Report Assignment I - Kinematics I

Nils Becker

Fundamentals of Robotics - FS 2021

-

November 6^{th} 2021

1 Robot description

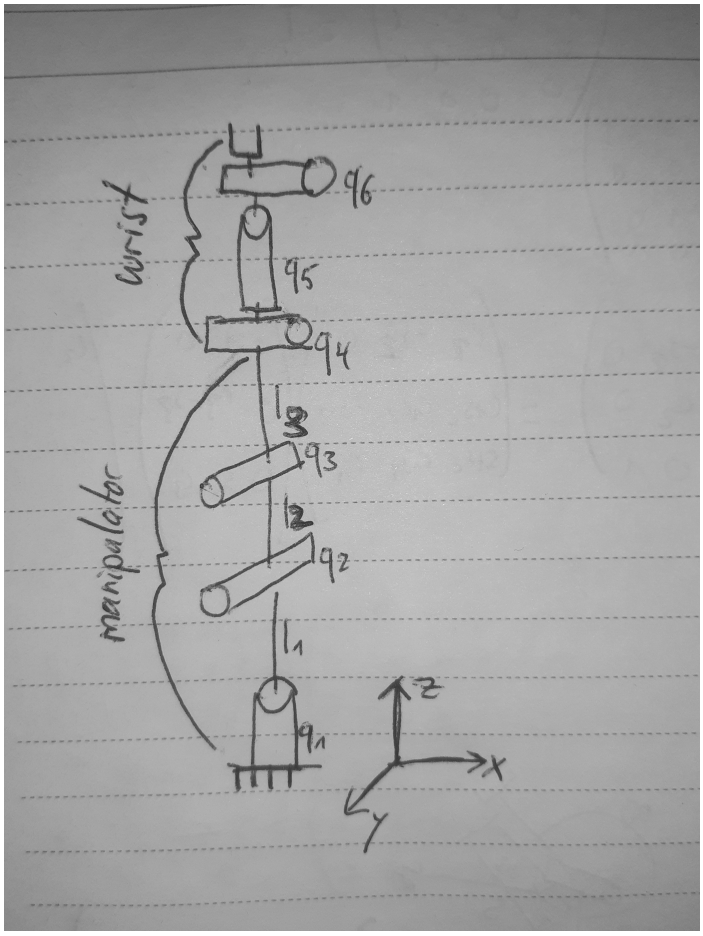
1.1 Selected Robot

The chosen robot is a Antropomorphic robot which provides 6 degrees of freedom with a spherical wrist.

1.2 Wrist configuration

The wrist of the robot's arm follows rotation around XZX axes. As the wrist is responible for the rotation of the endeffector, the selected configuration must ensure, that all rotations can be achieved. Rotations about first the X axis, then about the new Z = Z' axis and then about the new X = X" axis satisfies this requirement.

1.3 Model



2 Kinematics

2.1 Forward Kinematics

Forward Kinematics is solved by using Rotation- and Translation matrices directly from the zero configuration.

$$\mathrm{T} = \mathrm{T}_{Base}\mathrm{R}_Z(\mathrm{q1})\mathrm{T}_Z(\mathrm{l1})\mathrm{R}_Y(\mathrm{q2})\mathrm{T}_Z(\mathrm{l2})\mathrm{R}_Y(\mathrm{q3})\mathrm{T}_Z(\mathrm{l3})\mathrm{R}_X(\mathrm{q4})\mathrm{R}_X(\mathrm{q5})\mathrm{R}_X(\mathrm{q6})\mathrm{T}_{Tool} \tag{1}$$

2.2 Inverse Kinematics

To compute all possible input angles given the endeffector pose, the following procedure is applied.

Let's say the endeffector position is given by a matrix M of form:

$$\begin{pmatrix} nx & sx & ax & px \\ ny & sy & ay & py \\ nz & sz & az & pz \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2.2.1 Finding q1

To find q1, we can compute atan2(py, px). This gives us one solution. To find the second solution, we can add π to the first solution.

$$q1_1 = atan2(py, px) \tag{2}$$

$$q1_2 = atan2(py, px) + \pi \tag{3}$$

2.2.2 Finding q2

$$q2_1 = \pi/2 - \alpha - \beta \tag{4}$$

$$q2_2 = \pi/2 - \alpha + \beta \tag{5}$$

To compute α and β we use the following equations:

$$\alpha = atan2((pz - l1)^2, \sqrt{(px^2 + py^2)}) \tag{6}$$

$$\beta = acos((s^2 + l2^2 + l3^2)/(2 * l2 * s^2) \tag{7}$$

$$s = (pz - l1)^2 + \sqrt{(px^2 + py^2)} \tag{8}$$

2.2.3 Finding q3

$$q3_1 = \pi - \gamma \tag{9}$$

$$q3_2 = \pi + \gamma \tag{10}$$

To compute γ we use the following equation:

$$\gamma = acos((l2^2 + l3^2 - s^2)/(2 * l2 * l3)) \tag{11}$$

2.2.4 Finding q4, q5, q6

Note that 3x3 rotaion matrices will be used.

