Exercise 3, Robot Studio, using an arc welding gun

For Robot Studio version 6.06

Gunnar Bolmsjö, [Rev. Sept 2018 ARo/ERo]

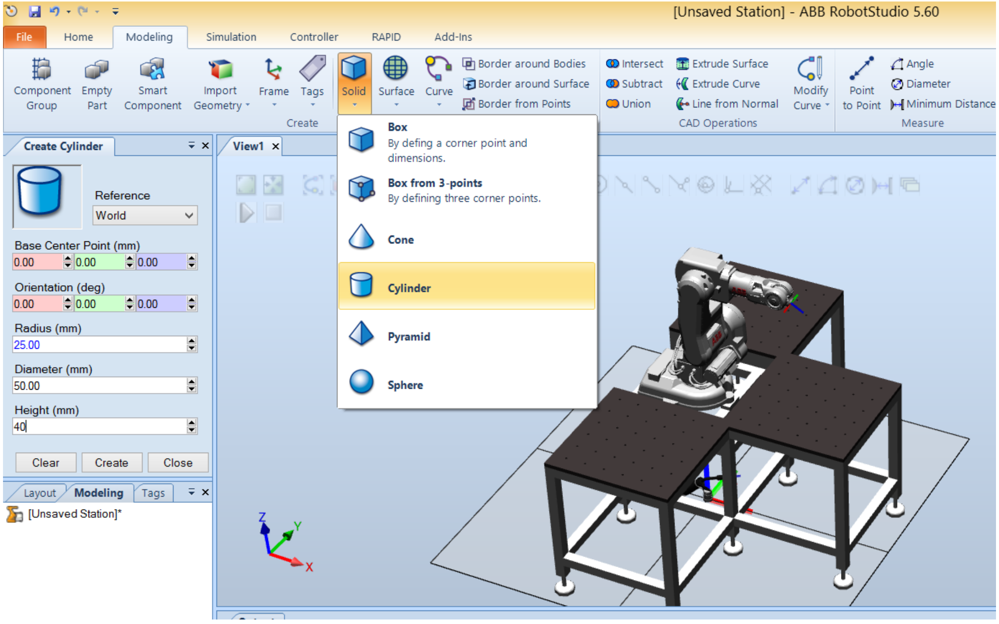
The aim of this exercise is to explore some issues related to simulation and programming of a task which requires precise control of both the target positions and the path between these. In addition, we will introduce collision checks between objects, as this is a common problem in this type of applications. The work scenario is a robot that welds workpieces (plates) arranged as a normal structure for any heavy construction found in steel bridges or shipyards. The robot station is the same as in exercise 1 and 2 (ABB IRB140). The instruction of this exercise focuses on the new issues introduced but all steps to conclude the exercise is mentioned.

1. Setting up the robot station

Start RobotStudio 6.05 (64-bit) and start a new station using a template system. Select the first robot (IRB140_6kg_0.81m_typeC) and click on OK. After some time, the system will come up with a robot. Import some geometries form the library (Import Library), select the library folder "Lab3 weld" (S:\Courses\design\MMKF15 Robotteknik\RobotStudio_exercises_2016\Lab3 Weld). Import rob-table, welding-gun and welding-case1. Set the task frame to z value 670 mm. Instead of attach the welding gun directly to the robot, as before, we will have a tool changer in between the robot and the welding gun. A simple way we solve this is to attach the welding gun to the robot, but to offset it the 40mm from the flange, which corresponds to height of the tool changer, and then add a graphical representation for the tool changer, which we also attach to the robot flange:

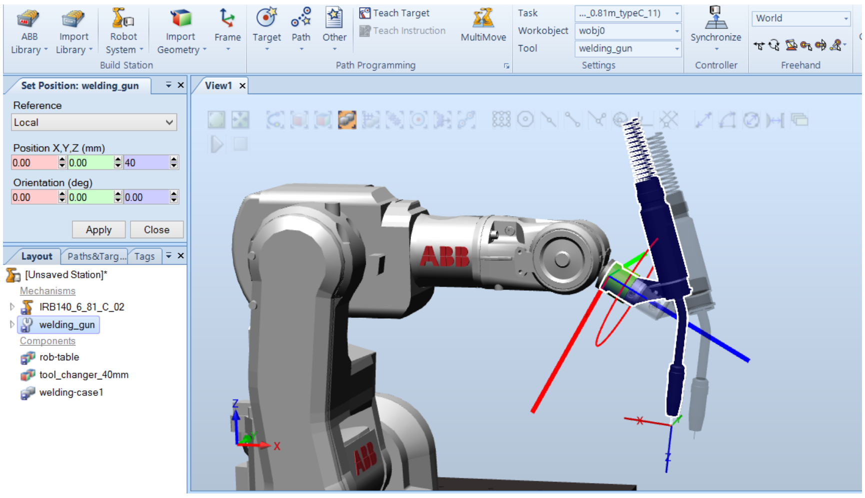
To define and attach a tool changer:

Choose the Modeling-tab and choose Solid -> Cylinder. Enter Diameter = 50mm and Height=40mm and press "Create". Change the name to "tool_changer_40mm".



