Elektronik ve Haberleşme Mühendisliği

Haberleşme Laboratuvarı Deney Föyü [1]

Elektronik ve Haberleşme Mühendisliği için MATLAB

Lab. Sorumlusu:
Yasin Yıldırım
Emin Akpınar
Emir Aslandoğan

E-posta:
{ *ysnyldrm, emin.akpinar, emira*} @ *yildiz.edu.tr*

Lab Kuralları :

- Deney föyüne dikkatlice bireysel olarak çalışılması
- Föy içerisinde bulunan konu anlatımdaki kodların yazılması ve yorumlanması
- Araştırılması gereken kod veya konu varsa araştırılması
- Laboratuvara föy haricinde (flash, pc tablet vb.) herhangi bir aracın getirilmemesi
- Bilgisayar düzenine göre yerleşilmesi
- Deney bitiminde bilgisayarların yeniden başlatılması

---

[1] LaTeX *v0.1*

- Laboratuvara her hangi bir yiyecek ve içeceğin getirilmemesi

1 Deneyin Amacı ve Çalışma Soruları

Giriş

Bu deneyin amacı MATLAB'in EHM alanındaki temel işlevlerini öğretilmesi ve etkin olarak kullanılmasıdır. Buna yönelik deney kapsamında temel matematiksel ifadelerin, matris ve vektörlerin nasıl oluşturulduğu işlenecektir. Konu anlatımı içerisinde örnek olarak verilen problem çözümlerinin bireysel kodlanması pratiklik kazanma anlamında faydalı olacaktır. Ayrıca "Öneri" olarak verilen soruların yapılması örnek sorunun kavranmasına veya farklı çeşitlerine karşı çözüm önerisi geliştirilmesine yardımcı olacaktır.

Çalışılması Gereken Konular

- *cos*, *square*, *sawtooth*, *rand*, *randn*, *awgn* gibi temel fonksiyonlar
- Element ve matris çarpımları, indis işlemleri ve çizim yöntemleri
- Uniform ve Normal dağılımın matematiksel gösterimleri ve MATLAB uygulamaları
- "Signal Noise Ratio" (SNR), histogram ve otokorelasyon kavramları

Önemli Bağlantılar ve Kaynaklar

■ `www.ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-094-introduction-to-matlab-january-iap-2010/lecture-notes/`

▲ `www.mccormick.northwestern.edu/documents/students/undergraduate/introduction-to-matlab.pdf`

● `www.eecs.umich.edu/courses/eecs206/public/lab/`

- Kendi emeğinize ve çalışan arkadaşlarınızın emeğine saygı gösteriniz. Cevaplarınızı paylaşmayınız.
- Final sınavında iyi not alabilmek için soruları bireysel cevaplamalısınız.
- Kopyanın en buyuk zararının kendinize olduğunu unutmayınız.
- Cevapları başkalarından 'almak' yerine konuları 'anlamaya' çalışınız.
- Başarılar dileriz...

2 MATLAB’a Başlarken...

MATLAB mühendislik işlemleri için yüksek performanslı bir dildir. Programlama ve görselleştirme ortamlarını bünyesinde oldukça güzel harmanlamaıştır. Bu sayede teknik hesaplama ve çözümlerin yanı sıra ilgilenilen problemin görselleştirilmesi de rahatlıkla yapılabilmektedir. Ayrıca, MATLAB modern programlama dili olarak gelişmiş veri yapılarına sahiptir. Bunlarla birlikte yerleşik düzenleme, hata ayıklama araçları ve nesne yönelimli programlamayıda destekler. Bu faktörler MATLAB’ı öğretim ve araştırma için elverişli bir araç haline getirir. Teknik problemlerin çözümünde MATLAB’nin geleneksel bilgisayar dillerine kıyasla birçok avantajı vardır (ör. C, FORTRAN). Hesaplama ve görselleştirme avantajları sayesinde MATLAB, NASA General Motors, ASELSAN, TAI vb gibi şirketlerde modelleme için en çok tercih edilen programların başında gelir. Özellikle içerisinde barındırdığı Haberleşme, Sinyal İşleme, Görüntü İşleme, Makine Öğrenmesi ve Derin Öğrenme gibi işlev paketleri ile kullanıcıların işini kolaylaştırmaktadır.

YTU-Haberleşme Laboratuvarında Windows 10 tabanlı bilgisayarlarda MATLAB *2013a*, *2015a*, *2017b* veya *2018a* kullanılmaktadır. Yıldız e-postanız ve şifreniz ile okulun sağladığı lisansı kullanarak MATLAB kurulumu yapabilirsiniz. Ayrıntılı bilgi :MATLAB

MATLAB içerisinde barındırdığı yüzlerce fonksiyon sayesinde karmaşık mühendislik hesaplamalarını yapmakta kolaylık sağlar. Fakat bu kadar çok foknsiyonun ezberlenmesi (**Haberleşme için temel fonksiyonları hariç !**) mümkün değildir. Bu nedenle internet üzerinden ilgilenilen konu için MATLAB araştırmasının etkin yapılabilmesi gerekmetkedir. Bu alanda en sık kullanılan fonksiyonlar *help* ve *lookfor*’dur. *help* komutu herhangi bir fonksiyonun nasıl kullanılacağı hakkında detaylı açıklamaları içeren bir döküman sunar.

Bu deney föyü MATLAB programının Haberleşme Laboratuvarı için kullanımına yönelik bir başlangıç niteliğindedir. Bu nedenle aşağıdaki listede verilen MATLAB kavramlarının iyi bir şekilde bilinmesi gerekmektedir. Bu kavramlar föy içerisinde anlatılmamıştır.

- Command Window nedir?
- Command History nedir?
- Workspace nedir? Nasıl kullanılır?
- New Script (Yeni .m dosyası açmak)
- Kodların çalıştırılması
- Yazılan .m dosyalarının kaydedilmesi

3 Temel İşlemler

3.1 Değişken Atamaları ve İndis İşlemleri

MATLAB içerisinde tanımlanan her değişken hafızada matris olarak tutulmaktadır. Köşeli parantez içersinde değişkene atanan değerler aynı sınıf cinsinde olmalıdırlar. `A = [2.1 , [1 2 3] , 'lab' ]` gibi tanımlamalara MATLAB izin vermez. Değişken atamaları Tablo. (1)'de gösterildiği gibi yapılabilir.

Resim (2)'de, *Command Window* üzerinde tanımlanan $\boldsymbol{x} \in \mathbb{R}^{1\times100}$ vektörü normal dağılımlı rasgele sayılar içeren bir satır vektörüdür. $\boldsymbol{y} \in \mathbb{K}^{1\times23}$ ise karakter dizisinden oluşan 23 uzunluklu bir satır vektörüdür. $\boldsymbol{X} \in \mathbb{C}^{20\times100}$ kompleks sayılarda tanımlı matrisi göstermektedir. Dikkat edilirse MATLAB harfe duyarlı (*case-sensitive*) bir dil olduğu için aynı harfe ait büyük ve küçük ile tanımlanan değişkenlere farklı atamalar yapılabilir. Değişkenlere atamalar ile ilgili örnekler Table (1)'de verilmiştir.

```
>> whos x y X
  Name        Size             Bytes  Class     Attributes

  X          20x100            32000  double    complex
  x           1x100              800  double
  y           1x23                46  char
```

Figure 1: MATLAB üzerinde tanımlanan bazı değişkenler ve özellikleri. MATLAB harfe duyarlı bir dil olduğu için x ve X değişkenleri farklı büyüklükleri tanımlamak için kullanılabilir.

Notasyon Bilgisi

Föyde reel sayılardan oluşan matrisler $\mathbb{R}$, kompleks değerli matrisler $\mathbb{C}$, ve karakter içeren matrisler ise $\mathbb{K}$ kümeleri ile gösterilecektir. Örnek olarak `x = [-1.2 -2.3 0.12 4.11]` satır vektörü $\boldsymbol{x} \in \mathbb{R}^{1\times4}$; `y = [0.1; 0.2; 0.3; 0.4]` olarak tanımlanan sütun vektörü için ise $\boldsymbol{x} \in \mathbb{R}^{4\times1}$ notasyonu kullanılacaktır. `A = [1 2 3; 4 5 6]` matrisi $\boldsymbol{A} \in \mathbb{R}^{2\times3}$; `c = 2.1 + 1i*5.1` skaler sayısı $c \in \mathbb{C}$ olarak ifade edilecektir.

Tanımlanan matrislerde indis işlemleri (.) parantezleri ile olmaktadır ve indis değerleri mutlaka 0 içermeyen doğal sayılardan oluşmalıdır. Örnek olarak $\boldsymbol{A} \in \mathbb{R}^{50\times50}$ büyüklüğünde bir matris olsun. $\boldsymbol{A}$ matrisinin satır : 25, sütun : 42 değerine `A(25,42)` ile ulaşılabilir. Eğer 25. satırın tamamı isteniyorsa `A(25,:)` yazılarak ilgili satırdaki tüm sütun değerleri çekilebilir. Aynı işlem `A(:,25)` olarak yazılırsa 25. sütundaki tüm değerler alınır. Burada dikkat edilmesi gereken önemli yer elde edilen vektörlerin boyutlarıdır. `x = A(25,:)` $\boldsymbol{x} \in \mathbb{R}^{1\times50}$ ile; `x = A(:,25)` ise $\boldsymbol{x} \in \mathbb{R}^{50\times1}$ ile ifade edilir. Matematiksel işlemlerin yanlış olmaması veya MATLAB hatalarının alınmaması için matris boyutlarına dikkat edilmelidir.

