Lab Assignment № 2

Alexandra Rebecca Marie Zimmer
Adina Zell
Ali Alouane

06/12/2020

Task A: Forward Kinematics

1.

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	q_1
2	$+180°$	L_1	0	q_2
3	0	L_2	0	q_3
$4(E)$	0	L_3	0	0

Table 1: DH parameters

2.

The frame to frame transformation can be calculated according to the following formula :

$$ {}^{i-1}_{i}T = R_X(\alpha_{i-1})D_X(a_{i-1})R_Z(\theta_i)D_Z(d_i) = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Therefore with the help of the DH parameters in 1, we calculate the Homogeneous Transformation (HT) between adjacent links ${}^{i-1}_{i}T$:

$$ {}^{0}_{1}T = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} ; {}^{1}_{2}T = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & L_1 \\ -s\theta_2 & -c\theta_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} ; {}^{2}_{3}T = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & L_2 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{3}_{E}T = \begin{bmatrix} 1 & 0 & 0 & L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

In order to calculate the forward kinematics ${}^0_ET$, we apply the concatenation rules between each adjacent links and the trigonometric identities from Craig book:

$$
\begin{aligned}
{}^0_ET &= {}^0_1T\,{}^1_2T\,{}^2_3T\,{}^3_ET \\
&= \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & L_1 \\ -s\theta_2 & -c\theta_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & L_2 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} 1 & 0 & 0 & L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} c_1c_2+s_1s_2 & -c_1s_2+s_1c_2 & 0 & c_1L_1 \\ s_1c_2-c_1s_2 & -s_1s_2-c_1c_2 & 0 & s_1L_1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} c_3 & -c_3 & 0 & c_3L_3+L_2 \\ 0 & c_3 & 0 & s_3L_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} c_{21} & s_{21} & 0 & c_1L_1 \\ s_{21} & -c_{21} & 0 & s_1L_1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} c_3 & -c_3 & 0 & c_3L_3+L_2 \\ 0 & c_3 & 0 & s_3L_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} c_{21}c_3 & -c_{21}c_3+s_{21}c_3 & 0 & c_{21}(c_3L_3+L_2)+s_{21}s_3L_3+c_1L_1 \\ s_{21}c_3 & -c_3(s_{21}+c_{21}) & 0 & s_{21}(c_3L_3+L_2)-c_{21}s_3L_3+s_1L_1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

with $c_{21} = c_1c_2 + s_1s_2 = cos(\theta_1 - \theta_2)$ and $s_{21} = s_1c_2 - c_1s_2 = sin(\theta_1 - \theta_2)$

3.

The forward kinematic can be describe $\mathbf{x} = F(\mathbf{q}) = {}^0_ET(\mathbf{q})$, where $\mathbf{x} = \begin{pmatrix} x \\ y \\ \alpha \end{pmatrix}$ contains the position and orientation of the End-Effector in the base frame O.

Using the HT matrix ${}^0_ET(\mathbf{q}) = \left(\begin{array}{ccc|c} & R & & \vec{t} \\ \hline 0 & 0 & 0 & 1 \end{array}\right)$ we can calculate the forward kinematic function by determine the position from the vector $\vec{\mathbf{t}}$ and the orientation from the rotation matrix **R** :

$$
f(\mathbf{q}) =) \begin{pmatrix} c_{21}(c_3L_3+L_2)+s_{21}s_3L_3+c_1L_1 \\ s_{21}(c_3L_3+L_2)-c_{21}s_3L_3+s_1L_1 \\ -atan2(\frac{c_{21}}{s_{21}}) \end{pmatrix}
$$

4.

The Jacobian matrix of the forward kinematic $F(\mathbf{q})$ is derived as follow:

- $\frac{\partial f_1}{\partial \theta_1} = -s_{21}(c_3 L_3 + L_2) + c_{21} s_3 L_3 - s_1 L_1$
- $\frac{\partial f_1}{\partial \theta_2} = s_{21}(c_3 L_3 + L_2) - c_{21} s_3 L_3$
- $\frac{\partial f_1}{\partial \theta_3} = -s_3 c_{21} L_3 + c_3 s_{21} L_3$
- $\frac{\partial f_2}{\partial \theta_1} = c_{21}(c_3 L_3 + L_2) + s_{21} s_3 L_3 - c_1 L_1$
- $\frac{\partial f_2}{\partial \theta_2} = -c_{21}(c_3 L_3 + L_2) - s_{21} s_3 L_3$
- $\frac{\partial f_2}{\partial \theta_3} = -s_3 c_{21} L_3 - c_3 c_{21} L_3$

