An approach for solving industrial robot forward and inverse kinematics

PENG Cheng, LI Jun-qiang, SU Wen-long，ZHU Shuai-chen，GUO Xiong

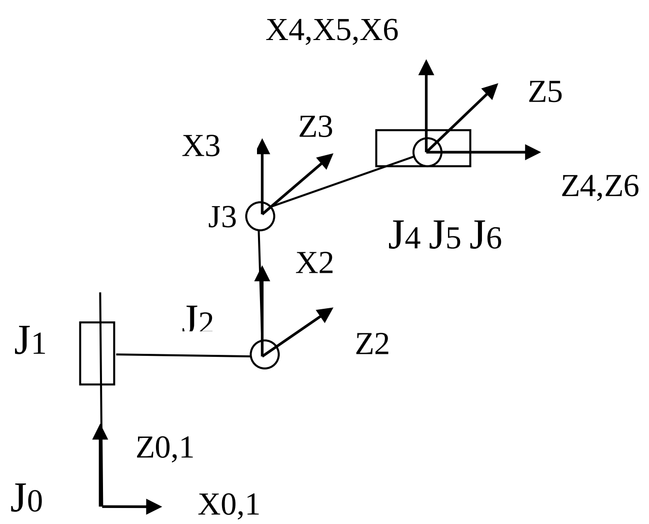
ABSTRACT

Solving the forward/inverse-pose kinematic transformation is the basic problem in the study of mechanical manipulation. In this paper we introduced a widely used approach to working out the forward-pose transformation by using the D-H parameter table, together with a geometric approach to solve the inverse-pose transformation. All the approaches are implemented via MATLAB programs, and the programs are tested by given input data. Results are checked by some functions in the Corke MATLAB robotics Toolbox. The model is established according to the widely used industrial robot, FANUC 2000ib/165EW.

Key word: industrial robot, forward-pose transformation, inverse-pose transformation, kinematic, geometric approach.

Forward-pose kinematics

D-H table. By analyze the drawing of the FANUC Robot, we can built its model as below, which is simplified by some common rules:



Built coordinates on each joint and we can get the D-H parameter table as below

$Link_i$	a_{i-1}	d_i	a_i	θ_i
1	0	0	0	0
2	312	0	-90°	0
3	1075	0	0	0
4	725	1280	-90°	0
5	0	0	90°	0
6	0	0	-90°	0

Homogeneous transformation matrix. the standard process to derive the Homogeneous transformation matrix, which is what we used, is as below:

Rotate about X_{i-1} by α_i

Translate along X_{i-1} by a_{i-1}

Rotate about Z_i by θ_i

Translate along Z_i by d_i

Implement standard process in MATLAB code:

```
A1=rotx(alpha(1))*transl(a(1),0,0)*rotz(theta(1))*transl(0,0,d(1))
A2=rotx(alpha(2))*transl(a(2),0,0)*rotz(theta(2))*transl(0,0,d(2))
A3=rotx(alpha(3))*transl(a(3),0,0)*rotz(theta(3))*transl(0,0,d(3))
A4=rotx(alpha(4))*transl(a(4),0,0)*rotz(theta(4))*transl(0,0,d(4))
A5=rotx(alpha(5))*transl(a(5),0,0)*rotz(theta(5))*transl(0,0,d(5))
A6=rotx(alpha(6))*transl(a(6),0,0)*rotz(theta(6))*transl(0,0,d(6))
```

Multiply the matrix to get transformation matrix ${}^0_3T$, ${}^0_6T$

```
B1=A1*A2*A3    %T (03)
B2=A1*A2*A3*A4*A5*A6    %T (06)
```

