



**BİLECİK ŞEYH EDEBALI
ÜNİVERSİTESİ**

**Fen Bilimleri Enstitüsü
Enerji Sistemleri Mühendisliği Anabilim Dalı**

**LEAP MOTİON İLE 2B ROBOT KOLU
SİMÜLASYONU**

**Seren EMİR
Yüksek Lisans Tezi**

**Tez Danışmanı
Yrd. Doç. Dr. Salim CEYHAN**

BİLECİK 2017

Ref.No: 10152674



**BİLECİK ŞEYH EDEBALI
ÜNİVERSİTESİ**

**Fen Bilimleri Enstitüsü
Enerji Sistemleri Mühendisliği Anabilim Dalı**

**LEAP MOTİON İLE 2B ROBOT KOLU
SİMÜLASYONU**

**Seren EMİR
Yüksek Lisans Tezi**

**Tez Danışmanı
Yrd. Doç. Dr. Salim CEYHAN**

BİLECİK 2017



**BİLECİK ŞEYH EDEBALI
UNIVERSITY**

Graduate School of Sciences

Department of Energy Systems Engineering

**2D ROBOT ARM SIMULATION WITH USING LEAP
MOTION**

**Seren EMİR
Master's Thesis**

**Thesis Advistor
Ass. Prof. Salim CEYHAN**

BİLECİK 2017



BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS JÜRİ ONAY FORMU

Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulunun 31/05/2017 tarih ve 28 sayılı kararıyla oluşturulan jüri tarafından 15/06/2017 tarihinde tez savunma sınavı yapılan Seren EMİR' in "*Leap Motion ile 2B Robot Kolu Simülasyonu*" başlıklı tez çalışması Enerji Sistemleri Mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak oy birliği/ oy çokluğu ile kabul edilmiştir.

JÜRİ

ÜYE

(TEZ DANIŞMANI) : Yrd. Doç. Dr. Salim CEYHAN

ÜYE

: Doç. Dr. Kürşat AYAN

ÜYE

: Yrd. Doç. Dr. Nazım İMAL

ONAY

Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulunun
.../.../..... tarih ve/..... sayılı kararı.

İMZA/ MÜHÜR

TEŞEKKÜR

Bilimsel çalışmalarına her zaman destek olan ve hiçbir zaman yardımlarını esirgemeyen Fethi CANDAN' a çok teşekkür ederim. Fikirleriyle ve bilgileriyle beni yönlendiren İTÜ IPC Takımı' na teşekkür ederim. Sayın hocam Yrd. Doç. Dr. Salim CEYHAN' a yanımda olduğu için teşekkür ederim.

Seren EMİR

Mayıs, 2017



ÖZET

Bu çalışmada Leap Motion ile robot kolunun gerçek zamanlı hareket etmesi gereken noktalar kullanıcı tarafından dışarıdan verilir. Alınan bu noktalar yardımıyla robot kolunun hareketini sağlamak için önce ters kinematikle serbest değişkenler ve daha sonra ileri kinematikle de robot kolunun kartezyen koordinatları bulunur. Üç serbestlik derecesine sahip RRP tipli 3B bir robot kolunun hareketi z-ekseni boyunca sabit düzlem üzerinde kalması sebebiyle üstten görünümü baz alınarak, iki serbestlik derecesine sahip 2B simülasyonu LabVIEW ortamında oluşturuldu. İki serbestlik derecesine sahip robot kolun hareketi sadece kinematik yöntemler kullanarak kontrol edildi. Denavit-Hartenberg yöntemi ile iki serbestlik derecesine sahip robot kolun parametreleri belirlendi ve bu parametrelerden yararlanılarak ileri ve ters kinematik modeller LabVIEW ortamında oluşturuldu. Leap Motion ile elde edilen konum bilgisinden yararlanarak, ters kinematik modelden eklem açılarının alabileceği değerler LabVIEW ortamında hesaplandı. Ters kinematik probleminden bulunan çözümler ileri kinematik problemde yerine konularak en uygun çözüm robot kolunun yaklaşık konumu olarak alındı. Leap Motion'dan alınan n adet konum verisi ve LabVIEW ortamında kinematik hesaplarla elde edilen robot kolunun kartezyen koordinatları arasındaki hatalar iki boyutta Mutlak Hata Oranı Ortalaması(MAPE) ile hesaplandı.

Anahtar Kelimeler: 2B Robot Simülasyonu, Leap Motion, LabVIEW, Mutlak Hata Oranı Ortalaması(MAPE).

ABSTRACT

In this study, the user gives Leap Motion and desired points where the robot arm must move in real time from the outside. With the aid of these points, first, in order to provide motion of the robot arm, there are free variables with inverse kinematics and then Cartesian coordinates of the robot arm with forward kinematics. A 2D robot simulation with two degrees of freedom was created in the LabVIEW Program, based on the top view, because the RRP type robot arm with three degrees of freedom stays on a constant plane along the z-axis of motion. The movement of the robot arm that has two degrees of freedom only was controlled using kinematic methods. By using the Denavit-Hartenberg method, robot arm parameters with two degrees of freedom were determined and by using these parameters, forward and inverse kinematic models were created in LabVIEW program. By using the position information obtained by Leap Motion, the values that the joint angles can take in the inverse kinematic model are calculated in the LabVIEW program. The solutions derived from the inverse kinematics problem were substituted in the advanced kinematic model and the most appropriate solution was taken as the approximate position of the robot arm. The errors between the Cartesian coordinates of the robot arm obtained from Leap Motion with n includes position datas and the kinematic calculations in the LabVIEW Program were calculated using the Mean Absolute Percentage Error(MAPE) in two dimensions.

Key : 2D Robot Simulation, Leap Motion, LabVIEW, Mean Absolute Percentage Error(MAPE)

İÇİNDEKİLER

Sayfa No

JÜRİ ONAY SAYFASI

TEŞEKKÜR

ÖZET i

ABSTRACT ii

ÇİZELGE DİZİNİ..... iv

ŞEKİLLER DİZİNİ v

SİMGELER VE KISALTMALAR..... vii

1. GİRİŞ..... 1

1.1. Literatür Araştırması 1

2. KİNEMATİK ANALİZ 3

2.1. İleri Kinematik..... 3

2.2. Ters Kinematik 6

3. LEAP MOTİON 10

4. LABVIEW PROGRAMI..... 14

4.1. LabVIEW ile Grafiksel Programlamanın Faydaları..... 14

4.2. LabVIEW Kullanıcı Arayüzü..... 21

4.3. 2B Robot Kolunun Tasarlanması 25

4.4. Leap Motion Entegrasyonu 26

5. MUTLAK HATA ORANI ORTALAMASI..... 30

6. SONUÇ VE ÖNERİLER..... 36

KAYNAKLAR..... 37

ÖZGEÇMİŞ

ÇİZELGE DİZİNİ

	Sayfa No
Çizelge 2.1 : Denavit-Hartenberg parametreleri.....	4
Çizelge 2.2 : Projeye ait Denavit-Hartenberg parametreleri.....	5



ŞEKİLLER DİZİNİ

	Sayfa No
Şekil 2.1 : İki serbestlik derecesine sahip robot kolun eksen takımları.....	3
Şekil 2.2 : RR eklem yapısına sahip düzlemsel robot.....	6
Şekil 2.3 : Puma-560 robotuna ait çözümler.....	9
Şekil 2.4 : İleri kinematik ve ters kinematik problemler.....	9
Şekil 3.1 : Leap Motion Cihazının şematik gösterimi.....	10
Şekil 3.2 : Leap Motion Cihazının koordinat sistemi.....	10
Şekil 3.3 : Leap Motion Cihazının görüş alanı ve çalışma alanı.....	11
Şekil 3.4 : Leap Motion Cihazının etkileşim kutusu.....	12
Şekil 3.5 : 2B robot kolunun simülasyonunun çalışma sistemi.....	13
Şekil 4.1 : LabVIEW de yapılmış grafiksel programlama örneği.....	14
Şekil 4.2 : LabVIEW' de düğüm noktaları, veri girişlerinin ve çıkışlarının gösterildiği block diagram.....	15
Şekil 4.3 : LabVIEW ara yüzüne ait araç çubuğu.....	16
Şekil 4.4 : a) Kırılmış çalıştır düğmesi. b) Error list paneli.....	17
Şekil 4.5 : Döngüler arası veri iletimi.....	18
Şekil 4.6 : LabVIEW mathscript node terminali ve G programlama.....	19
Şekil 4.7 : Leap Motion Cihazının entegrasyon blok şeması.....	20
Şekil 4.8 : Gösterge oluşturma.....	21
Şekil 4.9 : a) Front panel. b) Block diagram.....	22
Şekil 4.10 : a) Controls. b) Functions.....	23
Şekil 4.11 : a) Göstergeler ve kontroller. b) Terminaller.....	24
Şekil 4.12 : 2B robot kolun simülasyonunun tasarımı.....	25
Şekil 4.13 : 2B robot kolunun kullanıcı ara yüzü.....	26
Şekil 4.14 : Leap Motion konum algılama.....	26
Şekil 4.15 : Lokal değişken kullanımı.....	27
Şekil 4.16 : Leap Motion entegrasyonu.....	28
Şekil 4.17 : 2B robot kolun simülasyonu.....	29
Şekil 5.1 : LabVIEW ile mutlak hata oranının ortalaması.....	30
Şekil 5.2 : a) Gerçek veriler-I. b) Yaklaşık veriler-I.....	31
Şekil 5.3 : Verilerin doğruluğu-I.....	32

Şekil 5.4 : a) Gerçek veriler-II. b) Yaklaşık veriler-II.....	33
Şekil 5.5 : Verilerin doğruluğu-II.....	34
Şekil 5.6 : Hesabı yapılmamış veri tablosu.....	35



SİMGELER VE KISALTMALAR

Simgeler

${}^{i-1}_iT$	: Dönüşüm Matrisi
${}^{i-1}_iR$	: Dönme Matrisi
${}^{i-1}_ip$	: Pozisyon Vektörü
a_{i-1}	: Uzunluk
α_{i-1}	: Uzunluk Açısı
d_i	: Kayar Sistemlerde Eklem Kaçıklığı
θ_i	: Döner Sistemlerde Eklem Açısı
0iT	: İleri Kinematik Probleminin Çözümü
a_1	: Birinci Uzunluk
a_2	: İkinci Uzunluk
θ_1	: Birinci Eklem Dönme Açısı
θ_2	: İkinci Eklem Dönme Açısı
x_i	: Leap Motion' dan Alınan x_i Verileri
$\hat{x}_i$	: Kinematik İşlemlerden Elde Edilen $\hat{x}_i$ Verileri
y_i	: Leap Motion' dan Alınan y_i Verileri
$\hat{y}_i$	: Kinematik İşlemlerden Elde Edilen $\hat{y}_i$ Verileri
n	: Alınan Veri Sayısı

Kısaltmalar

SCARA	: Selective Compliant Articulated Robot Arm(Seçilebilir Uyumlu Eklemli Robot Kolu)
TCP	: Transmission Control Protocol(İletişim Kontrol Protokolü)
IP	: Internet Protocol(İnternet Protokolü)
IR	: Infrared Radiation(Kızılötesi)
LED	: Light Emitting Diode(Işık Yayan Diyot)
2B	: İki Boyutlu
3B	: Üç Boyutlu
NI	: National Instruments
LabVIEW	: Laboratory Virtual Instrument Engineering Workbench

G	: Grafiksel Programlama
USB	: Universal Serial Bus(Evrensel Seri Veri Yolu)
PCI	: Peripheral Component Interconnect(Çevresel Bileşen Arabirimi)
PXI	: PCI Extensions for Instrumentation Express(Enstrümantasyon Express için Çevresel Bileşen Arabirim Uzantıları)
RF	: Radio Frequency(Radyo Frekansı)
FPGA	: Field Programmable Gate Array(Alanda Programlanabilir Kapı Dizileri)
MAPE	: Mean Absolute Percentage Error(Mutlak Hata Oranı Ortalaması)
VI	: Virtual Instrument(Görsel Enstrümantasyon)
RRP	: SCARA Robot Eksen Tipi

1. GİRİŞ

Günümüzde robotlar hem ticari hem de ev kullanıcıları için en popüler mühendislik alanı haline gelmiştir. Farklı uygulamalar için yaygın olarak kullanılan geniş robot çeşitleri vardır. Bunlardan biri olan SCARA robot, imalat sanayisinde genellikle lehimleme, kesme, delme, diş açma, baskı devre, montaj ve frezeleme için kullanılır.

Bu çalışmada 2B robot kolu üç serbestlik derecesine sahip 3B SCARA robot kolunun hareketi z-ekseni boyunca sabit düzlem üzerinde kalması sebebiyle üstten görünümü baz alınarak LabVIEW programında tasarlanmıştır. Tasarlanan 2B robot kolun Leap Motion ile gerçek zamanlı olarak hareketi sağlanmıştır. Bu çalışma ile çalışılması zor ve tehlikeli kimyasal maddelerin, Leap Motion ile uzaktan kullanımı sağlanarak daha güvenli bir çalışma ortamının oluşturulması amaçlanmıştır.

1.1. Literatür Araştırması

SCARA robot tasarımı yapılarak statik ve doğal frekans analizleri hesaplanmış olup çeşitli programlarla hareketinin simülasyonu sağlanmıştır (Kucuk, 2005).

Farklı serbestlik derecesine sahip SCARA robotların ters kinematik çözümleri yapılmış ve hareketin en hızlı biçimde hesaplanması sağlanmıştır (Demirci, 2012).

Endüstriyel robotların serbest şekil yollarının uygulamada sınırlı olmasından kaynaklı sorunlar oluşturduğu ele alınarak, robot denetleyici üzerinde uygulanan algoritmalarla ve interpolasyon hesaplarıyla robotların serbest şekil yollarının nasıl kullanılacağı, SCARA robotlar üzerinde test edilmiştir (Moctezuma, 2012).

İşitme engelli kişilerle toplumda işaret dili bilmeyen kişiler arasında iletişimi kolaylaştırmak için Leap Motion cihazı ile Arap diline yönelik çalışmalar yapılmıştır (Mohandes, 2014).

Sanal bir simülatördeki iki ameliyat çubuğunun hareketi için Leap Motion ile bir mimik tanıma sistemi tasarlanmış ve cerrahi bir eğitimin temelini sağlamak için laparoskopi yapmak üzere tasarlanmıştır (Vargas, 2014).

SCARA robotun Matlab&Simülink tasarımı ve kontrolü gerçek zamanlı olarak sağlanmıştır (Çelikbağ, 2015).

FUM SCARA adı verilen bir robot için özel bir lineer-döner eklemin mekanik tasarım süreci gösterilmektedir. Tasarlanan bu özel eklem sayesinde üretim hatlarında seçim, yerleştirme ve montaj uygulamalarını yapması deneysel ve simülasyon olarak

sağlanmıştır (Mousavi, 2015).

SCARA robotların mekanik ve elektriksel olarak tasarlanması ve gerçekleşmesi dışında ters kinematik denklemler, manipülatör hareketini kontrol etmek için cebirsel ve geometrik bir yöntem kullanarak türetilmesi sağlanmıştır (Gulzar, 2015).

Plastik ve alüminyum malzemeler kullanarak mekanik maliyeti daha düşük SCARA robot tasarımı gerçekleştirilmiştir. Yapılan bu SCARA robot tasarımıyla merkezinden belirli bir uzaklıktaki nesnelerin görüntüleri alınmış ve bu görüntülere kalibrasyon yapılmıştır. Bu işlemin ardından yerleştirilen nesnelerin görüntüsü, örüntü tanıma yöntemlerinden biri ile renk bilgisi kullanılarak sınıflandırılması sağlanmıştır. Son olarak bu nesnelerin merkezi, uzunluk ve eğim bilgileri elde edilerek Modbus-TCP iletişim arabirimi üzerinden kontrol cihazına iletimi sağlanmıştır (Chia-Yu, 2015).

