Robotik Manupilator Matlab Dosyası Çalıştırma

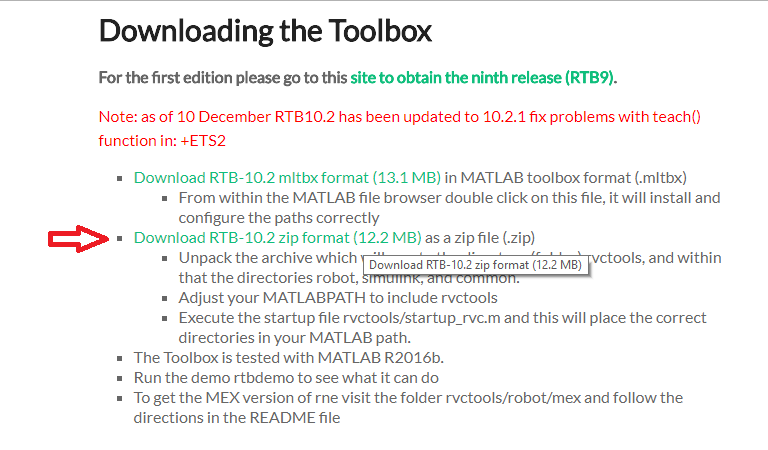
Öncelikle Proje dosyamız yardımcı bir toolbox dosyasıyla çalışmaktadır. Aşağıda verilen linke tıklayarak fotğraflarda gösterilen yolu izleyerek öncelikle toolbox dosyasını matlaba yüklenmesi gerekir. Bu toolbox Matlab 2016b sürümü için geçerlidir.

Bu proje çalıştırma işlemi 2 aşamadan oluşur ;

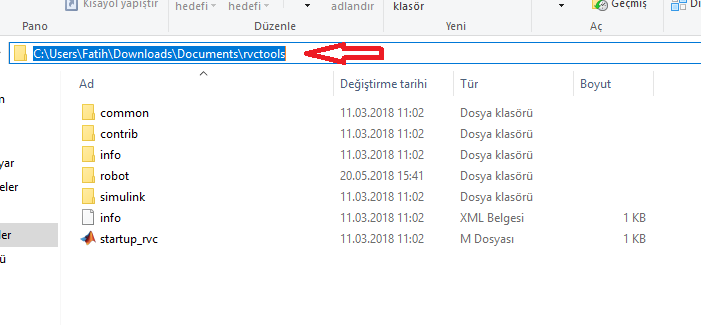
1. Toolbox İndirme ve Yükleme

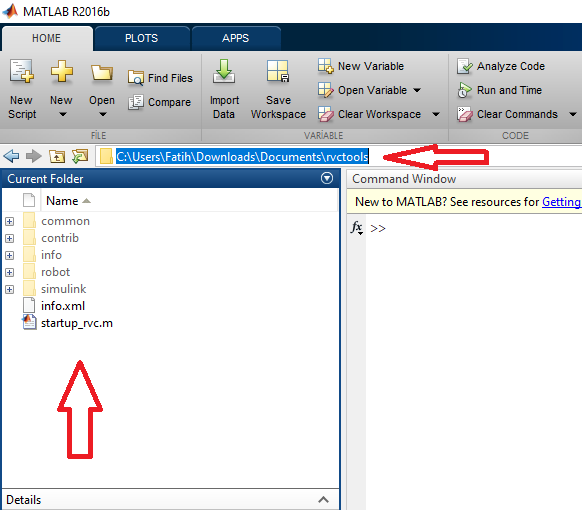
1.Toolbox Linki : <http://petercorke.com/wordpress/toolboxes/robotics-toolbox#Downloading_the_Toolbox>

2. Toolbox dosyasının indirilmesi : Aşağıdaki fotoğrafta gösterilen bölümdeki linke tıklayarak zip formatındaki dosya indirilmelidir.



3. Toolbox dosyasının matlaba yüklenmesi: Zip dosyasından çıkarılan dosyaların dizinini kopyalayıp; matlab içindeki adres çubuğu bölümüne yapıştırarak enter tuşuna basarak dosya dizinini tanımlamış oluruz.