Picture 1: Where to find the Solid -> Cylinder

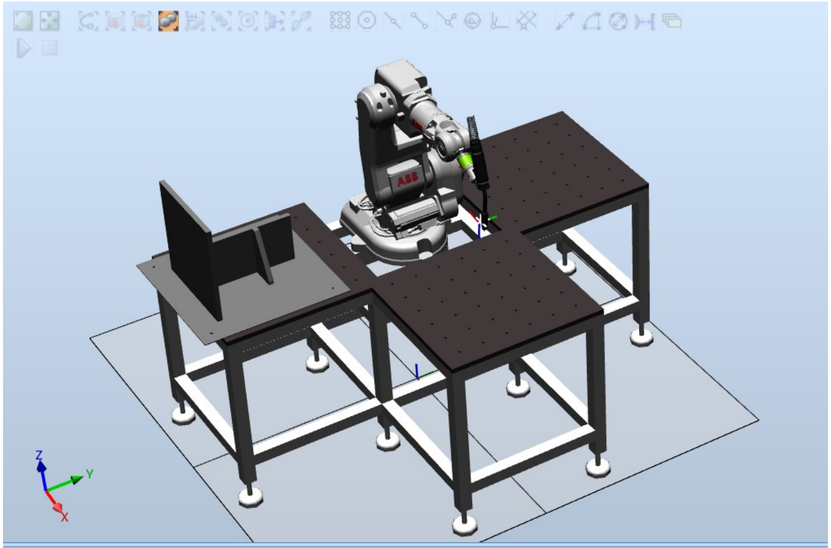
In the Home -> Layout tab, *right-click* on the tool changer and change its color for better visibility ("Modify" -> "Set color…") and then *right-click* again to attach it to the robot. Then *right-click* on the welding gun and attach it to the <u>robot</u> (!). To add the 40 mm a simple way is to refer to a local frame. *Right-click* on the welding gun and "Set position". By choosing "Reference = Local" and z-position = 40mm we get the correct welding gun position.



Picture 2: The correct position for the welding gun is the ghostly shape to the right

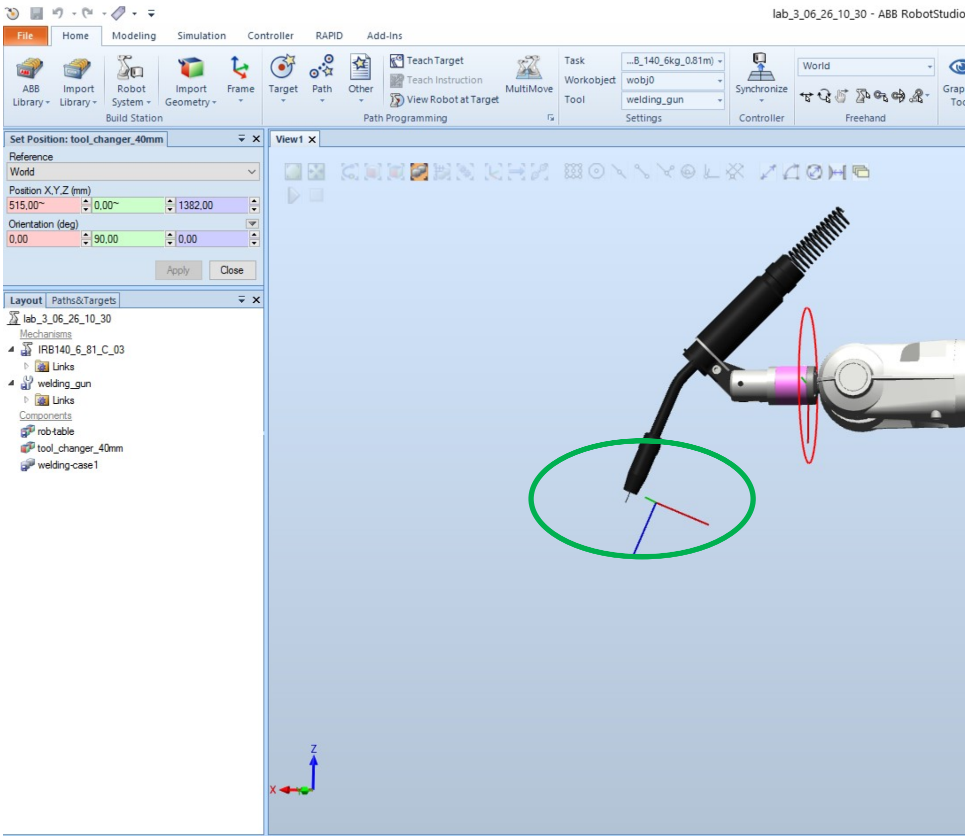
Save the station.

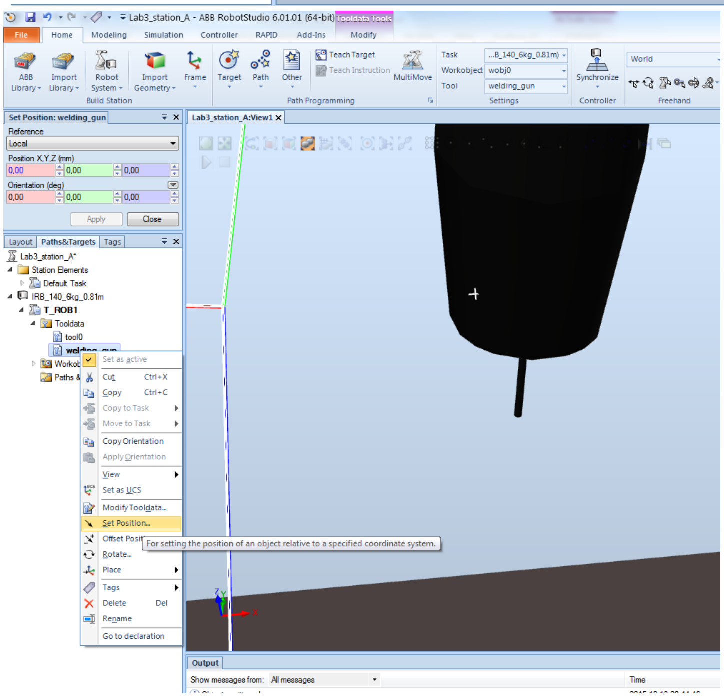
The station should now look like the following picture:



Picture 3: How the station should look at this point

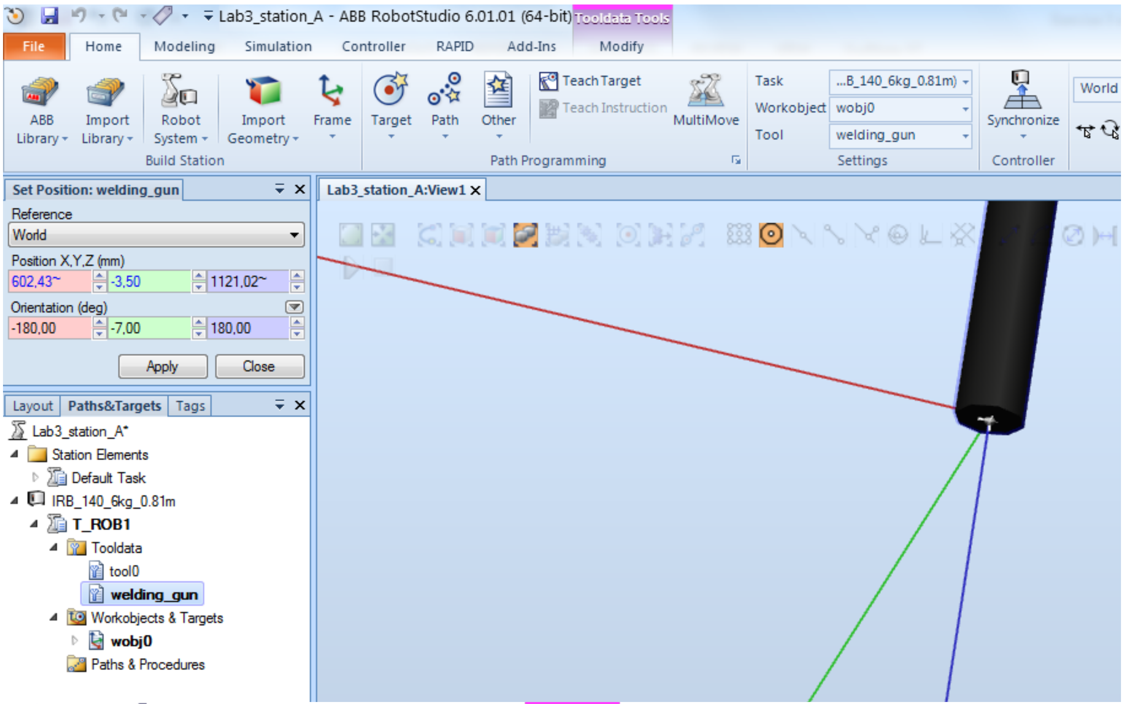
However, looking closer, we see that at the tool tip, we can see that there are two frames according to pictures below.

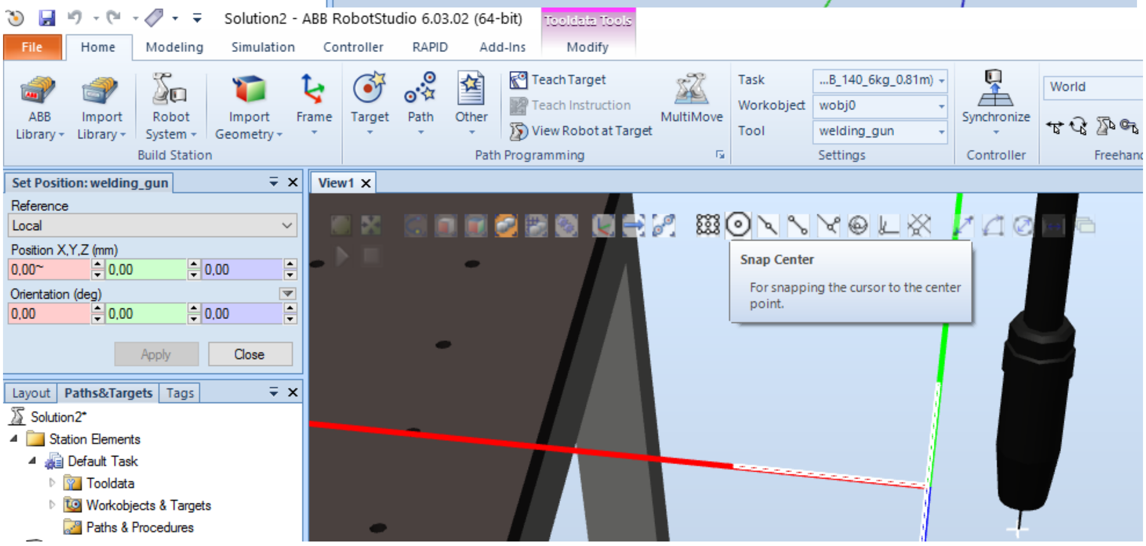




Right-click on the welding gun in the “Paths&Target”-tab and choose “Set position”.

Click in one of the coordinate edit fields (red, green or blue), choose “Snap center”, zoom in on the tool tip of the welding gun and click to place the tool frame.

