The difference between Atan2 in book and Matlab: in the text book it is written as Atan2(x, y) but in Matlab it is Atan2(y, x).

Problem 1: Consider the PRP robot with spherical wrist shown in Fig. 1. Consider all 3 DOF of the spherical wrist to be concentric (zero lengths between joints).

1. Assign a complete set of coordinate frames following the D-H convention and include a summary table. Sketch all frames. (please see above the figure)

Table 1. DH convention table. (Note: the transformation called b is not considered as joint. It is just a transformation in order to comply with DH convention.)

<table>
<thead>
<tr>
<th>Link</th>
<th>a</th>
<th>( \alpha )</th>
<th>( d )</th>
<th>( \theta )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>( +\frac{\pi}{2} )</td>
<td>( d_1 + q_1^* )</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>( q_2^* )</td>
</tr>
<tr>
<td>( b )</td>
<td>0</td>
<td>( +\frac{\pi}{2} )</td>
<td>0</td>
<td>( +\frac{\pi}{2} )</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>( d_2 + q_3^* )</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>( -\frac{\pi}{2} )</td>
<td>0</td>
<td>( q_4^* )</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>( +\frac{\pi}{2} )</td>
<td>0</td>
<td>( q_5^* )</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>( d_3 )</td>
<td>( q_6^* )</td>
</tr>
</tbody>
</table>
2. For a point \( p \) with local coordinates \( p_6 = (0, 0, d_3) \), determine the forward kinematic transformation \( H_6^0 \) (write code that can calculate the transformation for any given joint coordinate vector \( q \) and parameters \( d_1, d_2, d_3 \); don’t try to show the symbolic expression !).

The following is the code is written for this system:

```matlab
%%
t=0:0.001:10;
d1=1;
d2=1;
d3=0.25;
for i = 1 : length(t)
    q1(i)=0.5 + 0.25*sin(t(i));
    q3(i)=0.5 + 0.25*sin(t(i));
    q2(i)= 0.5*sin(2*t(i));
    q4(i)= 0.5*sin(2*t(i));
    q5(i)= cos(t(i));
    q6(i) = 2*t(i);
    \% Transformation:
    H10=Transz(d1+q1(i))*Rotx(pi/2);
    H21=Rotz(q2(i));
    Hb2=Rotz(pi/2)*Rotx(pi/2);
    H3b=Transz(d2+q3(i));
    H43=Rotz(q4(i))*Rotx(-pi/2);
    H54=Rotz(q5(i))*Rotx(pi/2);
    H65=Rotz(q6(i))*Transz(d3);
    H60=H10*H21*Hb2*H3b*H43*H54*H65;

    \%Find world position of point of interest:
    Wposition(i,:)=H60*[0;0;d3;1];
    Wposition= Wposition (:,1:3);
    \%Plot point of interest in the world frame
    plot3(Wposition (i,1), Wposition (i,2), Wposition (i,3),'b+')
    hold on
    plot(t,q1)
end
```

Use the following numerical values: \( d_1 = d_2 = 1 \), \( d_3 = 0.25 \). Suppose that the joints are moved according to the following functions:
- \( q_1(t) = q_3(t) = 0.5 + 0.25 \sin(t) \)
- \( q_2(t) = q_4(t) = 0.5 \sin(2t) \)
- \( q_5(t) = \cos(t) \)
- \( q_6(t) = 2t \)

4. Make a 3D plot of the world position of \( p \) for \( t \) from zero to 10 time units.
Figure 1. 3D plot of the world position P
Figure 2. Plot of the world position of x, y, z.
Figure 2. Plot of the world position of x, y, z.
Problem 2:

1. Assign coordinate frames starting from the information given in Fig. 3 and the dimensions at the end of this document. Use the D-H convention.

![Diagram](image)

Table 2. DH convention table for 4-DOF SCARA system

<table>
<thead>
<tr>
<th>Link</th>
<th>$a$</th>
<th>$\alpha$</th>
<th>$d$</th>
<th>$\theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$L_1$</td>
<td>0</td>
<td>0</td>
<td>$q_1^*$</td>
</tr>
<tr>
<td>2</td>
<td>$L_2$</td>
<td>0</td>
<td>0</td>
<td>$q_2^*$</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>$d_3^*$</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0.075</td>
<td>$q_4^*$</td>
</tr>
</tbody>
</table>

2. Develop the necessary forward kinematics as Matlab code.

Please find the Matlab code in the following pages for the forward kinematics.
3. Consult SHV for the solutions to the inverse kinematics for this robot and make necessary adjustments or simplifications as you see fit.