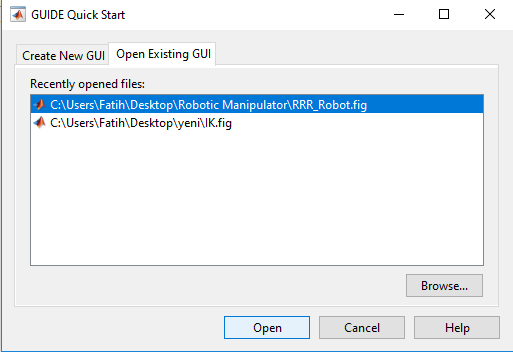




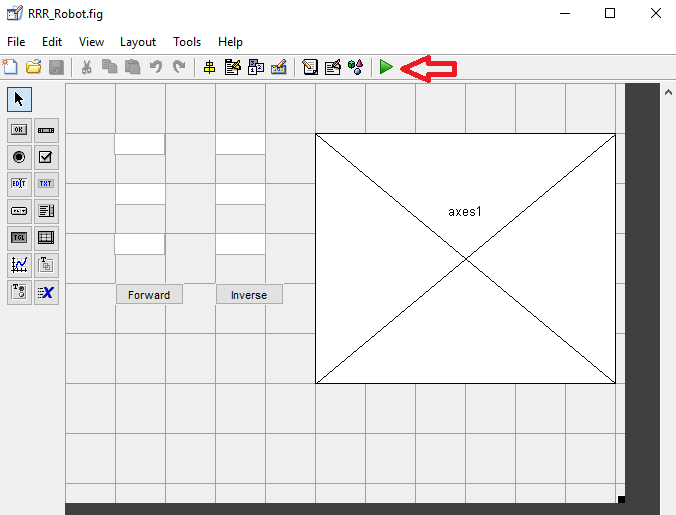
4. Dosyalar sol bölümde gözüktükten sonra "startup\_rvc.m" dosyasına sağ tıklanarak run komutuna basılır ve toolbox dosyaları matlaba yüklenir ve işlevsel hale gelir.

2. Proje Simülasyonunun Çalıştırılması

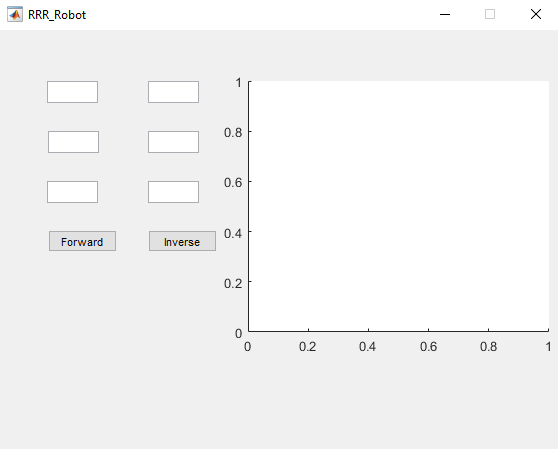
1. Toolbox dosyası yüklendikten sonra sıra Robotik Manupilator proje dosyalarını açmaya gelir.Bunun için matlab komut kısmına(command window) "guide" yazılarak Matlab Guide açılır.



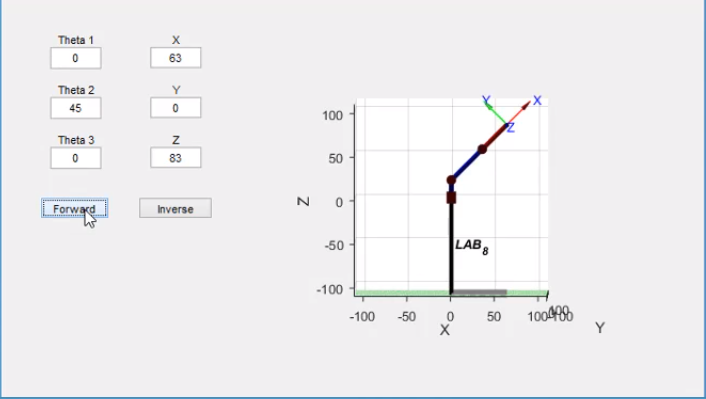
2.Bu kısımdan "browse.." butonuna tıklayarak proje dosyası olan ".fig" uzantılı dosya seçilerek proje taslak dosyası open butonuna basarak açılır.