Tablo 1: SKALER, VEKTÖREL VE MATRIS TANIMLAMALARI, İNDIS İŞLEMLERI
MATLAB üzerinde her değişken matris olarak tanımlanmaktadır. Skaler değerler 1×1 matris olarak nitelendirilir. Sayıların türleri belirtilmediği sürece otomatik olarak *double* alınır

MATLAB Kodu	**Matematiksel Gösterim**	**Küme Tanımı**
`a = 0.1; veya a = .1;`	$a = 0.1$	$\mathbb{R}^{1\times1}$ veya $\mathbb{R}$
`a = 3.2 + 1i*5.1`	$a = 3.2 + j5.1$	$\mathbb{C}^{1\times1}$ veya $\mathbb{C}$
`x = exp(-1i*20)`	$x = e^{-j20}$	$\mathbb{C}$
`v = [1+1i*2; 0.3; 0.1-1i*5]`	$\boldsymbol{v} = \begin{bmatrix} 1+j2 \\ 0.3 \\ 0.1-j5 \end{bmatrix}$	$\mathbb{C}^{3\times1}$
`x = exp(-1i*2*pi/32)`	$x = e^{-j\frac{2\pi}{32}}$	$\mathbb{C}$
`x = []`	$x =$ Boş Küme Matrisi	$\mathbb{R}^{0\times1}$
A = [1 2 3; 4 5 6]	$\boldsymbol{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$	$\mathbb{R}^{2\times3}$
$\boldsymbol{x}$ = A(:,1)	A'nın 1. sütunu	
$\boldsymbol{x}$ = A(1,:)	A'nın 1. satırı	

`D = [1 2 3;`
`4 5 6;`
`7 8 9;]`
	3x3'lük matris	

Notasyon Bilgisi

İndis kullanımında `end` değerinin özel bir yeri vardır. MATLAB `end` kullanılan matrisin son değerini alır. Bu özel değer sayeinde matris boyutları bilinmese bile sondaki değerlere `end` veya `end-1`, `end-2` gibi indisler kullanılarak ulaşılabilir. Örnek olarak `D = [5 11 23;25 45 74; 23 66 98]` $\in \mathbb{R}^{3\times3}$ bir matris olsun. `x = D(end,:)` ile `x = [23 66 98];` `x = D(end,2)` ile `x = 66` ve `x = D(:,end-1)` ile `x = [11; 45; 66]` değerlerine ulaşılır.

```
>> D = [5 11 23;25 45 74; 23 66 98]

D =

     5    11    23
    25    45    74
    23    66    98

>> D(end,:)

ans =

    23    66    98

>> D(end,2)

ans =

    66

>> D(:,end-1)

ans =

    11
    45
    66
```

Figure 2: MATLAB üzerinde tanımlanan bazı değişkenler ve özellikleri. MATLAB harfe duyarlı bir dil olduğu için x ve X değişkenleri farklı büyüklükleri tanımlamak için kullanılabilir.

3.2 Matris Atamaları ve İşlemleri

Matris Atamaları

```
D = [1 2 3;
     4 5 6;
     7 8 9;]

D_transpose = D' % D'nin
    transpozu
```

`D(:,end)` => $[3,6,9]^T$
`D(end,:)` => [7,8,9]

D^T :
$1,4,7$
$2,5,8$
$3,6,9$

Tablo 2: Matrisler ile kullanılan operatörler

Operatör	Amaç	Açıklama
+	Toplama	-
-	Çıkarma	-
/	Bölme	Matrisler aynı boyda olmalı
*	Matris çarpımı	Matrisler mxn ve nxk boyutlu olmalı
.*	Nokta Çarpım	Matrisler aynı boyda olmalı
.^	Tüm elemanların n. Kuvvetini alma	-
'	Tranpozu alma	-
.'	Transpozu alma*	Komplex sayılarda konjugesini almaz

Operatörlerin kullanımı

```
A = [1 2 3;
     4 5 6;
     7 8 9];
B= [3 2 1;
    6 5 4;
    9 8 7];
toplam = A+B;
fark = A-B;
matris_carpimi = A*b;
nokta_carpim = A.*B;
A_transpozu= A';
A_karesi = A.^2
```

Tablo 3: Matrisler atamalarında kullanılan bazı fonksiyonalr

Fonksiyon	Açıklama
randn(n,m)	mxn boyutlu normal dağılımlı [0,1] rastgele bir matris oluşturur
rand(n,m)	mxn boyutlu uniform [0,1] dağılımlı rastgele bir matris oluşturur
ones(m,n)	mxn boyutlu 1 değerlerinden oluşan bir matris oluşturur
zeros(m,n)	mxn boyutlu 0 değerlerinden oluşan bir matris oluşturur
eye(n)	nxn boyutlu birim matris oluşturur
inv(A)	Matrisin tersini alır (Eğer varsa)
norm(A)	Matrisin normunu verir
size(A)	Matrisin boyutunu verir
det(A)	Matrisin determinantını verir
max(A)	Matrisin maksimum değerli elemanını verir
min(A)	Matrisin minimum değerli elemanını verir
sum(A)	Matrisin değerlerinin toplamını verir
mean(A)	Matrisin değerlerinin ortalamasını verir
diag(A)	Matrisin diyagonal elemanlarını verir

3.3 Matematiksel İfadelerin MATLAB'da oluşturulması

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

```
%Ornek: n = 10
% n, 1'den 10'a kadar sayilari iceren bir vektor.
n= 1:10;
sum(n)
% Yada formul kullanilarak yapilabilir
n=10;
n*(n+1)/2
```

$$\sum_{i=1}^{10} \frac{i^2+i}{i^3+sin(i)}$$

```
x = 0;
for i = 1:10
    x = (i^2+i)/(i^3+sin(i))+x;
end
%veya
i = 1:10;
x = sum( (i.^2 + i)./...
         (i.^3 + sin(i)) )
```

$$y = \sin 2\pi 100t + \cos 2\pi 200t$$

```
% Ts: Ornekleme periyodu
Ts = 1/1000; t  = 0:Ts:1;
F0 = 100;    F1 = 200;
y  = sin(2*pi*F0*t)+...
     cos(2*pi*F1*t);
```

$$\Gamma(\alpha) = \int_{x=0}^{\infty} x^{\alpha}.e^{-x}.dx$$

Not:Bu integral $\alpha \in \Re$ değerinin faktoriyel hesabını yapan özel bir integraldir. Sayılar teorisinde ve olasılık hesaplarında sıklıkla kullanılır.

```
alfa = pi;
xmin = 0; % alt sinir
xmax = alfa*50; % ust sinir
fun = @(x) x.^alfa.*exp(-x);
val = integral(fun,xmin,xmax)
```

$$\begin{array}{ll} \text{alfa} = 3 & \text{val} = 6.0000 \\ \text{alfa} = \pi & \text{val} = 7.1881 \\ \text{alfa} = e & \text{val} = 4.2608 \end{array}$$

Kullanım İpuçları

MATLAB'te "..." ifadesi uzun olan eşitlik tanımlarını bir alt satıra kaydırmaya yarar. "f = @(x)" ifadesi ise "handle" fonksiyon olarak geçer. Kullanım : f = @(x) (x.^2+1); olarak tanımlansın. f([3, 4])'in sonucu [10, 17]

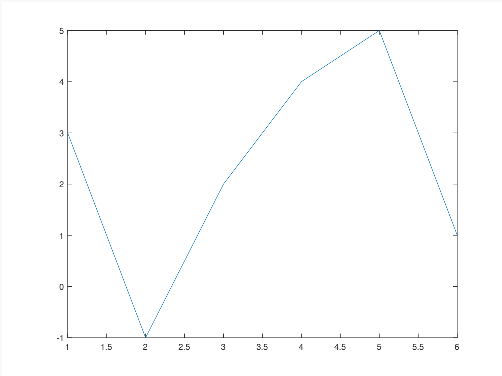
4 Hesaplamaların Görselleştirilmesi

MATLAB iyi grafik araçlarına sahiptir. Belirli bir veri kümesini veya hesaplama sonuçlarını çizdirmek çok az komutla mümkündür. Matematiksel denklemleri grafiklerle anlamaya çalışmak, MATLAB'ı öğrenmenin keyifli ve etkili bir yoludur. Matematiksel fonksiyonları ve verileri kolayca görselleştirebilmek bu bölümün amaçlarındandır.

4.1 Basit Grafikler Oluşturma

MATLAB'da 2-boyutlu fonksiyonları çizdirmek için `plot` komutu kullanılır. X-Y koordinatında bulunan bir işaretin çizdirilmesi için aynı uzunluklu x ve y vektörü gerekir. Örnek olarak $x = [1, 2, 3, 4, 5, 6]$ ve $y = [3, -1, 2, 4, 5, 1]$ vektörlerini çizdirmek için `plot(x,y)` komutu kullanılmalıdır.

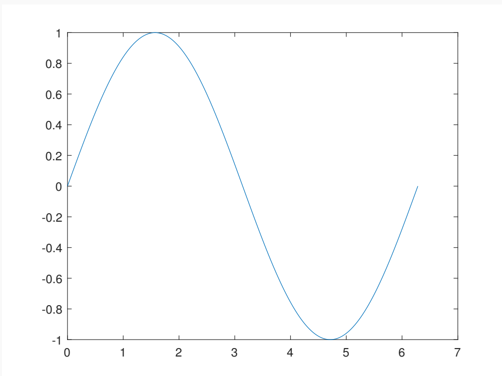
```
x = [1 2 3 4 5 6];
y = [3 -1 2 4 5 1];
plot(x,y)
```



İkinci örnek; bir sinüsoidal işaretin bir periyot boyunca çizdirilmesi olsun. Bir sinüs işareti oluşturmak için `sin(x)` fonksiyonu kullanılmalıdır ve `x` vektörü, sinüsün bir periyodu olarak seçilmelidir ([0,2π]).

NOT: Aşağıda, "0 :pi/100:2*pi" komutu 0'dan başlayan ve 2π'ye $\pi/100$ aralıklarla giden bir vektörü tanımlar.

```
x = 0:pi/100:2*pi;
y=sin(x);
plot(x,y)
```

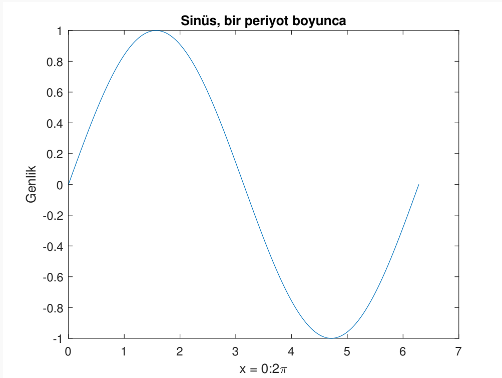


4.2 Grafiklere Başlık ve Eksen Ekleme

MATLAB ile eksenlere etiket ekleyebilir, grafiklere başlık koyabilirsiniz. Böylece grafikleriniz daha anlaşılır olur. Bir önceki örnekte, bir periyot boyunca çizdirilen sinüs işaretinin eksenlerine isim vermek için aşağıdaki komutlar kullanılır.

