

SELÇUK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
ELEKTRİK-ELEKTRONİK
MÜHENDİSLİĞİ BÖLÜMÜ
ROBOTİĞE GİRİŞ DERSİ
PROJE ÖDEVİ

3 DOF ROBOTİK KOL MATLAB UYGULAMASI

HAZIRLAYAN

YASİN ATAY

141202042

ÖĞRETİM GÖREVLİSİ

ÖĞR.DR. AKİF DURDU

KONYA 2018

3 DOF ROBOTİK KOL VE MATLAB SİMULASYONU

1.0 KONU VE GİRİŞ

Bu çalışmada üç serbest eksenli(3DOF) bir robot kolun Matlab ortamında Simülasyonu gerçekleştirilmiştir.

Burada hazırlanan robot kol ters ve düz kinematik olarak çalışıyor yani ya robotun olması gerektiği konum bilgisi girilebilir ya da robotun her bir linkinin açı bilgileri girilerek robotun istenen hareketi yapması sağlanabilir.

2.0 TRANSFORMASYON MATRİSİ

Robot kolun herhangi bir konumu ile transformasyon matrisinin çarpılması sonucu robotun bir sonraki konumu bulunur.

Transformasyon matrisini elde edebilmek için bir “A” matrisi oluşturulur. A matrisi her bir link için ayrı ayrı hesaplanır ve çarpılır. A matrisi $A_n = \text{Rot}(z, \text{Theta}) * \text{Trans}(0, 0, d) * \text{Trans}(a, 0, 0) * \text{Rot}(x, \text{Alfa})$ şeklinde hesaplanır. Bu matrisdeki değişkenleri bizim kullandığımız Robot kol için yerine yazarsak ve her bir link için bulduğumuz değerleri çarparsak Robot kolumuz için transformasyon matrisini elde ederiz.

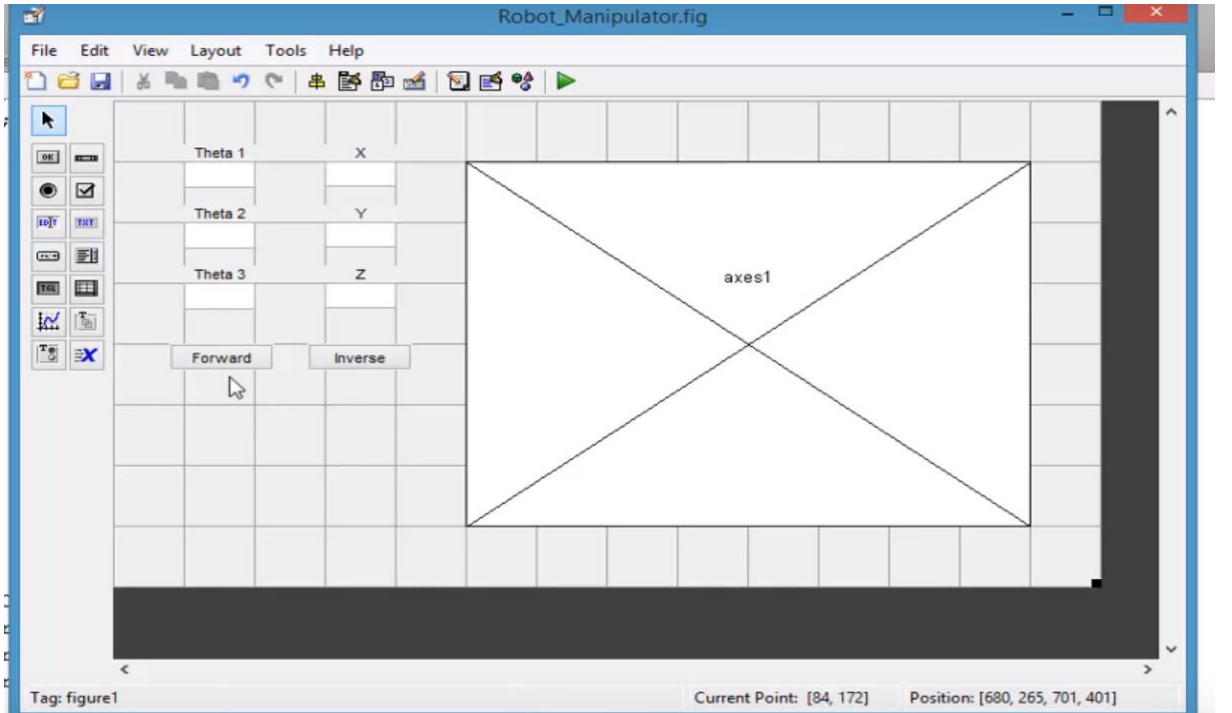
Robot kolumuzun link bilgileri aşağıdaki tablodaki gibidir.

