

# Endüstriyel Nesnelerin İnterneti Kullanılarak Delta Robot Uygulamalarına Yönelik Gerçek Zamanlı Çizim Arayüzü Geliştirilmesi

## Implementing Real-time Commands through a Drawing Application in Delta Robot through IIoT Communications

Klaus Gani<sup>1</sup>, Ramazan Kaya<sup>1</sup>, Berkan Ergün<sup>1</sup>, Ali Fuat Ergenç<sup>1</sup>

<sup>1</sup>Kontrol ve Otomasyon Mühendisliği Bölümü  
Elektrik Elektronik Fakültesi, İstanbul Teknik Üniversitesi

{gani18}@itu.edu.tr

{kayara16}@itu.edu.tr

{ergunb17}@itu.edu.tr

{aergenc}@itu.edu.tr

### Özetçe

Bu bildiride, endüstriyel robotlardan olan Delta robotuna Python programlama dili ile oluşturulan gerçek zamanlı bir çizim programı entegre edilerek robotunun rotasının belirlenmesidir. Bu doğrudan hareket denetimini sağlamak için Node-RED programlama ile GUI (Grafiksel Kullanıcı Arayüzleri) kullanarak yenilikçi bir Nesnelerin İnterneti (IoT) arayüzü tabanlı yöntem sunulmaktadır. Robot hareketini daha pratik bir şekilde denetlemek için yapılan hesaplamalarda ileri kinematik, ters kinematik, ileri jakobiyen ve ters jakobiyen kullanılmıştır. Sonuç olarak, kullanıcılara bir Delta robotunun hareketini gerçek zamanlı olarak kontrol etmenin basit ve etkili bir yolunu sunarak imalat ve otomasyon da dahil olmak üzere çeşitli endüstrilerde kullanımını kolaylaştırmaktadır.

### Abstract

In this paper, the purpose of this paper is to determine the route of the Delta robot, one of the industrial robots, by integrating a real-time drawing program created with Python programming language. To achieve this direct motion control, an innovative Internet of Things (IoT) interface-based method is presented using GUI (Graphical User Interfaces) with Node-RED programming. Forward kinematics, inverse kinematics, forward jacobian and inverse jacobian were used in the calculations to control the robot's movement in a more practical way. The end result is to provide users with a simple and effective way to control the movement of a Delta robot in real-time, facilitating its use in a variety of industries, including manufacturing and automation.

### 1. Giriş

Robotik, iş ve teknoloji üzerinde önemli bir etkiye sahip olmaktadır. Üretim, montaj, taşıma ve bir dizi farklı işleri yap-

mak üzere programlanabilen robotlar, görevlerini otomatik veya kısmen otomatik olarak tamamlayabilen makineler olarak bilinmektedir. Robotik olarak adlandırılan mühendislik alanı, özellikle otomasyon alanında, robotların üretilmesi, bakımı, kullanımı ve uygulanması ile ilgilenmektedir.

Otomasyon ve robotik, otomatik ve kısmi otomatik sistemlerin araştırma ve geliştirilmesiyle ilgilenen konulardır. Aynı zamanda internet ve bulut bilişim teknolojisinin entegrasyonu olasılığını da göz önünde bulunduran bu alanlar, insanların bu sistemlerle nasıl etkileşimde bulunduğu da dikkat etmektedir [1, 2]. Son yıllarda işletmeler, verimliliği artırmak, maliyetleri düşürmek ve üretim kalitesini yükseltmek için yollar ararken, robotların otomasyonda kullanımı geniş kapsamda artırmaktadır. Robotlar yüksek derecede doğruluk ve tutarlılıkla işleri yapabilmekte, yorgunluk hissetmeden sürekli ve dinlenmeksizin çalışabilmektedirler. Bu nedenle, özellikle büyük miktarda ürün üretiminin gerektiği imalat gibi alanlarda son derece faydalıdır. Dağıtım robotları, montaj robotları, işleme robotları ve elektrikli kaynak robotları, endüstriyel üretim süreçlerinde kullanılan sadece birkaç robottan bazılarıdır [3].

Bu çalışmanın odak noktası, yüksek hassasiyete sahip bir yapı olan delta robot denetimidir. Delta robotu denetlemek için çeşitli teknikler ve yöntemler vardır, uygulamaya ve hedefe bağlı olarak değişiklik göstermektedir. Bu durumda, kullanıcının girişine bağlı olarak çeşitli şekiller çizmek istenmektedir. Allen-Bradley PLC'ler (Programlanabilir Mantıksal Denetleyici) ve yazılımları bu robotu kontrol etmek için kullanılmaktadır. Çözüm, uç efektör için yeterli ve pratik bir rota planı oluşturmaktır. Robot, bu rota planlamasını kullanarak hataları ve sapmaları azaltmak ve nihai çizimin kalitesini artırmak için değiştirilmiş yolu izleyebilmektedir.

