1. Connections between the forward and inverse kinematics

(1). Comparison between DH and POE in forward kinematics.

The DH in robotics and the POE.

Note that vector w should keep aligned with the positive rotation in DH frames.

```
% 如下，对于theta=[60,50,40,30,20,10,0]/180*pi;dh和poe结果完全相同。
% T =
%
%      0.8883     0.3511    -0.2962   368.3227
%      0.3123    -0.9345    -0.1710   637.9536
%     -0.3368     0.0594    -0.9397  -283.6111
%           0          0          0     1.0000
%
%
% T =
%
%      0.8883     0.3511    -0.2962   368.3227
%      0.3123    -0.9345    -0.1710   637.9536
%     -0.3368     0.0594    -0.9397  -283.6111
%           0          0          0     1.0000
```

The codes are respectively called googoledirectdh and googoldirectpoe

Note: both method we omit the terminal of the last link, we can attach the parameters of the terminal in both methods. (The file from Liu is not completely right.)

Note2: we can define variables theta for symbolic solution.

(2).Additionally，we compare the transformation matrix.

$$g_{st}(\theta) = e^{\hat{\xi}_1\theta_1} e^{\hat{\xi}_2\theta_2} e^{\hat{\xi}_3\theta_3} e^{\hat{\xi}_4\theta_4} e^{\hat{\xi}_5\theta_5} e^{\hat{\xi}_6\theta_6} g_{st}(0) := g_d$$

$$T_6 = A_1 A_2 A_3 A_4 A_5 A_6 = \begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The book of Li points out the relationship:

$$\xi_i = \mathrm{Ad}_{g_{l_0 l_{i-1}}(0)}\, \xi_{i-1,i}.$$

$$g_{st}(\theta) = e^{\widehat{\xi}_{01}\theta_1} \left(g_{l_0 l_1}(0) e^{\widehat{\xi}_{12}\theta_2} g_{l_0 l_1}^{-1}(0) \right) \cdot \left(g_{l_0 l_2}(0) e^{\widehat{\xi}_{23}\theta_3} g_{l_0 l_2}^{-1}(0) \right) \cdots \left(g_{l_0 l_{n-1}}(0) e^{\widehat{\xi}_{n-1,n}\theta_n} g_{l_0 l_{n-1}}^{-1}(0) \right) g_{l_0 t}(0). \quad (3.9)$$

```
T =

    0.8883    0.3511   -0.2962  368.3227
    0.3123   -0.9345   -0.1710  637.9536
   -0.3368    0.0594   -0.9397 -283.6111
         0         0         0    1.0000
```

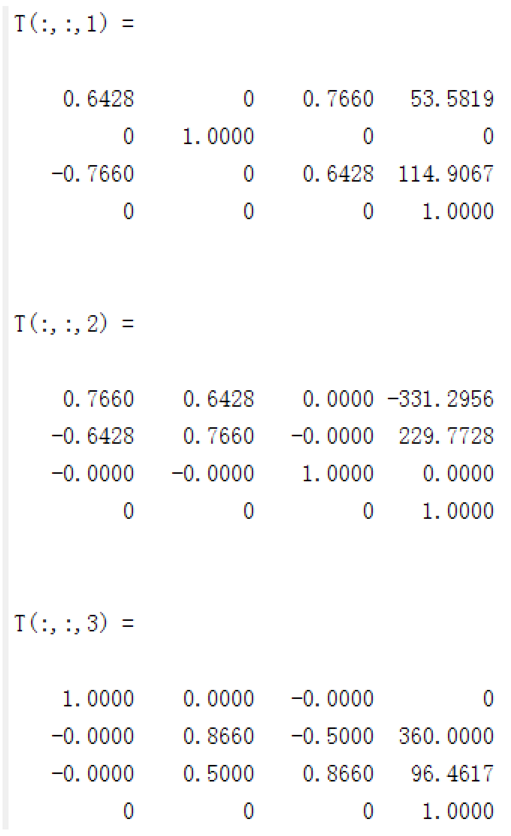
The DH method: each transformation matrix.

```
Tdh(:,:,1) =

    0.5000   -0.0000   -0.8660   75.0000
    0.8660    0.0000    0.5000  129.9038
         0   -1.0000    0.0000         0
         0         0         0    1.0000


Tdh(:,:,2) =

    0.7660    0.6428         0  436.6453
   -0.6428    0.7660         0 -366.3889
         0         0    1.0000         0
         0         0         0    1.0000


Tdh(:,:,3) =

    0.7660   -0.0000   -0.6428  114.9067
    0.6428    0.0000    0.7660   96.4181
         0   -1.0000    0.0000         0
         0         0         0    1.0000


Tdh(:,:,4) =

    0.8660   -0.0000    0.5000         0
    0.5000    0.0000   -0.8660         0
         0    1.0000    0.0000  650.0000
         0         0         0    1.0000


Tdh(:,:,5) =

    0.9397   -0.0000   -0.3420         0
    0.3420    0.0000    0.9397         0
         0   -1.0000    0.0000         0
         0         0         0    1.0000


Tdh(:,:,6) =

    0.9848   -0.1736         0         0
    0.1736    0.9848         0         0
         0         0    1.0000         0
         0         0         0    1.0000
```

