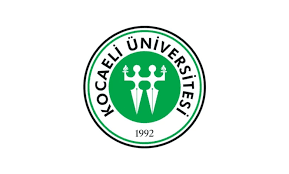
MKT434 Çok Gövdeli Robot Modelleme ve Analizi Dersi Proje Raporu



Ad Soyad: Enes Arıfinan

No: 200223056

E-posta: [enesarifinan@gmail.com](mailto:enesarifinan@gmail.com)

Tel no: 05459480554

Robot Kinematiği ve Dinamiği Simülasyonu Raporu

**1. Giriş**

Bu projede amaç, robot kinematiği ve dinamiği konularında bir simülasyon çalışması gerçekleştirmektir. Projede kullanılan iki farklı MATLAB programı mevcuttur. Birinci program, iki serbestlik dereceli (iki kollu) bir robot kolunun hareketini animasyon şeklinde göstermektedir. İkinci program ise tek kollu bir robot kolunun dinamik analizini yapmaktadır. Birinci kodda robotun kinematik modeli oluşturularak eklem hareketleri görselleştirilmekte, ikinci kodda ise robot koluna etkiyen torklar hesaplanarak zaman eksenine karşı eğrileri çizdirilmektedir.

**2. Kod Açıklamaları**

**2.1. İki Kollu Robot Animasyon Kodu**

İki kollu robot kodu genellikle şu adımlardan oluşur: Başlangıçta MATLAB çalışma ortamı temizlenir, robot parametreleri tanımlanır, simülasyon için zaman vektörü oluşturulur ve döngü içinde robotun pozisyonu hesaplanarak çizilir. Kodun başında yer alan temel komutlar tipik olarak şunlardır:

clc; clear; close all;

Burada clc komutu komut penceresini temizler, clear komutu değişken hafızasını boşaltır, close all ise açık grafik pencerelerini kapatır. Bu şekilde simülasyon her çalıştırıldığında eski verilerden arınmış bir başlangıç yapılır.

Ardından robotun fiziksel özellikleri tanımlanır. Örneğin, birinci kolun ve ikinci kolun uzunlukları veya kütleleri aşağıdaki gibi belirlenebilir:

% Robot bağlantılarının uzunlukları ve ilk konum açıları

L1 = 5.0; % Birinci kol uzunluğu (metre)

L2 = 3.0; % İkinci kol uzunluğu (metre)

theta1 = 0; % Başlangıçta birinci eklem açısı (radyan)

theta2 = 0; % Başlangıçta ikinci eklem açısı (radyan)

Bu kod bloğu, robotun kinematik yapısını tanımlamak için kullanılır. L1 ve L2 bağlantıların uzunluklarını belirtirken, theta1 ve theta2 değişkenleri eklem açılarını ifade eder. Dönüş eksenleri ise genellikle her eklemde z-ekseni etrafında dönen eklemler varsayılarak ayarlanır. Hareket denklemleri kinematik hesaplamaları içerir; örneğin öne doğru kinematik denklemleri ile eklem açılarına göre her bağlantı ucunun uzaydaki koordinatı hesaplanır:

x1 = L1 \* cos(theta1);

y1 = L1 \* sin(theta1);

x2 = x1 + L2 \* cos(theta1 + theta2);

y2 = y1 + L2 \* sin(theta1 + theta2);

Bu hesaplamalarda birinci eklemin ucunun (x1,y1), ikinci eklemin ucunun ise (x2,y2) koordinatları bulunur. Yükselen veya 3B gösterim varsa benzer şekilde z koordinatları hesaplanır.

Bir sonraki aşamada simülasyon için zaman vektörü oluşturulur ve sabit veya zamana bağlı açısal hızlar belirlenir. Örneğin benim kodumda:

T = 2;

dt = 0.01;

t = 0:dt:T;

omega1 = 0.5;

omega2 = -0.3;

Burada t vektörü simülasyon boyunca ilerleyecek zamanı, omega1 ve omega2 ise her eklemin sabit açısal hızlarını temsil eder. Hareket denklemleri kodda bu hızlarla açılar güncellenerek işlenir.