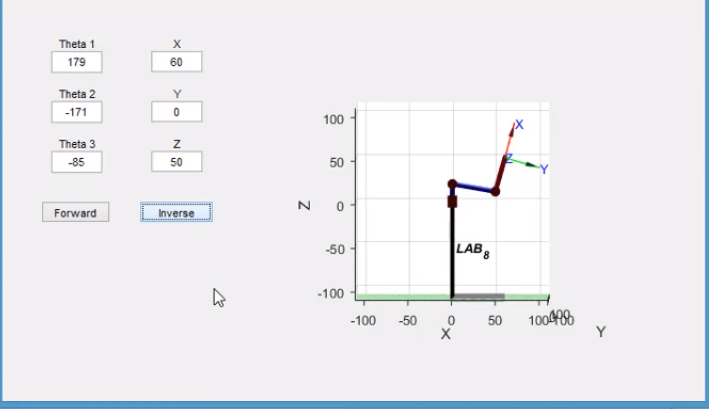
3.Bundan sonra yeşil buton olan "run" butonuna basılıp gelen pencerede "Change folder" butonuna basılarak proje dosyası açılır.

4.Bu açılan pencerede "Forward" butonu üzerindeki boşluklara Teta1,Teta2,Teta3 değerleri girilerek "Forward" butonuna basılır ve ileri kinematik yapılır.

5.Daha sonra "İnverse" butonu üzerindeki boşluklara X,Y,Z değerleri girilerek "İnverse" butonuna basılır ve geri kinematik yapılır.



İleri Kinematik



Geri Kinematik

MATLAB KODU

function varargout = RRR\_Robot(varargin)

% RRR\_ROBOT MATLAB code for RRR\_Robot.fig

% RRR\_ROBOT, by itself, creates a new RRR\_ROBOT or raises the existing

% singleton\*.

%

% H = RRR\_ROBOT returns the handle to a new RRR\_ROBOT or the handle to

% the existing singleton\*.

%

% RRR\_ROBOT('CALLBACK',hObject,eventData,handles,...) calls the local

% function named CALLBACK in RRR\_ROBOT.M with the given input arguments.

%

% RRR\_ROBOT('Property','Value',...) creates a new RRR\_ROBOT or raises the

% existing singleton\*. Starting from the left, property value pairs are

% applied to the GUI before RRR\_Robot\_OpeningFcn gets called. An

% unrecognized property name or invalid value makes property application

% stop. All inputs are passed to RRR\_Robot\_OpeningFcn via varargin.

%

% \*See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one

% instance to run (singleton)".

%

% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help RRR\_Robot

% Last Modified by GUIDE v2.5 28-Apr-2018 21:29:56

% Begin initialization code - DO NOT EDIT

gui\_Singleton = 1;

gui\_State = struct('gui\_Name', mfilename, ...

'gui\_Singleton', gui\_Singleton, ...

'gui\_OpeningFcn', @RRR\_Robot\_OpeningFcn, ...

'gui\_OutputFcn', @RRR\_Robot\_OutputFcn, ...

'gui\_LayoutFcn', [] , ...

'gui\_Callback', []);

if nargin && ischar(varargin{1})

gui\_State.gui\_Callback = str2func(varargin{1});

end

if nargout

[varargout{1:nargout}] = gui\_mainfcn(gui\_State, varargin{:});

else

gui\_mainfcn(gui\_State, varargin{:});

end

% End initialization code - DO NOT EDIT

% --- Executes just before RRR\_Robot is made visible.

function RRR\_Robot\_OpeningFcn(hObject, eventdata, handles, varargin)

% This function has no output args, see OutputFcn.

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% varargin command line arguments to RRR\_Robot (see VARARGIN)

% Choose default command line output for RRR\_Robot

handles.output = hObject;

% Update handles structure

guidata(hObject, handles);

% UIWAIT makes RRR\_Robot wait for user response (see UIRESUME)

% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.

function varargout = RRR\_Robot\_OutputFcn(hObject, eventdata, handles)

% varargout cell array for returning output args (see VARARGOUT);

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure

varargout{1} = handles.output;

function Theta\_1\_Callback(hObject, eventdata, handles)

% hObject handle to Theta\_1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Theta\_1 as text

% str2double(get(hObject,'String')) returns contents of Theta\_1 as a double

% --- Executes during object creation, after setting all properties.

function Theta\_1\_CreateFcn(hObject, eventdata, handles)

% hObject handle to Theta\_1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

function Theta\_2\_Callback(hObject, eventdata, handles)

% hObject handle to Theta\_2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Theta\_2 as text

% str2double(get(hObject,'String')) returns contents of Theta\_2 as a double

% --- Executes during object creation, after setting all properties.

function Theta\_2\_CreateFcn(hObject, eventdata, handles)

% hObject handle to Theta\_2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

function Theta\_3\_Callback(hObject, eventdata, handles)

% hObject handle to Theta\_3 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Theta\_3 as text

% str2double(get(hObject,'String')) returns contents of Theta\_3 as a double

% --- Executes during object creation, after setting all properties.

function Theta\_3\_CreateFcn(hObject, eventdata, handles)

% hObject handle to Theta\_3 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

% --- Executes on button press in btn\_Forward.

function btn\_Forward\_Callback(hObject, eventdata, handles)

%İleri kinematiğin yapıldığı fonksiyon kımı.