NOT: `xlabel` x eksenine isim verir, `ylabel` y eksenine isim verir.

```
x = 0:pi/100:2*pi;
y=sin(x);
plot(x,y)
xlabel('x = 0:2\pi')
ylabel('Genlik')
title('Sinus isareti, bir periyot
    boyunca')
```



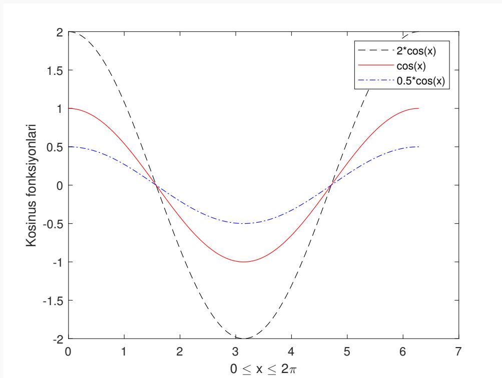
Grafiklerde varsayılan renk mavidir, ancak diğer renkleri kullanmakta mümkündür. İstenilen renk üçüncü argüman olarak eklenir.(Örnek: Kırmızı için `plot(x,y,'r')`)

4.3 Çoklu Grafik Çizdirme

Simulasyon yapılırken birden çok fonksiyonun aynı grafikte çizdirilmesi gerekebilir. Bu gibi durumlarda fonksiyonlar `plot(x,y1,x,y2,x,y3,.....)` şeklinde aynı figür üzerinde gösterilebilir. Renk ve eğrinin stili, `plot(x,y1,'r-',x,y2,'k-.')` şeklinde belirlenebilir.

```
x  = 0:pi/100:2*pi;
y1 = 2*cos(x); y2 = cos(x);
y3 = 0.5*cos(x);

plot( x,y1,'k—',...
      x,y2,'r–' ,...
      x,y3,'b–.')
xlabel('0 \leq x \leq 2\pi')
ylabel('Cosine functions')
legend('2*cos(x)','cos(x)','0.5*
    cos(x)')
```



Fonksiyonları tek bir grafikte alt alta (veya yan yana) çizdirmek için `subplot` komutu kullanılır. Bu komut ile mxn'lik bir ızgara oluşturulur ve ızgaralar sırayla seçilerek grafikler çizdirilir.

Tablo 4: Plot komutu için ayarlar

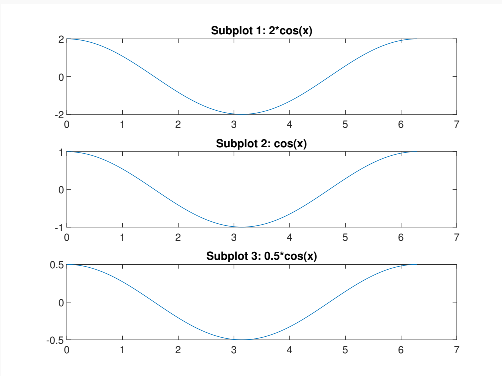
Sembol	**Renk**	**Sembol**	**Stil**	**Sembol**	**İşaret**
k	Siyah			+	Artı
r	Kırmızı	-	Düz	o	Çember
b	Mavi	- -	Parçalı	*	Asteriks
g	Yeşil	:	Noktalı	.	Nokta
m	Eflatun	-.	Parçalı-Noktalı	X	Çarpı
y	Sarı			s	Kare

```
x= 0:pi/100:2*pi;
y1=2*cos(x);
y2=cos(x);
y3=0.5*cos(x);

subplot(3,1,1)
plot(x,y1)
title('Subplot 1: 2*cos(x)')

subplot(3,1,2)
plot(x,y2)
title('Subplot 2: cos(x)')

subplot(3,1,3)
plot(x,y3)
title('Subplot 3: 0.5*cos(x)')
```



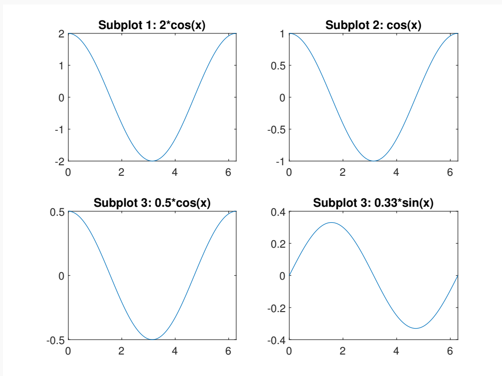

```
x= 0:pi/100:2*pi;
y1=2*cos(x);
y2=cos(x);
y3=0.5*cos(x);
y4=0.33*sin(x);

subplot(2,2,1)
plot(x,y1)
title('Subplot 1: 2*cos(x)')

subplot(2,2,2)
plot(x,y2)
title('Subplot 2: cos(x)')

subplot(2,2,3)
plot(x,y3)
title('Subplot 3: 0.5*cos(x)')

subplot(2,2,4)
plot(x,y4)
title('Subplot 3: 0.33*sin(x)')
```



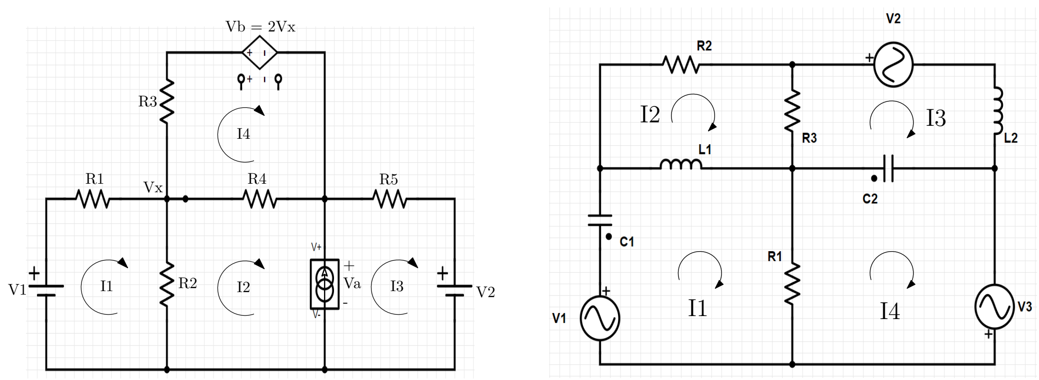
5 Mühendislik Hesaplamaları

MATLAB mühendislik hesaplamalarında ve simülasyon ortamlarının geliştirilmesinde kullanılan önemli bir programdır. Elektronik ve Haberleşme Mühendisliği kapsmında şirketler, tasarladıkları sistemlerin teori ile tutarlı olup olmadığını görmek için mutlaka sistemlerin simülasyonunu oluştururlar. Bu bölümde mühendislik alanında basit problemlerin çözümünü MATLAB programı yardımı ile yapılacaktır.

5.1 Devre Analizi

Devre analizinde göz-akım veya düğüm-gerilim yöntemleri sistemlerin çözümü için kullanılmaktadır. Önemli olan dervreleri, bilinmeyen sayısı kadar denklem üreterek matrissel bir formda yazılmasıdır. Bu sayede MATLAB veya benzer programlar ile doğrusal sistemler çözülerek devreye ait bilinmeyenler kolaylıkla bulunabilir. Örnek-1'de bulunan devreler el ile çözümü zor ve karmaşık olan sistemlerdir. Fakat devreleri göz-akım metodu ile matris formuna dönüştürülebilirse MATLAB'ı kullanarak hızlıca çözülecektir.

Örnek: 1 Devre Çözümlemesi



(a) (b)

$$
\begin{aligned}
V1 &= I1\times[R1+R2] + I2\times[-R2] \quad + I3\times[0] \quad + I4\times[0] \\
-Va &= I1\times[-R2] \quad + I2\times[R2+R4] + I3\times[0] \quad + I4\times[-R4] \\
Va - V2 &= I1\times[0] \quad + I2\times[0] \quad + I3\times[R5] + I4\times[0] \\
-Vb &= I1\times[0] \quad + I2\times[-R4] \quad + I3\times[0] \quad + I4\times[R3+R4]
\end{aligned}
$$

Figure 3: (a) devresi için göz-akım denklemleri

Denklemlerde I_1, I_2, I_3, I_4 bilinmeyenlerdir. Ayrıca devrede V_a ve V_b değerleri de bilinmediği için ek olarak 2 denkleme daha ihtiyaç vardır. Ayrıca I_a bilinen bir değer olduğu için de oradan ek bir denklem daha çıkartılabilir.