**2.2. Tek Kollu Robot Dinamik Analiz Kodu**

Tek kollu robot dinamik analiz kodu genel olarak robotun kütlesel özellikleri ve dinamik denklemleri üzerinden tork hesaplaması yapar. Kodun başlangıcında benzer şekilde ortam temizlenir ve robot parametreleri tanımlanır:

clc; clear; close all;

% Robot parametreleri (dinamik özellikler)

L = 4.0; % Kolun uzunluğu (metre)

m = 2.0; % Kolun kütlesi (kg)

I = 0.1; % Kolun atalet momenti (kg\*m^2)

g = 9.81; % Yerçekimi ivmesi (m/s^2)

% Hareket profili için başlangıç koşulları

theta = 0; % Başlangıç açısı (radyan)

omega = 0; % Başlangıç açısal hızı (rad/s)

Bu blokta tanımlanan L, m, I gibi değişkenler robotun fiziksel dinamiği için gereklidir. Örneğin, m ve I dönme hareketine direnç (atalet) etkisini belirler. Dinamik analiz genellikle Euler-Lagrange veya Newton-Euler yöntemleriyle yapılır. Örneğin birinci eklem için tork (tau) hesaplaması Lagrange yöntemiyle şu şekilde ifade edilebilir:

Τ(tau) =Iα+mgL​cos(θ) / 2

burada (α) alpha açısal ivmeyi, g ise yerçekimi ivmesini gösterir. Kodda bu dinamik denklem, her bir zaman adımında α ve θ değerlerine göre hesaplanır. Örnek bir kod bloğu tork hesaplaması için şu şekilde olabilir:

for k = 1:length(t)

% Açısal hız ve ivmeyi güncelle (örneğin referans komutuna göre)

omega = omega + alpha \* dt;

theta = theta + omega \* dt;

% Dinamik model: tork hesapla

tau(k) = I \* alpha + m \* g \* (L/2) \* cos(theta);

end

Bu döngüde alpha açısal ivme, omega hız ve theta açı değişkenleri güncellenir; ardından Lagrange denklemi ile anlık tork değeri tau(k) olarak hesaplanır. Hesaplanan tork değerleri daha sonra grafik çizimi için saklanır.

Her iki koddaki clc; clear; gibi başlangıç komutlarının açıklaması ve kod parçaları bu şekilde yapılmıştır. Bağlantı tanımları, dönme eksenleri ve hareket/dinamik denklemler yukarıdaki örneklerde gösterildiği gibi ayrı ayrı ele alınmıştır.

**3. Simülasyon Döngüsü**

Simülasyon döngüsü, her iki kod için de zaman tabanlı hesaplamaların yapıldığı kısımdır. İki kollu robot animasyon kodunda döngü genellikle for veya while yapısıyla aşağıdaki işlemleri gerçekleştirir:

* Açı Güncelleme: Her zaman adımında birinci ve ikinci eklemin açıları önceden belirlenen açısal hızlarla güncellenir. Örneğin theta1 = theta1 + omega1\*dt; ve theta2 = theta2 + omega2\*dt;.
* Kinematik Hesaplama: Güncel açı değerleri ile robotun her bağlantısının uç noktası için uzay koordinatları bulunur. Örneğin iki boyutlu bir düzlemde ilk eklemin ucu (x1,y1) ve ikinci eklemin ucu (x2,y2) formülleriyle hesaplanır.
* Çizim: Elde edilen koordinatlar kullanılarak robot kolları grafik üzerinde çizilir. Her adımda önceki çizimin üzerine yeni çizim yapılacağından hold on komutuyla tutturma sağlanır. Örneğin:
* plot3([0, x1], [0, y1], [0, z1], 'r', 'LineWidth', 2);
* plot3([x1, x2], [y1, y2], [z1, z2], 'b', 'LineWidth', 2);

Bu kodda birinci kol kırmızı ('r'), ikinci kol ise mavi ('b') renkle çizilir.

