04- MERVE TELLEKEN

# BENZETİM ve MODELLEME (1. hafta)

Benzetim; gerçek hayattaki bir sistemin veya sürecin çalışmasının taklit edilmesidir. (genellikle bilgisayar üzerinde)
Benzetim, sistemin yapay geçmişinin üretilmesine ve gerçek sistemin karakteristik özelliklerine dair çıkarımlar yapmak üzere bu geçmişin gözlenmesine olanak verir.
Genel anlamda benzetim; zaman içinde sistemin işleyişinin taklididir.

Benzetimin gerçekleştirilebilmesi için; karmaşıklık ve rassallık belirli bir düzeyde olmalıdır. Karmaşıklık arttıkça performans düşeceğinden benzetim azalır.

Bir benzetim modeli geliştirildikten ve geçerliliği sağlandıktan sonra, gerçek sistem hakkındaki çeşitli sorulara cevap aramak için kullanılır.

Bir benzetim modeli;
- gerçek sistem üzerinde yapılacak değişikliklerin etkilerini (analiz aracı) ve
- yeni kurulacak bir sistemin performansını tahmin etmek için (tasarım aracı) analiz ve tasarım aracı olarak kullanılırlar.

Model; Genelde matematiksel bir modeldir. Bir sistemin karakterize edilmesidir. Bir sistemin gösterimi olarak tanımlanabilir.
Fiziksel model; bir matematiksel sistemin analog bir devre olarak elde edilmesidir.

## Benzetim ve Modelleme Amaçları

- Değerlendirme
- Karşılaştırma
- Tahmin
- Duyarlılık Analizi
- Optimizasyon
- Darboğaz analizi

## Benzetim ve Modelleme Ne Zaman Kullanılır?

1) Üzerinde çalışılacak sistem çalışmaya, deney yapmaya uygun değilse
2) Sistem henüz tasarım aşamasında ise,
3) Problemin analitik çözümü mümkün değilse,
4) Problemin analitik çözümü mümkün olmasına rağmen matematiksel modelin verebileceği sonuçlar dışında farklı sonuçlarla ilgileniliyorsa,
5) Sistemin davranış analizi yapılacaksa, benzetim kullanılır.

## Benzetim ve Modelleme Ne Zaman İyi Bir Fikir Değildir?

1) Problem sağduyulu bir analiz ile çözülebiliyorsa
Problem analitik olarak çözülebiliyorsa
Gerçek sistem üzerinde değişiklik ve deney yapmak daha kolaysa
Simülasyon Maliyeti sağlanacak kazancın üzerinde ise,
Proje için yeterli kaynak mevcut değilse
Model sonuçlarından faydalanmaya yetecek süre yoksa,
Gerekli veriler (hatta tahmin bile) yoksa
Modelin doğrulanması ve sağlaması yapılamıyorsa
Projeden beklentiler karşılanabilir düzeyde değilse
Sistem davranışı çok karmaşık ise veya sistem modelle-

nebilir değilse benzetim iyi bir fikir değildir

Benzetim ve Modelleme Avantajları

1) Benzetim modeli kurulduktan sonra, önerilen yeni tasarımların veya yeni politikaların analizinde kullanılabilir.
2) Yeni bir sistem analizine yardımcı olmak için kullanılır.
3) Benzetim modelinden veri elde etmek, gerçek sistemden aynı verileri elde etmekten daha ucuzdur.
4) Benzetim tekniği, analitik metotları uygulamaktan daha kolaydır.
5) Analitik modellerde çözüme ulaşabilmek için birçok basitleştirici kabullerin yapılması gerekirken, benzetim modellerinde böyle bir kısıtlama yoktur.
6) Analitik modeller ile kısıtlı sayıda performans ölçütleri hesaplanabilir. Benzetim modelleri ile akla gelebilen herhangi bir performas ölçütü tahmin edilebilir.
7) Bazı durumlarda, benzetim, bir çözümün elde edilmesi için tek araçtır.

Benzetim ve Modelleme Dezavantajları

1) Benzetim modellerinin kurulması ve geçerliliğinin araştırılması, zaman alıcıdır. Bu nedenle bilgisayarlarda benzetim modellerinin koşum maliyeti yüksek olabilir.
2) Maliyeti etkileyen diğer bir faktör benzetim modellerinin birden fazla (n kez) çalıştırılması ihtiyacıdır. Bu durumda bilgisayar maliyeti artmaktadır.
3) Benzetimin analitik tekniklerin yeterli olabileceği durumlarda da zaman zaman kullanıldığı gözlenmektedir.
4) Genel olarak, tüm benzetim modelleri "girdi-çıktı" modelleri olarak adlandırılır. Verilen bir girdi seti için sistemin çıktısını elde ederler. Yani, benzetim modelleri matematiksel modellerde olduğu gibi çözülmezler, çalıştırılırlar.
5) Belirli koşullar altında sistemin tavrını incelemek için kullanılırlar.