Ek Denklemler

$$V_b = 2 \times (I1 - I2) \times R2 \qquad V_a = I3 \times R5 + V2 \qquad I_a = -I2 + I3$$

Ek denklemler kullanılarak genişletilmiş çözüm matrisi Fig. (4) elde edilebilir. Bu matriste bilinmeyen vektörü $I_1, I_2, I_3, I_4, V_a, V_b$ olarak yazılabilir.

$$\begin{aligned}
V1 &= I1 \times [R1+R2] + I2 \times [-R2] + I3 \times [0] + I4 \times [0] + Va \times [0] + Vb \times [0]\\
0 &= I1 \times [-R2] + I2 \times [R2+R4] + I3 \times [0] + I4 \times [-R4] + Va \times [1] + Vb \times [0]\\
-V2 &= I1 \times [0] + I2 \times [0] + I3 \times [R5] + I4 \times [0] + Va \times [-1] + Vb \times [0]\\
0 &= I1 \times [0] + I2 \times [-R4] + I3 \times [0] + I4 \times [R3+R4] + Va \times [0] + Vb \times [1]\\
0 &= I1 \times [2 \times R2] + I2 \times [-2 \times R2] + I3 \times [0] + I4 \times [0] + Va \times [0] + Vb \times [-1]\\
-V2 &= I1 \times [0] + I2 \times [0] + I3 \times [R5] + I4 \times [0] + Va \times [-1] + Vb \times [0]\\
Ia &= I1 \times [0] + I2 \times [-1] + I3 \times [1] + I4 \times [0] + Va \times [0] + Vb \times [0]
\end{aligned}$$

Figure 4: Genişletilmiş Çözüm Matrisi

Çözüm matrisi ve kaynak ve kaynak matrisleri elde edildikten sonra sistem koda dönüştürülebilir.

```
clear all, close all; clc
V1 = 5;     V2 = 10;     Ia = 2;
R1 = 2;     R2 = 3;      R3 = 4;     R4 = 5;     R5 = 6;

% Kaynak Vektoru :
b = [V1 ; 0 ; -V2 ; 0 ; 0 ; -V2; Ia];

% Simetrik A Matrisi :
A = [    (R1+R2) ,      (-R2)     ,  (0)      ,  (0)     ,  (0)   ,  (0)
          (-R2)  ,     (R2+R4)    ,  (0)      , (-R4)    ,  (1)   ,  (0)
           (0)   ,       (0)      , (R5)      ,  (0)     , (-1)   ,  (0)
           (0)   ,      (-R4)     ,  (0)      , (R3+R4)  ,  (0)   ,  (1)
          (2*R2) ,     (-2*R2)    ,  (0)      ,   (0)    ,  (0)   , (-1)
           (0)   ,       (0)      ,  (R5)     ,   (0)    , (-1)   ,  (0)
           (0)   ,      (-1)      ,  (1)      ,   (0)    ,  (0)   ,  (0)];

% Sistem Cozumu
B = A\b;
% B = [I1 I2 I3 I4 Va Vb]
%% Saglamasinin Yapilmasi
Vx = R2*(B(1)-B(2));

Iy10 = (V1 - Vx)/R1;    Iy11 = B(1);
```

```
disp([Iy10 ; Iy11])

Iy20 = (Vx - B(5))/R4 ; Iy21 = B(2) - B(4);
disp([Iy20 ; Iy21])

disp([2*Vx ; B(6)])
disp([(V2-B(5))/R5 ; -B(3)])
disp([Ia ; B(3)-B(2)])
```

Göz-akım yöntemi ile ayrıca fazörel devrelerin çözümünde de kullanılabilir. Devre için verilen sabit bir frekans altında kapasite ve endüktans elemanlarının empedans değerleri Z_c ve Z_L bulunur. Arından Fig. (5) devre denklemleri ede edilir. Devre içinde bilinmeyenler sadece göz akımları olduğu için 4 denklem ile sistem çözülebilir. Devre çözümünde bir diğer önemli konu ise akım yönlerinin aynı seçilmesidir. Bu sayede devre matrisi simetrik çıkmakta ve hızlı bir şekilde türetilebilmektedir.

$$\begin{aligned}
V1 &= I1\times[Zc1+Zl1+Zr1] + I2\times[-Zl1] && + I3\times[0] && + I4\times[-Zr1]\\
0 &= I1\times[-Zl1] + I2\times[Zl1+Zr2+Zr3] && + I3\times[-Zr3] && + I4\times[0]\\
-V2 &= I1\times[0] + I2\times[-Zr3] && + I3\times[Zr3+Zl2+Zc2] && + I4\times[-Zc2]\\
V3 &= I1\times[-Zr1] + I2\times[0] && + I3\times[-Zc2] && + I4\times[Zr1+Zc2]
\end{aligned}$$

Figure 5: (b) Devresi İçin Çözüm Matrisi

Aşağıdaki kodlar incelendiğinde gerilim değerleri *Genlik* ve *Faz* olarak verilmektedir. Faz değerleri açı cinsinden olduğu için *polar2complex* handle fonksiyonunda açı değişkeni olan y 180'e bölünmüştür. Bu fonksiyon ile *Genlik-Açı* polar düzleminde verilen değer $a+i\times b$ formuna dönüştürülür.

```
clear all, close all; clc
polar2comlex = @(x,y)(x*cos(pi*y/180)+1i*x*sin(pi*y/180));
comlex2polar = @(x)( [sqrt(real(x).^2+imag(x).^2) , atan( imag(x)./real(x) )
    ]);

Fo = 1e3;
wo = 2*pi*Fo;
% Acilar derece cinsindendir
V1 = polar2comlex(10,30);
V2 = polar2comlex(20,20);
V3 = polar2comlex(30,10);

Zr1 = 10;
Zr2 = 100;
Zr3 = 100;

C1 = 1e-4;  Zc1 = 1/(1i*wo*C1);
```

```
C2 = 1e−3;  Zc2 = 1/(1i∗wo∗C2);

L1 = 2e−2;  Zl1 = 1i∗wo∗L1;
L2 = 1e−3;  Zl2 = 1i∗wo∗L2;
%% Devre Denklemlerinin Yazilmasi

% Kaynak Vektoru :
b = [V1 ; 0 ; −V2 ; V3];

% Simetrik A Matrisi :
A = [ (Zc1 + Zl1 + Zr1) ,    (−Zl1)     ,      (0)      ,  (−Zr1)
          (−Zl1)       , (Zl1+Zr2+Zr3) ,    (−Zr3)     ,   (0)
           (0)         ,    (−Zr3)     , (Zr3+Zl2+Zc2) ,  (−Zc2)
          (−Zr1)       ,      (0)      ,    (−Zc2)     , (Zr1+Zc2)  ];

I = A\b;

%% Saglamasinin Yapilmasi
Vx = Zr1∗(I(1)−I(4));

Iy10 = I(4) − I(3); Iy11 = (Vx+V3)/(Zc2);
disp([Iy10 ; Iy11])

Iy20 = I(1) − I(2); Va   = V1−I(1)∗Zc1; Iy21 = (Va−Vx)/Zl1;
disp([Iy20 ; Iy21])
```

Öneri

2. devre için akım ve elemanlar üzerindeki gerilim grafikleri çizdirilebilir. Bu sayede faz farkları gözlemlenmiş olur. Ayrıca frekans arttıkça devre elemanlarının karekteristikleri incelenebilir. Ayrıca iki gözlü el ile çözümü kolay bir devre oluşturularak çözümünü MATLAB ile yapmaya çalışabilirsiniz.

5.2 Temel Olasılık ve İstatistik

Olasılık teorisi 17. yüzyılda Fransa'da matematikçi olan Blaise Pascal ve Pierre de Fermat tarafından oyunlardaki şanslarını matematiksel olarak modelleyebilmek amacıyla ortaya atılmıştır. Günümüzde olasılık teorisi, matematiğin önemli bir dalıdır ve haberleşme uygulamalarından yapay zekaya, hava durumu tahmininden yeni ilaçların risk hesaplamalarına kadar bir çok alanda kullanılmaktadır. MATLAB olasılıksal hesaplamalar yapmak için uygun araçlardan biridir. Örnek olarak *rand(.)*, *randn(.)*, *randi(.)* ve *randperm(.)* komutları ile rastgele değişken matrisleri oluşturabilirsiniz.

Örnek: 1 Bir Rasgele Değişkeninin Histogram Analizi

$X \sim \mathcal{N}\{10, 5\}$ olan rasgele değişken olsun. MATLAB ile bu rasgele değişkenin histogram analizini yapınız ve $N_s \in \{1e1, 1e2, 1e3, 1e4, 1e5\}$ gerçekleme sayısına göre teorik ve pratik değerler arasındaki farkı gözlemleyiniz.

```
clear all, close all; clc
Ns  = 1e5;      sgm = 5;        mu  = 10;
x = sgm*randn(Ns,1) + mu;

[mu , sgm ; mean(x) , std(x)]

figure,
hist(x,100),grid on,
xlabel('X Rasgele Degiskeninin Degeri')
ylabel('Bulunma Yogunlugu')
title('X Rasgele Degiskeninin Histogram Grafigi')
```

```
       Ns = 1e1 icin:                      Ns = 1e5 icin:
        == ans ===                          == ans ===
Teorik : 10.0000    5.0000         Teorik : 10.0000    5.0000
Pratik : 11.0894    5.8334         Pratik : 10.0133    5.0047
```

Yukardaki değerlerden de görüleceği üzere N_s gerçekle sayısı düşük olduğunda istatistikse özelliklerde teori ile pratik arasında sapma önemli oranlarda olmaktadır. Gerçekleme sayısı arttırıldığında bu değerler birbirlerine daha benzemektedirler. Fig.(6) grafiğinde de görüleceği üzere $N_s = 1e5$ olduğu durumda $\mu = 10$ olarak kolaylıkla ölçülebilir.

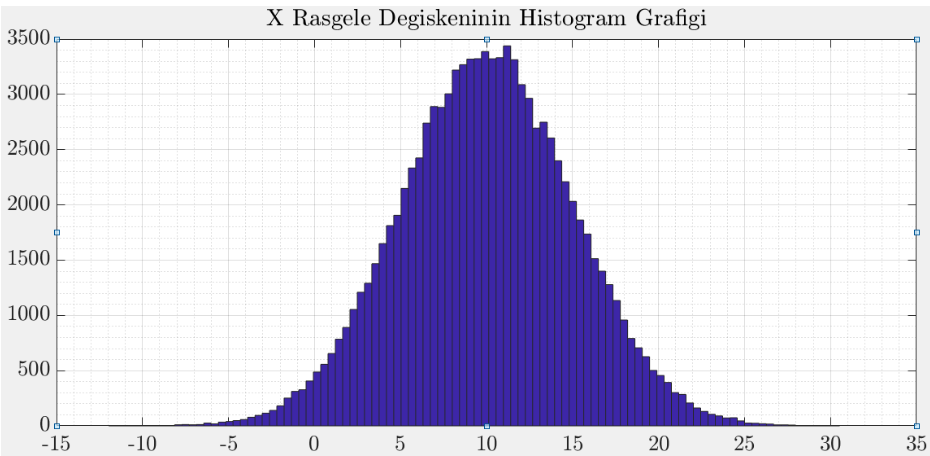


Figure 6: $X \sim \mathcal{N}\{10, 5\}$ değişkeni için histogram analizi

Öneri

Bu örneği $X \sim uniform\{-5,5\}$ rasgele değişkeni için tekrar edebilirsiniz. Ayrıca $\mathcal{N}\{\mu,\sigma\}$ ve $uniform\{a,b\}$ gibi matematiksel gösterimlerin ne anlama geldiğini ve MATLAB'ta nasıl yazıldığını öğreniniz. hist fonksiyonunun kullanımını araştırınız.