SCARA robotların ters kinematik problemlerinin verimli çözülebilmesi için optimal bir toplama noktası yani yeni bir veri tabanı oluşturarak çözüme götüren sistemler yapılmıştır (Zeqiang, 2016).

Tasarlanan SCARA robota çeşitli yöntemler denenmiş, bunların verimliliği üzerinde durulmuştur. Geleneksel Newton-Euler ve yinelemeli prosedürle karşılaştırıldığında, otomatik ayırma metodundaki sonuçlara dayanarak Newton- Euler in yinelemeli formunda kolaylıkla yazıldığı fakat tersi durumun çok zor olduğu üzerinde durulmuştur (Ana, 2016).

SCARA robotlar tıbbi uygulamalarda da kullanılmıştır ve farklı algoritmalar üzerinde durulmuştur. Önerilen algoritmaların performansı değerlendirilmiş, bunların sonuçları üzerinde durulmuştur (Rascón, 2016).

Robot çeşitlerinin karşılaştırıldığı ve SCARA robotların avantajlarından ve dezavantajlarından bahsedilmiş olup performanslarının kıyaslanması yapılmıştır (Gosselin, 2016).

Leap Motion ile ev otomasyon sistemlerinin yaşlılara ve engellilere yardım kolaylık sağlaması amacıyla mimik kontrol ara yüzü geliştirilmiştir (Branstet, 2015).

Leap Motion cihazı ile engelli kişilerin işaret dili bilmeyen kişiler arasında iletişim kurmalarını kolaylaştıracak çalışmalar yapılmıştır (Demircioğlu, 2016).

Leap Motion cihazı ile servo motor kontrolü ilkesine dayanan, sistemin robot kolunun hareketi belirli komutlar verilerek sağlanmış ve kontrol edebileceği gösterilmiştir (Pititeeraphab, 2016).

2. KİNEMATİK ANALİZ

2.1. İleri Kinematik

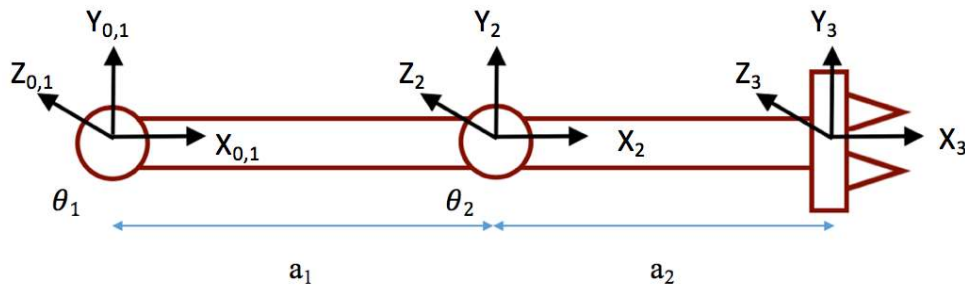
İleri kinematik problemi, robot kolun eklemleri ile uç-işlevcisinin pozisyon ve yönelimi arasındaki ilişki ile ilgilidir. Her ekleme bir eksen takımı yerleştirilirse, ard arda gelen iki eklem arasındaki ilişki ${}^{i-1}_iT$ dönüşüm matrisiyle elde edilebilir. Elde edilecek tüm dönüşüm matrislerinin çarpımı ileri kinematik probleminin çözümünü vermekte olup, denklemin genel gösterimi (2.1)' deki gibidir.

$${}^0iT = {}^0_1T {}^1_2T \dots {}^{i-1}_iT \quad (2.1)$$

2B robot kolun eksen takımları yerleştirilerek, kinematik problemlerin çözümü Denavit-Hartenberg yöntemine göre belirlenmiştir. Eksen takımları yerleştirilmesinde izlenen yol aşağıdaki gibi sırasıyla gerçekleştirilir.

- Eklem eksenlerinin dönme yönü belirlenir. Döner eksenler için dönme yönü Z eksenini olarak belirlenir.
- Z eksenine dik ve kol boyunca olan bağ uzunluğu X eksenini olarak kabul edilir.
- Z ve X eksenleri belirlendikten sonra sağ el kuralına göre Y eksenini bulunur.

Şekil 2.1.' de iki serbestlik derecesine sahip robot kolun eksen takımları gösterilmiştir.



Şekil 2.1. İki serbestlik derecesine sahip robot kolun eksen takımları.

Dönüşüm matrisi ${}^{i-1}_iT$, i numaralı eklemin $i-1$ numaralı ekleme göre dönüşümünü olarak ifade edilebilir. Dönüşüm matrisi, dönme matrisi ${}^{i-1}_iR$ ile pozisyon vektörü ${}^{i-1}_ip$ elde edilen 3×4 boyutundaki matrise birim elemanlar eklenerek (2.5) deki gibi 4×4 boyutunda kare matris elde edilir.

$${}^{i-1}_iR = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (2.2)$$

$${}^{i-1}_ip = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \quad (2.3)$$

$${}^{i-1}_iT = \begin{bmatrix} {}^{i-1}_iR & {}^{i-1}_ip \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

$${}^{i-1}_iT = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

Denavit-Hartenberg yöntemine göre dört ana parametre kullanılarak robot kinematiği çıkarılabilir. Bu parametreler uzuv uzunluğu a_{i-1} , uzuv açısı α_{i-1} , kayar sistemlerde eklem kaçıklığı d_i ve döner sistemlerde eklem açısı θ_i ' dir. Her bir eklem için Çizelge 2.1' deki gibi Denavit-Hartenberg parametreleri belirlenir ve her bir satırdaki Denavit-Hartenberg parametreleri ile dönüşüm matrisleri oluşturulur.

Çizelge 2.1 : Denavit-Hartenberg parametreleri.

Eksen Numarası	Denavit-Hartenberg Değişkenleri				Eklem Değişkeni
i	α_{i-1}	a_{i-1}	d_i	θ_i	d_i veya θ_i
1	α_0	a_0	d_1	θ_1	d_i veya θ_i
2	α_1	a_1	d_2	θ_2	d_i veya θ_i
3	α_2	a_2	d_3	θ_3	d_i veya θ_i

(2.6)' de gösterilen genel denklemde Denavit-Hartenberg parametreleri yerine yazılarak bulunur.

$${}^{i-1}_iT = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & a_{i-1} \\ \sin \theta_i \cos \alpha_{i-1} & \cos \theta_i \cos \alpha_{i-1} & -\sin \alpha_{i-1} & -\sin \alpha_{i-1} d_i \\ \sin \theta_i \sin \alpha_{i-1} & \cos \theta_i \sin \alpha_{i-1} & \cos \alpha_{i-1} & \cos \alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

Bu çalışmada iki serbestlik derecesine sahip robot kolun dönme yönleri aynı olduğundan uzuv açısı ve sistemde sadece döner yapı olduğundan eklem kaçıklığı bulunmamaktadır. 2B robot koluna ait Denavit-Hartenberg parametreleri Çizelge 2.2' de ve her bir eklem dönüşüm matrisleri (2.7) (2.8) (2.9) denklemlerinde verilmiştir.

Çizelge 2.2 : Projeye ait Denavit-Hartenberg parametreleri.

Eksen Numarası	Denavit-Hartenberg Değişkenleri				Eklem Değişkeni
i	α_{i-1}	a_{i-1}	d_i	θ_i	d_i veya θ_i
1	0	0	0	θ_1	θ_1
2	0	a_1	0	θ_2	θ_2
3	0	a_2	0	0	0

$${}^0_1T = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

$${}^1_2T = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & a_1 \\ \sin \theta_2 & \cos \theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

$${}^2_3T = \begin{bmatrix} 1 & 0 & 0 & a_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.9)$$

${}^0_iT = {}^0_1T {}^1_2T \dots {}^{i-1}_iT$ çözümünden yararlanarak (2.10)' daki gibi 0_3T ileri kinematiği bulunur.

$${}^0_3T = {}^0_1T {}^1_2T {}^2_3T \quad (2.10)$$

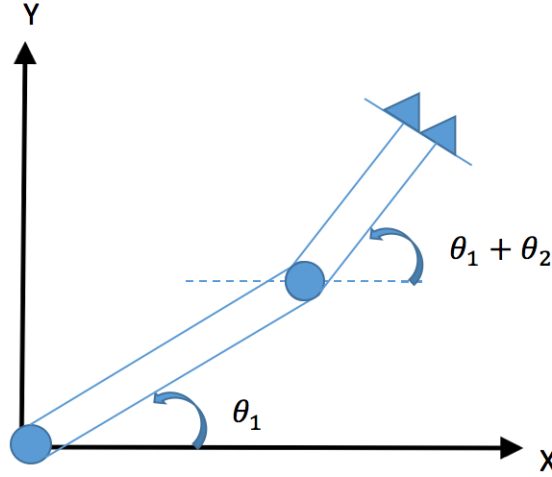
$${}^0_3T = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & a_1 \\ \sin\theta_2 & \cos\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$

$${}^0_3T = \begin{bmatrix} \cos(\theta_1+\theta_2) & \sin(\theta_1+\theta_2) & 0 & \cos(\theta_1+\theta_2)a_2 + \cos\theta_1 a_1 \\ \sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) & 0 & \sin(\theta_1+\theta_2)a_2 + \sin\theta_1 a_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.12)$$

2.2. Ters Kinematik

Ters kinematik problem, robot kolun uç-işlevcisinin konum ve yönelim verilerinden yararlanarak eklem değişkenlerinin bulunmasıdır.

İki serbestlik derecesine sahip düzlemsel robot kolun eklem açıları Şekil 2.2.' de gösterilmiştir. Şekil 2.2.' den yola çıkılarak ters kinematik çözümüyle eklem açıları belirlenmiştir.



Şekil 2.2. RR eklem yapısına sahip düzlemsel robot.

(2.10)' daki denklemin her iki tarafını ${}^0_1T^{-1}$ ile çarpılır.

$${}^0_1T^{-1} {}^0_3T = {}^0_1T^{-1} {}^0_1T {}^1_2T {}^2_3T \quad (2.13)$$

Bu çarpımda ${}^0_1T^{-1} {}^0_1T = I$ olduğundan denklem (2.14)' deki gibi olacaktır.

$$[{}^0_1T]^{-1} {}^0_3T = {}^1_2T {}^2_3T \quad (2.14)$$

$$\begin{bmatrix} \cos\theta_1 & \sin\theta_1 & 0 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.15)$$

$$\begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & a_1 \\ \sin\theta_2 & \cos\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.16)$$

(2.15) ve (2.16) matrisleri birbirine eşitlenir. Eşitlenen matrisler sonucunda aşağıdaki denklem takımları elde edilir.

$$\cos\theta_1 r_{11} + \sin\theta_1 r_{21} = \cos\theta_2$$

$$-\sin\theta_1 r_{11} + \cos\theta_1 r_{21} = \sin\theta_2$$

$$r_{31} = 0$$

$$\cos\theta_1 r_{12} + \sin\theta_1 r_{22} = -\sin\theta_2$$

$$-\sin\theta_1 r_{12} + \cos\theta_1 r_{22} = \cos\theta_2$$

$$r_{32} = 0$$

$$\cos\theta_1 r_{13} + \sin\theta_1 r_{23} = 0$$

$$-\sin\theta_1 r_{13} + \cos\theta_1 r_{23} = 0$$

$$r_{33} = 0$$

$$\cos\theta_1 p_x + \sin\theta_1 p_y = a_2 \cos\theta_2 + a_1$$

$$-\sin\theta_1 p_x + \cos\theta_1 p_y = a_2 \sin\theta_2$$

$$p_z = 0$$

Ters kinematik çözümü için robor kolun uzunluklarının ve konumların olduğu denklemler göz önüne alınır.

$$\cos\theta_1 p_x + \sin\theta_1 p_y = a_2 \cos\theta_2 + a_1 \quad (2.17)$$

$$-\sin\theta_1 p_x + \cos\theta_1 p_y = a_2 \sin\theta_2 \quad (2.18)$$

Denklemlerin kareleri alınıp taraf tarafa toplanmasıyla aşağıdaki (2.19) eşitlik oluşturulur.

$$\begin{aligned} \cos^2 \theta_1 p_x^2 + \sin^2 \theta_1 p_y^2 + \sin^2 \theta_1 p_x^2 + \cos^2 \theta_1 p_y^2 \\ = a_2^2 \cos^2 \theta_2 + 2a_1 a_2 \cos \theta_2 + a_1^2 + a_2^2 \sin^2 \theta_2 \end{aligned} \quad (2.19)$$

Eşitliğin sol tarafı p_x^2 ve p_y^2 parantezine, sağ tarafı a_2^2 parantezine alınırsa $\cos^2 \theta_{1,2} + \sin^2 \theta_{1,2} = 1$ olacağından denklem (2.20)' deki gibi olur.

$$p_x^2 + p_y^2 = a_2^2 + 2a_1 a_2 \cos \theta_2 + a_1^2 \quad (2.20)$$

$$\cos \theta_2 = \frac{p_x^2 + p_y^2 - (a_2^2 + a_1^2)}{2a_1 a_2} \quad (2.21)$$

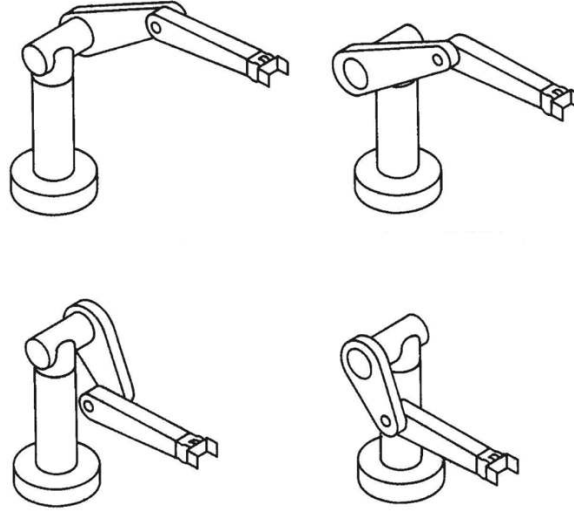
(2.21) denkleminde θ_2 değeri bulunur.

$$\theta_2 = \arg \tan 2 \left(\pm \sqrt{1 - \left[\frac{p_x^2 + p_y^2 - (a_2^2 + a_1^2)}{2a_1 a_2} \right]^2}, \frac{p_x^2 + p_y^2 - (a_2^2 + a_1^2)}{2a_1 a_2} \right) \quad (2.22)$$

θ_2 değeri bilindiğine göre (2.17) denkleminde yararlanılarak θ_1 değeri (2.23)' deki gibi bulunur.

$$\theta_1 = \arg \tan 2(p_x, p_y) \pm \arg \tan 2(\sqrt{p_x^2 + p_y^2 - (a_2 \cos \theta_2 + a_1)^2}, a_2 \cos \theta_2 + a_1) \quad (2.23)$$

Robot kolun uç işlevcisi dört farklı biçimde aynı noktaya ulaşabilir. Şekil 2.3.' de Puma-560 robotu için dört farklı şekilde geldiği görülmektedir.

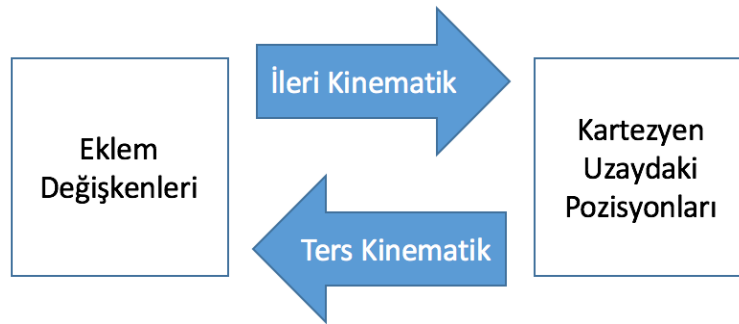


Şekil 2.3. Puma-560 robotuna ait çözümler.