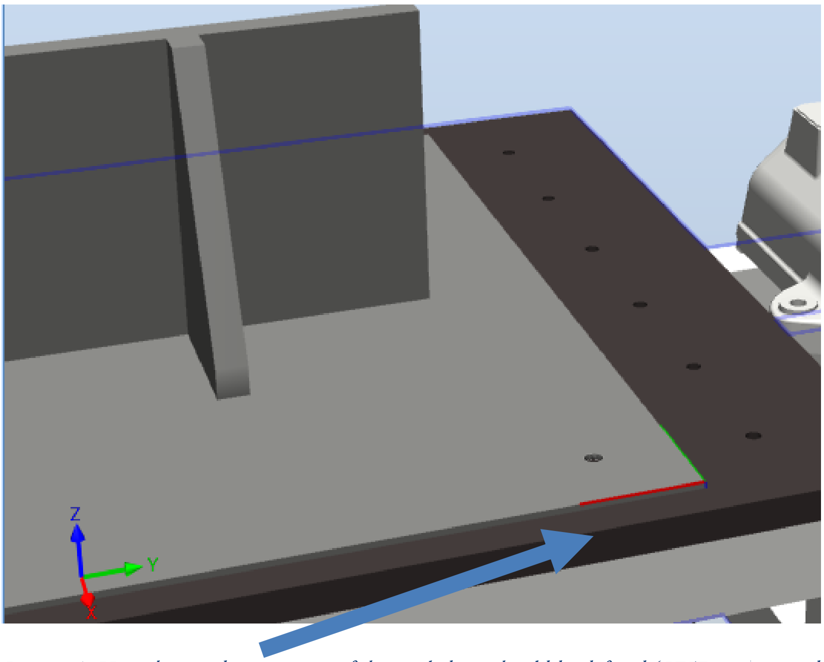




2. Create targets and paths for the welds

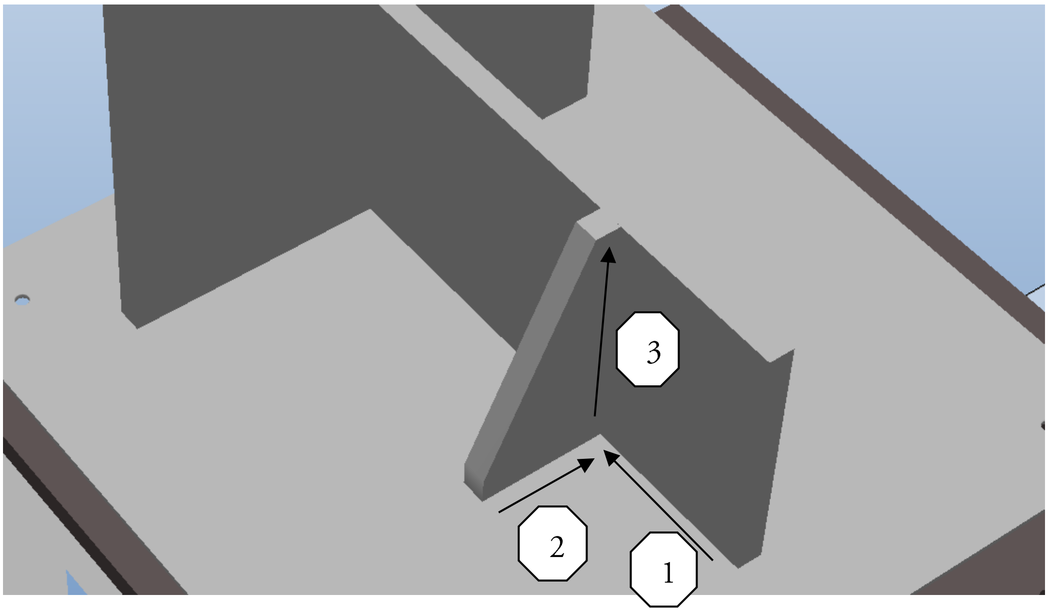
Create a workobject for the welding part. Set User Frame values at (300, -400, 124) and Rotation values at (-180, 0, -90). Keep the values for the workobject at zero and the workobject frame should now be at the corner of the welding object base plate according to Picture 4 below (z-axis pointing downwards).

See Appendix A for how you can teach the corresponding frames

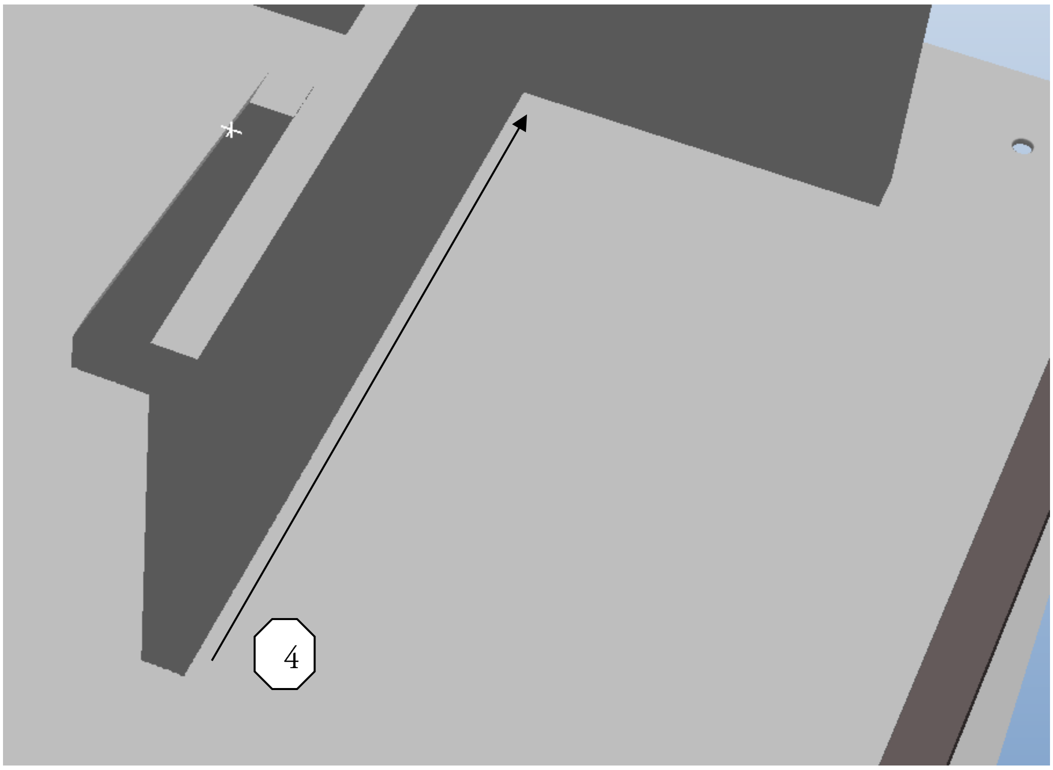


Picture 4: How the coordinate system of the workobject should be defined (XYZ : red-green-blue (rgb)).