Örnek: 2 Üç Farklı Olasılık Yönteminin Karşılaştırılması

Bir kasada 1000 dondurmadan 400 tanesinde "Bedava" yazmaktadır. Seçilen 50 dondurmadan kaç tanesinde bedava yazabileceğini gösteren olasılık kütle fonksiyonunu hesaplayınız.

Problemin çözümü için üç farklı yol izlenebilir. *a.* Kombinasyon Hesabı, *b.* Binomial Dağılım, *c.* Normal Dağılım.

a: Kombinasyon hesabunda i tane bedava çıkma olasığılı $p(i) = \frac{\mathbf{(400\ C\ i)\times(600\ C\ (50-i)))}}{\mathbf{(1000\ C\ 50)}}$

b: Binom dağılım kullanıldığında bedava çıkma olasığı $p = 400/1000$. Buradan i. seçenek için binom dağılımı $p(i) = (\mathbf{50C(i)}) \times (\mathbf{p})^{(\mathbf{i})} \times (\mathbf{1-p})^{(\mathbf{50-i})}$

c: Normal dağılım için ortalama : $\mu = n \times p$ ve standart sapma : $\sigma = \sqrt{n \times p \times q}$ olarak bulunur. Buradan i. seçenek için olasılık $p(i) = \frac{\mathbf{1}}{\sqrt{\mathbf{2}\times\pi\times\sigma}} \times \mathbf{e}^{-\frac{(\mathbf{i}-\mu)^{\mathbf{2}}}{\mathbf{2}\times\sigma^{\mathbf{2}}}}$ olarak hesaplanabilir.

```
clear all, close all; clc

N = 1000; % kasadaki dondurma sayisi
D = 400; % bedava
B = N-D; % tekrar dene (td)
p = D/N; % bedava cikma olasiligi
q = B/N; % tekar dene cikma olasiligi
n = 50; % secilen dondurma sayisi
% binom dagilimdan gauss dagilimina gecis
mu = n*p; % gauss icin ortalama
sd = sqrt(n*p*q); % standart sapma
for i = 0:n
comb(i+1) = (nchoosek(D,i)*nchoosek(B,n-i))/nchoosek(N,n); % kombinasyon uzerinden hesaplama
% ((400 C i)*(600 C (50-i)))/(1000 C 50)
binm(i+1) = nchoosek(n,i)*(p)^i*(q)^(n-i); % Binomial dagilim
norm(i+1) = 1/(sqrt(2*pi)*sd)*exp( -(i-mu)^2/(2*sd^2) ); % Normal dagilim
end
figure,
stem(comb,'b'),hold on,
```

```
stem(binm,'r'),hold on,
stem(norm,'g'),grid on,
legend('comb','binm','norm')
```

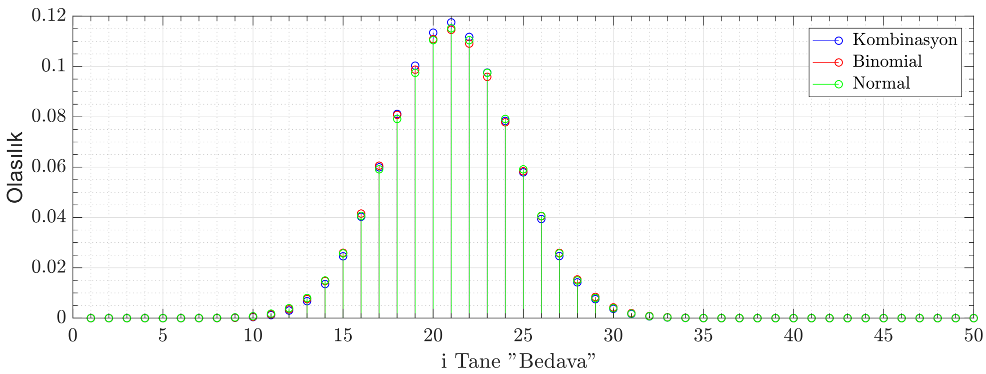


Figure 7: Üç olasılık hesaplama yönteminin sonuçlarının karşılaştırılması

Fig. (7)'de görüleceği üzere uygun sayılar verildiğinde dağılımlar birbirlerine benzemektedir.

Örnek: 3 Eğik Atış Problemleri İçin Menzil Olasılığı

Fig. (8) grafikte V_o ilk hızı ile atılan bir cismin menzil eğrisi verilmiştir. Cisim $\theta \sim uniform\{15^o, 25^o\}$ rasgele değişkeni ile fırlatıldığına göre menzil rasgele değişkeni R'nin olasılık yoğunluk fonksiyonunu hesaplayın. Menzil denklemi : $R = \frac{V_o^2 \times sin\{2\theta\}}{g_o}$

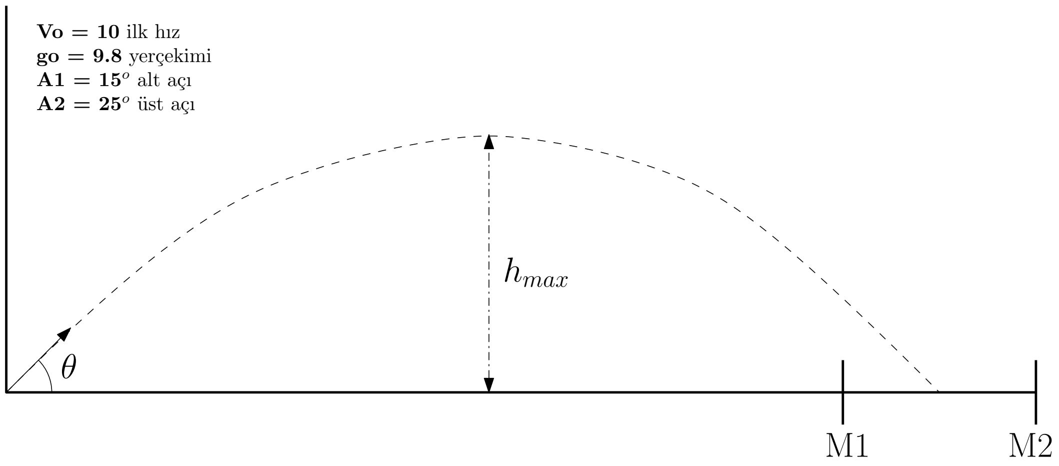


Figure 8: Devre Analizi

Olasılık teorisinde rasgele değişkenler arasında dönüşümler önemli bir konudur. Tasarlanan fiziksel sistemlerin sonuçları, girdilerin istatistiklerine göre değişkenlik gösterir. Sonuçlar hakkında öngörüde bulunabilmek için değişkenler arası olasılık dağılım fonksiyonu dönüşümlerinin yapılması gerekir. Bu problemde eğik atışta θ açısı $uniform$ dağılıma sahiptir. Buna göre cismin nereye düşeceğine dair istatistikler bulunacaktır.

$$\mathbb{P}\{R \le r\} => \mathbb{P}\left\{\frac{V_o^2 \times sin\{2\theta\}}{g_o} \le r\right\}$$
$$\mathbb{P}\left\{\theta \le 0.5sin^{-1}\left(\frac{r \times g_o}{V_o^2}\right)\right\} => \int_{A_1}^{0.5sin^{-1}\left(\frac{r\times g_o}{V_o^2}\right)} \frac{1}{A_1 - A_2} d\theta => \frac{1}{A_1 - A_2} \times \left[0.5sin^{-1}\left(\frac{r \times g_o}{V_o^2}\right) - A_1\right]$$
$$\mathbb{F}\{r\} => \frac{1}{A_1 - A_2} \times \left[0.5sin^{-1}\left(\frac{r \times g_o}{V_o^2}\right) - A_1\right]$$
$$\mathbf{p}(r) = \frac{d\mathbb{F}\{r\}}{dr} => \frac{1}{A_1 - A_2} \times \frac{0.5 \times \frac{g_o}{V_o^2}}{\sqrt{1 - \left(\frac{r\times g_o}{V_o^2}\right)^2}} \quad r \in \left[\frac{V_o^2 \times sin\{2A_1\}}{g_o}, \frac{V_o^2 \times sin\{2A_2\}}{g_o}\right]$$

Bunulan $\mathbf{p}(r)$ menzil olasılık yoğunluk fonksiyonudur. Yukarıdaki hesaplar eğer doğru ise $\mathbf{p}(r)$'nin $r \in \left[\frac{V_o^2 \times sin\{2A_1\}}{g_o}, \frac{V_o^2 \times sin\{2A_2\}}{g_o}\right]$ aralığındaki integralinin 1 olması gerekir.

```
clear all, close all; clc
Vo = 10;        % ilk hiz
g  = 9.8;       % yercekimi

A1 = (pi/180)*15;
A2 = (pi/180)*25;
Pa = 1/(A2-A1);

alfa = g / Vo^2;

M1 = ( (Vo^2) * sin( 2*A1 ) ) / g
M2 = ( (Vo^2) * sin( 2*A2 ) ) / g

Pr = @(x)(0.5*Pa*alfa* 1./( sqrt( 1 - (alfa*x).^2) ) ); % r degiskeni icin pdf
Er = @(x)(x) .* Pr(x); % r degiskeni beklenen deger fonksiyonu

sum_Pr = integral(Pr,M1,M2);  % olasilik Toplami
exp_r  = integral(Er,M1,M2);  % beklenen deger
Vr     = @(x)( x - exp_r ).^2 .* Pr(x); % varyans fonksiyonu
var_r  = integral(Vr,M1,M2); % varyans
[sum_Pr,exp_r,var_r]
%===========================================%
ans =
    1.0000    6.5258    0.6175
```

Kodlar çalıştırıldığında bulunan teorik değerler [1.0000 6.5258 0.6175]. Bulunan değerlerin pratik sonuçlar ile karşılaştırılması için N_{trial} tane [15 , 25] aralığında

$uniform$ değerler üretilmiştir. Bu değerler üzerinden menziller bulunmuştur. Menzillerin pratik değerleri üzerinden istatistiksel değerleri ölçülmüştür.

```
Ntrial = 1e5;
a      = A1 + (A2-A1)*rand(Ntrial,1); %[A1,A2] arasi uniform dagilim.
R      = ( (Vo^2) * sin( 2*a ) ) / g;

[   sum_Pr            ,  exp_r    , var_r
 sum(hist(R))/Ntrial , mean(R) , var(R)]
%===========================================%
 ans =
    1.0000    6.5258    0.6175
    1.0000    6.5237    0.6158
```

Öneri

A_1 ve A_2 değerlerini değiştirerek ortalama ne standart sapmanın değişimini gözlemleyiniz. N_{trials} değerini değiştirerek sonuçları inceleyiniz. Önemli: A_2 sınırı 45^o'yi geçmemesi gerekir. *handle* fonksiyonların nasıl kullanıldığına dikkat ediniz.

