Welding Task in DELMIA V5-6

Optimal Kinematic Design of Robots Lab 3

Guillaume JEANNEAU, Debaleena MISRA

École Centrale de Nantes

November 18, 2017

1 Objective

This practical work is aimed at design and operation of robots in the DELMIA V5 software environment. A robot is kinematically modeled to perform an arc welding task. By course of this, we get familiar with handling the interaction of robot with the environment through process building, task definitions, resolving model constraints and workspace limitations, and eventually performing an efficient execution of spot welding. The diagram below demonstrates this objective.

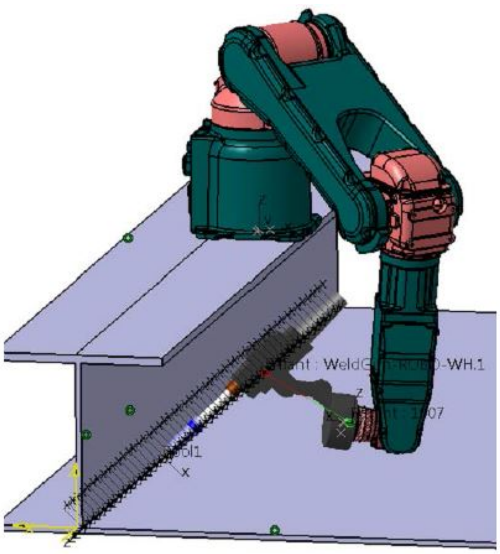

Figure 1: Example of Welding process

2 Methodology

The robot model used for this task is **Motoman MH5LS**. The device build-up is started using the .CATPart files for the links and base of the robot.

2.1 Definition of the frame of interest

The welding task involves attaching a tool rigidly to the robot end-effector. The direct and inverse geometric model of the robot must apply to the tool as well. To ensure that, we use the **frame of interest** feature in the DELMIA framework.

We define a tool frame attached to the end-effector solid **T_AXIS**. The same way, a base frame should be attached to the robot base **BASE_AXIS**. This has been done by defining these two .CATPart objects as .CATProduct objects. The procedure is described below.

A **frame of interest** subtree is added to the product object. This subtree will include the frame of the tool and of the base. Next, the construction of the frame is done using 3 points. Those three points have been defined previously in each of the two CAD models. One point defines the center, the second point defines the first axis direction and the third defines the plane and the orientation of the z axis. This third axis is directed outward to the exterior for the tool frame. It is defined as going inwards to the robot for the base. Figure 2 shows the defined frames of interest once the robot has been assembled.

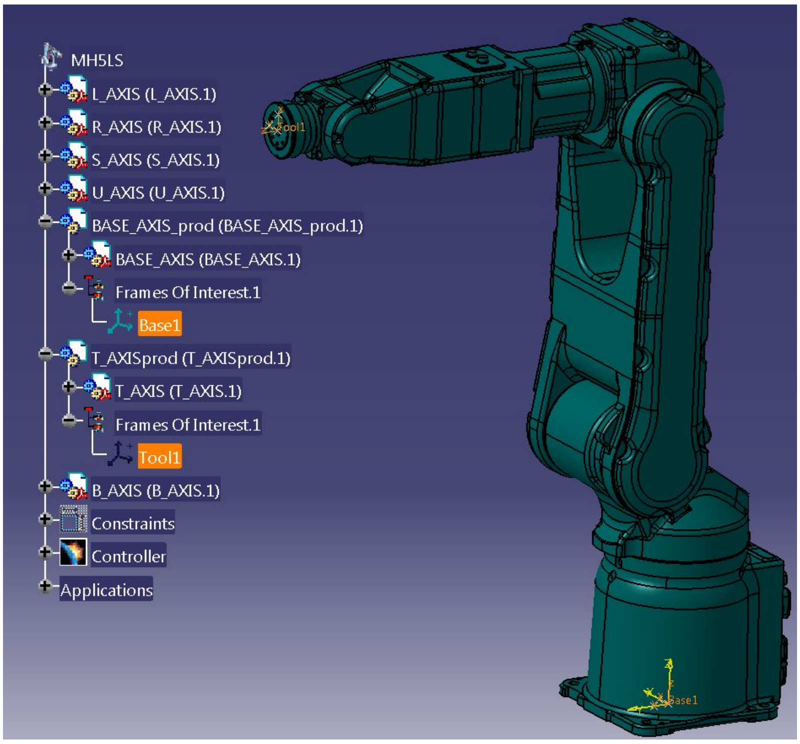

Figure 2: Defined frames of interest for the robot at home position

2.2 Kinematic modelling of the robot

At the beginning of the session, the body have been modeled, but no relation have been defined between the different body. The kinematic must be defined. To do it, we use the device building environment in DELMIA software. A brief summary of steps is listed below:

