KUKA KR 6R 700 sixx HM-SC İleri-Ters Kinematik ve Jakobiyen Matrisi Hesaplanması ve GUI Tasarımı

Mekatronik Mühendisliği Bölümü

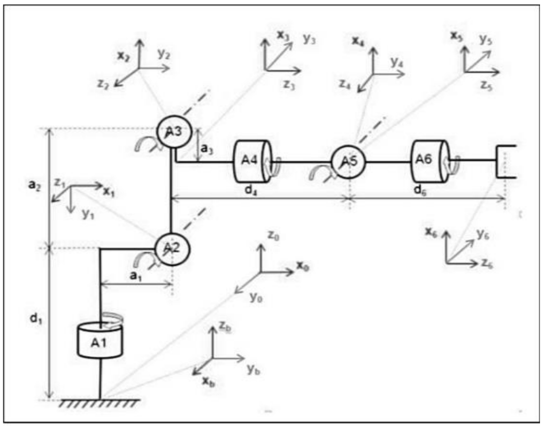
Kocaeli Üniversitesi,KOCAELİ

Özhan YILDIZ 150224074 *ozhan.yldz@gmail.com*

Robotun tanıtımı(Özet)

Ödevde kullan robot KUKA firması tarafından endüstriyel amaçlı üretilmiş olan KUKA KR 6R 700 sixx HM-SC modelli 6 serbestlik derecesine sahip seri bir robot koludur.

Robotun eklem yerleştirmesi aşağıdaki görüntüdeki gibidir:



D-H Tablosunu çıkarılması

Robotun datasheet'inden yararlanarak uzuv uzunlukları veya eklem kaçıklıkları bulunur ve yukarıdaki koordinat yerleştirmesinden yararlanılarak aşağıdaki D-H tablosu çıkarılır:

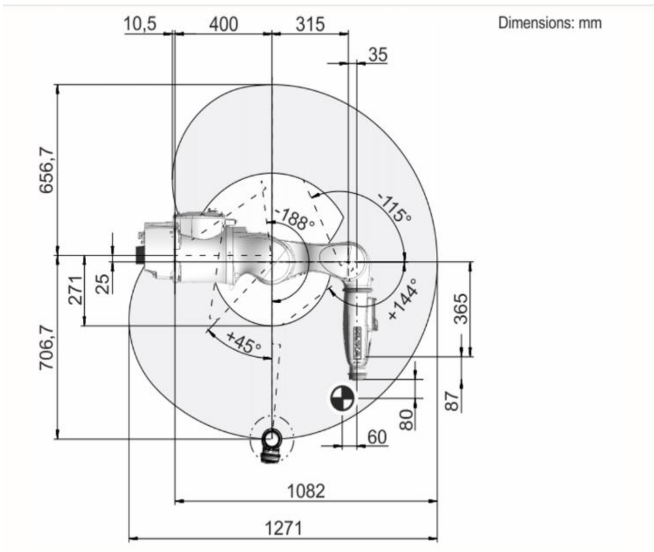
i	α_{i-1}	a_{i-1}	d_i	θ_i
1	-90	a_1	d_1	θ_1
2	0	a_2	0	θ_2
3	90	a_3	0	θ_3
4	90	0	d_4	θ_4
5	-90	0	0	θ_5
6	180	0	d_6	θ_6

Bu tablodaki sembolik değerler aşağıdaki gibidir ve robotun hangi uzuv uzunluğuna tekabül ettikleri aşağıdaki iki boyutlu teknik resimden bakılabilir:

a_1 = 260 mm a_2 = 680 mm

a_3 = 35 mm d_1 = 675 mm

d_4 = 670 mm d_6 = 115 mm



İleri Kinematiğin hesaplanması

D-H tablosundaki değerlere bakılarak her bir eklemin transformasyon matrisi aşağıdaki gibi elde edilmiştir:

$$ {}^{0}_{1}T = \begin{bmatrix} \cos(\theta_1) & 0 & -\sin(\theta_1) & a_1 * \cos(\theta_1) \\ \sin(\theta_1) & 0 & -\cos(\theta_1) & a_1 * \sin(\theta_1) \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{1}_{2}T = \begin{bmatrix} \cos(\theta_2) & 0 & -\sin(\theta_2) & a_2 * \cos(\theta_2) \\ \sin(\theta_2) & 0 & -\cos(\theta_2) & a_2 * \sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{2}_{3}T = \begin{bmatrix} \cos(\theta_3) & 0 & -\sin(\theta_3) & a_3 * \cos(\theta_3) \\ \sin(\theta_3) & 0 & -\cos(\theta_3) & a_3 * \sin(\theta_3) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{3}_{4}T = \begin{bmatrix} \cos(\theta_4) & 0 & -\sin(\theta_4) & 0 \\ \sin(\theta_4) & 0 & -\cos(\theta_4) & 0 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{4}_{5}T = \begin{bmatrix} \cos(\theta_5) & 0 & -\sin(\theta_5) & 0 \\ \sin(\theta_5) & 0 & -\cos(\theta_5) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

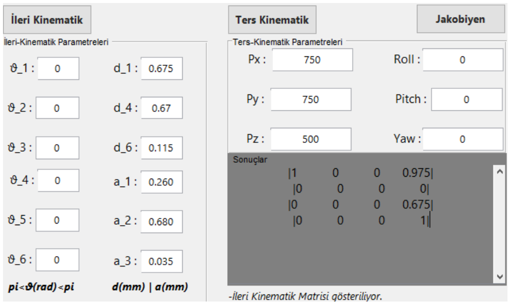
$$ {}^{5}_{6}T = \begin{bmatrix} \cos(\theta_6) & 0 & -\sin(\theta_6) & 0 \\ \sin(\theta_6) & 0 & \cos(\theta_6) & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Bu transformasyon matrisleri Matlab kodları şeklinde yazılarak beş farklı eklem açı grubu için ${}^{0}_{6}T$ ileri kinematiği hesaplanmıştır

MATLAB .m file:

```
clear all;
clc;

syms  theta1 theta2 theta3 theta4 theta5
theta6

d1=.675;     d4=.670;     d6=.115;    a1=.260;
a2=.680;     a3=.035;

T0_1=[cos(theta1)   -sin(theta1)  0  a1  ;  0   0
1 d1  ;
      -sin(theta1)  -cos(theta1)  0  0   ;  0  0
0 1]

T1_2=[cos(theta2)   -sin(theta2)  0  a2  ;
      sin(theta2)   cos(theta2)   0  0  ;
       0  0  1  0   ;  0  0  0  1];

T2_3=[cos(theta3)   -sin(theta3)  0  a3;    0  0
-1  0  ;
      sin(theta3)    cos(theta3)  0  0  ;    0  0
0  1]

T3_4=[cos(theta4)  -sin(theta4)  0  0;    0   0
1  d4  ;
      -sin(theta4)    -cos(theta4)  0  0;    0
0  0  1];

T4_5=[cos(theta5)  -sin(theta5)  0  0  ;
      sin(theta5)   cos(theta5)   0  0  ;
       0  0  1  0   ;  0  0  0  1];

T5_6=[cos(theta6)  -sin(theta6)  0  0  ;
      sin(theta6)    cos(theta6)  0  0  ;
       0  0  1  0   ;  0  0  0  1];
TR_H=T0_1*T1_2*T2_3*T3_4*T4_5*T5_6
```

Eklem açıları ile Matlab'te ileri kinematik matrisi aşağıdaki gibi elde edilmiştir:



Ters Kinematiğin hesaplanması

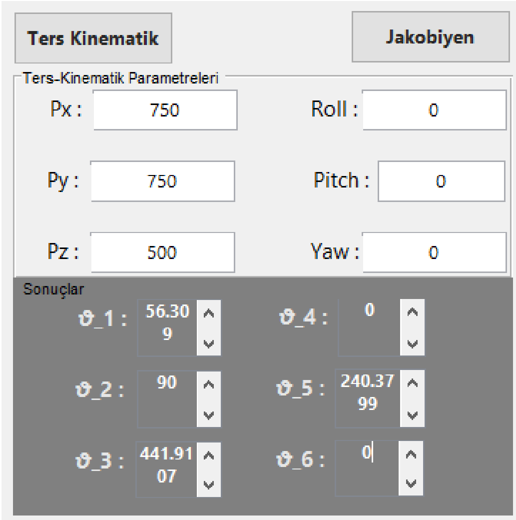
Robotun ters kinematiğinin hesaplanması için aşağıda verilen Matlab kodu kullanılmıştır. Bu kodun daha iyi anlaşılabilmesi için çalışma mantığını açıklayalım:

Öncelikle ileri kinematikteki gibi robotun transformasyon matrisleri yazılır ve ardından uç işlevcisinin gitmesini arzuladığımız konum bilgileri ve bu konumdaki yönelim bilgileri yazılır. Bu bilgiler sembolik bir TR_H ileri kinematik matrisini oluşturmak için kullanılır (bu matris Robot Kinematiği kitabındaki formülden elde edilmiştir). Sonra ilk iki transformasyon matrisinin (T0_1 ile T1_2) tersi alınır ve bu sembolik ileri kinematik matrisiyle çarpılır. Dikkat edilirse TR_H matrisinde bütün elemanların değeri belliyken, tersleri alınan ilk iki transformasyon matrisinin içinde belli olmayan θ açı değişkenleri mevcuttur. Robot Kinematiği kitabındaki teoriğe göre bir sonraki adım bu çarpım sonucu elde edilen matrisin elemanlıyla eşitliğin diğer tarafında kalan T2_6 matrisinin (ki bu matrisinin de elemanları belli olmayan θ açı değişkenleri içermektedir) elemanlarını tek tek birbirine eşitleyip bilinmeyen açıların değerlerini bulmaktır. Bunun için eşitliğin her iki tarafındaki matrislerin karşılıklı bazı elemanları (toplamda 6 tane) seçilip birbirinden çıkartılıp sıfıra eşitlenmiştir. Sonraki adımda bu eşitlikler Matlab'ın çalışma alanından kopyalanıp ayrı bir m-dosyasında oluşturulan bir fonksiyonun elemanları olarak yazılmıştır. Doğrusal olmayan adet adet denklemden ibaret olan bu fonksiyonda aynı zamanda altı adet bilinmeyen θ açı değişkeni mevcuttur. Son adımda da doğrusal olmayan denklemleri çözen bir Matlab fonksiyonuyla bu fonksiyon çözdürülüp bulunan açı değişkenleri çalışma ekranına yazdırılmıştır.

MATLAB .m file:

```
clear all;
clc;

%% Ters kinematik
syms theta1 theta2 theta3 theta4 theta5
theta6  ...
    r11 r12 r13 r21 r22 r23 r31 r32 r33;
T1_6 = T1_2*T2_3*T3_4*T4_5*T5_6;
T1_6 = simplify(T1_6);
T2_6 = T2_3*T3_4*T4_5*T5_6;
T2_6 = simplify(T2_6);
T0_6s = [ r11 r12 r13 px;   r21 r22 r23 py;
...
          r31 r32 r33 pz;   0 0 0 1];
T_eqv1 = inv(T0_1)*T0_6s;
T_eqv2 = inv(T1_2)*T_eqv1;

alfa=pi;
beta=pi;
gama=pi;
RX_Z=[ cos(alfa)*cos(beta)
cos(alfa)*sin(beta)*sin(gama)-
sin(alfa)*cos(gama)
cos(alfa)*sin(beta)*cos(gama)+sin(alfa)*sin
(gama); ...
       sin(alfa)*cos(beta)
sin(alfa)*sin(beta)*sin(gama)+cos(alfa)*cos
(gama) sin(alfa)*sin(beta)*cos(gama)-
cos(alfa)*sin(gama); ...
       -sin(beta) cos(beta)*sin(gama)
cos(beta)*cos(gama)]

R0_6 =T4_5(1:3,1:3)*T5_6(1:3,1:3)
R0_4=T0_1(1:3,1:3)*T1_2(1:3,1:3)*T2_3(1:3,1
:3)*T3_4(1:3,1:3)
R1_6 = simplify(R0_6)
R_eqv4=inv(R0_4)*RX_Z
R_eqv4=simplify(R_eqv4)

eqv1 = simplify(T1_6(1,4)-T_eqv1(1,4));
eqv2 = simplify(T1_6(2,4)-T_eqv1(2,4));
eqv3 = simplify(T1_6(3,4)-T_eqv1(3,4));
eqv4 = simplify(T2_6(1,4)-T_eqv2(1,4));
eqv5 = simplify(T2_6(2,4)-T_eqv2(2,4));
eqv6 = simplify(T2_6(3,4)-T_eqv2(3,4));
theta3=(solve(eqv3,theta3));
eqv11= subs(eqv1,{'theta3'},{theta3});
eqv22= subs(eqv2,{'theta3'},{theta3});
[theta1, theta2] = solve(eqv11, theta1,
theta2);
theta1=double(abs(theta1));
theta2=double(abs(theta2));
theta3=double(abs(theta3));

 R_eqv1 = simplify(R1_6(1,3)-R_eqv4(1,3))
 R_eqv2 = simplify(R1_6(1,2)-R_eqv4(1,2))
 R_eqv3 = simplify(R1_6(2,1)-R_eqv4(2,1))
 theta4=angle(abs(solve(R_eqv1,theta4)))
  R_eqv22= subs(R_eqv2,{'theta4'},{theta4})
[theta5, theta6] = solve(R_eqv22, theta5,
theta6)
% theta4=(abs(theta4)*180/pi)
% theta5=(abs(theta5)*180/pi)
% theta6=(abs(theta6)*180/pi)
Px,Py,Pz değerleri yerine yazıldğında
robotun eklem açıları elde edilmiştir.
```

Eklem açıları ile Matlab'te ileri kinematik matrisi aşağıdaki gibi elde edilmiştir:



-Ters Kinematik-Açı değerleri gösteriliyor.

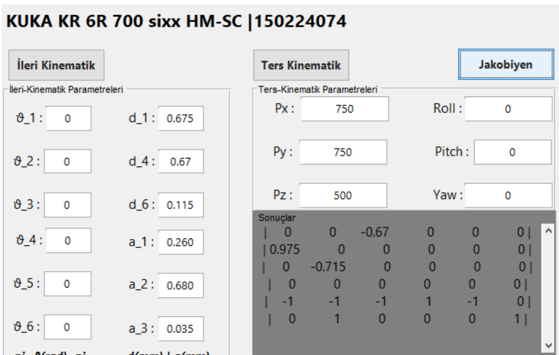
Jakobiyen Matrisinin hesaplanması

Robotun Jakobiyen Matrisinin hesaplanması için aşağıda verilen Matlab kodu kullanılmıştır. Bu kodun daha iyi anlaşılabilmesi için çalışma mantığını açıklayalım:

Transformasyon matrisinde elde ettiğimiz TR_H ileri kinematik matrisi türev alma yöntemi ile DH tablosunda elde ettiğimiz değişken parametrelere göre kismi türev alma yöntemi kullanılarak hesaplanmıştır.

MATLAB .m file:

```
T06_ja=jacobian(TR_H(1:3,4),[theta1;
theta2; theta3; theta4; theta5; theta6])
    A=[0;0;1]
    J0_1= T0_1(1:3,1:3)*A
    J1_2= T1_2(1:3,1:3)*A
    J2_3= T2_3(1:3,1:3)*A
    J3_4= T3_4(1:3,1:3)*A
    J4_5= T4_5(1:3,1:3)*A
    J5_6= T5_6(1:3,1:3)*A
    J0_6=[J0_1 J1_2 J2_3 J3_4     J4_5
J5_6]
    J=[T06_ja;J0_6]
```



-Jakobiyen Matrisi gösteriliyor.