- $\frac{\partial f_3}{\partial \theta_1} = \frac{1}{1+\frac{c_{21}}{s_{21}}} + \$\partial f_{3\,\overline{\partial \theta_2 = 1}}$
- $\frac{\partial f_3}{\partial \theta_3} = 0$

$$J(\mathbf{q}) = \frac{\partial F(\mathbf{q})}{\partial \mathbf{q}} = \begin{bmatrix} -s_{21}(c_3 L_3 + L_2) + c_{21} s_3 L_3 - s_1 L_1 & s_{21}(c_3 L_3 + L_2) - c_{21} s_3 L_3 & -s_3 c_{21} L_3 + c_3 s_{21} L_3 \\ c_{21}(c_3 L_3 + L_2) + s_{21} s_3 L_3 - c_1 L_1 & -c_{21}(c_3 L_3 + L_2) - s_{21} s_3 L_3 & -s_3 c_{21} L_3 - c_3 c_{21} L_3 \\ -1 & 1 & 0 \end{bmatrix}$$

5.

a) & b)

i.) For the joint configuration $q_a = (0, 0, -\frac{\pi}{2})^\top$.

we get the jacobian matrix $J = \begin{bmatrix} -L_3 & L_3 & L_3 \\ L_2 - L_1 & -L_2 & L_3 \\ -1 & 1 & 0 \end{bmatrix}$

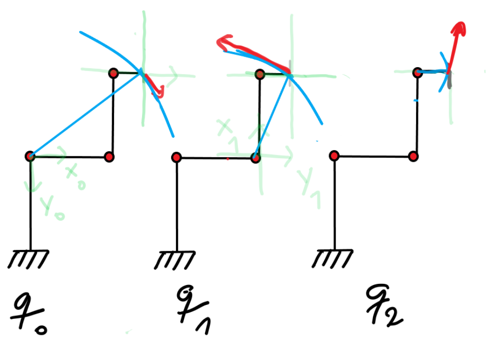


Figure 1: joint configuration

ii.) For the joint configuration $q_a = (-\frac{\pi}{2}, \frac{\pi}{2}, 0.1)^\top$.
we get the jacobian matrix
$$J = \begin{bmatrix} 0.99L_3 - L_1 & 0.099L_3 & -0.99L_3 \\ -0.99L_3 - L_2 & 0.99L_3 + L_2 & 1.09L_3 \\ -1 & 1 & 0 \end{bmatrix}$$
The figure 1 shows that the robot is close to a singularity for the joint 2 and 3.

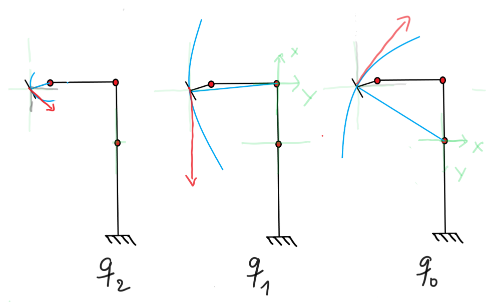


Figure 2

iii.) For the joint configuration $q_a = (-\frac{\pi}{2}, -\frac{\pi}{2}, 0)^\top$.

we get the jacobian matrix $J = \begin{bmatrix} L_1 & 0 & 0 \\ L_3 + L_2 & -(L_3 + L_2) & -L_3 \\ -1 & 1 & 0 \end{bmatrix}$

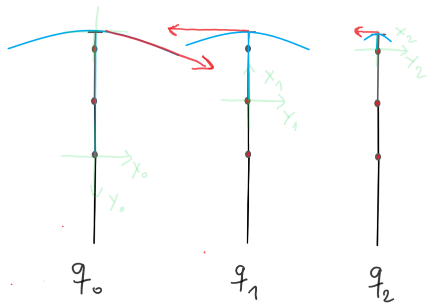


Figure 3

the jacobian matrix of the given joint configuration $q_a = (-\frac{\pi}{2}, -\frac{\pi}{2}, 0)^\top$ is not full rank because $det J(\mathbf{q}) = 0$. This implies that the puma robot is in singularity.