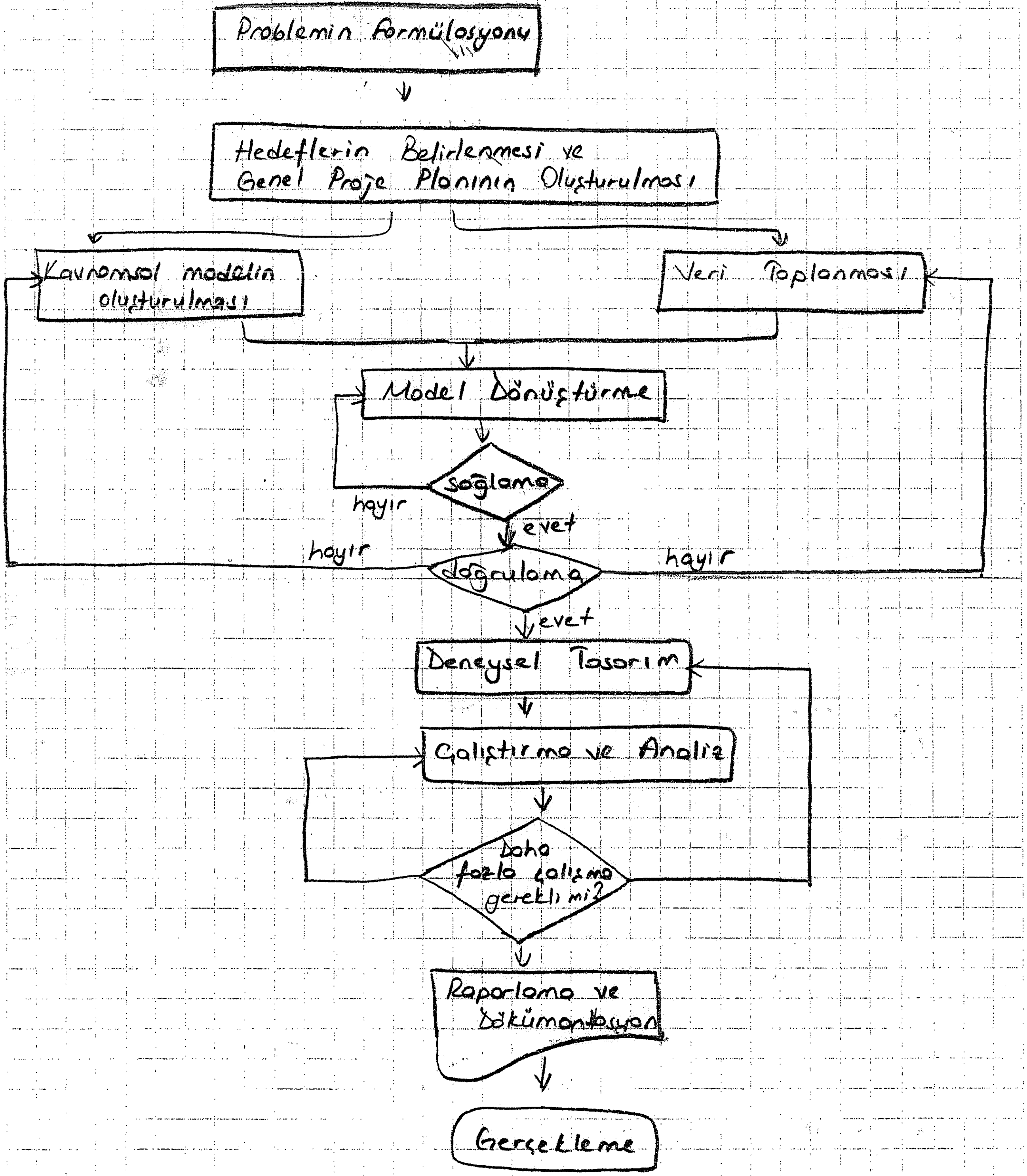
# Bu nedenle; analitik modellerin aksine en iyi çözümü üretmezler. #

Benzetim ve Modellemenin Genel Olarak Uygulama Alanları

- Bilgisayar Sistemleri
- Üretim
- İşletme
- Kamu Hizmeti
- Ekoloji ve çevre
- Sosyoloji
- Biyoloji

Sistem: Bir amacı gerçekleştirmek için aralarında düzenli bir etkileşimin veya bağımlılığın bulunduğu nesneler topluluğudur.

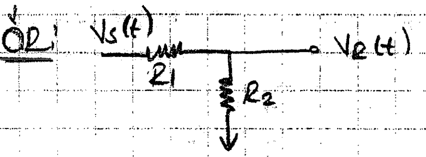
Benzetim Çalışmasının Adımları




# Bir sistem, kendisi dışında ortaya çıkan değişikliklerden etkilenir. Sistemlerin modellerinin kurulabilmesi için, sistem ve sistemin çevresi arasındaki sınıra karar vermek gerekir. Bu karar, sistemin özelliğine ve çalışmanın amacına bağlıdır.

# Geri besleme sistemi dinamikleştirir.

ÖR:




$x(t) = V_s(t)$ giriş

$z(t) = V_R(t)$ çıkış

↳ Tek girişli elektrik devresi sistemi (dinamik değil)

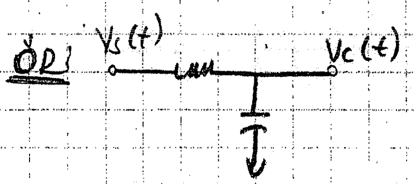
$$z(t) = \frac{R_2}{R_1 + R_2} \cdot x(t)$$

Çıkış ve durum denklemlerini tanımlamak için durum değişkeni tanımlanmalıdır. Durum değişkeni akım olarak belirlenirse;

$y(t) = i(t)$

durum denklemi $y(t) = \dfrac{x(t)}{R_1 + R_2}$ $\qquad z(t) = R_2 \cdot y(t)$

ÖR:




Kondansatör enerji depolayan bir eleman olduğundan sistem dinamiktir.

$x(t) = V_s(t)$ giriş $\qquad y(t) = V_c(t)$ çıkış

Birinci dereceden tek girişli ve tek girişli RC devresi

Temel fizik kanunlarından;

$$RC\,(\underbrace{dV_c/dt}_{\dot{y}(t)}) + \underbrace{V_c}_{y(t)} = \underbrace{V_s}_{x(t)}$$

$$\dot{y}(t) = \frac{1}{RC} \cdot [x(t) - y(t)]$$

$$z(t) = y(t)$$

(#) Sistem uygulamalarından en popüleri kontrol sistemleridir.

## Ayrık Zamanlı Sistemler

Zaman-sürümlü modeller girişi bütün zaman dilimlerinde değişen sistemlerdir. Bu özel durumda $t$ zamanı süreklidir. Çünkü çıkış diferansiyel denklem ile gösterilir. Eğer sistem olayları sadece belirli zamanlarda oluşuyorsa ayrık zamanlı bir sistemden bahsedilir.

Bu tür sistemde $t_k = t_0 + hk$ for $k = 0, 1, \ldots$ ile ifade edilir. $t_k$, $t_0$ başlangıç zamanı ile başlar ve sistem sinyali $h$ birim zamanına kadar değişmez.

$t_{k+1} - t_k = h \rightarrow$ örnekleme aralığının adım büyüklüğü

# Eğer sistem olayları belirli bir zamanda oluşuyorsa sistem ayrıktır

**Örnek:** Sürekli zamanlı bir $x(t)=\cos(\pi t)$ sinyalinin $t_k = 3+(1/2)k$ ayrık zamanlarda örneklendiğini düşünelim. Burada örnekleme uzunluğu $h=(1/2)$ olup ve başlangıç zamanı $t_0=3$ olsun.

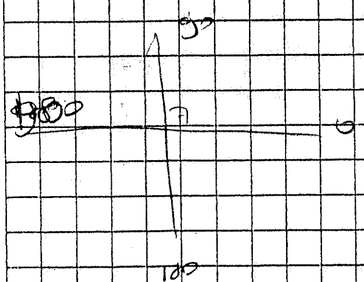
$$x(t_k) = \cos\left[\pi\left(3+\tfrac{1}{2}k\right)\right]$$

$$= -\cos\left(\tfrac{1}{2}\pi k\right)$$

$$= \begin{cases} 0, & k = \text{tek} \\ (-1)^{(k+2)/2}, & k = \text{çift} \end{cases}$$

Dolayısıyla $x(k)$ aşağıdaki gibi yazılır.

$$x(k) = \begin{cases} 0, & k = \text{tek} \\ (-1)^{(k+2)/2}, & k = \text{çift} \end{cases}$$



**Örnek:** Her 10 sn'de bir bir kutunun ulaştığı bir fabrika taşıyıcı sistemi düşünelim. Kutuların ağırlıkları sırasıyla 5, 10, 15 kg'dır. Fakat 5kg ve 15kg kutuların gelme olasılığı 10kg'ın gelme olasılığından iki kat fazladır. Bu sistem nasıl modellenir ve benzetimi nasıl yapılır?