- In the **Device Building** work-bench, a new product is created by inserting the .CATPart files as existing components.
- A new **mechanism** is created for the robot, and base is chosen as the **Fixed part**.
- Kinematic joints are defined between each parts from the base to the end-effector. In this model, all joints are **Revolute** and defined as rotations with respect to line and plane of the previous body. The lines are easily the rotation axes and planes are the plane of contact between the two joints. The **swap visible space** and **Hide/Show** features are useful in this step, to ease visibility during plane/line selection. Those joints are set as active joints.
- A **Home position** is defined using the dedicated tool. The home position is defined with only zero angle. We can note that this position is singular. The T_AXIS and R_AXIS are aligned. It is a wrist singularity. This position is visible on figure 2
- Using **Travel Limits** tool, the joint limits are set (using values from the Flyer-Robot-MH5S.pdf that lists the device specifications) as shown on figure 3.
- Similarly, using **Joint/TCP speed and acceleration limits** in the diagram below figure 4, we specify the permitted values as per catalogue.

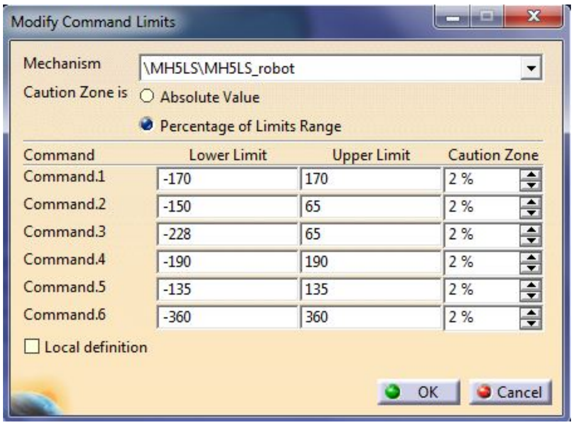

Figure 3: Setting the joint limits

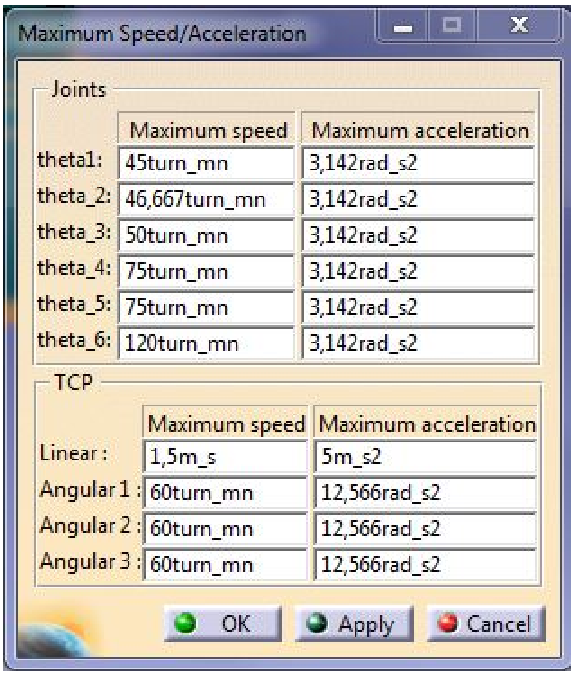

Figure 4: Setting the limits for joint speeds

2.3 Inverse Kinematic model of the robot

Once all the joints and frames have been defined, we can deduce the Inverse Kinematic model. The steps are listed below.

Inverse kinematic model definition We use the generic inverse to have all the posture of the robot available. The robot is specified as a **6-R** robot. We verify that the principle axis for each body is the Z-Axis. The frame of interest are used to define the model in order to be able to take into account the tool in the kinematic model.

Verification of the IKM The **jog mechanism** feature is used for verifying that the inverse kinematic model works properly. Moving the end-effector automatically changes the robot's joint positions. The inverse kinematic model has therefore been properly calculated. We also observe the different postures of the robot. We notice that the robot's initial configuration is indeed singular. Figure 5 shows this.

