UR Log Viewer - Analysis of Trapeze Velocity Profile of a UR robot

Jean-Louis Boimond
University of Angers

This document highlights the trapezoidal speed profile at the joints of a Universal Robots (UR) robot, in this case a UR3e robot, which appears when the movement of the robot arm is calculated in the *joint space*; the interest of such a profile is to minimize the travel time of the arm. The position, speed, and acceleration of the joints during a movement are measured using the *UR Log Viewer* software, installed on a PC connected to the robot controller, see §1. Initially, to facilitate understanding, the movement of the robot arm, programmed using *Polyscope* (see §2.a), consists of moving a single joint, namely joint no. 3[1] (called the *elbow*), which amounts to performing a rotational movement around this joint, see §2, §3.1. The simultaneous movement of the six joints of the robot arm is discussed in §3.2 and will require synchronization of the joint movements so that all joints reach their final angular value at the same time.

Table of Contents

1 UR Log Viewer Software

1.1 UR Log Viewer Overview

UR Log Viewer is a Windows software available on the robot manufacturer's website. It enables users to view/analyze/record, using a PC connected to the robot controller, certain data contained in several files automatically generated during the execution of a task performed within UR robot controllers. The aim is to troubleshoot/improve an application under development by better understanding the robot's behavior. In our case, we will simply use this software to view the

[1] This is the third joint, the first being the *base*, the second being the *shoulder*.

trajectories of the position, speed, and acceleration of certain joints during the movement of the robot arm.

1.2 Installing UR Log Viewer on a PC

UR Log Viewer is downloadable <here> (it can also be accessed by typing '*log viewer*' in the search field of a window that appears when selecting the ***Training&Support>Resources&Downloads*** tab on the Universal Robots website: https://www.universal-robots.com/).

Remark: It is not necessary to disable Windows Firewall to use UR Log Viewer, however you will probably need to indicate to the firewall that you authorize the software to access network traffic when using the software for the first time.[2]

1.3 Connection between the PC and the robot

The connection between the PC and the robot controller requires knowledge of the robot IP in order to configure a PC IP address that is compatible with that of the robot.

1.3.a Identification of the robot IP address

Click on the icon (located at the top right of *Teach Pendant* of the robot), then select **Settings**, then the ***System>Network*** tab, which gives access to the following information, namely in the case of the UR3e robot of the school:

Static IP address: 169.254.123.187,
Subnet mask: 255.255.255.0,
Default gateway: 0.0.0.0,

as shown in the figure below.

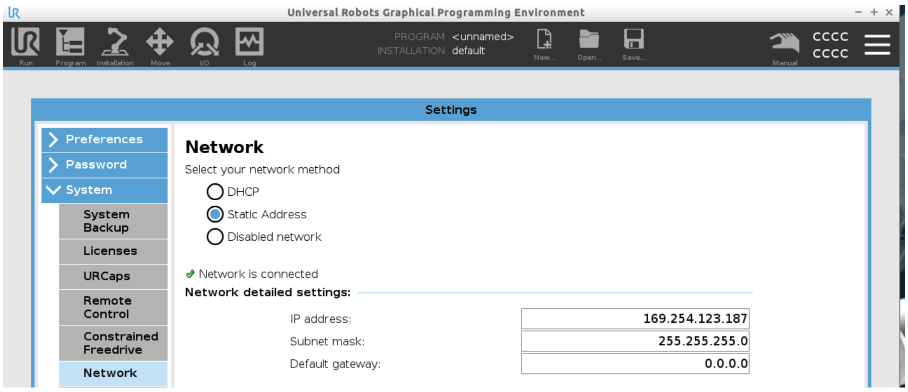

Figure 1: Static IP address of the robot.

1.3.b Configuration of the PC IP address to be compatible with that of the robot

To configure the PC IP address in Windows 11, open the **Control Panel** and follow these steps:

- Click on **Network and Internet>Network and Sharing Center**,
- Click on **Change map settings**,
- **Right-click** on the **Ethernet** card to which the Ethernet cable will be plugged (in this case Ethernet card 2) to click **Properties**,

[2] The Real-Time Data Exchange (RTDE) interface of the robot controller is supposed to be enabled, which is the default setting during controller execution.

