MECHATRONICS ENGINEERING

TYPE-2 SCARA MANIPLATOR

İLERİ KİNEMATİK, TERS KİNEMATİK VE JAKOBİYEN

MATLAB GUI

– UFUK ŞİMŞEK

İçindekiler

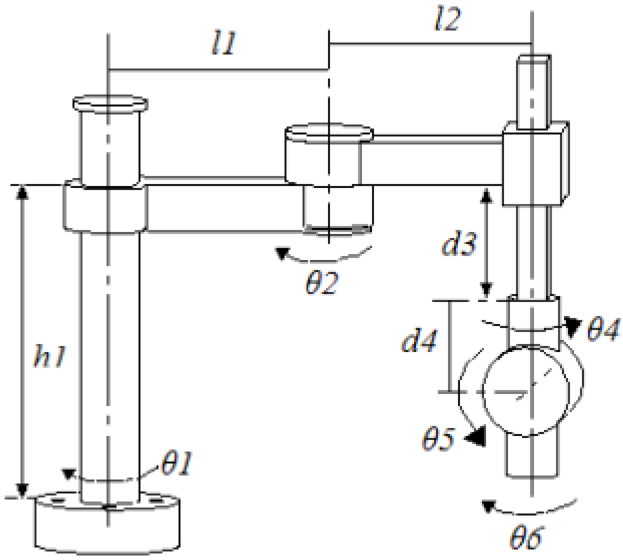
Figürler

TYPE-2 SCARA MANIPLATOR TYPE-2 SCARA MANIPLATOR İLERİ KİNEMATİK, TERS KİNEMATİK VE JAKOBİYEN MATRİSİ

1. GİRİŞ

Bir robotun kolları arasındaki açı veya konum ilişkilerini iki şekilde formüle ederiz. Bunlar ileri kinematik ve ters kinematiktir. İleri kinematikte her bir eklem arasındaki dönüşüm matrisleri sırasıyla birbiriyle çarpılır ve bu matris, orijindeki ana eksenlere göre uç işlevcinin oryantasyonunu ve pozisyonunu belirleriz. Robotun ters kinematiği ise end efektörün oryantasyonu ve pozisyon verileri yardımıyla eklem değişkenlerinin belirlenmesi olarak tanımlanabilir. Sunulan bu çalışmada MATLAB GUI'de geliştirilen bir arayüz bulunmaktadır.

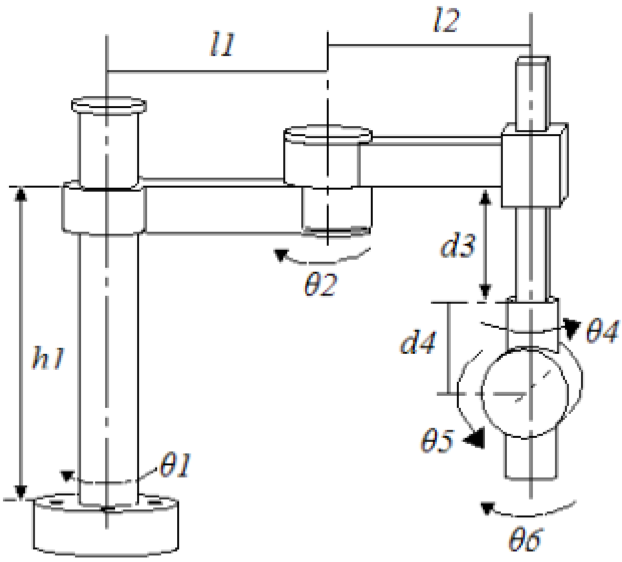
2. TİP-2 SCARA MANIPULATOR



1981 yılında Sankyo Seiki, Pentel ve NEC montaj robotları için tamamen yeni bir konsept sundu. Robot, Yamanashi Üniversitesi'nde profesör olan Hiroshi Makino'nun rehberliğinde geliştirildi. Kolu Z ekseninde rijitti ve XY eksenlerinde bükülebilirdi, bu da XY eksenlerinde deliklere adapte olmasına izin veriyordu. [1] [2]

SCARA'nın ikinci özelliği, insan koluna benzer birleşik iki bağlantılı kol düzenidir. Bu özellik , parçaları bir hücreden diğerine aktarmak veya yükleme / boşaltma işlemleri için avantajlıdır.