The welds to be made are shown in the following pictures:



Picture 5: How the three first welds look



Picture 6: How the 4th and final weld looks (backside).

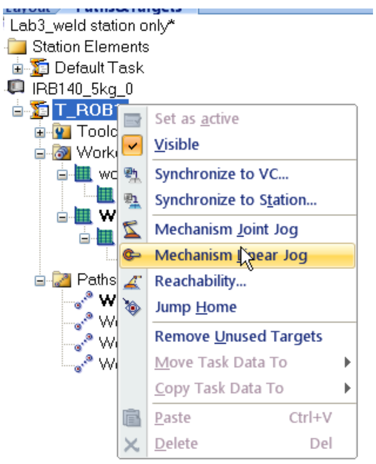
The general workflow for each weld is to produce an approach movement close to the weld, about 30 mm, move straight to the start point of the weld with a "fine" MoveL (perpendicular, 45 degrees angle between the plates except for the start in the corner which have to 45 degrees to all plates). Then, at the weld start point start the weld motion to the end of the weld via intermediate weld targets if necessary (in such case with a zero value greater than "fine"). Use velocity v50 for the weld movements.

Note: Typical welding speed for a case like this is usually in the order of 4-8 mm/s, and in several layers and performed in a weaving pattern. Thinner plates are produced with a higher welding speed, in the order of up to 25 mm/s, but this is highly dependent on many interrelated parameters including plate material, thickness, joint design, tolerances and so on.

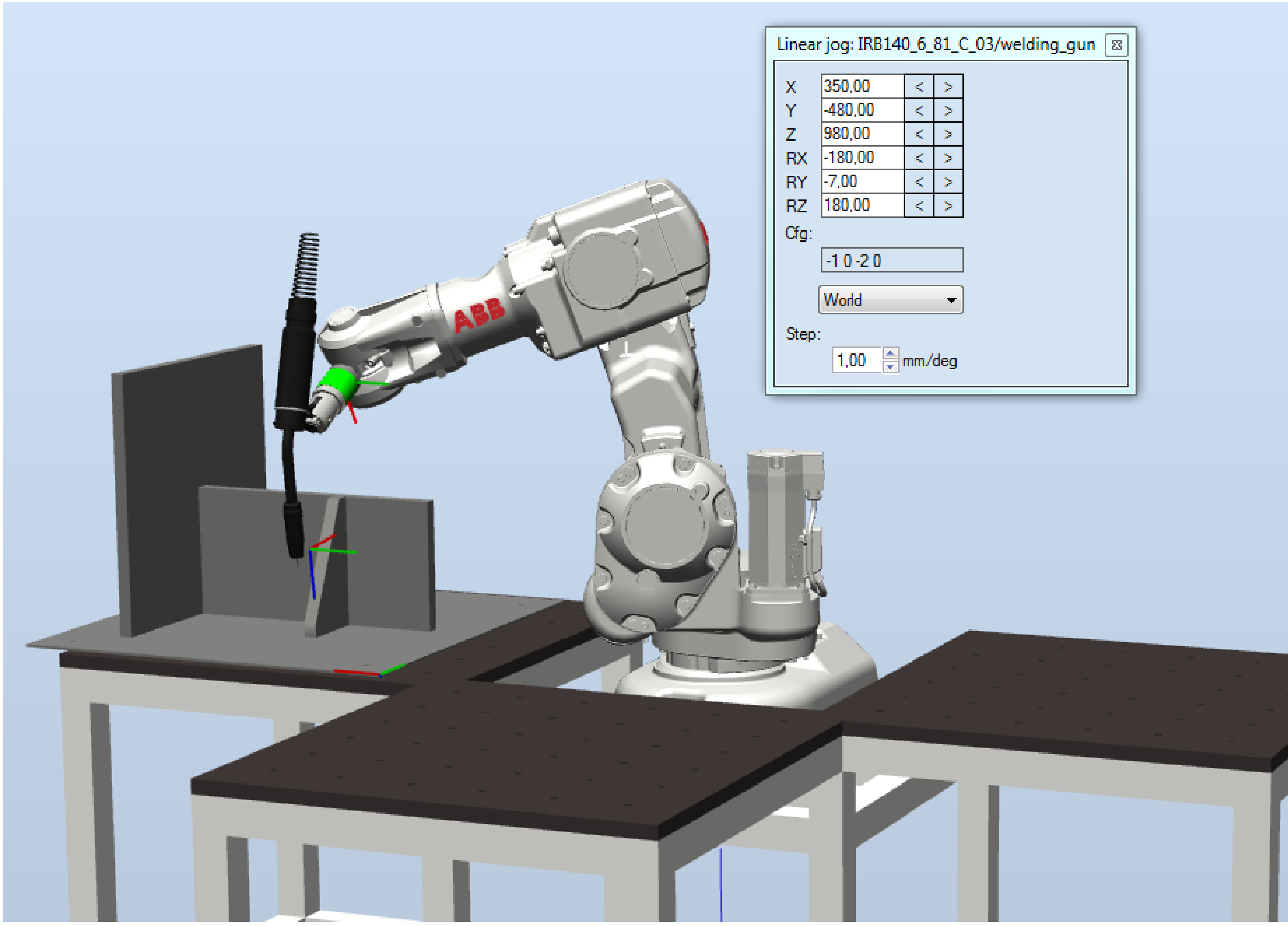
Create four weld paths and name them for example as Weld1, …, Weld4.

Create a target at the "Home" position.