- Select **Internet Protocol version 4 (TCP/IPv4)** from the list, then click **Properties**,
- Check **Use the following IP address** (*i.e.*, a static address, not a dynamic one) to set an IP address in the same address range as the robot, for example, 169.254.123.100, respecting the subnet mask, *i.e.*, 255.255.255.0, as shown in the following figure (in French!):

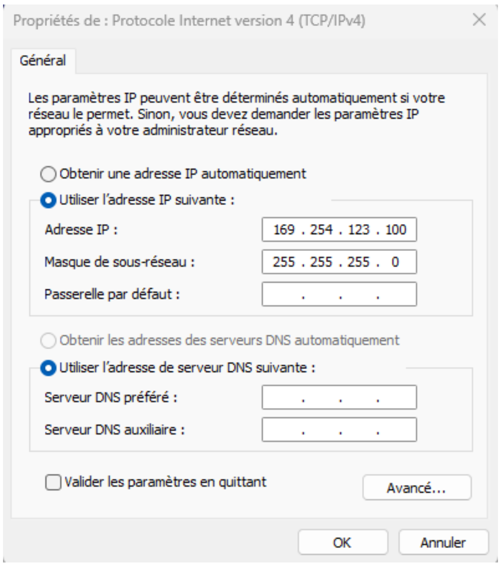

Figure 2: Static IP address of the PC.

1.3.c Wiring between the PC and the robot

Connect an Ethernet cable between the PC and the robot controller. Test the connection by pinging the robot IP address (in our case: 169.254.123.187) in the PC command prompt.

2 Programming Movement Using Polyscope, Taking Measurements

2.a Programming Movement

As mentioned in the introduction, the movement of the robot arm that we are going to study simply consists of rotating joint no. 3. from 0° to 80°. The resulting movement is shown in the *html* animation accessible <here>.

We define a starting point, named **A**, and an end point, named **B**, in the program, such that: joint no. 3 is equal to 0° at point **A** and 80° at point **B**, knowing that the values of the other joints are the same for both points[3], namely: −90° for joints no. 1 (base) and 2 (shoulder), 0° for joints no. 4, 5, 6. The program must be such that:

- The movement to move the robot arm from point **A** to point **B** is calculated in joint space, using the **moveJ** instruction, knowing that the robot arm must be stationary at these two points;

[3] There is no movement of a joint between points **A** and **B** if the values of the joint at these points are the same, since the joint value is constant during the movement from **A** to **B**.

- The maximum speed, known as *cruising speed*, is set at $60^{\circ}/s$, the maximum desired acceleration is set at $80^{\circ}/s^2$. Note that these speed and acceleration are distinct from the maximum (permissible) speeds and accelerations of the robot arm joints[4].

Realization 1: Create a program that moves the robot arm from point **A** to point **B** using Polyscope software[5]. You will note, as illustrated in Figures 3.a, 3.b, 3.c below, that:

- The **Program** tab allows the creation of a Polyscope program by adding commands available in the **Basic** commands list, we will use in our case the **Move** and **Waypoint** commands;
- The **Move** command controls movement between the waypoints defined *under* the command. In our case, two waypoints are used, one named **Waypoint_1** (rather than **A**), the other named **Waypoint_2** (rather than **B**), defined below. Three specifications are given in the window on the right relative to the **Move** command:
    - The type of movement in the joint space *via* **MoveJ**,
    - The speed and acceleration when moving the robot arm between its two crossing points in the **Joint speed** and **Joint acceleration** fields respectively. This is where the cruising speed and the maximum desired acceleration will be defined,

  see the following figure for the settings for this command:

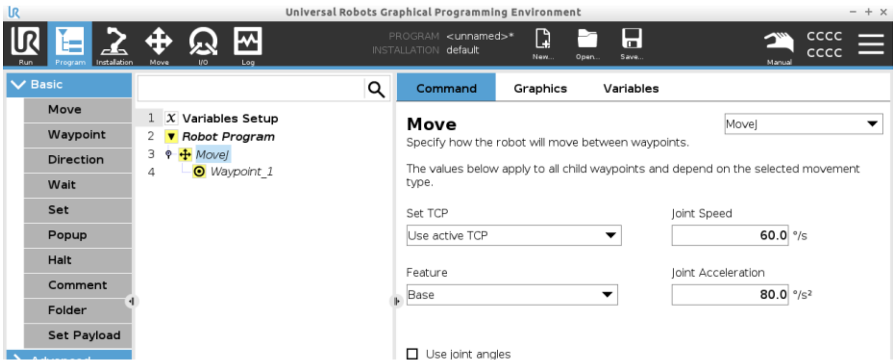

Figure 3.a: Definition of the **Move** command.

