

Matlab GUI Yardımıyla Görsel bir Robotik Araç Kutusu Tasarımı

Sibel Kaplan¹, Hüseyin Canbolat²

¹ Elektrik ve Elektronik Mühendisliği Bölümü
Mersin Üniversitesi, Çiftlikköy
sibel_izmir@yahoo.com

² Elektrik ve Elektronik Mühendisliği Bölümü
Mersin Üniversitesi, Çiftlikköy
huseyinc@mersin.edu.tr

Özetçe

Bu çalışmada, komut tabanlı geliştirilmiş ücretsiz dağıtılan *Robotics Toolbox*.Version 7 April-2002 araç kutusu içerisinde yer alan tüm komutların, MATLAB editörü olmadan görsel ekranlar aracılığı ile işletilmesi amaçlanmıştır. *Robotics Toolbox*, komutlarının gerektirdiği parametrelerin programa rahatça verilebilmesi amacıyla MATLAB'ın grafiksel kullanıcı arayüzü (GUI) kullanılmıştır. GUI ile geliştirilen ara yüz kullanıcıya gerekli parametrelerin her biri için bir alan ayırdığından kullanıcının komuttaki parametre sayısı ve hangi parametreler olduğu gibi bilgileri önceden ayarlamadan daha hızlı komut işleme ve sonuç almasına yardımcı olmaktadır.

Abstract

In this study, it is aimed that the commands of *Robotics Toolbox*.Version 7 April-2002 is executed through visual windows outside the standard MATLAB editor. The command windows are created using MATLAB graphical User Interface (GUI) facilities. Since the visual command window is set to ask every parameter required to run the command, the user does not need to know the details of the command parameters and other settings.

1. Giriş

Robotics Toolbox, dünya çapında birçok robotik dersinde kullanılmaktadır ve *Robotics Toolbox*'ı kullanan birçok çalışma bulunmaktadır [1,2]. Yazılımın yaygın kullanılan MATLAB ile uyumlu olması amaçlanmıştır [1,3]. Bu çalışmadaki amaç *Robotics Toolbox*'ı mevcut kullanımından çıkarıp kullanıcıların, görsel bir arayüz ile kullanmalarını sağlamaktır. Mühendislik araştırmalarında çok sık kullandıkları bir program olan Matlab'da, bu konuda görsel bir araç kutusu henüz geliştirilmemiştir. Fakat, Mikrodalga alanında görsel bir araç kutusu daha önceden tasarlanmıştır [4]. Bu çalışmada, Matlab ortamında görsel olarak geliştirdiğimiz "Görsel Robotik Araç Kutusu" tasarımı tanıtılmaktadır.

Robot manipülatörlerinin simülasyonu ve programlanması için farklı yaklaşımlara dayalı paketler geliştirilmiştir [5,6,7]. Robot sistemleri özellikle otomotiv sanayinde yaygın biçimde kullanılmaktadır. Bu gibi yerlerde robotlar arası koordinasyon ve işbirliğinin sağlanması etkin bir veri alışverişine bağlıdır.

Bunun için bilgisayar iletişimi önemli bir faktördür [6]. Bu sistemlerin tasarımında teorik performansın elde edilmesi için bilgisayar ortamında sistemi tanımlayarak simülasyonlarının yapılması önemli yer tutmaktadır. Fakat her araştırmacı kendi sistemini kendisi bilgisayar ortamında hazırlayarak gerekli testlerden elde edilecek sonuçları görmekte ve fiziksel prototiplerde deneyler yapılmaktadır. Genel olarak kullanılan robot manipülatörleri benzer özelliklere sahiptir. Üstelik simülasyonlar genellikle MATLAB paket programı kullanılarak gerçekleştirilmektedir [8]. Bunların defalarca bilgisayar ortamında yeniden tanımlanması zaman ve emek kaybına neden olmaktadır. Robot simülasyon yazılımı bu gibi ön hazırlıkları önemli ölçüde kısaltacaktır.

Matlab Grafiksel Kullanıcı Arayüzü, diğer bir söylemi ile Matlab GUI, Matlab programcısı tarafından hazırlanan grafik tabanlı uygulamaların, son kullanıcıya fare ve klavye arabirimi ile interaktif olarak hitap etmesini sağlayan bir platformdur [9]. Matlab GUI uygulamalarının gerekliliğinin temel sebeplerinin başında günümüzde hazırlanan uygulamaların grafik tabanlı oluşu ve bu uygulamaların son kullanıcı tarafından kullanım kolaylığına sahip olması gelmektedir [9].

