

## 2 SERBESTLİK DERECELİ RR ROBOT KOL İÇİN GERÇEK ZAMANLI BİR MATHEMATİCA SİMÜLASYONU

Samet YAVUZ<sup>1, a)</sup>

<sup>1</sup>Türk Hava Kurumu Üniversitesi, İzmir Havacılık MYO, Motorlu Araçlar ve Ulaştırma Teknolojileri  
Bölümü 35920, İZMİR

<sup>a)</sup>syavuz@thk.edu.tr

Bu çalışmada, iki serbestlik derecesine sahip düzlemsel bir RR robot kolunun ileri ve ters kinematik analizlerinin etkileşimli biçimde gerçekleştirilebildiği bir görselleştirme ve simülasyon aracı geliştirilmiştir. Geliştirilen sistem, Wolfram Mathematica ortamında kullanıcı dostu bir arayüzle tasarlanmış olup, mafsal açıları ve bağlantı uzunlukları üzerinde gerçek zamanlı kontrol olanağı sunmaktadır. Simülasyon ortamı, uç efektörün izlediği yolu dinamik olarak gösteren bir “trajectory trace” modülüne, Jacobian matrisinin hesaplanmasına ve singularite durumlarının analizine olanak sağlamaktadır. Ayrıca kullanıcı, hedef bir konum tanımlayarak sistemin ters kinematik çözümünü görebilmekte ve uç efektörün hedefe yönelmesini izleyebilmektedir. Bunun yanında, sistemde robot kolun belirli açı aralıklarında otomatik hareketini sağlayan bir animasyon modülü de yer almaktadır. Çalışma, robotik sistemlerin hem eğitsel hem de tasarımsal amaçlarla sezgisel olarak anlaşılmasına katkı sağlayacak bir etkileşimli platform sunmaktadır.

**Anahtar Kelimeler:** 2 serbestlik dereceli robot kol, ileri kinematik, ters kinematik, Mathematica, görselleştirme, etkileşimli simülasyon

### A REAL TIME MATHEMATICA SIMULATION FOR A 2-DEGREE-OF-FREEDOM RR ROBOT ARM

#### ABSTRACT

In this study, an interactive visualization and simulation tool is developed for a planar 2-DOF RR robot arm, enabling both forward and inverse kinematic analyses. The system is implemented in Wolfram Mathematica with a user-friendly interface that allows real-time adjustment of joint angles and link lengths. The simulation environment includes a dynamic trajectory trace module that visualizes the path of the end-effector, as well as analytical tools for computing the Jacobian matrix and identifying singular configurations. Additionally, users can define a target position to initiate inverse kinematic computation and observe the robot's motion toward the goal. An animation module is also integrated, enabling the robot arm to move automatically within defined angle ranges. This work offers an intuitive and interactive platform for understanding and analyzing robotic mechanisms, suitable for both educational and design-oriented applications.

**Keywords:** 2-DOF robot arm, forward kinematics, inverse kinematics, Mathematica, visualization, interactive simulation

#### 1. GİRİŞ

Mekanik sistemlerin modellenmesi, simülasyonu ve görselleştirilmesi, mekanizma teorisinin hem eğitim hem de mühendislik uygulamaları açısından temel unsurlarındandır. Özellikle robotik mekanizmalar gibi çok serbestlik dereceli sistemlerde, kinematik

analizlerin yalnızca cebirsel ifadelerle değil, aynı zamanda etkileşimli görsel araçlarla da desteklenmesi, kullanıcıya daha sezgisel bir anlayış kazandırmaktadır.

Bu bağlamda, düzlemsel iki serbestlik dereceli (2-DOF) RR robot kolları, hem teorik

analizlerin uygulanabilirliği hem de görsel olarak kolay anlaşılabilir yapıları sayesinde araştırmalarda ve eğitimde sıkça tercih edilmektedir. Son yıllarda, bu tür robot sistemlerinin farklı yazılım ortamlarında modellenerek simülasyonlarının gerçekleştirilmesi üzerine birçok çalışma yapılmıştır.

Becenen ve Tuna [1], Simulink ortamında geliştirdikleri 2-DOF robot kol modelinde fiziksel ve matematiksel sonuçların uyumunu göstermiştir. Mayetin ve Küçük [2], benzer bir yapı üzerinde farklı kontrol algoritmalarını karşılaştırarak PSO ile optimize ettikleri sistem davranışlarını analiz etmiştir. Diğer bazı çalışmalarda, klasik PID tabanlı kontrol [3] ve Euler-Lagrange yöntemine dayalı modellemeler [4] de başarıyla uygulanmıştır. Kızılhan ve arkadaşları [5] ise SimMechanics ortamında dış iskelet sistemi konum kontrolünü gerçekleştirmiştir.