It applies:

$$^3R_6 =^0 R_3^\top *^0 R_6 \tag{12}$$

The matrix $^3R_6$ will be of form:

$$\begin{pmatrix} c5 & -s5c6 & s5s6 \\ c4s5 & c4c5c6 & -s4c6 - c4c5s6 \\ s4s5 & c4s6 + s4c5c6 & c4c6 - s4c5s6 \end{pmatrix}$$

While the values will be determined by the used q1-q3.

The values q5 can thus be computed by:

$$q5 = acos(c5) \tag{13}$$

The values of q4 will be:

$$q4 = atan2(s4s5, c4s5) \tag{14}$$

And the values of q6 will be:

$$q6 = atan2(s5s6, -s5c6) \tag{15}$$

3 Code

First of all, the my code computes the value of q2 in Inverse Kinematics wrong. Therefore q4, q5, q6 will be computed wrong, because their calculation is based on q2. If you manually put the correct value for q2 in line 162 of the code, q4, q5 and q6 will be computed correctly.

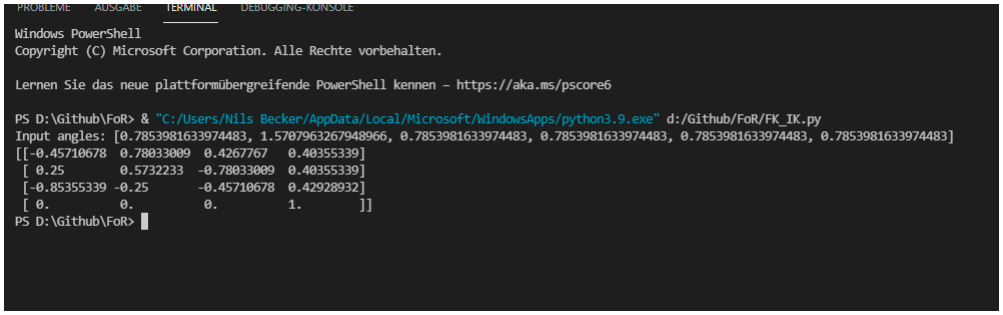
3.1 Code Output

Let q1 - q6 be: $q1 = \pi/4$, $q2 = \pi/2$, $q3 = \pi/4$, $q4 = \pi/4$, $q5 = \pi/4$, $q6 = \pi/4$

And l1 - l3: $l1 = 0.5$, $l2 = 0.5$, $l3 = 0.1$

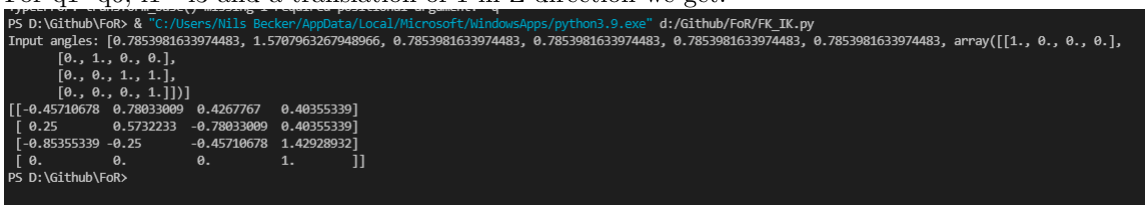
3.1.1 Forward Kinematics

For q - q6 and l1 - l3 we get:



3.1.2 Transform Base

For q1 -q6, l1 - l3 and a translation of 1 in Z direction we get:



3.1.3 Inverse Kinematics

For the matrix calculated in Forward Kinematics we get the following solution if we use the (wrongly calculated) q2:

```
[[-0.45710678  0.78033009  0.4267767   0.40355339]
 [ 0.25        0.5732233  -0.78033009  0.40355339]
 [-0.85355339 -0.25       -0.45710678  0.42928932]
 [ 0.          0.          0.          1.        ]]
r = 0.5707106781308576
pz2 = -0.07071067811999998
s = 0.5750744978981189
alpha = 0.00876078242843278
beta = 0.12327112425372624
gamma = 2.356194489928527
-1.178097244964263
q1 = [0.7853981633974483, -2.356194490192345]
q2 = [1.4387644201127374, 1.68530666862019]
q3 = [0.7853981636612661, -2.3561944899285265]
q4 = [0.67785338124028 68, -0.67785338124028 68, 0.85721708503 4633, -0.85721708503 4633, 2.237709899762092, -2.237709899762092, 2.1529395931752275, -2.1529395931752275]
q5 = [0.6969916525315746, 2.444601001058219, 0.869320567047827, 2.272272086541966, 0.9412997993919932, 2.2002928541977997, 1.1418297567005409, 1.9997628968892522]
q6 = [0.9309302373720522, -2.210662416217741, 0.6794332616083745, -2.46215939 19814186, 2.8775042530161974, -0.2640884005735957, 3.0442786250132863, -0.09731402857650681]
[[0.7853981633974483, -2.356194490192345], [1.4387644201127374, 1.68530666862019], [0.7853981636612661, -2.3561944899285265], [0.67785338124028 68, -0.67785338124028 68, 0.8
57217085034633, -0.85721708503 4633, 2.237709899762092, -2.237709899762092, 2.1529395931752275, -2.1529395931752275], [0.6969916525315746, 2.444601001058219, 0.869320567047
827, 2.272272086541966, 0.9412997993919932, 2.2002928541977997, 1.1418297567005409, 1.9997628968892522], [0.9309302373720522, -2.210662416217741, 0.6794332616083745, -2.46
21593919814186, 2.8775042530161974, -0.2640884005735957, 3.0442786250132863, -0.09731402857650681]]
PS D:\Github\FoR>
```

And for a corrected q2:

```
[[-0.45710678  0.78033009  0.4267767   0.40355339]
 [ 0.25        0.5732233  -0.78033009  0.40355339]
 [-0.85355339 -0.25       -0.45710678  0.42928932]
 [ 0.          0.          0.          1.        ]]
r = 0.5707106781308576
pz2 = -0.07071067811999998
s = 0.5750744978981189
alpha = 0.00876078242843278
beta = 0.12327112425372624
gamma = 2.356194489928527
-1.178097244964263
q1 = [0.7853981633974483, -2.356194490192345]
q2 = [1.4387644201127374, 1.68530666862019]
q3 = [0.7853981636612661, -2.3561944899285265]
q4 = [0.7853981635935081, -0.7853981635935081, 0.7853981635935081, -0.7853981635935081, 2.1862760353773445, -2.1862760353773445, 2.1862760353773445, -2.1862760353773445]
q5 = [0.7853981635345802, 2.356194490054991, 0.7853981635345802, 2.356194490054991, 1.047197551388152, 2.0943951022016414, 1.047197551388152, 2.0943951022016414]
q6 = [0.7853981630955807, -2.3561944904942123, 0.7853981630955807, -2.3561944904942123, 2.9716741990729454, -0.16991845451684773, 2.9716741990729454, -0.16991845451684773]
[[0.7853981633974483, -2.356194490192345], [1.4387644201127374, 1.68530666862019], [0.7853981636612661, -2.3561944899285265], [0.7853981635935081, -0.7853981635935081, 0.7
853981635935081, -0.7853981635935081, 2.1862760353773445, -2.1862760353773445, 2.1862760353773445, -2.1862760353773445], [0.785398163534802, 2.356194490054991, 0.785398163
534802, 2.356194490054991, 1.047197551388152, 2.0943951022016414, 1.047197551388152, 2.0943951022016414], [0.7853981630955807, -2.3561944904942123, 0.7853981630955807, -2.
3561944904942123, 2.9716741990729454, -0.16991845451684773, 2.9716741990729454, -0.16991845451684773]]
PS D:\Github\FoR>
```

3.2 Code Validation

For the computation q1 - q6 and l1 - l3 will be used again.

Again we look at the computed and corrected q2 seperatly. For the computed q2 IK will find the correct angles for q1 and q3 as expected:

```
Input angles: [0.7853981633974483, 1.5707963267948966, 0.7853981633974483, 0.7853981633974483, 0.7853981633974483, 0.7853981633974483]
[[-0.45710678  0.78033009  0.4267767   0.40355339]
 [ 0.25        0.5732233  -0.78033009  0.40355339]
 [-0.85355339 -0.25       -0.45710678  0.42928932]
 [ 0.          0.          0.          1.        ]]
r = 0.5707106781308576
pz2 = -0.07071067811999998
s = 0.5750744978981189
alpha = 0.008760782428432 78
beta = 0.12327112425372624
gamma = 2.356194489928527
-1.178097244964263
q1 = [0.7853981633974483, -2.356194490192345]
q2 = [1.4387644201127374, 1.68530666862019]
q3 = [0.7853981636612661, -2.3561944899285265]
q4 = [0.6778533812402868, -0.6778533812402868, 0.857217085034633, -0.857217085034633, 2.237709899762092, -2.237709899762092, 2.1529395931752275, -2.1529395931752275]
q5 = [0.6969916525315746, 2.444601001058219, 0.869320567047827, 2.272272086541966, 0.9412997993919932, 2.2002928541977997, 1.1418297567005409, 1.9997628968892522]
q6 = [0.9309302373720522, -2.210662416217741, 0.6794332616083745, -2.4621593919814186, 2.8775042530161974, -0.2640884005735957, 3.0442786250132863, -0.09731402857650681]
Found correct angle for q0
Found correct angle for q2
[0.7853981633974483, [], 0.7853981636612661, [], [], []]
PS D:\Github\FoR>
```