İki serbestlik dercesine sahip robot kolun θ_1 ve θ_2 değerlerinden yararlanılarak dört farklı çözüm kümesi elde edilmiştir. Bulunan bu çözüm kümeleri ileri kinematik probleminin çözümünde yerine yazılarak doğruluğunun tespit edilebilmesi sağlanır. Çözüm kümelerinden yalnızca iki tanesi robot kolunu doğru konuma ulaştırır.

Bu çalışmada ters kinematik çözümündeki konum değerlerimiz Leap Motion tarafından alınmıştır. Elde edilen eklem değişkenlerinden oluşan dört farklı çözüm ileri kinematik denkleminde yerine yazarak yaklaşık çözüm kümesi bulunmuştur.

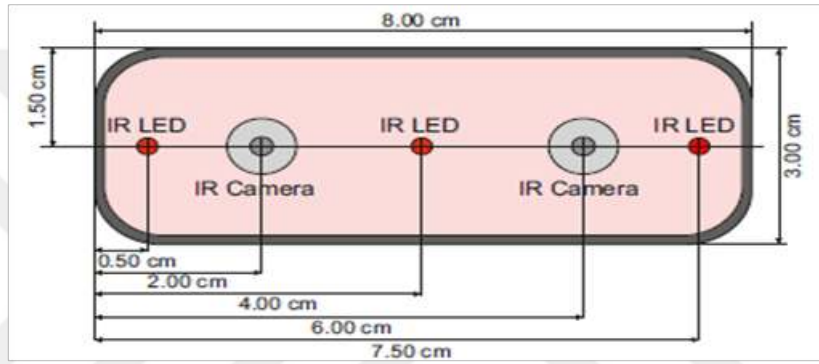
Şekil 2.4.' deki döngüde kartezyen uzaydaki pozisyonları verilen sistemin eklem açılarının ters kinematik problem ile bulunduğu, eklem değişkenleri verilen sistemin ise kartezyen uzaydaki pozisyonları ileri kinematikle problemle bulunduğu yani birbirine ters çalışan sistemler olduğu gösterilmiştir.



Şekil 2.4. İleri kinematik ve ters kinematik problemler.

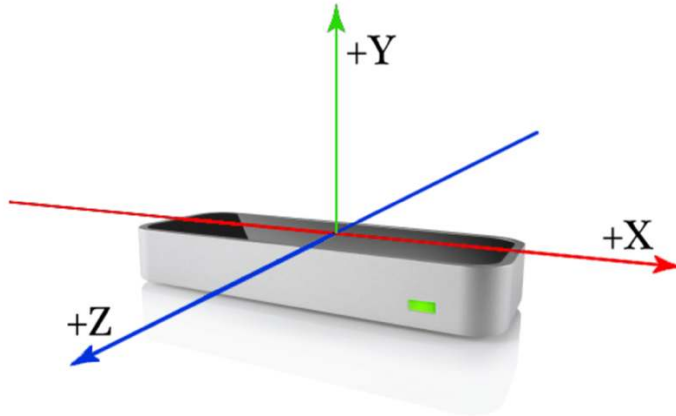
3. LEAP MOTİON

Leap Motion denetleyicisinde, üç IR(Infrared Radiation) LED(Light Emitting Diode) ve iki IR monokromatik kamera bulunmaktadır. Bu sensörler ile elleri ve parmakları tanır ve izler. Yüksek hassasiyet ve izleme hızıyla, ellerin ve parmakların konumları ve hareketleri hakkında bilgi alınabilmektedir. Standart çalışma konumundayken y eksenı boyunca yaklaşık 150 derecelik bir görüş alanına sahiptir. Cihazın etkin menzili yaklaşık 25 ila 600 milimetre arasındadır. Şekil 3.1.' de Leap Motion Cihazının şematik gösterimi verilmiştir.



Şekil 3.1. Leap Motion Cihazının şematik gösterimi.

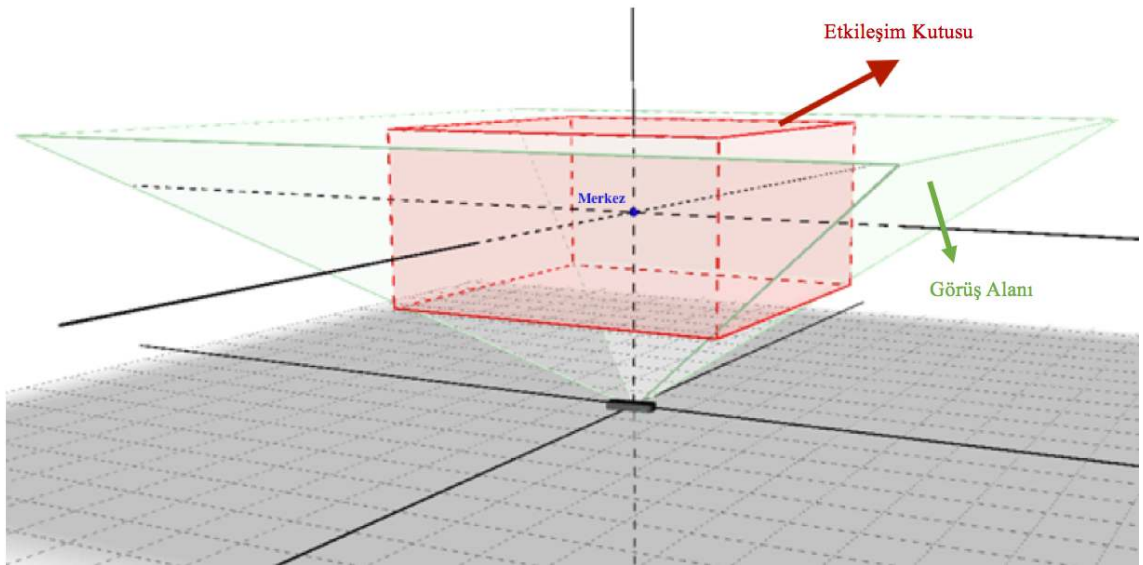
Leap Motion sistemi sağ elle bir Kartezyen koordinat sistemi kullanmakta olup merkezi üst kısmında ortalanmaktadır. x ve z eksenleri yatay düzlemde bulunur, x eksenı cihazın uzun kenarına paralel uzanır. y eksenı dikeydir ve pozitif değerler yukarı doğru artar. z eksenı kullanıcıya doğru artan pozitif değerlere sahiptir. Şekil 3.2.' de Leap Motion Cihazının koordinat sistemi gösterilmiştir.



Şekil 3.2. Leap Motion Cihazının koordinat sistemi.

Leap Motiondan gelen verileri uygulamamızda kullanmak için gelen verileri mantıklı olacak şekilde yorumlamamız gerekir. Leap Motion ile koordinatlarını uygulama koordinatlarına eşlemek, hangi eksenlerin kullanılıp kullanılmayacağına karar vermek, çalışma alanının hangi noktadan yararlanacağına ve mutlak bir haritalama veya görelî haritalamanın kullanılıp kullanılmayacağına karar vermek mümkündür.

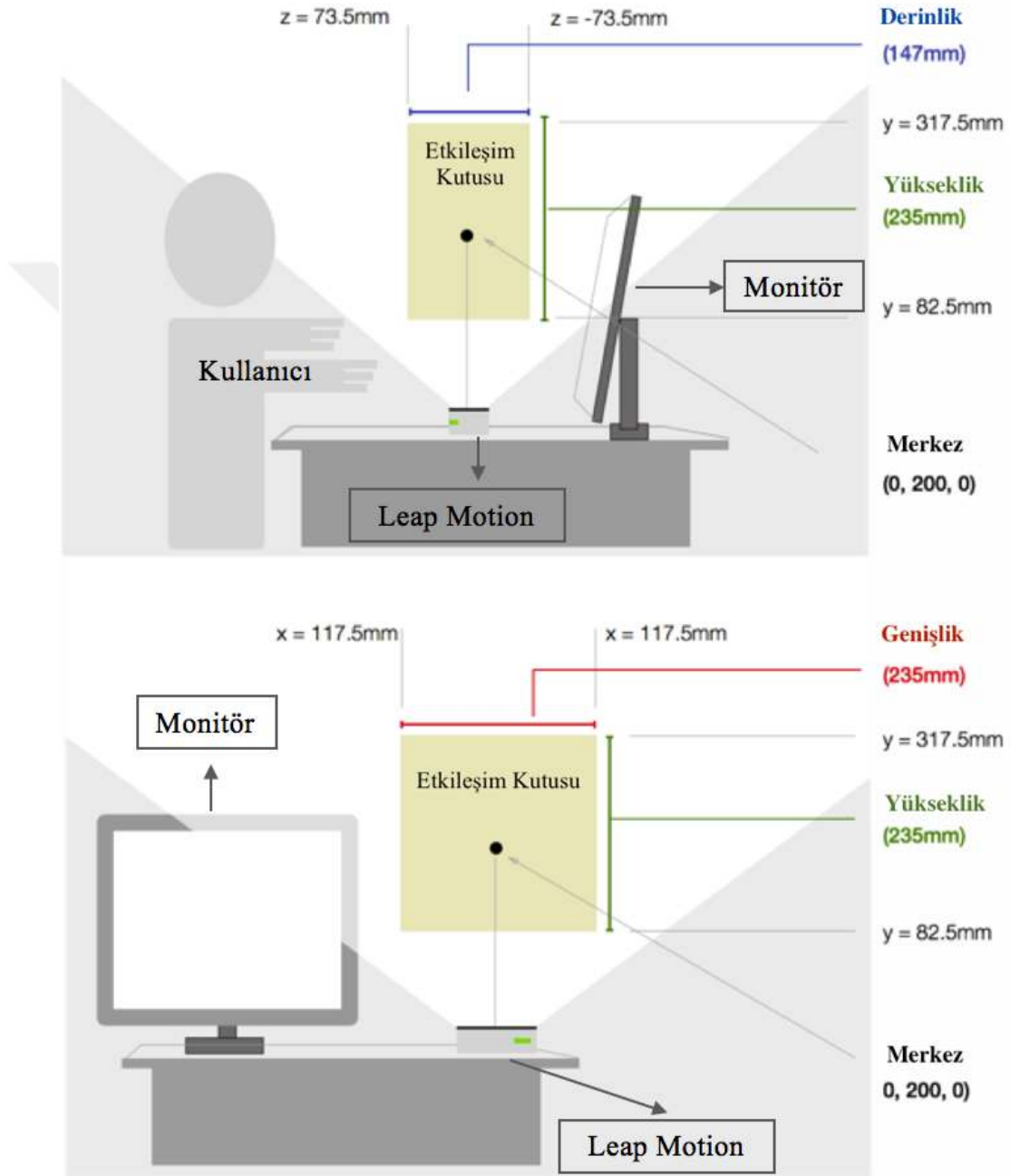
3B uygulamalarında ve 2B uygulamalarında Leap Motion kullanılabilir. 2B uygulamaları için genellikle z eksenî atılır ve girdi kalan iki eksenî uygulama koordinatlarına eşleştirilir. 3B uygulamalarında veya 2B uygulamalarında Leap Motion' ın görüş alanının ne kadarının kullanılacağına karar verilmelidir. Leap Motion' ın görüş alanı ters piramit olması sebebiyle x ve z eksenlerinde mevcut aralık cihazinkinden çok daha küçüktür. Çok geniş bir aralık kullanılırsa veya çok düşük bir yükseklikte kullanılırsa, uygulamanın alt köşelerine erişim sağlanamaz. Çok küçük bir aralık değeri kullanılırsa uygulamada fiziksel hareketin bir kenardan diğerine geçişi kolaylaşacaktır. Şekil 3.3.' de Leap Motion Cihazının görüş alanı ve çalışma alanı gösterilmiştir.



Şekil 3.3. Leap Motion Cihazının görüş alanı ve çalışma alanı.

Kullanıcı elinin veya parmağının belirtilen etkileşim kutusu içinde kaldığı süreç, Leap Motion' ın çalışma alanı bu bölgede daha garanti sonuçlar elde edilmiş olur. Etkileşim yüksekliğini daha yüksek olarak ayarlanırsa, etkileşim kutusu daha büyük olur ve kullanıcı otomatik olarak yüksekliğini ayarlayabilir. Kullanıcı elini etkileşim

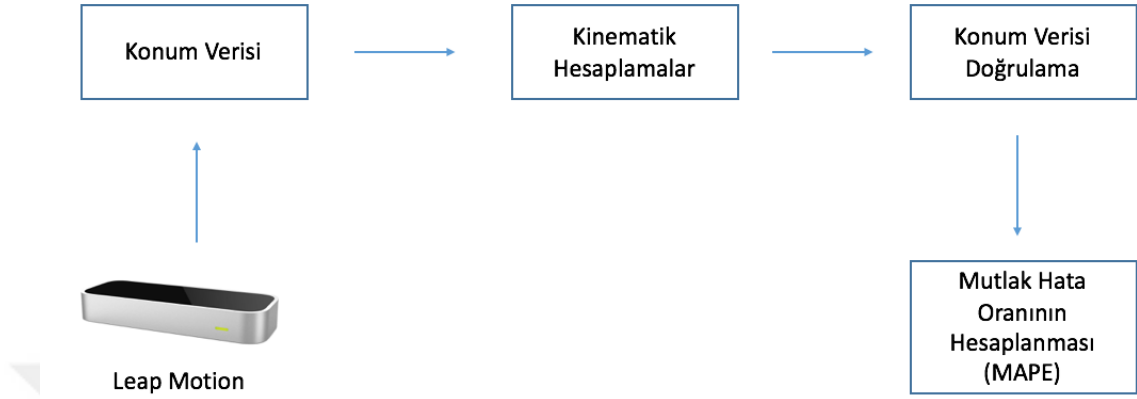
kutusunun geçerli bölgesinin altına getirirse, denetleyici yazılımı yüksekliği düşürür ve otomatik olarak yüksekliği ayarlanabilir. Etkileşim yüksekliğinin otomatik olarak değişme işlemi, minimum yüksekliğe ve maksimum yüksekliğe ulaşana kadar yapılabilir. Şekil 3.4.' de etkileşim kutusu ve çalışma alanıyla ilgili örnek gösterilmiştir.



Şekil 3.4. Leap Motion Cihazının etkileşim kutusu.

Etkileşim kutusu ile daha hassas işler yapmak mümkündür fakat hassasiyet arttıkça kontrol edilmesi zorlaşabilir.

2B robot kolu simülasyonunun çalışma sistemi Şekil 3.5.' de verilmiştir.



Şekil 3.5. 2B robot kolunun simülasyonunun çalışma sistemi.

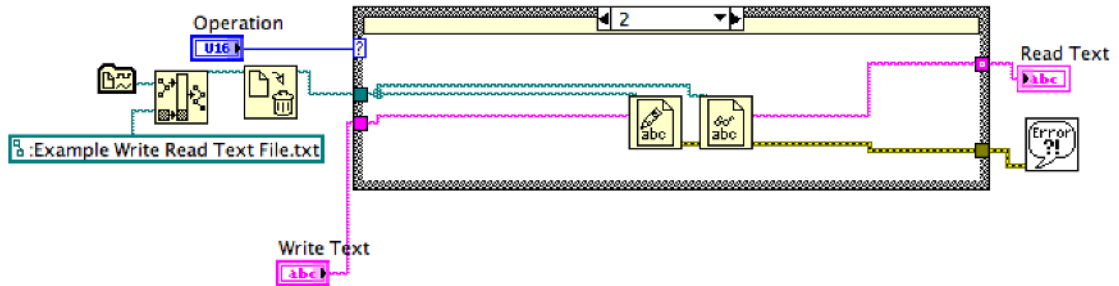
Leap Motion cihazı ile elde edilen konum bilgisinden yararlanarak, ters kinematik modelden eklem açıları hesaplanır. Ters kinematik probleminden elde edilen değerler ileri kinematikte yerine konularak uygun çözüm bulunur ve yaklaşık konum olarak alınır. Leap Motion' dan alınan veriler ile yaklaşık olarak bulunan veriler arasındaki mutlak hata oranı ortalaması(MAPE) hesaplanır.

