Kinematic control of a robotic manipulator with redundant degrees of freedom: Trajectory tracking and obstacle avoidance

(Redundant manipulators: Path Following and Obstacle Avoidance)

DESCRIPTION OF THE ROBOT

A. The robot

Cobots, or *collaborative robots*, are robots that are intended for direct and safe interaction with humans, unlike traditional *industrial robots* which *are* designed to operate autonomously, isolated from human presence. This is achieved thanks to the sensors with which they are equipped, the limitation of the operating speed, or the force exerted, but also due to other properties related to their construction material, etc. At the same time, the cost remains low, making them affordable solutions for small and medium-sized enterprises.

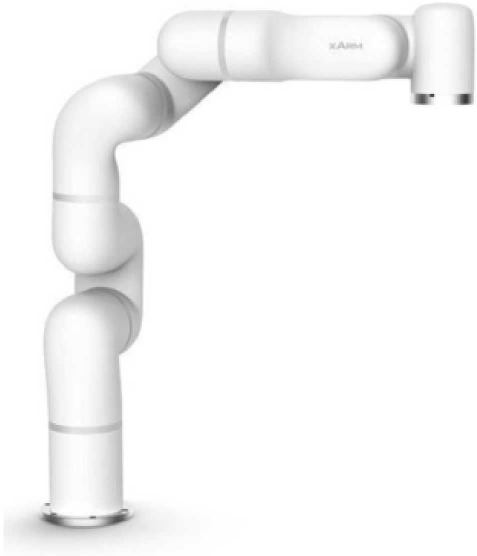


Figure 1 - Cobot xArm 7

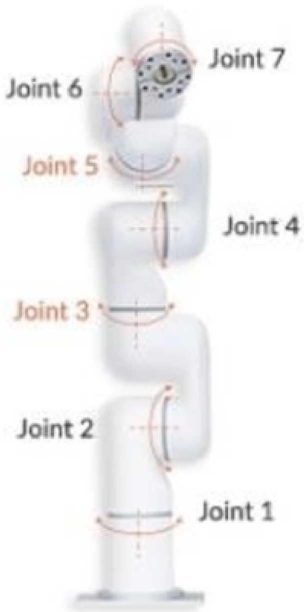


Figure 1 shows **xArm 7** which belongs to the above category *(cobot)*. It has a payload capacity of *3.5 kg*, its workspace extends to ***70 cm*** from its base, it achieves a maximum speed of *1 m/s* at the final action element, and it has ***7 degrees of freedom (DOFs)***, as indicated by its name. It is currently used not only in university centres for research purposes, but also in industry as an individual system in production lines. One of the main reasons why it seems to be very popular in the near future, apart from its low cost, is the possibility to use it in combination with the widely used software environment for the development of robotic applications ***ROS***. **Figure 2** shows the positions of the joints of the robotic arm, as well as the direction of rotation.

Figure 2 - Joints of xArm 7

B. Reference frames, kinematic analysis and dimensions

Figure 3 shows the reference frames in the initialization setup, according to which the kinematic analysis of the robotic arm *xArm 7* was performed.