Delta Robot'un bir çizim içine entegre edilmesi ve aynı yolu izlemesi, çizimi veya yazıyı gerçek zamanlı olarak otomatikleştirmeye ve aynı zamanda otomatik sanatın geliştirilmesine yardımcı olmaktadır. Bu amaçla, bu ve diğer olası uygulamaların işletilebilmesine yardımcı olacak bir GUI ve IoT'nin entegre edilmesi gerekmektedir.

## 1.1. Literatür Taraması

Endüstriyel amaçlar için kullanılan delta robotlarının denetimine yönelik bir dizi farklı yöntem bulunmaktadır. Bu konuda literatürde yayınlanmış birçok makale ve dergi bulunmaktadır, bu çalışmaların çoğu delta robotlarının daha etkili bir şekilde denetlenmesi amacıyla geliştirilmiş farklı denetim stratejilerini içermektedir. Bu çeşitli yöntemler, robotların hızlarını, hassasiyetlerini ve verimliliklerini artırmak için tasarlanmıştır.

### 1.1.1. Klasik PID Denetleyici

Delta robotunun denetimi için kullanılan ana ve en eski yöntemlerden biridir. Üç denetim parametresi (oransal-integral-türevsel), PID denetleyicisi gerçek zamanlı olarak denetim sinyalini değiştirmesini sağlamak, hataları azaltmakta ve kararlılığı korumaktadır. Ancak ideal performans elde etmek için PID parametreleri hassas olarak ayarlanmalıdır. PID denetleyici, yol planlama, yörünge takibi, uç efektör pozisyon denetimi, hız denetimi ve konveyör bantlarındaki taşıma işlemleri gibi robotikte birçok kullanımı bulunmaktadır. Özellikle endüstriyel otomasyon ve imalat süreçlerinde, basitliği, etkinliği ve kurulum kolaylığı nedeniyle robot manipülatörlerin kullanılması tercih edilen bir seçenektir [4]. PID denetleyici her zaman zorlu robotik görevlerle basa çıkmak için yeterli olmayabilmektedir. Bu nedenle, modern robotik, belirli problemleri veya talepleri ele almak için model tabanlı denetleyici, adaptif denetleyici veya bulanık mantık denetleyici gibi gelişmiş kontrol yaklaşımlarını sıkça kullanmaktadır.

### 1.1.2. Bulanık Mantık Denetleyici

Bulanık mantık denetleyicisi, geleneksel yaklaşımlardan farklı olarak matematiksel modellere dayanmak yerine dilsel kurallar ve yaklaşık akıl yürütmeye dayalı bir kontrol yöntemi sunar. Bu, özellikle karmaşık ve belirsiz sistemlerde, örneğin delta robotları gibi çok serbestlik derecesine sahip yapılar için idealdir. Bulanık mantık denetleyicileri, bu tür sistemlerin dinamiklerini daha iyi anlamak ve etkili bir şekilde kontrol etmek için kullanılır ve bu nedenle günümüzde yenilikçi bir denetleme yöntemi olarak öne çıkar [5].

### 1.1.3. Yapay Zeka (AI)

Robotik araştırmaları, özellikle konum denetimi için yapay zeka tabanlı tekniklere odaklanmaktadır. Bu yaklaşım, hedeflenen sonuçları elde etmek için verilere dayalı bir işlevi yaklaşık olarak hesaplar. Ayrıca, robot manipülatörlerinin ters kinematiki yerine yapay zeka yöntemlerinin kullanılması da denemektedir. Bu gelişmeler, robotik alanında daha hassas ve akıllı denetleyici yöntemlerinin araştırılmasında önemli bir rol oynamaktadır [6].

Bu yöntemler, çoğu zaman oldukça kullanışlı olabilirler. Ancak gerçek zamanlı bir sistemde güvenlik özelliklerine sahip PLC kullanarak bu yöntemleri uygulamak zorlu olabilir. Bu nedenle, bu tür çalışmalarda ve araştırmalarda yaygın olarak GUI kullanılır [7-9]. Python, sıfırdan bir GUI oluşturmak ve aynı zamanda gerçek zamanlı çalışan bir PLC üzerinde uygulamak için mükemmel bir platform sunar. Ancak GUI'lerin birçok potansiyel avantajına rağmen, onları gerçek zamanlı denetim durumlarında kullanmak ek zorluklar doğurabilir. Bu zorluklar arasında kullanıcı girişi ile robot tepkisi arasındaki gecikmeyi yönetmek

ve denetleyici sistemiyle birlikte GUI'nin hesaplama yükünü dengelemek gibi ek sorumluluklar bulunmaktadır.