Matlab GUI' nin çalışması belirli üç temel özellik içerir. Bunlar GUI Yüzeyi, GUI Objeleri ve İşlevlerdir. GUI yüzeyi programda kullandığımız bütün objelerin bulunduğu kısımdır. GUI yüzeyine elemanların yerleştirildiği ve görsel temanın sağlandığı kısımda denilebilir. GUI objeleri programı oluştururken kullandığımız buton, slider, axes gibi her birinin kendine ait bir işlevi olan ve bu işlevlere göre programcının oluşturduğu program yapısında çalışan elemanlardır. En önemli kısım olan işlevler kısmı (*Callback*) bir nesnenin ne yapması gerektiği belirtilen kısımdır. Gerçekte eğer m-function şeklinde yazılan grafik tabanlı programlarda nesnelerin *Callback* (işlev)' nin belirtilmesi gerekir.[8,9]

2. Görsel Robotik Araç Kutusu Tasarımı

Geliştirilen görsel robotik araç kutusunun (GRAK) grafiksel kullanıcı ara yüzünün hazırlanmasında, bir ana ekran kullanılmıştır. Bu kısmın ayrıntıları tasarım bölümünde verildikten sonra bazı örnek uygulamalar takip eden bölümde verilecektir.

Robotics Toolbox kinematikler, dinamikler ve yörünge türetme gibi robotikte geçerli birçok işlevi oluşturmada yararlı bir modelleme aracıdır. Robotics Toolbox gerçek robotlarla yapılan deneyleri çözümlmek kadar önemli olan 3D benzetimi yerine getirir. Peter I. Corke(CSIRO Manufacturing Science and Technology-Australia) tarafından 1996 yılında ilk versiyonu çıkarılmıştır. Şu anda kullanılan versiyon Release6'dır ve yazılım tarihi 2001'dir. Yazılım <http://www.mathworks.com/> adresinden ücretsiz olarak edinilebilir. Robotics Toolbox seri uzuvlu manipülatörler için yazılmıştır. Matlab'ın genel özelliği olarak yazılıma müdahale edilebilmektedir. Böylece kullanıcı belirli konular için kendi benimsediği algoritmaları kullanarak gerekli hesaplamaları yapabilir. Robotics Toolbox'ta kullanılan ve GRAK ile görselleştirilen *.m file dosyalarında kullanılan komutların listesi Tablo1 de verilmiştir.

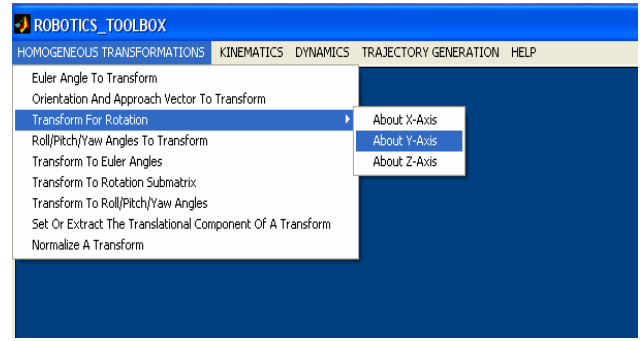
Tablo 1. GRAK'da görselleştirilen Robotik Toolbox Komutları

Homojen Dönüşümler	
eul2tr	Euler açılarından homojen dönüşüme geç
oa2tr	Yönelim ve yaklaşım vektörü homojen dönüşüme geç
rot2tr	Homojen dönüşümden 3x3 dönme altmatrisini al
rotx	X-ekseni etrafında dönme için homojen dönüşüme geç
roty	Y-ekseni etrafında dönme için homojen dönüşüme geç
rotz	Z-ekseni etrafında dönme için homojen dönüşüme geç
rpy2tr	Roll/pitch/yaw açılarından homojen dönüşüme geç
tr2eul	Homojen dönüşümden Euler açılarına geç
tr2rot	Homojen dönüşümden dönme altmatrisine geç
tr2rpy	Homojen dönüşümden roll/pitch/yaw açılarına geç
transl	Homojen dönüşümden öteleme altmatrisini al
trnorm	Homojen dönüşümün normalini al
Yörünge Türetimi	
ctrj	Kartezyen yörünge türet
jtraj	Eklem uzayı yörüngesi türet
trinterp	Homojen dönüşümleri interpolate et
Kinematikler	
diff2tr	Diferansiyel hareket vektöründen homojen dönüşüme geç
fkine	Düz kinematikleri hesapla
ikine	Ters kinematikleri hesapla
ikine560	Puma 560 için ters kinematikleri hesapla
ikinetl2	Twolink için ters kinematikleri hesapla
jacob0	Temel koordinat çerçevesinde Jakobyeni hesapla
jacobn	Sonlandırıcı koordinat çerçevesinde Jakobyeni hesapla
tr2diff	Homojen dönüşümden diferansiyel hareket vektörüne geç
tr2jac	Homojen dönüşümden Jakobyene geç
Dinamikler	
accel	Düz dinamikleri hesapla
cinertia	Kartezyen manipülatör eylemsizlik matrisini hesapla
coriolis	Coriolis/merkezkaç momentlerini hesapla
friction	Sürtünme ekle
frans	Kuvvet/moment dönüştür