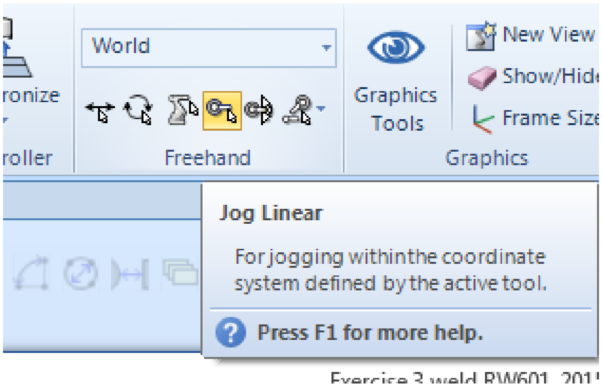
Jog the robot to a rough start position for the first weld without changing the tool orientation. Jog linear by right-click on the T_ROB1 (in the Layout-tab) and select Linear Jog:

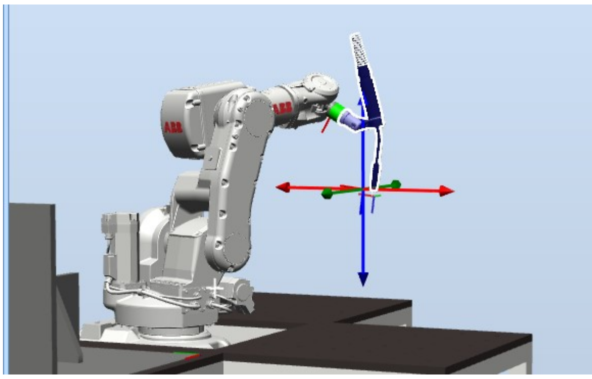


Picture 7: How to select Linear Jog



There is also a possibility to jog the robot in the graphics window by choosing "Jog Linear-button" in the Home-tab, click the tool-tip of robot in the Graphics window and then drag along any of the red/green/blue arrows, see figure below. (You can also re-orient or jog joint-wise in a similar manner).

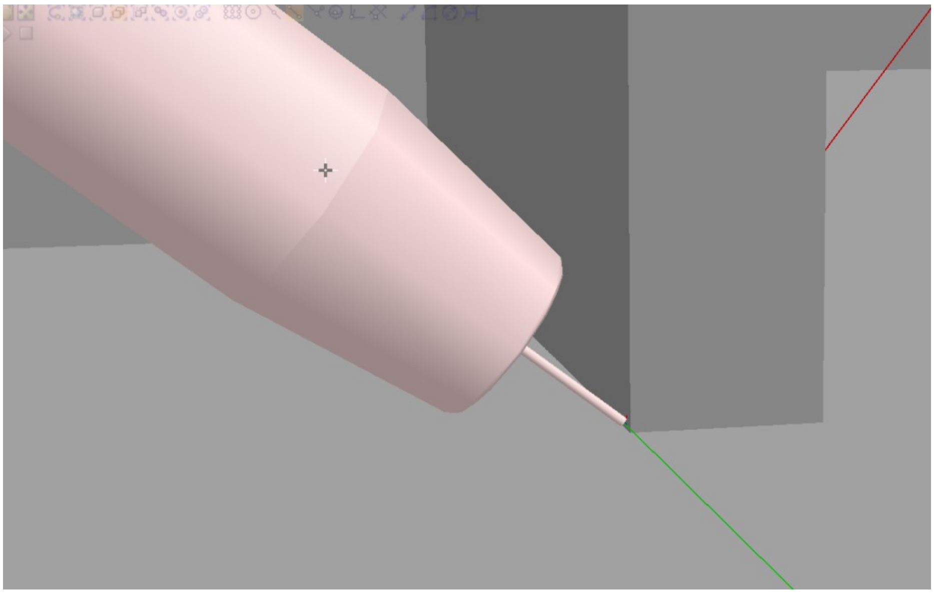




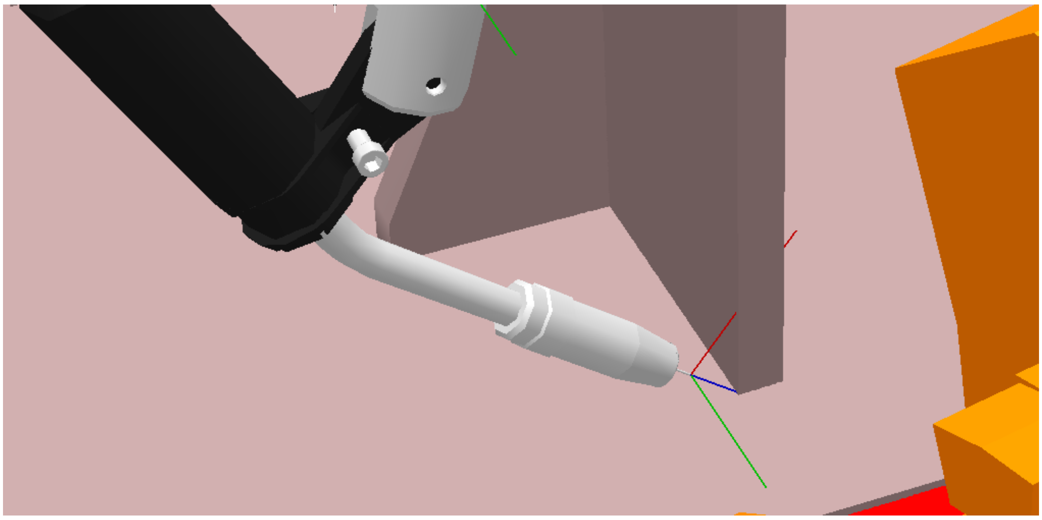
Teach a new target.

Now, select the welding gun as the jogging reference. Rotate around the Y axis to get a tool orientation for the weld operation. Tip: to get a 45 degrees orientation, use the fixed step size to increase the angle to get an exact orientation.

Jog the robot to the start of the first weld. To get to the start you must change the Step to 1 mm.