Delta robotunun denetimi, kinematik modelleme ve geribesleme denetleyicisinin bir kombinasyonu ile sağlanmakta, bu sayede motorlar ve eyleyiciler kullanılarak eklem konumları istenilen hareketi gerçekleştirecek şekilde ayarlanmaktadır. Bir sonraki yöntem ise hibrit konum/hız denetleyicisidir [7, 10]. Burada, konum denetimi ve hız denetimi ile birleştirilmekte, böylece robot istenilen yolu izlerken aynı zamanda sabit bir hızda kalmaktadır. Başka bir yöntem ise empedans denetleyicisidir. Empedans denetleyicisi, robotun son etkisörde istenen empedansı korumak için hareketini değiştirmesiyle, dışsal kuvvetlere tepki vermesine izin vermektedir [8].

Bu bildirinin geri kalanında cihazın modellemesi, ileri ve ters kinematiki, delta robotunun ileri ve ters jakobiyenleri, problem için kullanılan çizim kodu, IoT altyapısı ve sonuçlara yer verilmiştir.

## 2. Sistem Modeli ve Kinematikler

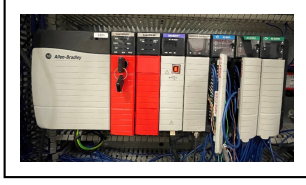
### 2.1. Sistemin Modellenmesi ve Kurulumu

Bu uygulamada kullanılan delta robot, Şekil 1'de gösterilmiştir. Delta opsiyonel olarak, bir parçayı araç ucunda döndürmek için kullanılan dördüncü bir serbestlik derecesine sahip üç serbestlik derecesine sahip bir robottur.

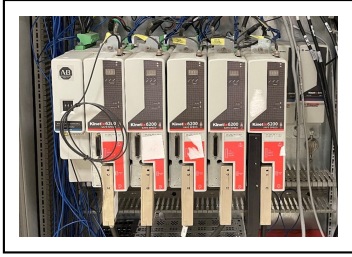


Şekil 1: Bu uygulamada kullanılan Delta Robot

Kolları karbon fiberden yapılmış olup, her kolu ayrı bir servo motor tarafından denetlenmektedir. Her servo motor, Şekil 3'te gösterilen Kinetix 6200 SERCOS sürücüsü tarafından çalıştırılmaktadır. Tüm SERCOS sürücüler, Şekil 2'de görülen ana şase ile kontaklı sürücülerini ve diğer bileşenleri içeren kontrol paneline yerleştirilmektedir. Ana şase içinde, 0 numaralı yuva için bir emniyet PLC'si bulunur ve diğer yuvalar iki dijital giriş kartı, bir dijital çıkış kartı ve bir EtherNet/IP (Ethernet Endüstriyel Protokolü) kartına ayrılmıştır.



Şekil 2: Kontrol panelinin ana şasesi

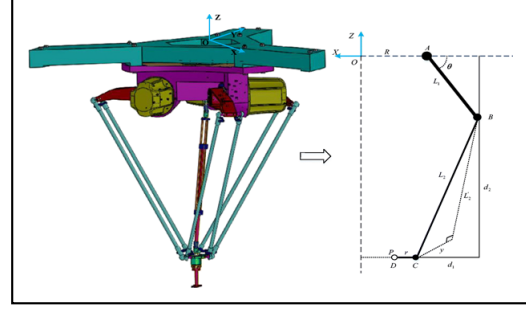


Şekil 3: Kinetix 6200 SERCOS sürücülere

Emniyet PLC'leri normal bir PLC ile aynı özelliklere ek olarak güvenlikle ilgili yeteneklere sahiptirler. Bu PLC'ler, tehlikeli maddeler veya yüksek enerjili süreçlerle ilgili sistemleri izlemek ve denetlemek için kullanılmaktadır. Özel giriş ve çıkış modülleriyle donatılmışlardır. Böylece acil durdurma veya ekipman arızası gibi güvenlikle ilgili olayları algılayabilir ve buna tepki verebilirler. Aynı zamanda uyarı ışıklarını veya alarmları etkinleştirmek, tehlikeli ekipmana gücü kesmek veya acil frenleri bırakmak gibi güvenlikle ilgili görevleri gerçekleştirmek için programlanabilirler.

## 2.2. Kinematik

Robotik uygulamanın geliştirilmesinde ileri ve ters kinematik çalışmaları son derece önemlidir. Robotun kinematik davranışının tanımsız ya da sonsuz hale geldiği noktaları yani robotun iş alanında bulunan tekil noktaları veya durumları belirlemek için kullanılır. Bu tekillikler uygun bir şekilde dikkate alınmazsa, robotun iş alanındaki sınırlarını aşmasına ve başarısızlık veya arıza durumlarına neden olabilmektedir [11]. Konformal geometrik cebir (CGA) adı verilen, üç boyutlu uzayın beş boyutlu temsilinin özel bir şekli olan cebirde bu çalışmalarda kullanılmaktadır [1]. Delta robot mekanizmasına dahil edilen kısıtlamaların geometrisini açıklamak için CGA temsil gücünden yararlanılmaktadır. İleri ve ters kinematik çalışmaları ve hesaplamaları ile CGA yaklaşımını kullanarak ileri ve ters jakobiyen hesaplamaları [1] kitabından alınmıştır. Denklemlerde kullanılan eklem adları ve sembolleri Şekil 4'te gösterilmiştir.