5.3 İşaretler ve Istatistiksel Ozellikleri

İşaret, bir parametrenin başka bir parametreye göre nasıl değiştiğinin matematiksel olarak modellenmesidir. Örnek olarak, elektronik bir devrede zaman içinde voltaj değişimi veya görüntüdeki mesafeye göre değişen parlaklık verilebilir. İşaretleri oluşturan parametreler bilindiğinde, işaret deterministik olur. Ancak pratikte her parametre bilinemez. Örneğin, kablosuz haberleşme sistemlerinde antenden gönderilen işaret alıcıya ulaşana kadar bir çok etkiye maruz kalır. Gürültü bu etkilerden biridir. İşarete gürültü eklenmesi, kablosuz haberleşme esnasında oluşan bozunmanın modellenmesi olarak kabul edilebilir. Gürültü hiç bir bilgi içermeyen, ancak olasılıksal olarak modellenebilen bir değişkendir. İşaretlerin bazı parametreleri bilinmediği zaman, bilinmeyen parametreler rastgele değişken olarak modellenebilir. Bu durumda işaret stokastik bir işaret olur.

```
clear all, close all; clc
%===========================================================================
% Ayrik isaretlerin olusturulmasi
%===========================================================================
% COS Isareti
n  = -50:50;
fo = 1/50;
y1 = cos(2*pi*fo*n);
%===========================================================================
% Gauss Isareti
```

```
mu  = 10;
sgm = 10;
y2 = exp(-(n-mu).^2/sgm^2);
%==========================================================================
% Gurultulu Kare Dalga
fo = 1/25;
x = square(2*pi*fo*n);
y3 = x + 0.1*randn(1,numel(x));
%==========================================================================

figure,
subplot(311),stem(n,y1,'b'),grid on
title('Ayrik COS Isareti'),
xlabel('Ayrik Zaman'),ylabel('Genlik')
subplot(312),stem(n,y2,'r'),grid on
title('Ayrik Gauss Isareti'),
xlabel('Ayrik Zaman'),ylabel('Genlik')
subplot(313),stem(n,y3,'m'),grid on
title('Gurultulu Kare Isareti'),
xlabel('Ayrik Zaman'),ylabel('Genlik')
```

Öneri

$x[n] = 0.2^n \times u[n] + 2^n \times u[-n-1]$ işaretini oluşturmayı deneyiniz. Bu işarete $\nu \sim \mathcal{N}\{1, 0.5\}$ gürültüsünü ekleyiniz. İpucu: 0.2 için n_1, 2 için n_2 düzlemlerini ayrı ayrı oluşturup x_1 ve x_2'yi elde ettikten sonra bu fonksiyonları toplayabilirsiniz.

Bu kısma sürekli işaretlerin nasıl oluşturulacağı linspace komutu ve süreklilik için örneklemenin çok yüksek seçildiğinden bahsedilebilir. Ayrıca kodların çalıştırılarak görsellerin bireysel incelenmesi ve yorumlanması istenebilir.

```
clear all, close all; clc
Fs = 1e4; % Surekli zamani modellemek icin ornekleme fekansi
          % Bu frekans simulasyon esnasinda kullanilacak olan maksimum
          % frekanstan en az 20-30 kat fazla olmalidir.
Ts = 1/Fs;
t = -2:Ts:2;
F0 = 0.5:0.5:2; % 1-10 araliginda farkli frekanslar
x = cos(2*pi * F0' * t); % 10xnumel(t) boyutlarinda sinyal matrisi

for i = 1:numel(F0)
    legend_isim{i} = ['Frekans: ',num2str(F0(i))];
end

figure,plot(t,x),grid on,
xlabel('Zaman'),ylabel('Genlik')
title('Surekli Zaman COS Isaretleri')
legend(legend_isim{:})
```

```
%% ========================================================================
% Gurultulu Ucgen Dalga
F0 = 2;
x  = sawtooth(2*pi*F0*t);
SNR_orani = [20,10,5,0,-5];

figure
for i = 1:numel(SNR_orani)
y(i,:) = awgn(x,SNR_orani(i),'measured');  % Sinyale Gurultu Ekleme
subplot(numel(SNR_orani),1,i),plot(t,y(i,:),'color',rand(1,3)),grid on
ylabel(['SNR:',num2str(SNR_orani(i))])

snr_hesaplanan = snr_hesapla(x,y(i,:));
title(['Hesaplanan SNR: ',num2str(snr_hesaplanan)])
end
xlabel('Zaman'),suptitle('Gurultulu Isaretler')
%% ========================================================================
% Farkli Frekanslardaki Sinusoidallerin Toplami ve Genlik Spektrumu
F0 = 1:5;
A0 = linspace(1,5,numel(F0));

x = bsxfun(@times,A0',cos(2*pi*F0'*t));
y = sum(x,1);

figure
FY = 1/numel(y)*fftshift(abs(fft(y)));
F = linspace(-Fs/2,Fs/2,numel(y));
figure,
subplot(211),plot(t,y),grid on
xlabel('Zaman [sn]'),ylabel('Genlik')
title('Sinusoidallerin Toplami')

subplot(212),plot(F,FY),grid on
xlim([-2*max(F0),2*max(F0)])
xlabel('Frekans [Hz]'),ylabel('Genlik')
title('Genlik Spektrumu')
```

```
function SNR = snr_hesapla(x,y)
sinyal_gucu  = sum(x.^2);
gurultu_gucu = sum((y-x).^2);  % y(t) = x(t) + n(t)

SNR = 10*log10(sinyal_gucu/gurultu_gucu);
end
```

Öneri

SNR hesaplamasını harici bir fonksiyon olarak yazmak yerine handle fonksiyonu olarak ifade etmeye çalışınız. $bsxfun$ fonksiyonunu araştırınız.

Örnek: 3 Gürültülü ve Gürültüsüz İşaretlerin Otokorelasyon Analizi

$x(t) = rect(2\pi \times 2 \times t)$ işaretine $0dB$ değerinde bir gürültü etki etsin. Gürültülü ve gürültüsüz işaretlerin otokorelasyon analizini yapınız.

Bir işaretin otokorelasyonu (1)'teki gibi işaretin ters çevrilmeden kendisi ile konvolüsyonu olarak tanımlanmaktadır. Elde edilen $R_{xx}(t)$ işaretinin Fourier Dönüşümü işaretin Güç Spektrumunu vermektedir. Otokorelasyon modern haberleşme sistemlerinde işaretlerin tanımlanması, kestirimi, algılanması veya yön bulma gibi konularda kullanılan çok önemli bir matematiksel araçtır.

$$R_{xx}(t) = \int_{-\infty}^{\infty} x(\tau) \times x(t+\tau) \tag{1}$$

Gürültü eklenmiş bir işareti $y(t) = x(t) + n(t)$ olarak tanımlansın. Bu $y(t)$ işaretinin otokorelasyonu $R_{yy}(t) = R_{xx}(t) + R_{nn}(t) + 2 \times R_{xn}(t)$ olarak ifade edilir. $R_{xn}(t)$, $x(t)$ ile gürültü $n(t)$ arasındaki korelasyonu simgeler. Gürültü ile herhangi bir işaret ilintisiz olduğu için bu değer teorik olarak 0 olarak kabul edilir ve pratikte de etkisi diğer iki bileşenin yanında çok düşüktür. Ayrıca $R_{nn}(t)$ ifadesi gürültünün kendisi ile korelasyonunu simgeler. Gürültü $n(\tau)$ teorik olarak $n(\tau + t)$ ile t'nin 0'dan farklı değerleri için ilintisizdir. Bu nedenle pratik uygulamalarda da gürültünün etkisi $t = 0$ noktasından en şiddetli olarak hissedilir.

```
clear all, close all; clc
Ts = 0.01;
t = -1:Ts:1;
F0 = 2;

x = 5*square(2*pi*F0*t);  % Kare Dalga
y = awgn(x,0,'measured'); % 0 dB Gurultu Ekleme

Rx = xcorr(x); % x'in otokorelasyonu
Ry = xcorr(y); % y'in otokorelasyonu

tfark = linspace(-2*max(t),2*max(t),numel(Rx));

figure,
subplot(211), plot(t,x),ylim([0 5.1]),grid on
xlabel('Zaman [sn]') , ylabel('Genlik [V]')
subplot(212), plot(t,y),ylim([0 5.1]),grid on
xlabel('Zaman [sn]') , ylabel('Genlik [V]')

figure,
plot( tfark , Rx,'b'),hold on,plot( tfark , Ry,'r'), grid on
xlabel('[t_0 - t_1] Zaman Farki'), ylabel('Otokorelasyon Sonucu')
title('R_x ve R_y Icin Otokorelasyon Analizi')
legend('x(t) : Gurultusuz Isaret', 'y(t) : Gurultulu Isaret')
```