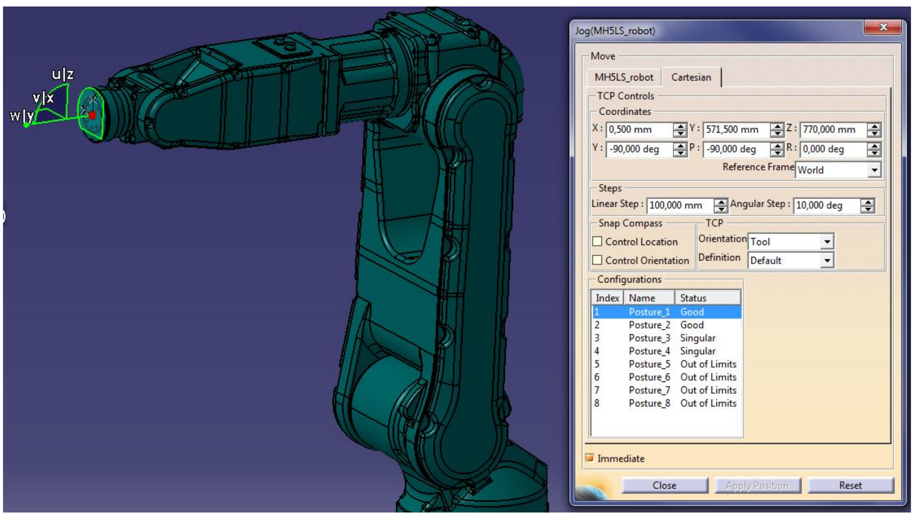

Figure 5: Robot available posture at home position

3 Programming and Optimization of the welding task

3.1 Robotic cell

The **Device Task Definition** Workbench of the DELMIA framework is used in this section. A robotic cell is first defined by inserting the robot and the tool as **product** and the environment as **resource** in our process definition. Then we mount the tool on the end effector using **Set Tool** and orient the tool and robot in a good posture. Next, a new **Tag Group** is created to build a set of tags that will correspond to the desired tool locations for spots to be welded.

Welding frame definition To define the welding task, we define a list of frames in the tag group, which the robot is going to follow. The position and orientation of the robot tool will correspond to that of the tag. If the frames are close enough, the robot will go from one frame to the next following the line we want to weld on. At first the orientation of all frames is the same. The **Tags** are defined at the intersection of the two surface we want to weld together. Here it is a line because it is the intersection of two planes. The inter-distance between the frames is kept at 40mm. With **Orient Tag/Tag Group** we can alter the tag orientations suitably, and it was set at 45° for weld angle, 20° for rake angle and 0° for wire angle.

Once the frame is created, we create the task of the robot. The robot will move from the first frame to the last and in the end, go back to the home position.

3.2 Initial simulation results

It is possible to now launch the robot for its welding task. We observe that at this stage, none of the points are reachable by the robot from its home position. The robot tries to reach the points by moving through the platform. Several reasons explain why this task is incorrectly defined.

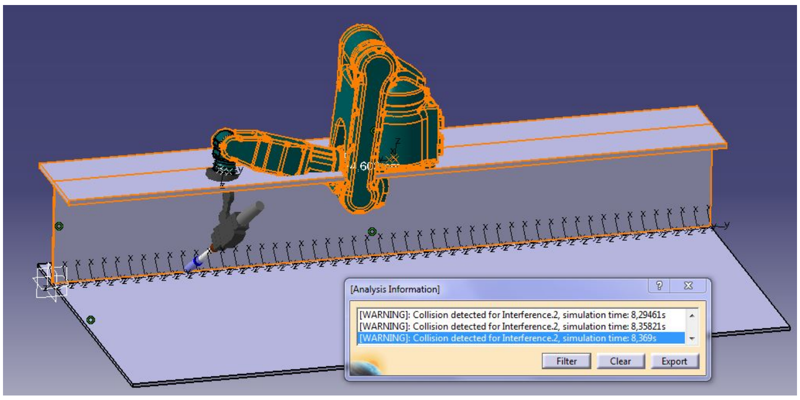

Figure 6: The robot goes through the platform with the first task definition

Firstly, the orientation of the tag makes the tool go through the platform. Secondly, we see that the robot is on its elbow down posture. It creates a collision between the robot and the environment and between the robot and its base. We must thus change the posture of the robot.