3. SKARA MANİPULATOR’ÜN İLERİ KİNEMATİĞİ



Figür 1: Scara Manipulator Tip-2’nin Sabit ve Değişkenleri

İlk olarak ileri kinematik matrislerini bulmak için D-H tablosunu oluşturuyoruz.

EKSEN	**D-H DEĞİŞKENLERİ**			
i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	h_1	θ_1
2	0	l_1	0	θ_2
3	180	l_2	d_3	0
4	0	0	d_4	θ_4
5	-90	0	0	θ_5
6	90	0	0	θ_6

Figür 2: Scara Manipulator Tip-2 D-H Tablosu

Bir sonraki adımda DH tablosu verilerini kullanarak her bir eklem için transformasyon matrisini hesaplayarak uç işlevcinin konum ve oryantasyon bilgisine ulaşacağız.

Bu işlemleri yapacak olan matlab kodu şu şekildedir:

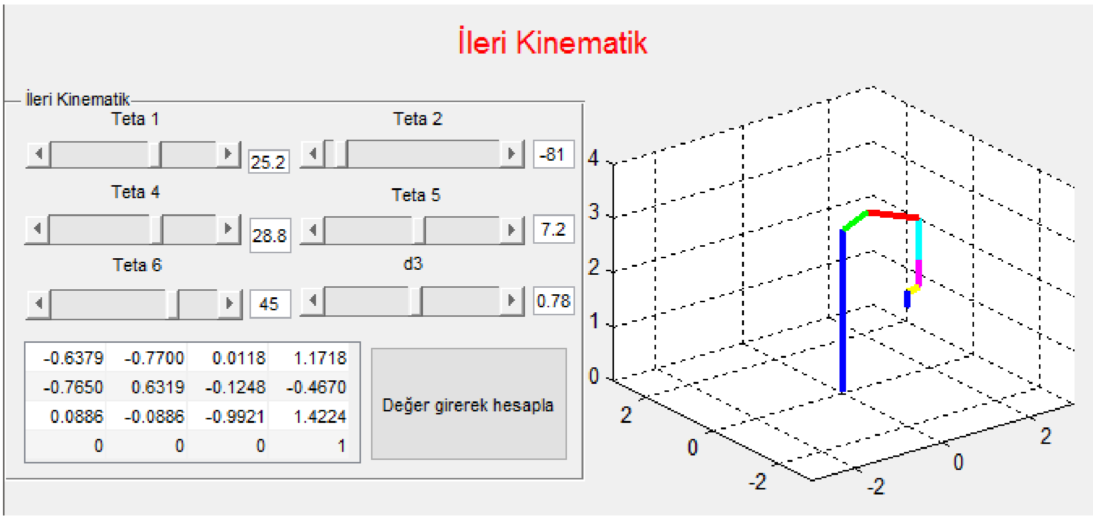
a. Matlab Kodu:

```
function ilerifcn(teta1,teta2,teta4,teta5,teta6, d3)

%________________Tanımlamalar__________________
global T06
h1=3;
d4=0.5;
I1=1;
I2=1;
% d3=1;
% teta1 = 0;
% teta2 = 0;
% teta4 = 0;
% teta5 = 0;
% teta6 = 0;
renk=['b' 'g' 'r' 'c' 'm' 'y' 'b'];

%_______Burda çizimde gözükmesi için bir eklem ekledim
alpha = [0 0 180 0 0 -90 90] ;
a = [0 I1 I2 0 0 0 0];
d = [h1 0 0 d3 d4 0.3 0.3]; % uç işlevcinin belli olması icin
0.3 kaçıklık atadım. 0 verilebilir
t = [teta1 teta2 0 0 teta4 teta5 teta6];

%________________Transfer matrixlerinin hesaplanması (
Structure yapısı
%________________var. .m ile biten normal ileri yol
matrisi________________
for i=1:7
       T(i).m = [cosd(t(i)) -sind(t(i)) 0 a(i);
sind(t(i))*cosd(alpha(i)) cosd(t(i))*cosd(alpha(i)) -
sind(alpha(i)) -sind(alpha(i))*d(i); sind(t(i))*sind(alpha(i))
cosd(t(i))*sind(alpha(i)) cosd(alpha(i)) cosd(alpha(i))*d(i);
0 0 0 1];
end
%_______________ İleri yol matrislerinin çarpımları. Structure
da .p ile kayıtlı__________
T(1).p=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0]; %çizim için 0
konumda gerekli
T(2).p=T(1).m;
T(3).p=T(1).m*T(2).m;
T(4).p=T(1).m*T(2).m*T(3).m;
T(5).p=T(1).m*T(2).m*T(3).m*T(4).m;
T(6).p=T(1).m*T(2).m*T(3).m*T(4).m*T(5).m;
T(7).p=T(1).m*T(2).m*T(3).m*T(4).m*T(5).m*T(6).m;
T(8).p=T(1).m*T(2).m*T(3).m*T(4).m*T(5).m*T(6).m*T(7).m;
T06=T(8).p; %bu aslında önceki t6 ya eşittir

%________________Sonunda döngüde 3d çizim. ________
```

```
for i=1:7
    plot3( [T(i).p(1,4) T(i+1).p(1,4)], [T(i).p(2,4)
T(i+1).p(2,4)], [T(i).p(3,4) T(i+1).p(3,4)], 'LineWidth',3,
'Color', renk(i))
    hold on;
end
grid;
axis([-3 3 -3 3 0 4]);
hold off;

end
```