Şekil 4: Delta robot yapısı ve kolunun geometrik modeli [2]

### 2.2.1. İleri kinematik

Robotun eklemlerinin açıları ve bağlantı uzunlukları verildiğinde, robotun son efektör konumunu ve yönelimini belirlemek için ileri kinematik hesaplamaları, her bir kol için tek tek yapılmaktadır. Gereken motor açı konumu için düğüm noktası, aşağıdaki denklem 1'de gösterildiği gibi hesaplanabilmektedir [1].

$$\omega_i = (r_b + l \cos(\theta_i))s_i + l \sin(\theta_i)e_3 \quad (1)$$

Uç efektör plakasının yarıçapı ve yönü, her bir kol için gerçek eklem ile sahte düğüm arasında yatay bir sapma görevi görür; bu sapma, aşağıdaki denklem 2'de gösterildiği gibi hesaplanabilir [1].

$$a_i = (r_b - r_e + l \cos(\theta_i))s_i + l \sin(\theta_i)e_3 \quad (2)$$

Bundan bir CGA nokta çifti olarak bir eşdeğer CGA nokta bulunabilmekte, denklem 3 ve 4 ile bulunabilmektedir [1].

$$A_i = \frac{1}{2} a_i^2 n_\infty + a_i + n_0 \quad (3)$$

$$\sum_{n=1}^* A_i - \frac{1}{2} \rho^2 n_\infty \quad (4)$$

Bu geometrik yaklaşımla, kol ve eklem tarafından oluşturulan üç kısıtlı geometrik kürenin kesişiminden oluşan T adlı bir nokta çifti, uç efektör plakasının iki konfigürasyonunu gösterir ve denklem 5'te gösterildiği gibi hesaplanabilir [1].

$$T = I_5 \bigwedge_{i=1}^{i=3} \Sigma_i^* \quad (5)$$

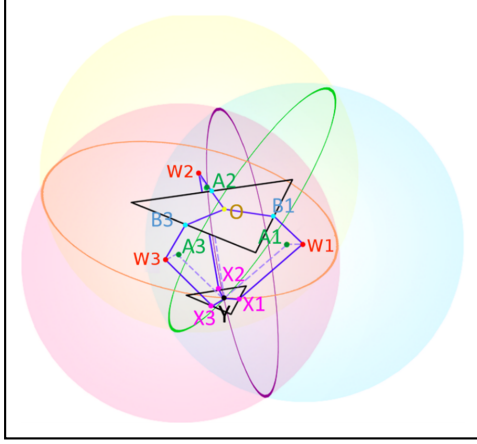
Gerekli noktanın üç boyutlu konumu (y), projeksiyonlar ile örtüştürülerek bulunabilir, denklem 6,7 ve 8'de olduğu gibi:

$$P = \frac{1}{2} \left( 1 + \frac{T}{\sqrt{T^2}} \right) \quad (6)$$

$$y = \frac{-\sum_{j=1}^{j=3} (Y \cdot e_j) e_j}{Y \cdot n_\infty} \quad (7)$$

$$Y = -P \tilde{(T n_\infty)} P \quad (8)$$

İleri kinematik hesaplama probleminin geometrik temsili Şekil 5'te gösterilmiştir [1].



Şekil 5: İleri kinematik problemi geometrik temsili [1]

### 2.2.2. Ters kinematik

Ters kinematik, ileri kinematiğin tam tersine çalışan bir kavramdır. Bu bağlamda, belirli bir son etkisör konumu ve yönelimi elde etmek için bir robotun eklemlerinin hangi açılarla pozisyonlandırılması gerektiği hesaplanır. Yani, istenen sonuca ulaşmak için robotun eklemlerinin ne olması gerektiği ters kinematik analizi ile bulunur. Bu, robotik sistemlerde özellikle hedeflenen konum ve yönelime ulaşmak için kullanışlı bir yöntemdir [11]. Ters kinematik, son etkisör plakasının hedef konumunu belirleyerek motor açılarını hesaplar ve böylece robotun istenen konuma ulaşmasını sağlar. Bu hesaplamalar, robotun geometrisini temel alır [1]. Üç eklemler plakasının üç boyutlu koordinatları, plakanın yarıçapıyla belirtilen yönde başlar. Robotun geometrisi, dirsek noktasının bir kürenin merkezine yerleştirilmesini ve bu kürenin yarıçapının önkol uzunluğuna eşit olduğunu gerektirir. Bu küre denklemi CGA'da çift küre olarak ifade edilir ve denklem 9'da gösterilir.