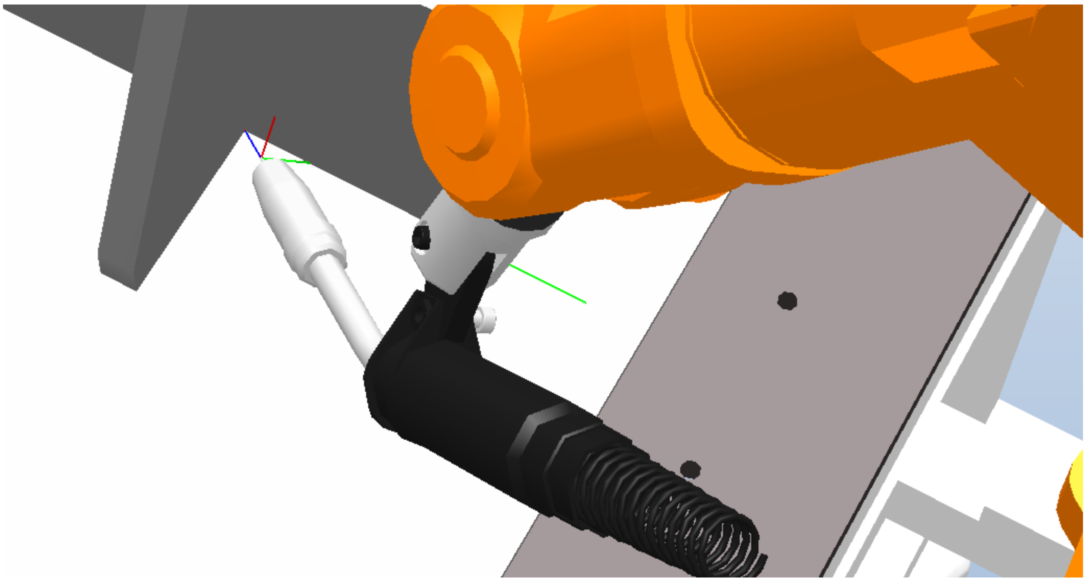
Jog out from the weld start 30 mm (e.g., change the Step to 30 mm and click once). Teach a target.



Jog in to the weld start again and teach the weld start target.

?

Jog to the corner and teach the end of the weld, move out form the corner about 30 mm and teach the weld end target. Continue with a target some 30 mm out from the weld end position. Put zone when moving back to home position to fine to avoid a collisions and warning messages. If your robot does collide, create another position even further away from the working object before moving your robot back to the Home -position.



Check first weld

Save the station.

This is a good time to check the station with the first weld and check that it works. Put the targets in the first weld path; finish with the first target (the home position). Change the move instruction so the motion to the first, second and last target is a MoveJ. Set velocities and zone as described above in this document. Use "Auto configuration", and then add the path in the simulation setup and run the simulation. When all works, save the station.

Repeat the workflow with weld 2, 3 and 4.

Note 1: Use World as reference when jogging in the X / Y plane. When jogging in the direction of the welding gun, welding gun is the preferred reference. This is also the case when making orientation changes.

Weld 4 workflow

Note 2: To get a proper orientation for weld 4 you might find it better to switch to joint jog mode and then back to linear jog again.

Save the station.

Creating the targets and the paths is a bit tricky in this case, which also is quite typical for a real situation, but in this case, we are only dealing with a few targets.

When you are finished with the paths, check these with "Auto Configuration" and the robot position at each target for any collisions.

Synchronize to RAPID by adding one path at a time. Make sure that the first target is a MoveJ, otherwise a synchronization error is likely to occur when trying to Synchronize to Station.

Note: If stuck in the sync process: Make a "P-start", Delete Module1 containing your program, create a "New Module" and name it Module 1. Everything should be fresh now and if still not working, make all Move instructions to MoveJ. Include only "Save Paths" and start from there.

Finally, add the "Home" position to the last weld path at the end and make it a MoveJ instruction.

When the "Synchronization to RAPID" is working, add a main procedure (either in a separate module or as a procedure in Module1) and run the simulation.

You will notice some strange motions and suspected collisions between weld paths. We will now make some fine-tuning to check and manage collisions.

3. Check the program for collisions and configuration errors

We will now check the program for collisions. In general, there are two types of collisions to take care of:

1. At weld start / end, typically corner targets when the welding gun is re-oriented. Solution: change the Z-value in the welding gun direction and move is a bit away (a few mm).
2. Between weld paths. Include so called via targets. Depending on the motion you need to make, one or more targets may be necessary. You may also try to switch between MoveL and MoveJ.

If you get a configuration error you will also need to modify instruction to MoveJ and include a so called "via target" which is an intermediate point along the path you want to follow.

To check for collisions, the system can assist you to detect those during simulation of a motion.

In this exercise we will use two collision sets, one to check for collision between the robot and the welding gun, and one to check for collision between the (robot + welding gun) and the weld object.

A collision set is created as follows: In the **Simulation** tab, select **Create Collision Set**. A "CollisionSet_1" appears in the Layout tree. Expand the collision set and drag the Links Base, Link1, …, Link5 to "ObjectsA" and the welding gun to "ObjectsB".

Note 1: Link6 is not included because that would create a collision all the time as that link is connected to the welding gun.

Note 2: It does not matter what object is dragged to ObjectsA or ObjectsB.

Note 3: To change a target, you can do as follows: Move to the target you want to change ("Jump to Target"). Jog the robot to the new location. Teach a new target. Right-click on the target you want to change and select "Modify Target" → "Place" → "Frame". In the box "Select Frame" click on the new modified target you just created, and "Apply". You may then delete the last made target.

Note 4: Collision sets are computer demanding. You may notice that the simulation runs slower. In this case, the station is extremely simple. In a realistic scenario, great care must be taken when working with collision sets and limit the collision study as necessary. Collision sets can be made active / not active by right-click on the respective set.

Create one more collision set and drag both the robot and the welding gun to ObjectsA and the weld-case to ObjectsB.

Run the simulation again and you will most likely notice a number of collisions. Collisions will be highlighted in red and recorded in the Output log.