Figür 3: ilerifcn.m İleri Kinamtik Hesapları Fonksiyonu

Yukarıdaki fonksiyon burada kullanılmıştır:



Figür 4: İleri Kinematik Arayüzü

4. Scara Manipulator'ün Ters Kinematiği

İleri kinematik çözümünden elde edilen transformasyon matrisleri şu şekildedir:

$$
{}^{0}_{1}T = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & h_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\qquad
{}^{3}_{4}T = \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & 0 \\ s\theta_4 & c\theta_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{1}_{2}T = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & l_1 \\ s\theta_2 & c\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\qquad
{}^{4}_{5}T = \begin{bmatrix} c\theta_5 & -s\theta_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_5 & -c\theta_5 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{2}_{3}T = \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\qquad
{}^{5}_{6}T = \begin{bmatrix} c\theta_6 & -s\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ s\theta_6 & c\theta_6 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Figür 5: Transformasyon Matrisleri

$$\theta_1 = Arctan\left(\frac{p_y}{p_x}\right) \pm Arctan\left(\frac{\sqrt{4p_y{}^2 l_1{}^2 + 4p_x{}^2 l_1{}^2 - c^2}}{c}\right) \qquad \theta_2 = Arctan\left(\frac{-s(\theta_1)p_x + c(\theta_1)p_y}{c(\theta_1)p_x + s(\theta_1)p_y - l_1}\right)$$

$$\theta_4 = Arctan\left(\frac{r_{23}c(\theta_1+\theta_2) - r_{13}s(\theta_1+\theta_2)}{-r_{13}c(\theta_1+\theta_2) - r_{23}s(\theta_1+\theta_2)}\right)$$