Eğitsel uygulamalar açısından Ghayoor [6], MATLAB tabanlı sanal bir robotik laboratuvar ortamı geliştirerek, öğrencilerin simülasyon ortamında PID, MPC ve bulanık mantık gibi kontrol yöntemlerini deneyimlemelerini sağlamıştır. Yine yapay zekâ temelli uygulamalarda Sharkawy ve Kahairullah [7], 3 serbestlik dereceli bir robot manipülatörün ileri ve ters kinematik çözümünü yapay sinir ağları ile gerçekleştirmiş; geleneksel çözümlere alternatif yollar sunmuştur. Benzer şekilde, Shah ve arkadaşları [8], 2-DOF düzlemsel bir robotun kinematik çözümlemesini hem analitik hem de yapay zekâ destekli yöntemlerle karşılaştırmış; özellikle eğitim amaçlı uygulamalar için sezgisel ve hızlı çözümler üretmenin önemini vurgulamıştır. Bu tür yaklaşımlar, sistemlerin hem hesaplamalı hem de etkileşimli yönlerini bir arada ele alan hibrit platformların gerekliliğini göstermektedir.

Sabit robot kol sistemlerinin yanı sıra mobil ve hibrit sistemlerin benzetimi üzerine de çeşitli çalışmalar yapılmıştır. Koca ve Yılmaz [9], tank sürüş özellikli bir mobil robotu ROS/Gazebo ortamında simüle ederek, çevresel koşulların hareket performansına etkisini analiz etmiştir. Kavitha ve arkadaşları [10], çok serbestlik dereceli bir robot kolun ROS tabanlı açık kaynaklı bir simülasyon ortamında entegrasyonunu sağlamıştır. Jayasekara ve çalışma arkadaşları ise [11], ROS tabanlı bir

çoklu robot simülasyon ortamı geliştirerek, paralel öğrenme ve görev tabanlı kontrol sistemlerinin nasıl test edilebileceğini ortaya koymuştur. Bu tür platformlar, robot kinematığı üzerine yapılan çalışmalarda simülasyonun yazılım mimarisiyle entegrasyonunu öne çıkarmakta önemli rol oynamaktadır. Mengacci ve arkadaşları [12] ise uyumlu aktüatörlü robotların ROS-Gazebo ortamında gerçekçi şekilde modellenenebilmesi için bir araç kutusu geliştirmiştir.

Takashi [13] tarafından gerçekleştirilen bir çalışmada, Python ile modüler robot yapılandırılmalarının simülasyonu sağlanmış ve çok eksenli yapıların bir araya getirilebileceği deneysel bir platform önerilmiştir. Hibrit sistem örneği olarak Karaağaçlı [14], bir quadrotor sistemine entegre edilen 2-DOF robot kolunun hem uçuş hem de görev kontrolünü içeren bütünlük bir simülasyon modeli sunmuştur.

Bu çalışmalar göstermektedir ki, robotik sistemlerin yalnızca analitik değil, aynı zamanda görsel, etkileşimli ve gerçek zamanlı olarak da modellenmesi, eğitsel ve araştırma odaklı ortamlarda önemli kazanımlar sunmaktadır. Bu çalışmada sunulan ve Wolfram Mathematica ortamında geliştirilen RR robot kolu simülasyonu, etkileşimli arayüzü, uç efektörün izini dinamik olarak çizme özelliği, ters kinematik çözümleyicisi, Jacobian analiz modülü ve otomatik animasyon kabiliyetiyle, hem sezgisel hem de analitik bir deneyim sunmayı hedeflemektedir. Bu çalışmanın literatüre katkısı, 2 serbestlik dereceli RR robot kolları için Wolfram Mathematica ortamında geliştirilmiş, ileri ve ters kinematik analizlerini, Jacobian/tekillik tespitini, çalışma uzayı görselleşmesini ve iz takibi modülünü tek bir gerçek zamanlı, etkileşimli arayüzde birleştiren bütünlük bir simülasyon platformu sunmasıdır. Mevcut çalışmalar çoğunlukla MATLAB/Simulink veya ROS/Gazebo tabanlıdır [1,2,5,6,9]; Mathematica ortamında bu kapsam ve entegrasyonda bir uygulama literatürde yer almamaktadır.

## 2. MATEMATİKSEL MODELLEME

### 2.1. İleri Kinematik

Düzlemsel bir RR robot kolu, iki döner eklemden oluşur ve uç efektörün konumu aşağıdaki denklemlerle hesaplanır.