	θ	d	a	α
Link 1	θ_1^*	L1	0	90
Link 2	θ_2^*	0	L2	0
Link 3	θ_3^*	0	L3	0

Bu değerler ile elde hesap yapmak yerine MATLAB programında hesap yapılacaktır.

3.0 MATLAB ARAYÜZÜNÜN OLUŞTURULMASI

Matlab ile uygulama geliştirebilmek için GUI denilen kullanıcı arayüzü uygulaması kullanılacaktır. Bunun için komut satırına guide yazın ve açılan pencerelerden OK tıklayarak arayüze ulaşıldı. Buradan 6 tane “Edit Text” ,2 tane “Push Button” ve 1 tane “Axes” açıldı.



İleri kinematik hesaplarının yapıldığı yere istenildiğinde “Forward” butonuna sağ tıklanıp Callbacks den Editör kısmına ulaşıp hesaplamalarda gerekli değişiklikler yapılabilir.

```
Editor - C:\Users\Yasin Atay\Desktop\Robot\untitled.m
untitled.m
145 % --- Executes on button press in btn_Foward.
146 function btn_Foward_Callback(~, ~, handles)
147 % hObject    handle to btn_Foward (see GCBO)
148 % eventdata  reserved - to be defined in a future version of MATLAB
149 % handles    structure with handles and user data (see GUIDATA)
150
151 %% İleri Kinematik için yapılan hesaplamalar
152
153 Th_1=str2double(handles.Theta_1.String)*pi/180; %"Edit Text" bölümüne girilen string değeri Th_1 açısına çeviren fonksiyon
154 Th_2=str2double(handles.Theta_2.String)*pi/180; %"Edit Text" bölümüne girilen string değeri Th_2 açısına çeviren fonksiyon
155 Th_3=str2double(handles.Theta_3.String)*pi/180; %"Edit Text" bölümüne girilen string değeri Th_3 açısına çeviren fonksiyon
156
157 L_1=20; % "Axes" kısmına çizilecek robot kolun birinci link uzunluğu
158 L_2=50; % "Axes" kısmına çizilecek robot kolun ikinci link uzunluğu
159 L_3=40; % "Axes" kısmına çizilecek robot kolun üçüncü link uzunluğu
160
161 L(1)=Link([0 L_1 0 pi/2]); % Robot kol için hesaplanan 1.link parametreleri
162 L(2)=Link([0 0 L_2 0]); % Robot kol için hesaplanan 2.link parametreleri
163 L(3)=Link([0 0 L_3 0]); % Robot kol için hesaplanan 3.link parametreleri
164
165 Robot=SerialLink(L); % Yukarıda hesabı yapılan bu üç linkin birleştirip robot kol haline getiren fonksiyon
166 Robot.name='Yasin_Robot'; % Robot kol'a verdiğimiz isim
167 Robot.plot([Th_1 Th_2 Th_3]); % Hesaplanan açılara göre robot kol'u "Axes" kısmına çizen fonksiyon
168
169 T = Robot.fkine([Th_1 Th_2 Th_3]); % Robot kolun transformasyon matrisi
170 handles.Pos_X.String=num2str(floor(T(1,4))); % Hesaplanan açı değerlerinin string türüne dönüştürülüp Pos_X değişkenine atayıp ilgili "Edit Text"e yazar
171 handles.Pos_Y.String=num2str(floor(T(1,4))); % Hesaplanan açı değerlerinin string türüne dönüştürülüp Pos_Y değişkenine atayıp ilgili "Edit Text"e yazar
172 handles.Pos_Z.String=num2str(floor(T(1,4))); % Hesaplanan açı değerlerinin string türüne dönüştürülüp Pos_Z değişkenine atayıp ilgili "Edit Text"e yazar
173
174
```

