Lab 1 - KUKA KR3 R540: Trajectory Programming

Jean-Louis Boimond
University of Angers

Objective of the lab: After training on the robot, create a script to enable the robot arm, equipped with a pencil, to draw the picture shown below.

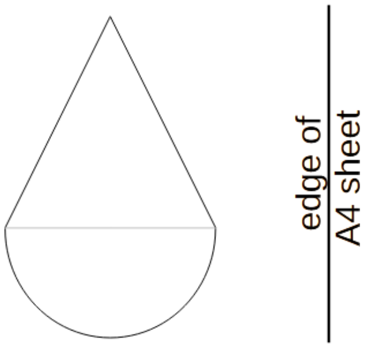

Figure 1: Representation of the picture to be reproduced by the robot.

The height of the triangle is equal to 80 mm and is parallel to one of the edges of the A4 sheet as shown in the previous figure. The radius of the half-circle is equal to 40 mm.

⇒ A first step is to locate the reference frame of the robot (denoted R_0), also named as the World frame, as well as the frame associated with the flange (denoted R_F) by using the following figure. In this figure, the posture of the robot results from the execution of the PTP HOME (PTP for Point-to-Point) move instruction in which $A1 = 0, A2 = -90, A3 = 90, A4 = A5 = A6 = 0$ (in °) (which is different from its *initial posture* in which $A1 = A2 = \cdots = A6 = 0°$).

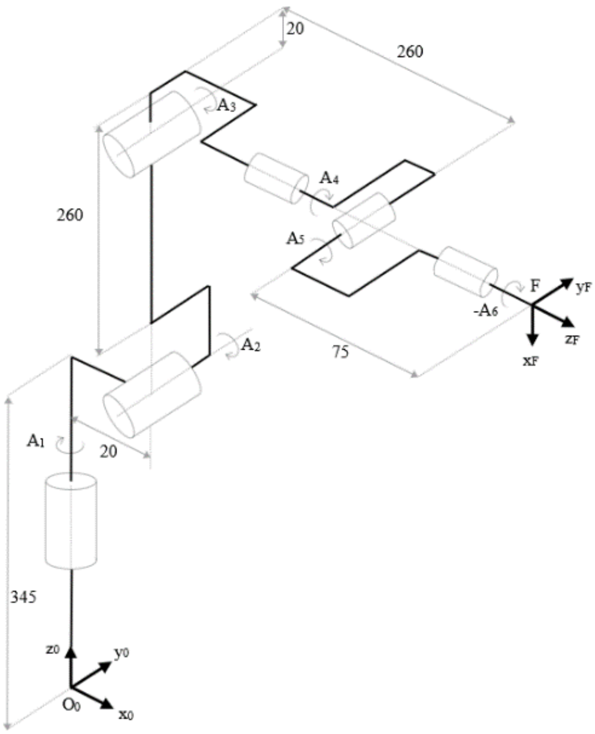

Figure 2: Representation of the posture of the robot corresponding to the HOME instruction.

Before programing the trajectory (made in §II), you need to learn how to handle the robot with the *Teach Pendant* of the robot (also called SmartPAD for Kuka robots) to:

- manually set the robot arm in motion, see §I.A,
- display the coordinates of the *Tool Center Point*, see §I.B,
- write and run a script to automatically move the robot arm, see §I.C.

I) GETTING STARTED WITH THE ROBOT

In this section, you'll learn how to use the basic manipulations of the robot required to realize the lab. It's up to you to test these functionalities **with caution**.

First, you need to be in the 'Expert' user group, rather than the default 'Operator', to be able to create/modify/delete a script. To do this, select the key  in the main menu (see the figure on the left below), then select **Configuration>User Group** (see boxes in red in the figure located in the center), then enter the password: **kuka** by validating it with the 'Return' key, or 'Save', circled in red in the figure on the right.