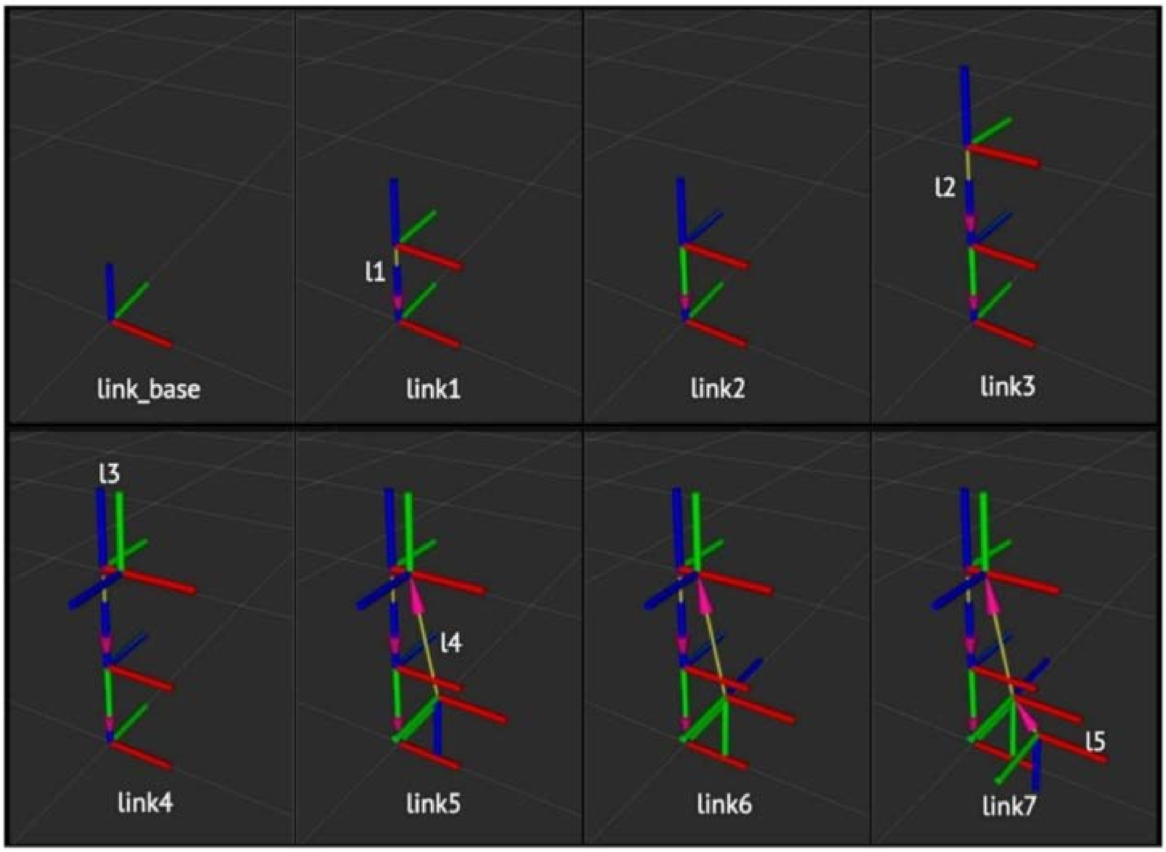


Figure 3 - Reference frames and links of xArm 7 (by Rviz)

$$A_1^0(\boldsymbol{q_1}) = Rot(z, q_1) \cdot Tra(z, l_1), \quad \boldsymbol{l_1} = 26.7cm$$

$$A_2^1(\boldsymbol{q_2}) = Rot\left(x, -\frac{\pi}{2}\right) \cdot Rot(z, q_2)$$

$$A_3^2(\boldsymbol{q_3}) = Rot\left(x, +\frac{\pi}{2}\right) \cdot Rot(z, q_3) \cdot Tra(z, l_2), \quad \boldsymbol{l_2} = 29.3cm$$

$$A_4^3(\boldsymbol{q_4}) = Rot\left(x, +\frac{\pi}{2}\right) \cdot Tra(x, l_3) \cdot Rot(z, q_4), \quad \boldsymbol{l_3} = 5.25cm$$

$$A_5^4(\boldsymbol{q_5}) = Rot\left(x, +\frac{\pi}{2}\right) \cdot Tra(x, l_4 \sin\theta_1) \cdot Rot(z, q_5) \cdot Tra(z, l_4 \cos\theta_1), \quad \begin{matrix} \boldsymbol{l_4} = 35.12\ cm \\ \boldsymbol{\theta_1} = 0.2225\ rad \end{matrix}$$

$$A_6^5(\boldsymbol{q_5}) = Rot\left(x, +\frac{\pi}{2}\right) \cdot Rot(z, q_6)$$

$$A_7^6(\boldsymbol{q_6}) = Rot\left(x, -\frac{\pi}{2}\right) \cdot Tra(x, l_5 \sin\theta_2) \cdot Rot(z, q_7) \cdot Tra(z, l_5 \cos\theta_2), \quad \begin{matrix} \boldsymbol{l_5} = 12.32\ cm \\ \boldsymbol{\theta_2} = 0.6646\ rad \end{matrix}$$

Γ. Running programs

The execution of robot control programs is done in a *ROS (Robot Operating System)* environment. The ROS application development environment software is a collection of tools, libraries and working conventions which has the main goal of simplifying the creation of complex and reliable robotic software, while also facilitating portability, interoperability and code sharing.