$$\theta_5 = Arctan \pm \left(\frac{\sqrt{1 - r_{33}{}^2}}{-r_{33}}\right)$$

$$\theta_6 = Arctan\left(\frac{-r_{32}}{r_{31}}\right)$$

Figür 6: Açıların ters kinematik çözümleri

a. Matlab Kodu:

```
function tersfcn(px , py , pz , alfa , beta , gamma, psbt,
kaciklik)

%_________________Tanımlamalar___________________
global T06
h1=3;
d4=0.5;
I1=1;
I2=1;

syms teta1 teta2 teta4 teta5 teta6 d3;
renk=['b' 'g' 'r' 'c' 'm' 'y' 'b'];

alpha = [0 0 180 0 -90 90] ; % Burda çizdirmedeki fazla eklemi
hızlı olması açısından eklemedim
a = [0 I1 I2 0 0 0];
d = [h1 0 d3 d4 0 0]; % Kaçıklığı burda verirsem denklemler
değişir
t11 = [teta1 teta2 0 teta4 teta5 teta6];

% px = 1;
% py =1;
% pz =1;
% alfa=30;
% beta=60;
% gamma=89;

%_________________Transfer matrixlerinin hesaplanması (
Structure yapısı
```

```
%________________var. .m ile biten normal ileri yol
matrisi________________
for i=1:6
       T(i).m = [cos(t11(i)) -sin(t11(i)) 0 a(i);
sin(t11(i))*cosd(alpha(i)) cos(t11(i))*cosd(alpha(i)) -
sind(alpha(i)) -sind(alpha(i))*d(i);
sin(t11(i))*sind(alpha(i)) cos(t11(i))*sind(alpha(i))
cosd(alpha(i)) cosd(alpha(i))*d(i); 0 0 0 1];
end

%_______________ İleri yol matrislerinin çarpımları. Structure
da .p ile kayıtlı__________
T(01).p=T(1).m;
T(02).p=T(1).m*T(2).m;
T(03).p=T(1).m*T(2).m*T(3).m;
T(04).p=T(1).m*T(2).m*T(3).m*T(4).m;




T(05).p=T(1).m*T(2).m*T(3).m*T(4).m*T(5).m;
T(06).p=T(1).m*T(2).m*T(3).m*T(4).m*T(5).m*T(6).m;
T06=T(06).p;

%% Dönüşüm matrisinin saf hali rotasyonların bulunması için uç
işlevcinin açısı girilmelidir
r11 = cosd(alfa)*cosd(beta);
r12 = cosd(alfa)*sind(beta)*sind(gamma)-
sind(alfa)*sind(gamma);
r13 =
cosd(alfa)*sind(beta)*cosd(gamma)+sind(alfa)*sind(gamma);
r21 = sind(alfa)*cosd(beta);
```

```
r22 =
sind(alfa)*sind(beta)*sind(gamma)+cosd(alfa)*cosd(gamma);
r23 = sind(alfa)*sind(beta)*cosd(gamma)-
cosd(alfa)*sind(gamma);
r31 = -sind(beta);
r32 = cosd(beta)*sind(gamma);
r33 = cosd(beta)*cosd(gamma);

T06s = [r11 r12 r13 px; r21 r22 r23 py; r31 r32 r33 pz; 0 0 0
1];

%__________ Birkaç ters alıp karşıya atılmış denlemin
oluşturulması __________
T(1).im = inv(T(1).m);
T(2).im = inv(T(2).m);
T(3).im = inv(T(3).m);
% T(4).im = inv(T(4).m);
% T(5).im = inv(T(5).m);
```