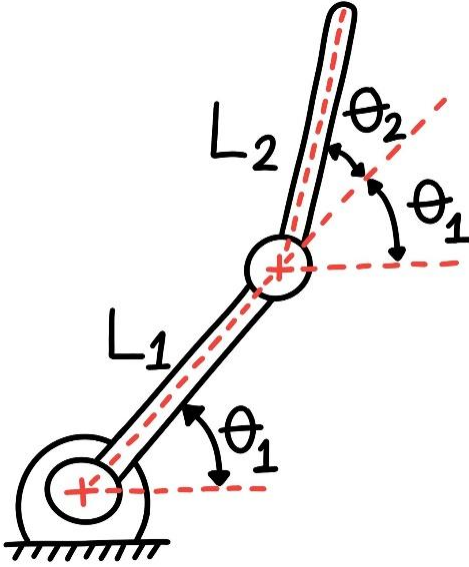
$$x = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) \quad (1)$$

$$y = L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) \quad (2)$$

Burada;

+  $L_1, L_2$  uzuv uzunlukları,

+  $\theta_1, \theta_2$  yere göre eklem açıları olarak verilmiştir (Şekil 1).



Şekil 1. Düzlemsel RR robot kolunun boyutsal parametreleri

### 2.2. Ters Kinematik

Uç efektörün hedef konumu ( $x_d, y_d$ ) verildiğinde, buna karşılık gelen eklem açıları  $\theta_1$  ve  $\theta_2$  aşağıdaki verildiği gibi kapalı formda çözülebilir.

$$\cos(\theta_2) = \frac{x_d^2 + y_d^2 - L_1^2 - L_2^2}{2L_1L_2} \quad (3)$$

$$\theta_2 = \arccos(\cos(\theta_2)) \quad (4)$$

$$\theta_1 = \arctan2(y_d, x_d) - \arctan2\left(\frac{L_2 \sin(\theta_2)}{L_1 + L_2 \cos(\theta_2)}\right) \quad (5)$$

Bu iki çözüm, mekanizmanın aynı hedef konuma farklı geometrik yollarla ulaşmasını sağlar: biri 'dirsek yukarı' (elbow-up), diğeri 'dirsek aşağı' (elbow-down) pozisyonudur. Simülasyonda bu çözüm çiftlerinden yalnızca biri (pozitif sinüs değeri) dikkate alınmıştır.

### 2.3. Jacobian Matrisi ve Tekillik

Bahsi geçen robot kolun hız ilişkilerinden elde edilen Jacobian matrisi aşağıdaki denkleme verilmiştir.

$$J = \begin{bmatrix} -L_1 \sin(\theta_1) - L_2 \sin(\theta_1 + \theta_2) & -L_2 \sin(\theta_1 + \theta_2) \\ L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) & L_2 \cos(\theta_1 + \theta_2) \end{bmatrix} \quad (6)$$

Bu matrisin determinantı, sistemin tekillik durumlarını verir.

$$\det(J) = L_1 L_2 \sin(\theta_2) \quad (7)$$

Eğer  $\sin(\theta_2)$  ifadesi sıfıra eşit ya da sıfıra yakın olursa robot kol tekillik konumunda ya da tekillikğe yakın konumda olur. Bu durumda uç efektörün bazı yönlerde hareket kabiliyeti kaybolur ve kontrol zorluğu artar.

## 3. SİMÜLASYON ARAYÜZÜ

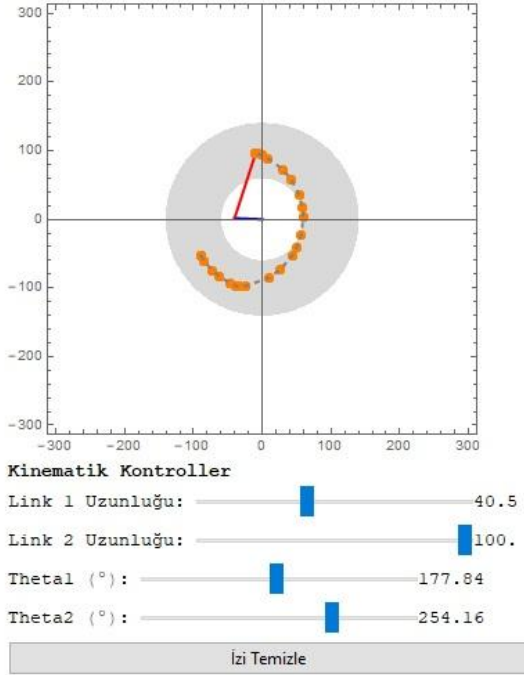
Bu çalışmada geliştirilen simülasyon, Wolfram Mathematica ortamında hazırlanmış etkileşimli bir kullanıcı arayüzünü içermektedir. Uygulama, 2 serbestlik dereceli RR robot kolunun ileri ve ters kinematik çözümlerini görsel olarak sunmakta ve aşağıdaki özellikleri barındırmaktadır:

- Gerçek zamanlı kontrol paneli,
- İz takip özelliği,
- Ters kinematik modülü,
- Jacobian ve tekillik göstergesi,
- Çalışma uzayı modülü

### 3.1. Gerçek Zamanlı Kontrol Paneli

Simülasyon arayüzünün temel bileşenlerinden biri, kullanıcıya uzuv boyutlarını ve eklem açılarını gerçek zamanlı olarak değiştirme imkânı tanıyan etkileşimli kontrol panelidir. Bu panelde dört adet kaydırıcı (slider) yer almaktadır: uzuv boyutları  $L_1$ ,  $L_2$  ve eklem açıları  $\theta_1$ ,  $\theta_2$ . Her bir kaydırıcı, kullanıcının görsel geri bildirim eşliğinde parametreleri kolayca değiştirmesine olanak verir (Şekil 2).

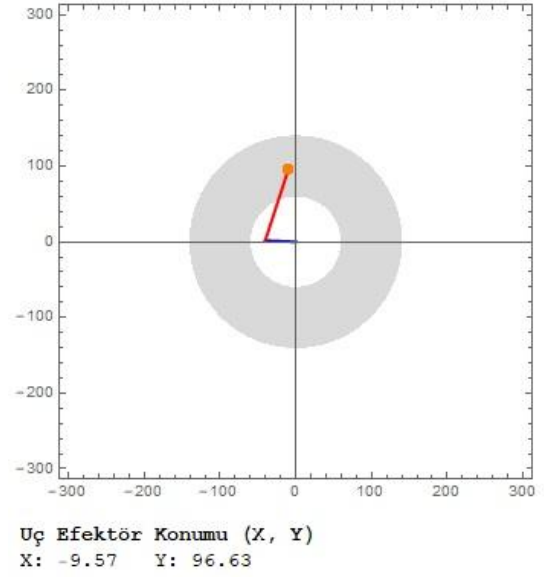
Link boyutları için belirlenen aralıklar 1 ile 100 birim arasında olup, sistemin fiziksel ölçeklendirmeye göre esnetilebilmesine izin verir. Eklemlerin döndürme açıları ise  $0^\circ$ – $360^\circ$  arasında tanımlanmış ve trigonometri hesaplamalarında otomatik olarak radyan dönüşümü yapılacak şekilde düzenlenmiştir.



Şekil 2. Gerçek zamanlı kontrol paneli

Kullanıcı bu parametreleri değiştirdikçe, sistemin ileri kinematik modülü anında yeniden çalışmakta ve robot kolunun yeni durumu, bağlantı çizgileriyle birlikte güncellenmiş olarak grafik alanına yansıtılmaktadır. Aynı anda uç efektör konumu da hesaplanarak sayısal değerlerle metin olarak ekranda sunulmaktadır (Şekil 3). Bu yapı sayesinde kullanıcı; simülasyon ortamında herhangi bir hesaplama komutu girmeksizin parametreleri değiştirerek

robot kolunun davranışlarını gözlemleyebilmektedir.



Şekil 3. Uç efektörün gerçek zamanlı konum bilgisi

### 3.2. İz Takip Özelliği

Simülasyon arayüzünde yer alan Trajectory Trace (İz Takibi) modülü, uç efektörün zaman içerisindeki hareketini kaydederek geçmiş konumlarını grafik üzerinde görsel olarak sunmaktadır (Şekil 1). Bu modül, her yeni ileri kinematik hesaplamasında uç efektörün güncel pozisyonunu bir listeye ekler ve bu pozisyonları simülasyon ortamında renkli noktalar ve kesikli çizgilerle gösterir.

Uygulamada uç efektörün ( $x$ ,  $y$ ) koordinatı her hesaplamada trace adlı bir listeye eklenmekte ve bu liste sabit bir maksimum uzunlukta tutulmaktadır (örneğin 1000 nokta). Bu sınırlama, performans sorunlarını engellemek amacıyla uygulanmıştır. Uç efektörün bu şekilde birikimli olarak gösterilen izi; önceki hareketlerin görsel hafızasını oluşturarak kullanıcıya sistemin dinamik davranışı hakkında bilgi sunmaktadır.