- You will note that the **Waypoint_1** point has been defined by default, the **Waypoint_2** point will be defined using the **Waypoint** command. For these two points, we will indicate that:
    - The robot arm stops at the point (by checking the box **Stop at this point**) and the joint speed and acceleration are those defined in the previous **Move** command (by checking the **Use shared parameters** box). See the following figure for the **Waypoint_1**:

---

[4] The security level of the 'Factory Presets' must be high enough to allow the robot to reach cruising speed. In our case, the level is set to 3, which allows a maximum speed of around $190^{\circ}/s$ for the first three joints and around $370^{\circ}/s$ for the last three.

[5] Polyscope enables the robot arm to be moved by learning a series of points acquired using the *Teach Pendant*. Consult the UR3e robot user manual by typing '*user manual UR3e*' in the search field of a window accessible at: https://www.universal-robots.com/support/

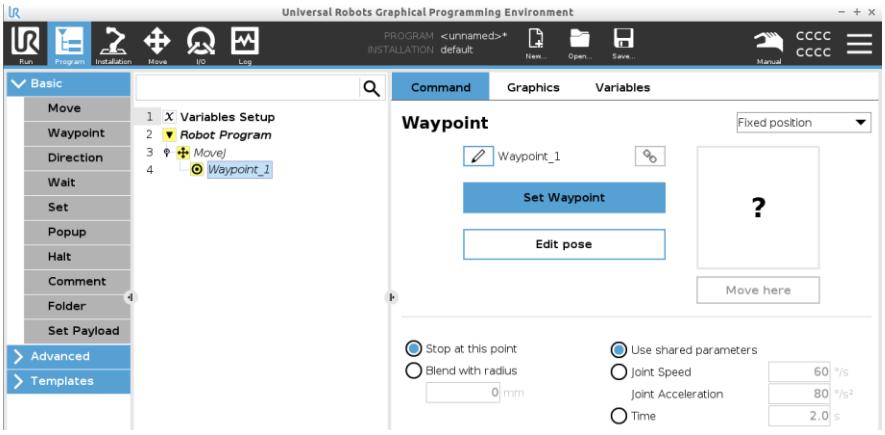

Figure 3.b: Setting point **Waypoint_1**.

- The joint values for **Waypoint_1** point are defined by selecting the **Edit pose** button (accessible in the **Command** tab that appears by default) *via* the fields **Base**, **Shoulder**, **Elbow**, **Wrist 1**, **Wrist 2**, **Wrist 3**, see the following figure:

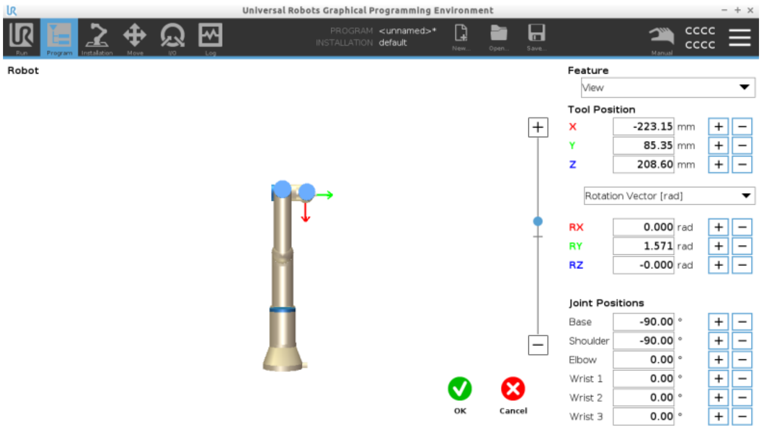

Figure 3.c: Definition of the 6 joint values of the **Waypoint_1** point.

- The **Halt** command added at the end of the program stops its execution (otherwise, the robot arm movement would repeat).

2.b Taking measurements during movement

Run the UR Log Viewer software. The steps described below are performed in the home window to measure and visualize the position, speed, and acceleration of joint no. 3 during the movement resulting from **Realization 1**:

- Select the **Record Data** tab to check the boxes corresponding to **Joint Angle**, **Joint Velocity**, and **Joint Acceleration** (if necessary, uncheck the other boxes). Note that the measurement frequency is set to $500\ Hz$ by default;
- Enter the IP address of the robot controller (namely 169.254.123.187) in the **IP** field (located at the top center of the window);
- Click on the **Start** button (located at the top right of the window) to start taking measurements: in practice, a few seconds before executing (from the *Teach Pendant*) the movement of the robot arm, *i.e.*, before the arm leaves point **A**;
- Click the **Stop** button to stop taking measurements once the arm has stopped at point **B**;

- Select the **Log Reader** tab and double-click on the *Flightrecord* file containing the measurements taken previously. Click on **UR3** in the window that appears to view the position, speed, and acceleration curves for the six joints.