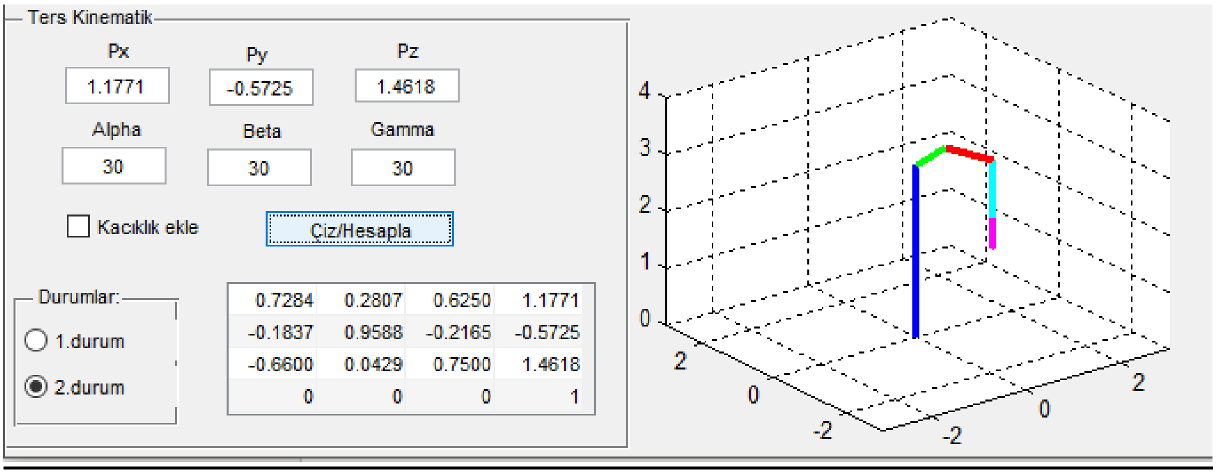
```
left1=T(1).im*T06s;
left2=T(2).im*left1;
left3=T(3).im*left2;
% left4=T(4).im*left3;
% left5=T(5).im*left4;

T(16).p=T(2).m*T(3).m*T(4).m*T(5).m*T(6).m;
T(26).p=T(3).m*T(4).m*T(5).m*T(6).m;
T(36).p=T(4).m*T(5).m*T(6).m;
% T(46).p=T(5).m*T(6).m;
% T(56).p=T(6).m;


eqv1 = simplify(T(16).p(1:3,4)-left1(1:3,4));
% eqv2 = simplify(T(26).p(1:3,4)-left2(1:3,4))
% eqv3 = simplify(T(36).p(1:3,4)-left3(1:3,4))




% eqv4 = simplify(T(46).p(1:3,4)-left4(1:3,4))
% eqv5 = simplify(T(56).p(1:3,4)-left5(1:3,4))

d3r=solve(eqv1(3),d3); % d3 kolay bir şekilde bulunur tek
olasılığı vardır

[teta1r teta2r ] = solve(eqv1(1), eqv1(2), teta1, teta2);
teta1r=double(teta1r); % Teta 1 ve 2 iki olasılığa sahip
teta2r=double(teta2r);
```

```
%eqv31 = simplify(T(36).p(1:3,1)-left3(1:3,1))
eqv32 = (T(36).p(1:3,2)-left3(1:3,2));
eqv32 = subs(eqv32,{'teta1' , 'teta2', 'd3'},{teta1r(psbt),
teta2r(psbt), d3r});
eqv33 = (T(36).p(1:3,3)-left3(1:3,3));
eqv33 = subs(eqv33,{'teta1' , 'teta2', 'd3'},{teta1r(psbt),
teta2r(psbt), d3r});
% Euler bileğinin açılarıda bulunur
[teta4r teta5r ] = solve(eqv33(1), eqv33(2), teta4, teta5);
teta4r=double(teta4r);
teta5r=double(teta5r);

eqv32 = subs(eqv32,{'teta1' , 'teta2', 'd3', 'teta4',
'teta5'},{teta1r(psbt), teta2r(psbt), d3r, teta4r(1),
teta5r(1)});
[teta6r ] = solve(eqv32(1), teta6);
teta6r=double(teta6r);

%____________Bundan sonra çizdirme______________
```

```
t1=teta1r(psbt);
t2=teta2r(psbt);
t4=teta4r(1);
t5=teta5r(1);
t6=teta6r(1);

%_______Burda çizimde gözükmesi için bir eklem ekledim
alpha = [0 0 180 0 0 -90 90] ;
a = [0 I1 I2 0 0  0 0];
d = [h1 0 0 d3r d4 kaciklik kaciklik ];  % uç işlevcinin belli
olması icin 0.2 kaçıklık atadım. 0 verilebilir
t11 = [t1 t2 0 0 t4 t5 t6];







%________________Transfer matrixlerinin hesaplanması (
Structure yapısı
%________________var. .m ile biten normal ileri yol
matrisi________________
for i=1:7
       T(i).m = [cos(t11(i)) -sin(t11(i)) 0 a(i);
sin(t11(i))*cosd(alpha(i)) cos(t11(i))*cosd(alpha(i)) -
sind(alpha(i)) -sind(alpha(i))*d(i);
sin(t11(i))*sind(alpha(i)) cos(t11(i))*sind(alpha(i))
cosd(alpha(i)) cosd(alpha(i))*d(i); 0 0 0 1];
end
```

```

%_______________ İleri yol matrislerinin çarpımları. Structure
da .p ile kayıtlı__________
T(01).p=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0]; %çizim için 0
konumda gerekli
T(02).p=T(1).m;
T(03).p=T(1).m*T(2).m;
T(04).p=T(1).m*T(2).m*T(3).m;
T(05).p=T(1).m*T(2).m*T(3).m*T(4).m;
T(06).p=T(1).m*T(2).m*T(3).m*T(4).m*T(5).m;
T(07).p=T(1).m*T(2).m*T(3).m*T(4).m*T(5).m*T(6).m;
T(08).p=T(1).m*T(2).m*T(3).m*T(4).m*T(5).m*T(6).m*T(7).m;
T06=double(T(08).p) %bu aslında önceki t6 ya eşittir

%________________Sonunda döngüde 3d çizim. ________
    for i=1:7
```

```
        plot3( [T(i).p(1,4) T(i+1).p(1,4)], [T(i).p(2,4) T(i+1).p(2,4)], [T(i).p(3,4) T(i+1).p(3,4)], 'LineWidth',3, 'Color', renk(i))
        hold on;
    end
    hold off;
grid;
axis([-3 3 -3 3 0 4]);
end
```