Grafik üzerinde turuncu renkle gösterilen noktalar, uç efektörün geçmişte bulunduğu konumları temsil ederken; bu noktaları birleştiren gri kesikli çizgi ise izlenen yolu bütünsel olarak göstermektedir. Kullanıcı, sistemin önceki hareketlerini gözlemleyerek belirli bir hedefe ulaşım şeklini, sistemin osilasyon eğilimlerini veya sınır davranışlarını

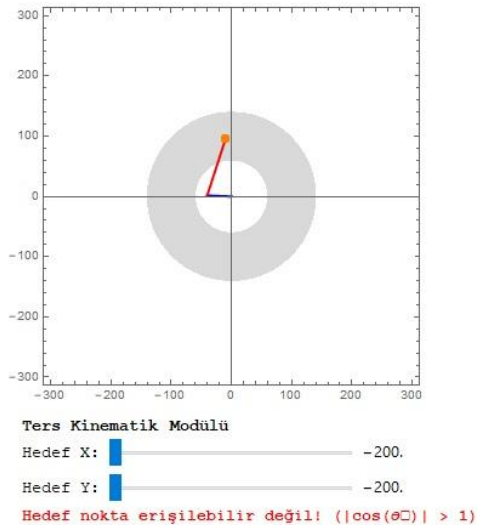
inceleyebilir. Ayrıca, “İzi Temizle” düğmesi aracılığıyla kullanıcı bu izleri manuel olarak sıfırlayarak yeni bir izleme süreci başlatabilir.

### 3.3. Ters Kinematik Modülü

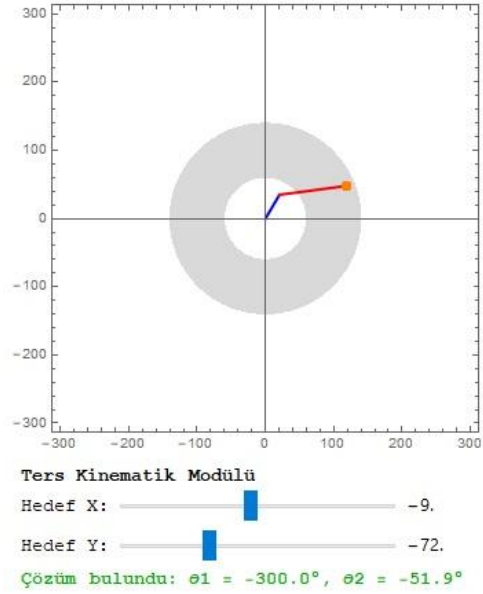
Simülasyon arayüzüne entegre edilen ters kinematik modülü, kullanıcı tarafından belirlenen bir hedef noktanın ( $x_{\text{hedef}}, y_{\text{hedef}}$ ) robot kolunun uç efektörü tarafından erişilip erişilemeyeceğini kontrol etmekte ve erişim mümkünse buna karşılık gelen eklem açılarını ( $\theta_1, \theta_2$ ) hesaplamaktadır. Bu hesaplama, iki uzuvlu düzlemsel RR robot kolu için kapalı formda gerçekleştirilen klasik ters kinematik çözümüne dayanmaktadır.

Ters kinematik çözüm, öncelikle hedef pozisyon ile bağlantı uzunlukları arasındaki geometrik ilişkiye göre  $\cos(\theta_2)$  bileşeninin hesaplanması ile başlar. Hesaplanan değer mutlak değeri 1’den büyükse, fiziksel olarak ulaşılamayan bir hedef seçilmiş demektir ve arayüz kullanıcıyı kırmızı renkli bir uyarı mesajı ile bilgilendirir (Şekil 4). Eğer çözüm mümkünse,  $\theta_2$  açısı arctan fonksiyonu ile hesaplanır ve ardından  $\theta_1$  açısı elde edilir.

Bu hesaplama sonucunda elde edilen açı değerleri simülasyona aktarılır ve robot kolu otomatik olarak bu hedefe yönelir. Ayrıca, kullanıcıya yeşil renkli bir bilgi mesajı ile çözülen açı değerleri gösterilir (Şekil 5). Bu görsel geribildirim sayesinde kullanıcı hem sistemin erişilebilirlik sınırlarını test edebilir hem de ters kinematik çözümlemesinin sonucunu gözlemleyebilir.



Şekil 4. Ters kinematik çözümleme ile ulaşılamayacak konumun belirtilmesi



Şekil 5. Ters kinematik çözümleme ile ulaşılabilecek konumun belirtilmesi

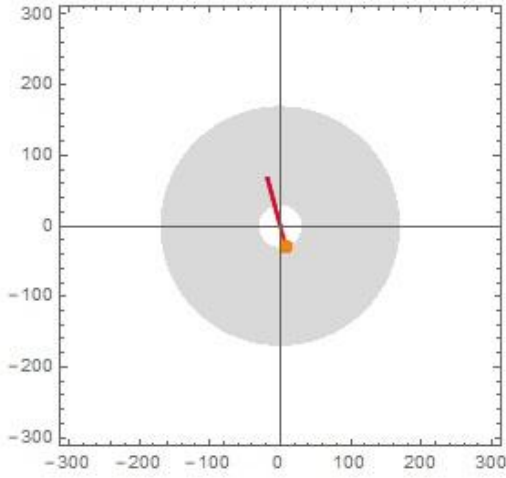
### 3.4. Jacobian ve Tekillik Göstergesi