Realization 2: Using UR Log Viewer, measure the position, speed, and acceleration of the joints during the movement corresponding to the Polyscope program carried out in **Realization 1**. View those corresponding to joint no. 3 (the elbow) in order to perform their analyses below.

3 Measurement Analysis

The calculation of the movement corresponding to the **MoveJ** instruction, used in the program carried out in **Realization 1**, is performed in the joint space and results in a trapezoidal profile of the speed of joint no. 3, which minimizes the movement time of the joint, taking into account the cruising speed and the desired maximum acceleration.

3.1 Validation of the trajectories of position, speed, and acceleration of joint no. 3.

We denote the trajectory of position by q, the trajectory of speed by $\dot{q}$, and the trajectory of acceleration by $\ddot{q}$ for joint no. 3. These trajectories are described in the following figure.

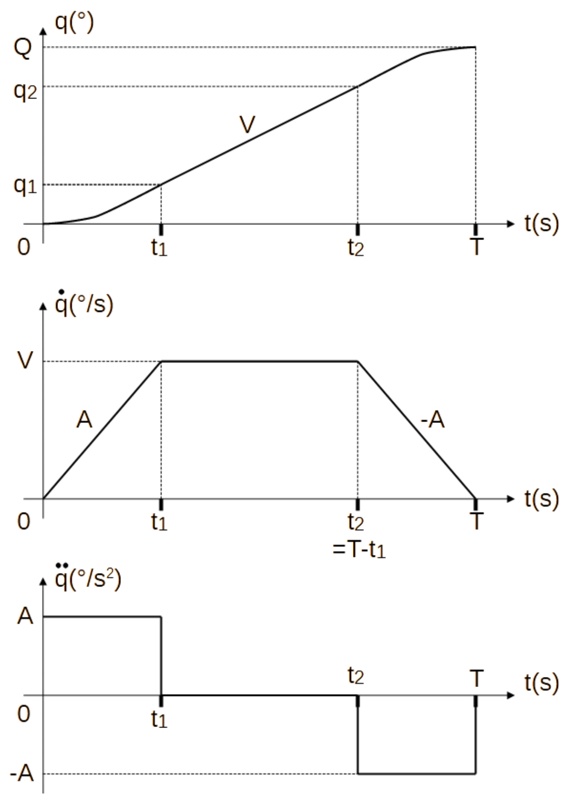

Figure 4: Position (q), speed ($\dot{q}$), and acceleration ($\ddot{q}$) of joint no. 3.

The speed trajectory is called *trapezoidal*, its trapezoid shape comes from its three-segment construction:

- The first segment whose slope is equal to the maximum desired acceleration (A) of the joint during a *constant acceleration* phase (between times 0 and t_1);

- The second segment with zero slope (due to zero acceleration) and whose value is equal to the cruising speed (V) of the joint during a cruising phase (between times t_1 and t_2);
- The third segment whose slope is equal to the maximum desired deceleration ($-A$) of the joint during a *constant deceleration* phase (between times t_2 and T) knowing that the duration of this segment is equal to that of the first segment, *i.e.*, $t_2 = T - t_1$.

The position trajectory, resulting from the integration of the speed trajectory, consists of a segment of slope equal to V between times t_1 and t_2, connected to two parabolic segments at the initial position (between times 0 and t_1) and final position (between times t_2 and T). Let:

$$q(t) = \begin{cases} \frac{1}{2}At^2 & \text{for } 0 \leq t \leq t_1, \\ V(t - \frac{t_1}{2}) & \text{for } t_1 < t \leq t_2 \ (= T - t_1), \\ Q - \frac{1}{2}A(t - T)^2 & \text{for } t_2 < t \leq T. \end{cases}$$

We observe a symmetry of the trajectory q with respect to the point $(T/2\,, Q/2)$.

The acceleration trajectory, resulting from the derivation of the speed trajectory, consists of a segment with a (constant) value of zero between times t_1 and t_2, connected to two segments with a (constant) value equal to: A between times 0 and t_1, and $-A$ between times t_2 and T. We observe a symmetry of the trajectory with respect to the point $(T/2\,, 0)$.

