RCS - Laboratory work 5

The trajectory is defined as a path followed by a point subjected to time constraints. A robot trajectory is described by the trajectory of the robot gripper or of the manipulated object carried by the robot gripper.

The design of a robot trajectory is generally made in the Cartesian coordinate system. Anyway, the robot control has to be made in the robot coordinates – operational coordinates. The coordinate's transformation is made using the inverse kinematics model of the robot. In practice there are two main types of trajectories: PTP – point to point and CP – continuous path.

1.1. Point to point trajectories

The trajectory is saved as an array of points in the robot working space. These points will be followed by the robot gripper during motion but there is no definition of the space between the consecutive points. PTP motion is realized based on one of the following methods:

- ***Unconstrained motion*** – all the robot axes are moving in the same time with the highest velocity and acceleration. Function of the required motion distance of each joint, the arrival time of each joint can vary (the motion of one axis can end before the motion of another axis);
- ***Joint by joint motion*** – the robot joints are moving one after another. The resulting trajectory is different if the of the joint motion order is different. Generally, the trajectory generated by this type of motion is different from a straight line.
- ***Linear interpolated motion*** – is realized by moving the robot joints by pre-calculated velocities and accelerations, all joints starting and ending the motion in the same time. The resulting trajectory is a straight line, the shortest path from a start point to an end point.
- ***Polynomial trajectory*** – the trajectory is designed by using equations which determine an irregular path, but each point on the trajectory is saved and transformed into robot coordinates by the inverse kinematic model.

1.2. Continous path trajectories

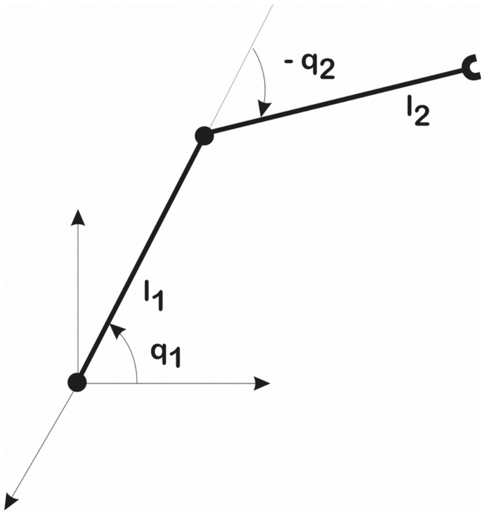
The continuous path trajectory is saved as a parametric function and not as an array of points. During the motion, there is a continuous control of the robot position. The used memory in this case is much lower, but the processing time is higher than in the case of PTP trajectories. One of the commonly used methods in continuous path trajectory generation is Taylor's Algorithm, defined by the following steps:

a) Apply the inverse kinematics for the start and target points;
b) Calculate the midpoint between start and target in robot coordinates and apply direct kinematics model on this point.
c) If the error in offset is higher that an imposed one:
   - Locate the midpoint in the Cartesian coordinates
   - Calculate the midpoints between start and 1st midpoint and between 1st midpoint and target point
   - Call recursively this algorithm for both the left side and right side of the midpoints

   Else go to (d).
d) Return the sequence of points.

1.3. Exercises

Consider a robotic structure with 2 Degrees-of-Freedom:



a) Verify the direct kinematics model:

$$\begin{cases} X &= 0 \\ Y &= l_2 \cos(q_1 + q_2) + l_1 \cos(q_1) \\ Z &= l_2 \sin(q_1 + q_2) + l_1 \sin(q_1) \end{cases}$$

b) Verify the inverse kinematics model:

$$\begin{cases} a &= \frac{Y^2+Z^2-l_1^2-l_2^2}{2l_1l_2} \\ q_2 &= \pm \arctan \frac{\sqrt{1-a^2}}{a} \\ q_1 &= \arctan \frac{Z}{Y} - \arctan \frac{l_2 \sin(q_2)}{l_1+l_2\cos(q_2)} \end{cases}$$

c) Considering the following Start and Target points, and based on the verified direct and inverse kinematics models, make a script which generates the ***joint by joint trajectory*** and the ***linear interpolated trajectory*** between the Start and Target points:

$$a)\ \begin{cases} S(0, 30, 30\sqrt{3}) \\ T(0, 20\sqrt{3} - 10, 10\sqrt{3} + 20 \end{cases} \qquad b)\ \begin{cases} S(0, 10, 40 + 10\sqrt{3}) \\ T(0, 40, 20\sqrt{3} \end{cases}$$