SIMULATION ENVIRONMENT

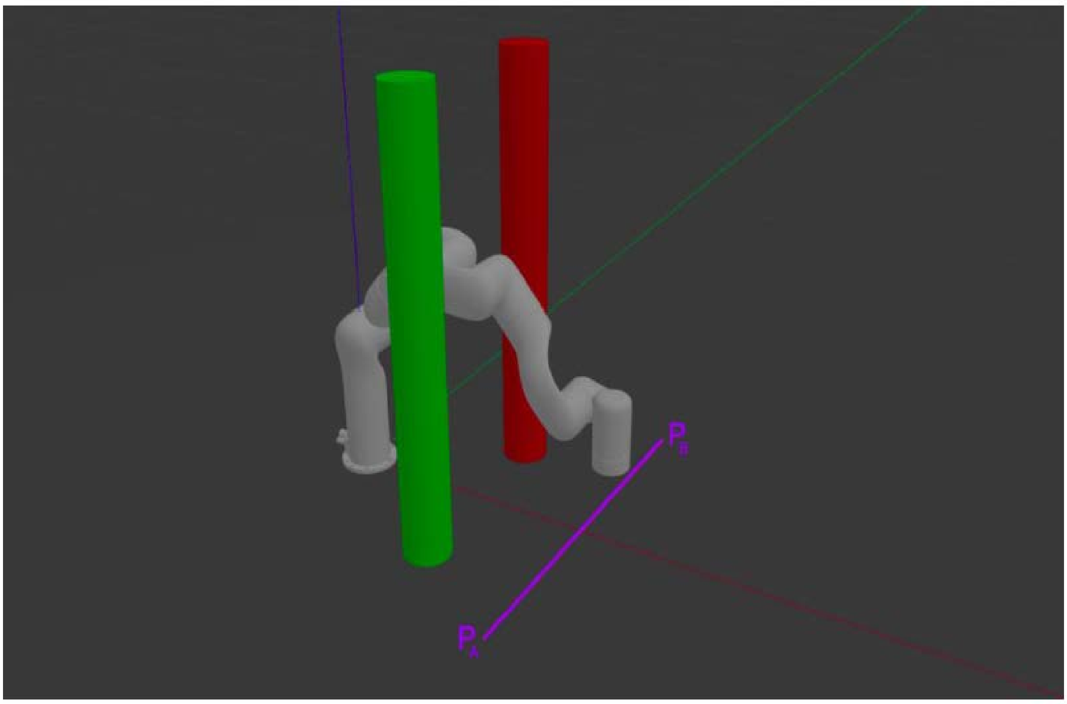
The programmatic implementation for the needs of the project can be done using the simulation environment ***"gazebo"***: http://gazebosim. org/

This simulation environment provides realistic implementations of a large number of robotic devices and can therefore be used (instead of or in addition/precursor to an experimental/laboratory implementation on the real robot) for the development and testing of the robot control programs and the completion of the objectives of the work described below. This type of simulator has the ability to work with *ROS.*

REQUIRED PROCESSES

The aim of this work is to implement an algorithm for tracking a trajectory from the final action element of the aforementioned robotic device *(xArm 7)* while avoiding obstacles, whose movement is interactively controlled by the user through the keyboard. More specifically, the arm's final action element is asked to perform rectilinear periodic motion between positions P_A and P_B (with position and NOT orientation control), as shown in **Figure 4** where a representation of the overall system in the *Gazebo* simulation environment is depicted. Moreover, the two end positions defining the linear motion segment of the final robot action element are symmetric with respect to the x-axis, are located on the line $\{x = 0.6043 | z = 0.11508\}$, while they are **40 cm apart**. At the same time, avoidance of obstacles is sought by considering their positions in space continuously known. At this point it should be noted that the diameter of the cross-section of each link of the arm is **12.6 cm**, while the two obstacle-cylinders are identical with a cross-sectional diameter of ***10 cm***, their centers are **40 cm apart,** and their motion in space is carried out on the plane

$x = 0.3.$



A. Theoretical Analysis

Describe in detail a way of applying the theoretical methodology of kinematic control of redundant degrees of freedom by splitting robotic work into sub-tasks, for the specific problem consisting of avoiding obstacles while performing the desired motion of the final action element in the plane, as described above.

B. Simulation

To implement this method for this problem, and to perform its kinematic simulation. In addition to the visualization of the robotic arm during the execution of the required task, record in time several other variables that you consider important to better capture, explain and evaluate the operation of the algorithm (such as distances from obstacles, position error, etc.), the annotation of which will be made within the report.