Check. To check the matrix derived, function link()、robot()、fkine() in robot toolbox are used to build the robot model and derive its forward-pose transformation matrix.

```
sigma=0    %0表示运动关节
L1=link([alpha(1),a(1),theta(1),d(1),sigma],'mod')
L2=link([alpha(2),a(2),theta(2),d(2),sigma],'mod')
L3=link([alpha(3),a(3),theta(3),d(3),sigma],'mod')
L4=link([alpha(4),a(4),theta(4),d(4),sigma],'mod')
L5=link([alpha(5),a(5),theta(5),d(5),sigma],'mod')
L6=link([alpha(6),a(6),theta(6),d(6),sigma],'mod')
R=robot({L1,L2,L3,L4,L5,L6},'Rlan3R')
q=[0 0 0 0 0 0]
%q=theta
IR=fkine(R,q)
```

Input the given data as below:

a) $[\theta_1\theta_2\theta_3\theta_4\theta_5\theta_6]^T = [0\ 0\ 0\ 0\ 0\ 0]^T$

b) $[10°\ \ 20°\ \ 30°\ 10° - 30°\ 10°]^T$

c) $[90°\ \ -35°\ \ 79° - 80°\ 10°\ 120°]^T$

All of the matrix fit the check result (check ${}^0_6T$ only for simplification):

a) $[\theta_1\quad \theta_2\quad \theta_3\quad \theta_4\quad \theta_5\quad \theta_6]^T = [0\quad 0\quad 0\quad 0\quad 0\quad 0]^T$

```
IR =

   1.0e+03 *

    0.0010         0         0    1.6120
         0   -0.0010    0.0000    0.0000
         0   -0.0000   -0.0010   -1.2800
         0         0         0    0.0010

>> B2

B2 =

   1.0e+03 *

    0.0010         0         0    1.6120
         0   -0.0010    0.0000    0.0000
         0   -0.0000   -0.0010   -1.2800
         0         0         0    0.0010
```

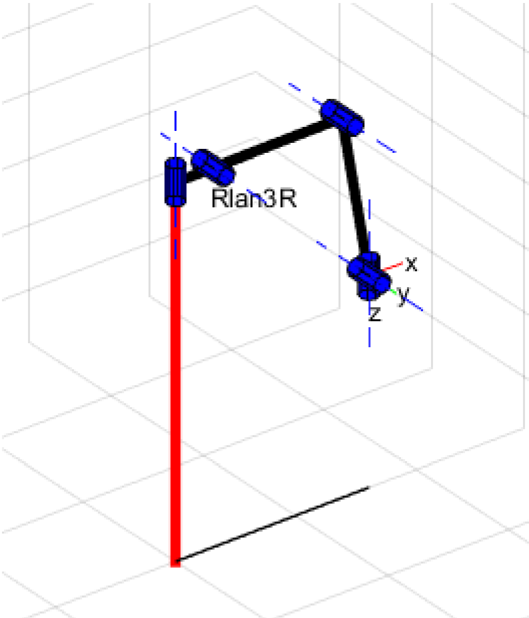
b) $[\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad \theta_5 \quad \theta_6]^T = [10° \quad 20° \quad 30° \quad 10° \quad -30° \quad 10°]^T$

```
B2 =

   1.0e+03 *

    0.0009   -0.0001   -0.0003    0.4789
   -0.0002   -0.0010   -0.0001    0.0844
   -0.0003    0.0002   -0.0009   -1.3628
         0         0         0    0.0010

>> IR

IR =

   1.0e+03 *

    0.0009   -0.0001   -0.0003    0.4789
   -0.0002   -0.0010   -0.0001    0.0844
   -0.0003    0.0002   -0.0009   -1.3628
         0         0         0    0.0010
```

c) $[\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad \theta_5 \quad \theta_6]^T = [90° \quad -35° \quad 79° \quad -80° \quad 10° \quad 120°]^T$