4. LABVIEW PROGRAMI

NI(National Instruments) LabVIEW(Laboratory Virtual Instrument Engineering Workbench), 20 yılı aşkın bir süredir mühendislerin ve bilim insanlarının test, ölçüm ve kontrol sistemlerini oluşturmak için kullandığı bir geliştirme ortamıdır. Doğal bir grafik programlama dili olan LabVIEW, veri analizi ve sinyal işleme için dahili IP ve açık bir mimari ile herhangi bir donanım cihazının ve herhangi bir yazılım yaklaşımının entegrasyonunu sağlayan uygun çözümün oluşturulmasını sağlayan yazılımdır. Genel amaçlı araçların aksine LabVIEW ile herhangi bir donanımı kapsamlı analiz ve sinyal işleme kütüphaneleriyle sıkı bir şekilde bütünleştirir. LabVIEW özel grafik kullanıcı arayüzü ile sistemleri karmaşıklığını basitleştiren bir platform sunar.

4.1. LabVIEW ile Grafiksel Programlamanın Faydaları

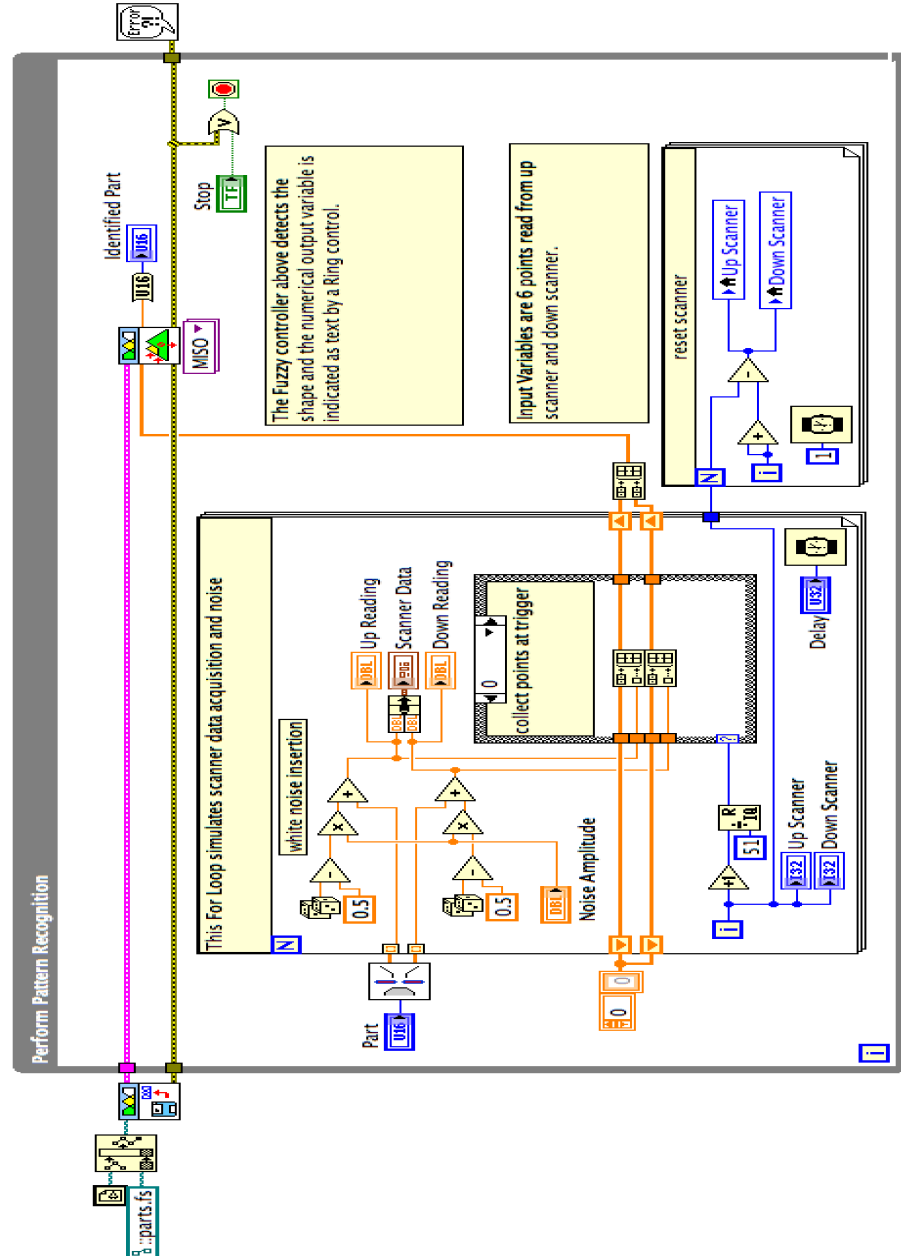
LabVIEW' i genel amaçlı diğer programlama dillerinden ayıran iki ana farklılık vardır. Birincisi, grafiksel simgelerle birlikte bir diyagram üzerinde gerçekleştirilir ve daha sonra doğrudan makine koduna derlenir. Grafiksel olarak gösterilen ve çoğu geleneksel dilde bulunan programlama kavramlarını içerir. Şekil 4.1.' de LabVIEW de yapılmış bir grafiksel programlama örneği gösterilmiştir.



Şekil 4.1. LabVIEW de yapılmış grafiksel programlama örneği

İkinci ana farklılık ise LabVIEW ile geliştirilen ve “G” olarak ifade edilen grafiksel programlamanın, veri akışı kurallarına göre yerine getirilmesidir. C ve C++ gibi çoğu metin tabanlı programlama dilinde bulunan daha geleneksel prosedürel yaklaşım yani başka bir deyişle sıralı bir dizi komut vardır. LabVIEW ile sıralı metin dizgileri değil programdaki düğümler arasındaki veri akışını yürütme sırası belirlenir. LabVIEW programındaki düğümlerde diğer bir deyişle işlevler, döngüler, altprogramlar ve benzeri

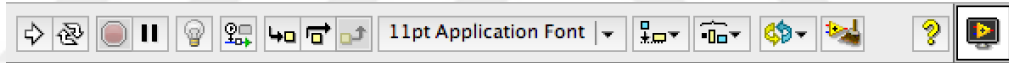
yapıların girdileri vardır. Belirli bir düğüm, girişlerinin tümü geçerli veri içeriyorsa, bu düğüm mantığını yerine getirir ve çıktı verisi üretir, üretilen bu veriler veri akışı yolundaki bir sonraki düğüme aktarılır. Başka bir düğümden veri alan bir düğüm, yalnızca diğer düğüm yürütmeyi tamamladıktan sonra yürütmesini tamamlayabilir. Şekil 4.2. LabVIEW’ de yapılmış bir programa ait düğüm noktaları, veri girişlerinin ve çıkışlarının gösterildiği block diagram verilmiştir.



Şekil 4.2. LabVIEW’ de düğüm noktaları, veri girişlerinin ve çıkışlarının gösterildiği block diagram.

Çoğu insan bilinçli düşünmeye hiç ihtiyaç duymadan görüntüleri görerek ve işleyerek öğrenirler. Görsel olarak oluşan bu süreç diyagramları genellikle problemlere çözüm üretmeye olanak sağlar. Bununla birlikte, çoğu genel amaçlı programlama dilleri, o dille ilişkili belirli metin tabanlı sözdizimini öğrenmek için önemli zaman harcamanızı ve daha sonra dilin yapısını çözmekte olan sorunla eşleştirmenizi gerektirir. G kodu ile daha sezgisel bir deneyim sunar. G kodunu blok diyagramları, akış süreçleri ve görevleri açısından diyagramsal olarak görselleştirmeyi anlamak kolaydır.

LabVIEW geleneksel programlama araç çubuğunda bulunan özellikleriyle karşılaştırılabilir özellikler sunar. Araç çubuğu ile programın birçok bölümünde aynı anda verileri inceleyebilir, yürütmeyi duraklatabilir ve karmaşık programlama olmadan bir altprogramı adım adım uygulayabilirsiniz. Diğer programlama dillerinde mümkün olmakla birlikte, programın durumunu ve kodun paralel kısımları arasındaki ilişkileri görselleştirmek daha kolaydır. Şekil 4.3.' de LabVIEW ara yüzüne ait araç çubuğu verilmiştir.

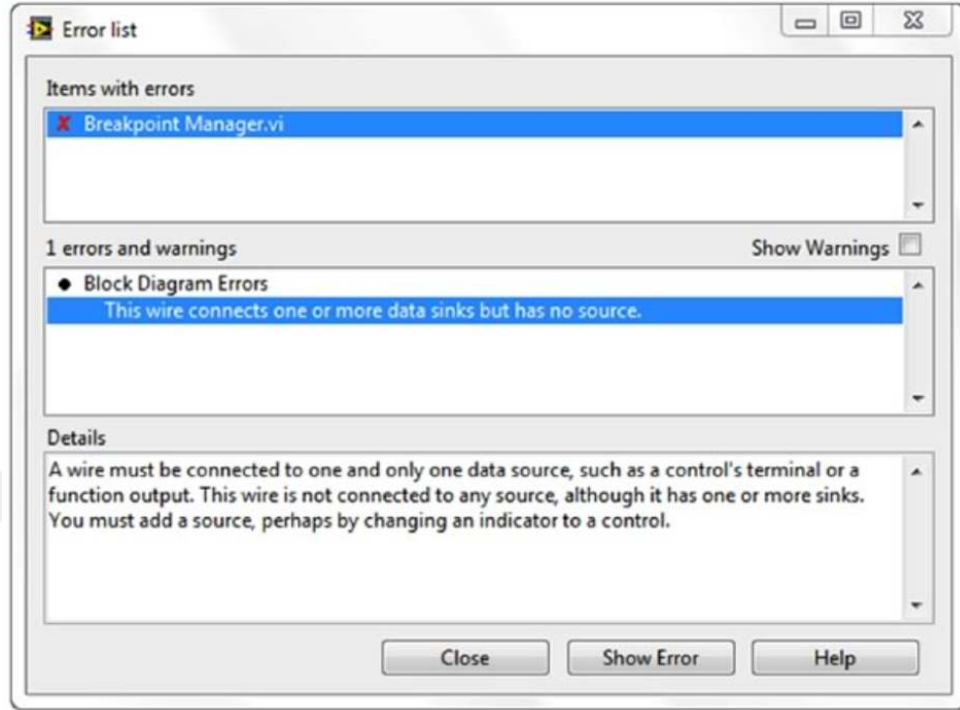


Şekil 4.3. LabVIEW ara yüzüne ait araç çubuğu.

LabVIEW uygulamasında kullanılan en yaygın hata ayıklama özelliklerinden biri daima derleyici üzerinde olmasıdır. Bir program geliştirirken, derleyici hataları sürekli olarak denetler ve uygulama hakkında anlamsal ve söz dizimsel geribildirim sağlar. Bir hata varsa programı çalıştıramazsınız ve araç çubuğunda yalnızca kırılmış bir çalıştır düğmesini görürsünüz. Kırılmış çalıştır düğmesi, kontrol etmeniz gereken bir sorunların listesi olan “Error List” panelini açar. Bu sorunlara çözüm bulduktan sonra LabVIEW derleyicisi programınızı makine koduna derleyebilir. Bir kez derlendiğinde G programlarının performansı, C gibi daha geleneksel metin tabanlı dillerle karşılaştırılabilir niteliktedir. Şekil 4.4.' de kırılmış çalıştır düğmesinin ve error list panelinin gösterimleri verilmiştir.



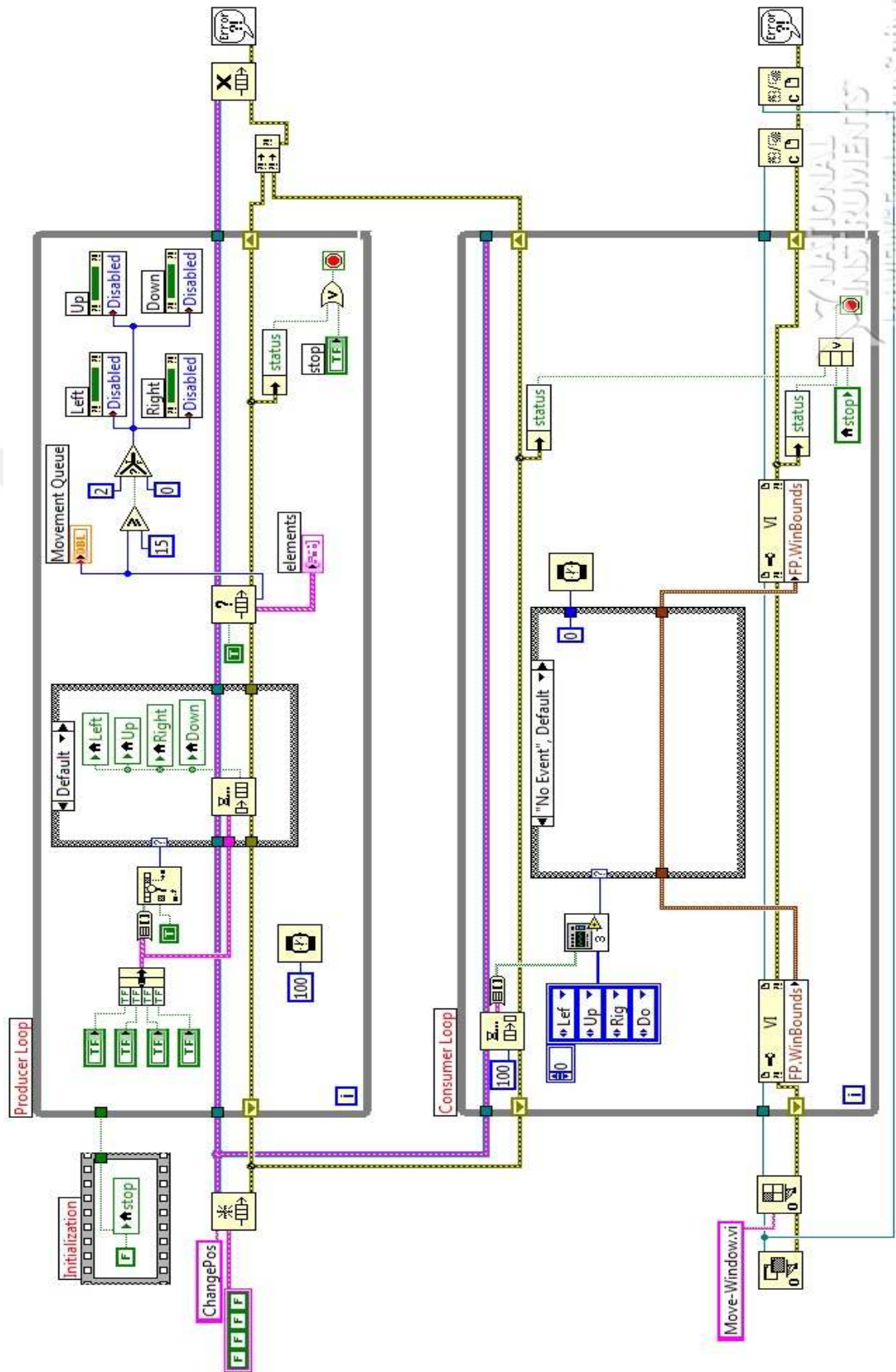
(a)



(b)