gravload	Yerçekimi yükünü hesapla
inertia	Manipülatör eylemsizlik matrisini hesapla
itorque	Eylemsizlik momentini hesapla
nofriction	Robot nesnesinden sürtünmeyi çıkart
rne	Ters dinamikleri hesapla

2.1. Tasarım

GRAK ana ekranında Robotik Toolbox içerisinde kullanılan komutlar mөнüler halinde listelenmiştir. Görsel robotik araç kutusunun geliştirilmesinde Matlab GUI baz alınmıştır. GUI içerisinde var olan tüm görsel öğeler kullanılarak, Robotikte kullanılan denklemleri çözüp sonuçlar üreten birer figür ve Matlab *.m file dosyaları hazırlanmış ve bunlar ana ekranda oluşturulan mөнü aracılığı ile çağrılmıştır. Bunun ardından, robotikte kullanılan bazı olaylar için animasyonları hazırlanmış, hazırlanan animasyonlar programa eklenmiştir. Ayrıca son aşamada araç kutusunun nasıl kullanılacağını içeren bir yardım mөнüsü hazırlanmış ve programa bir mөнü elemanı olarak eklenmiştir. Şekil 1'de program çalıştırıldığında karşılaşılan ilk ekran görülmektedir.



Şekil 1. Programın çalışır ekran görüntüsü

Programın en belirgin özelliği kullanıcıların işlemlerini yapmak için Matlab editöründen herhangi bir işlem yapmaması, komut yada parametre girmemesidir. Her bir işlem için hazırlanan görsel ekran aracılığı ile işlemlerini kolaylıkla gerçekleştirebilir. Bu programda kullanıcıların Matlab Programlama Dilini bilmesi de gerekmemektedir. Çünkü, kullanılacak olan *.m file dosyaları görsel mөнüler ve görsel ekran öğeleri şeklinde, içlerindeki gereken parametre sorgularıyla beraber kullanıcıya sunulmaktadır. Kullanıcının sadece bu parametreleri uygun değerlerle doldurup, gerekli düğmelere basması yeterlidir.

Programda Robotik Toolbox içerisinde işlem gören tüm komutlar için gerekli figür ve *.m file dosyaları oluşturulmuştur. Aşağıdaki listede programda kullanılan mөнü elemanları listelenmiştir. Ayrıca bu mөнülerin işlevlerine göre alt elemanları da programda mevcuttur.

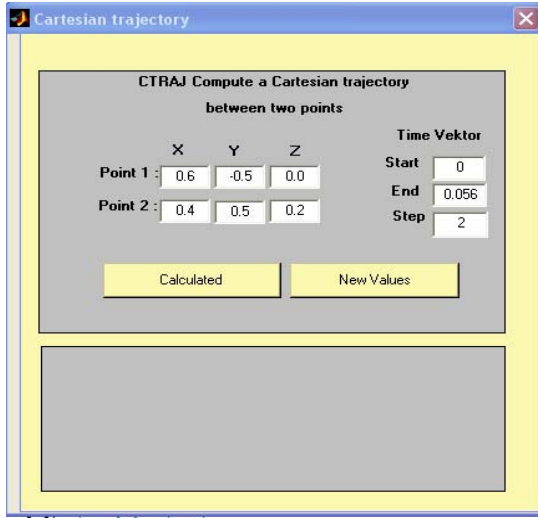
- Homojen Dönüşümler (*Homogeneous transformations*)
- Yörünge Türetimi (*Trajectory generation*)
- Kinematikler (*Kinematics*)
- Dinamikler (*Dynamics*)

2.2. Örnek Uygulamalar

Hazırlanan görsel robotik araç kutusu için bazı örnekler aşağıda verilmiştir.