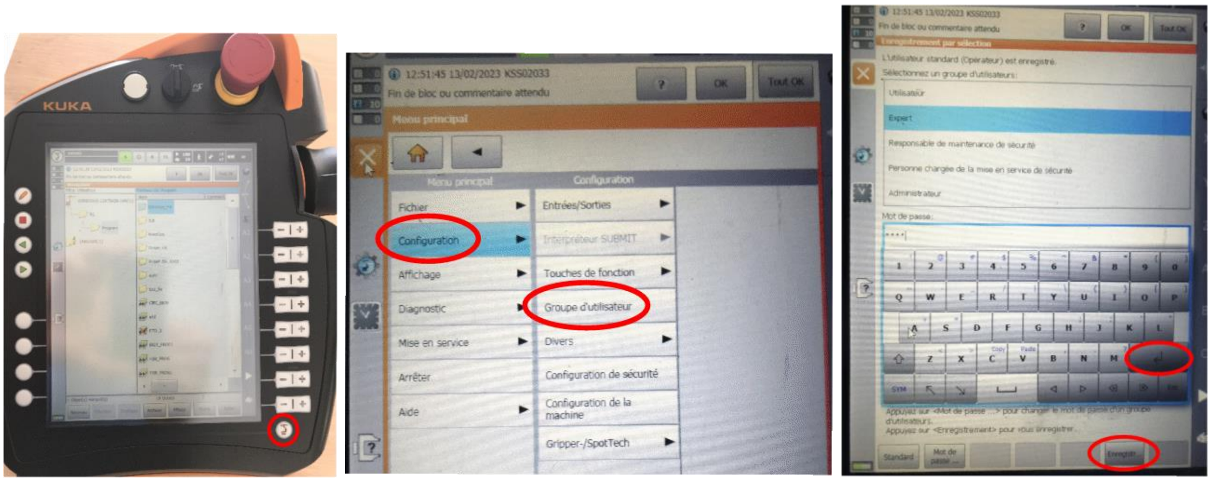

Figure 3: User group selection.

I.A) Manually setting the robot arm in move

Manual movement of the robot arm is conditioned by a (not too pronounced) pressure on the 'dead body', located at the rear side of the *Teach Pendant* on the right or left (as desired), see 2.2.*ii*, p. 8 of the document 'Robot Kuka KR C3'.

I.A.1) Movement in the Joint space

To manually set a robot joint in motion (through the angles $A1, A2, A3, A4, A5$ or $A6$):

- If you are not already in the main menu, select the key  (see the previous figure on the left);
- Select the icon circled in red in the figure on the left that follows, then on the one circled in red in the figure on the right, to indicate your wish to make movements according to the 'Axes';

Figure 4: Manual movement of the joints.

- With the 'dead body' activated, press the + (or -) key for the joint you want to move ($A1, A2, A3, A4, A5$ or $A6$) in the positive (or negative) direction of rotation.

NB: The – and + keys circled in blue in the previous figure enable to adjust the speed of the movements. So, you can reduce this speed when the tip of the pencil will be close to the carriage tray... **so as not to crush the pencil**.

I.A.2) Movement in the Tasks/Operational space

To manually set the *Tool Center Point* (TCP) in motion (for the moment the TCP corresponds to the frame associated with the robot flange (R_F), see Figure 2) along an axis (X, Y or Z), or around the axes (A, B either or C), of the reference frame of the robot (R_0/World):

- If you are not already in the main menu, select the key  (see Figure 3 on the left);
- Select the icon circled in red in the figure on the left that follows, then on the one circled in red in the figure on the right, to indicate your wish to make movements along the axes of the R_0/World frame;

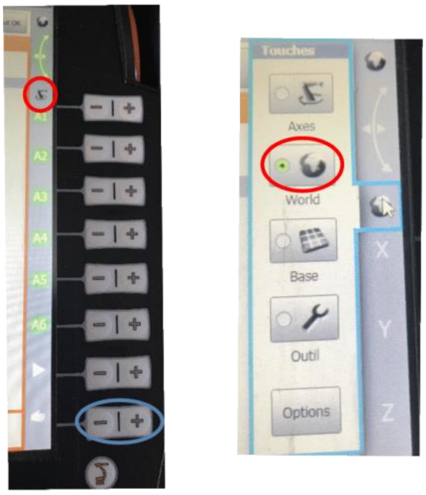

Figure 5: Manual movement of the TCP along/around the axes of the reference frame of the robot (R_0/World).

- Knowing that the 'dead body' is enabled, press the + (or -) key on the axis X, Y or Z along, or around, which you want to move the TCP in the positive (or negative) direction.

NB: The – and + keys circled in blue in the previous figure enable to adjust the speed of the movements. This way, you can reduce this speed when the tip of the pencil is close to the carriage tray... **so as not to crush the pencil**.