Go through the collisions one by one and fix the problems as indicated above.

4. Reflection

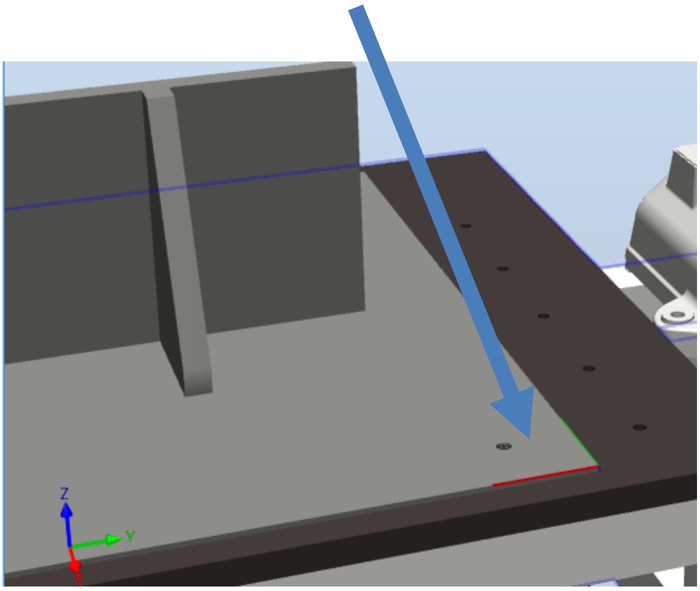
Think about the following issues:

1. Time to make a full arc welding program (or for any similar work process). A real workpiece may include between 20-100 welds, most of them at difficult positions
2. Real welding include a lot of process related considerations which put great demand on the programmer
3. Many targets or weld paths generate configurations close to joint limits, collisions or singularities. This is a common case, especially for processes like arc welding that runs a process through precisely defined paths
4. Read the procedure for defining workobject frames in Appendix A and make sure you understand the mathematics behind it.
5. How can you verify that the tool center point of the real welding gun is as defined in the program without doing a full tool calibration?

In practical situations, it is convenient to add some simple routines and tests to assure that workobjects and tool definitions seem to be correct (e.g., make reorientation around tooltip and verify it stays at same position; easiest to see if close to stationary reference).

Appendix A

How to define/calibrate a work object coordinate system in the real robot cell:



Description taken from

http://developercenter.robotstudio.com/BlobProxy/manuals/IRC5FlexPendantOpManual/doc105.html

Defining the work object coordinate system <-

-

Overview

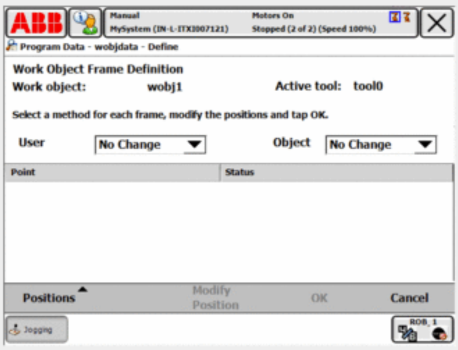
Defining a work object means that the robot is used to point out the location of it. This is done by defining three positions, two on the x-axis and one on the y-axis.

When defining a work object you can use either the user frame or the object frame or both. The user select frame and the object frame usually coincides. If not, the object frame is displaced from the user frame.

How to select method

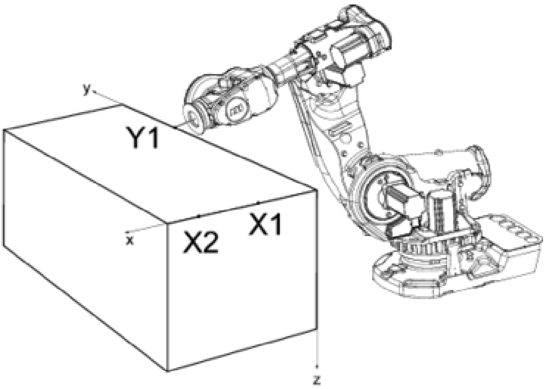
This procedure describes how to select method for defining either user frame or object frame or both. Note that this only works for a user created work object, not the default work object, wobj0. Defining work object can also be done from the **Program Data** window.

	Action
1	On the **ABB** menu, tap **Jogging**
2	Tap **Work object** to display the list of available work objects.
3	Tap the work object you want to define, then tap **Edit**.
4	In the menu, tap **Define.....**
5	Select method from the **User method** and/or the **Object method** menu. See How to define the user frame and How to define the object frame



How to define the user frame

This section details how to define the user frame.

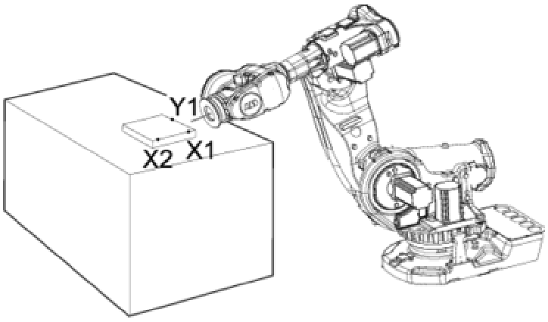


The x axis will go through points X1-X2, and the y axis through Y1.

	Action	Info
1	In the **User method** pop up menu, tap **3 points**.	
2	Press the enabling device and jog the robot to the first (X1, X2 or Y1) point you want to define.	Large distance between X1 and X2 is preferable for a more precise definition.
3	Select the point in the list.	
4	Tap **Modify Position** to define the point.	
5	Repeat steps 2 to 4 for the remaining points.	

How to define the object frame

This section describes how to define the object frame if you want to displace it from the user frame.



The x axis will go through points X1-X2, and the y axis through Y1.

	Action
1	In the **Object method** pop up menu, tap **3 points**.
2	See steps 2 to 4 in the description of How to define the user frame.