$$\sum_i^* X_i - \frac{1}{2} \rho^2 n_\infty \quad (9)$$

CGA'da, her iki işlenen de çift formda olduğundan, iki nesnenin kesişimini belirlemek için meet operatörü kullanılır. Çift dairenin hesaplanması denklem 10'da, nokta-çift ise denklem 11'de beş boyutlu yarı ölçek katsayısının çarpılmasıyla gösterilmektedir ve böylece bir vektör hesaplanır.

$$C_i^* = (B_i - \frac{1}{2} l^2 n_\infty) \wedge (I_3(s_i \wedge e_3)) \quad (10)$$

$$T_i = (C_i^* \wedge \Sigma_i^*) I_5 \quad (11)$$

Yine projeksiyon operatörlerini kullanılmakta ve aşağıdaki denklem 12,13 ile tekil çözümler bulunmaktadır.

$$P_i = \frac{1}{2} (1 + \frac{T_i}{\sqrt{T_i^2}}) \quad (12)$$

$$W_i = -\tilde{P}_i(T_i \cdot n_\infty) P_i \quad (13)$$

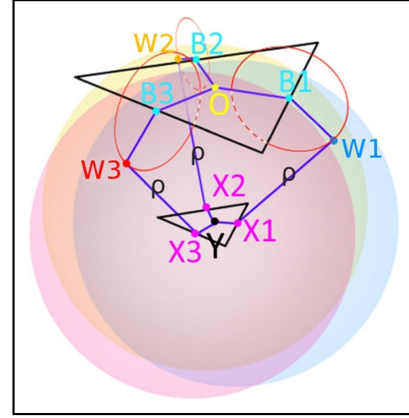
Aşağıdaki denklem 14,15 16 ile CGA'yı üç boyutlu vektör noktaya, ardından motor açısına dönüştürülmektedir.

$$\omega_i = - \frac{\sum_{j=1}^{j=3} (W_i e_j) e_j}{W_i n_\infty} \quad (14)$$

$$\theta_i = \text{atan2}(z_i e_3, z_i s_i) \quad (15)$$

$$z_i = \omega_i - r_b s_i \quad (16)$$

Ters kinematik hesaplama probleminin geometrik temsili Şekil 6'da gösterildiği gibidir.



Şekil 6: Ters kinematik problemi geometrik temsili [1]

### 2.2.3. Ters Jakobiyen

Ters jakobiyen kullanılarak, robotun uç efektörünün çalışma alanında ne kadar hızlı hareket edeceği bilindiğinde, her bir eklemin hangi hızda hareket etmesi gerektiği belirlenebilir. Statik ve kinematik çözümlerini anlamak önemlidir, ancak türevlere bakmak, Delta robot mekanizmasının daha detaylı bir şekilde incelenmesine olanak tanır [1, 11]. Denklem 17'de gösterildiği gibi jakobiyen matrisi oluşturulmaktadır.

$$J^* = \begin{bmatrix} \frac{\partial \theta_1}{\partial \alpha_1} & \frac{\partial \theta_1}{\partial \alpha_2} & \frac{\partial \theta_1}{\partial \alpha_3} \\ \frac{\partial \theta_2}{\partial \alpha_1} & \frac{\partial \theta_2}{\partial \alpha_2} & \frac{\partial \theta_2}{\partial \alpha_3} \\ \frac{\partial \theta_3}{\partial \alpha_1} & \frac{\partial \theta_3}{\partial \alpha_2} & \frac{\partial \theta_3}{\partial \alpha_3} \end{bmatrix} \quad (17)$$

Bu matris kullanılarak, Denklem 18'de gösterildiği gibi, hız vektöründen bir dizi motor hızının uç noktası oluşturulur [1].

$$\begin{bmatrix} \frac{\partial \theta_1}{\partial t} \\ \frac{\partial \theta_2}{\partial t} \\ \frac{\partial \theta_3}{\partial t} \end{bmatrix} = J^* \begin{bmatrix} \frac{\partial \alpha_1}{\partial t} \\ \frac{\partial \alpha_2}{\partial t} \\ \frac{\partial \alpha_3}{\partial t} \end{bmatrix} \quad (18)$$

### 2.2.4. İleri Jakobiyen

Özellikle motor hızlarının uç nokta plakası hızını nasıl etkilediğini belirlenmesi gerekmektedir. [1]. İleri kinematik çözümünde ilk adım, belirli bir kol için dirsek noktasının konumunu hesaplamaktır. Denklem 19'da gelişmiş jakobiyen matrisi ifade edilmektedir [1].