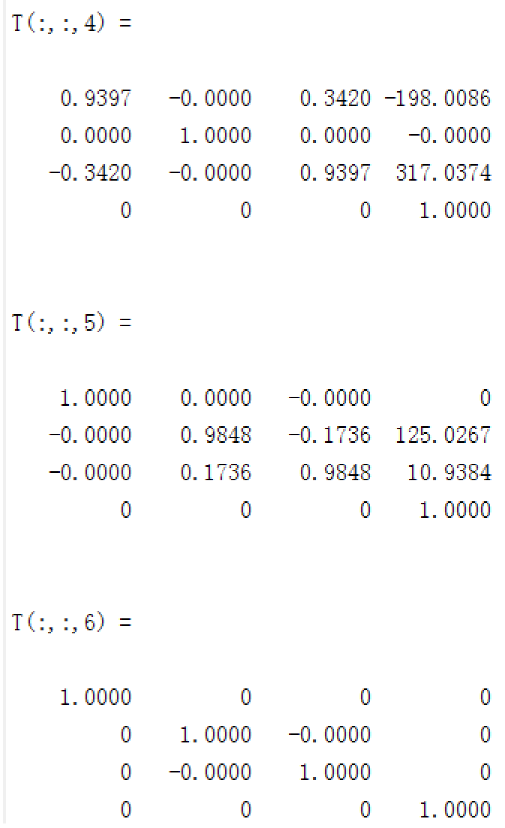
The POE method: each matrix.

```
Tpoe(:,:,1) =

    0.5000   -0.8660         0         0
    0.8660    0.5000         0         0
         0         0    1.0000         0
         0         0         0    1.0000


Tpoe(:,:,2) =

    0.6428         0    0.7660   53.5819
         0    1.0000         0         0
   -0.7660         0    0.6428  114.9067
         0         0         0    1.0000


Tpoe(:,:,3) =

    0.7660         0    0.6428 -331.2956
         0    1.0000         0         0
   -0.6428         0    0.7660  229.7728
         0         0         0    1.0000


Tpoe(:,:,4) =

    1.0000         0         0         0
         0    0.8660   -0.5000  360.0000
         0    0.5000    0.8660   96.4617
         0         0         0    1.0000


Tpoe(:,:,5) =

    0.9397         0    0.3420 -198.0086
         0    1.0000         0         0
   -0.3420         0    0.9397  317.0374
         0         0         0    1.0000


Tpoe(:,:,6) =

    1.0000         0         0         0
         0    0.9848   -0.1736  125.0267
         0    0.1736    0.9848   10.9384
         0         0         0    1.0000
```

Note that Tpoe7 is eye (4).
From the data above,
We conclude that: neither the position nor the orientation of the corresponding matrix is the same. We can use the formula to make connections between the two modeling methods.
The code is called googoldhtopoe.m. The result:



```
T(:,:,1) =

    0.6428         0    0.7660   53.5819
         0    1.0000         0         0
   -0.7660         0    0.6428  114.9067
         0         0         0    1.0000


T(:,:,2) =

    0.7660    0.6428    0.0000 -331.2956
   -0.6428    0.7660   -0.0000  229.7728
   -0.0000   -0.0000    1.0000    0.0000
         0         0         0    1.0000


T(:,:,3) =

    1.0000    0.0000   -0.0000         0
   -0.0000    0.8660   -0.5000  360.0000
   -0.0000    0.5000    0.8660   96.4617
         0         0         0    1.0000
```

