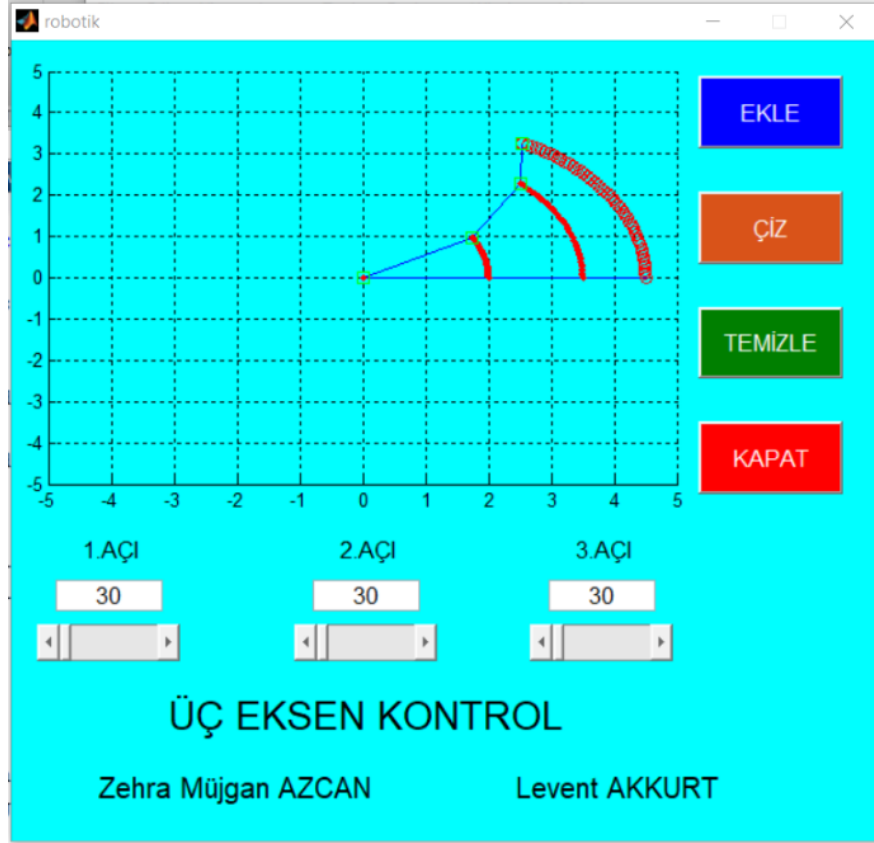


3 LİNKLİ ROBOT KONTROLÜ

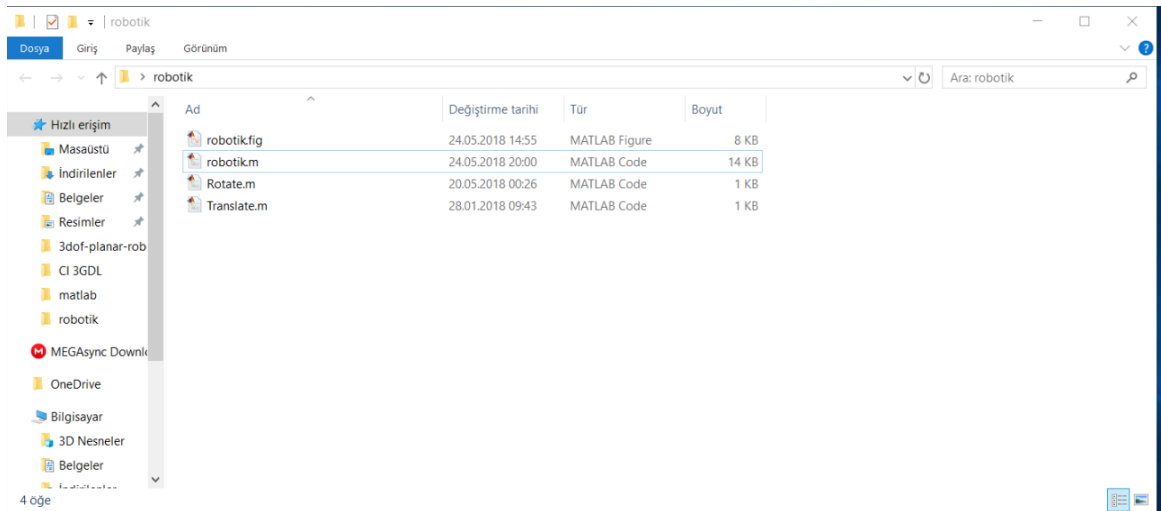
Projenin Amacı:

Üç serbestlik noktasına sahip robot kolun ileri kinematik denklemler yardımıyla girilen joint açısı değerlerine göre çalışmasını simüle etmek. Şekilde programın arayüzü görülmektedir.