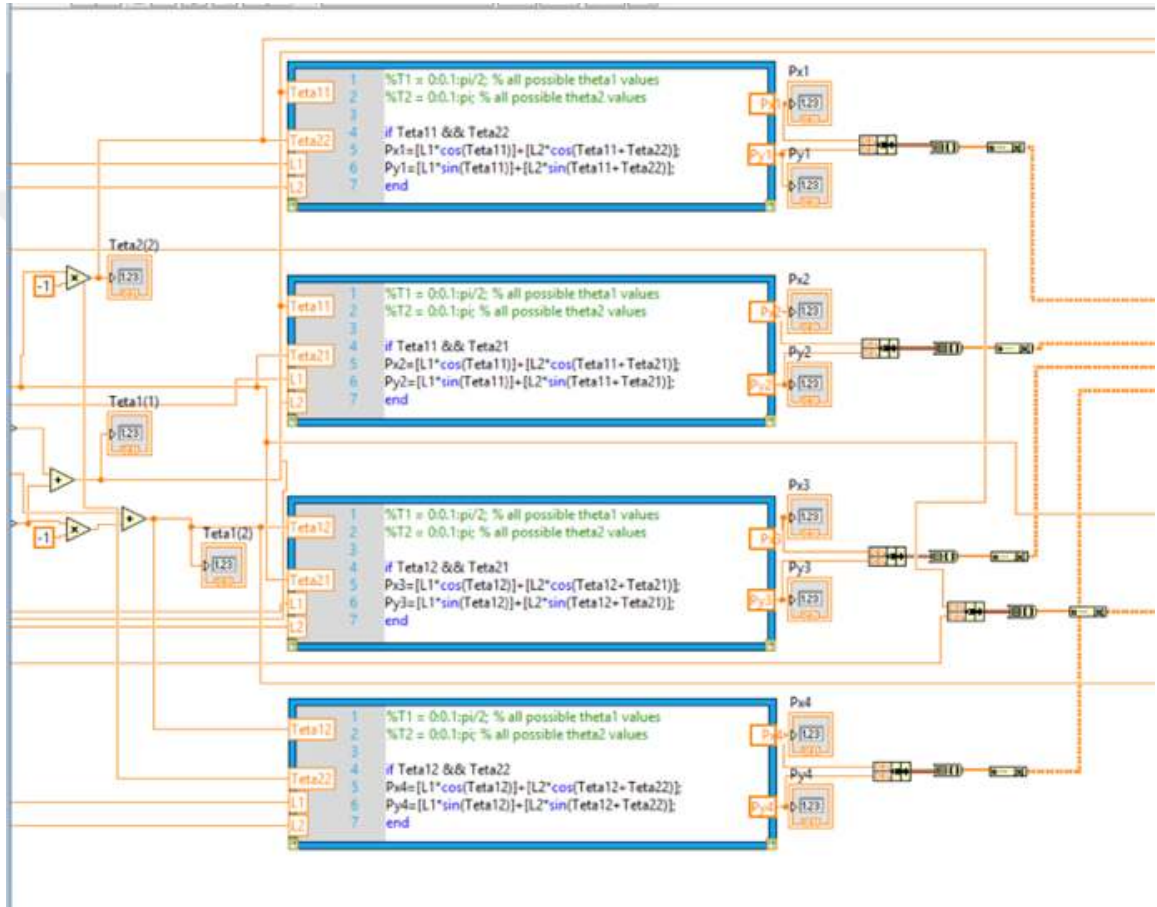
Şekil 4.4. a) Kırılmış çalıştır düğmesi. b) Error list paneli.

LabVIEW gibi veri akışı dilleri otomatik paralelleştirmeye izin verir. C ve C++ gibi sıralı dillerin aksine LabVIEW, kodun hangi bölümlerinin paralel olarak yürütüleceği hakkında bilgi verir. Farklı hızlarda paralel olarak yürütülmesine rağmen, genel amaçlı programlama dillerindeki gibi standart veri yapıları kullanılarak iki döngü arasında veri iletilir. Şekil 4.5.' de iki while döngüsü kullanılarak yapılmış veri iletimi örneği gösterilmiştir.



Şekil 4.5. Döngüler arası veri iletimi.

G kodunuzu diğer dillerle birleştirmek mümkündür. Özellikle, matematiksel formüller ve denklemler genellikle metin ile daha özlü bir şekilde temsil edilebilir. Bu sebeple LabVIEW'i kullanarak grafiksel programını çeşitli metin tabanlı programlama biçimleriyle birleştirilebilir. LabVIEW içinde çalışırken, metinsel, grafiksel veya ikisi birlikte bir yaklaşım seçilebilir. Şekil 4.6.' da LabVIEW mathscript node terminali ile metin tabanlı programlama ve G programlama birlikte verilmiştir.



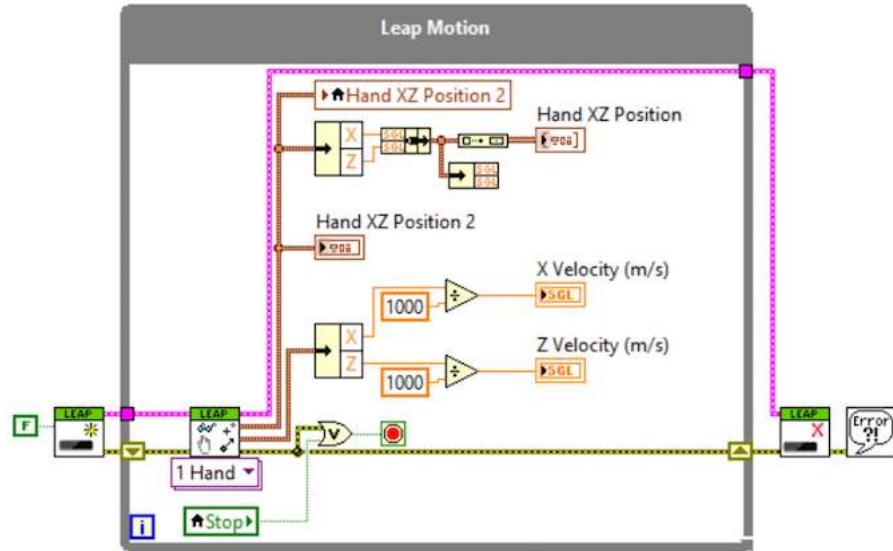
Şekil 4.6. LabVIEW mathscript node terminali ve G programlama.

LabVIEW ile tüm donanımınızı tek bir geliştirme ortamında kullanmak mümkündür. LabVIEW ile donanımınız arasındaki iletişimi sağlayan sürücü yazılımı ile mümkün hale gelir. LabVIEW sürücü yazılımı, veri toplama cihazları da dahil olmak üzere çok fazla araç ve sensör tipi arasında sorunsuz entegrasyon sağlar. Modüler aletler, hareket kontrol cihazları, motor tahrikleri, makine öğrenme ve görüntü işleme donanımları, kablosuz sensörlerin entegrasyonu mümkündür. Kullanılacak entegrasyonun bir LabVIEW sürücüsü mevcut değilse, sürücüler diğer programlama

dillerinden aktarabilir veya kendi sürücünüzü uygulamak için bir iletişim protolü kullanılarak aktarılabilir.

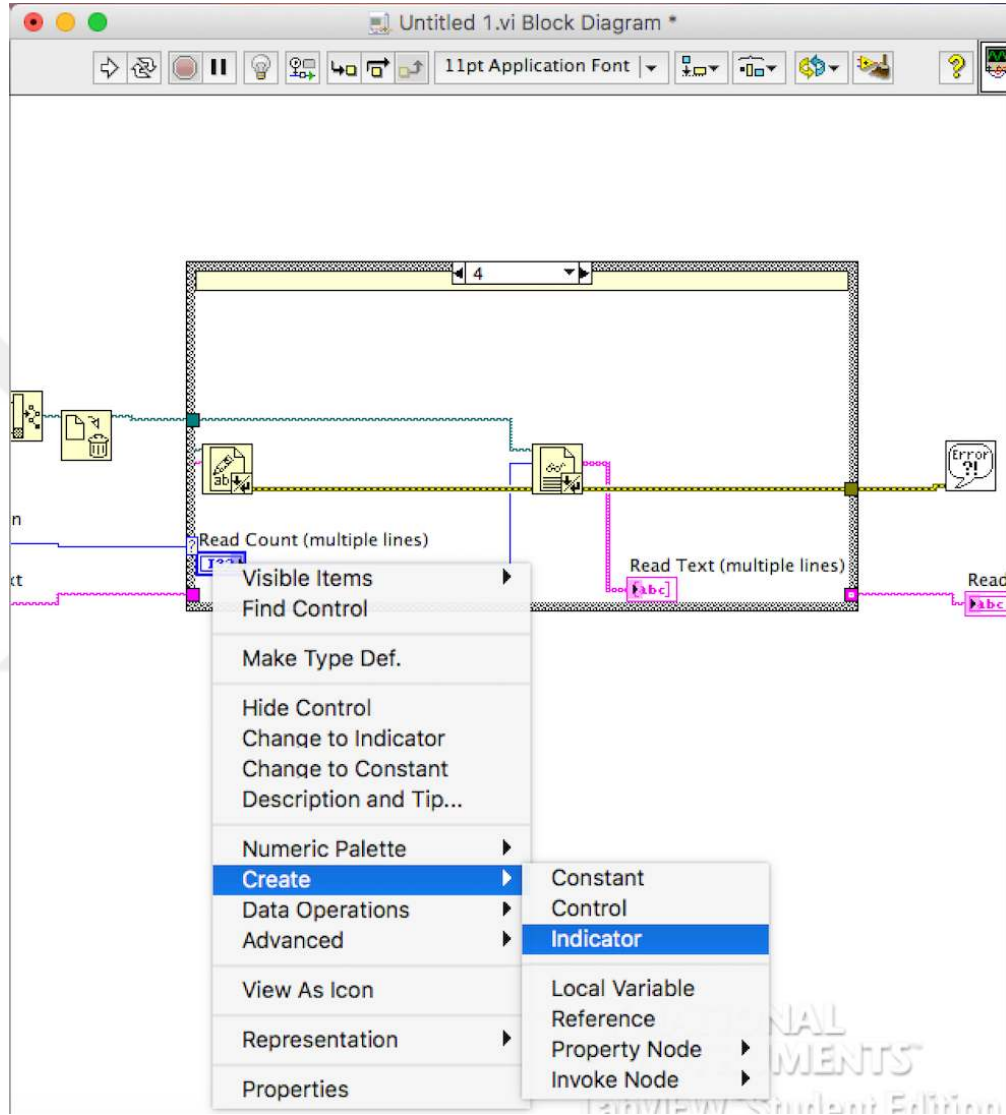
LabVIEW ile veri toplama cihazları USB(Universal Serial Bus), PCI(Peripheral Component Interconnect), PCI Express, PXI(PXI Extensions for Instrumentation), PXI Express, Ethernet gibi çeşitli haberleşme yollarıyla entegre etmek mümkündür. Veri toplama cihazları dışında dijital multimetreler, yüksek hızlı osiloskoplar, RF(Radio Frequency) sinyal analizatörleri, RF sinyal jeneratörleri, sinyal üreticileri, yüksek hızlı dijital giriş-çıkışlar, anahtarlama elemanları, programlanabilir güç kaynakları, yeniden yapılandırılabilir FPGA(Field Programmable Gate Array) giriş-çıkışları, hareket kontrol cihazları, görüş sistemleri gibi çeşitli donanımları LabVIEW'e entegre ettiğiniz donanım aygıtını yapılandırmak ve denetlemek için işlevleri blok şemasına sürükleyip bırakmanız yeterlidir.

Donanım sürücülerinizin tamamı yüklendikten sonra özel olarak talimatları eklemenize gerek kalmadan kullanılabilir. Dokümantasyona bakılmaksızın her bir LabVIEW işlevi kendi bağlamsal yardım girişine sahiptir. İşlevin üzerine farenizi getirdiğinizde işlevinin ne yaptığını ve hangi parametrelerle çalıştığını görebilirsiniz. Şekil 4.7.' de Leap Motion donanımıyla konum bilgisi veren program verilmiştir.



Şekil 4.7. Leap Motion Cihazının entegrasyon blok şeması.

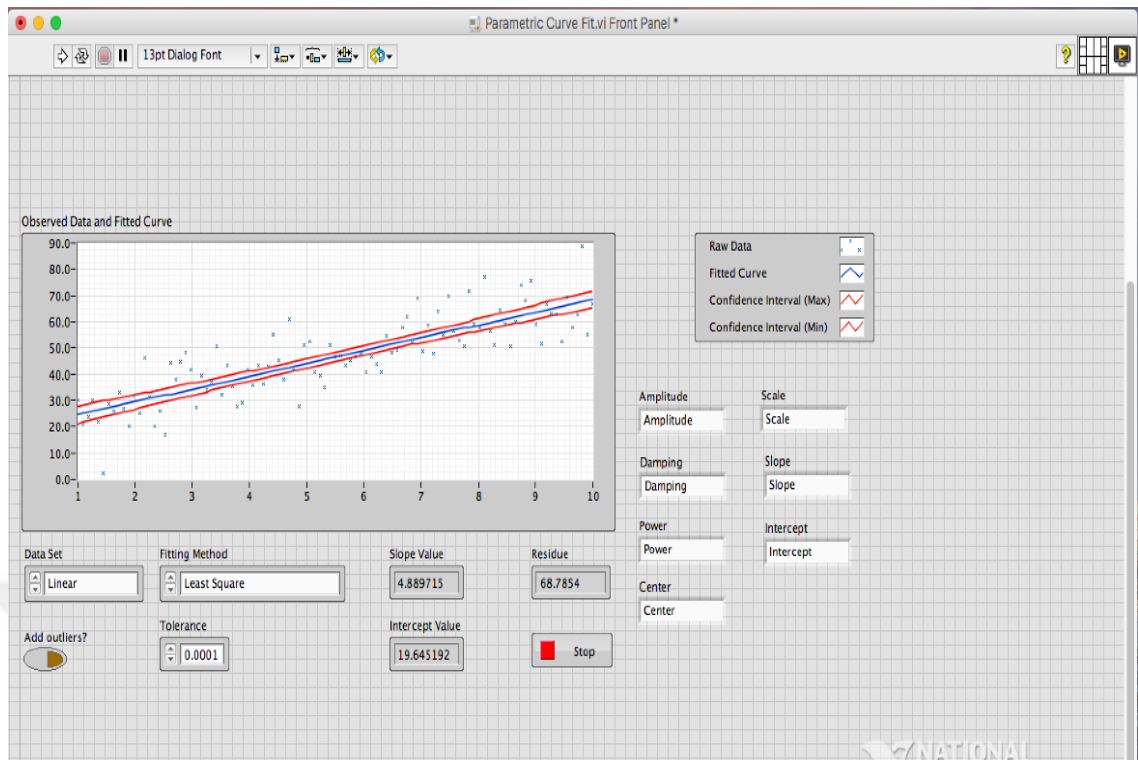
LabVIEW ile edinmiş olduğunuz ya da işlediğiniz tüm verileri blok şemasında görüntüleyebilir ve yeni bir gösterge oluşturmak için herhangi bir şeyi dönüştürmeniz veya kod yazmanız gerekmez. Block diagramda seçtiğiniz bloğa sağ tıklayıp “Create→Indicator” seçerek gösterge oluşturabilirsiniz. Şekil 4.8.’ de gösterge oluşturma gösterilmiştir.