İkinci kodda (tek kollu dinamik analiz), simülasyon döngüsü genel olarak aşağıdaki adımları içerir:

* Tork Hesaplama: Her zaman adımında, ilgili kinematik ve dinamik denklemler kullanılarak eklem torkları hesaplanır. Bu hesaplamada genellikle güncel açı, hız ve ivme değerleri dikkate alınır.
* Durum Güncelleme: Eğer simülasyon süreklilik içeriyorsa, bu torklar uygulandıktan sonra açılar ve hızlar Euler yöntemi gibi basit entegrasyon teknikleriyle bir sonraki adıma aktarılabilir. Örneğin:
* omega = omega + alpha \* dt;
* theta = theta + omega \* dt;
* Veri Kaydı: Hesaplanan tork değerleri bir dizi içerisinde saklanır. Her döngü sonunda tau(k) gibi bir vektöre yeni değer eklenir; böylece simülasyon bitiminde her bir eklem için tork-zaman verileri elde edilir.

Özetle, simülasyon döngüsünde her zaman adımında gerekli fiziksel hesaplamalar yapılarak robotun hareketi ve dinamik tepkisi sürekli olarak güncellenir ve çizim/çizgi olarak ekrana yansıtılır. Eş zamanlı olarak açılar ve hızlar formül bazlı veya entegrasyonla hesaplanarak bir sonraki adıma geçilir.

**4. Görselleştirme**

Görselleştirme kısmında robotun uzaydaki hareketleri grafiksel olarak gösterilir. MATLAB’da 3 boyutlu çizimler için plot3 komutu sıkça kullanıldı. Bu komut, belirtilen x, y, z koordinatlarındaki noktaları üç boyutlu düzlemde çizer. Örneğin ilk bağlantı şu şekilde çizilebilir:

plot3([0, x1], [0, y1], [0, z1], 'LineWidth', 3);

Bu kodda [0, x1] şeklindeki vektör birinci kolun başlangıç noktasından ucuna kadar x-ekseni koordinatlarını, [0, y1] y-ekseni koordinatlarını, [0, z1] ise z-ekseni koordinatlarını içermektedir. 'LineWidth', 3 parametresi çizginin kalınlığını ayarlar.

Ayrıca view komutu ile görüntünün açısı ve perspektifi ayarlanabilir. Örneğin view(45, 30) komutu ile grafik 45° yatay, 30° dikey bakış açısıyla görüntülenir. axis equal komutu, tüm eksenlerde ölçeği eşit tutar ve robotun gerçek oranlara uygun gösterilmesini sağlar. grid on komutu ise ızgara çizgilerini açarak görselleştirmeyi kolaylaştırır.

Robot bağlantılarının çizimi sırasında, tipik olarak her bağlantı bir çizgi ile temsil edilir ve eklemler arası bağlantı noktaları belirgin hale getirilir. Uç efektör (son kol ucu) noktası bir işaret veya çizgi ile vurgulanabilir. Örneğin, her zaman adımında uç efektörün konumunu işaretlemek için:

plot3(x2, y2, z2, 'ko', 'MarkerSize', 5, 'MarkerFaceColor', 'k');

bu kod siyah bir daire ('ko') ile uç noktasını gösterir. Böylece izleyici robotun hareketini takip edebilir. Görselleştirme adımları, simülasyon döngüsünde güncellenen her koordinat setiyle yeni pozisyonları ekrana çizer ve bu şekilde robotun animasyonu oluşturulur.

**5. Tork Grafikleri (İkinci Kod)**

İkinci kodda yapılan dinamik analizin sonuçları tork-zaman grafikleri olarak gösterilir. Bu kısımda öncelikle hangi verilerin kaydedildiğine değinilmelidir. Simülasyon sırasında genellikle her zaman adımında hesaplanan eklem açısı, açısal hız ve ivme gibi büyüklükler ile eklem torkları bir dizi içerisinde depolanır. Özellikle grafik çizimi için tork değerleri aşağıdaki gibi iki boyutlu vektörler hâline getirilir:

tau1(k) = ...; % Birinci eklem torku

tau2(k) = ...; % İkinci eklem torku

Burada tau1 ve tau2 dizileri, her bir zaman adımı için hesaplanan tork değerlerini sırasıyla birinci ve ikinci eklemde saklar. Simülasyon tamamlandıktan sonra, bu veriler plot komutu ile grafik üzerine aktarılır. Örneğin:

plot(t, tau1, 'r', t, tau2, 'b');

xlabel('Zaman (s)'); ylabel('Tork (Nm)');

legend('Eklem 1', 'Eklem 2');

Bu grafikte kırmızı eğri birinci eklemin, mavi eğri ise ikinci eklemin torkunu zamanla gösterir.