Task B: Trajectory Generation in Joint Space

1.

a)

From the exercise sheet, there is given:

- $q_a = (0, 0, 0)^\top$
- $q_b = (\frac{-\pi}{4}, \frac{\pi}{2}, 0)^\top$
- $q_c = (\frac{-\pi}{2}, \frac{\pi}{4}, 0)^\top$

We derived these spline functions from the tutorial:

- $q_{ab}(t) = a_1 + a_2(t - t_{\text{start}}) + a_3(t - t_{\text{start}})^2 + a_4(t - t_{\text{start}})^2$
- $q_{bc}(t) = b_1 + b_2(t - t_{\text{start}} - t_{\text{via}}) + b_3(t - t_{\text{start}} - t_{\text{via}})^2 + b_4(t - t_{\text{start}} - t_{\text{via}})^2$

where $q_{xy}(t)$ is the 3-dimensional cubic spline for the trajectory from q_x to q_y over time t and a_i, b_i are the 3-dimensional spline component vectors.

The desired conditions for the splines are:

- $q_{ab}(t_{\text{start}}) = q_a$
- $\dot{q}_{ab}(t_{\text{start}}) = \dot{q}_a = (0,0,0)^\top$
- $q_{ab}(t_{\text{start}} + t_{\text{via}}) = q_b$
- $\dot{q}_{ab}(t_{\text{start}} + t_{\text{via}}) = \dot{q}_b = v_{\text{via}}$
- $q_{bc}(t_{\text{start}} + t_{\text{via}}) = q_b$
- $\dot{q}_{bc}(t_{\text{start}} + t_{\text{via}}) = \dot{q}_b = v_{\text{via}}$
- $q_{bc}(t_{\text{start}} + t_{\text{f}}) = q_c$
- $\dot{q}_{bc}(t_{\text{start}} + t_{\text{f}}) = \dot{q}_c = (0,0,0)^\top$

where $t_{\text{via}} = 2.5$, $t_{\text{f}} = 5$ and v_{via} is the desired velocity vector at the via point, which is calculated using the heuristic from the tutorial. Since there is a sign change (as defined in the tutorial) in the second and third vector component of the configuration series q_a, q_b, q_c, but not in the first component, v_{via} is computed as:

- $v_{\text{via}}[0] = \frac{s_{ab}+s_{bc}}{2} = \frac{\frac{-\pi}{10}+\frac{-\pi}{10}}{2} = \frac{-\pi}{10}$, where s_{xy} is the slope between q_x and q_y:

  $$s_{ab} = \frac{q_b[0]-q_a[0]}{t_{\text{via}}-t_{\text{start}}} = \frac{-\pi/4}{2.5} = \frac{-\pi}{10}$$
  $$s_{bc} = \frac{q_c[0]-q_b[0]}{t_{\text{f}}-t_{\text{via}}} = \frac{-\pi/4}{2.5} = \frac{-\pi}{10}$$

- $v_{\text{via}}[1] = 0$
- $v_{\text{via}}[2] = 0$

  $\Rightarrow v_{\text{via}} = (\frac{-\pi}{10}, 0, 0)^\top$

We derived the following spline components a_i, b_i like in the tutorial, but as vectors, because we calculate 3-dimensional splines. By this, we get one 1-dimensional spline for every joint of the robot:

- $a_1 = q_a = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$
- $a_2 = \dot{q}_a = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$
- $a_3 = \frac{3}{t_{\text{via}}^2} \cdot (q_b - q_a) - \frac{2}{t_{\text{via}}} \cdot \dot{q}_a - \frac{1}{t_{\text{via}}} \cdot v_{\text{via}} = \begin{pmatrix} -2\pi/25 \\ 6\pi/25 \\ 0 \end{pmatrix}$
- $a_4 = \frac{-2}{t_{\text{via}}^3} \cdot (q_b - q_a) + \frac{1}{t_{\text{via}}^2} \cdot (\dot{q}_a + v_{\text{via}}) = \begin{pmatrix} 2\pi/125 \\ -8\pi/125 \\ 0 \end{pmatrix}$