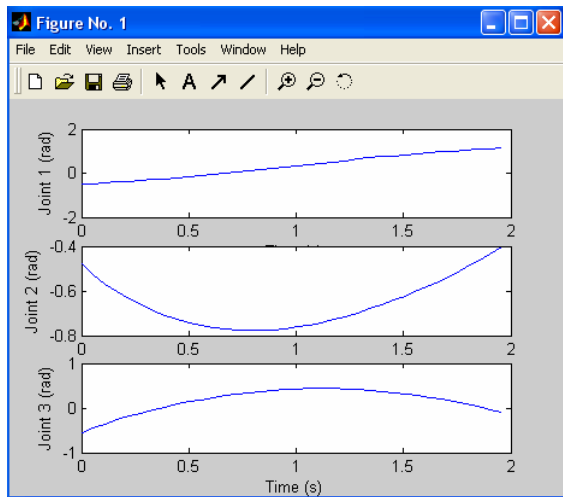
2.2.1. Örnek 1 (Kartezyen Yörünge Türetimi)

Manipülörün uç noktasının başlangıç noktasından son durumuna kadar hareketi esnasında, yer değiştirme ve dönme yollarını belirleyen noktalar kümesi “Yörünge” olarak adlandırılır. Kartezyen yörünge türetiminde esas olan robot ucunu, o anki başlangıç değerinden (Point1), istenilen son değere (Point2) taşımaktır. Programda görsel olarak ilk ve son nokta değerleri ve t'ye göre zaman vektörü girilecektir. Şekil2’de Kartezyen yörünge türetimi için GUI’de hazırlanan ekran görüntüsü görülmektedir.



Şekil2. Kartezyen Yörünge Türetimi Ekran Görüntüsü

Yukarıdaki ekran görüntüsüne gerekli değerler girilip “Calculated” düğmesine tıklandığında Şekil3’de görüldüğü gibi uç noktanın izlediği yolun kartezyen koordinatlardaki çizimi ekranda görülür.



Şekil3. Uç Noktanın Kartezyen Koordinatlarda İzlediği Yol

Kartezyen Yörünge Türetimi için yazılmış olan *.m dosyası aşağıdaki gibidir:

```
function Calculated_Callback(hObject, eventdata, handles)
p1_x=findobj(gcf,'Tag','Point1_X');
p1_y=findobj(gcf,'Tag','Point1_Y');
p1_z=findobj(gcf,'Tag','Point1_Z');

p2_x=findobj(gcf,'Tag','Point2_X');
p2_y=findobj(gcf,'Tag','Point2_Y');
p2_z=findobj(gcf,'Tag','Point2_Z');

t1=findobj(gcf,'Tag','T1');
t2=findobj(gcf,'Tag','T2');
t3=findobj(gcf,'Tag','T3');

p1_x=str2num(get(p1_x,'String'));
p1_y=str2num(get(p1_y,'String'));
p1_z=str2num(get(p1_z,'String'));

p2_x=str2num(get(p2_x,'String'));
p2_y=str2num(get(p2_y,'String'));
p2_z=str2num(get(p2_z,'String'));

alınan_t1=str2num(get(t1,'String'));
alınan_t2=str2num(get(t2,'String'));
alınan_t3=str2num(get(t3,'String'));

P1 = transl(p1_x, p1_y, p1_z) % define the start point
P2 = transl(p2_x, p2_y, p2_z) % and destination
Tm=[alınan_t1:alınan_t2:alınan_t3] % create a time vector

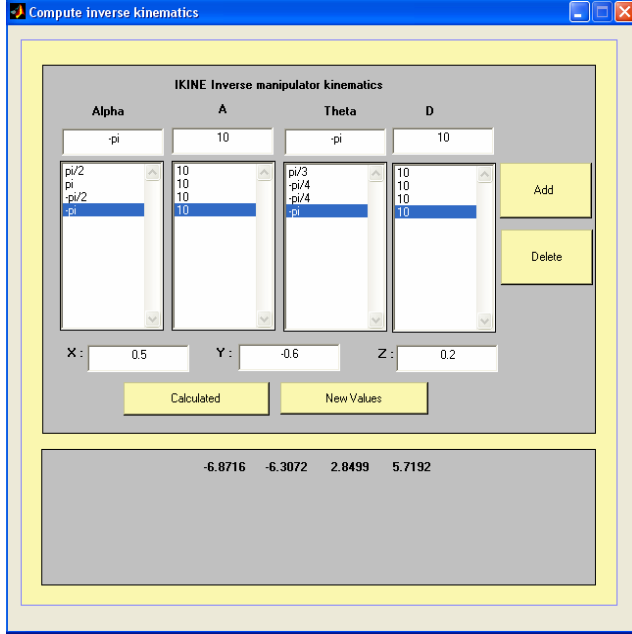
str4=findobj(gcf,'Tag','sonuc');
set(str4,'String','');
TT=ctraj(P1, P2, length(Tm));
s=ctraj(P1, P2, length(Tm));
s=num2str(s);
set(str4,'String',s);
```

```
puma560;
q = ikine(p560, TT);
```

```
figure(1);
subplot(3,1,1);
plot(Tm,q(:,1));
xlabel('Time (s)');
ylabel('Joint 1 (rad)');
subplot(3,1,2);
plot(Tm,q(:,2));
xlabel('Time (s)');
ylabel('Joint 2 (rad)');
subplot(3,1,3);
plot(Tm,q(:,3));
xlabel('Time (s)');
ylabel('Joint 3 (rad)');
```