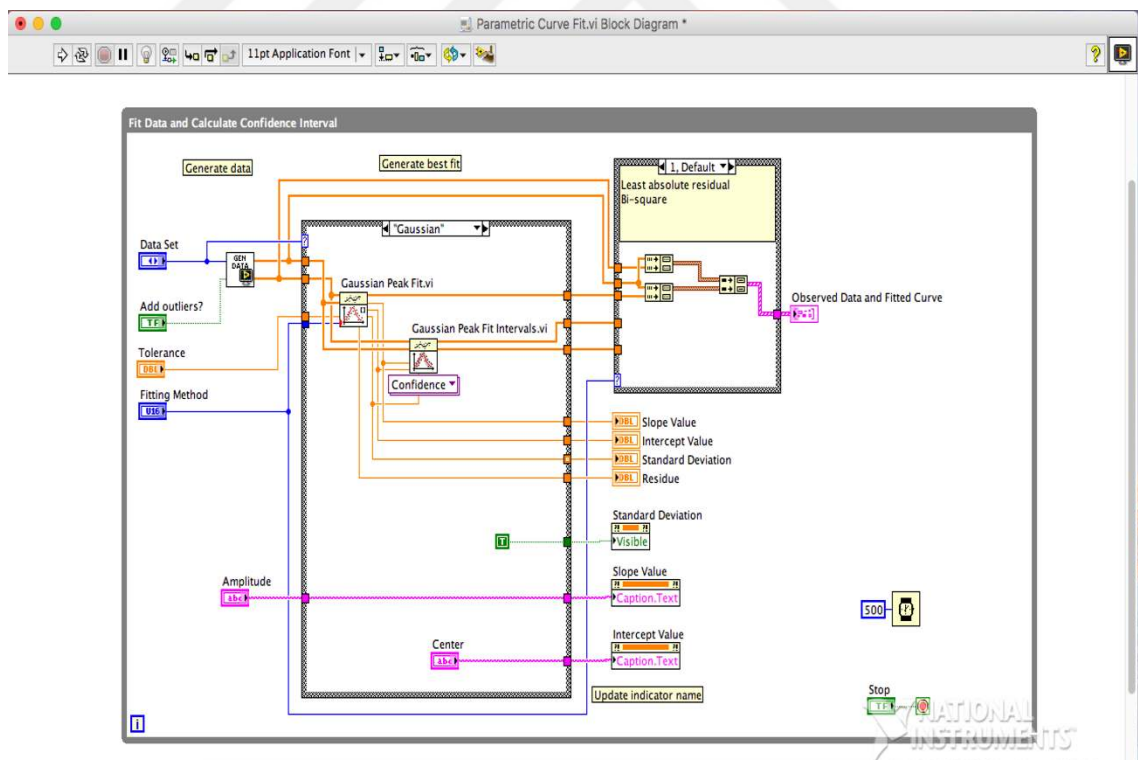
Şekil 4.8. Gösterge oluşturma.

4.2. LabVIEW Kullanıcı Arayüzü

LabVIEW arayüzü “Block Diagram” ve “Front Panel” olmak üzere iki ana kısımdan oluşur. Front panel kullanıcı ara yüzünün hazırlandığı ve göstergelerin bulunduğu kısımdır. Block diagram ise görsel programlamanın yapıldığı kısımdır. Şekil 4.9.’ da Block diagram ve front panel gösterilmiştir.



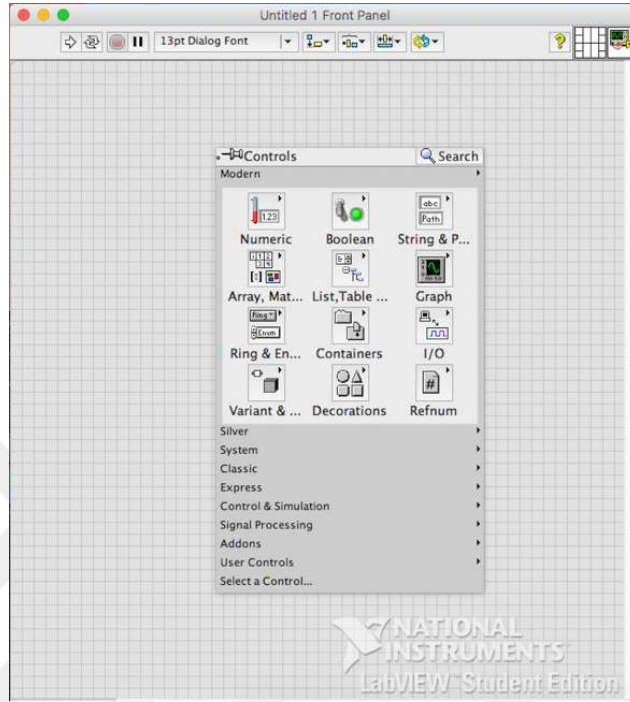
(a)



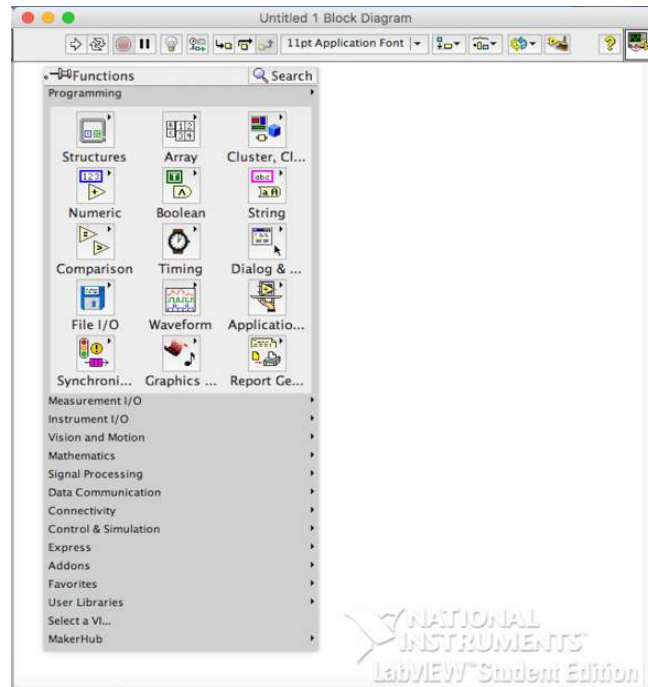
(b)

Şekil 4.9. a) Front panel. b) Block diagram.

LabVIEW’ de 3 ayrı araç paleti bulunmaktadır. Front panelde kullanılan palete “Controls”, block diagramda kullanılan palete “Functions” ve her iki panelde de bulunan block diagram ve front paneldeki programın çalışmasını ve düzenlenmesini sağlayan “Tool” bulunmaktadır. Şekil 4.10.’ da controls ve functions gösterilmiştir.



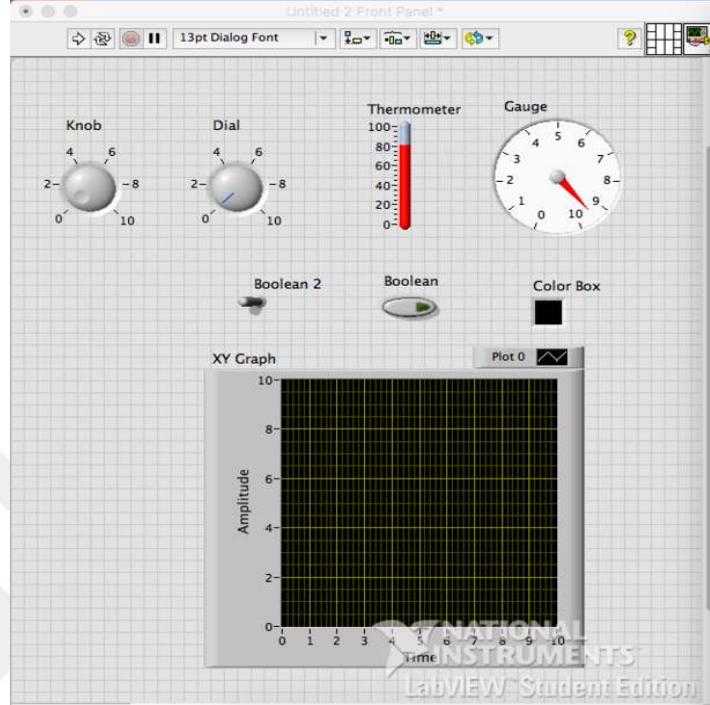
(a)



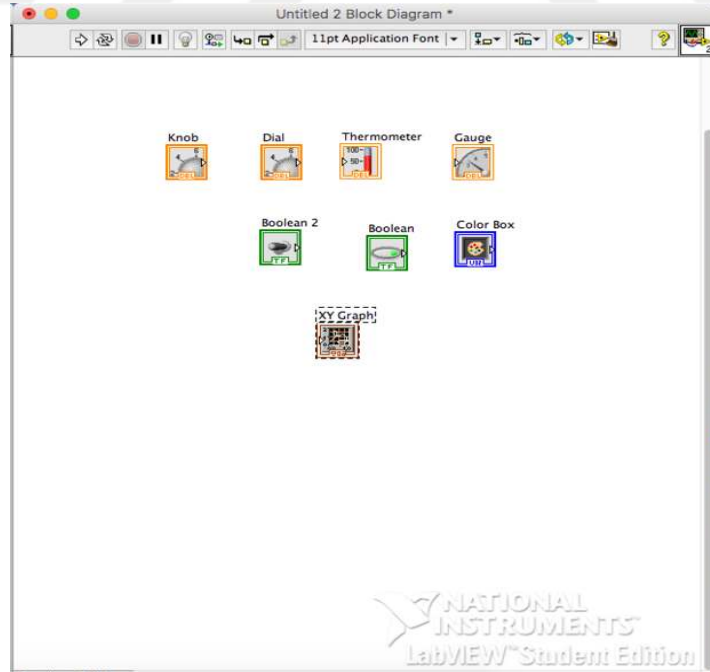
(b)

Şekil 4.10. a) Controls. b) Functions.

Front panelde kullanılan göstergelerin ve kontrollerin her biri, block diagramda terminal olarak tanımlanmıştır. Şekil 4.11.' de Front panelde seçilmiş her bir göstergelerin ve kontrollerin block diagramdaki terminal karşılığı gösterilmiştir.



(a)



(b)

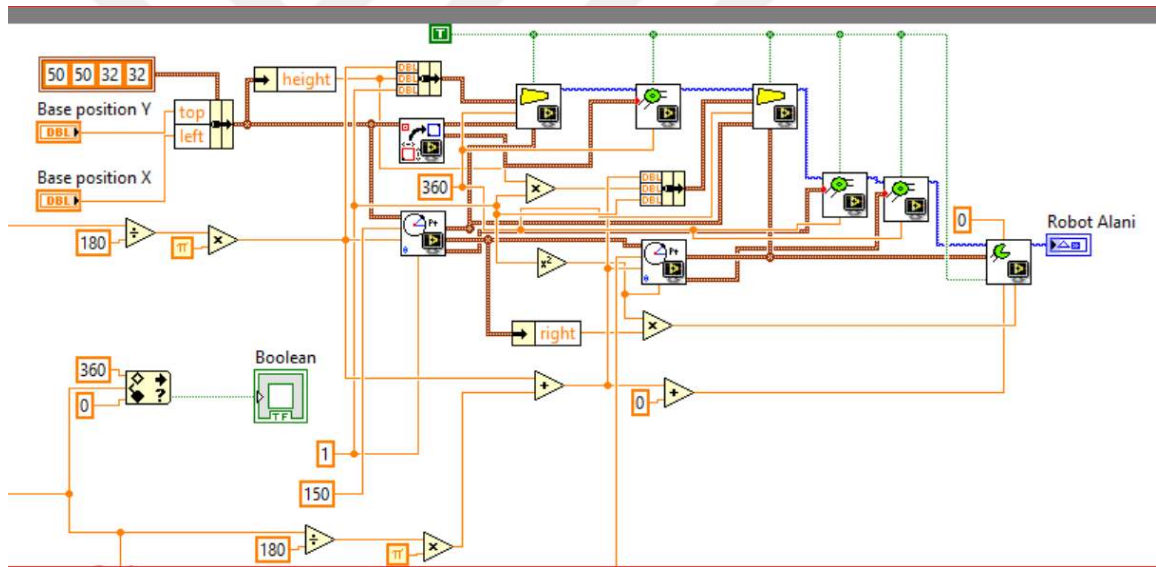
Şekil 4.11. a) Göstergeler ve kontroller. b) Terminaller

Bu çalışmada 2B robot kolu tasarımı, leap motion cihazının entegrasyonu ve robot kol kinematik hesapları LabVIEW ile yapılmıştır.

4.3. 2B Robot Kolunun Tasarlanması

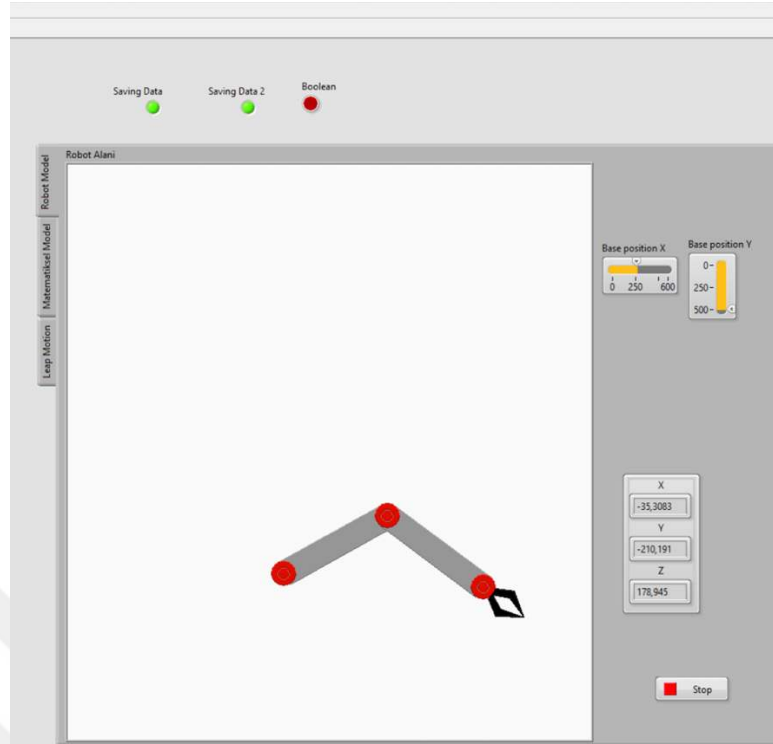
Üç serbestlik derecesine sahip RRP tipli 3B bir robot kolunun hareketi z-ekseni boyunca sabit düzlem üzerinde kalması sebebiyle üstten görünümü baz alınarak, iki serbestlik derecesine sahip 2B simülasyonu LabVIEW ortamında oluşturuldu.

Tasarlanan 2B robot kolu düzlemde dairesel olarak hareket etmekte olup robotumuza ait her bir uzuv ve eklem kendine özgü özellikleriyle ve arzu edildiğinde bu özellikleri değiştirebileceğimiz şekilde tasarlandı. Tasarlanan robot RR tipte ve 2B robot kolu simülasyonun block diagramı Şekil 4.12.' de verilmiştir.



Şekil 4.12. 2B Robot kolun simülasyonunun tasarımı.

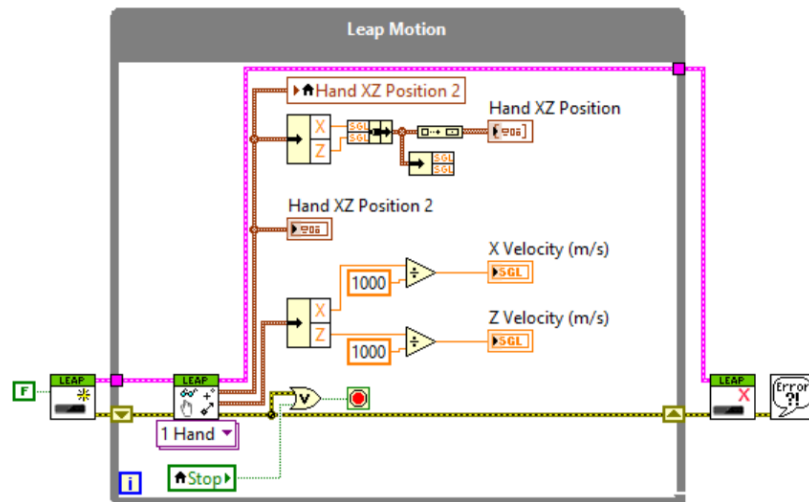
LabVIEW programında yeni veya var olan bir VI(Virtual Instrument) dosyasını açtığınızda, VI' in front panel penceresi açılır. Front panel penceresinde matematiksel modeller, Leap Motion' dan alınan konum bilgileri ve 2B robot kolunun hareketinin görülebileceği ara yüz oluşturulmuştur. Şekil 4.13.' de 2B robot kolunun kullanıcı ara yüzü bulunmaktadır.