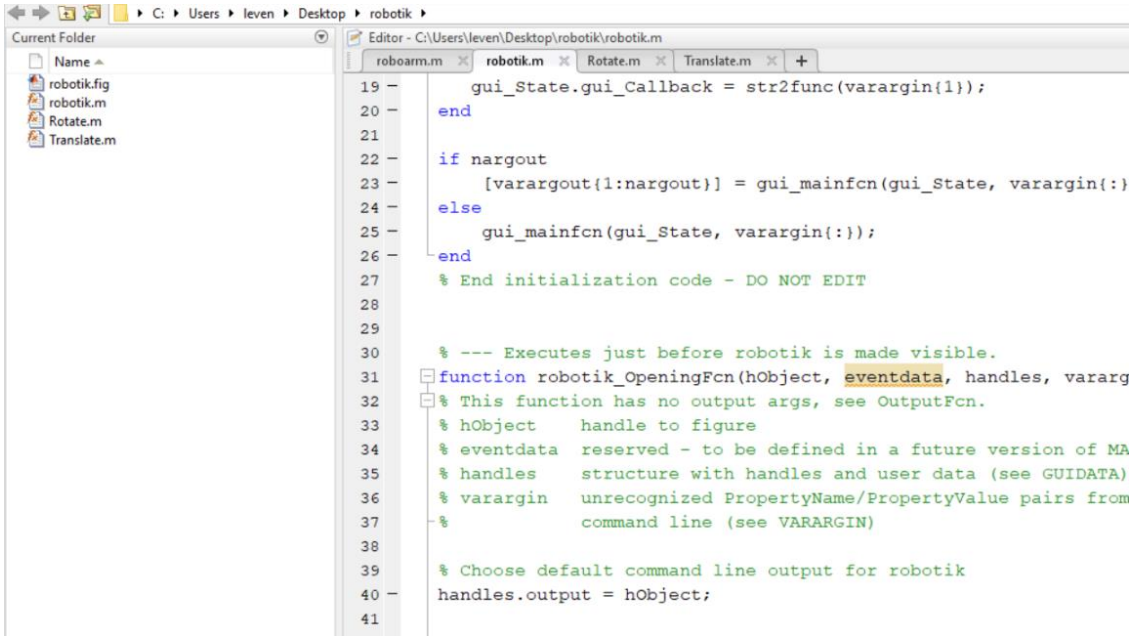


Projenin Çalıştırılması:

1. Dosyaları şekildeki gibi aynı dizine indirin



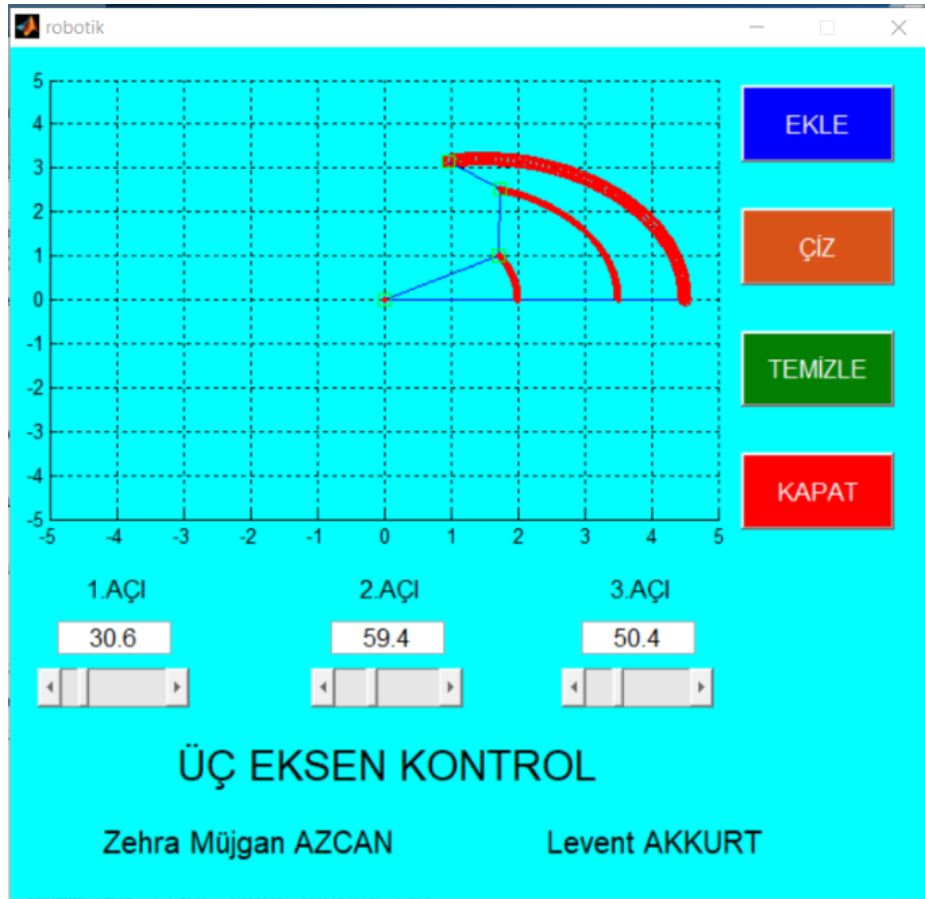
2. Dört dosyayı seçip matlab'da açın



```
19 gui_State.gui_Callback = str2func(varargin{1});
20 end
21
22 if nargin
23     [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:})
24 else
25     gui_mainfcn(gui_State, varargin{:});
26 end
27 % End initialization code - DO NOT EDIT
28
29
30 % --- Executes just before robotik is made visible.
31 function robotik_OpeningFcn(hObject, eventdata, handles, varargin)
32 % This function has no output args, see OutputFcn.
33 % hObject    handle to figure
34 % eventdata  reserved - to be defined in a future version of MA
35 % handles    structure with handles and user data (see GUIDATA)
36 % varargin   unrecognized PropertyName/PropertyValue pairs from
37 %            command line (see VARARGIN)
38
39 % Choose default command line output for robotik
40 handles.output = hObject;
41
```