**Çözüm:**

$W = \{5, 10, 15\}$ ağırlık değişkeni

$Pr(W=w)$ w olma olasılığı

| $w$ | $Pr[W=w]$ Olasılık |
|---|---|
| 5 | 0.4 |
| 10 | 0.2 |
| 15 | 0.4 |
| | 1.0 |

Her 10sn'de bir kutu geldiğine göre $t=10k$ olarak alınır.

```
for k=1 to n
    r=10*RND
    if r<4 then w(k)=5            (0,1,2,3)
    if 4<=r<6 then w(k)=10        (4,5)
    if r>=6 then w(k)=15          (6,7,8,9)
next k
```

**Örnek:** İki taşıyıcıya sahip bir fabrika sistemi düşünelim. Bir önceki örnekte olduğu gibi kutuları almaktadır. Fakat bu kez ulaşan kutular sadece 3 kutu tutan kısa taşıyıcıya yerleştirilmektedir. Yeni gelen kutular düşen kutuların yerine gelmektedir. Bu durumda her bir kutunun ağırlığı yerine ilgili taşıyıcıdaki toplam kutu ile ilgilenirse. Bununla birlikte z(k) çıkışını belirlemek için önceki iki girişte hatırlanmalıdır. Bu sistemi karakterize edelim.

**Çözüm:**

```
for k=1 to n
    r = 10*RND
    if r<4 then x(k)=5          }
    if 4<=r<6 then x(k)=10      }  tek taşıyıcı için tanımlanmış
    if r>=6 then x(k)=15        }
    if k=1 then z(k)=x(1)
    if k=2 then z(k)=x(1)+x(2)
    if k>2 then z(k)=x(k)+x(k-1)+x(k-2)
next k
```

Sistem girişi rasgele değişken olduğundan çıkış da nondeterministik olmalıdır. Aynı zamanda taşıyıcı yüklendikten sonra üç noktadan ikisi bilinmelidir.

$$x(k) = \begin{cases} x(1) & k=1 \\ x(1)+x(2) & k=2 \\ x(k)+x(k-1)+x(k-2) & k>2 \end{cases}$$

İkinci dereceden fark denklemi. Bu 2 hafıza elemanına ihtiyaç duyulduğu anlamındadır.

Olay Sürümlü Modeller

Olay sürümlü bir modelde sistem düzensiz planlanmış olaylar oluşumunda hareketsiz bekleme durumunda kalır.

Zaman sürümlü modellerde zaman aralığı sabittir. Burada ise genellikle değişir ve non-deterministiktir. Bu yüzden olay-sürümlü benzetimler stokastik metotlara dayalıdır.

Burada $t_k = k$ olayın oluştuğu zamanı ifade eder.

Basit bir şekilde varış zamanları arası [0,1) aralığında rastgele dağılım gösterebilir. Yani $t_{k+1} - t_k = RND$

n adet rastgele dağılım için;

```
tk = 0
for i=1 to n
    tk = tk-1 + RND
next k
```

ÖR: Daha önceki taşıma sistemini göz önüne alalım. Bu kez kutular aynı fakat farklı zaman aralıklarında ulaşmaktadır. Amaç t zamanına kadar kaç kutu ulaşmıştır.

```
t0=0
for k=1 to n
    tk = tk-1 + RND
next k
for t=0 to tn step h
    for k=1 to n
        if tk-1 < t < tk then  N=K
    next k
    print t, N
next t
```

(#) Burada dikkat edilmesi gereken giriş zamanları sürekli, fakat olay zamanları ayrıktır.

Sistem Bileşenleri

1- • Varlık (Entity): Sistemle ilgilenen nesnedir.
Nesneler 3 temel sınıf altında toplanabilir: 1) insan
2) parça, evrak
3) elektronik mektup, proje...

2- • Özellik (Atribute): Bir nesnenin sahip olduğu özellik

3- • Faliyet (Activity): Belirli bir zaman dilimindeki bir işlemin tanımlanması

4- • Kaynaklar (Resources): Personel, alet, enerji, zaman, para

5- • Kontrol (Control): faaliyetlerin nerede, ne zaman, nasıl ortaya çıktığını planla

6- • Sistemin Durumu (System Statement): Çalışmanın amacına bağlı olarak herhangi bir anda sistemi tanımlamak için gerekli olan değişkenlerin toplamıdır.

7- • Olay (Event): herhangi bir anda ortaya çıkarak sistemin duru-

200

[redacted]

Quiz sorusuydu.

Bir banka sistemini düşünürsek;

| Sistem | Nesne | Özellik | Faaliyet |
|---|---|---|---|
| Banka | Müşteri | Hesap Kontrolü | mevduat hesabı açma |

| Olay | Durum Değişkeni |
|---|---|
| Varış, çıkış | meşgul vezne sayısı<br>Bekleyen müşteri sayısı<br>varış zamanı |

S
N
Ö
F
Ö
D

# BENZETİM ve MODELEME (2. hafta)

## 1. DİNAMİK SİSTEMLER

Dinamik model: Zamana bağlı modeldir. Zamanla ilgili değişken tutar.

### BAŞLANGIÇ DEĞER PROBLEMLERİ

Eğer başlangıç koşulları ile birleştirilmiş diferansiyel denklemler sistemi tanımlıyorsa başlangıç koşulları ile tanımlanan değişkenler sistem durum değişkenlerini oluşturur.

* Birinci dereceden diferansiyel denklemlerin her birinin çıkış değişkenleri, bir başlangıç durumları ile birlikte sistem durum değişkenlerinin bir kümesini oluşturur.

Birinci dereceden bir başlangıç değer problemi için;

$\frac{dx}{dt} = f(t, x)$ $\quad x(t_0) = x_0$ ise;

$x(t) = [x_1(t), x_2(t), \ldots x_n(t)] \rightarrow$ Sistem durum vektörü
$x(0) = [x_1(0), x_2(0), \ldots x_n(0)] \rightarrow$ İlgili başlangıç durumları

**ÖR!** $\ddot{\alpha} + 2\beta\dot{\alpha} + \beta^2\alpha = \cos t$ $\quad$ Başlangıç durumları:
$\dot{\beta} + \alpha \cdot \beta = 4$ $\quad$ $\alpha(0) = 2 \quad \alpha'(0) = -1 \quad \beta(0) = 1$

İki dinamik durum değişkeni ve birinci ve ikinci dereceden diferansiyel denklemler olduğu için; üçüncü dereceden bir sistemdir. Bu yüzden 3 tane birinci dereceden diferansiyel denklem sistemi olarak tekrar tanımlamak mümkündür.