2.2.2. Örnek 2 (Ters Kinematikleri Hesaplama)

Düz kinematiklerin tersi biçimde robotikte ters kinematikler verilen sonlandırıcı konum ve yönelimi için gerekli eklem değişken değerlerini bulmak olarak tanımlanabilir. Ters kinematik problemlerinde düz kinematiklerin tersi biçimde homojen dönüşüm matrisleriyle oluşturulan doğrusal olmayan denklemlerin çözülmesi istenir. Şekil4’de Ters Kinematikleri hesaplamak için GUI’de hazırlanan ekran görüntüsü görülmektedir.



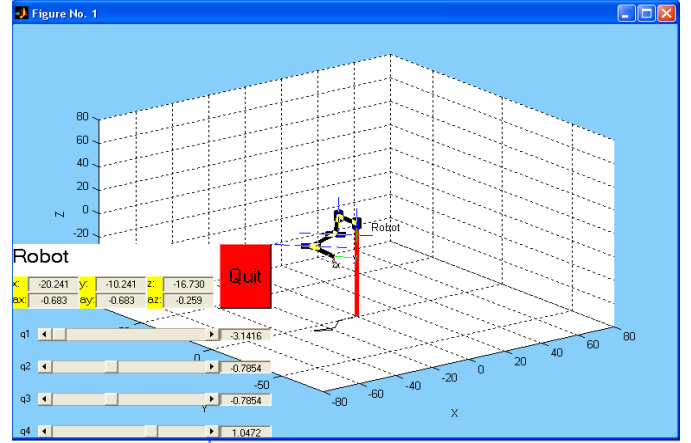
Şekil4. Ters Kinematik Hesaplama Ekran Görüntüsü

Yukarıda ekranda görüldüğü ters kinematik hesaplamak için gerekli olan robotu kullanıcı istediği kadar eklem noktası olacak şekilde kendisi oluşturmaktadır. Ayrıca hesaplamak için gerekli olan uç birim pozisyonu X,Y,Z koordinatları girdikten sonra “Calculated” düğmesine tıkladığında hesaplama sonucu aşağıdaki Şekil4’de görüldüğü gibi metin kutusu içerisinde gözükmektedir. Tablo2’de her bir eklem için kullanılan parametreler ve neye karşılık geldikleri belirtilmiştir.

Tablo 2: Bir Eklem İçin Alınan Parametreler

Robot Parametresi	Sembol	Tanımı
alpha	α	Eklem burulma(twist) açısı
A	A	Eklem uzunluğu
theta	θ	Eklem dönme açısı
D	D	İki eklem arası uzaklık
sigma	σ	Mafsal Tipi (Dönel İçin 0, prizmatik için 0’dan farklı)

Yukarıdaki örnekte 4 eklemli bir robot oluşturulmuştur. Bu robotun simülasyonu da Şekil5 görülmektedir. Bu programda “Calculated” düğmesine tıklanınca oluşturulur.