For the corrected q2, all angles except of q2 will be found correctly:

```
Input angles: [0.7853981633974483, 1.5707963267948966, 0.7853981633974483, 0.7853981633974483, 0.7853981633974483, 0.7853981633974483]
[[-0.45710678  0.78033009  0.4267767   0.40355339]
 [ 0.25        0.5732233  -0.78033009  0.40355339]
 [-0.85355339 -0.25       -0.45710678  0.42928932]
 [ 0.          0.          0.          1.        ]]
r = 0.5707106781308576
pz2 = -0.07071067811999998
s = 0.5750744978981189
alpha = 0.008760782428432 78
beta = 0.12327112425372624
gamma = 2.356194489928527
-1.178097244964263
q1 = [0.7853981633974483, -2.356194490192345]
q2 = [1.4387644201127374, 1.68530666862019]
q3 = [0.7853981636612661, -2.3561944899285265]
q4 = [0.7853981635935081, -0.7853981635935081, 0.7853981635935081, -0.7853981635935081, 2.1862760353773445, -2.1862760353773445, 2.1862760353773445, -2.1862760353773445]
q5 = [0.785398163534802, 2.356194490054991, 0.785398163534802, 2.356194490054991, 1.047197551388152, 2.0943951022016414, 1.047197551388152, 2.0943951022016414]
q6 = [0.7853981630955807, -2.3561944904942123, 0.7853981630955807, -2.3561944904942123, 2.9716741990729454, -0.16991845451684773, 2.9716741990729454, -0.16991845451684773]
Found correct angle for q0
Found correct angle for q2
Found correct angle for q3
Found correct angle for q4
Found correct angle for q5
[0.7853981633974483, [], 0.7853981636612661, 0.7853981635935081, 0.785398163534802, 0.7853981630955807]
```

3.3 Addidtional Functionalities

3.3.1 Parameter "fromFK" for Inverse Kinematics

For the function IKsolve the additional parameter fromFK indicates if the Inverse Kinematics will be solved based on previous Forward Kinematics solve. Therefore, the input angles q1-q6 are well defined and can be later verified. This helps to find the excact angles for q1-q6 and not just a list of all possible solutions. If not set to True, fromFK will be False and the output of this function will be a list of all possible soultions.

Important: If fromFK is used, the list of input angles have to be saved in global variable q. If not, the angles can not be verified.

3.4 Class "Robot"

The file "Robot.py" holds a robot class. This class will be initialized with a list of 6 input angles and a list of 3 link lengths. From this value it will generate Antropomorphic robot with a spherical XZX wrist.

In this class Forward- and Inverse Kinematics can be calculated. Furthermore, it can perform a series of Forward Kinematics as well as plot the robot.

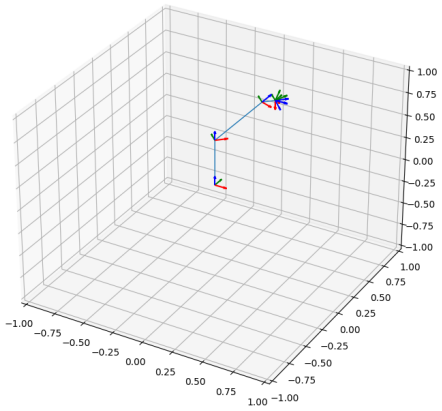
Running Robot.py will demonstrate a plot for a robot, then apply a series of Forward Kinematics and then a plot of the new robot position.

3.4.1 Plotting

The function plot plots the current robot configuration into a 3d-space.

For q1 = q2 = q3 = q4 = q5 = q6 = $\pi/4$ and l1 = 0.5, l2 = 0.5, l3 = 0.1

We get this plot:



3.4.2 Multiple Forward Kinematics

With this function a series of Forward Kinematics can be calculated.

Let l1 = 0.5, l2 = 0.5, l3 = 0.1

The first list of input angles: [0, $\pi/2$, 0, $\pi/4$, $\pi/2$, $\pi/4$] and

The second list of input angles: [$\pi/2$, $\pi/2$, 0, $\pi/2$, $\pi/4$, $\pi/4$]

Plotting the calculated robot configuration gives: