Modern Robotics practice

Argyrios Kokkinis
s2252406
a.kokkinis@student.utwente.nl

June 2019

1 Introduction

In this practical exercise we had to work on the kinematics of a 3-DOF manipulator arm and control its position.As a first step we had to define a reference configuration $q_i = 0$ for the arm and calculate the homogeneous matrix H_{ee}^0 from the end effector to our reference frame Ψ_0. Secondly. we had to find the Jacobian matrix for any configuration of the joints.The last step of the exercise was to apply a control law using our previous findings and test it in the actual setup.

2 Forward Kinematics

The easiest reference configuration was to have all 3 bodies of the arm aligned vertically on the z-axis. Doing so it was easy to find the homogeneous matrices from the one frame to the other on $q_i = 0$. $H_{ee=3}^0 = H_1^0 H_2^1 H_3^2$. Knowing that

$$H_n^{n-1}(0) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_n \\ 0 & 0 & 0 & 1 \end{pmatrix}, L_n = length_n$$

Using the chain rule it is easy to find out that

$$H_{ee}^0(0) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_1 + L_2 + L_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Also at $q_i = 0$ is easy to find the unit twists.We have 3 rotations one along z-axis and two along x-axis

$$\hat{T_1}^{0,0} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \hat{T_2}^{0,1} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \hat{T_3}^{0,2} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

While the twist $\hat{T_1}^{0,0}$ is constant for any configuration of the joints the other twists are not.However,in the Brockett's formula the twists at the reference configuration are the ones that are used.
From the Brockett's formula we have that:

$$H_{ee}^0(q) = e^{\tilde{\hat{T_1}}^{0,0} q_1} e^{\tilde{\hat{T_2}}^{0,1} q_2} e^{\tilde{\hat{T_3}}^{0,2} q_3} H_{ee}^0(0)$$

Using this expression we can move from the end-effector's frame to our reference frame at any configuration of the joints by knowing only the q_i

The Jacobian is a 6 by 3 matrix because we are having a 3-DOF robot

$$J(q) = \begin{pmatrix} T_1 & T_2 & T_3 \end{pmatrix}$$

with $T_1 = \hat{T}_1{}^{0,0}, T_2 = \hat{T}_2{}^{0,1}, T_3 = \hat{T}_3{}^{0,2}$ for any q_i. So, we have to calculate $\hat{T}_2{}^{0,1} = Ad_{H_1^0}\hat{T}_2{}^{1,1}$ and $\hat{T}_3{}^{0,2} = Ad_{H_2^0}\hat{T}_3{}^{2,2}$ for every configuration.

3 Position Control

The velocity at the end-effector can be found $\dot{p}_{ee} = K_u(p_{sp} - p_{ee})$ with p_{sp} being the desired position of the end-effector and p_{ee} its current position.The gain K_u was set to be 10.

We had to create a fourth frame Ψ_4 located at the same position with Ψ_3 having the same orientation with the base frame Ψ_0 and use it as a reference to express the velocities of the end-effector.We can now express the Jacobian in that frame and calculate $T_3{}^{4,0}$. Knowing that the frame Ψ_4 doesn't rotate with respect to Ψ_0 we can find that.

$$H_0^4 = \begin{pmatrix} I_{3x3} & -p_3^0 \\ 0_3 & 1 \end{pmatrix}$$

Thus, the new Jacobian $\grave{J} = Ad_{H_0^4}J(q)$ maps the twists expressed in frame Ψ_0 to Ψ_4 at any change of the joint configuration $\dot{q}$. Since we are interested only in transnational velocities we keep only the bottom three elements of all twists in the Jacobian and we define this new 3x3 matrix as J_u.

Calculating the pseudo-inverse of J_u and knowing that $u_3{}^{4,0} = \dot{p}_{ee}$ we can find that $\dot{q}_{set} = J_u^{\dagger}\dot{p}_{ee}$ which are the velocities that we send to the robot for following the desired trajectory.

4 Results

The above analysis was implemented in MATLAB with the desired trajectory of the end-effector being a 8-figure in the xz-plane.For calculating the homogeneous matrices for any q_i I used the expm function as provided from MATLAB.The calculation of the pseudo-inverse was also done using the pinv function that returns the Moore-Penrose pseudo-inverse.

At the `robot_skeleton_control.m` file some extra variables were defined that save the desired trajectory of the robot on the xz-plane, the calculated trajectory using the above analysis and the real robot's trajectory.Plotting all of them at the end. As running the program and sending the velocities to the robot has real-time calculations , any lag at the computer may cause different (dangerous) trajectories.

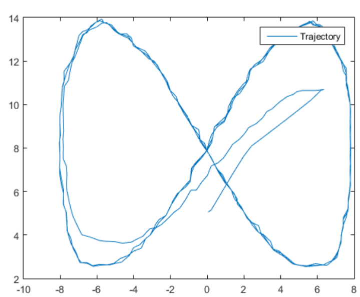

Figure 1: Robot's trajectory

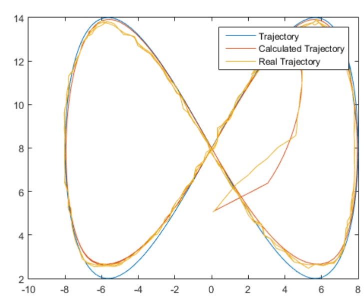

Figure 2: All three trajectories

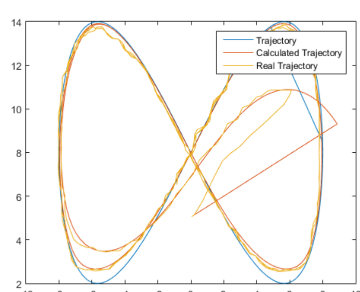

Figure 3: All three trajectories with some lag at the beginning