$x_1(t) = \alpha(t) \quad x_2(t) = \alpha'(t) \quad x_3(t) = \beta(t) \Rightarrow$ tanımlanan bu denklemler denklem sisteminde yerine yazılırsa;

$$\dot{x}_2 + 2x_2 \cdot x_3 + x_3^2 \cdot x_1 = \cos t$$
$$\dot{x}_3 + x_1 \cdot x_3 = 4$$

$x_2$, $x_1$'in türevidir. Denklem buna göre tekrar düzenlenirse;

$$\dot{x}_1 = x_2 \qquad x_1(0) = 2$$
$$\dot{x}_2 = -2x_2 \cdot x_3 - x_3^2 \cdot x_1 + \cos t \qquad x_2(0) = -1$$
$$\dot{x}_3 = -x_1 \cdot x_3 + 4 \qquad x_3(0) = 1$$

3 birleşik durum vektörü $x = [x_1, x_2, x_3]$ olarak tanımlanırsa;

$$f = [x_2, -2x_2 \cdot x_3 - x_3^2 \cdot x_1 + \cos t, -x_1 \cdot x_3 + 4]$$
$$x_0 = [2, -1, 1]$$
$$t_0 = 0$$

Çok sayıda sistemi bu teknikle genelleştirmek mümkündür. Sistemler bu şekle getirilip matlabla çözülebilir.

$\rightarrow x(k+1) = \cancel{x} + h^{*}x^{k}$

## EULER YÖNTEMİ

Türevin tanımından;

(3) $\lim \dfrac{x(t+h)-x(t)}{h} = f(t,x) \qquad \left(f(t,x) = \dfrac{dx}{dt}\right)$

(1) $x(t+h) = x(t) + h.f(t,x) \Rightarrow$ sürekli zamanlı

(2) $x(k+1) = x(k) + h.f[t(k), x(k)] \Rightarrow$ ayrık zamanlı $(k=0,1,2,\ldots,n)$

Zaman değişkeni t ise; sinyal sürekli ya da analog olarak alınmıştır. Zaman ayrık ise; durum değişkeni $x(k)$'dır. Sürekli zamanın ayrık zamanla yer değiştirdiği bu işleme "ayrıklaştırma" adı verilir. Buna göre yeterince küçük h adım uzunluğu için $x(k) = x(tk) = x(t)$'dir.

Denklem (2)'yi kullanarak denklem (1)'i çözmek denklem (3)'ün yaklaşık çözümü olacaktır.

* Denklem (3)'ün sağ tarafında değişkenler k zamanda olmasına karşın, sol taraftakiler $k+1$ zamandadır. Bu durum recursive x değişkeninin bir güncellemesi olarak ifade edilir. Denklemin sağ tarafındaki değişkenler "eski", sol tarafındaki değişkenler "yeni" olarak adlandırılır. Bu teknik Euler Yöntemi'dir.

**ÖR:** $\dot{x} = x^2 t \qquad x(1) = 3$ şeklinde tanımlanan bir sistem için;

t'nin ilk değeri — x'in ilk değeri

Bu sistem temel tekniklerle şöyle çözülür.

$$\frac{dx}{dt} = x^2 t \quad \text{ve} \quad x(1) = 3 \rightarrow \frac{dx}{x^2} = t\,dt$$

$$\int_3^x \frac{dx}{x^2} = \int_1^t t\,dt = -\frac{1}{x}\Big|_3^x = \frac{t^2}{2}\Big|_1^t$$

$$= -\frac{1}{x} + \frac{1}{3} = \frac{t^2}{2} - \frac{1}{2} \Rightarrow \boxed{x(t) = \frac{6}{5-3t^2}} \rightarrow \text{gerçek çözüm}$$

Açık çözümün bilinmediğini farz ederek Euler yöntemini çıkarabiliriz. Keyfi verilmiş $h=0{,}05$ adım uzunluğu ile eşit ayrık sistem başlangıç koşulları ile karakterize edilmiştir.

$t(0) = 1, \quad x(0) = 3$

ve fark denklemleri $t_{k+1} = t_k + 1/20$

$x(k+1) = x(k) + 1/20 \cdot x^2(k) \cdot t_k \quad (k=1,2,\ldots n)$

? Dikkat edilirse t'nin eski değeri x güncellemesine ihtiyaç duyar fakat x, t'nin güncellemesine ihtiyaç duymaz.

Sözde kod;

```
t=1   x=3   print t,x
for k=1 to n
  x = x+hx^2t
  t = t+h
  print t,x   next k
```

Burada recursive bir yöntem vardır.

Euler Yöntemi Matlab Kodu,

```
t0=1;    % başlangıç zamanı
x0=3;    %   "   koşulu
x=[x0];     t=[t0];
xg=[6/(5-3*t0^2)]    % temel yöntemle bulunan mutlak çözüm
h=0,05
for i=1:5
        xyeni=x0+h*x0^2*t0;
        t0=t0+h;
        xg=[xg 6/(5-3*t0^2)];
        x=[x xyeni]
        t=[t  t0]
        x0=xyeni;
end
plot (t,x,'r-d')
hold on
plot (t,xg,'k-s')
```

# Euler yönteminde doğru çözüm elde etmenin yolu h adım uzunluğunun azaltılmasıdır. Ancak h değerinin azaltılması iki büyük engele sahiptir;

1) verilen bir noktadaki çözümü tahmin etmek için daha fazla hesaplama gerekecektir.

2) Veri gösteriminde bir sonraki makine sınırlamalarından dolayı h çok fazla küçük olabilir.

# Her hesaplamanın sonucunda bir prosedürün sonuçlarının çıktısını almaktan ziyade, hesapların periyodik sonuç çıktılarının alınması daha yararlıdır. Bu, "kontrol break" adı verilen bir işlem uygulanarak gerçekleştirilir. Bir kontrol break iş ise döngüler yoluyla çalışır.

```
t=1
x=3
print t,x
for i=1 to n
    for j=1 to m
        x=x+hx²t
        t=t+h
    next j
    print t,x
next i
```

Dış döngü i (i=1,2,...,n) indeksi kullanarak kontrol eder. iç döngü j (j=1,2,...,m) indeksi kullanır.

