METR4202 -- Robotics
Tutorial 3 – Week 4: Forward Kinematics

Ekka Day Tutorial[1]

Solutions

The objective of this tutorial is to explore homogenous transformations. The MATLAB robotics toolbox developed by Peter Corke might be a useful aid[2].

Reading

Please read/review Please read/review chapter 7 of Robotics, Vision and Control.

Review

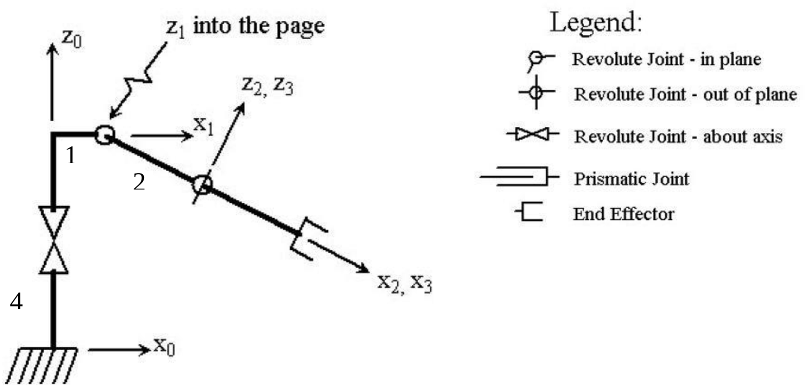
Useful commands:

```
Transl,trotx, troty, trotz, rotx, roty, rotz, tr2eul, DHFactor
```

Familiarise yourself with the link class

Questions

1. For the robot shown in the following figure, find the table of DH parameters according to "Standard" DH conventions.
(**note**: you are allowed to move the initial frame to fit convention(s))



Answers:

Link	${}_{From}$Frame	^{To}Frame	θ_i	d_i	a_i	α_i
1	0	1	θ1*	4	1	-90°
2	1	2	θ2*	0	2	90°
3	2	3	θ3*	0	0	0

➔ Note that the position of the end effector (the gripper) may be viewed as a position vector ($\mathbf{P}^{\mathbf{end\_effector}}$) in Frame 3.

---

[1] As this tutorial is on Ekka Day, it is not being held / assessed. The material is posted as reference.
[2] http://petercorke.com/Robotics_Toolbox.html

2.

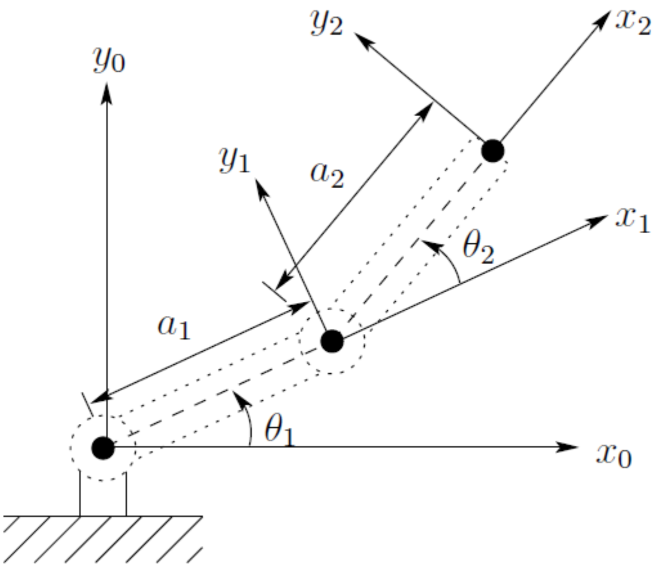


Figure 1: Two-link Planar Robot

a.) Determine the joint angles of the two-link planar arm.

The joint space of the robot is (θ_1, θ_2).
The forward kinematics may be solved directly using the vector-loop method or somewhat more mechanically using the DH convention (see slides 24 and 42 of Lecture 3). This gives:
$(p_x, p_y) = (a_1 c\theta_1 + a_2 c\theta_{12},\ \ a_1 s\theta_1 + a_2 s\theta_{12})$
The inverse kinematics involves solving the above simultaneous equation for θ_1 and θ_2.
A geometric way of solving this is to observe that the distance from {0} to {2} is independent of θ_1. Thus, sum of squares gives:

$$p_x^2 + p_y^2 = a_1^2 + a_2^2 + 2a_1a_2c\theta_2$$

$$\theta_2 = \arccos\left(\frac{p_x^2 + p_y^2 - a_1^2 - a_2^2}{2a_1a_2}\right)$$

If θ^* is an answer to the above, the, in general, $-\theta^*$ will also be an answer. This is corresponds to the “elbow up” and “elbow down” configurations.
Substituting this back into the kinematic equations gives:

$$p_x = (a_1 + a_2c\theta_2)c\theta_1 - (a_2s\theta_2)s\theta_1,\ p_y = (a_2s\theta_2)c\theta_1 + (a_1 + a_2c\theta_2)s\theta_1$$

$$c\theta_1 = \frac{p_x(a_1 + a_2c\theta_2) + p_y(a_2s\theta_2)}{a_1^2 + a_2^2 + 2a_1a_2c\theta_2}$$

$$s\theta_1 = \frac{-p_x(a_2s\theta_2) + p_y(a_1 + a_2c\theta_2)}{a_1^2 + a_2^2 + 2a_1a_2c\theta_2}$$

θ_1=Atan2($s\theta_1$, $c\theta_1$)

If a1 = 2 and a2 = 3 what are the joint angles corresponding to an end effector position of (x,y)=(1, 1).

θ_1= 167.028^o, θ_2 =-156.44^o (Elbow down)
Or θ_1= -77.028^o, θ_2 =156.44^o (Elbow up)

To verify using the Robotics Toolbox:

```
L(1) = Link([ 0      0    2   0], 'standard')
L(2) = Link([ 0      0    3   0], 'standard')
twolink = SerialLink(L, 'name', 'two link')
T=rpy2tr(0,0,0); T(1:2, 4)=[1 1]
Qsol=twolink.ikine(T, zeros(1,2), [1 1 0 0 0 0])

Qsol =

    2.9152   -2.7305
```