

**TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ**

**ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ**

ELE515

Ödev1

Hande YILDIRIM

141201047

**SORU 1**

Bu soruda bizden 2 adet N-puzzle matrisi (N-yapboz matrisi) arasındaki Manhattan Uzaklığı (Manhattan Distance) değerini hesaplayan bir fonksiyonun yazılması istenmiştir.

N-puzzle matris tanımı şu şekildedir:

* + N-puzzle matrisleri NxN kare matrislerdir, mesela 3x3, 4x4, 10x10, …
  + Elemanları [0, 𝑁^2 − 1] aralığında tamsayılardır.
  + Her eleman birbirinden faklıdır.

Manhattan uzaklığı, iki matrisin birbirinden ne kadar farklı olduğunun bir ölçümüdür.

Kodun başlangıcında Manhattan Uzaklığı hesaplanacak olan matrisin yukarıdaki üç şartı sağlayıp sağlamadığı kontrol edilmiştir. İlk olarak matrisin kare matris olup olmadığını anlayabilmek için A ve B matrislerinin sizeları (boyutları) kontrol edilmiştir. Bunu yaparken “size” komutu kullanılmıştır. Böylece ayrı ayrı A ve B matrislerinin satır ve sütun sayıları bulunmuştur. Bunlardan “nRowA” ve “nRowB” satırları “nColA” ve “nColB” ise sütunları gösteren değişkenlerdir. Burada satır sayıları sütun sayılarına eşit olmazsa matris kare matris olamaz ve “N-puzzle Matris değildir” denir.

A screenshot of a cell phone

Description automatically generated

İkinci olarak, matrisin elemanlarının [0, 𝑁^2 − 1] aralığında tamsayılar olup olmadığı kontrol edilmiştir. Yani buradan şu anlaşılır her matrisin minimum değere sahip elemanı 0, maksimum değere sahip olan elemanı matrisin satır veya sütun sayısının (kare matris olduğu için) karesinin 1 eksiği kadar değere sahip olabilir. Bunun kontrolünü yapabilmek için öncelikle max(A(:)), max(B(:)), min(A(:)), min(B(:)) [5] komutları ile matrislerin maksimum değeri bulunup N^2-1 ve minimum değeri bulunup 0 ile kıyaslanmıştır. Eğer matris bu şartı sağlayamazsa yine N-puzzle Matris olamaz ve fonksiyon error verir.

A screenshot of a cell phone

Description automatically generated

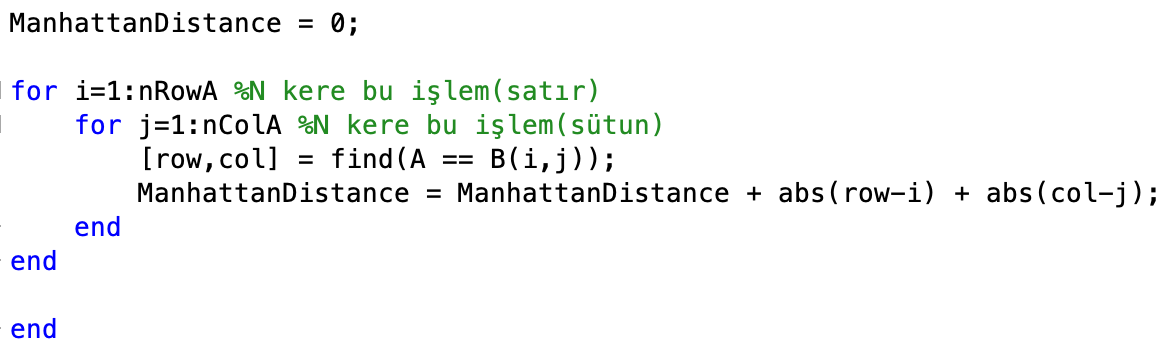
Son olarak ise matrislerin her bir elemanı kendi içlerinde kıyaslanarak aynı değerlere sahip olup olmadıklarına bakılmıştır. Bunu yaparken şu şekilde bir yol izlendi: Öncelikle matrislerin satır ve sütun elemanlarının hepsi “sort(A(:))” ve “sort(B(:))”komutunu kullanarak minimumdan maksimuma sıralandı. Daha sonra bir for döngüsünü “length(siraliA)” komutu kullanarak ve toplam eleman sayısı kadar loopa sokarak her bir eleman kendi içinde kıyaslanmıştır. Yine aynı şekilde bu şartta sağlanamadığı takdirde matris, N-puzzle matris olamaz ve Manhattan Distance hesaplanamaz.

A screenshot of a cell phone

Description automatically generated

Tüm bu controller yapıldıktan sonra en son Manhattan Distance hesaplanmıştır. Bunun için şu formül kullanılmıştır:

Bu formülü biraz daha açıklamak ve kodu da daha net anlamak gerekirse; öncelikle B matrisinde (1,1) pozisyonundaki eleman bulunur. Daha sonra bu elemanın A matrisindeki pozisyonu bulunur. Bulunan pozisyonlardan satır pozisyon değerleri kendi içerisinde, sütun pozisyon değerleri kendi içerisinde çıkarılır ve mutlak değer içerisine alınır ve elde edilen bu 2 değer toplanır. Daha sonra aynı işlem tüm satır ve sütun değerleri için tekrar edilir. En son elde edilen tüm değerler de son kez toplanınca Manhattan Distance bulunmuş olur.



Yukarıda Manhattan Distance kodu açıkça yer almaktadır. Burada ilk olarak Manhattan Distance’a 0 değeri verilir. Daha sonra (i,j) şeklindeki tüm elemanlarda değerler bulunur. Burada kullanılan bazı önemli komutlar şu şekildedir:

find : Bu komut, B’nin i. satır j. sütundaki değerin A’nın hangi posizyonunda(row = hangi satırda bulunduğu, col = hangi sütunda bulunduğu) olduğunu verir.

abs : Bu komut sayesinde mutlak değer alabiliriz.

Böylece program sonucu örneği şu şekilde elde edilir:

A screenshot of a cell phone

Description automatically generated

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**SORU 2**

Bu soruda değişik bir yaklaşımla 𝜋 sayısı, noktadan sonra 2 basamağa kadar tahmin edilmeye çalışıldı. Bu kodu yazarkan “th” değişkeni kullanarak merkezi orijin olan ve 1 yarıçaplı grafik çizdirildi.

A close up of a map

Description automatically generated

Ardından “rand” komutu kullanılarak elemanları [−1,1] aralığında değişen 2 boyutlu 1000 adet rastgele vektör oluşturuldu ve bu vektörlerin her biri için, formundaki vektörlerin uzunlukları formülü yardımıyla bulundu:

A screenshot of a cell phone

Description automatically generated

Bu vektörleri uzunluğu, eğer 𝑟 = 1 ’den büyük olduğu zaman siyah renk ile, küçük olduğu zaman ise kırmızı renk ile çizildi ve bu işlem oluşturulan 1000 adet vektörün hepsi için tek tek gerçekleştirildi. Bu kodu yazarken vektörlerin uzunluklarının 1’den büyük olup olmadıklarını anlamak için “ones” komutu kullanılarak 1000x1’lik (veya verilen toplam nokta sayısı kadar x 1 ‘ lik) bir matris bir matris (r1 matrisi) oluşturuldu. Böylece uzunluk matrisinin her bir elemanını tek tek 1 ile kıyaslayabilme imkanı bulundu. Bu kıyaslamayı yaparken for döngüsü kullanılmasının sebebi ise 1000 kere aynı işlemin yapılacak olunmasıydı. Daha sonra ise “plot” komutu kullanılarak çizilen grafiğin içerisine “scatter”[3] komutu kullanılarak kırmızı (uzunluk vektörünün değeri 1’den küçük olunca) ve dışına yine aynı komut ile siyah renkteki (uzunluk vektörünün değeri 1’den büyük olunca) noktalar çizildi.

A screenshot of a cell phone

Description automatically generated

Ayrıca kodda, bizden istenildiği gibi “waitbar”[2] komutunu yani yükleme çubuğunu kullandık. Bu durumda grafik çizilirken bekleme esnasında aşağıdaki görüntü elde edilmiştir.

A picture containing sky, different

Description automatically generated

En son olarak ise formülünü kullanarak değeri elde edildi. Daha sonra bu değer grafiklere “title” komutu kullanılarak başlık olarak yazıldı.

A screenshot of a cell phone

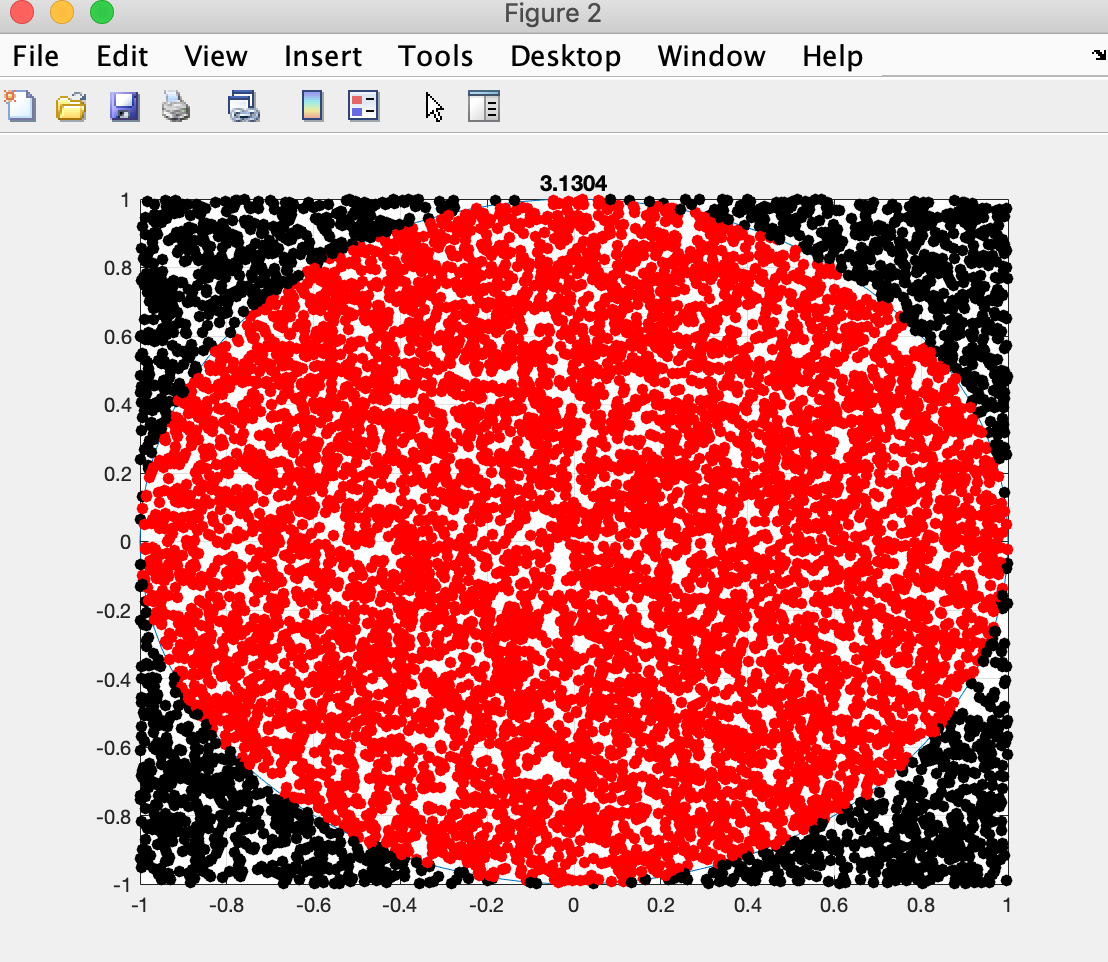
Description automatically generated

Başta döngü çalıştığı zaman yaklaşık olarak 8 dakika içerisinde şu grafiği elde edildi:

A close up of a logo

Description automatically generated

Toplam nokta sayısı artırıldığı zaman ise aşağıdaki grafiği yaklaşık olarak 45 dakika elde edilmiştir.



**SORU 3**

Üçüncü soruda 4 adet görev verilmiştir ve genel olarak aşağıda verilen sistemin simülasyonları yapılacaktır:

A picture containing object

Description automatically generated

Sistem parametrelerinin nominal değerleri ise aşağıdaki tabloda verilmiştir:

A screenshot of a cell phone

Description automatically generated

**GÖREV 1:**

İlk görevde aşağıdaki gibi verilen başlangıç değeri için, 𝑇 = 500 𝑠 için Matlab altında simüle edilmesi istenmiştir. Simulasyon sonucunda elde edilen sistem durumlarını, tek bir figürde 4 ayrı alt şekle “subplot” komutu kullanılarak çizildirilmiştir.

Bu kod yazılırken ilk olarak türev isimli ve çıktıları Kd1, Kd2, M1, M2, Ks ve tut olan bir fonksiyon tanımlandı. Sonra x1(t), x2(t), v1(t) ve v2(t) değişkenleri ilerde veri olarak alıp kullanmak amacıyla “syms” komutu ile sembol olarak tanımlandı. Yukarıdaki sistemde verilen sistem A\*x+B şeklinde bir diferansiyel denklem olduğu çok net bir şekilde anlaşılabilmektedir. Bu bilgiyi kullanarak kod o formatta yazılmıştır. Fakat bizim yazdığım bu denklem aslında türevi alınan vektörlerden oluşan bir sisteme aittir. Bu nedenle diferansiyel denklemi yazarken “diff” komutunu kullanarak türevinin alındığı gösterilmiştir. Yani Matlab’da diferansiyel denklem çözümü yapılmıştır.[1] Ve bu denklem daha sonra “dsolve” komutu kullanılarak başlangıç koşulları yardımıyla çözdürülmüştür.

A screenshot of a cell phone

Description automatically generated

Bu işlem bittikten sonra aynı figüre içerisinde tüm sembollerin grafikleri gözlemlenmiştir. Burada “hold on” komutu sayesinde tüm grafikler kapanmadan aynı anda çizdirilebilmiştir ve “fplot” komutu[1] kullanılarak T=500 saniye için gerekli olan verilere ulaşılmıştır. Burada şunu belirtmek isterim ki hemen aşağıda blunun kod parçasını yazarken çok ciddi bir süre harcadım. Başta sadece real ve vpa kodlarını kullandım fakat buna rağmen kodum çalışmadı. Sonrasında arkadaşımın tavsiyesi ile simplify kodunu ekledim ve kod çalıştı. Neden diye sorarsanız hala bizde anlamış değiliz açıkçası.

A close up of a device

Description automatically generated

A screenshot of a cell phone

Description automatically generated

Bu durumda elde edilen grafikler aşağıdaki gibi olmuştur:

A screenshot of a cell phone

Description automatically generated

**GÖREV 2:**

Bu görevde sistemin farklı parametrelerde nasıl davrandığının araştırılması için bir fonksiyon yazılacaktır.

A screenshot of a cell phone

Description automatically generated

Bu nedenle Perturb işlemi yapabilen bir fonksiyon yazıldı ve bunu yazarken şu formül kullanıldı:

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

**GÖREV 3:**

Bu görevde ise yukarıda yazılmış olan fonksiyon kullanılarak, 𝑀1, 𝑀2,𝐾𝑑1, 𝐾𝑑2, 𝐾𝑠 parametrelerinin her birini, yüzde 10 oranında pertürbe edip, sistem simülasyonunu 5 defa art arda (pertürbe edilmiş bu parametre değerleri için) tekrarlanması (aynı süre ve aynı başlangıç koşulu için) istenmiştir.

Bunu yapabilmek için bu parametrelerin değerlerin kodun en başına aşağıdaki gibi parametre değerleri eklendi. Burada M1 = 300, M2 = 25000, Ks = 1000, Kd1 = 5000, Kd2 = 2000 değerleri bu parametreleri pertürbe edebilmek için 5 kere loopa girmişlerdir.

Burada kodun tamamı ekran görüntüsüne sığmadığı için kod kontrolünü soru3 isimli script dosyasından açarak inceleyebilirsiniz:

A screenshot of a cell phone

Description automatically generated

Bu değerler kullanıldığı zaman elde edilen grafik aşağıdaki şekildedir:

A screenshot of a cell phone

Description automatically generated

**GÖREV 4:**

Burada Matlab Simulink kullanılarak Görev 1 tekrar edilmiştir ve aynı grafik Scope ile elde edilmiştir. Simulinkte çizilmiş olan devreyi aşağıda bulabilirsiniz:

A close up of a map

Description automatically generated

Burada bazı önemli noktalar şu şekildedir:

* En başta “Step Function” kullanılarak 1 adet F değeri için (F=1000) fonksiyon çalıştırılmıştır.
* Üçgen şekiller “Gain” i ifade ederler ve verilen parametreleri ifade etmek amacıyla kullanılan gösterim şekilleri gibi düşünülebilirler.
* Türev işlemi için “Integrator kullanarak” denklemi tersten düşündük ve bu sayede de Geri Beslemeli Bir Sistem tasarlamış olduk.
* Verilen başlangıç değerlerinde sistemi simüle edebilmek için “Constant” kullanıldı ve gerekli yerlere gerekli değerlerde bağlandı.
* Tüm bu işlemleri yapabilmek için öncelikle en başta verilen sistemin denklemi matris çarpımı yapılarak açıldı ve T=500 s’de Simulink yardımı ile modellendi.

En son olarak ise elde edilen grafik aşağıdaki şekilde elde edildi. Bu grafik Scriptte yazılan fonksiyonun çıktısı olan grafiklerle kıyaslandığı zaman, simülasyonun her iki yoldan da başarılı bir şekilde gerçeklenmiş olduğu anlaşılmaktadır.

A close up of a computer

Description automatically generated

**SORU 4**

Bu soruda aşağıdaki iki denklem ile ifade edilen dinamikler verilmişlerdir:

A picture containing object

Description automatically generated

**GÖREV 1:**

Yukarıdaki gibi verilen bir sistem “ode45” fonksiyonu kullanılarak, 𝑇 = 10 𝑠 boyunca, = başlangıç koşulu için Matlab altında (Simulink kullanmadan) simüle edilmiştir. Sistem durumları olan 𝑥1 ve 𝑥2 sinyalleri aynı grafiğe, x-y eksenleri olacak şekilde çizilmişlerdir. Bunların yapımı kod şekline aşağıdaki gibi aktarılmıştır. Bu kod x0 başlangıç koşulu için verilen zamana göre grafiği çizer.

Burada öncelikle “x0” isimli bir vektör tanımlanmıştır. Bu vektör başlangıç değerlerini ifade etmektedir. 10 saniyelik bir simülasyon istendiği için zaman değişkeni 10 olarak alınmıştır. Daha sonrasında öncelikli olarak, ode45 kullanılacağı için fonksiyon f değişkeninin içindeki formdaki gibi yazılmıştır. Bu f değişkeni, t değişkenine bağlı olan x fonksiyonlarını ifade etmektedir.

Verilen denklemde x1 ve x2 olmak üzere iki vektör vardır. Kodda da bu gösterim yapılmıştır. Daha sonrasında fonksiyonu çözdürmek için ode45 kullanılır. Kodun bu kısmında çizdirilen grafiklerde zaman olmadığından yani x1’e bağlı olan x2 çizdirildiğinden zaman değişkeni output kısmına yazılmamıştır. “hold on” komutu ile grafiği tutup, “grid” komutu ile grafiği çizildiği figure’e grid ekleyip, “plot” komutu ile de x2 ile x1’in birbirlerine göre değişimin olduğu grafik çizdirilmiştir. Burada grafiğin başlangıç noktası yine plot komutu ile işaretlenmiştir. Bu işaretleme yapılırken x1(t)’nin 1. elemanı (“res(1,1)”) ve x2(t)’nin 1. elemanı (“res(1,2)”) seçilmiştir ve ona göre kod yazılmıştır. Son olarak ise “xlabel” komutu ile grafiğin x-ekseni, “ylabel” komutu ile de grafiğin y-ekseni isimlendirilip ve “title” komutu ile de grafiğe başlık eklenmiştir. En son “hold off” ile grafiği tutma bırakılmıştır.

A screenshot of a cell phone

Description automatically generated

Yazılan bu kod sonucu aşağıdaki gibi bir grafik elde edilmiştir:

A close up of a map

Description automatically generated

**GÖREV 2:**

Bu bölümde verilen V(x) sistemi için, aşağıdaki şartı sağlayabilen 100 adet şeklinde olan başlangıç koşulu vektörünün elde edilip, 𝑐 parametresi için verilen bazı durumların denenmesi istenmiştir.

A picture containing object

Description automatically generated

Bu durumda öncelikle aşağıdaki her bir durumdaki c değerlerine uygun olarak grafik çizdirilebilecek bir fonksiyon yazılmalıdır.

(Kısaca Not: Bu fonksiyonu yazarken başta kolay geldi fakat sonrasında bilmediğim bazı ufak komutlardan dolayı gerçekten zorlandım. Fakat en son gerekli. Komutları arkadaşlarıma sorup internetten araştırarak buldum ve ne için gerekli olduklarını öğrendim.)

Öncelikle gorev2 isimli fonksiyon verilen c inputuna bağlı olarak başlangıç değerlerini döndürmektedir. Bu başlangıç değerleri kodun devamında kullanılmak üzere “syms” komutu ile sembolik olarak tanımlanmışlardır. Sonra denklemi yazıp çözdürebilmek için sırasıyla başlangıç koşullarını içere x vektörü, bu x vektörünün transpozu ve verilen 2x2 boyutlarındaki matris tanımlanmıştır. Daha sonra denklemin f değişkenine denklemin c’den küçük olma koşulu tanımlanmıştır. Başlangıçta fonksiyon doğrudan elde edilen başlangıç değerlerine göre çizdirilmeye çalışıldı. Fakat sonrasında verilen sistemin maksimumuna ulaşabilmek amacıyla x1 ve x2’nin başlangıç koşulları birbirlerine eşitlendi. Son olarak, fonksiyonun result outputuna ulaşabilmek amacıyla “solve” komutu kullanılarak denklem çözüldü ve başlangıç koşulları elde edildi. Burada yazılan “ ‘ReturnConditions’ ” parametresi aracılığıyla başlangıç değerleri true değerinde elde edildi.

A screenshot of a cell phone

Description automatically generated

Bu fonksiyon yazıldıktan sonra run edildiğinde CommandWindow’da başlangıçta c için bir parametre değeri istendi. Bu da şu komut (“result = gorev2(c)”) verilerek sağlanmış oldu:

A picture containing screenshot

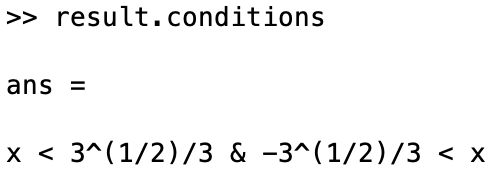
Description automatically generated

Bunun sonucunda CommandWindow’da şöyle bir görüntü elde edildi:

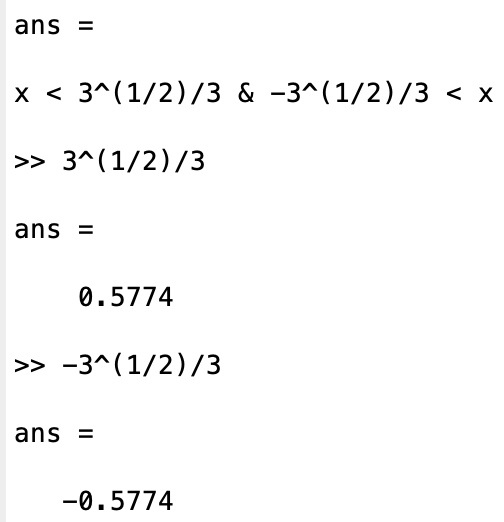
A screenshot of a cell phone

Description automatically generated

Daha sonrasında ise başlangıç değerlerini verilen c değerine göre bulmak için CommandWindow’da “result.conditions” komutu yazılmıştır. Mesela A durumu için:



**A durumu:** 𝑐 = 0.5 değeri için 100 tane başlangıç koşulu elde edilip, 𝑇 = 10 𝑠 boyunca her bir başlangıç koşulunun simüle edilip ve çizilmesi istenmiştir.



Bu değer için yazılan gorev2 fonksiyonu kullanıldığında yukarıdaki gibi (0.5774) ve (-0. 5774) değerleri elde edildi. Bu değerler için grafik çizdirme kodu:

A screenshot of a cell phone

Description automatically generated

Bu kodu kısaca açıklamak gerekirse:

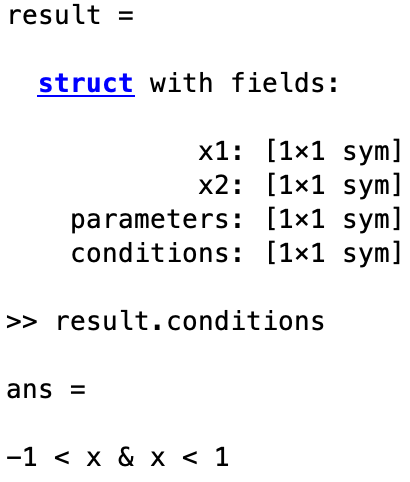
Öncelikle “linspace” komutu kullanılarak eşit aralıklarla değerlerin alınması sağlanmıştır. Daha sonra x vektörü tanımlanmıştır. 100 tane değer alınabilmesi yani 100 tane başlangıç değerinin oluşabilmesi için fonksiyon 100 kere loop’ta döndürülmüştür. Burada önemli bir detay olarak, x vektörünün her seferinde 1. ve 2. Satırları kullanılmıştır daha sonra da grafik çizdirilmiştir. En son olarak grafiği daha estetik olarak görebilmek adına “axis square komutu” kullanılarak grafiğin eksenleri eşit uzunluklara sahip yapılmıştır. Burada özellikle şunu belirtmek isterim ki, bu kodun sadece ama sadece “axis square” [4] kısmını arkadaşımdan öğrenerek yazdım. Ayrıca daha sonraki 3 durumda da aynı algoritma kullanılmıştır.

Buradan aşağıdaki grafik elde edilir:

A picture containing screenshot

Description automatically generated

**B durumu:** 𝑐 = 1.5 değeri için 100 tane başlangıç koşulu elde edilip, 𝑇 = 10 𝑠 boyunca her bir başlangıç koşulunun simüle edilip ve çizilmesi istenmiştir.



Bu değer için yazılan gorev2 fonksiyonu kullanıldığında yukarıdaki gibi (1) ve (-1) değerleri elde edildi. Bu değerler için grafik çizdirme kodu:

A screenshot of a cell phone

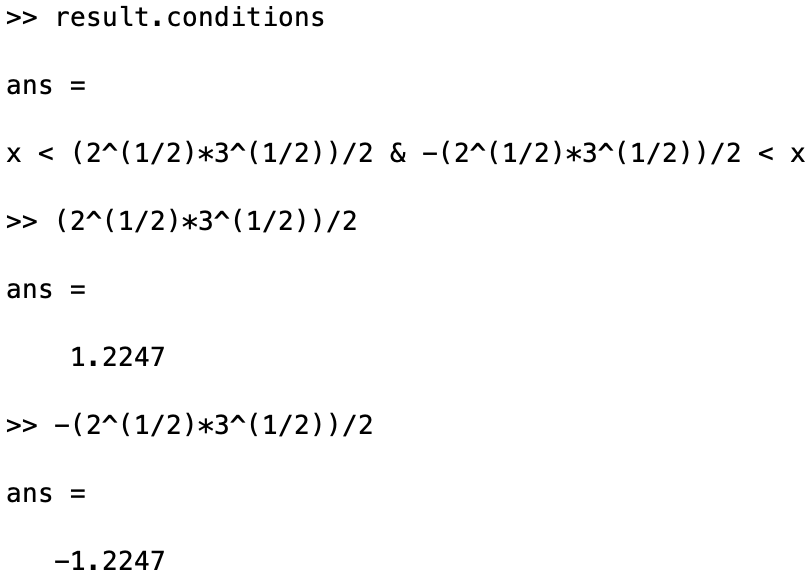
Description automatically generated

Buradan aşağıdaki grafik elde edilir:

A screenshot of a social media post

Description automatically generated

**C durumu:** 𝑐 = 2.25 değeri için 100 tane başlangıç koşulu elde edilip, 𝑇 = 10 𝑠 boyunca her bir başlangıç koşulunun simüle edilip ve çizilmesi istenmiştir.



Bu değer için yazılan gorev2 fonksiyonu kullanıldığında yukarıdaki gibi (1.2247) ve (-1.2247) değerleri elde edildi.

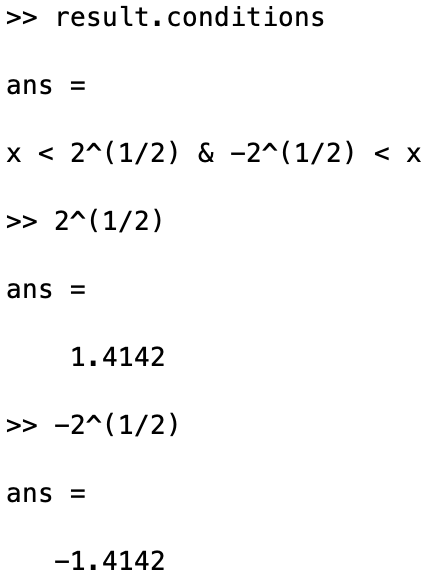
A screenshot of a cell phone

Description automatically generated

**A picture containing screenshot

Description automatically generated**

**D durumu:** 𝑐 = 3.0 değeri için 100 tane başlangıç koşulu elde edilip, ayrıca diğer tüm durumlardan farklı olarak 𝑇 = 2.6 𝑠 boyunca her bir başlangıç koşulunun simüle edilip ve çizilmesi istenmiştir.