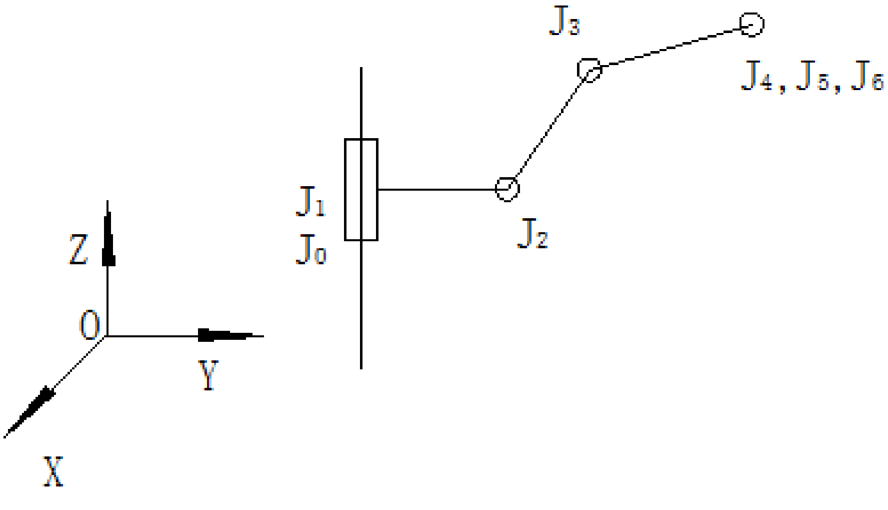
```
IR =

    0.6353    0.7531    0.1710   -0.0000
    0.6123   -0.3563   -0.7058  465.2772
   -0.4706    0.5531   -0.6875 -460.4584
         0         0         0    1.0000

>> B2

B2 =

    0.6353    0.7531    0.1710   -0.0000
    0.6123   -0.3563   -0.7058  465.2772
   -0.4706    0.5531   -0.6875 -460.4584
         0         0         0    1.0000
```

Function drivebot(R, q) can be used to draw the figure of the robot model, which is coincident with the model we built before.

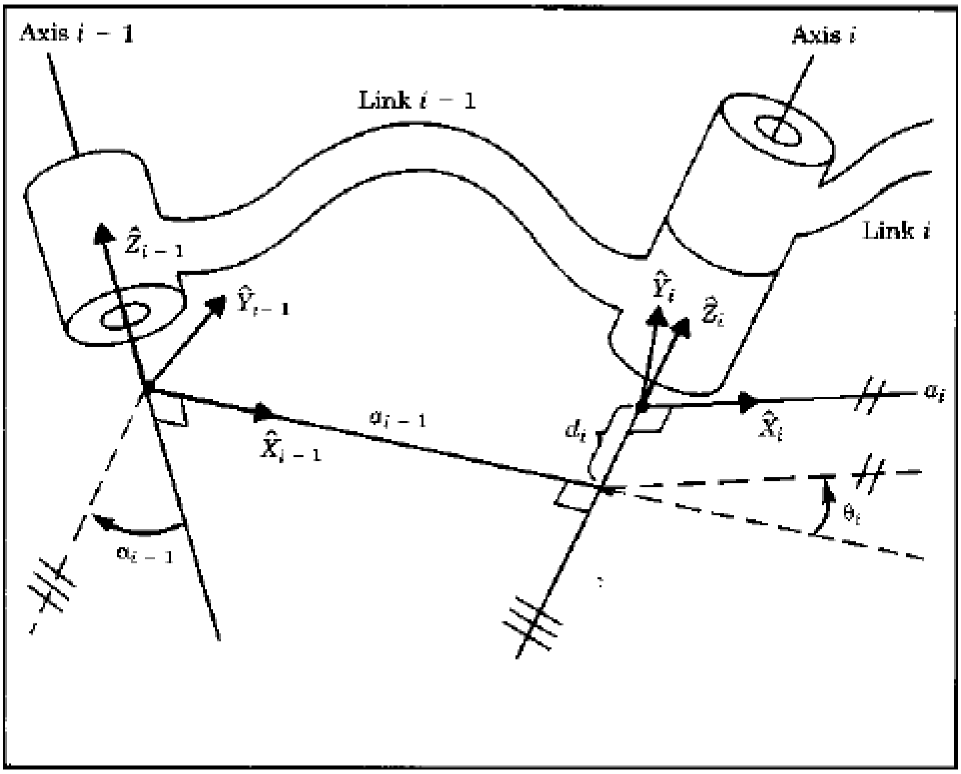


Inverse-pose kinematics

Inverse-pose solution. To derive the solution of a given ${}^{0}_{H}T$,we decided to used a partially geometric and partially algebraic method. Build a coordinate as below:

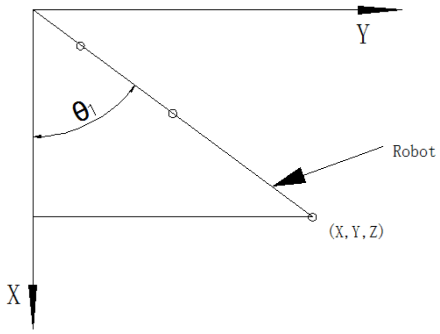


As we know, derive the inverse-pose solution means find the equation of θ_i ($i=1,\cdots,6$) , θ_i is the intersection angle of the two common normal line between axis(i-1)、axis(i) and axis(i)、axis(i+1)



① θ_1

Express θ_1 in X-Y plane



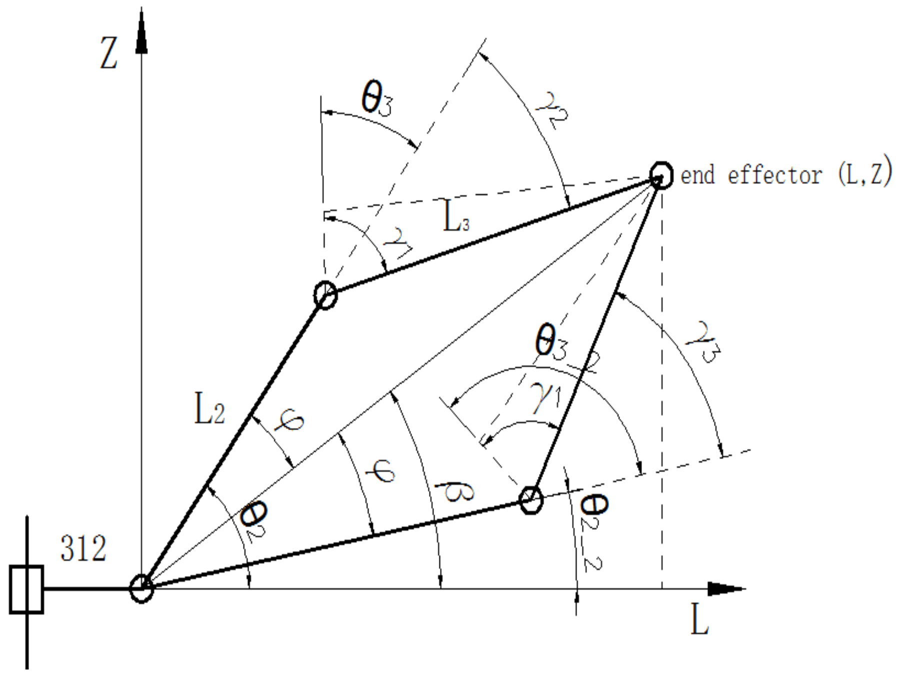
Form the ${}^{H}_{0}T$ we can easily get the coordinates (x, y, z) of the end effector,

in which $x={}^{6}_{H}T(1,4)$， $y={}^{6}_{H}T(2,4), z={}^{6}_{H}T(3,4)$

Then $\theta_1 = A\tan 2(y,x)$

② θ_2, θ_3

To derive the equation of θ_2, θ_3 ,the robot is analyzed in vertical plane.



Relationships among angles are as above .

In this case, we can usually find a solutions of a ${}^{0}_{H}T$,but when the manipulator is at its singularity, only one solution will be found.