I.B) Access to Tool Center Point coordinates

The Tool Center Point (TCP) corresponds (for the moment) to the frame associated with the robot flange. To access the TCP coordinates, press the key in the main menu (located at the bottom right) circled in red in the left figure below.Then select **Display>Actual position** (see boxes in red in the figure in the center) to access the coordinates of the point in the articular space (i.e. $A1, A2, \dots, A6$) or in the operational space (i.e. X, Y, Z, A, B, C), by successively pressing the '**Cartesian/Axis specific' icon** circled in red in the figure on the right.

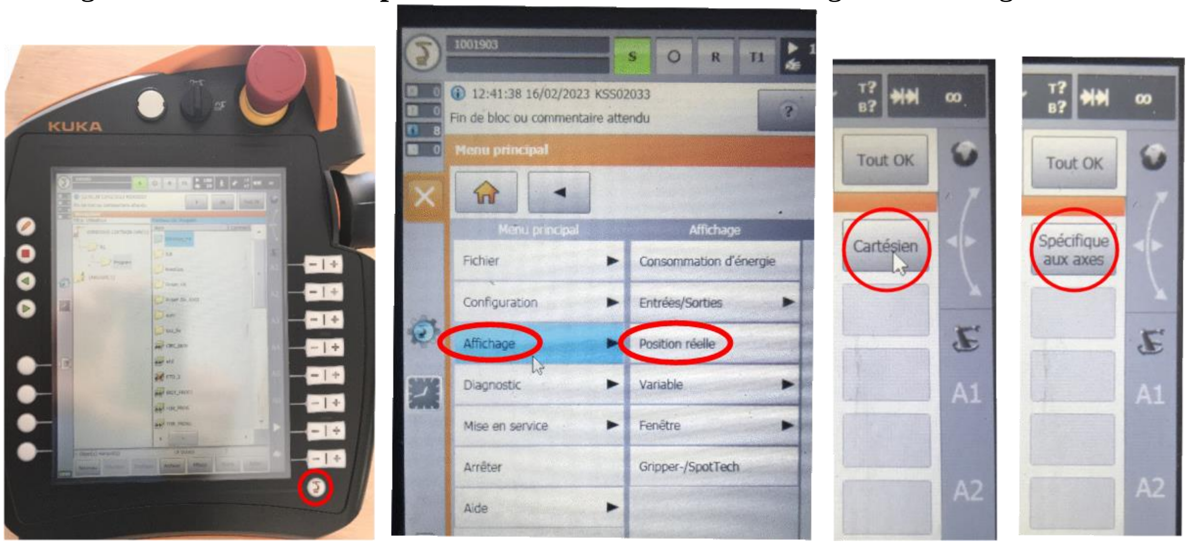

Figure 6: Display of the Joint or Operational coordinates of the TCP.

Tap the key to return to the main menu.

I.C) Automatic movement of the robot arm

I.C.1) Writing the script

<u>**Precautions for use:**</u>

1) In order not to modify the scripts installed in the robot controller, work only in the file '**TPEtudiants.src**', located in the 'KRC:\R1\Program' directory, whose command lines are as follows:

```
DEF TPEtudiants( )
; Reserved area for declaring variables
INI
; Reserved area for initializing variables
PTP  HOME   Vel= 100 % DEFAULT
; Reserved area for coding
PTP  HOME   Vel= 100 % DEFAULT
END
```

2) For security reasons, the program will always be run in manual mode **T1** (see §2.3.*i*, p. 13 of the document 'Kuka KR C3 Robot') in order to restrict the speed of movement to less than

250 mm/s. The current mode, T1 or T2, is indicated in the status bar of the *Teach Pendant* (see the blue box in the figure below).

Once the **TPEtudiants.src** file has been selected (see box in red in the left figure below), press the 'Open' key circled in red in the right figure that follows to edit the program.