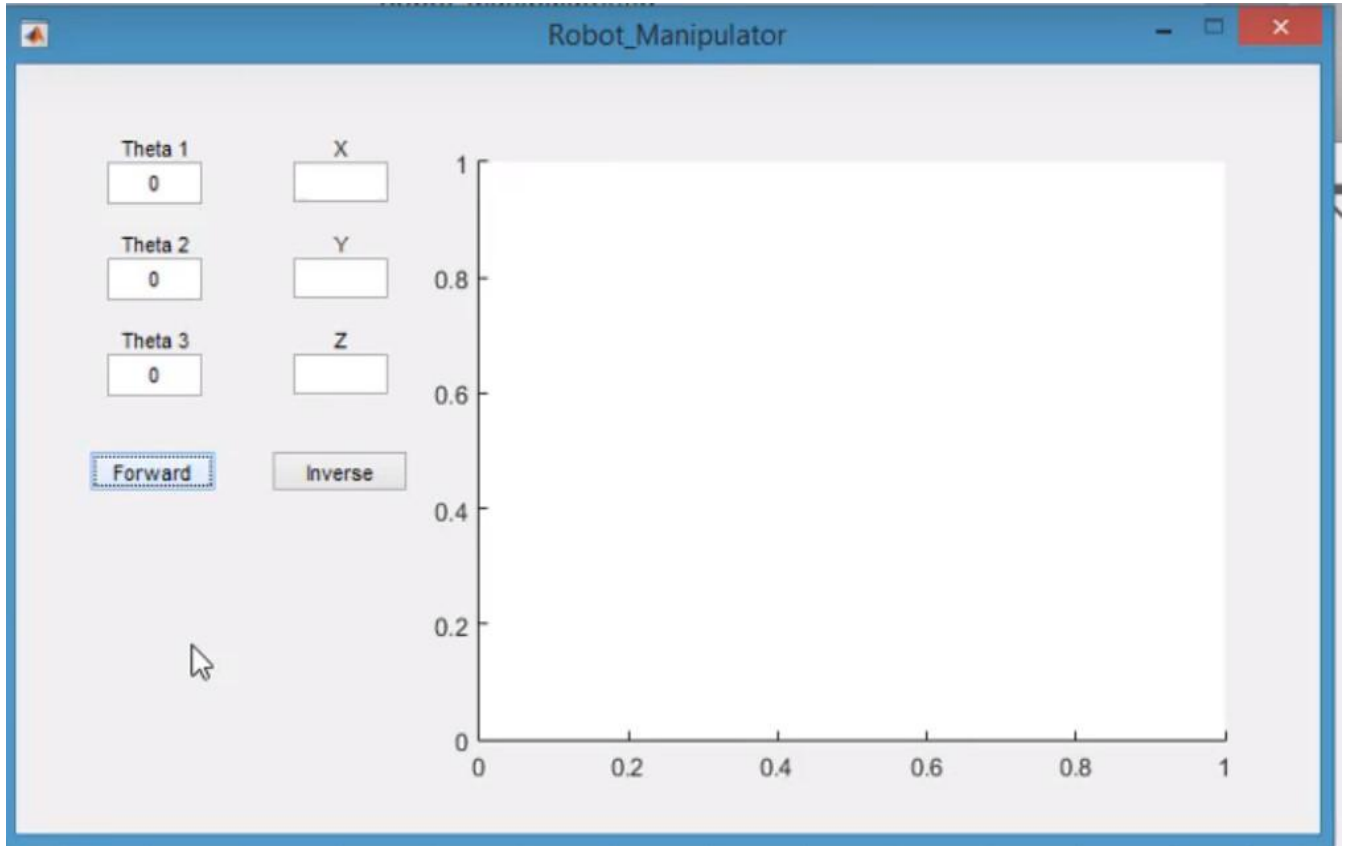
Burada yapılan MATLAB hesaplarının ayrıntılı açıklamasına yukarıdaki görselden ulaşabilirsiniz.

İleri Kinematik hesaplamalarından sonra ters kinematik hesapları için GUI den “Inverse” Butonuna sağ tık callbacks ile ters Kinematik hesaplamaları yapılacaktır.

```
250 %% Ters Kinematik Kısım İçin Hesaplamalar
251
252 PX=str2double(Pos_X.String); % Pos_X string türünü okuyup double türüne dönüştür ve PX değişkenine atar
253 PY=str2double(Pos_Y.String); % Pos_Y string türünü okuyup double türüne dönüştür ve PY değişkenine atar
254 PZ=str2double(Pos_Z.String); % Pos_Z string türünü okuyup double türüne dönüştür ve PZ değişkenine atar
255
256 L_1=20; % 1.Link için Link Uzunluğu belirlendi
257 L_2=50; % 2.Link için Link Uzunluğu belirlendi
258 L_3=40; % 3.Link için Link Uzunluğu belirlendi
259
260 L(1)=Link([0 L_1 0 pi/2]); % Robot kol için hesaplanan 1.link parametreleri
261 L(2)=Link([0 0 L_2 0]); % Robot kol için hesaplanan 2.link parametreleri
262 L(3)=Link([0 0 L_3 0]); % Robot kol için hesaplanan 3.link parametreleri
263
264
265 Robot=SerialLink(L); % Yukarıda hesabı yapılan bu üç linkin birleştirip robot kol haline getiren fonksiyon
266 Robot.name='Yasin_Robot'; % Robot kol'a verdiğimiz isim
267 Robot.plot([Th_1 Th_2 Th_3]); % Hesaplanan açılara göre robot kol'u "Axes" kısmına çizen fonksiyon
268
269
270
271
```

Yukarıdaki Matlab kesitinden Ters Kinematik hesaplamalarına ve her bir fonksiyonun ayrıntılı açıklamasına ulaşabilirsiniz.

Bundan sonra programı çalıştırıp açılı değerleri girilebilir.



Burada sadece aç ı değ erleri girilip “Forward” butonu ile robot kolun konum bilgisi hesaplanıp robot kol çizdirildi.