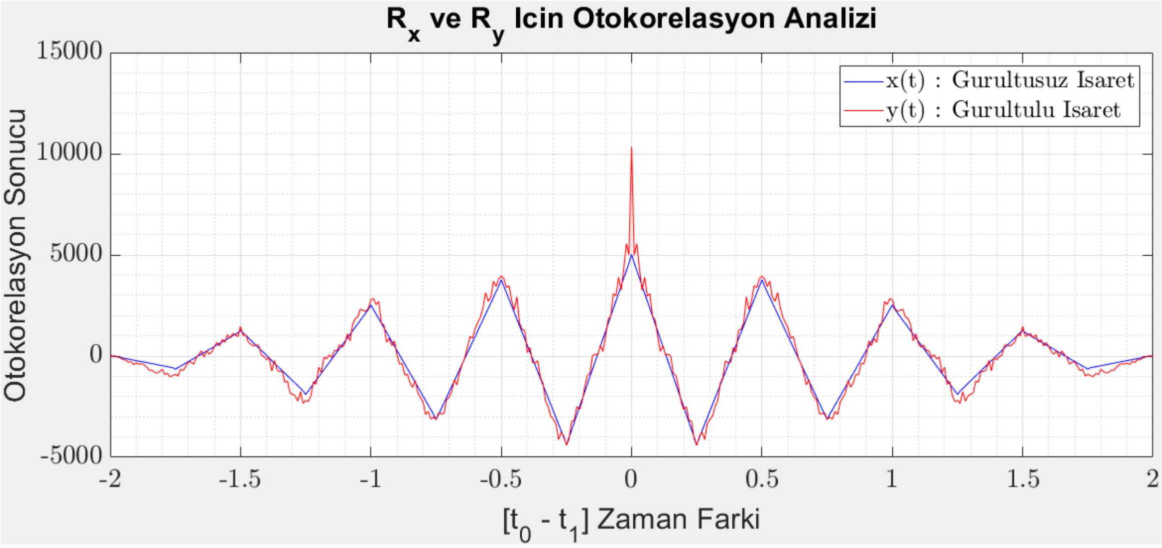


Figure 9: $x(t)$ ve $y(t)$ surekli isaretlerinin otokorelasyon grafikleri.

Fig. (9) grafiğinde gürültünün en çok 0. değeri bozduğuna dikkat ediniz. Bunun nedeni gürültünün kendisi ile korelasyonundan en yüksek bozucu ektikinin $R_{nn}(0)$ değerinde olmasıdır.

Öneri

Otokorelasyon kavramını araştırabilirsiniz. Periyodik işaretlerin otokorelasyon özeliklerini inceleyiniz. Ayrıca yukarıdaki kodda $dB \in \{-5, -3, 3, 10, 20\}$ için analizi tekrarlayıp çıkan sonuçları yorumlayınız.

6 Deneyde Yapılacaklar

Deney kapsamında 3 soru bulunmaktadır. Her soru için mutlaka Fig. (10)'de gösterildiği gibi yeni sekme açılmalı ve S_1 olarak isimlendirilmelidir. Sorular altında bulunan alt başlıklar için sekme sayfası %% ile bölümlere ayrılmalıdır ve S_1a olarak isimlendirilmelidir. $CTRL + ENTER$ ile her bir bölüm bağımsız olarak çalıştırılabilir.

```
S_1.m   S_2.m   +
%% S-2a
clear all, close all; clc
A = [1 2 3; 4 5 6; 7 8 9];
a = A(1,2); disp(['a = ',num2str(a)])
%% S-2b
b = A*A*A;
c = A(:);
%% S-2c

d = A.^2*A';
e = sum(d(:))
```

Figure 10: Örnek çözüm sistematiği

6.1 İndis ve Çarpma İşlemleri (30 pt)

S1: Bu sorunun amacı MATLAB'ta yoğun olarak kullanılan indis ve çarpma işlemlerinde pratiklik kazanılmasıdır. $\mathcal{A} \in \Re^{4\times4}$ boyutlarında rasgele oluşturulan bir matris olsun. Bu matris ile ilgili aşağıdaki soruları cevaplayınız. Önemli: Her deney grubuna deney esnasında bu matrislerden *sadece birisi* rasgele atanacaktır.

42	15	40	20	44	42	30	13	55	89	5	65	97	70	25	86
72	9	54	88	3	33	27	51	71	90	44	28	55	22	43	98
0	19	42	3	55	20	62	18	29	13	3	68	97	98	78	16
30	35	69	67	44	62	53	79	51	21	46	59	71	1	20	60
(A)				(B)				(C)				(D)			

Figure 11: Soru-1 için $\mathcal{A}$ matrisleri

1A: a = $\mathcal{A}$[2,5] , b = $\mathcal{A}$[2,:] , c = $\mathcal{A}$[end,:] değerlerini disp komutu ile ekrana yazdırınız. Ipucu: disp(['a = ' , num2str(a)]) Örn. çıktı : a = 3, b = 1 2 3 4 **(3pt)**

1B: $\mathcal{A}$ matrisi ile transpozesinin element çarpımını bulunuz. **(3pt)**

1C: $\mathcal{A}$ matrisi ile transpozesinin matris çarpımını bulunuz. **(4pt)**

1D: 3. sütun ile 2. satırın element çarpımı İpucu: Element çarpmanın doğru olması için iki vektörün aynı boyutlarda olması gerekir ve bu çarpmadan yine aynı boyutlarda bir vektör elde edilmelidir. **(4pt)**

1E: 1 satır ile 4. sütunun matris carpımı **(4pt)**

1F: $\mathcal{A}$[2:3,2:3] ile $\mathcal{A}$[2:3,end] matis çarpımı **(4pt)**

1G: A matrisi ile A'nın 2. sütununu matris çarpımı **(4pt)**

1H: $\mathcal{A}$[2 , 2:end] ile $\mathcal{A}$[2:end , 2] element çarpımı **(4pt)**

6.2 Grafik Üzerinden İşaretlerin Kodlaması (30 pt)

Grafik üzerinden işaretlerin okunması birçok alanda karşılaşılan bir iştir. Bu doğrultuda 2. soruda grafiği verilen işaretlerin kodunun yazılması istenmektedir. Deney esnasında buradaki örnek **VERİLMEYECEKTİR.** Her grup için farklı olarak bu örneğe benzer bir soru sorulacaktır. Örneğin amacı ön-çalışmaya yardımcı olmaktır. Deney esnasında verilen grafiğin *birebir olarak aynısın* elde edilmesine bakılacaktır.

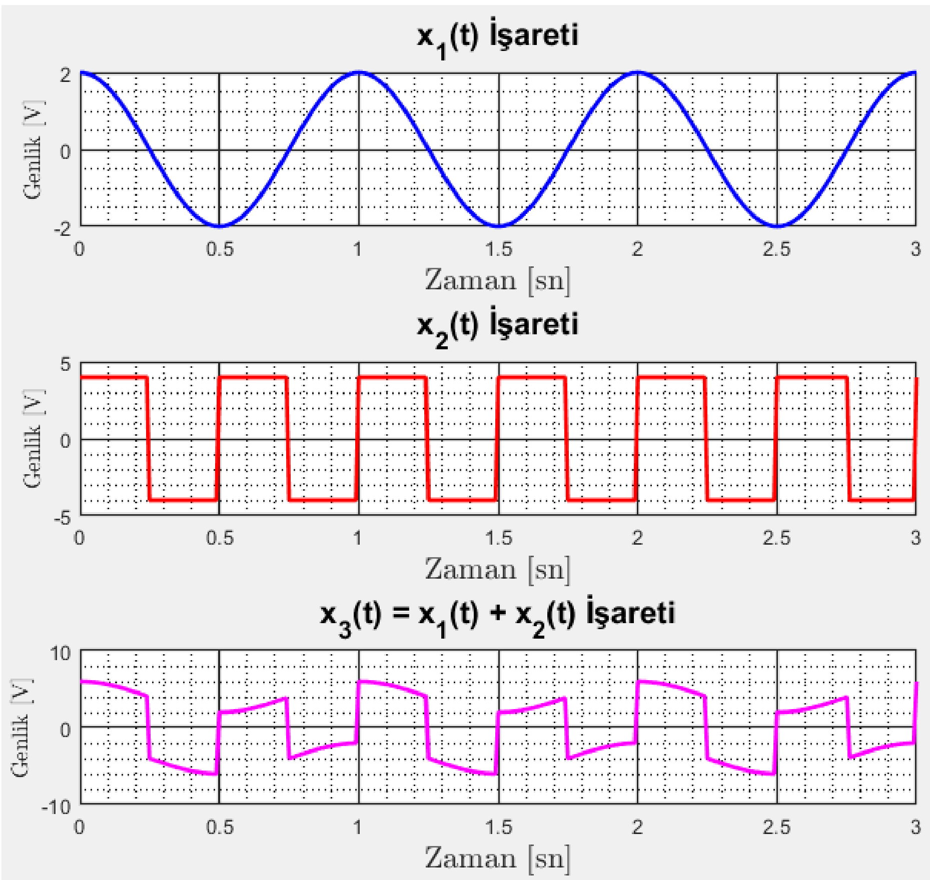


Figure 12: Soru-2 için grafikler

S2: Fig. (12)'te verilen grafiğin aynısını üretecek olan kodu yazınız. Zaman örneklemesi için $F_s = 100$ olarak kullanabilirsiniz. Deney esnasında eksen skalası ve isimleri, grafik renkleri ve başlıklar puan alacağınız yerlerdir. Bu noktaları ayrıntı olarak görmeyip mutlaka yapmalısınız. Font büyüklük ve kalınlıkları önemli değildir.

6.3 Temel İşaretlerin Oluşturulması ve Gürültü Analizi (40 pt)

Bir işaret alıcıya gelinceye kadar birçok bozucu etkene maruz kalır. Isıl gürültü, faz kaymaları, sönümleme ve girişim gibi etkenler orjinal işareti olumsuz yönde etkilerler. Bilginin alıcı tarafında kayıpsız olarak elde edilebilmesi için işaret işleme tekniklerine ihtiyaç vardır. Bu soru kapsamında temel işaretlerin nasıl oluşturulabileceği ve gürültü analizinin hangi metotlar ile yapılabileceği öğrenilecektir. Soruda örnek olarak verilen işaret çalışma amaçlıdır. Her deney grubu için farklı işaret verilecektir.