% hObject handle to btn\_Forward (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

Th\_1 = str2double(handles.Theta\_1.String)\*pi/180; %Bu üç satırda gi-

Th\_2 = str2double(handles.Theta\_2.String)\*pi/180; %rilen teta değer-

Th\_3 = str2double(handles.Theta\_3.String)\*pi/180; %leri alınır.

L\_1 = 20; %L değerleri

L\_2 = 50; %linklerin

L\_3 = 40; %uzunlukları

L(1) = Link([0 L\_1 0 pi/2]); %Toolbox da yüklenilen Link komutu

L(2) = Link([0 0 L\_2 0]); %ile ileri kinematik için gerekli

L(3) = Link([0 0 L\_3 0]); %matematiksel işlemler yapılır

Robot = SerialLink(L);

Robot.name = 'RRR\_Robot';

Robot.plot([Th\_1 Th\_2 Th\_3]); %plot komutu ile manupilator %çizimi yapılır

T = robot.fkine([Th\_1 Th\_2 Th\_3,varargin]);

handles.Pos\_X.String = num2str(floor(T(1,4)));

handles.Pos\_Y.String = num2str(floor(T(2,4)));

handles.Pos\_Z.String = num2str(floor(T(3,4)));

function Pos\_X\_Callback(hObject, eventdata, handles)

% hObject handle to Pos\_X (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Pos\_X as text

% str2double(get(hObject,'String')) returns contents of Pos\_X as a double

% --- Executes during object creation, after setting all properties.

function Pos\_X\_CreateFcn(hObject, eventdata, handles)

% hObject handle to Pos\_X (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

function Pos\_Y\_Callback(hObject, eventdata, handles)

% hObject handle to Pos\_Y (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Pos\_Y as text

% str2double(get(hObject,'String')) returns contents of Pos\_Y as a double

% --- Executes during object creation, after setting all properties.

function Pos\_Y\_CreateFcn(hObject, eventdata, handles)

% hObject handle to Pos\_Y (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

function Pos\_Z\_Callback(hObject, eventdata, handles)

% hObject handle to Pos\_Z (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Pos\_Z as text

% str2double(get(hObject,'String')) returns contents of Pos\_Z as a double

% --- Executes during object creation, after setting all properties.

function Pos\_Z\_CreateFcn(hObject, eventdata, handles)

% hObject handle to Pos\_Z (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

% --- Executes on button press in btn\_Inverse.

function btn\_Inverse\_Callback(hObject, eventdata, handles)

%Geri kinematiğin yapıldığı fonksiyon kımı.

% hObject handle to btn\_Inverse (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

PX = str2double(handles.Pos\_X.String); %Bu üç satırda

PY = str2double(handles.Pos\_Y.String); %girilen X,Y,Z

PZ = str2double(handles.Pos\_Z.String); %değerleri alınır.

L\_1 = 20; %L değerleri

L\_2 = 50; %linklerin

L\_3 = 40; %uzunlukları

L(1) = Link([0 L\_1 0 pi/2]); %Toolbox da yüklenilen Link komutu

L(2) = Link([0 0 L\_2 0]); %ile geri kinematik için gerekli

L(3) = Link([0 0 L\_3 0]); %matematiksel işlemler yapılır

Robot = SerialLink(L);

Robot.name = 'RRR\_Robot';

T = [ 1 0 0 PX;

0 1 0 PY;

0 0 1 PZ;

0 0 0 1];

J = Robot.ikine(T, [0 0 0], [1 1 1 0 0 0]) \* 180/pi;

handles.Theta\_1.String = num2str(floor(J(1)));

handles.Theta\_2.String = num2str(floor(J(2)));

handles.Theta\_3.String = num2str(floor(J(3)));

Robot.plot(J\*pi/180); %plot komutu ile manupilator %çizimi yapılır