Bu değer için yazılan gorev2 fonksiyonu kullanıldığında yukarıdaki gibi (1.4142) ve (-1.4142) değerleri elde edildi.

A screenshot of a cell phone

Description automatically generated

Burada ise diğer durumlardan farklı olarak eksenlerin boyutları “xlim” ve “ylim” komutları kullanılarak belirleştirildi

A screenshot of a social media post

Description automatically generated

**GÖREV 3:**

1. Bu görevdeki ilk kısımda Matlab Simulink kullanılarak Görev-1’deki simülasyonun tekrarlanması istenmiştir. Bu nedenle öncelikle soruda verilen sistemin modellenmesi yapılmıştır. Daha sonrasında verilen başlangıç koşullarına bağlı olarak aşağıdaki simulink devresi çizlmiştir.

Burada bazı önemli noktalar şu şekildedir:

* x1 ve x2 parametreleri birbirlerine bağlı olan iki parametredir. Bir nevi geri beslemeli ve birbirlerine bağlılardır.
* Üçgen şekiller “Gain” i ifade ederler ve verilen parametreleri ifade etmek amacıyla kullanılan gösterim şekilleri gibi düşünülebilirler.
* Türev işlemi için “Integrator kullanarak” denklemi tersten düşündük ve x1 ve x2 parametrelerinin türevleri bu şekilde elde edildi.
* Verilen başlangıç değerlerinde sistemi simüle edebilmek için “Constant” kullanıldı ve gerekli yerlere gerekli değerlerde bağlandı.
* Tüm bu işlemleri yapabilmek için öncelikle en başta verilen sistemin denklemi matris çarpımı yapılarak açıldı ve T=10 s’de Simulink yardımı ile modellendi.

A screenshot of a social media post

Description automatically generated

1. Bu ödevin son kısmında ise sonuçların Matlab Workspace’e aktarılarak simülasyon bitiminde Matlab ortamında çizdirilmesi istenmiştir. Ayrıca buradaki grafiğin detaylı ve Görev-1’deki grafiğin tıpatıp aynısı olması beklenmiştir. Bunu yapmak için öncelikle Simulink ortamında tasarlanmış olan sistem için kısa bir kod yazılmıştır. Bu kod parçası aşağıdaki şekildedir ve kısaca açıklayacak olursak :

Öncelikle ‘Gorev3’ olarak adlandırılan Simulink dosyasından “sim” komutu ile tasarım çekildi. Zaman olarak ise 10s yerine direk simülasyondan zaman değeri alındı. Sonrasında “simout” komutu ile tasarlanan devrenin çıktıları workspace’e getirildiler. Burada 2 kere simout komutu kullanılmıştır. Bunun sebebi sistemin 2 çıkışının olmasıdır. Daha sonra ise plot komutu kullanılarak x1 ve x2’nin dataları çizdirilmiştir. Görev-1’deki ile birebir aynı grafik oluşması için title ve labellar adlandırılıp, grid koyulup, başlangıç değeri işaretlendirilmiştir.

A screenshot of a cell phone

Description automatically generated

Bunun sonucunda elde edilen grafik ise şu şekildedir:

A screenshot of a cell phone

Description automatically generated

**REFERANSLAR**

1. <https://www.mathworks.com/help/symbolic/dsolve.html?s_tid=doc_ta>
2. <https://www.mathworks.com/matlabcentral/answers/446011-i-want-to-add-a-waitbar>
3. <https://www.mathworks.com/matlabcentral/answers/?term=scatter>
4. <https://www.mathworks.com/matlabcentral/answers/131340-what-s-different-between-axis-square-and-axis-equal-in-matlab>
5. <https://www.mekatronikmuhendisligi.com/matlab-temel-islemler.html>
6. Genel olarak her türlü bilgi için dersin kendi sayfası kullanılmıştır.