Simülasyon arayüzünde robot kolunun anlık Jacobian matrisini hesaplayan ve sistemin tekillik durumlarını kullanıcıya anlık olarak bildiren bir gösterge modülü de bulunmaktadır. Bu modülde, ileri kinematik denklemlerden elde edilen pozisyon bilgisi kullanılarak hız kinematiğine geçişi sağlayan  $2 \times 2$  boyutlu Jacobian matrisi dinamik olarak oluşturulur. Bu matris:

$$J = \begin{bmatrix} -L_1 \sin(\theta_1) - L_2 \sin(\theta_1 + \theta_2) & -L_2 \sin(\theta_1 + \theta_2) \\ L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) & L_2 \cos(\theta_1 + \theta_2) \end{bmatrix} \quad (8)$$

şeklinde tanımlanmakta ve her parametre değişiminde yeniden hesaplanmaktadır. Kullanıcı arayüzünde bu matrisin kendisi görsel olarak MatrixForm ile sunulmakta, aynı zamanda determinantı da hesaplanarak doğrudan kullanıcıya bildirilmektedir. Bu determinant değeri belirli bir eşik ( $|\text{Det}(J)| < 0.01$ ) altında ise sistemin tekillik (singularite) bölgesinde çalıştığı uyarısı kırmızı renkli ve kalın bir yazı ile verilmektedir (Şekil 6 ve Şekil 7). Aksi durumda sistemin düzenli (nonsingular) bir bölgede çalıştığı bilgisi yeşil tonlu bir ibareyle gösterilir (Şekil 8). Bu yapı sayesinde kullanıcı, seçtiği parametre kombinasyonunun sistemin kontrol yeteneğini

azaltıp azaltmadığını doğrudan görebilmektedir.



Jacobian ve Singularite Analizi

Jacobian Matrisi

$$\begin{pmatrix} 29.1545 & 96.5382 \\ 7.87741 & 26.0842 \end{pmatrix}$$

$$\text{Det}(J) = 2.401 \times 10^{-12} \rightarrow \text{TEKİLLİK!}$$

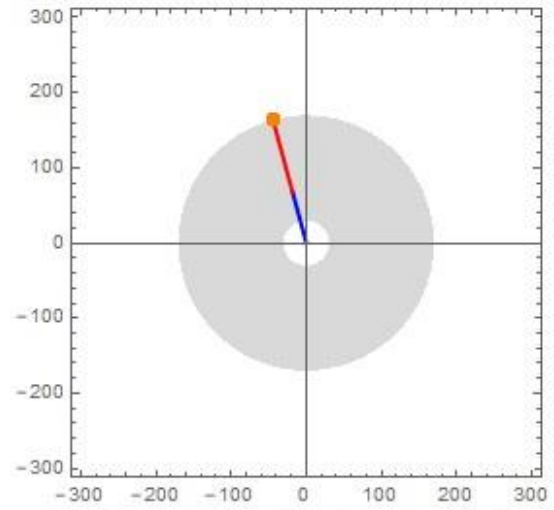
Şekil 6. Simülasyon tekillik tespiti (dirsek aşağı)

### 3.5. Çalışma Uzayı Modülü

Simülasyona ayrıca, robot kolunun ulaşabileceği tüm konumları temsil eden çalışma uzayını görsel olarak sunan bir modül de entegre edilmiştir. Bu modül, L1 ve L2 uzuv boyutlarına göre hesaplanan bir halka bölgeyi (annulus) esas alır. Halka bölge, dış sınırı L1 + L2 yarıçaplı, iç sınırı ise  $|L1 - L2|$  yarıçaplı iki dairenin farkından oluşur. Mathematica'nın RegionPlot komutu kullanılarak elde edilen bu bölge, sistemin erişebileceği tüm uç efektör pozisyonlarını temsil eder. Kullanıcılar, bu görsel rehber sayesinde seçtikleri açı ve bağlantı uzunluklarının fiziksel olarak anlamlı (erişilebilir) olup olmadığını kolayca değerlendirebilmektedir. Ayrıca, bu modülün hesaplama yükünü azaltmak için çizim işlemi manuel olarak tetiklenmekte; "Çalışma Uzayını Yenile" butonuna basıldığında yalnızca o anki parametreler için yeniden oluşturulmaktadır (Şekil 9). Bu sayede hem performans korunmakta hem de sezgisel kullanım desteklenmektedir.

Bu arayüz; mekanizma teorisi, robotik ve kontrol sistemleri gibi disiplinlerde öğrenciler ve araştırmacılar için hem sezgisel hem de

matematikselsel olarak bir analiz ve eğitim ortamı sunmaktadır.



