points, but joint angles for the robot. It is therefore not universal and varies from one manufacturer to the other.

Loading the Post-Processor

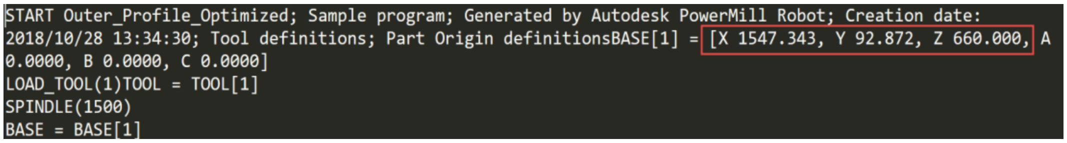
Each of the default robots in the library has a post-processor. However, if you have a different robot, the post-processor has to be customized for that robot. The default posts can be found in this location: `C:\Program Files\Autodesk\Autodesk PowerMill Robot Plugin 2019\Library\PostProcessors`. Follow these steps to load the post-processor:

1. Expand the **Robot Cell** tab and click **Robot Configuration.**
2. In the dialog box, click on **Postprocessor.**
3. Choose the post from the ones available in the drop-down.

Writing the Robot NC Program

The NC program is generated from the **Robot Program** tab as follows:

1. In the **Program Name** field, enter "Outer Profile Optimized." The program name has to be different from that of the toolpath being written.
2. Leave the output workplane blank so that the workplane we measured to position the part is used. The code will be posted with respect to the chosen workplane. If a rotary external axis is involved, the workplane at the center of the axis is usually used. PowerMill Robot will output this location in the NC code so that it is clear for the robot where the values are being measured from:



```
START Outer_Profile_Optimized; Sample program; Generated by Autodesk PowerMill Robot; Creation date:
2018/10/28 13:34:30; Tool definitions; Part Origin definitionsBASE[1] = [X 1547.343, Y 92.872, Z 660.000, A
0.0000, B 0.0000, C 0.0000]
LOAD_TOOL(1)TOOL = TOOL[1]
SPINDLE(1500)
BASE = BASE[1]
```

3. Right-click anywhere in the white space in the Robot Program tab and select **Add a Toolpath.**
4. Under **All Simulations** choose "Outer Profile Local Control".
5. Click **Write Robot NC Program.**

The file location where the program is saved will automatically be presented and two files will be generated; one with the output workplane information and one with the actual code.

The program can then be loaded onto the robot. This example had only one toolpath, but there are situations where there are multiple toolpaths and safe transitions between the end and start of each toolpath are required.

Transitioning Safely Between Toolpaths

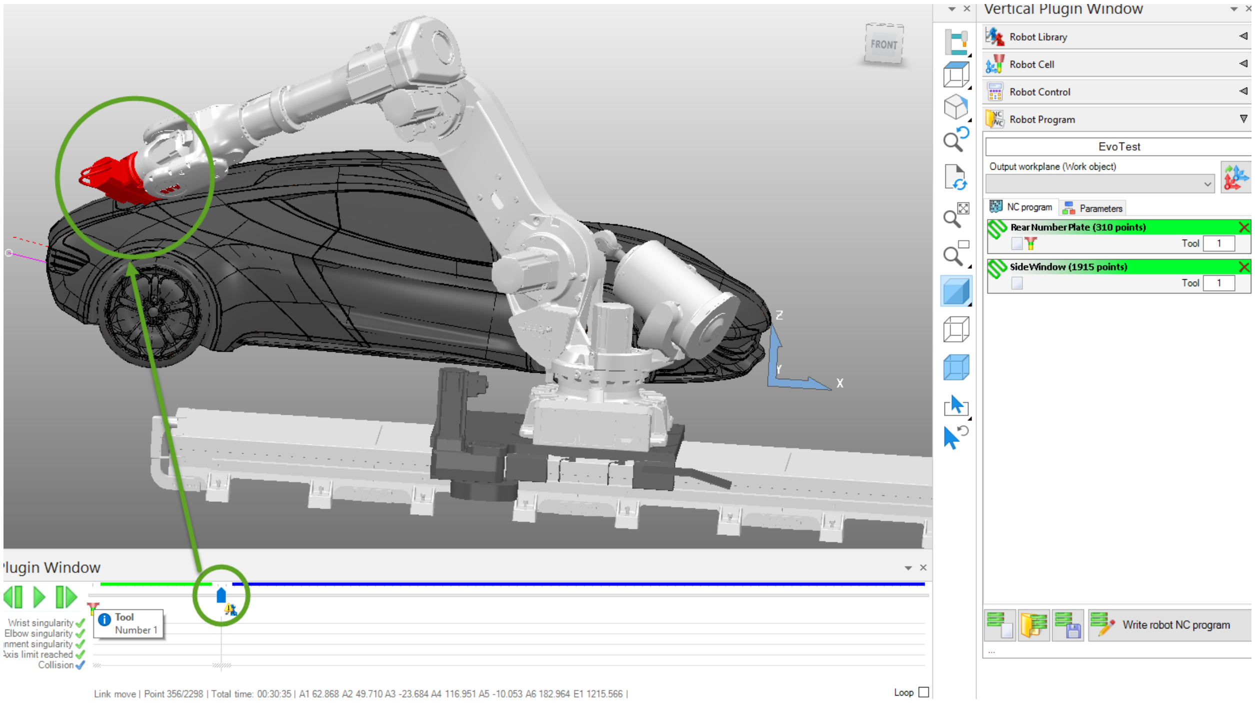
- Close the current project and open the **AutodeskEvoCar** `from C:\Program Files\Autodesk\Autodesk PowerMill Robot Plugin 2019\Library\Projects\AutodeskEvoCar`. This project is a model of a car and therefore bigger than the normal workspace of a single robot. It will require a robot on a linear track to expand this workspace so that all toolpaths can be machined in the same cell in a single setup.