Şekil 4.13. 2B robot kolunun kullanıcı ara yüzü.

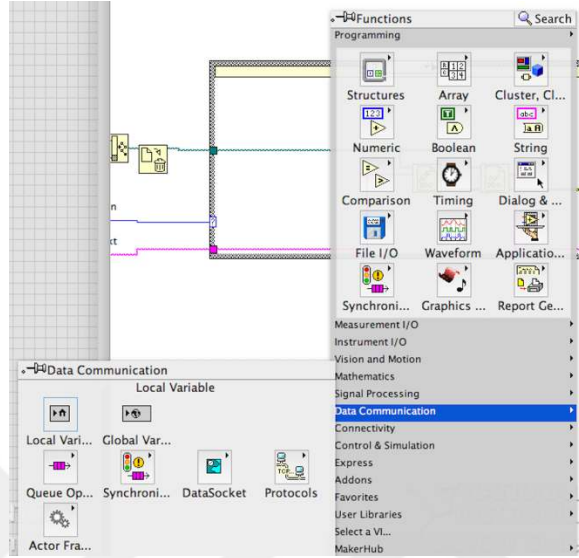
4.4. Leap Motion Entegrasyonu

2B robot kolun, Leap Motion cihazı ile bağlantısı LabVIEW ile sağlanır. Bağlantısı sağlanan Leap Motion cihazı ile elin algılanması sağlanır ve elin konum verisi elde edilir. Bu çalışmada Şekil 4.14.' deki gibi Leap Motion ile konum verisi elde edilmiştir.

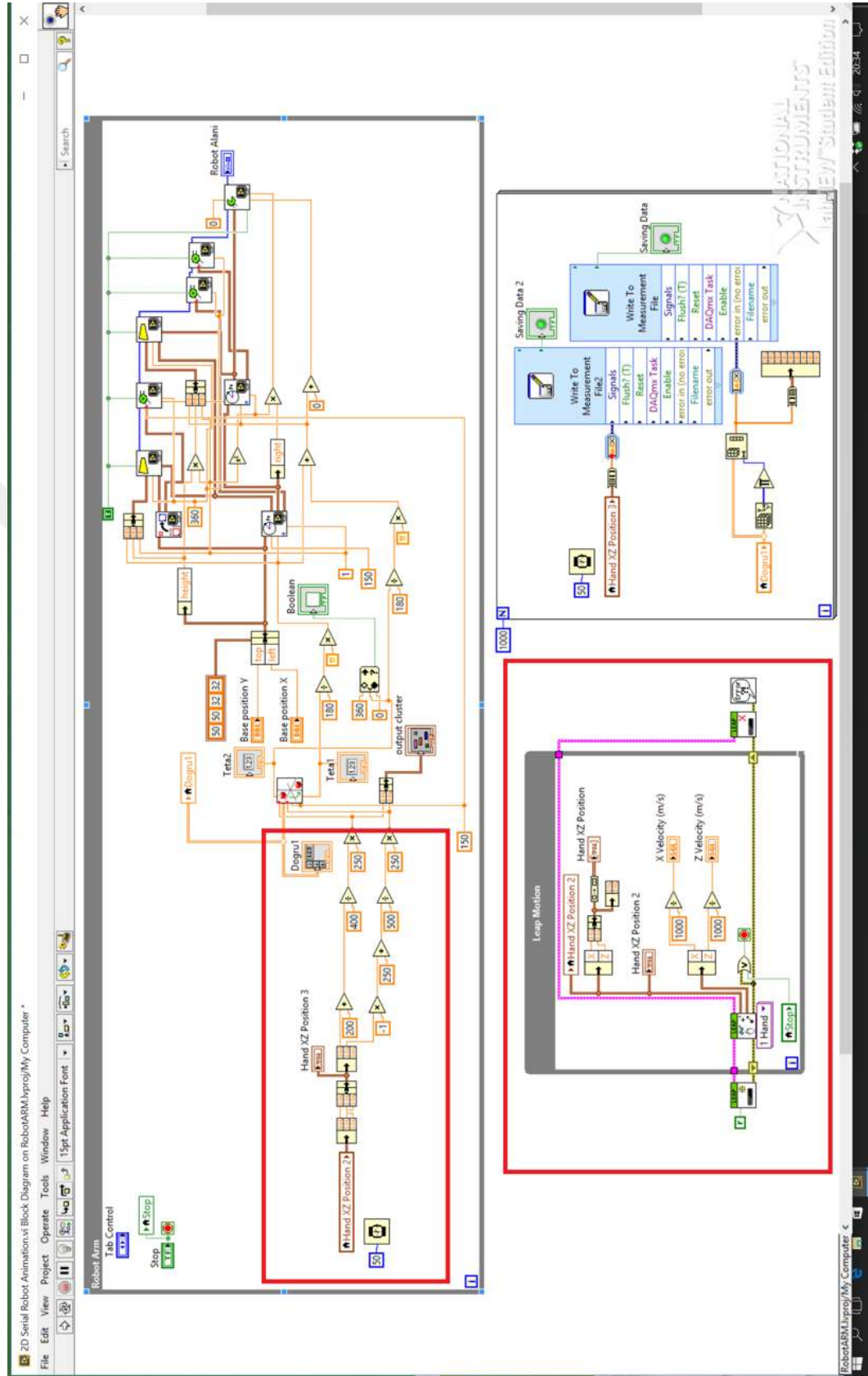


Şekil 4.14. Leap Motion konum algılama.

LabVIEW programında Leap Motion ile elde edilen konumları yine LabVIEW ile Şekil 4.15.' deki gibi lokal değişken kullanılarak robot kola entegresini sağlandı. Şekil 4.16.' da Leap Motion' ın robot kola entegre edilmesi gösterilmiştir.

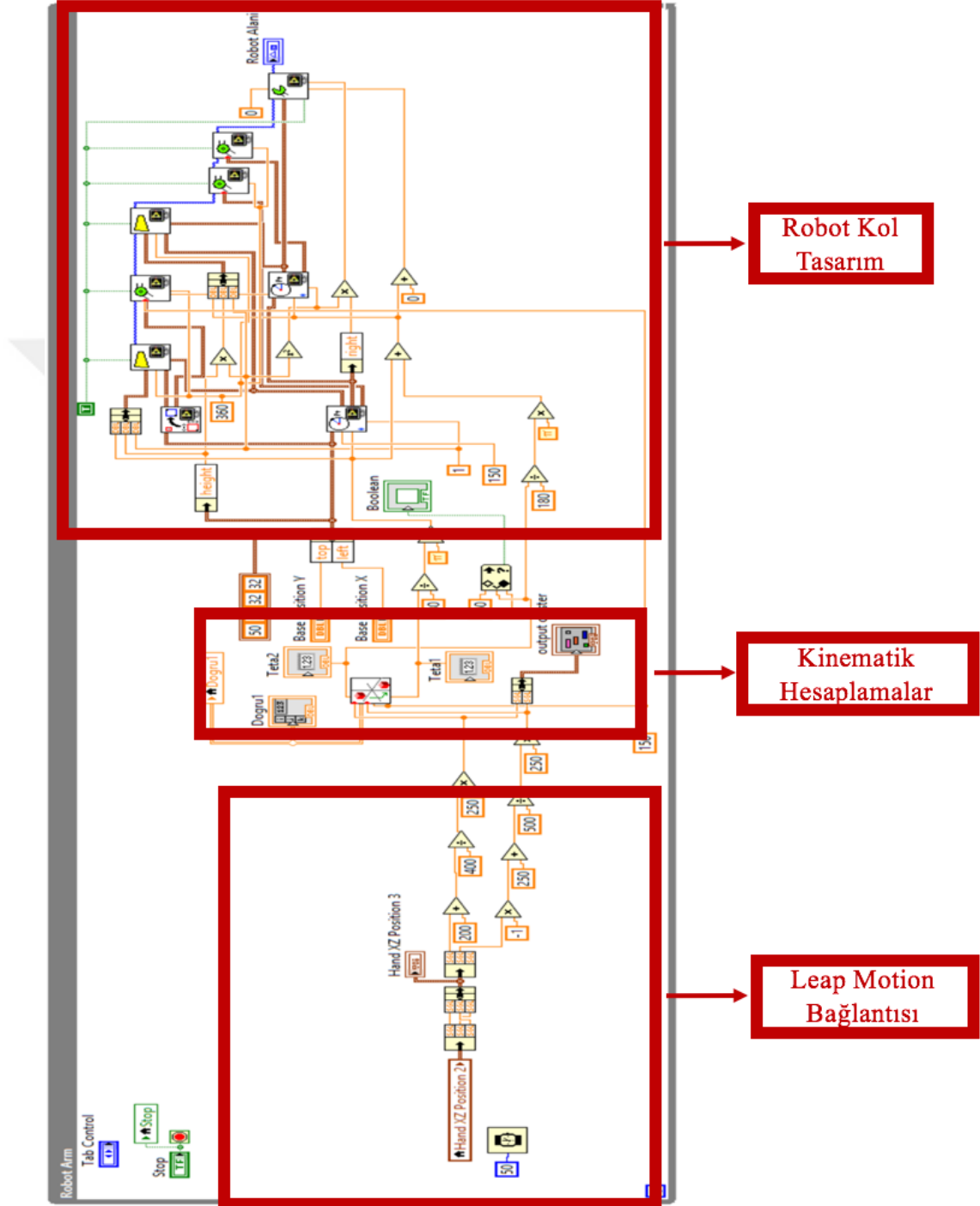


Şekil 4.15. Lokal değişken kullanımı.



Şekil 4.16. Leap Motion entegrasyonu.

Şekil 4.17’ de Leap Motion cihazı bağlantısı yapılarak alınan veriler ile kinematik hesaplamalar yapılarak robot kol simülasyonunun hareketi sağlanmıştır.



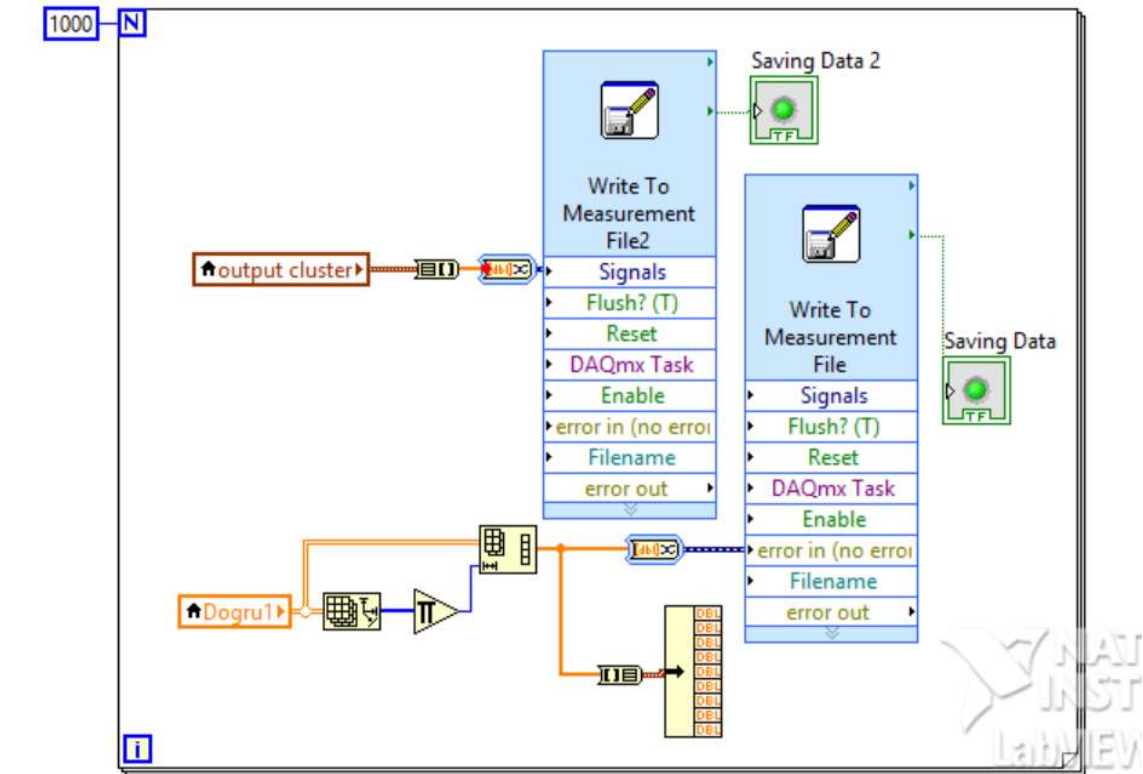
Şekil 4.17. 2B robot kolun simülasyonu.

5. MUTLAK HATA ORANI ORTALAMASI

Mutlak hata oranı ortalaması(MAPE) hata oranının doğruluğunu tanımlar. Hesaplanan değer yüzde olarak değerlendirilir ve elde edilen sonucun değeri ne ise sistemin yanılma değeri o kabul edilir. (5.1) deki gibi hesaplanır.

$$\sum_{i=1}^n \left| \frac{(y_i - \hat{y}_i)}{y_i} \right| \times 100 \quad (5.1)$$

Burada x_i ve y_i Leap Motion’ dan alınan verilere, $\hat{x}_i$ ve $\hat{y}_i$ kinematik veriler sonucunda elde edilen verilere ve n ise alınan veri sayısına karşılık gelmektedir. Bu denklem x ve y konumlarının doğruluğunun hesaplanması için ayrı ayrı uygulanmıştır. MAPE hesabı için kullanılacak verilerin LabVIEW ile kaydedilmesi Şekil 5.1.’ deki gibi sağlanmıştır.



Şekil 5.1. LabVIEW ile mutlak hata oranının ortalaması.

Sistem üzerinde yapılan bir deneme sonrasında Leap Motion’ dan alınan gerçek verilere karşılık kinematik hesaplamalar ile bulunan yaklaşık veriler Şekil 5.2.’ de gösterilmiştir.