Another important concern is to take care of the reachability of the robot. We see that with this initial orientation of the tags, some points on the two extremities of the platform are not reachable at all. They are out of the robot workspace. So, attention must be given to find the correct robot configuration that avoids singularity and collisions, and the tag orientation to attain maximum possible reachability and avoid collision.

With the initial tag orientation, it is impossible for the moment to avoid collision between the tool and the platform. This practical work subject mentions a specific 45° for weld angle and 20° for the rake angle, which defines this particular welding task. The only angle that can be controlled is the **wire angle**, so we can only alter the z-axis orientation to avoid collision of the tool with the platform. So suitably changing the z-axis, we maybe able to avoid such collisions and the next section describes the changes done in this regard.

Hence, summarizing the insights from initial simulations, the variables we can act on for optimally performing the welding task, are listed below:

- The base position of the robot
- The posture of the robot
- Orientation of the frames of all the tags that (correspond to spots for welding)

3.3 Correction of the task definition

The following changes were introduced to make the welding task successful.

Optimal base placement The robot is placed centrally on the wall, at the exact middle location, so that the extremities of the workspace are symmetrically distanced from here. This maximizes our chances to reach more points, especially the ones towards the extreme ends. The robot is also put on the edge that is closer to the side of the wall that is to be welded with the platform.

Setting the orientation of the tags in welding trajectory We choose to modify the orientation of the z-orientation of the frames of the tags that are at the extreme points in the welding trajectory.

Interpolate the frames After setting the orientation of the extreme points, we use the **Interpolate Frames** feature between these two tags. Thus all frames between these two points have now been set to correct orientation.

Changing the posture of the robot After the interpolation is done, we perform the Jogging mechanism at each frame throughtout the trajectory to check that the robot has good posture without any singularity, at each tag point. It is to be noted here, that this **same** posture is maintained **constant** throughout the trajectory, so that there is no singularity crossing due to changing of postures during the welding.

Collision detection We add an interference constraint using the **Clash** feature, which creates an alert if at any point the robot encounters a collision with the platform or any other part in the environment. We activate this interference in the **Analysis Configuration**.

Creating a start point for the welding task The robot is initially kept at its home position. However it is observed that if the robot tries to reach an end point directly from this configuration, then it faces a collision with the wall (as the tool goes through it) even before actually starting the welding job. So, an intermediate start point is created. The robot moves from the home position to first this point, and then goes to an end point of the welding trajectory to begin the welding operation. This ensures that there is no collision encountered before the welding trajectory is reached.

Our welding task has been modified and now is ready to be implemented again with the robot. A point to note here is that due to the orientation of the tool with respect to the robot, the number of points in the welding trajectory that can be successfully welded without any collision, is *different* from the two ends of the trajectory, i.e it depends on which end of the trajectory we choose to start our simulation. We report our findings for the case when starting from the right-end as well as for starting from the left-end of the welding trajectory.

3.4 Final results and discussions

Welding starts from left end We observe that the first two tag points on the left-end, are out of reach of the robot since the arm is fully stretched. So the welding actually starts from the 3^{rd} tag point. The tool is always on the left of the arm as seen in the figure. Then we count how many tag points the robot can successfully weld without any collision in the trajectory defined by the tags. On figure 7, we can observe the final welding point (the 33^{rd} tag point) that the robot is able to reach without any collision. We end the welding task at this stage. The total number of points successfully welded is 31.