- Load the **ABB IRB6640 + Linear Track** robot from the library. The robot will position itself on the part since that's where the world origin is. To position the part, go to the Robot Cell tab and choose the "Center Right (Rail)" workplane.

- Simulate the "RearNumberPlate" and "SideWindow" toolpaths from the Robot Control tab in that order and save them.

- Create a robot program with both toolpaths then simulate both of them in sequence.

During the simulation, a collision will occur as the robot moves from the rear of the car to the side window. A safe transition between the toolpaths must be added to ensure the collision doesn't occur.



EvoTest

Output workplane (Work object)

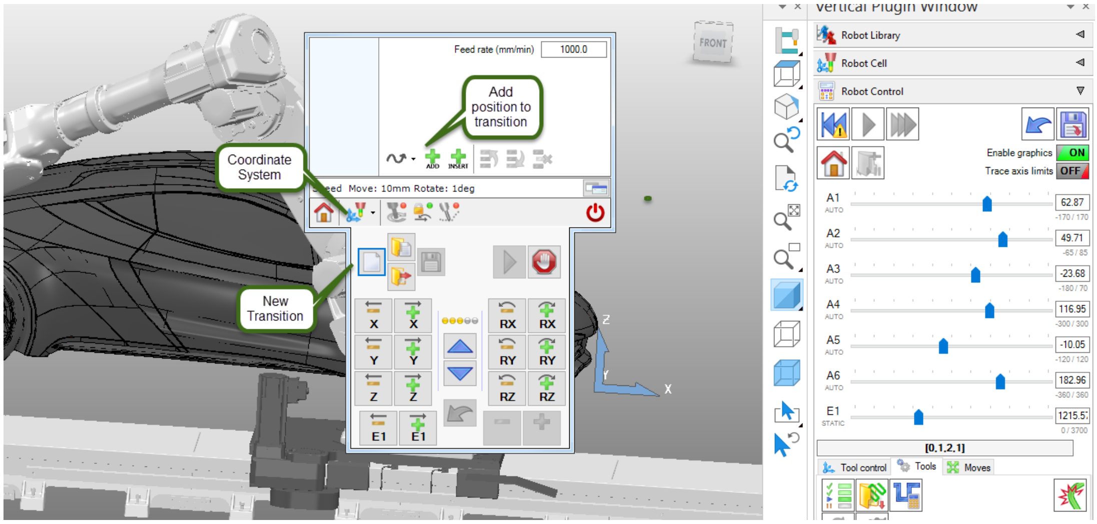
RearNumberPlate (310 points) Tool 1

SideWindow (1915 points) Tool 1

Link move | Point 356/2298 | Total time: 00:30:35 | A1 62.868 A2 49.710 A3 -23.684 A4 116.951 A5 -10.053 A6 182.964 E1 1215.566 |

How to add a transition:

1. Right-click anywhere in the Robot Program tab and select **Add A Transition**
2. Select **Virtual Teach Pendant**. This will display the teach pendant so that the robot's transition motion can be recorded "manually" by jogging the axes.
3. In the virtual teach pendant, click the **New** icon to begin the recording. The axis control tools can be used to move each robot axis or the tool axis in discrete movements and the resulting position added to the transition using the **Add** icon**.**



4. Move the simulation slider to the end of the RearNumberPlate toolpath and add the position to the recording. This is a safe location as the robot is off the part.
5. Switch the coordinate system to the robot axis and jog axis A3 such that the arm is above the height of the car and add this position to the transition.
6. Now that the robot is clear of the part, use the Robot Control tab to move it to the home position and add the position to the transition.
7. Switch the coordinate system to the external axis (rail) and move the robot such that the base is past the front of the car. Add this position to the transition as well.
8. Finally, activate the SideWindow toolpath and attach the tool to start. Add this final point to the transition.
9. Click the save icon in the teach pendant and name the transition "T1".
10. Right-click anywhere in the Robot Program tab and select **Add A Transition.**
11. Choose the T1 transition you just saved and it will be added to the bottom of the program.
12. Drag the transition and place it in-between the toolpaths. Re-run the simulation and it will now be free of the collision.