- $b_1 = q_b = \begin{pmatrix} -\pi/4 \\ \pi/2 \\ 0 \end{pmatrix}$
- $b_2 = v_{\text{via}} = \begin{pmatrix} -\pi/10 \\ 0 \\ 0 \end{pmatrix}$
- $b_3 = \frac{3}{(t_{\text{f}}-t_{\text{via}})^2} \cdot (q_c - q_b) - \frac{2}{t_{\text{f}}-t_{\text{via}}} \cdot v_{\text{via}} - \frac{1}{t_{\text{f}}-t_{\text{via}}} \cdot \dot{q}_c$

  $= \begin{pmatrix} -1\pi/25 \\ -3\pi/25 \\ 0 \end{pmatrix}$
- $b_4 = \frac{-2}{(t_{\text{f}}-t_{\text{via}})^3} \cdot (q_c - q_b) + \frac{1}{(t_{\text{f}}-t_{\text{via}})^2} \cdot (v_{\text{via}} + \dot{q}_c)$

  $= \begin{pmatrix} 2\pi/125 \\ 4\pi/125 \\ 0 \end{pmatrix}$

b)

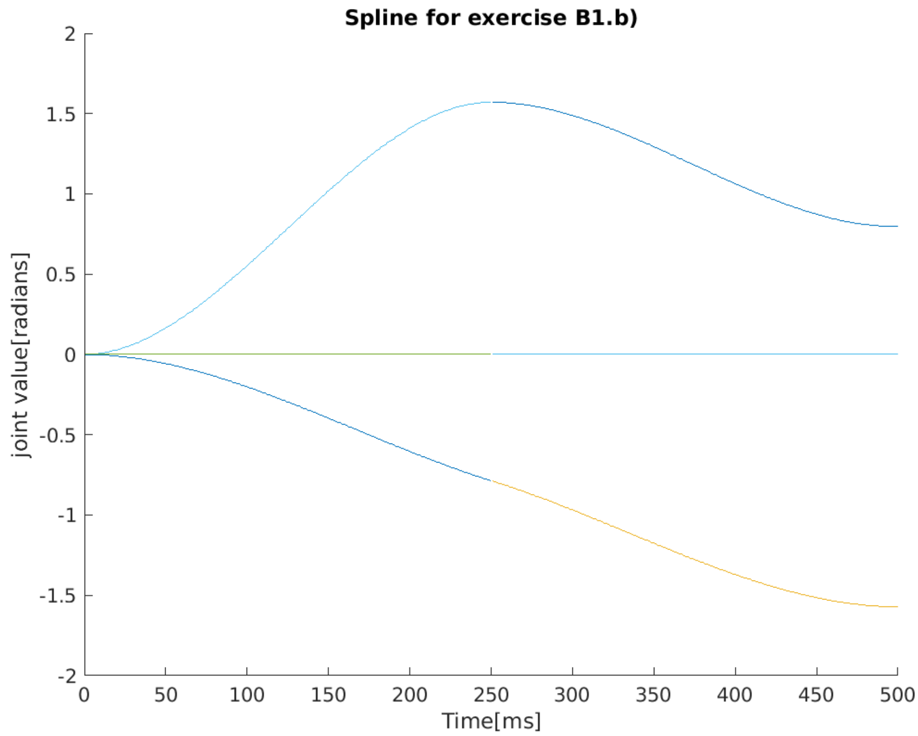


Figure 4: Diagram of the joint angles with respect to time.

In Fig. 4 the two cubic splines for the desired trajectory are plotted over time.

2.

- in: current and desired joint configuration (q_0, q_f)
- out: cubic spline parameters $(a_0, a_1, a_2, a_3, t_0, t_f)$

a)

Here, initNjtrackControl() is implemented. It takes current and desired joint configuration (q_0, q_f) as input and outputs for each joint $a_0, a_1, a_2, a_3, t_0, t_f$.