$$J = \begin{bmatrix} \frac{\partial y}{\partial \theta_1} e_1 & \frac{\partial y}{\partial \theta_2} e_1 & \frac{\partial y}{\partial \theta_3} e_1 \\ \frac{\partial y}{\partial \theta_1} e_2 & \frac{\partial y}{\partial \theta_2} e_2 & \frac{\partial y}{\partial \theta_3} e_2 \\ \frac{\partial y}{\partial \theta_1} e_3 & \frac{\partial y}{\partial \theta_2} e_3 & \frac{\partial y}{\partial \theta_3} e_3 \end{bmatrix} \quad (19)$$

Denklem 20’de gösterildiği gibi İleri jakobiyen matrisi ve ters jakobiyen matrisi birbirinin ters matrisi olarak kabul edilir ve bunların matris çarpımı birim matrisi verir.

$$JJ^* = I \quad (20)$$

### 3. Çizim Uygulaması

Bu projenin amacı özet ve giriş bölümlerinde anlatıldığı gibi, kullanıcının fare imleci yardımı ile istedi herhangi bir dijital objeyi çizerek bir Python GUI’sinin oluşturulması ve ve oluşturulan Python kodunu PLC ile iletişim kurarak kullanıcının oluşturduğu şeklin uç efektörlerde gerçekleştirilmesini sağlamaktır.

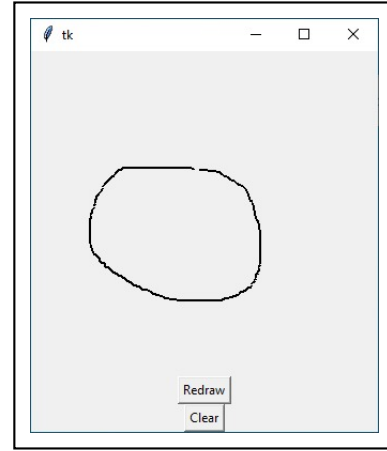
#### 3.1. Python Programlama Dili ve PLC ile İletişim için Kullanımı

Python programlama dili; anlaşılması kolay, basit bir sözdizimine sahip ve geniş bir standart kütüphane içerdiği bilinmektedir. Yapay zeka, makine öğrenimi, görüntü işleme, veri analizi, yazılım geliştirme ve diğer alanlarda sıkça kullanılmaktadır [10]. Python kullanarak GUI oluşturmak, yazılım programlarıyla kullanıcı etkileşimini ve deneyimini büyük ölçüde arttırmaktadır. Örnek olarak, Python’un standart kütüphanesinin bir parçası olan Tkinter kütüphanesi kullanıcı dostu ara yüzü sayesinde hafif ve basit GUI’ler oluşturmak için sıkça kullanılmaktadır. Tkinter kütüphanesi, kullanıcı eylemlerini pencere bileşenleri üzerinde almakta ve işletim sistemi tarafından kontrol edilen olay döngüsünü yönetmektedir. Bu kütüphane, düğme itme, tuş vuruşları, pencere yeniden boyutlandırma ve fare hareketleri gibi kullanıcı eylemlerini kontrol etmektedir [9]. Bu sebeple, GUI geliştirmek için Python öğrenmek, temel ve gelişmiş kullanıcı ara yüzleri geliştirmek için geniş bir seçenek yelpazesi sunmakta ve günümüz yazılım geliştirme ortamında değerli bir beceri haline gelmektedir. Bu proje için, Python’da GUI oluşturmak amacıyla Tkinter kütüphanesi kullanılmaktadır, çünkü standart kütüphanenin içinde mevcuttur ve kolay kodlanabilmektedir. Çizim arayüzü, Python’da Tkinter kütüphanesi kullanılarak oluşturulmuştur.

Python ayrıca PLC programlaması ve haberleşmesi için özel olarak geliştirilen güçlü kütüphaneler sunmaktadır. Böylece geliştirme sürecini önemli ölçüde kolaylaştırmakta ve mühendislerin PLC’lerle ara yüz oluşturma becerilerini geliştirmektedir. Bu proje için, özellikle EtherNet/IP protokolü aracılığıyla Allen-Bradley PLC’lerle iletişim kurmak için oluşturulan pycomm kütüphanesi tercih edilmektedir. Bağlantı kurulumu, PLC etiketi okuma ve yazma, etiket değişiklik izleme ve kontrol komutu iletimi için geniş bir yetenek yelpazesi sunmaktadır. Modül, EtherNet/IP protokolünün karmaşıklıklarını basitleştirerek, Python kullanarak Allen-Bradley PLC’lerle haberleşmeyi kolaylaştırmaktadır. Bu kütüphane, alta yatan karmaşıklıkları soyutlayan ve karmaşık iletişim protokollerini kapsayan yüksek seviyeli işlev sunmakta; böylece PLC’lerle olan haberleşme basitleştirilmektedir. Pycomm3, bu projede ihtiyaç duyulan mantığı ve işlevselliği kolayca uygulanmasına olanak sağlamakta, haberleşme bağlantılarını oluşturmada, veri formatlarını yönetme ve iletişim sorunları ile ilgilenmek gibi görevlerle sahiptir.