Tork-zaman grafikleri, robotun çalışma esnasında eklem torklarının nasıl değiştiğini yorumlamaya yardımcı olur. Örneğin grafikteki zirveler, yüksek açısal ivme veya yük altında olunan anları işaret eder. Eğer sinüzoidal bir hareket verisi kullanıldıysa, torklar da genellikle sinüzoidal olarak değişir; hareketin dönme ivmesi arttıkça tork değeri yükselir. Ayrıca birinci eklem ikinci ekleme göre genellikle daha fazla yüke maruz kalacağından, tau1 eğrisi tau2’ye kıyasla daha büyük değerlerde olabilir. Tork grafikleri bu nedenlerle eklem başına uygulanan kuvvet ihtiyacını açıkça göstermekte ve tasarım analizinde önemli bilgi sağlamaktadır.

**6. Karşılaşılan Sorunlar**

Kod geliştirme sürecinde bazı eksiklikler veya hatalı çalışan kısımlar ortaya çıkmış olabilir. Bunlar dürüstçe belirtilmeli ve olası nedenleri üzerinde durulmalıdır. Örneğin ikinci kodda gözlenen bazı karışıklıklar şunlardır:

* Sadece 6 eklem sayısına sahip robotlarda çalışması: Her ne kadar da n sayısı programda bulunsa da n 6 dan farklı olduğu anda program çalışmamakta hata vermektedir
* Değişken Tanımlama Eksiklikleri: Kod içinde kullanılan bazı değişkenlerin önceden tanımlanmadığı fark edilmiş olabilir. Örneğin ağırlık, atalet momenti gibi parametreler kodda yer almıyorsa hesaplamalarda hata yaşanır. Çözüm olarak tüm parametrelerin açıkça tanımlandığı bir başlangıç bloğu eklenmelidir.
* Çizim Koşulları: Animasyon sırasında grafik güncellemeleri doğru yapılmamışsa robot parçaları doğru çizilmeyebilir. Örneğin, her döngüde clf komutunun kullanılması veya hold on unutulması çizim bozulmalarına yol açabilir. hold on kullanılarak önceki çizimlerin korunması ve eksen ayarlarının (axis equal) düzgün yapılması tavsiye edilir.
* Zaman Adımı ve Stabilite: Simülasyondaki zaman adımı (dt) çok büyük seçilmişse hesaplanan değerlerde sapmalar oluşabilir. Daha küçük bir dt kullanılarak simülasyonun doğruluğu artırılabilir.

**7. Sonuç**

Sonuç olarak, her iki simülasyon da belirlenen amaçlar doğrultusunda çalışmaktadır. İki kollu robot animasyon kodu, robotun bağlantıların uzaydaki hareketini başarılı bir şekilde grafiksel olarak göstermiştir. Tek kollu robot dinamik analiz kodu ise, robot kolu üzerine etkiyen torkları hesaplayarak zamana bağlı tork profilini elde etmiştir

Dürüstlük ve Şeref Beyanı;

Bu projede sunduğum tüm çalışmalar tarafımdan, kendi emeğimle hazırlanmıştır. Üniversitemin akademik dürüstlük ilkelerine uygun olarak hareket ettiğimi, herhangi bir kaynaktan izinsiz alıntı yapmadığımı, kopya çekmediğimi ve başkasının çalışmasını kendi çalışmam gibi sunmadığımı beyan ederim.

Bu proje kapsamında herhangi bir öğrenciyle fikir alışverişinde bulunduysam, yardım aldıysam veya birlikte belirli kısımlarda çalıştıysam, aşağıda açıkça belirtilmiştir.

İş Birliği

AHMET YİĞİT KARAOĞLU 190224068

- Tek kollu robot dinamiğinde fikir alış verişi yaptım

Ad Soyad: ......................................................

Öğrenci Numarası: ...........................................

Tarih: .............................................................

İmza: ..............................................................