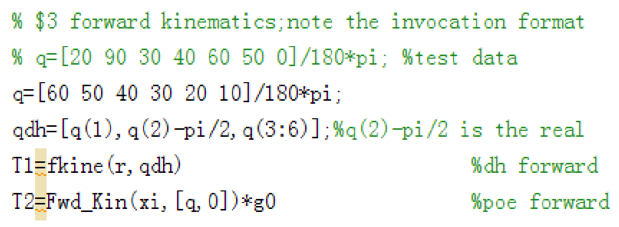


```
T(:,:,4) =

    0.9397   -0.0000    0.3420 -198.0086
    0.0000    1.0000    0.0000   -0.0000
   -0.3420   -0.0000    0.9397  317.0374
         0         0         0    1.0000


T(:,:,5) =

    1.0000    0.0000   -0.0000         0
   -0.0000    0.9848   -0.1736  125.0267
   -0.0000    0.1736    0.9848   10.9384
         0         0         0    1.0000


T(:,:,6) =

    1.0000         0         0         0
         0    1.0000   -0.0000         0
         0   -0.0000    1.0000         0
         0         0         0    1.0000
```

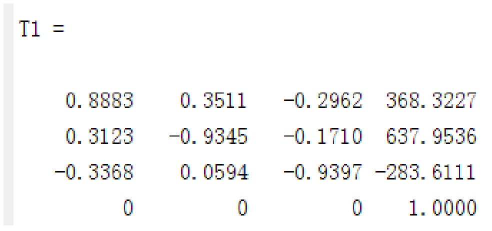
By comparison with Tpoe, we have shown the formula:

(3). Comparison between DH and POE in the inverse kinematics.

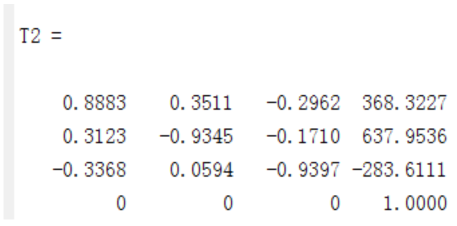
A significant file that compares the inverse kinematics is googolinversecom.m
We first reexamine the forward kinematics:



```
% $3 forward kinematics;note the invocation format
% q=[20 90 30 40 60 50 0]/180*pi; %test data
q=[60 50 40 30 20 10]/180*pi;
qdh=[q(1),q(2)-pi/2,q(3:6)];%q(2)-pi/2 is the real
T1=fkine(r,qdh)                       %dh forward
T2=Fwd_Kin(xi,[q,0])*g0               %poe forward
```

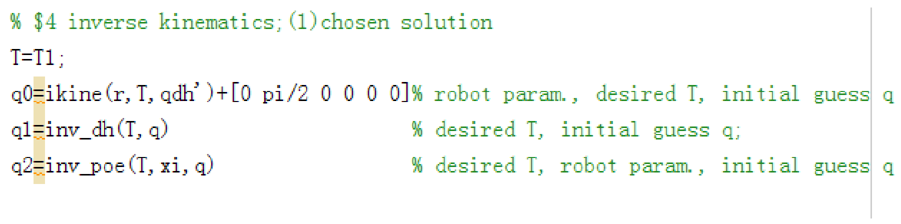


```
T1 =

    0.8883    0.3511   -0.2962  368.3227
    0.3123   -0.9345   -0.1710  637.9536
   -0.3368    0.0594   -0.9397 -283.6111
         0         0         0    1.0000
```