Şekil 5. Dört Eklemli Örnek Robot Simülasyonu

Ters kinematik hesaplamak için yazılmış olan m dosyası aşağıdaki gibidir:

```
function pushbutton1_Callback(hObject, eventdata, handle)
global L;
global aci;
global r;
l1=findobj(gcf,'Tag','listbox1');
l2=findobj(gcf,'Tag','listbox2');
l3=findobj(gcf,'Tag','listbox3');
l4=findobj(gcf,'Tag','listbox4');

str1 = get(l1, 'String');
str2 = get(l2, 'String');
str3 = get(l3, 'String');
str4 = get(l4, 'String');

str5=findobj(gcf,'Tag','sonuc')
len = length(str1);
sayac=len
if len > 0
    for index = 1:len
        sonuc1=str1(sayac);
        sonuc2=str2(sayac);
        sonuc3=str3(sayac);
        sonuc4=str4(sayac);

        sonuc1=str2num(sonuc1);
        sonuc2=str2num(sonuc2);
        sonuc3=str2num(sonuc3);
        sonuc4=str2num(sonuc4);

        L{index}=link([sonuc1 sonuc2 sonuc3 sonuc4 0]);
        aci(1,index)=sonuc3;
        sayac=sayac-1 ;
    end
end
```

```

strx=findobj(gcf,'Tag','edit5');
stry=findobj(gcf,'Tag','edit6');
strz=findobj(gcf,'Tag','edit7');

xl=str2num(get(strx,'String'));
yl=str2num(get(stry,'String'));
zl=str2num(get(strz,'String'));

Tl = transl(xl, yl, zl)

r=robot(L);
r.name='Robot';
t=ikine(r,Tl,aci);
t=num2str(t);
set(str5,'String','');
set(str5,'String',t);
drivebot(r,aci);

```

3. Sonular

Bu alıřmada Matlab GUI ile hazırlanmıř “Görsel Robotik Ara Kutusu Tasarlanmıřtır”. Bylelikle robot sistemleri iin genel bir simülasyon ortamı hazırlanmıřtır. Yazılım, belli bařlı robotlar iin gerekli matematiksel karakteristiklerini tanımakta ve yeni bir robot manipölatorü tanımlanmasına da olanak saėlamaktadır. Görsel ekranlar aracılıėı ile girilen parametrelere göre robotun hareketlerinin görsel bir animasyonu ve gerekli diėer analizleri yapılmaktadır. Kullanıcıların Matlab Programlama Dili, herhangi bir formül ya da komut bilmeksizin işlemlerini yapmaları saėlanmıřtır. Robotun grafiksel animasyonu ile ilgili alıřmalar halen devam etmektedir.

alıřmanın ilerleyen ařamalarında, yazılımın diėer mekatronik sistemleri kapsayacak řekilde genişletilmesi

amalanmaktadır. Ayrıca *Robotics Toolbox* veya bir simülasyon paketiyle elde edilen verilerin oluřturacaėı hareketin üzerinde alıřılan robotun grafiksel bir görüntüsü yoluyla bilgisayar ekranında animasyonunun oluřturulması hedeflenmektedir.

4. Kaynaka

- [1] *Matlab Users Guide, The MathWorks Inc., USA, <http://www.mathworks.com>*
- [2] *Robotics Toolbox For Matlab(Release 6) Peter Corke, April 2001*
- [3] *User Reference For Matlab Ver.6.5, Mathworks Inc., 2003.*
- [4] Vural M.S.,*Matlab Grafiksel Arabirimi Yardımıyla “Görsel Mikrodalga Ara Kutusu” Tasarımı* , Yüksek Lisans Tezi, Mersin Üni. Fen Bil. Ens. ,2005
- [5] Gourdeau R., *"Object Oriented Programming for Robotic Manipulator Simulation"*, IEEE Robotics and Automation Magazine, 1997;4(3):21-29.
- [6] Halsall F., *"Data Communications, Computer Networks and Open Systems"*, Third Edition, Addison-Wesley, 1992.
- [7] Pires J.N., Sá da Costa J.M.G., *"Object Oriented and Distributed Approach for Programming Robotic Manufacturing Cells"*, IFAC Journal on Robotics and Computer Integrated Manufacturing, 1999
- [8] İ.Yüksel, *Matlab İle Mühendislik Sistemlerinin Analizi ve Çözümü*, Türkmen Yayınevi, 2000.
- [9] M.Uzunoėlu , A.Kızıl, Ö. Ç. Onar, *Kolay Anlatımı İle İleri DüzeydeMatlab 6.0-6.5*, Türkmen Yayınevi, 2002.
- [10] K.S.Fu, R.C.Gonzalez,C.S.G.Lee, *Robotics Control, Sensing,Vision, and Intelligence*, McGraw-Hill Book Company, 1987.