By referring to the SHV for the inverse kinematic of the SCARA robot and some adjustment. The following formulas are used to find the angles $q_1, q_2, q_4$ and the displacement $d_3$.

$$H_4^0 = \begin{bmatrix}
\cos(q_1 + q_2 + q_4) & -\sin(q_1 + q_2 + q_4) & 0 & a_2\cos(q_1 + q_2) + a_1\cos(q_1) \\
\sin(q_1 + q_2 + q_4) & \cos(q_1 + q_2 + q_4) & 0 & a_2\sin(q_1 + q_2) + a_1\sin(q_1) \\
0 & 0 & 1 & d_3 + 0.075 \\
0 & 0 & 0 & 1
\end{bmatrix}$$

$$q_1 + q_2 + q_4 = \text{atan2}(-r11, r12)$$

After projecting the manipulator configuration onto the $x_0−y_0$ plane and consider it as a two-link manipulator, the following equations is obtained.

$$D = \frac{a_x^2 + a_y^2 - L_1^2 - L_2^2}{2L_1L_2}$$

$$\theta_2 = \text{atan2}(D, \pm\sqrt{1 - D^2})$$

$$\theta_1 = \text{atan2}(a_x, a_y)$$

And finally the displacement variable is obtained as:

$$d_3 = a_z - 0.075$$

4. Find the mathematical expressions for the space curve to be followed, as well as the end frame orientation and write code to generate them numerically, using the required welding speed.

We considered the 30mm length of the electrode in the curve space for compensation of the final position of the end effector. Therefore, instead of considering the 50mm radius for the cylinder, we assumed that there is a 50mm+30mm cylinder that should be welded. Each revolution should take 4 second only; therefore the 4 revolute will take 16 seconds. That’s why we found $\omega = 0.5\pi$ . $a$ is used to consider the rate of wearing off (2 mm per revolute). Therefore the value for $a$ is 0.0005. The curve radius is $r = 0.05 + 0.03 = 0.08$ m. The $z$ axis of the final position is increased by the rate $c = 0.04/16$.

The mathematical curve that can express the space curve in the world position is:

$$a_x = 0.4 + (r - at) \times \cos(0.5\pi t)$$

$$a_y = (r - at) \times \sin(0.5\pi t)$$

$$a_z = 0.085 + ct$$

The curve space is shown for the case that the electrode length is constant and equal to 30m and the case where it is wearing off by the rate 2 mm per revolute.
Figure 2. Curve space for welding the cylinder (constant electrode length).

Figure 3. Top and side view of curve space for welding the cylinder (electrode length is decreased by 2 mm/rev).
The end frame orientation for the curve space is:

The end effector should be normal to the circumference of the workpiece during the welding operation. Therefore the orientation of the end effector is a rotation by $Z$ axis with angle:

$$\gamma(t) = \pi + 0.5\pi t$$

$$Rot_{Z,\gamma} = \begin{bmatrix} \cos(\gamma(t)) & -\sin(\gamma(t)) & 0 \\ \sin(\gamma(t)) & \cos(\gamma(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

5. Use exact inverse kinematic formulas to find the four required joint angle histories.

By using the exact inverse kinematics from the part 3 of the problem. The joint angles ($q_1, q_2, q_4$) and the joint displacement ($d_3$) histories are sketched as the following figure.

![Graph showing joint angles and joint displacement histories](image)

Figure 4. The joint angles ($q_1, q_2, q_4$) and the joint displacement ($d_3$).

6. Verify by feeding your solutions back into the forward kinematics.

The solutions obtained in part 5 are fed into $H_4^0$ in order to see if both the inverse kinematic and forward kinematic show the same curve space.

$$O_4^0 = \begin{bmatrix} a2\cos(q_1 + q_2) + a1\cos(q_1) \\ a2\sin(q_1 + q_2) + a1\sin(q_1) \\ d_3 + 0.075 \end{bmatrix}$$

The end effector positions is sketched for all calculated joint angles and are compared to curve space in part 4. The following figure shows both end effector position from the direct and inverse kinematics.
Figure 5. The end effector position obtained from direct and inverse kinematics for the case that the electrode length is constant.
Figure 6. The end effector position obtained from direct and inverse kinematics for the case that the electrode is wearing off.
7. Attempt a direct numerical solution using Corke’s Robotics Toolbox (work in meters to prevent numerical issues). We used the Corkes’s Toolbox and we defined the Robot with the DH table. Then we built the transformation matrix from the analytical solution we got from the inverse kinematics. We plotted the result of Toolbox (joint angles) as \( q \) guesses and the positions we obtained from the forward kinematic. As the result show both positions lie in the same space curve and the results are verified. The following figure shows the Robotic Toolbox solutions and the forward kinematic positions.