Figür 7: Ters Kinematik Arayüzü

5. Scara Manipulator Tip-2 Jakobiyen Matrisi

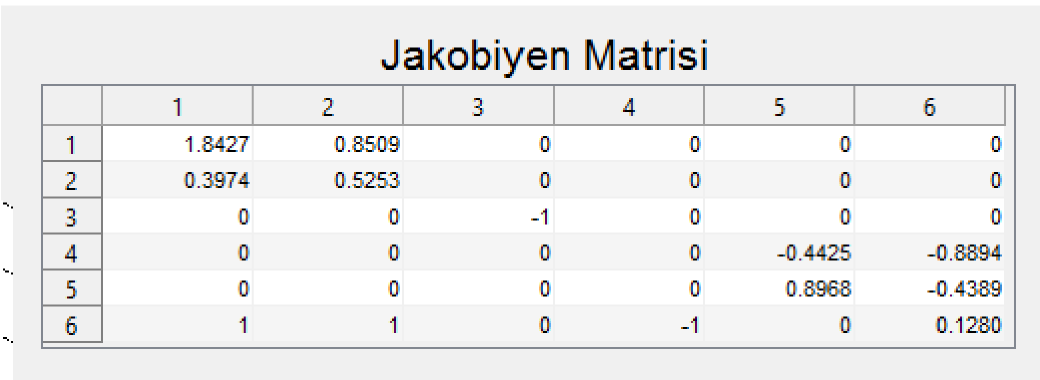
Ters Kinematik Problemleri analitik yöntemler olmadan çözülebilir. Jakobiyen matrisi, eklemlerin açısındaki değişikliklere göre end efektördeki hız değişiklikleri gösterir. Jakobiyen matrisi için bir fonksiyon geliştirmedim. Basitçe, şimdi bir tabloda gösterdim.

GUI m dosyasında.

a. Matlab Kodu:

```
R01 = [cos(teta1) -sin(teta1) 0 ; sin(teta1) cos(teta1) 0; 0 0 1];

R12 = [cos(teta2) -sin(teta2) 0 ; sin(teta2) cos(teta2) 0; 0 0 1];

R23 = [1 0 0 ; 0 1 0; 0 0 -1];
```

```
R34 = [cos(teta4) -sin(teta4) 0 ; sin(teta4) cos(teta4) 0; 0 0
1];

R45 = [cos(teta5) -sin(teta5) 0 ; 0 0 -1; sin(teta5)
cos(teta5) 0];

R56 = [cos(teta6) -sin(teta6) 0 ; 0 0 1; -sin(teta6) -
cos(teta6) 0];

R01=R01;
R02=R01*R12;
R03=R01*R12*R23;
R04=R01*R12*R23*R34;
R05=R01*R12*R23*R34*R45;
R06=R01*R12*R23*R34*R45*R56;

Z01=R01*[0;0;1]
Z02=R02*[0;0;1]
Z03=R03*[0;0;0]
Z04=R04*[0;0;1]
Z05=R05*[0;0;1]
Z06=R06*[0;0;1]

J = [ - sin(teta1) - cos(teta1)*sin(teta2) -
cos(teta2)*sin(teta1), - cos(teta1)*sin(teta2) -
cos(teta2)*sin(teta1), 0, 0, 0, 0;
      cos(teta1) + cos(teta1)*cos(teta2) -
sin(teta1)*sin(teta2),   cos(teta1)*cos(teta2) -
sin(teta1)*sin(teta2), 0, 0, 0, 0;
      0, 0, -1, 0, 0, 0;
      Z01,Z02,Z03,Z04,Z05,Z06]
  set(handles.uitable7,'Data',J);
```