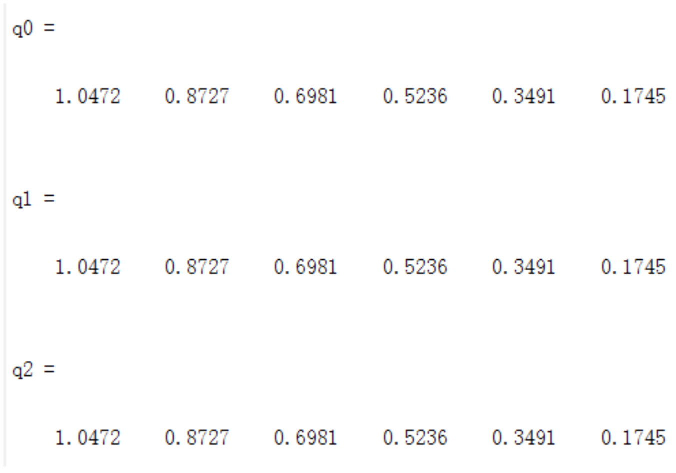


```
T2 =

    0.8883    0.3511   -0.2962  368.3227
    0.3123   -0.9345   -0.1710  637.9536
   -0.3368    0.0594   -0.9397 -283.6111
         0         0         0    1.0000
```

Then, we verify the inverse kinematics respectively in numerical, DH, POE methods. Note that three results are the same in many situations. However, in others situations, as we do not define the limits of the limits of the revolute joints, the solution of numerical method may have some difference. The three methods for inverse kinematics are respectively ikine.m, inv_dh.m, inv_poe.m. We can separately verify the inverse via fkine.m, for_dh.m, for_poe.m

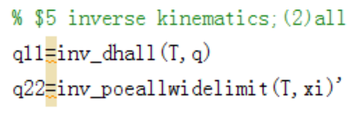


```
% $4 inverse kinematics;(1)chosen solution
T=T1;
q0=ikine(r,T,qdh')+[0 pi/2 0 0 0 0]% robot param., desired T, initial guess q
q1=inv_dh(T,q)                     % desired T, initial guess q;
q2=inv_poe(T,xi,q)                 % desired T, robot param., initial guess q
```

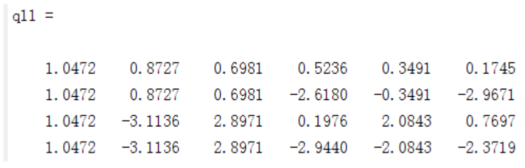


```
q0 =

    1.0472    0.8727    0.6981    0.5236    0.3491    0.1745


q1 =

    1.0472    0.8727    0.6981    0.5236    0.3491    0.1745


q2 =

    1.0472    0.8727    0.6981    0.5236    0.3491    0.1745
```

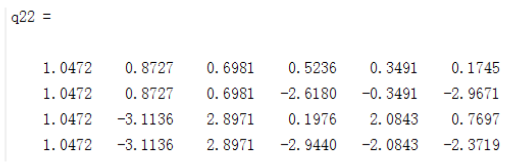
Further, we should compare all the solutions in two methods. Inv_dhall.m , inv_poeallwidelimit.m
The numerical method can only solve one convergent value. Therefore, we only list the results of DH and POE
The results with four solutions are the same. As the 4th， 5th， 6th joints intersect, the number of inverse solutions are reduced form 16 to 8. Further, there is no distance between the 3th and 4th joints in the y direction, the possible number are reduced to 4. (Analysis of number of solutions)



```
% $5 inverse kinematics;(2)all solution
q11=inv_dhall(T,q)
q22=inv_poeallwidelimit(T,xi)'
```



```
q11 =

    1.0472    0.8727    0.6981    0.5236    0.3491    0.1745
    1.0472    0.8727    0.6981   -2.6180   -0.3491   -2.9671
    1.0472   -3.1136    2.8971    0.1976    2.0843    0.7697
    1.0472   -3.1136    2.8971   -2.9440   -2.0843   -2.3719
```



```
q22 =

    1.0472    0.8727    0.6981    0.5236    0.3491    0.1745
    1.0472    0.8727    0.6981   -2.6180   -0.3491   -2.9671
    1.0472   -3.1136    2.8971    0.1976    2.0843    0.7697
    1.0472   -3.1136    2.8971   -2.9440   -2.0843   -2.3719
```

Additionally, we list the numerical methods used in the robotics toolbox.