Therefore we solve the cubic spline equation, with new conditions:

$$q(t_0) = q_0$$

$$\dot{q}(t_0) = (0, 0, 0)^\top$$

$$q(t_f) = q_f$$

$$\dot{q}(t_f) = (0, 0, 0)^\top$$

speed and acceleration constraints

$$-\dot{q}_{\text{max}_i} \leq \dot{q}_i \leq \dot{q}_{\text{max}_i}$$

$$-\ddot{q}_{\text{max}_i} \leq \ddot{q}_i \leq \ddot{q}_{\text{max}_i}$$

information about where speed and/or acceleration is the highest

$$\dot{q}(t_f/2) = \dot{q}_{\text{max}}, \text{ or } = -\dot{q}_{\text{max}}$$
$$\ddot{q}(t_f) = -\ddot{q}_{\text{max}}, \text{ or } = \ddot{q}_{\text{max}}$$

Solving the last two equations leads to following values for t_f:

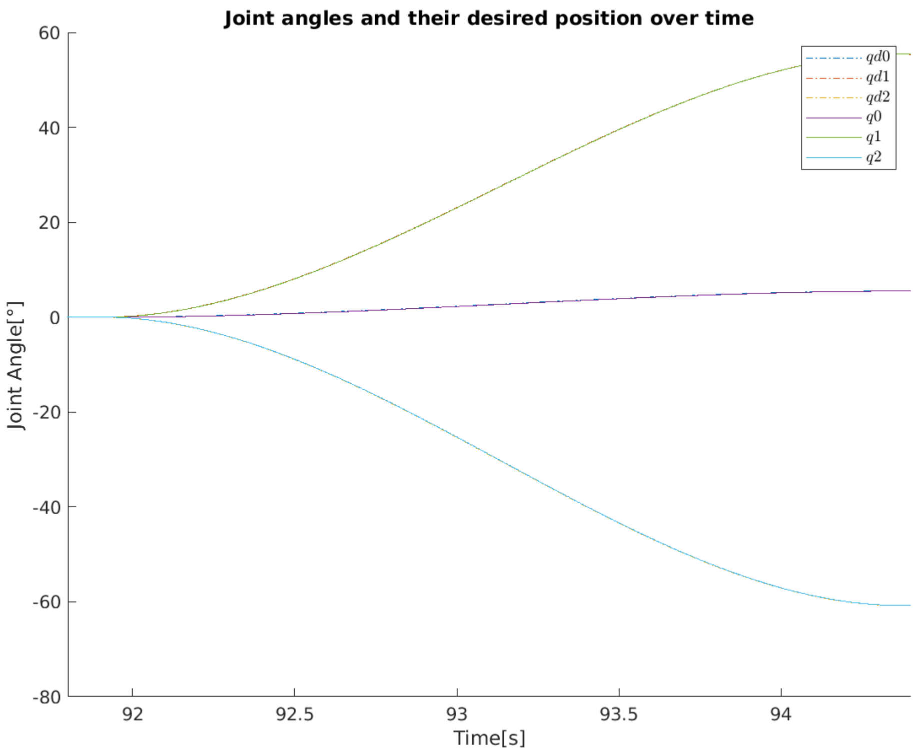
- $t_f = [-]\frac{3(q_f - q_0)}{2 \cdot \dot{q}_{\text{max}}}$
- $t_f = \sqrt{\frac{[-]6(q_f - q_0)}{\ddot{q}_{\text{max}}}}$

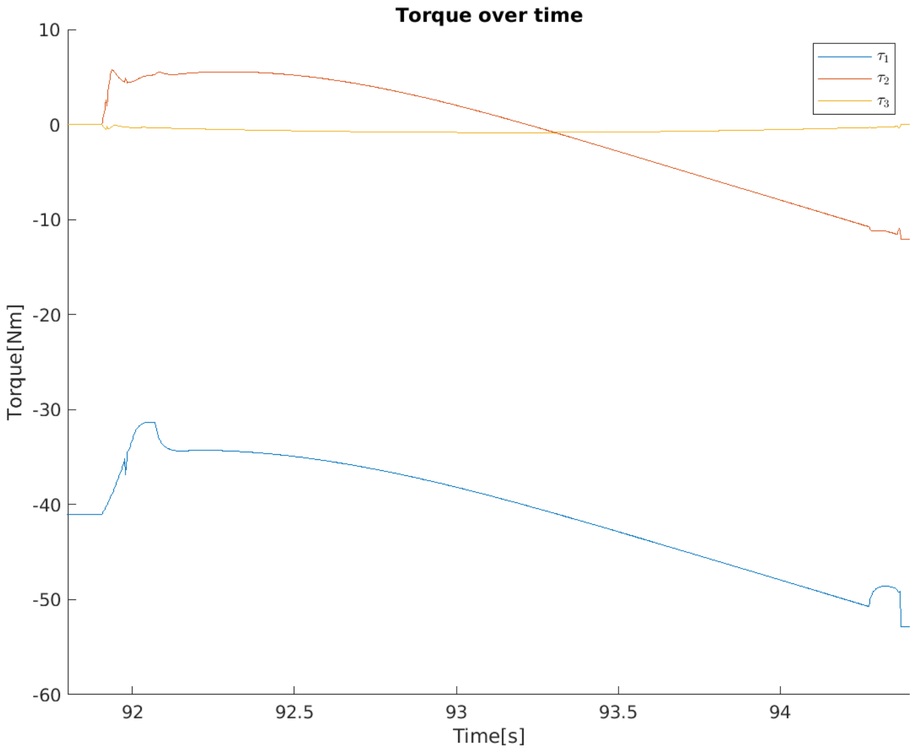
We take the maximum of all the six values, because t_f should be chosen such, that both conditions are fulfilled. The minus is in brackets, because depending on the direction of joint movement it has to be included or not.