We can apply the “law of cosines” to solve θ_3 .

$$Z^2 + L^2 = L_2{}^2 + L_3{}^2 - 2L_2L_3COS(180° - r_2)$$

Introduce

$$L = \sqrt{x^2 + y^2} - 312$$

$$\cos(180° - r_2) = -\cos(r_2)$$

We get

$$\cos r_2 = \frac{L^2 + Z^2 - L_2{}^2 - L_3{}^2}{2L_2L_3}$$

$$r_2 = a\cos\left(\frac{L^2 + Z^2 - L_2{}^2 - L_3{}^2}{2L_2L_3}\right)$$

Also we know

$$\mathrm{r}_1 = a\tan 2(1280{,}225)$$

So

$$\theta_3 = -(\mathrm{r}_1 - r_2) = r_2 - r_1$$

θ_3 in another solution has a different equation

$$\theta_{3\text{-}2} = -(r_1 + r_3)$$

In which

$$Z^2 + L^2 = L_2{}^2 + L_3{}^2 - 2L_2L_3\cos(180° - r_3)$$

Apply $L = \sqrt{X^2 + Y^2} - 312$

$$\cos(180° - r_3) = -\cos(r_3)$$

we get

$$r_3 = a\cos\left(\frac{L^2 + Z^2 - L_2{}^2 - L_3{}^2}{2L_2L_3}\right)$$

And

$$\mathrm{r}_1 = a\tan 2(1280{,}225)$$

For θ_2,we know

$$\theta_2 = \beta + \varphi$$

$$\theta_{2\text{-}2} = \beta - \varphi$$

In which

$$\beta = \mathrm{atan2}(\mathrm{Z}, \mathrm{L})$$

And

$$Z^2 + L^2 + L_2{}^2 - L_3{}^2 = 2L_2\sqrt{Z^2 + L^2} \bullet \cos\varphi$$

$$\varphi = a\ \cos\left(\frac{L^2 + Z^2 + L_2{}^2 - L_3{}^2}{2L_2\sqrt{Z^2 + L^2}}\right) \quad , \quad \varphi \in [0° \sim 180°]$$

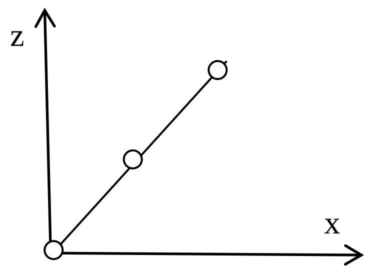
Before calculate θ_2 , θ_3 ,we have to make sure that the position of end effector expressed by ${}^{0}_{H}T$ is in the reach of the manipulator, which means ${}^{0}_{H}T$ has a solution.

So

$$L_2 + L_3 \geq \sqrt{Z^2 + L^2}$$

When $L_2 + L_3 = \sqrt{Z^2 + L^2}$

The manipulator is at the singularity point.



After getting the values of θ_1 , θ_2 , θ_3 by geometric approach, we used algebraic approach to derive the values of $\theta_4, \theta_5, \theta_6$.

Introduce the values of $\theta_1, \theta_2, \theta_3$ in the equation

$${}^{0}_{1}T\ {}^{1}_{2}T\ {}^{2}_{3}T\ {}^{3}_{4}T\ {}^{4}_{5}T\ {}^{5}_{6}T = {}^{0}_{H}T$$

$${}^{3}_{4}T\ {}^{4}_{5}T\ {}^{5}_{6}T = (\ {}^{0}_{1}T\ {}^{1}_{2}T\ {}^{2}_{3}T\ )^{-1}\ {}^{0}_{H}T$$

Assume

$$X_1 = (\ {}^{1}_{0}T\ \ {}^{1}_{2}T\ \ {}^{3}_{4}T\ )^{-1}\ {}^{0}_{H}T$$

$$X_2 = {}^{3}_{4}T\ \ {}^{4}_{5}T\ \ {}^{5}_{6}T$$

The results are below:

```
>> X2=A4*A5*A6

X2 =

[   cos(theta4)*cos(theta5)*cos(theta6) - sin(theta4)*sin(theta6), - cos(theta6)*sin(theta4) - cos(theta4)*cos(theta5)*sin(theta6), -cos(theta4)*sin(theta5),  225]
[                                       cos(theta6)*sin(theta5),                                        -sin(theta5)*sin(theta6),           cos(theta5), 1280]
[ - cos(theta4)*sin(theta6) - cos(theta5)*cos(theta6)*sin(theta4),   cos(theta5)*sin(theta4)*sin(theta6) - cos(theta4)*cos(theta6),  sin(theta4)*sin(theta5),    0]
[                                                             0,                                                               0,                        0,    1]
```