The inverse kinematics of robotics toolbox, is mainly referred to the Richard P.Paul’s article.

$$dT = \begin{bmatrix} dn_x & do_x & da_x & dp_x \\ dn_y & do_y & da_y & dp_y \\ dn_z & do_z & da_z & dp_z \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\Delta_{\text{revolute}} = \begin{bmatrix} 0 & -d\theta & 0 & 0 \\ d\theta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\Delta_T = \begin{bmatrix} 0 & -\delta_z & \delta_y & d_x \\ \delta_z & 0 & -\delta_x & d_y \\ -\delta_y & \delta_x & 0 & d_z \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

llows a prismatic joint then the
, corresponds to a translation al
ordinate frame:

$$\Delta_{\text{prismatic}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & dd \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

$$dT = T * \Delta_T.$$

$$\Delta_{T6} = U_n^{-1} * \Delta_{\text{revolute}} * U_n \qquad (10)$$

and

$$\Delta_{T6} =$$

$$\begin{bmatrix} 0 & o_x n_y - o_y n_x & a_x n_y - a_y n_x & p_x n_y - p_y n_x \\ n_x o_y - n_y o_x & 0 & a_x o_y - a_y o_x & p_x o_y - p_y o_x \\ n_x a_y - n_y a_x & o_x a_y - o_y a_x & 0 & p_x a_y - p_y a_x \\ 0 & 0 & 0 & 0 \end{bmatrix} d\theta_n \quad ,$$

$$\begin{bmatrix} T_6 d_x \\ T_6 d_y \\ T_6 d_z \\ T_6 \delta_x \\ T_6 \delta_y \\ T_6 \delta_z \end{bmatrix} = \begin{bmatrix} J \end{bmatrix} * \begin{bmatrix} dq_1 \\ dq_2 \\ dq_3 \\ dq_4 \\ dq_5 \\ dq_6 \end{bmatrix}$$

As is shown in the fourth part, the
It is sometimes possible to invert the Jacobian symbolically but this is difficult as the expression of the elements of the Jacobian are quite complicated. A numeric solution to (16) can be obtained, but this is typically far too slow and complicated by the Jacobian becoming singular whenever the manipuLlator becomes degenerate.
More details are introduced in the next part.

Introduction, analysis and simulation of the three kinds of singularities are shown in files singularities_sb, including the analysis of the inverse kinematics of the toolbox.(The test file is googolkinematicsdh)
And the last explanation is shown in the file singularity report.docx

Undoubtedly, the numerical method wastes more time and calculation than the closed-form solutions given by DH and POE mathods. The distinction between the latter two methods are shown below.

2. Comparison between the DH and POE methods

As is shown that the specific values of the forward and inverse kinematics are only related to the mechanical structure of the manipulators including motion limits of the joints. If the modeling method is correct and proper, **the solutions should be the same**. Especially when we have solved the closed-form inverse kinematics, the only job we should do is to transform the two kinds of formula into codes. Therefore, **there is no obvious merits algorithmically when we get the closed form solutions considering the calculation time and computational complexity**.
The DH and POE solve the same four solutions above, which verifies the conclusion.

However, it is still necessary to compare the advantages and disadvantages.

(1). In DH modeling, we should continuously transform the coordinates. We should consider the initial state of the joint variable. Whereas, the POE modelling has only two coordinates-the spatial frame and the tool frame. For instance, you may need to substitute theta-pi/2 for theta as the joint variables, which makes it more complex to analysis what is on earth the joint variable. You may easily forget to add more minus pi/2. The intuitive distinction is that in DH, we pretend to **make mistakes**, try and error to figure out the solutions.

(2).The geometry meaning of POE modeling makes the expressions and variables digestible. You may not need to know what the angles between joints mean. In fact, the concept is complicated when it comes to the number of the joints and the number of the variables. More importantly, it might not have practical significance. The expression in DH, for example, the wrist makes you easy to know what the orientation of each axes. The choice of parameter in POE is more intuitive. One example is that we are easier to obtain the singularities through the dependency of the vector ξ.

(3).POE modeling method is more convenient and more stylized to in the procedure of purchasing the inverse solution. Both methods are utilizing the thought bracket separation. But we need to work out the inverse form the complex sin-cos form equations. The method may leave out several solutions. Sometimes, you may need to use some tips to simplify the equations, which demands mathematical skills.

(4).In trajectory planning and error analysis, the calculation of POE is obvious prior to the DH modeling method. The related research is expected to be carried out. Recall the relationship between the DH and POE, they can be converted utilizing one formula, however, the expressions in POE seems more natural in both practical significance and mathematically calculating.