0	0
0	0
0	0
0	0
0	0
-190,507	213,14014
-141,347	205,74338
-138,499	204,14632
-137,551	193,36691
-136,785	184,95348
-134,884	173,40381
-105,247	167,12276
-101,524	154,68257
-98,864	153,6985
-97,9526	153,10767
-97,1861	149,84264
-96,5663	149,66429
-94,648	148,34439
-94,0532	147,76808
-88,2641	145,5721
-78,4433	145,5094
-51,4329	144,71611
-48,3839	144,22408
-45,118	142,49847
-29,306	141,13376
-27,1532	139,11733
-23,1739	137,31015
2,763831	134,40137
17,27769	134,75339
22,39094	133,00536
25,44671	132,3
27,50989	129,24229
30,1305	125,8326

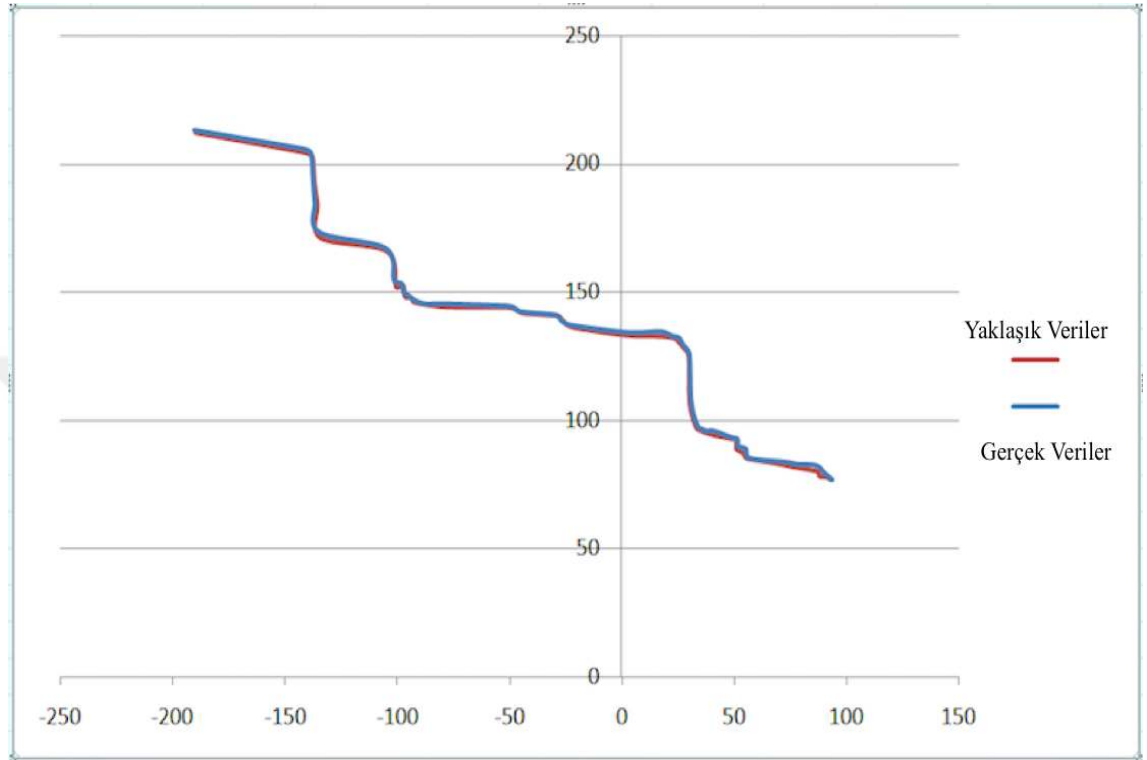
(a)

#NV	#NV
#NV	#NV
#NV	#NV
#NV	#NV
#NV	#NV
#NV	#NV
-189,968	212,1009
-141,194	204,3387
-138,008	202,9199
-136,952	192,7754
-135,701	184,0112
-134,234	171,1439
-104,745	165,9213
-100,576	152,0989
-97,99	152,779
-97,1977	151,9108
-96,5633	148,1008
-94,1669	148,0048
-93,1896	146,9916
-92,8789	146,1781
-87,6755	145,1789
-77,6788	144,1953
-50,7379	143,9891
-47,1568	143,1322
-44,1891	141,775
-28,7366	140,6767
-26,3457	138,7876
-22,4606	136,399
2,010987	133,1422
16,67727	132,7852
22,97765	132,1057
24,67655	131,199
27,15767	128,5694
29,76579	125,0129

(b)

Şekil 5.2. a) Gerçek veriler-I. b) Yaklaşık veriler-I.

Şekil 5.3.' de, Şekil 5.2.' deki verilerden yararlanılarak gerçek veriler ve yaklaşık verilerin grafiği çizdirilerek doğruluğu gösterilmiştir. Bu deneme sonucunda alınan 67 veri için x konumunda %3' lük, y konumunda %4' lük bir hata oranı hesaplanmıştır.



Şekil 5.3. Verilerin doğruluğu-I

Sistem üzerinde yapılan başka bir denemede sonucunda Şekil 5.4.' deki veriler elde edilmiştir. Leap Motion' dan alınan verilere karşılık gelen kinematik hesaplar sonrasında elde edilen veriler gecikmeli olduğunda, gecikmeli veriler silinerek MAPE hesabı yapılmıştır.

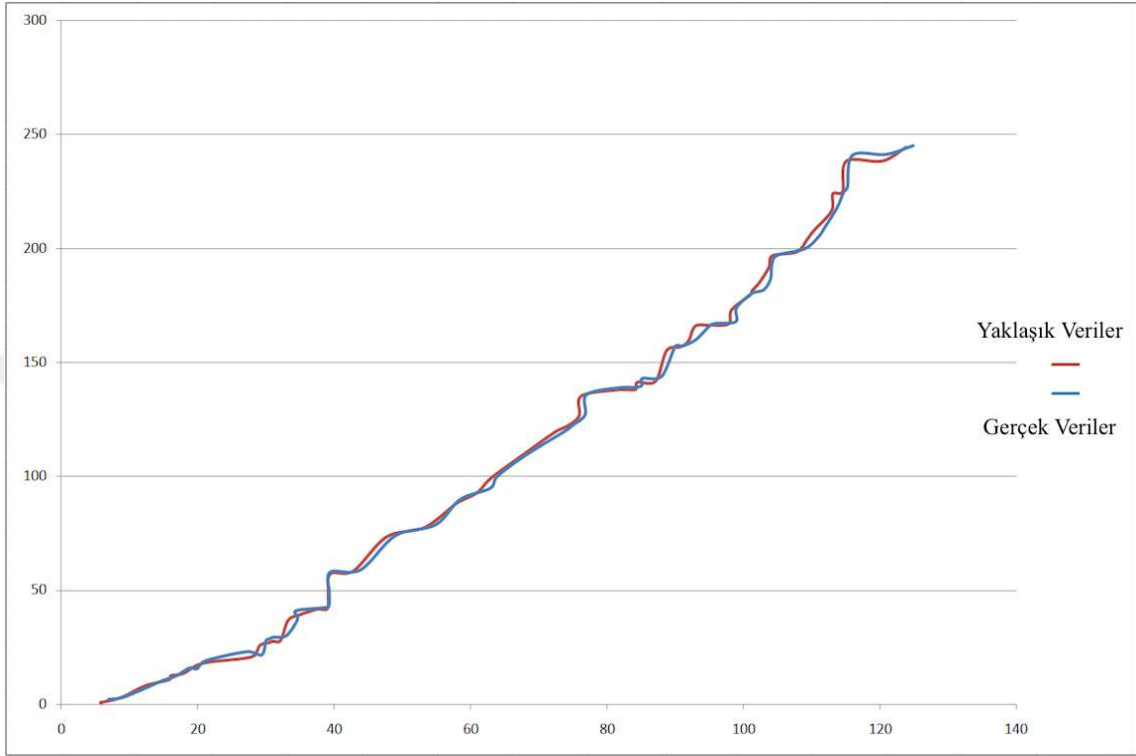
0	0	#NV	#NV
6,873863	1,989738	#NV	#NV
9,324782	3,198313	5,674755	0,730537
14,92372	10,36716	8,742364	2,996895
16,74544	12,1768	12,22563	8,079943
18,74827	15,77232	15,74659	10,77906
19,89474	15,49829	16,00368	12,55833
21,11364	18,92917	17,82362	13,50898
27,17183	22,93717	20,63514	17,95199
29,36416	21,51657	26,30827	19,95717
29,99238	27,72893	28,21874	21,34162
30,46612	28,61265	29,09283	25,88742
31,14271	29,35378	30,38267	26,99757
32,99278	30,01514	30,93862	27,67192
34,61879	37,18763	32,10193	28,02059
34,38172	40,91782	33,21934	36,82596
38,17812	42,18781	34,92837	39,23614
39,19819	42,9082	37,37489	41,60231
39,28928	48,82178	38,98383	42,0165
39,39864	57,92871	39,10387	48,21747
43,87238	58,90216	39,39864	57,2458
48,97182	73,92867	42,7824	58,47815
54,87892	78,71933	47,67323	73,45479
58,47971	89,88162	53,27773	77,57266
62,87382	94,72897	57,92782	88,12136
63,93748	99,99272	60,92246	92,72656
67,99851	109,109	62,92752	99,03628
73,92392	120,0002	67,39274	108,8096
74,92782	122,3656	72,35462	119,3743
76,79387	126,9824	74,10837	121,9161
77,01538	135,9287	75,89184	126,3968
81,92736	138,9284	76,20188	135,1428
83,99922	138,9893	81,293	137,8419
85,09199	139,9013	82,87494	137,943
85,17823	142,983	84,19975	138,1082
87,99991	143,8719	84,48365	141,2325
89,98987	157,0002	87,11837	141,6747
91,01334	157,1518	88,73647	155,0647
92,98199	159,9182	90,76373	156,7565
94,99798	165,9188	91,93781	159,3201
95,93362	167,1436	92,93126	165,8294

(a)

(b)

Şekil 5.4. a) Gerçek veriler-II. b) Yaklaşık veriler-II.

Şekil 5.5.' de, Şekil 5.4.' deki verilerden yararlanılarak yaklaşık verilerin doğruluğu gösterilmiştir. Bu deneme sonucunda alınan 102 gerçek veri için x konumunda %5' lik, y konumunda %7' lik bir hata oranı hesaplanmıştır.



Şekil 5.5. Verilerin doğruluğu-II

Kullanıcı Leap Motion cihazının çalışma alanı dışına çıktığında sık ve düzgün veriler alınamadığından MAPE hesabı yapılmamıştır. Şekil 5.6.' da MAPE hesabı yapılmamış veri tablosu gösterilmiştir.

101,2874	181,3763
102,3465	184,9465
#NV	#NV
#NV	#NV
103,7524	191,6374
104,2637	196,675
107,8493	198,3716
109,3152	203,6835
110,3262	207,6898
#NV	#NV
#NV	#NV
#NV	#NV
#NV	#NV
#NV	#NV
#NV	#NV
112,9387	216,3776
113,1198	223,8614
#NV	#NV
#NV	#NV
#NV	#NV
#NV	#NV
#NV	#NV
#NV	#NV
#NV	#NV
#NV	#NV
#NV	#NV
114,5747	224,8782
115,0932	238,162
120,3866	238,2264
123,8739	244,2698
#NV	#NV
#NV	#NV
#NV	#NV
#NV	#NV
#NV	#NV
#NV	#NV
#NV	#NV

Şekil 5.6. Hesabı yapılmamış veri tablosu

6. SONUÇ VE ÖNERİLER

Hareket boyunca Leap Motion' dan alınan verilerle, ters ve ileri kinematik ile LabVIEW ortamında hesaplanarak elde edilen kartezyen koordinatlar arasındaki hatalar iki boyutta Mutlak Hata Oranı Ortalaması(MAPE) ile hesaplanan sonuçlar göz önüne alındığında robot kolu hareketinin ileri bir teknoloji olan Leap Motion cihazı kullanılarak sağlanması, hem zaman hem de hareketin gerçek zamanlı kontrol edilebilmesi bakımından teknolojik uygulamalarda kullanımı önemli olacaktır.

x konumu için %3 ile %8 arasında hata oranı ortalaması, y konumu için %5 ile %15 arasında hata oranı ortalaması bulundu. Leap Motion' ın görüş alanının ters piramit olması sebebiyle uygulamanın alt köşelerine erişim sağlanamadığından, uzuv kontrolünün yapılacağı mesafe iyi ayarlanması gerektiği görüldü. Sistem sadece kinematik hesaplar yapılarak hareketi sağlandığından konum doğruluğunun hata oranı daha büyük çıktığı görülmüştür.

KAYNAKLAR

- Kucuk, S. and Bingul, Z., “The Inverse Kinematics Solutions of Fundamental Robot Manipulators with Offset Wrist” *Proceedings of the 2005 IEEE International Conference on Mechatronics*. Taipei, Taiwan: 2005.
- Demirci, A.E, “Design and Analysis of a Scara Robot”, Doktora Tezi, *Graduate School of Natural and Applied Sciences, Dokuz Eylül University*, 2012.
- Çelikbağ, Y., “Sayısal Sinyal İşlemci Kullanarak Scara Robot Denetimi”, Yüksek Lisans Tezi, *Fırat Üniversitesi*, 2015.
- Mousavi, A., Akbarzadeh, A., Shariatee, M. and Alimardani, S., “Design and Construction of Linear-Rotary Joint for Robotics Applications”, *Proceedings of the 3rd RSI International Conference on Robotics and Mechatronics*, Tehran, Iran, 2015.
- Gulzar M. M., Murtaza A. F., Ling Q., Javed M. Y., Rizvi S. T. H., Rana R. A., “Kinematic Modeling and Simulation of an Economical SCARA Manipulator by Pro-E and Verification using Matlab/Simulink”, *ICOSST*, 2015.
- Chia-Yu Hu, Chang-Ru Chen, Chin-Hsien Tseng, Andika Pramanta Yudha and Chung-Hsien Kuo, “Visual Servoing Spanner Picking and Placement with a Scara Manipulator”, *National Taiwan University of Science and Technology*, Taipei 106, Taiwan, 2015.
- Zeqlang He, Zonglin Li and Jiachen Ma, “Research on a High-Speed Picking-Placing Motion Of The Sorting Robot Based On The Optimal Picking Point”, *Harbin Institute of Technology*, Harbin 150001, China, 2016.
- Ana Djuric, Mirjana Filiovic, Vukica Jovanovic and Ljubinko Kevac, “Comparison Between Newton-Euler and Automatic Separation Method for SCARA Dynamic Modeling”, *IEEE*, 2016.

- R. Rascón, M.A. Reyna and J. Torres, “Tracking and Regulation Control of a Scara Robot with Application to Minimally Invasive Surgeries”, *Global Medical Engineering Physics Exchanges/Pan American Health Care Exchange*, 2016.
- Clément Gosselin, Mats Isaksson, Kristan Marlow, and Thierry Laliberté, “Workspace and Sensitivity Analysis of a Novel Nonredundant Parallel SCARA Robot Featuring Infinite Tool Rotation”, *IEEE Robotics and Automation Letters*, 2016.
- Moctezuma L.E.G., Lobov A. And Lastra J.L.M., “Free Shape Patsin Industrial Robots”, *Factory Automation Systems and Technology Lab, University of Technology Tampere*, Finland, IEEE, 2012.
- Demircioğlu B., Bülbül G., ve Köse H., “Leap Motion ile Türk Dili Tanıma”, *IEEE*, 2016.
- Branstett J., Gagneux V., Leleu A., Levadoux B., Pascale J., “ConductHome: Gesture Interface Control of Home Automation Boxes”, *International Scholarly and Scientific Research & Innovation 9*, 2015.
- Pititeeraphab Y., Choitkunnan P., Thongpance N., Kullathum K., Pintavirooj Ch., “Robot-Arm Control System Using Leap Motion Controller”, *International Conference on Biomedical Engineering*, 2016.
- Mohandes M., Aliyu S., and Deriche M., “Arabic Sign Language Recognition using the Leap Motion Controller”, *23rd International Symposium on Industrial Electronics*, 2014.
- Vargas H. F., Vivas A. O., “Gesture Recognition System for Surgical Robot’s Manipulation”, *Signal Processing and Artificial Vision Symposium on Image*, 2014.

ÖZGEÇMİŞ

Kişisel Bilgiler

Adı Soyadı : Seren EMİR

Doğum Yeri ve Tarihi : Samandağ / 1991



Eğitim Durumu

Lisans Öğrenimi : Elektrik Elektronik Mühendisliği

Bildiği Yabancı Diller : İngilizce

İş Deneyimi

Stajlar : FEB-Enerji Tic. Ltd. Şti / Reel Sistem Otomasyon Makina Tic. Ltd. Şti.

Çalıştığı Kurumlar : Eski İnşaat Mühendislik Mimarlık

İletişim

Adres : İsmetpaşa Mah. Salkım Sk. Kayıboyu Apt. Daire : 25 Bilecik/Merkez

Tel : 0(542) 548 44 61

E-Posta Adresi : serenemir@gmail.com / serenemir@engineer.com

Tarih : 29/05/2017