![Figure 7. The robot visualization in Corke’s Toolbox](image)

![Figure 8. The trajectory of the analytical solution and Toolbox solution for the constant electrode](image)
Figure 9. The trajectory of the analytical solution and Toolbox solution for the electrode wearing off by 2 mm/rev.
Matlab code for Problem 2

clc; clear;

r = 0.05 + 0.03; %cylindrical radius + electrode length
a = 0.0005; %wearing off rate of electrode
%a=0;
c = 0.04/16; %rate of increasing in the direction f z axis

%defining the space curve
r = 0 : 0.1 : 16;
ox = 0.4 + (r-a.*t).*cos(0.5*pi*t);
oy = (r-a.*t).*sin(0.5*pi*t);
oz = 0.085 + c*t;
figure(1)
plot3(ox,oy,oz,'r*','LineWidth',2)
xlabel('x'); ylabel('y'); zlabel('z');

% the SCARA parameters
L1 = 0.325;
L2 = 0.225;
d4 = 0.075;

gamma = pi + 0.5*pi*t; % the rate of rotation around z axis (w
want to find the end effector orientation and since it is normal
to the workpiece it is like rotating across z axis.
for i = 1 : length(t)
    R = [cos(gamma(i)) -sin(gamma(i)) 0; sin(gamma(i)) 0 0 1]; %end effector orientation matrix
    %inverse kinematics exact formulas
    D = (ox(i)^2 + oy(i)^2 - L1^2 -L2^2)/(2*L1*L2);
    q2(i) = atan2(sqrt(1-D^2),D);
    q1(i) = atan2(oy(i),ox(i))-atan2(L2*sin(q2(i)),
L1+L2*cos(q2(i)));
    q4(i) = -atan2(R(1,2),R(1,1)) - q1(i) - q2(i);
%#ok
    d3(i) = oz(i) - d4;
%#ok
    R40 = [cos(q1(i) + q2(i) + q4(i)) -sin(q1(i) + q2(i) + q4(i))
      sin(q1(i) + q2(i) + q4(i)) cos(q1(i) + q2(i) + q4(i)) 0;
      0 0 1];
\[ S = R_{40} - R; \] % difference between forward and inverse rotation matrices
end
% the end effector position from the forward kinematics formulas
x = L2*cos(q1 + q2) + L1*cos(q1);
y = L2*sin(q1 + q2) + L1*sin(q1);
z = d3 + d4;
hold on
plot3(x,y,z,'b')
% plotting the joint angles
figure(2)
subplot(2,2,1);
plot(t,q1,'LineWidth',4)
xlabel('time') % x-axis label
ylabel('q1') % y-axis label
subplot(2,2,2);
plot(t,q2,'LineWidth',4)
xlabel('time') % x-axis label
ylabel('q2') % y-axis label
subplot(2,2,3);
plot(t,q4,'LineWidth',4)
xlabel('time') % x-axis label
ylabel('q4') % y-axis label
subplot(2,2,4);
plot(t,d3,'LineWidth',4)
xlabel('time') % x-axis label
ylabel('d3') % y-axis label
% corkes toolbox
% startup_rvc
% load toolbox if not done earlier
L1=0.325; L2=0.225;
d4=0.075; % location of point of interest on z6 axis
L(1)=Link([0 0 L1 0]);
L(2)=Link([0 0 L2 0]);
L(3)=Link([0 0 0 0 1]);
L(4)=Link([0 d4 0 0 1]);
% For the toolbox solution, the last frame has origin
% at the point of interest o. For the analytical solution, the last frame
% has origin at wrist center.
robot=SerialLink(L,'name','myrobot');

figure(3)
W=[-10 10 -10 10 -10 10];
robot.plot([pi/4 pi/4 0 pi/2],'workspace',W); %plot some pose

% Toolbox solution
qguess=[q1(1) q2(1) d3(1) q4(1)]; %initial analytical solution
for i = 1 : length(t)
  % Build rotation matrix:
  R40 = [cos(gamma(i)) -sin(gamma(i)) 0; sin(gamma(i)) cos(gamma(i)) 0; 0 0 1];
  H=[R40 [ox(i);oy(i);oz(i)];0 0 0 1];
  qnum(:,i)=robot.ikine(H,qguess,[1 1 1 0 0 1])'; %ignore x,y components of orientation
  qguess=qnum(:,i)';
end

% Verification
figure(4)
for i=1:length(t),
  H=robot.fkine(qnum(:,i)'); %pull out transformation
  check=H(1:3,4); %look in 4th column to extract world coordinates of endpoint
  plot3(check(1),check(2),check(3),'ko')
  hold on
end
title('Trajectory: Desired, Analytical and Toolbox')
xlabel('x_0')
ylabel('y_0')
zlabel('z_0')
hold on
plot3(x,y,z,'b')