N.B.: Points **A** and **B** are assumed to be sufficiently distant from each other to allow a cruising speed to be established during the movement from **A** to **B** (which is the case if $t_2 > t_1$) and thus to allow a trapezoidal speed trajectory to appear for joint no. 3. If the distance between points **A** and **B** is too small, the result is a (only) triangular speed trajectory, known as *a bang-bang*, due to a lack of time for the joint to reach its cruising speed.

Calculation 3: The cruising speed V, the maximum desired acceleration A, and the angular value Q of joint no. 3 at point **B** (the angular value at point **A** being zero to simplify calculations) are assumed to be known. Express, as a function of V, A and Q, the values t_1, q_1, t_2, q_2, T shown in the previous figure where:
- t_1 is the time when the speed $\dot{q}$ reaches its cruising speed V, knowing that this speed was reached linearly and that the robot arm was stationary at time 0,
- q_1 is the position reached at time t_1,
- t_2 is the time when the speed $\dot{q}$ leaves its cruising speed V to linearly reach zero speed at time T (which causes the robot arm to stop at time T),
- q_2 is the position reached at time t_2,
- T represents the time taken by the joint to reach point **B** (the start of the arm movement occurring at point **A** at time 0).

Realization 4: Using the (literal) formulas determined in **Calculation 3**, calculate the numerical values of t_1, q_1, t_2, q_2, T for the values V, A, and Q considered when measuring the position, speed, and acceleration of joint no. 3 *via* UR Log Viewer. Compare them with those measured in **Realization 2**. What conclusions can you draw about the validity of the literal formulas?

3.2 Synchronization when moving several joints simultaneously

Until now, we have focused on moving a single joint (namely joint no. 3), which amounts to studying a simple rotational movement of the joint. Now, let us consider the case where moving the robot arm from a point **A** to a point **B** requires the joint movement of all six joints. The following calculation method is frequently used by robot manufacturers. The speed profile of the joints is always trapezoidal, but the calculation of these profiles must also take into account the fact that:

- The movement of the joints is synchronized so that all joints reach their final angular value at the same time (same value of T),
- The acceleration and deceleration times are the same for all joints (same value of t_1, and therefore also the same value of t_2 since $t_2 = T - t_1$).

The calculation is performed in several steps, namely:

- The trajectories of each of the six joints are calculated independently of one another. From this, deduce the joint, denoted j in the following, with the *slowest trajectory* to reach point **B**. This trajectory will be the one used to control joint no. j;
- The trajectory of each of the five other joints is *synchronized* relative to joint j so that its time t_1 (the time at which the joint reaches its cruising speed) is equal to time t_{1_j} and its time T (the time taken by the joint to reach point **B**) is equal to time T_j. Consider, for example, joint no. k ($k \neq j$), we need to calculate the speed v and acceleration a that define the *synchronized trajectory*, denoted ${}^{S}\dot{q}_k$. Let: $\dot{q}_k$ be the trajectory *to be synchronized*, shown in Figure 5.a; $\dot{q}_j$ be the *slowest* trajectory and ${}^{S}\dot{q}_k$ be the *synchronized* trajectory, shown in Figure 5.b:

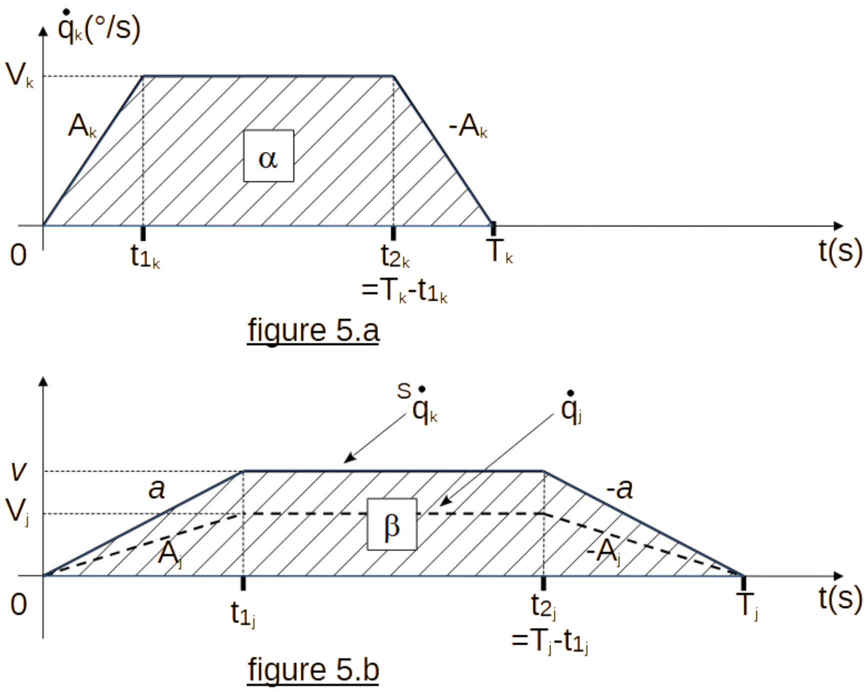

Figure 5: Trajectories $\dot{q}_k$ (see Figure 5.a), $\dot{q}_j$ and ${}^{S}\dot{q}_k$ (see Figure 5.b).

Note that the area α of the trapezoid relative to $\dot{q}_k$, shown in Figure 5.a, is equal to the angular value Q_k traveled by joint no. k in the interval $[0, T_k]$, indeed, we have:

$$\alpha = \int_0^{T_k} \dot{q}_k(\tau)\, d\tau = q_k(T_k) = Q_k.$$

Therefore, synchronizing trajectory $\dot{q}_k$ relative to trajectory $\dot{q}_j$ consists of calculating the speed v and acceleration a of the *synchronized* trajectory ${}^S\dot{q}_k$ as a function of V_j, A_j, Q_j, Q_k (data extracted from the joints j and k assumed to be known) so that:

- Its time to reach cruising speed (${}^St_{1_k}$) is equal to time t_{1_j},
- Its time to reach point **B** (${}^ST_k$) is equal to time T_j,
- The area of its trapezoid, denoted β in Figure 5.b, which corresponds to the angular value traveled by the joint in the interval $[0, T_k]$, is equal to the area α.

Thus, we have: $\alpha = \frac{1}{2}A_k t_{1_k}^2 + (T_k - 2t_{1_k})A_k t_{1_k} + \frac{1}{2}A_k t_{1_k}^2 = A_k t_{1_k}(T_k - t_{1_k})$,

similarly, we have: $\beta = a t_{1_j}\left(T_j - t_{1_j}\right)$ since ${}^St_{1_k} = t_{1_j}$ and ${}^ST_k = T_j$.

We can then deduce the values v and a such that $\alpha = \beta$, that is:

$$a t_{1_j}\left(T_j - t_{1_j}\right) = A_k t_{1_k}(T_k - t_{1_k}) \text{ with } t_{1_j} = \frac{V_j}{A_j}, t_{1_k} = \frac{V_k}{A_k}, T_j = \frac{V_j}{A_j} + \frac{Q_j}{V_j}, T_k = \frac{V_k}{A_k} + \frac{Q_k}{V_k}$$

It follows that: $a = A_j \frac{Q_k}{Q_j}$ and $v = a t_{1_j} = V_j \frac{Q_k}{Q_j}$.

Remark: We can see in Figure 5.b that $a > A_j$ and $v > V_j$, which is due to the fact that the area β $(= \alpha = Q_k)$ is greater than the area, equal to Q_j, of the trapezoid relative to the speed $\dot{q}_j$. This observation can also be deduced from the previous formulas.

We will now illustrate these calculations by considering a movement of the robot arm, from a point **A** to a point **B**, which requires the movement of joints no. 3 (elbow) and 4 (wrist 1). To do this, we redefine points **A** and **B** described in §2.a as follows:

- joint no. 3 is equal to $0°$ at point **A** and $80°$ at point **B**,
- joint no. 4 is equal to $0°$ at point **A** and $140°$ at point **B**,
- the values of the other joints are the same for both points, namely: $-90°$ for joints no. 1 (base), 2 (shoulder) and $0°$ for joints no. 5, 6.

As indicated in §2.a, the cruising speed is set at $60°/s$, and the maximum desired acceleration is set at $80°/s^2$.

Realization 5:

1) Create a Polyscope program that allows the robot arm to perform such a movement.
2) Use UR Log Viewer to view the position, speed, and acceleration during the movement of joints no. 3 and 4:
   - Which joint is the slowest? Which is the joint whose trajectory is synchronized?
   - Extract the values a and v from the synchronized trajectory.
3) Compare the measured values of a and v with those obtained from the previous calculations. Compare the measured values of t_1, t_2, and T with those obtained from the calculation.