1. dereceden denkleme → EULER
2. dereceden denkleme → TAYLOR
4. dereceden denkleme → RUNGE KUTTA

## TAYLOR YÖNTEMİ

Euler yöntemi, Taylor teoreminin özel bir durumu olarak düşünülebilir.

$$x(t+h) = \sum_{i=0}^{\infty} \frac{x^{\wedge} i(t)}{i!} h^{\wedge} i$$

$\dot{x} = f(t,x)$ ve $x(t+h) = x(t) + h.f(t,x)$; Bu formül daha yüksek mertebeli yaklaşım kullanılarak genişletilebilir. Mesela ikinci dereceden yaklaşım şu formülü verir;

$$\underbrace{x(t+h) \approx x(t) + h\dot{x}(t) + \frac{1}{2}.h^2.\ddot{x}(t)}_{h\left(\dot{x}(t) + \frac{1}{2}.h.\ddot{x}(t)\right)}$$

→ x'in türevi h'in integrali

$$\ddot{x} = \frac{d}{dt} f(t,x) = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial x}\frac{dx}{dt} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial x} f(t,x)$$

Buna göre;

$$x(t+h) = x(t) + h.f(t,x) + \frac{1}{2}.h^2\left[\frac{\partial f}{\partial t} + \frac{\partial f}{\partial x} f(t,x)\right]$$

bu formul aşağıdaki gibi euler ayrıklaştırılabilir:

$$t_{k+1} = t_k + h$$

$$x(k+1) = x(k) + h\underset{\dot{x}(t)}{f} + \frac{1}{2}.h^2\underset{\ddot{x}(t)}{\left(\frac{\partial f}{\partial t} + \frac{\partial f}{\partial x}.f\right)}$$

**ÖR!** $\dot{x} = x^2 t$ ve $x(1) = 3$ sistemine Taylor tekniğini uygulayınız.

$f(t,x) = \dot{x} = x^2 t$ $\qquad \frac{\partial f}{\partial t} = x^2 \qquad \frac{\partial f}{\partial x} = 2xt$

$$x(t+h) = x(t) + h.f(t,x) + \frac{1}{2}.h^2\left[\frac{\partial f}{\partial t} + \frac{\partial f}{\partial x} f(t,x)\right]$$

$$= x(t) + h.x^2(t).t + \frac{1}{2}h^2\left[x^2(t) + 2x(t).t.x^2(t).t\right]$$

$$= x(t) + hx^2(t)t + \frac{1}{2}.h^2\left[x^2(t) + 2x^3(t).t^2\right]$$

ayrıklaştırırsak;

$$x(k+1) = x(k) + x^2(k).t_k.h + x^2(k)\left[x(k)t_k^2 + 1/2\right].h^2$$

# Taylor, Euler yöntemine göre daha doğru sonuçlar elde eder.

# RUNGE - KUTTA YÖNTEMİ

Daha önceki örneklerde;
birinci dereceden denklemlere Euler;
ikinci " " Taylor uygulandı.
Dördüncü dereceden bir yaklaşım sıklıkla şu şekilde kullanılır;

$$x(t+h) = x(t) + h\dot{x}(t) + \frac{1}{2}h^2\ddot{x}(t) + \frac{1}{6}h^3 x^{(3)}(t) + \frac{1}{24}h^4 x^{(4)}(t)$$

↳ Bu formulü uygulamak için, Taylor için yapıldığı gibi ilk önce $f$'i birkaç kez ayırmak gereklidir.
$f$ için bir analitik formul kullanılamıyor olabileceği için bu çoğu zaman imkansızdır ya da en iyi ihtimalle sıkıcıdır.

ÖR: Klasik bir Runge - Kutta algoritması şöyledir.

$$K1 = f(tk, x(k)) \qquad K3 = f\left(tk + \frac{1}{2}h, x(k) + \frac{1}{2}hK_2\right)$$

$$K2 = f\left(tk + \frac{1}{2}h, x(k) + \frac{1}{2}hK_1\right) \qquad K4 = f(tk + h, x(k) + hK_3)$$

$$tk+1 = tk + h$$

$$x(k+1) = x(k) + \frac{1}{6}\cdot h\,(K1 + 2K2 + 2K3 + K4)$$

# Klasik Runge - Kutta algoritmasının iki avantajı vardır; doğru çözümdür (dördüncü dereceden Taylor'a eşdeğerdir) ve türev hesaplanması gerekmediğinden kullanımı kolaydır.

ÖR) $\dot{x} = x^2 t$ ve $x(1) = 3$ sistemine Runge-Kutta tekniğini uygulayınız.

! K1, K2'den; K2, K3'den; K3, K4'ten; K4, x'ten önce güncellenmelidir.

Runge - Kutta Yöntemi Matlab Kodu

```
t0=1;      % başlangıç zamanı
x0=3;      %    "    koşulu
x=[x0];    t=[t0];
xg = [6/(5-3*t0^2)]     % mutlak çözüm
h=0,05;
for  i=1:5
    K1= t0*x0^2;                                % K1 = t x^2
    K2 =(t0+(1/2)*h)*[x0+(1/2)*h*K1]^2;         % K2=(t+1/2h)(x+1/2hK1)^2
    K3= (t0+(1/2*h)*[x0+(1/2)*h*K2]^2;          % K3=(t+1/2h).(x+1/2hK2)^2
    K4 = (t0+h)*[x0+h*K3]^2;                    % K4=(t+h).(x+hK3)^2
    xyeni = x0+(1/6)*h*(K1+2*K2+2*K3+K4);
    t0=t0+h;
    xg=[xg  6/(5-3*t0^2)];
    x=[x  xyeni];     t=[t  t0];
    x0=xyeni;
end
plot (t,x,'r-d')
hold on
plot (t,xg,'k-s')
```

## 2. YÜKSEK DERECEDEN SİSTEMLER

İki tane birinci dereceden denklem sistemi şöyle tanımlanabilir;

$$\dot{x} = f(t,x,y)$$
$$\dot{y} = g(t,x,y)$$
$$x(t0) = x_0$$
$$y(t_0) = y_0$$

Daha önce anlatıldığı gibi x değişkenini K1, K2, K3, K4 değerleriyle y değişkenini $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ değerleriyle ilişkilendirerek bu sisteme Runge-Kutta uygulamak basittir.

Ayrık çözüm denklemi şöyle elde edilir;

$$K1 = f(t_k, x(k), y(k)),$$
$$\lambda_1 = g(t_k, x(k), y(k)),$$
$$K2 = f(t_k + 1/2h, x(k) + 1/2\,hK1, y(k) + 1/2\,h\lambda_1),$$
$$\lambda_2 = g(t_k + 1/2, x(k) + 1/2\,hK1, y(k) + 1/2\,h\lambda_1),$$
$$K3 = f(t_k + 1/2h, x(k) + 1/2\,hK2, y(k) + 1/2\,h\lambda_2),$$
$$\lambda_3 = g(t_k + 1/2h, x(k) + 1/2\,K2, y(k) + 1/2\,h\lambda_2),$$
$$K4 = f(t_k + h, x(k) + hK3, y(k) + h\lambda_3)$$
$$\lambda_4 = g(t_k + h, x(k) + hK3, y(k) + h\lambda_3)$$

$$x(k+1) = x(k) + 1/6\,h(K1 + 2K2 + 2K3 + K4)$$
$$y(k+1) = y(k) + 1/6\,h(\lambda_1 + 2\lambda_2 + 2\lambda_3 + \lambda_4)$$

**ÖD1** $\ddot{x} + 3x\dot{x} = u(t) \quad u(t) = t, \quad t \geq 0 \quad x(0) = 2 \quad \dot{x}(0) = 1$

$u(t) \longrightarrow$ [sistem] $\longrightarrow x(t)$

$\dot{x} = y$

$\dot{y} = t - 3xy \qquad \dot{x} = y$

İkinci dereceden bir diferansiyel denklem olduğu için iki durum değişkeni olmalıdır. Bunlardan birini x(t) çıkışı olarak, diğerini $y(t) = \dot{x}(t)$ olarak tanımlarız. Böylece;

$\dot{x}(t) = y, \quad \dot{y} = t - 3xy \quad x(0) = 0 \quad y(0) = 1$ olur.

Euler yöntemi kullanılarak ayrıklaştırılmış sistem;

$$t_{k+1} = t_k + h$$
$$x(k+1) = x(k) + h\,y(k)$$
$$y(k+1) = y(k) + h[t_k - 3x(k)\,y(k)]$$

Sistemin Euler çözümü;

$y = x'(t)$

$y' = t - 3xy$

```
t=0
x=2
y=1
print t,x,y
for k=1 to n
    x1 = x + hy
    y1 = y + h (t-3xy)
    x = x1
    y = y1
    t = t+h
    print t,x,y
next k
```

Sistemin Runge-Kutta çözümü;

```
t=0
x=2
y=1
print t,x,y
for k=1 to n
  k1=y          -> x'   -> y
  λ1= t-3xy     x''  -> y'
  k2= y+1/2h λ1
  λ2 =t+1/2h - 3(x+1/2 h k1) (y+1/2hλ2)
  k3= y + 1/2 h λ2
  λ3= t+ 1/2h - 3(x+1/2hk2) y+1/2hλ3)
  k4= y+hλ3
  λ4= t +h- 3(x+hk3).(y+hλ3)
  x= x+ 1/6 h (k1+ 2k2+ 2k3 + k4)
  y =y+ 1/6 h (λ1+2λ2+2λ3+λ4)
  t= t+h
  print t,x,y
next k
```

k1
λ1
k2
λ2

## 3. OTONOM DİNAMİK SİSTEMLER

Sistem
- Girişi Olan
  - Zaman odaklı
  - Olay tabanlı
- Girişi Olmayan (OTONOM) → lineerse 3 tip doğal tepki olur.
  - 1. Kararlı
  - 2. Kararsız
  - 3. Marjinal

**Zaman odaklı:** Girişler eş zamanlı

Örneğin; $\ddot{x}+2\zeta\omega \dot{x}+\omega^2 x=r(t)$

$r(t)$ girişine, $x(t)$ çıkışına, $\zeta$ ve $\omega$ sabitlerine sahiptir. $r(t)=4-t$ gibi bir deterministik zaman fonksiyonu ise sistem zaman odaklıdır.

**Olay tabanlı:** Girişler eş zamanlı değil

Örneğin, aynı denklem için $r(t)$;

$$r(t)=\begin{cases} 1, & \text{event occors} \\ 0, & \text{event does not occor.} \end{cases}$$

olursa sistem olay odaklıdır.

**Kararlı;** kısa bir geçiş evresinden sonra çıktı sıfıra doğru yaklaşır.

**Kararsız;** sınırlama olmaksızın doğal tepki artar.

**Marjinal;** tepki periyodik ve sınırlıdır.

# Otonom olmayan sistem durumunda süperpozisyon prensibi uygulanır → bilmeye gerek yok?

Süperpozisyon:

# $r(t)$'nin giriş olduğu sabit katsayıları olan ikinci dereceden bir denklem sistemi düşünelim: $\ddot{x}+2\zeta\omega\dot{x}+\omega^2 x=r(t)$. Doğal tepki durumunda $r(t)=0$dır ve $\zeta$ ve $\omega$ sabitlerine bağlı çözümün üç tipten biri olduğu gösterilebilir.

Özellikle;

$$x(t)=\begin{cases} A.e^{-\cos t}.\cos(\omega\sqrt{1-} & |\zeta|<1 \\ A.\cos(\omega t)+B\sin(\omega t) & |\zeta|=1 \\ A.e^{\zeta\omega t}+B.e^{-\zeta\omega t} & |\zeta|>1 \end{cases}$$

**Ör:** $\ddot{x} + 4\dot{x} + 5x = 15t + 22$   $x(0)=0$   $\dot{x}(0)=1$

sistemi süper pozisyon kullanarak çözün. Bunu yaparken (1) başlangıç koşullarını içeren fakat girişten bağımsız bir terim ve (2) giriş içeren fakat başlangıç koşullarından bağımsız bir terim şeklinde iki terimin toplamının toplam çözüm olduğunu gösteriniz.

Önceki denkleme benzetirsek;

$\ddot{x} + 2\zeta w \dot{x} + w^2 x = r(t)$   $r(t) = 15t + 22$   doğal tepki verilen denklemin çözümüdür.

$x(0)=0$   $x'(0)=1$

$w = \sqrt{5} \Rightarrow 2\zeta w = 4$

$\zeta = \frac{2}{\sqrt{5}}$

$x_{natural}(t) = A.e^{-2t}.\cos t + B.e^{-2t}.\sin t$

Giriş doğrusal ve türevi sabit olduğu için $x_{forced} = C + Dt$ olacaktır. Denklem $(\ddot{x} + 4\dot{x} + 5x = 15t + 22)$ ve katsayıları yerine koyduğumuzda

$x_{forced} = 3t + 2$ olur.

Süper pozisyonu kullanarak toplam çözüm;

$x(t) = 2.e^{-2t}.\cos t - e^{-2t}.\sin t + 3t + 2$ olarak elde edilir. (A ve B sabitleri başlangıç şartları yerine koyularak bulunmuştur.)

Genel olarak kararlı doğrusal sistemlerde;

$x(t) = x_{natural}(t) + x_{forced}(t)$

Burada $x_{natural}(t)$ yalnızca başlangıç koşullarına, $x_{forced}(t)$ ise yalnızca girişe bağlıdır.