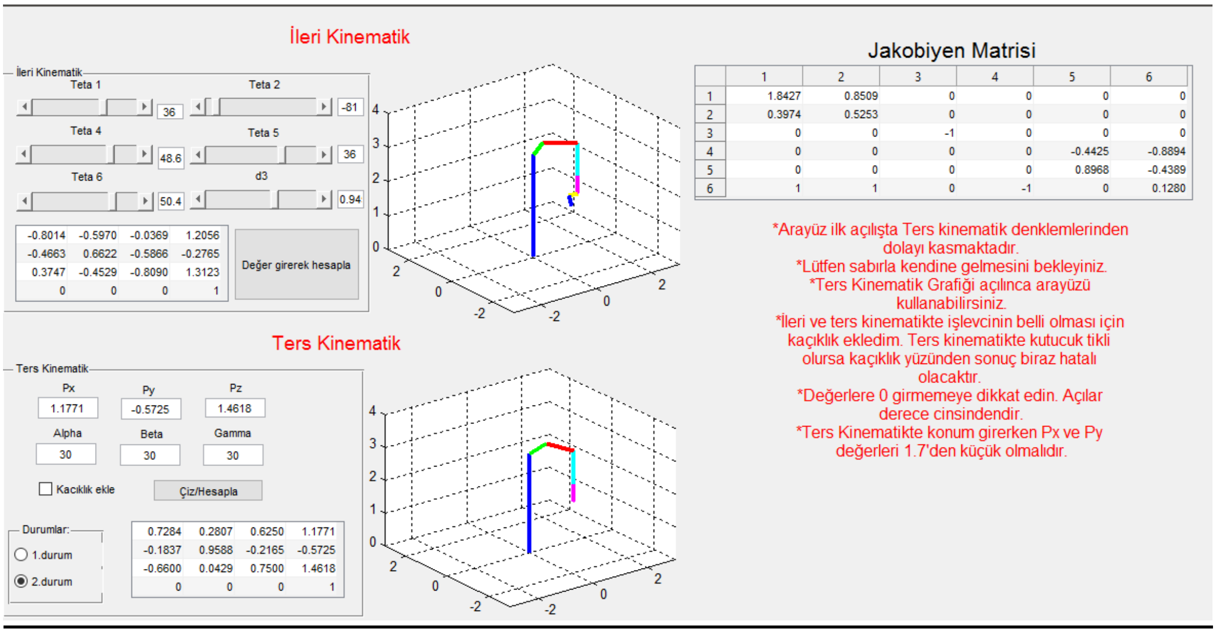


Jakobiyen Matrisi

	1	2	3	4	5	6
1	1.8427	0.8509	0	0	0	0
2	0.3974	0.5253	0	0	0	0
3	0	0	-1	0	0	0
4	0	0	0	0	-0.4425	-0.8894
5	0	0	0	0	0.8968	-0.4389
6	1	1	0	-1	0	0.1280

Figür 8: Jakobiyen Matrisi

6. Scara Manipulator Tip-2 Arayüzü



Referanslar

1. Assembly robot US Pat. 4,341,502 https://docs.google.com/viewer?url=patentimages.storage.googleapis.com/pdfs/US4341502.pdf
2. **Jump up^** *Westerland, Lars (2000). The Extended Arm of Man, A History of the Industrial Robot. ISBN 91-7736-467-8.*
3. *SCARA ROBOT MANİPULATÖR İLE SEÇME VE YERLEŞTİRME UYGULAMASI, Mehmet ÖZKARAKOÇ, Haziran,2009, İZMİR*