S3: $x(t) = 3.2 \times cos(2\pi 0.25 \times t) - 2.1 \times square(2\pi 2 \times t) + 5.3 \times sin(2\pi 0.5 \times t + \pi/17)$ olarak verilsin. $F_s = 100$ ve $t \in [-5, 5]$ aralığında üreteceğiniz işaret için:

3A: İşareti x değişkenine atayacak şekilde oluşturunuz. *plot* komutu ile çizdiriniz. *xlabel*, 'Zaman [sn]' ve *ylabel*, 'Genlik [V]' ve başlık 'Giriş İşareti' olacak şekilde yazdırınız. (**8pt**)

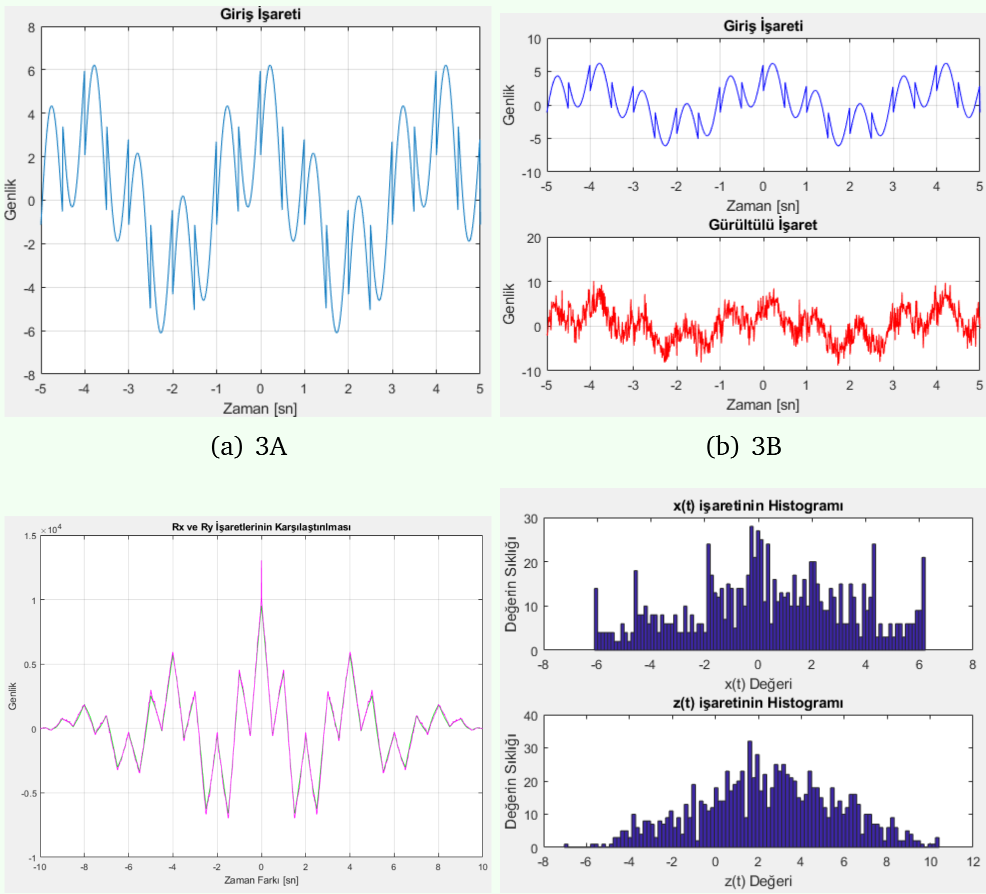
3B: $awgn(.)$ fonksiyonunu kullanarak $x(t)$ işaretine $5dB$ ve $'measured'$ gürültü ekleyiniz y değişkenine atayınız. Gürültüsüz ve gürültülü işaretleri $subplot$ komutu ile 'mavi' ve 'kırmızı' renklerde çizdiriniz. X ve Y eksenlerine uygun etiket isimlerini yazınız. Başlıkları unutmayınız. (**8pt**)

3C: SNR bulmak için $snr\_degeri(x, y)$ **handle** fonksiyonu olarak yazınız ve **4B** için x-y çifti üzerinden SNR değerini hesaplayınız. (**8pt**)

3D: $x(t)$ ve $y(t)$ işaretlerinin otokorelasyonlarını hesaplayınız. R_x ve R_y değişkenlerine atayınız. R_x ve R_y'nin grafiklerini $holdon$ kullanarak 'yeşil' ve 'magenta' olarak çizdiriniz. Uygun etiketleri yazınız. (**8pt**)

3E: $x(t)$ işaretine $\eta \sim \mathcal{N}\{2, 1\}$ istatistiğinde bir gürültü etki etsin. Gürültülü $z(t)$ işareti ile $x(t)$ işaretlerinin 100 noktalı histogramını $subplot$ kullanarak çizdiriniz. SNR değerini **handle** fonksiyonu ile bulunuz. (**8pt**)

DENEY SONU $\therefore$



(a) 3A (b) 3B

(c) 3D (d) 3E

Figure 13: Deney 6.3'ün Cevapları

7 Deney Harici Ek Çalışma İçin Problemler

Deneyde yapılan kodlamalara ek olarak bu bölümde bulunan alıştırmaları yapmanız MATLAB üzerinde programlama ve problem çözme yeteneğinizi pekiştirecektir. Zorunlu olmayan bu kısımda, deneyi tamamlayıcı sorular vardır.

7.1 Vektörlerde İç Çarpım

İç çarpım vektör analizinde sık kullanılan bir araçtır. Herhangin bir D boyutlu vektör $v \in \Re^{D\times 1}$ olarak tanımlansın. İki vektörün iç çarpımı $< v_1, v_2 >= \sum_{i=1}^{D} v_1(i) \times v_2(i)$ veya matris gösteriminde $< v_1, v_2 >= v_1^T \times v_2$ olarak bulunur. Diğer taraftan iç çarpım $< v_1, v_2 >= |v_1||v_2| \times cos(\theta)$ olarak yazılabilir. Burada θ D boyutlu uzaydaki vektörler arasındaki açıyı belirtir. $|.|$ ifadesi MATLAB'ta $norm(.)$ fonksiyonuna dek gelmektedir. Ayrıca yazılan vektörlerin **sütun** vektörü olmasına dikkat ediniz.

(A)		(B)		(C)		(D)	
-65	-57	-12	-31	7	32	235	-50
118	-56	-254	-113	95	-41	9	100
-76	18	28	19	-34	-64	221	-189
-111	-20	-20	-61	4	-66	74	-125
-85	59	-20	-83	112	-23	68	-23

Figure 14: Soru-3 için vektörler ilk sıra: v_1, ikinci sıra: v_2

S3: Fig.(14)'de bulunan 5 boyutlu vektörlerde birinci satırlar v_1, ikinci satırlar v_2'ye denk delmektedir. Buna göre:

3A: İki vektör arasında açıyı bulan fonksiyonu $vektor\_aci(v1, v2)$ olarak **handle** fonksiyonu olarak yazınız. Ardından θ açısını derece cinsine çevirerek bulunuz. Önemli: MATLAB içinde kullanacağınız $acos$ fonksiyonu radyan olarak çıkış vermektedir. Bu değeri açıya çevirmelisiniz.

3B: v_1 ve v_2 vektörlerini birim vektörü haline getiriniz. Buldugunuz birim vektörleri e_1 ve e_2 olarak atayınız. Birim vektör $e = \frac{v}{|v|}$ olarak hesaplanabilir.

3C: e_1 ve e_2 baz vektörleri kullanılarak tanımlanan uzayda $w = -3 \times e_1 + 5 \times e_2$ vektörünü bulunuz.

3D: w ile e_1, w ile e_2 arasındaki açıları *vektor_aci* fonksiyonu ile bulunuz ve θ_1 , θ_2 değişkenlerine atayınız.

3E: v_1 vektörü ile arasındaki açı 180^o olan v_3 vektörü olsun. v_3 vektörünün nasıl bulunacağınız sözel olarak *disp(.)* fonksiyonu kullanarak ekrana yazdırınız. Örnek: v3 = 'v1 vektörü 3 ile çarpılıp normu alındıktan sonra 4 ile bölünmelidir.

7.2 π Sayısını Keşfetmek...

π sayısı doğa ile matematiği birbirine bağlayan önemli bir sayıdır. Bu sayı her ne kadar bir çemberin çevresinin çapına oranı olarak tanımlansada farklı serilerin toplam olarak bulunmasıda mümkündür. Bunlardan bazıları aşağıda verilmiştir. Bu serileri MATLAB'ta oluşturup ∞ için $[1e2, 1e4, 1e6]$ gibi sayılar alarak hesaplanan sayının ne derece π değerine yaklaştığı gözlemlenebilir.

$$\frac{\pi^2}{6} = \sum_{k=1}^{\infty} \frac{1}{k^2} \tag{2}$$

$$\frac{\pi}{4} = \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{2 \times k - 1} \tag{3}$$

$$\pi = \int_0^1 \frac{16x - 16}{x^4 - 2x^3 + 4x - 4} dx \tag{4}$$

Bilgilendirme: π bilindiği üzere irrasyonel bir sayıdır. Birçok fonksiyonun temel hesaplamalarında sıklıkla kullanılan bu sayı mikroişlemci, mikrodenetleyici ve FPGA gibi donanım tasarımlarında da yoğun olarak kullanılır. Bu hesaplamalar Look-Up Table (LUT) tabanlı yapılabilindiği gibi CORDIC (coordinate rotation digital computer) algoritması ile de gerçeklenebilir. Sizlerde CORDIC ve LUT kavramlarını araştırıp sinusoidal işaretlerin donanımda nasıl kullanıldığını öğrenebilirsiniz. ♮

Kaynak :

■ `http://mathworld.wolfram.com/PiFormulas.html`

▲ `www.allaboutcircuits.com/technical-articles/an-introduction-to-the-cordic-algorithm/`