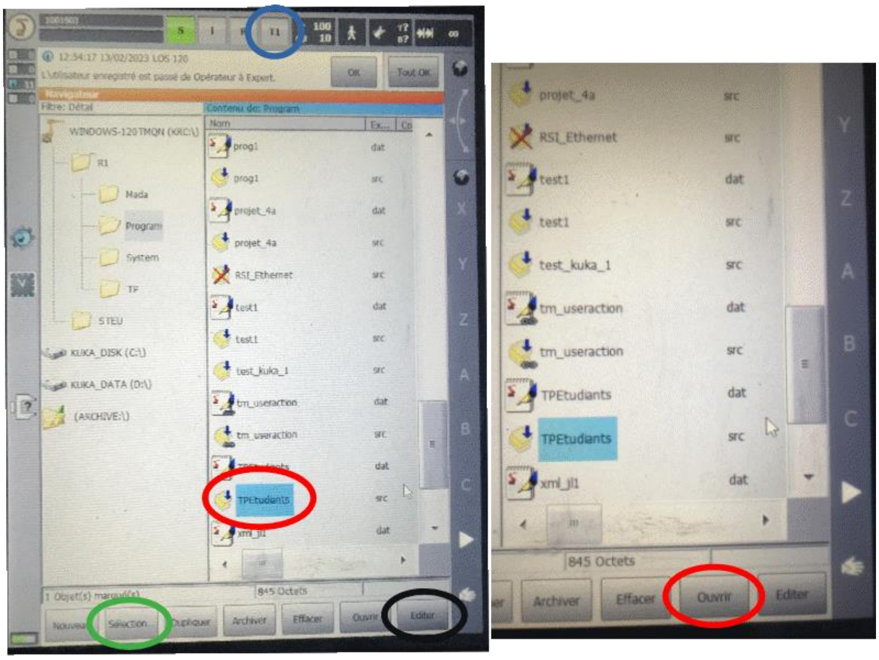

Figure 7: Editing the **TPEtudiants.src** file.

Press the key represented by a pencil (only accessible in 'Expert' mode, *see* p. 2 of the document), circled in red in the following figure, to allow you to write command lines (pressing the 'pencil' key again will close the window allowing you to write command lines).

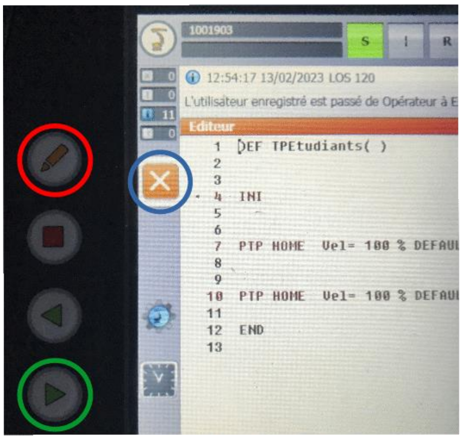

Figure 8: Manipulations related to the editing of the **TPEtudiants.src** file.

Clicking on the × key, circled in blue in the previous figure enable to save the changes made within the script before closing the window.

Note: A file with error(s) appears with a red cross through it. The '**Fault List**' button, located in the Buttons bar, can help you to understand the error(s).

I.C.2) Execution of the script corresponding to the TPEtudiants.src file

Select the **TPEtudiants.src** file through the "**Select**" key circled in green in Figure 7, all the necessary files and data lists are then *linked* to create an executable program.

The current position of the robot rarely coincides with the point indicated in the first move instruction to be performed (indicated in the script by the Block pointer), so a 'block coincidence' (denoted BCO) must be performed. Knowing that the execution of the movement allowing this correspondence does not represent a tested and programmed movement, the movement is always carried out at a reduced speed by pressing the **Start** key (circled in green in the previous figure) and activating the 'dead body' to be able to stop the robot if necessary. Hold down the **Start** key (to execute the block coincidence) until the 'Message Window' displays 'Block Coincidence Reached'. The robot will then stop and the script is ready to run. To run the script, hold down the **Start** key (there is no need to activate the 'dead body').

Simply stop pressing the **Start** key to stop running a script. Press and hold the **Start** key again to restart the sequence.

Press the **Edit** key, circled in black in Figure 7, then select (from the menu that appeared) the **Program Abandon** key to exit execution mode.

II) REALIZATION OF THE TRAJECTORY

II.A) 'Pencil + Pencil holder' tool consideration

Taking into account the 'Pencil holder + pencil' tool enable to *locate* (position and orientate) the TCP at the level of the tip of the pencil (and not at the level of the robot flange, as before).

The tool is described in the following figure.

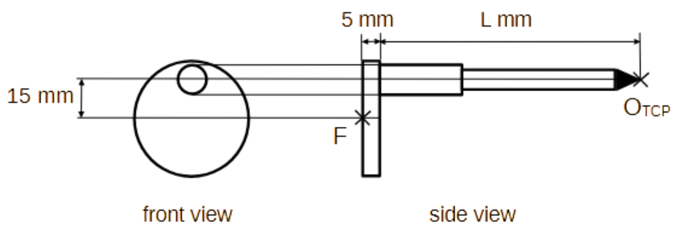

Figure 9: Front and side views of the tool.

The tool is attached to the robot flange as shown in the figure below.

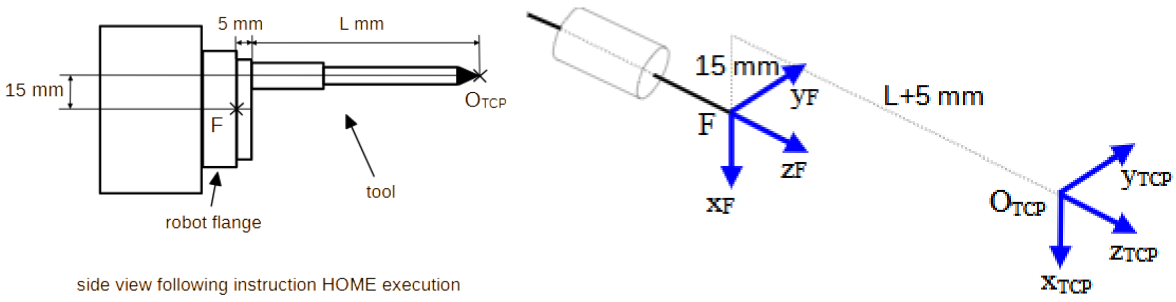

Figure 10: Attachment of the tool to the robot flange.

The tool is taken into account through the instruction

```
$TOOL={FRAME:X ??, Y ??, Z ??, A ??, B ??, C ??}
```

in which the $TOOL variable Kuka is defined where X, Y, Z set the coordinates (in mm) of O_{TCP} expressed in the R_F frame and A, B, C are the angles (in °), defined according to the convention z, y, x, that orientate the TCP with respect to the R_F frame. For example, the instruction $TOOL={FRAME:X 100, Y 100, Z 100, A 0, B 0, C 0} is such that $X = Y = Z = 100\ mm$ and $A = B = C = 0°$.

Q1) Give the values of X, Y, Z, A, B, C corresponding to the tool used. **CAUTION**: add 10 mm to the length (L) of the pencil so as not to risk damaging it. Thus, the TCP will be positioned $+10\ mm$ from the pencil tip along z_{TCP} axis.

⇒ Place the previous instruction just after the first instruction in the script (i.e., PTP HOME Vel= 100% DEFAULT) so that the points defined in the following take the tool into account.

II.B) Definition of a frame associated with the A4 sheet

A frame linked to the A4 sheet, fixed on the trolley tray, will be defined so that you do not have to relearn all the points (used to realize the trajectory) in the event of a change, voluntary or not, in the *situation* (position and orientation) of the A4 sheet in relation to the robot arm.

⇒ Fix/tape the A4 sheet (on which the drawing will be made) parallel to x_0 axis of R_0 frame (of course the sheet will be placed in the space that can be reached by the robot!!).

A first idea (to make it as simple as possible) is to define the frame associated with the A4 sheet, denoted $R\_Feuille$, by taking an orientation identical to that of the R_0 frame and an origin corresponding to a point, denoted $P0$, belonging to the drawing to be made, as shown in the figure that follows.

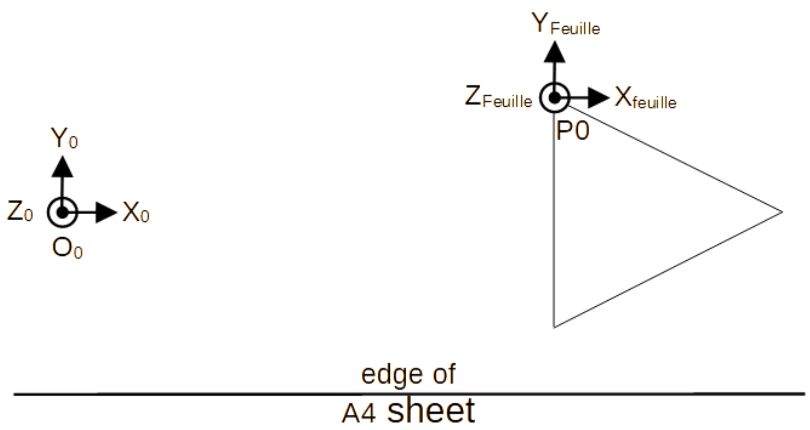

Figure 11: Orientation of $R\_Feuille$ frame identical to that of R_0 frame.

Q2) Give the values A, B, C of the resulting $R\_Feuille$ frame knowing that they correspond to the angles of rotation (in °) around the axes **successively** z, y, x capable of orienting the vectors of the $R\_Feuille$ frame with respect to the R_0 frame.

Q3) What would be the problem if we applied the PTP R_Feuille move instruction, which allows the frame associated with the Tool Center Point (TCP), in our case the robot flange, to merge with the $R\_Feuille$ frame (the PTP instruction is detailed in §2.7.*i*, 'Point-to-Point motions', p. 26 of the document 'Robot Kuka KR C3')?

Q4) How can the values A, B, C of the $R\_Feuille$ frame be modified to solve this problem, *i.e.* to enable the application of the PTP instruction?

Q5) Give the X, Y, Z values of the $R\_Feuille$ frame, knowing that they correspond to the coordinates (in mm) of $O_{Feuille}$ point in the R_0 frame. Explain why the value of Z is equal to 0. A solution to determine the values of X, Y is to *locate* (situate and orientate) the TCP at the point corresponding to the $R\_Feuille$ frame, to do this we proceed in 4 steps:

- declare and define in the **TPEtudiants.src** script a point, denoted P, such as:
  - its *position* (indicated through the values X, Y, Z) is close to that of the origin of the $R\_Feuille$ frame,
  - its *orientation* (indicated through the values A, B, C) corresponds to the desired orientation for the $R\_Feuille$ frame.

  See for example the script described on p. 23, as well as §2.6, 'Variables and declarations', p. 19;
- apply the PTP P instruction, so the *orientation* of the TCP will be the same as that of the $R\_Feuille$ frame ;
- move the pencil tip to the origin point of the $R\_Feuille$ frame by making manual movements along the axes of the R_0/World frame of the robot (*i.e.* in the Operational space, see §I.A.2) so as not to change the orientation of the TCP, so that the *position* of the TCP will be the same as that of the coordinate system;$R\_Feuille$
- take the values X, Y, Z, A, B, C of the $R\_Feuille$ frame to be those of the TCP (since the TCP corresponds to the $R\_Feuille$ frame).

⇒ Now, orientate the A4 sheet in order that the x_0 axis is no longer parallel to the $x_{Feuille}$ axis. Measure the angle, denoted α, made between these two axes using a protractor or a ruler.

Q6) Modify the values A, B, C of the $R\_Feuille$ frame found in question Q4 to take into account the angle α.

II.C) Script production

Let $P1, P2$ be the points defined in the $R\_Feuille$ frame, represented in the figure below, used to draw the picture.

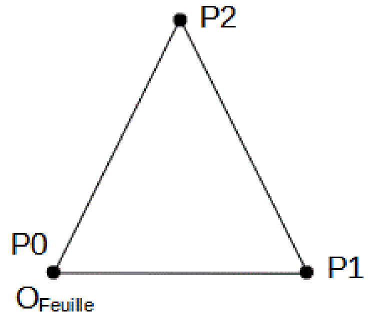

Figure 12: Representation of the $P1, P2$ points.

II.C.1) Robot Programming

⇒ 1. Moving the pencil to the $P0$ point: Complete the script to move the robot to the $P0$ point, see §2.7. *i*, 'Point-to-Point motions', p. 26 about the instruction to use;

⇒ 2. Drawing the triangle: Complete the script to draw the triangle. To do this, you will use the $P1, P2$ points (defined in the $R\_Feuille$ frame, see p. 22-23 concerning their use) knowing that the robot arm will have to move in a straight line to draw the triangle, see §2.7. *ii*, 'Linear motions', p. 27 concerning the instructions to use;

⇒ 3. Drawing the half-circle: Complete the script to reproduce the half-circle, taking care to define an auxiliary point ($P3$) defined in the $R\_Feuille$ frame, see §2.7. *ii*, 'Circular motions', pp. 27-28 concerning the instruction to use.

II.C.2) Calculation of the position of point $P2$ in the R_0/World frame

Q7) Calculate the position of the point $P2$ in the reference frame R_0 from the coordinates of this point in the $R\_Feuille$ frame and the coordinates of the $R\_Feuille$ frame in the R_0 frame. Remember that the angles A, B, C, expressing the orientation of a frame, are defined according to the convention (z, y, x).

⇒ Read the coordinates of the point $P2$ expressed in the R_0 frame to validate your calculation. To do this, insert the `WAIT SEC 5` instruction after the one enabling the arm to move to the point $P2$ (stopping the movement for 5 seconds after reaching this point will give you time to stop the execution of the script and thus allow you to read these coordinates).