# Belirli bir türün popülasyon dinamiğini modelleyen bir sistem düşünelim. Bu popülasyonu kontrol eden çok sayıda faktör olmasına rağmen bir popülasyon büyüklüğünün artma oranı, herhangi bir zamandaki popülasyon büyüklüğü ile kabaca orantılıdır.

$x(t)$, $t$ zamanındaki popülasyon büyüklüğünü ve $\lambda$ sabit orantıyı gösterir. $\dot{x} = \lambda.x$   $t=0$ zamanındaki başlangıç büyüklüğünü $x_0$ olarak kabul edersek $t \geq 0$ için $\boxed{x(t) = x_0.e^{\lambda t}}$ 'dir. ?

$\boxed{x(t) = x_0.e^{\lambda t}}$ → popülasyonun zaman içinde katlanarak artacağı direk görülür. Bu model Malthusian modeli olarak adlandırılır.

- Popülasyon büyüklüğüne ek olarak çevre kapasitesi, mücadele ve enerji gibi önemli faktörler vardır. Örneğin; bir popülasyon kendisine yiyecek temini için çok büyük olabilir veya yaşadığı bölgede avcılar olabilir.

* Sistemin desteklediği maksimum popülasyon büyüklüğü $X_m$ taşıma kapasitesidir. $x(t)/X_m$ sistemin doluluk oranıdır ve $1 - x(t)/X_m$ büyüme için geriye kalan kullanılabilir sistem oranıdır.

ezbere ← $\boxed{\dot{x} = \lambda.x\left(1 - \frac{x}{X_m}\right)}$ ⇒ Bu denklem "Lojistik denklem" olarak adlandırılır ve doğrusal değildir.

$x(t)$ küçük ise $(1 - x/X_m) \approx 1$ olduğundan denklem Malthusiandır.
$x(t)$ büyük ise $(1 - x/X_m) \approx 0$ olduğundan büyüme oranı düşer.

? Grafikler nasıl ?

$$\dot{x} = \lambda x \Rightarrow x = e^{\lambda}$$

temel matematik ile edilir.

benzer şekilde $\dot{x} = \lambda . x \left(1 - \frac{x}{x_m}\right) \Rightarrow x = \frac{x_m}{1-\left(1-\frac{x_m}{x_0}\right).e^{-\lambda t}}$ elde edilir.

* Lojistik model sınırlı yiyecek kaynağının ve bir popülasyonun olduğu ortamlarda iyi bir performans sağlar.
Lojistik sistemler başlangıç şartından bağımsız olarak belli bir noktaya kadar artar.

Bununla birlikte, yiyerek ve yenilmeyerek hayatta kalan av ve bir avcının var olduğu ortam daha tipik durumlardan biridir.

$x(t) \to$ av popülasyonun büyüklüğü
$y(t) \to$ avcı " " nü göstersin
av-avcı etkileşim sayısı $x(t).y(t)$
av-avcı popülasyonunun büyüme oranı;

$$\boxed{\dot{x} = \alpha_1 . x . \left(1 - \frac{y}{\beta_1}\right)} \qquad \boxed{\dot{y} = \alpha_2 . y \left(1 - \frac{x}{\beta_2}\right)}$$

$\alpha_1, \alpha_2, \beta_1, \beta_2$ : her popülasyon için pozitif kapasite sabitleridir.

Bu denklemler Lotka-Volterra denklemleri olarak adlandırılır, doğrusal değillerdir ve bilinen analitik bir çözümü yoktur. Bu yüzden daha önce anlatılan sayısal teknikler kullanılabilir.

* Lotka-Volterra modelinin benzetimi için; ~~Euler~~ yöntemi uygulanabilir. İki denklem çifti olduğu için geçici değişkenler kullanılır, değişkenler hesaplandıktan sonra güncellenir.

```
read t,x,y
print t,x,y
for k=1 to n
    x1=x(1+α1h-α1hy/β1)
    y1=y(1-α2h-α2hx/β2)
    x=x1
    y=y1
    t=t+h
    print t,x,y
next k
```

Sonuçların $[t_0, t_n]$ zaman aralığında ve h aralığı olduğunu düşünürsek $n=(t_n-t_0)/h$ oldukça büyük olur. Örneğin; $t_0=0$ $t_n=10$ $h=0{,}001$ ise $n=10\,000$ örnek gerekir. Muhtemelen sadece 50 tanesi yeterli olacaktır.

* Bu problem bir kontrol break ile çözülebilir. Böylece $[0,t_n]$ zaman aralığında m×n hesaplama yapılır. Adım boyutu h olmak üzere; $m=(t_n-t_0)/nh$ olarak seçilir.

```
read t,x,y   print t,x,y
for j=1 to n
    for i=1 to m
        x1=x (1
        y1=
        x=x1     y=y1     t=t+h
    next j
    print t,x,y
next i
```

ÖR: $\dot{x} = 3x\,(1 - 1/10\; y)$   $x(0) = 10$

$\dot{y} = 1{,}2\,y\,(-1 + 1/23\; x)$   $y = 5,$

verilen Lotka-Volterra sistemini $[0,5]$ aralığında çözün.

h=0,001 alalım. Grafik 5 birim zaman kadar olacak ve her bir aralıkta 10 örnek yeterlidir. Dolayısıyla 50 örnek iyidir;
$m = (5-0)/(50 * 0{,}001) = 100$ güncelleme olur. Toplamda 5000 güncelleme olur.

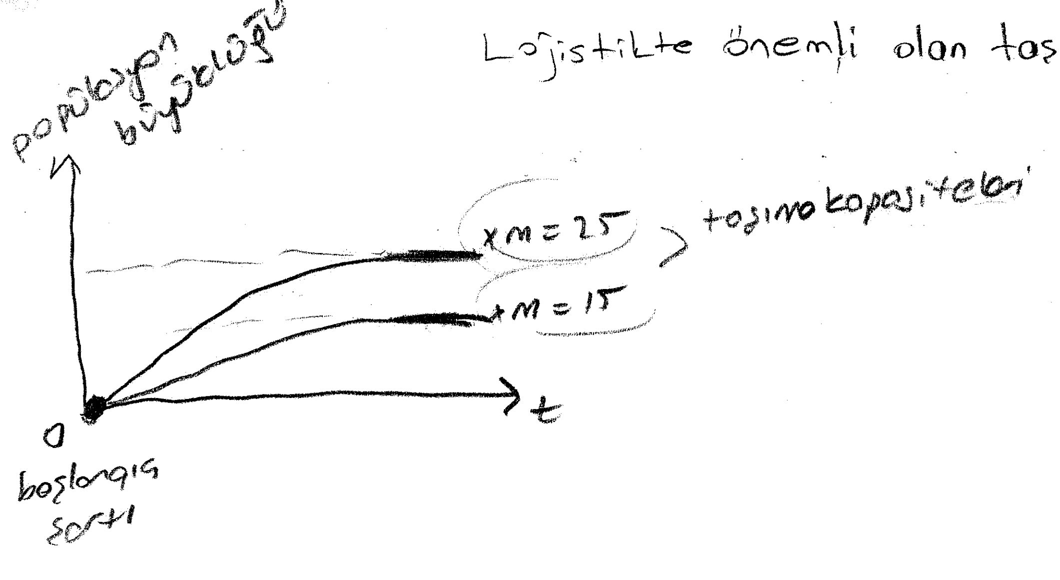
?

```
function [t,x,y]
x=[x0]
y=[y0]    h=0.001;  kontrol
t=0;                break
for i=1 to n
    for j=1:100
    xx = x0 + (1+3*h-3*h*y0/10);
    yy = y0 + (1-1,2*h-1,2*h*x0/25);
    x0 = xx;
    y0 = yy;
    x = [x ; xx];
    y = [y  yy];
    t = t+h;
    end
end
plot (x)
hold on
plot (y,'r--')
```