3. Matlab'da robotik.m dosyasını çalıştırın

4. Açılan arayüzden robot kol joint açılarını girerek robot kol simülasyonunu gerçekleştirebilirsiniz.



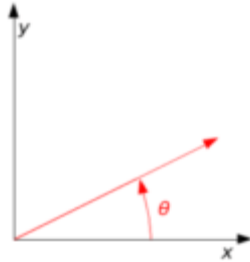
5. Arayüzde bulunan slider butonlarını kaydırarak veya direk değer girerek edit text'teki sayı değerlerini değiştirebilirsiniz.
6. Ekle butonuna tıklayarak edit text değerlerini joint açısı olarak kaydederiz.
7. Son olarak çiz butonu ile girilen açı değerlerine göre robot kol linkleri çizilmiş olur
8. Temizle butonu ile atanan bütün değerler sıfırlanır
9. Kapat butonu ile program kapatılır

Dönme Matrisi:

Doğrusal cebirde bir dönme matrisi, Öklid uzayında bir dönüş gerçekleştirmek için kullanılan matristir. Örneğin, iki boyutlu x-y kartezyen düzlemi içerisinde yer alan noktaları koordinat sisteminin orijini etrafında bir θ açısıyla saat yönünün tersine döndürmek için aşağıdaki gibi bir matris kullanılır.

$$R(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

Dönme işleminden sonraki koordinat noktası (x, y)'den (x', y') konumuna gelmektedir.



Bir vektörün açısı θ üzerinden saatin ters yönünde döndürülmektedir.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$x' = x \cos \theta - y \sin \theta$$

$$y' = x \sin \theta + y \cos \theta$$

Dönme Matrisi Uygulaması: Benzer şekilde üç boyutlu xyz koordinat sistemindeki dönüş hareketi içinde bir dönme matrisi tanımlayabiliriz. Koordinat sistemi eksenlerinden birinin (x, y veya z) etrafındaki dönüşü temel dönüşüm matrisi denir. Aşağıdaki üç temel döndürme matrisi, x, y veya z eksen etrafında θ açısı ile bir vektörü döndürür.

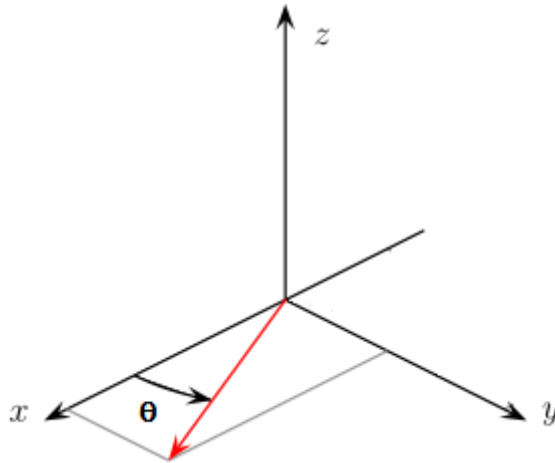
$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

$$R_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Örneğin x eksenine hizalanmış bir vektörü y-eksenine doğru döndürmek için R_z (z ekseninde 90 derecelik dönüş) yapmamız gereklidir. Yani $(1,0,0)$ vektörünü R_z yani z ekseninde 90 derece döndürürsek aşağıdaki gibi bir matris elde ederiz.

$$R_z(90^\circ) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos 90^\circ & -\sin 90^\circ & 0 \\ \sin 90^\circ & \cos 90^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$



Z Eksenine etrafında $[1,0,0]$ vektörünün dönüşü

Robotik Ders Ödevi

131222041 Zehra Müjgan AZCAN

141202119 Levent AKKURT