#### 3.2. Son Program

Tamamlanan yazılım, PLC ve Python arasındaki etkileşimleri içerir. Kullanıcı tarafından oluşturulan şeklin koordinatları yazılıma kaydedilir ve GUI tualinde yeniden çizim (şekil 7) talimatını vermeden önce kullanıcı, Node-RED paneline en az iki örnek değer girer. Yeniden çiz butonuna basıldığında örnekleme değeri kadar nicelenen şekil yeniden çizilir. Python kodu, kullanıcı tarafından belirlenen örnekleme değerine (şekil 8) bağlı olarak aynı miktarda noktayı aynı mesafede iletir; noktaları birleştirme prensibi budur, yalnızca kod noktaları gönderir. Kullanıcı Node-RED panosundaki bir butona basarak noktayı gönderir ve başka bir butona basarak uç efektörü gönderilen noktaya götürür. Bu işlem manuel olarak nokta değerinin gönderilmesi ve uç efektörün bu değere gönderilmesi şeklinde tekrarlanır. Tüm noktalar gönderildiğinde uç efektör otomatik olarak başlangıç noktasına döner. GUI’deki temizleme düğmesine basıldığında başlangıç koordinatları hareket bloğuna gönderilir. Node-RED panelindeki temizleme butonuna basıldığında uç efektör başlangıç noktasına gider.



Şekil 7: Python’da oluşturulan GUI’de kullanıcı tarafından çizilen şekil

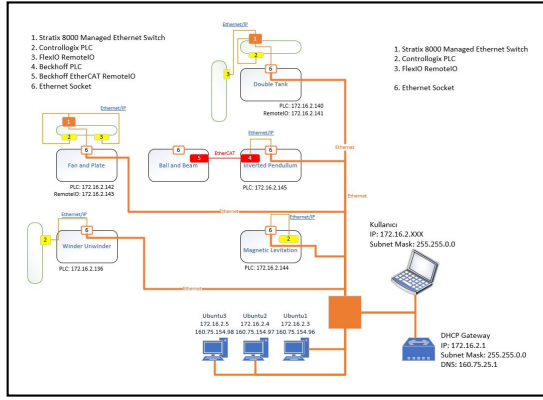


Şekil 8: Noktalar, örnekleme değeri 10 için uç efektör tarafından çizilir



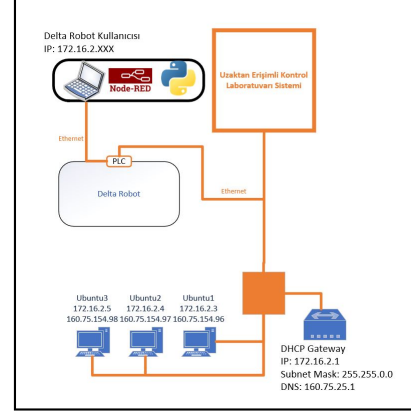
#### 4. Endüstriyel IoT Altyapısı

Delta robotun çizim programı ile uyumlu çalışabilmesi, delta robot yönetiminin internet üzerinden erişilebilir şekilde yapılabilmesi ve çalışma verilerinin kaydedilebilmesi adına bir endüstriyel IoT altyapısı kullanılmıştır. Delta robota ait nesnelere interneti haberleşmeleri bu altyapı aracılığı ile gerçekleştirilmiştir. Delta robotun bulunduğu laboratuvar ortamının sunucu ve haberleşme altyapısı Ayvaz tarafından 2022 yılında yapılan çalışmada açıklanmıştır. Bu çalışmada belirtildiği üzere sunucu ve endüstriyel nesnelere interneti haberleşme altyapısı bir uzaktan erişimli kontrol laboratuvarı sistemi için geliştirilmiş ve kullanılmıştır [12]. Bu çalışmada kullanılan altyapıya Delta Robot sisteminin eklenmiş hali Şekil 9’da görüldüğü gibidir. Şekil 9’da görülen altyapı üzerine Delta robot sistemi Şekil 10’da görüldüğü gibi eklenmiştir.

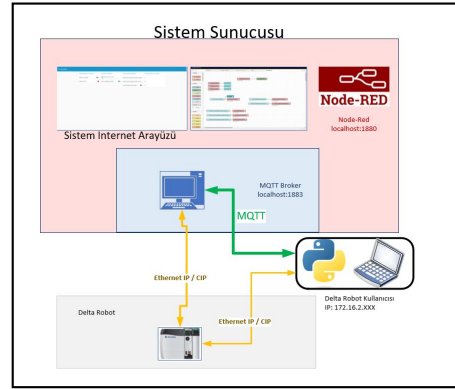


Şekil 9: Delta robotun bulunduğu ortama ait sunucu ve haberleşme altyapısı. [12]

Delta robot sisteminin Şekil 10’da görüldüğü gibi haberleşme ve sunucu altyapısına eklenmesiyle Delta robota ait endüstriyel denetleyiciler, sunucular ile aynı ağ uzayına alınmakta ve makineler arası haberleşmeden faydalanabilmektedir. Çizim yazılımı bulduran Delta robot kullanıcısı bu şekilde çizim yazılımı ve delta robotu aynı ağ almıştır. Kullanıcı aynı zamanda sistem sunucularında bulunan internet tabanlı kullanıcı arayüzüne de ulaşabilmektedir. Şekil 11’de geliştirilen uygulama içi haberleşmeler detaylı olarak gösterilmiştir. Sistem ile çizim yazılımı ve sistem sunucusu arasındaki haberleşmeler Ethernet/IP CIP protokolü ile sağlanmaktadır. Sistem sunucusunda bulunan Node-RED ve çizim yazılımı endüstriyel denetleyici ile Ethernet/IP CIP üzerinden haberleşebilecek özelliktedir. Bunun yanında Node-RED üzerinde bulunan kullanıcı arayüzü ile çizim yazılımı MQTT protokolü aracılığı ile haberleşmektedir.



Şekil 10: Delta robotun sunucu ve haberleşme altyapısına entegrasyonu



Şekil 11: Uygulama içi haberleşmeler.

#### 5. Sonuç

Bu proje; robot denetimi, hareket denetimi ve PLC’lerle iletişim için Python kullanımı ile birlikte IoT’nin uygulanması, Node-RED ve Python programlama dillerinin kullanımıyla başarıyla gerçekleştirilmiştir. Geliştirilen GUI sayesinde, Delta robotuna istenilen çizim yapılabilir veya istenilen yolu takip ettirebilir. Böylece istenilen amaca göre kullanılabilen oldukça esnek bir sistem tasarlanmıştır. Hesaplamalarda kullanılan kinematikler ve jakobiyenler sayesinde Delta robotu güvenli şekilde çalışma alanı içerisinde tekilliklere maruz kalmadan güvenli bir şekilde hareketleri denetlenmiştir. Bu çalışma; daha kolay, daha hızlı ve daha az hatalı yeni bir IoT iletişimi sağlayarak daha da geliştirilebilir.

#### 6. Kaynakça

- [1] Hadfield, H., Wei, L., & Lasenby, J. (2020). The forward and inverse kinematics of a delta robot. In Advances in Computer Graphics: 37th Computer Graphics International Conference, CGI 2020, Geneva, Switzerland, October 20–23, 2020, Proceedings 37 (pp. 447–458). Springer International Publishing.
- [2] Liu, C., Cao, G. H., Qu, Y. Y., & Cheng, Y. M. (2020).

An improved PSO algorithm for time-optimal trajectory planning of Delta robot in intelligent packaging. *The International Journal of Advanced Manufacturing Technology*, 107, 1091-1099.

- [3] "Motion Coordinate System." Retrieved from, Rockwell Automation Publication. , 2022.
- [4] Wen, J. T. & Murphy, S. H. (1990), PID control for robot manipulators.
- [5] Lu, X., & Liu, M. (2015, November). A fuzzy logic controller tuned with pso for delta robot trajectory control. In *IECON 2015-41st Annual Conference of the IEEE Industrial Electronics Society* (pp. 004345-004351). IEEE.
- [6] Lee, C., & An, D. (2022). AI-based posture control algorithm for a 7-DOF robot manipulator. *Machines*, 10(8), 651.
- [7] Engelberger, J. F. (2012). *Robotics in practice: management and applications of industrial robots*. Springer Science & Business Media.
- [8] Fu, K. S., Gonzalez, R. C., Lee, C. G., & Freeman, H. (1987). *Robotics: control, sensing, vision, and intelligence* (Vol. 1). New York: McGraw-Hill.
- [9] Amos, D. (2020). *Python gui programming with tkinter*. Real Python.
- [10] Jazar, R. N. (2010). *Theory of applied robotics*. Springer Science+ Business Media, LLC.
- [11] Craig, J. J. (2005). *Introduction to Robotics*.
- [12] Ayvaz, B., Gültekin, H. E., Ergenç, A. F. Pandemi Sırasında Uzaktan Erişimli Kontrol Laboratuvarı Uygulaması A Remote Access Control Laboratory Application During the Pandemic.