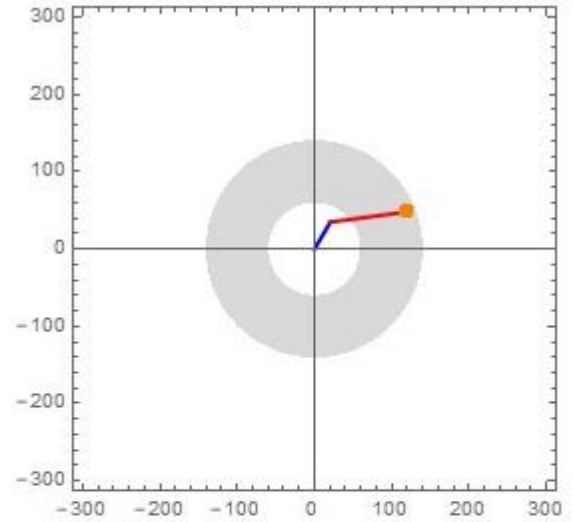
Jacobian ve Singularite Analizi

Jacobian Matrisi

$$\begin{pmatrix} -163.922 & -96.5382 \\ -44.2909 & -26.0842 \end{pmatrix}$$

$$\text{Det}(J) = -2.912 \times 10^{-12} \rightarrow \text{TEKİLLİK!}$$

Şekil 7. Simülasyon tekillik tespiti (dirsek yukarı)



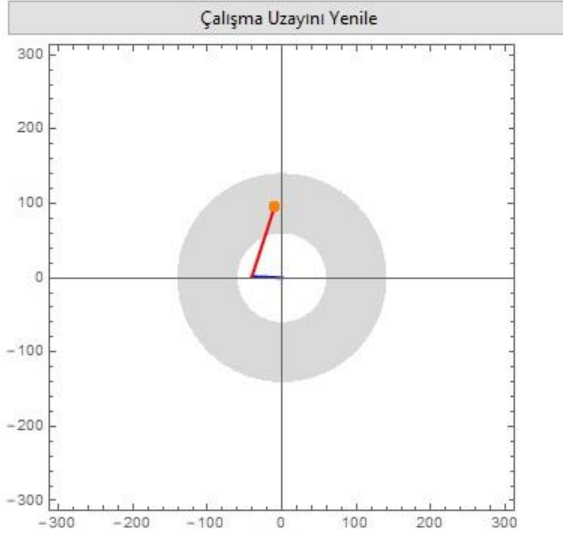
Jacobian ve Singularite Analizi

Jacobian Matrisi

$$\begin{pmatrix} -48.1433 & -13.24 \\ 119.662 & 99.1196 \end{pmatrix}$$

$$\text{Det}(J) = -3188.000 \rightarrow \text{Düzenli}$$

Şekil 8. Simülasyon tekillik tespiti (tekillik yok)



Şekil 9. Simülasyonda çalışma uzayının gösterilmesi

#### 4. SONUÇLAR

Bu çalışmada, iki serbestlik dereceli düzlemsel bir RR robot kolunun ileri ve ters kinematik analizlerini görsel ve etkileşimli biçimde sunan bir simülasyon arayüzü geliştirilmiştir. Wolfram Mathematica ortamında hazırlanan bu sistem, robot kinematığının temel prensiplerini hem nicel hem de görsel yollarla öğrenmeyi kolaylaştırmakta; özellikle mühendislik eğitimi alanında sezgisel bir deneyim sunmaktadır.

Gerçek zamanlı olarak değiştirilebilen parametreler sayesinde robot kolunun çalışma uzayı, pozisyonel davranışı ve tekillik noktaları doğrudan gözlemlenebilir hale gelmiştir. Özellikle aşağıdaki katkılar öne çıkmaktadır:

- İz takip özelliği, uç efektörün zamana bağlı hareketini kaydederek, kullanıcının robotun geçmiş hareketlerini analiz edebilmesini sağlamış; böylece kinematik davranışların daha sezgisel anlaşılmasına katkıda bulunmuştur.
- Ters kinematik modülü, hedef konum tanımlamalarına göre robot konfigürasyonlarını hesaplayarak hem kullanıcı etkileşimini artırmış hem de sistemin işlevselliğini genişletmiştir.
- Jacobian analizi ve tekillik göstergesi, sistemin kritik çalışma bölgelerinin görsel olarak belirlenmesini mümkün kılarak, özellikle mekanizma teorisi derslerinde konunun

somutlaştırılmasına katkı sağlama potansiyeli vardır.

#### 5. TARTIŞMA

Geliştirilen uygulama, robotik, mekanizma teorisi ve mekatronik sistemler gibi mühendislik alanlarında öğretim sürecine doğrudan katkı sağlayabilecek bir araçtır. Sürükle-bırak tabanlı kullanıcı arayüzü ve anlık görsel geri bildirim mekanizması sayesinde lisans düzeyindeki öğrenciler için anlaşılması kolay, etkileşimli bir öğrenim materyali sunmaktadır.