Chose 3 equations to derive $\theta_4\ \theta_5, \theta_6$.

$$\begin{cases} X_2(2,3) = X_1(2,3) \\ X_2(3,3) = X_1(3,3) \\ X_2(2,1) = X_1(2,1) \end{cases}$$

$$\Rightarrow \begin{cases} \theta_4 = a\ \cos(\dfrac{X_1(3,3)}{\sin\theta_5}) \\ \theta_5 = -a\ \cos(X_1(2,3)) \\ \theta_6 = a\ \cos(\dfrac{X_1(2,1)}{\sin\theta_5}) \end{cases}$$

Up to now, we have derived the solution of the given translation matrix

$$\theta_i = (\theta_1,\ \ \theta_2,\ \ \theta_3,\ \ \theta_4,\ \ \theta_5)$$

Check. Function ikine(), drivebot() can be used to check the result.

Test the program by given ${}^0_HT$:

T1:

$${}^0_HT = {}^0_6T = \begin{bmatrix} -0.11013 & 0.52562 & 0.84356 & 1604.7 \\ -0.96534 & -0.1455 & 0.21668 & 926.49 \\ 0.23663 & -0.83819 & 0.49138 & 1569.1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

T2:

$$
{}_{H}^{0}T = {}_{6}^{0}T = \begin{bmatrix} -0.73794 & 0.57972 & 0.34551 & 1499.3 \\ -0.63372 & -0.77128 & -0.059391 & 0 \\ 0.23205 & -0.26278 & 0.93654 & 2056.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

T3:

$$
{}_{H}^{0}T = {}_{6}^{0}T = \begin{bmatrix} 0.1658 & -0.1736 & -0.9708 & 2655 \\ 0.0292 & 0.9848 & -0.1712 & 866.5 \\ 0.9857 & 0 & 0.1683 & 806.3 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

It is also necessary to check if each joint angle result is in the working range as below table.

各轴工作范围	
	角度（度）
J1 轴	-180-0/0-180
J2 轴	-60-0/0-75
J3 轴	-128-0/0-80
J4 轴	-360-0/0-360（软件限位）
J5 轴	-125-0/0-125
J6 轴	-220-0/0-220

Due to the degree of accuracy of iteration , the ikine() function can only derived one solution of T1, as below :

```
q =[0.5236 -1.2217 -0.6196  0.7541 -0.3490 0.4362]
```

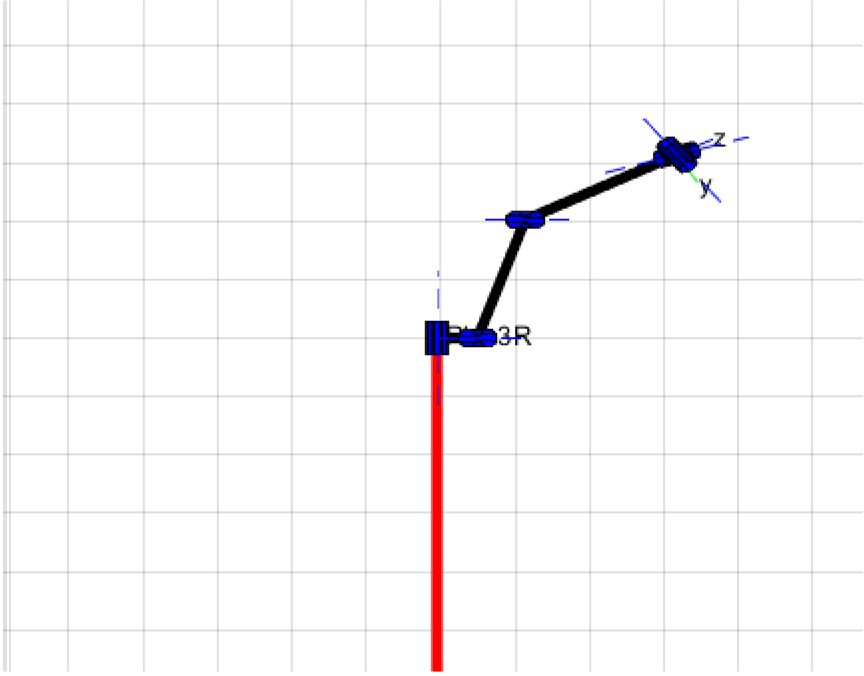
Following are the result derived by the partially geometric and partially algebraic algorithm:

One solution of the T1

```
theta =

    0.5236   -1.2217   -0.6196    0.8167   -0.3491    0.4363
```

Most values in the solution are coherent to the result of ikine.

Pose of the manipulator derived from the solution above:



Convert to angle system:

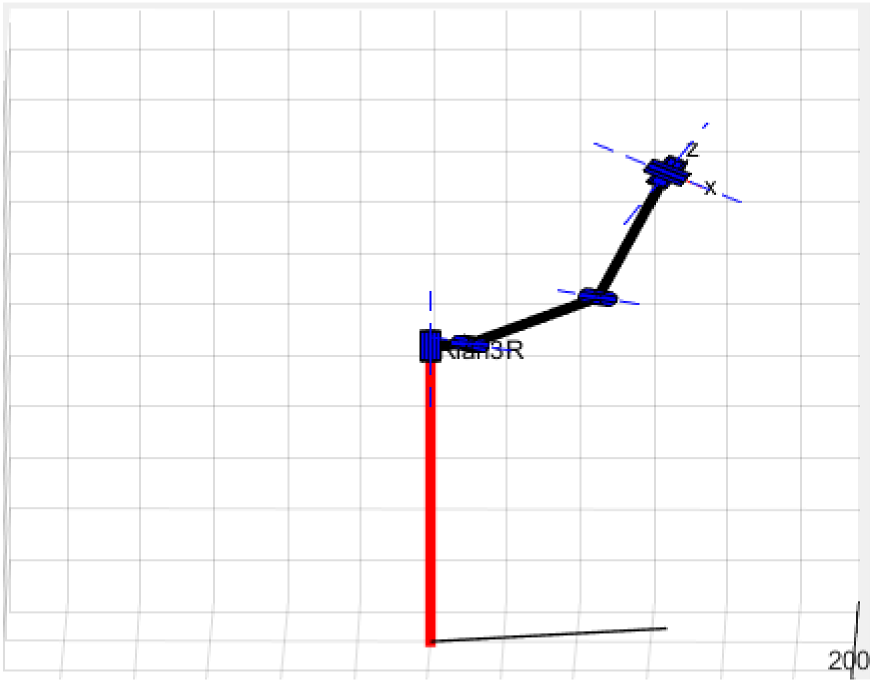
```
q =

   30.0004  -70.0008  -35.5014   46.7962  -19.9991   24.9986
```

We find theta2 is out of range, so it is an invalid pose for the manipulator.

Another solution of T1

```
theta =

    0.5236   -0.3672   -2.1740    1.0566   -0.4962    1.5212
```

Pose of the manipulator derived from the solution above:



It is easy to find that two solutions have the same position of the end effector and are in the pose as we expected.

Convert to angle system:

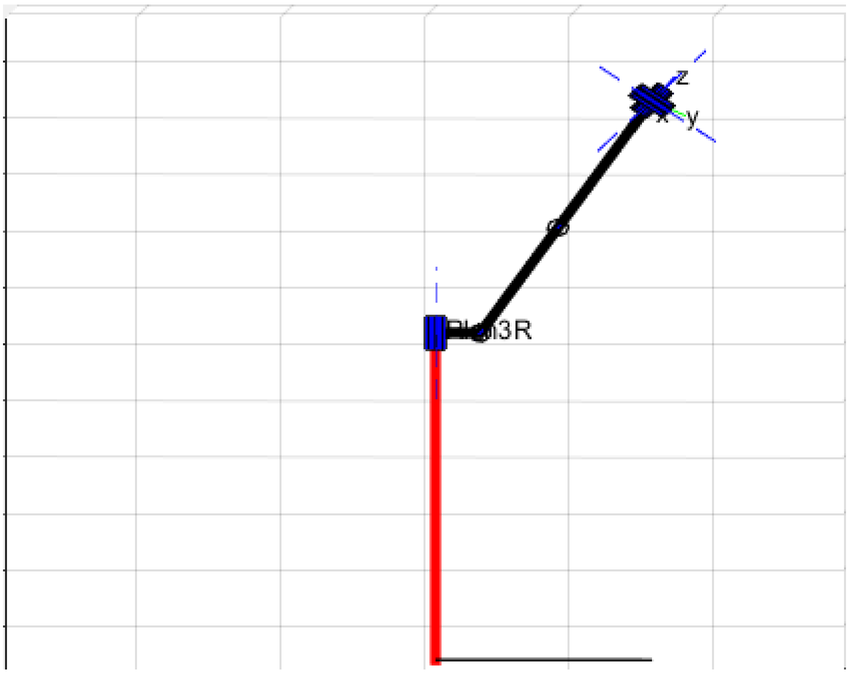
```
q =

   30.0004  -21.0362 -124.5593   60.5391  -28.4278   87.1582
```

All the angles are in working range.

solution of the T2 (T2 has only one solution because the end effector is at singularity point)

```
q =

         0   -1.0472   -1.3968    1.3962   -0.3490    0.5235
```

Figure derive by “drivebot()” function:



Convert to angle system:

```
q =

         0  -60.0001  -80.0307   79.9964  -19.9962   29.9943
```

All the angles are in working range.

T3 has no solution, because it doesn't satisfy the condition $L_2 + L_3 \geq \sqrt{Z^2 + L^2}$, when means the pose presented by the transformation matrix is out of the reach of the manipulator.

```
>> sqrt(z^2+Len^2)

ans =

   2.6086e+03

>> Len2+Len3

ans =

   2.3746e+03

>> if(cmp1>cmp2)

    fprintf('out of reach');
else
    theta3=acos((z^2+Len^2-Len2^2-Len3^2)/(2*Len2*Len3))-atan2(1280,225)
    fprintf('theta3=%d\n',theta3 )
end
out of reach>>
```

With the all the result we get, it is safe to say that our algorithms work well.

CONCLUSION

In this paper, focusing on the kinematics of the industrial robot, we discussed a standard way to get the forward transformation matrix based a 6 DOF manipulator. Also we introduced a partially geometric and partially algebraic method to derive the inverse solution of the robot. All the algorithms are implemented via MATLAB programs, and are proved correct by the result. The method has the two following distinguishing features: (1) simple and easy to understand. (2) small amount of calculation. (3) able to derive multiple solutions.

REFERENCES

[1] HUANG Chen-hua. "A geometric approach for solving a industrial robot inverse kinematics."[J] *Manufacturing automation*, http://magazine.cmanu-automation.cn/ch/reader/view_abstract.aspx?flag=1&file_no=242589&journal_id=zzyyzdh May. 27, 2014:36(8).

[2] John, J. Craig. *introduction to robotics: mechanics and control-2nd ed* [M] ,Wiley, USA.

[3] ZHOU Fang-fang & FAN Xiao-ping & YE Zhen."3-D virtual robotic model generated by D-H table."[J] *journal of system simulation*, http://kns.cnki.net/KCMS/detail/detail.aspx?dbcode=CJFQ&dbname=CJFD2006&filename=XTFZ200604034&v=MjQ4NzRxVHJXTTFGckNVUkwyZlllWnBGeTdoVnJ2TFBUbk5kTEc0SHRmTXE0OUdZSVI4ZVgxTHV4WVM3RGgxVDM= April.2006: 034