b)

q_d and the derivative of q_d are calculated, based on the spline parameters from part a).

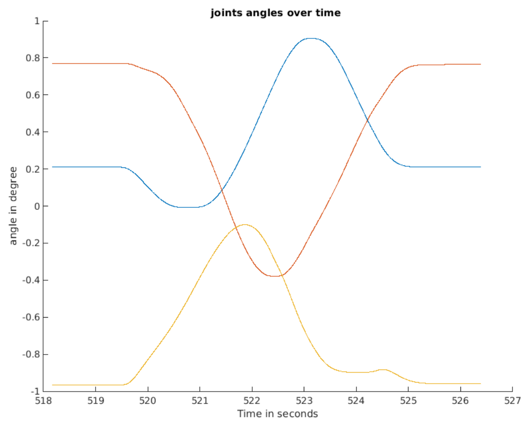
c)

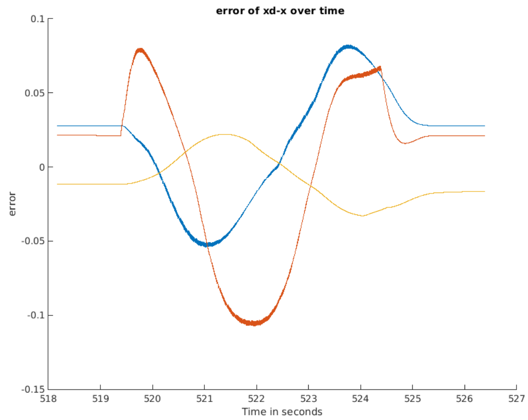


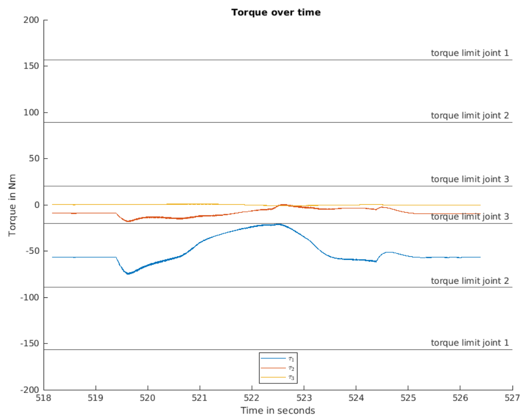


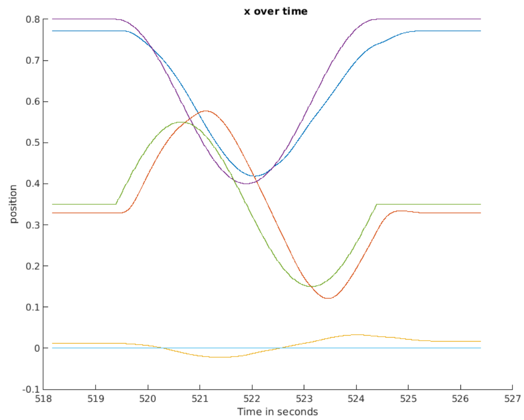
Task C: Operational Space Control

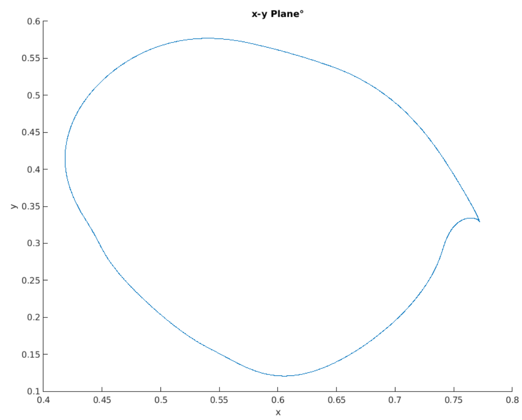
1.











If we increase the kp and kv the error gets smaller and if the kp and kv are too lower the error increases.

P3)

In Task 3 we should create a trajectory which stops after 3 circles and uses the parabolic blend.
We divided the trajectory in 3 Parts:
1) the accelaration until we reach the desired velocity of 2Pi/5s
2) the constant velococity trajectory with the velocity of 2Pi/5s
3) the brake part where we want to reach a velocity of 0 at the end

The 1) we can describe with the following formula

$$u(t) = 1/2 * u'' * t^2 + u_0$$

with u” is accelaration

The 3) we can describe with basically the same just with -u”

$$u(t) = 1/2 * -u'' * t^2 + u_2$$

The 2) we can describe with simple linear motion

$$u(t) = v * t + u_1$$

with v is the velocity

The question is at what time we should start with calculating the trajectory to 2) and 3).

For part 2) it is easy: with the given accelaration of u” = 2Pi/5s and the desired velocity of 2Pi/5s we need 5 sek to reach the desired velocitiy.The angle u1 we can compute with the fomular given for trajectory 1):

$$u(t) = 1/2 * 2Pi/25s^2 * 5s = Pi/5$$

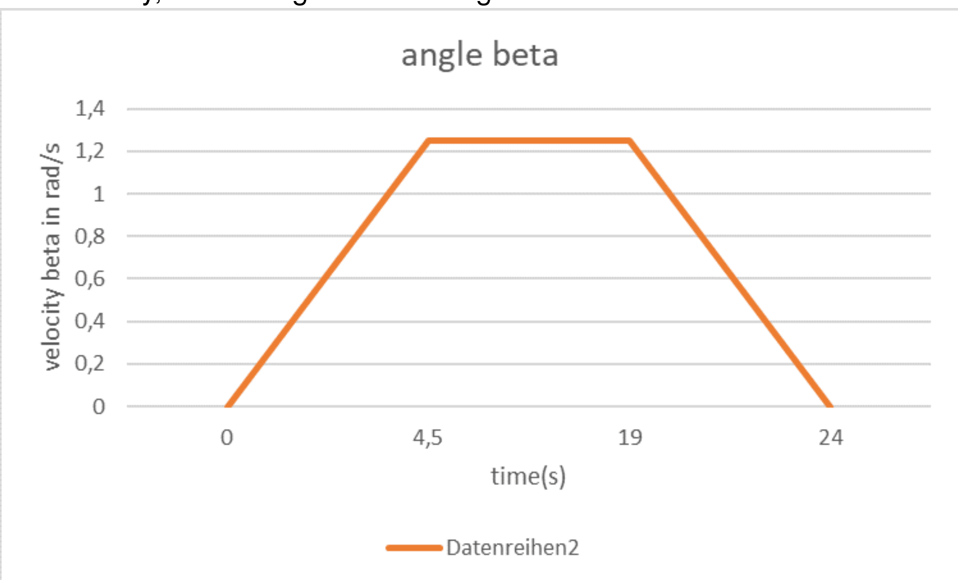
Then we have the following formular for part 2):

$$u(t) = 2 * Pi/5 * t + Pi/5$$

The last question is now at what time t we have to change the trajectory generation to 3). We know from trajectory 1) that in 5 seconds we reach the angle of Pi/5 rad, so we need to find the time trajectory 2 needs for 6Pi- 2*Pi/5. (Pi/5 = end angle of the blend).

$$u(t) = 2 * Pi/5 * t = 6Pi - 2 * Pi/5 => t = 14$$

Unfortunatly, we didn’t get the code right to draw 3 circles. The last half is missing.



Task C: Operational Space Control

1.

Student Name	A2	A3	A4	B2	C1	C2	C3	C4
Alexandra Zimmer	-	-	-	x	x	-	-	-
Adina Zell	-	-	-	-	x	x	x	x
Ali Alouane	x	x	x	-	x	x	x	x