##### 5.1. Gelecek Çalışmalar

Bu sistemin gelecekte aşağıdaki yönlerde geliştirilmesi planlanmaktadır.

- 3 serbestlik dereceli (RRP, RRR vb.) robot yapılarına genişletilmesi,
- Dinamik analizlerin (hız, ivme ve tork) entegrasyonu,
- Zaman tabanlı görev tanımlamaları ve uç efektör görev döngülerinin tanımlanması ve simülasyonu,
- Gömülü sistemlerle bağlantı kurularak fiziksel robot prototipleriyle eş zamanlı çalışma,
- Paralel robotların sentezlerine yönelik yeni ara yüz çalışmaları,
- Seri robotların sadece Denavit-Hatenberg parametreleri verilerek parametrik tanımlamayla otomatik simülasyon oluşturulması.

Bu çalışmanın, mekanizma teorisinin görsel ve interaktif yöntemlerle öğretimini desteklemesi ve robotik sistemlerin simülasyon tabanlı analizleri için bir temel teşkil etmesi amaçlanmaktadır.

#### 6. KAYNAKLAR

[1] Becenen, A., & Tuna, G. (2023). Simulation of Kinematic and Dynamic Model of Two Degrees of Freedom Robot Arm Using Simulink. *Kirklareli University Journal of Engineering and Science*, 9(2), 265-280.

[2] Mayetin, U., & Küçük, S. (2022). Comparative Study of Control Algorithms for 2-Degree of Freedom RR Planar Robot Manipulator Using Partical Swarm

Optimization. *Uluslararası Sivas Bilim ve Teknoloji Üniversitesi Dergisi*, 1(2), 35-62.

[3] Ghaleb, N. M., & Aly, A. A. (2018). Modeling and control of 2-DOF robot arm. *International Journal of Emerging Engineering Research and Technology*, 6(11), 24-31.

[4] Okubanjo, A. A., Oyetola, O. K., Osifeko, M. O., Olaluwoye, O. O., & Alao, P. O. (2017). Modeling of 2-DOF robot arm and control. *Futo J Series (FUTOJNLS)*, 3(2), 80-92.

[5] Kızıllhan, H., Şekerci, B., Kılıç, E., & Başer, Ö. (2022). A simulation study for position control of VSAnkleExo in MATLAB/SimMechanics environment. *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 28(4), 506-515.

[6] Ghayoor, F. (2020). A MATLAB-based virtual robotics laboratory: Demonstrated by a two-wheeled inverted pendulum. *The International Journal of Electrical Engineering & Education*, 57(4), 301-320.

[7] Sharkawy, A. N., & Khairullah, S. S. (2023). Forward and inverse kinematics solution of a 3-DOF articulated robotic manipulator using artificial neural network. *International Journal of Robotics and Control Systems*, 3(2), 330-353.

[8] Shah, J., Rattan, S. S., & Nakra, B. C. (2011). Kinematic Analysis of 2-DOF planer robot using artificial neural network. *International Journal of Mechanical and Mechatronics Engineering*, 5(9), 1720-1723.

[9] Koca, Y. B., Gökçe, B., & Aslan, Y. (2020). ROS/Gazebo Ortamında Tank Sürüş Özellikli Mobil Bir Robotun Simülasyonu. *Journal of Materials and Mechatronics: A*, 1(1), 29-41.

[10] Kavitha, S. (2024). Design and Simulation of a Multi-Degree-of-Freedom Robotic Arm Using Gazebo and ROS, *International Journal of Intelligent Systems and Applications in Engineering*, 12(4), 1588–1597.

[11] Kapukotuwa, J., Lee, B., Devine, D., & Qiao, Y. (2022, August). MultiROS: ROS-based robot simulation environment for concurrent deep reinforcement learning, *In 2022 IEEE 18th International Conference on Automation Science and Engineering (CASE) (pp. 1098-1103)*, IEEE.

[12] Mengacci, R., Zambella, G., Grioli, G., Caporale, D., Catalano, M. G., & Bicchi, A. (2021). An open-source ROS-gazebo toolbox for simulating robots with compliant actuators, *Frontiers in Robotics and AI*, 8, 713083.

[13] Hasbulah, M. H., Jafar, F. A., Nordin, M. H., & Yokota, K. (2019). Experimentation for Modular Robot Simulation by Python Coding to Establish Multiple Configurations, *International Journal of Advanced Computer Science and Applications*, 10(3).

[14] Bulut, N. (2019). *Modeling, simulation, and control of a quadrotor having a 2-DOF robotic arm*, Y. Lisans, ODTÜ.