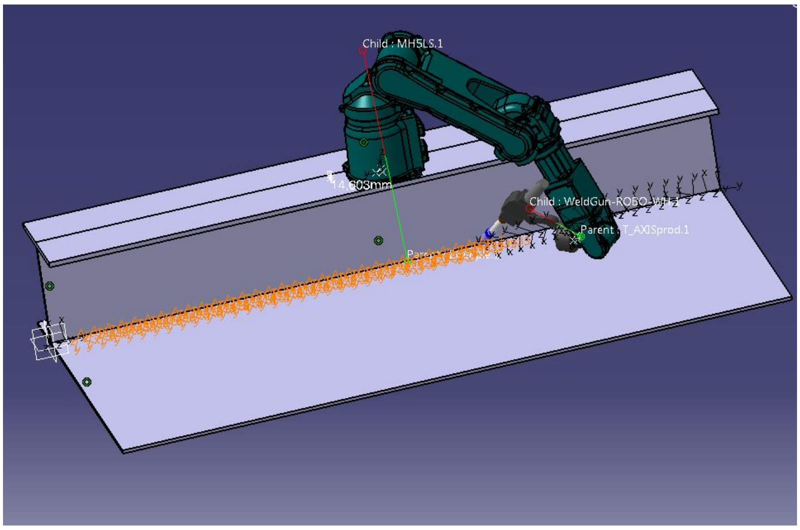

Figure 7: Trajectory defined on the left of the robot

The animation can be viewed at https://www.youtube.com/watch?v=-DJhwQVUUIc

Welding starts from right end Similarly, for the welding starting from right-end, the two extreme tag points at the start of the trajectory is out of reach of the robot. So welding starts from the 3^{rd} tag point. The tool is towards the right of the arm during the welding. It continues till the 39^{th} tag point, after which the tool encounters a collision with the platform and the welding task is terminated. The total count of points successfully welded is 37. Figure 8 demonstrates the trajectory successfully covered (tag points in yellow) and also shows the workspace volume of the robot.

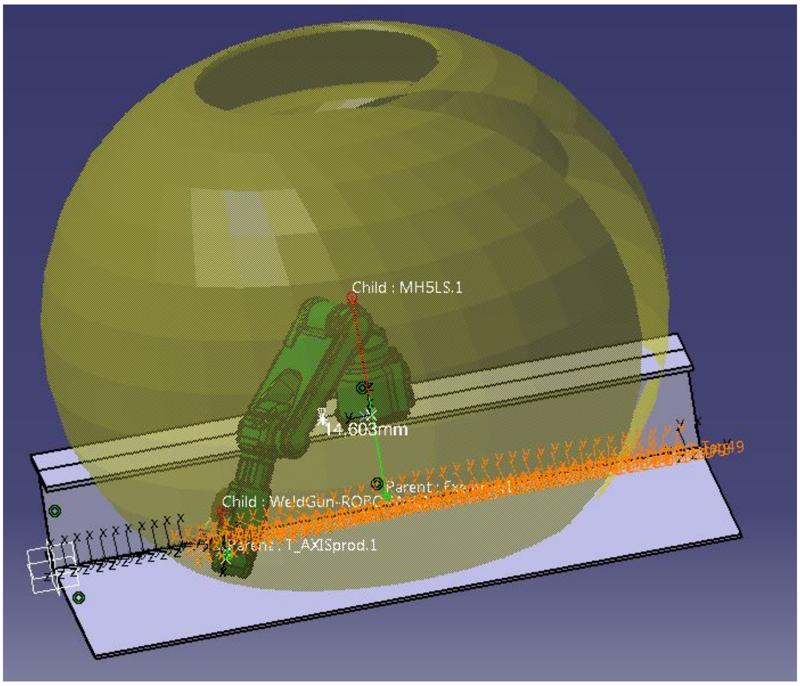

Figure 8: Trajectory defined on the right of the robot

The animation can be viewed at https://www.youtube.com/watch?v=QM8XxKoXPEM

Discussion of results We observe that more points can be successfully welded when starting from the right-end. The welding task encounters a collision between the tool and the platform while transitioning from the 37^{th} tag point to the next. The orientations of these consecutive tags are uniform. So, in order to align itself along both consecutive parallel frames, there is an unavoidable rotation that demands the last link must go through the platform. So the 38^{th} tag point is not accessible.

The same observation occurs earlier, at the 33^{rd} tag point for the first trajectory started from left-end. Since the collision is between the platform and the tool, this situation is unavoidable. Hence this is the terminating condition for the welding task in our work.

4 Comments on optimal placement of robot

The total count of welding points that can successfully accessed maybe increased by optimal placement of the robot. And the trajectory for welding also depends on this base placement. However this base placement optimization is highly linked to the orientation of the tool. This orientation varies along the trajectory, which therefore couples the system. Therefore, both the base placement as well as the orientation along the trajectory cannot be optimized at the same time. In our simulations, the robot had been placed at the edge so that it allows the possibility to accesss maximum end points, with fully stretched configurations. However, still some points were inaccessible as they were too far to be reached. If the robot was more inwardly positioned, the workspace would be even more limited than this.