! Avcının yiyecek kaynağı arttıkça avcı popülasyonu artmaktadır. Fakat avın harcanmasıyla avcı popülasyonu düşer. Dolayısıyla bir popülasyon bol bir besin kaynağına ulaştığında kendi başarısının kurbanı olur.

Lojistik sistemin, Lotka-Volterra'dan farkı;
- başlangıç şartlarından bağımsız olması
- belli bir noktaya kadar artması

Lojistikte önemli olan taşıma kapasitesi



## 4. ÇOKLU ZAMAN TABANLI SİSTEMLER

Dinamik sistemlerdeki önemli değişken zamandır. Bütün oranlar ve böylece bütün türevler zamana bağlı olarak değişir.

Bitkiler gerçek kronolojik zamandan ziyade ısı enerjisini (sıcaklık) miktarına göre büyür ve gelişirler. Bu ısı enerjisi fizyolojik zaman olarak düşünülür ve dinamik denklemler kronolojik zamandansa fizyolojik zamana göre oranları ölçerler.

Bazı sistemlerde her bir alt sistem zamanı farklı ölçer.

t zamanı içinde farklı T(t) sıcaklık değerleri gösteren bir ortamda belirli bakterilerin var olduğu biyolojik bir büyüme modelini düşünelim.

Belirli bir $T_0$ eşik sıcaklığı altında, büyümenin olmadığı gözlenmiştir. Fakat $T_0$'dan daha yüksek sıcaklıklar için bağlantılı büyüme oranı $\dot{x}/x$ $T(t)-T_0$ ile birlikte artmaktadır. Bu durum matematiksel olarak;

$$\frac{\dot{x}}{x} \begin{cases} 0 & T(t) < T_0 \\ r(t)[T(t)-T_0] & T(t) \geq T_0 \end{cases} \qquad (1)$$

Aşağıdaki gibi fizyolojik süre $\tau$ olarak adlandırılan yeni bir zaman belirlememiz gereklidir.

$$\tau(t) = \int (T(t) - T_0)\, dt$$

Burada $\tau(t) = \{ t \mid T(t) \geq T_0 \}$ sıcaklığının eşik değerini aşması durumudur. Bu tanımdan fizyolojik sürenin kümülatif bir sıcaklık birimi olduğunu açıkça görebiliriz. Böylece denklem (1)'i düzenleyerek $\tau(t)$ için günlük sıcaklık derecesi;

$$\frac{d\tau}{dt} = \begin{cases} 0 & T(t) < T_0 \\ T(t) - T_0 & T(t) \geq T_0 \end{cases}$$

$\frac{\dot{x}}{x} = r\frac{d\tau}{dt}$ olduğundan $x(t) = x_0 \cdot e^{rt}$ elde edilir.

Bu denklem, önceki kronolojik zaman için tanımlı üssel büyümedir ve fizyolojik zaman için de geçerlidir. Aslında lojistik ve Lotka Volterra modelleri de dahil olmak üzere daha önce gösterilmiş olan tüm büyüme modelleri aynı davranışı gösterir.

Uygun aktivite fonksiyonlarının olduğu tabloda, her bir büyüme modeli gösterilmiştir.

| Model ismi | Kronolojik zaman tanımı | Fizyolojik zaman tanımı |
|---|---|---|
| Malthusian | $\frac{\dot{x}}{x} = r\frac{d\tau}{dt}$ | $\frac{dx}{d\tau} = r\,x(t)$ |
| Lojistik | $\frac{\dot{x}}{x(1-x/x_m)} = r\frac{d\tau}{dt}$ | |
| Lotka-Volterra | | |

DENEYSEL VERİ İŞLEME

Modeller olayların idealleştirilmesi olmasına rağmen gerçekçi benzetimler oluşturmak için modele gerçek veriyi uygulamak istenir. (Örneğin; gerçek sıcaklık profili fiziksel ölçümlerden alınarak elde edilebilir.)

MERVE TELLEREN

# BENZETİM ve MODELLEME (3. hafta)

## Sistemin Performans Ölçütleri

**Çevrim zamanı:** Bir ürün üretilme zamanı

**Doluluk (kullanım) oranı:** Ekipmanın veya personelin üretken olduğu zaman yüzdesi

**Bekleme zamanı:** Bir müşterinin servis görebilmek için veya bir parçanın işlenebilmesi için kuyrukta geçirdiği ortalama zaman.

**Kalite:** Doğru özelliklere sahip ürün yüzdesi

**Maliyet:** Sistemin maliyeti

(#) Sistemler kesikli ve sürekli olarak ikiye ayrılır.

1) **Kesikli sistem:** Sistem durum değişkenleri, zamanın sadece kesitli noktalarında değişir. (Zaman eksenindeki bazı noktalar!)
Örneğin; banka

2) **Sürekli sistem:** Sistemin durum değişkenleri, zaman içinde sürekli olarak değişir. (Zaman eksenindeki tüm noktalar!)
Örneğin; bir uçağın havadaki hareketi

## Sistemlerin Çözümü ← sınav sorusu (5 puan)

Sistem
- Gerçek Sistem Üzerinde Deney
- Sistemin bir modeli ile Deney
  - Fiziksel Model → örn: prototip uçak
  - Matematiksel Model
    - Analitik Çözüm (hesaplamaları yapabiliyorsak kullanılır)
    - BENZETİM

## MODEL

Model; bir sistemin gösterimi olarak tanımlanabilir. Bir model gerçek sistem hakkındaki gerekli sonuçları çıkarmaya izin verecek detaya sahip olmalıdır.

Model
- Fiziksel Model (Fiziksel gerçek bir sisteme benzer (küçük ölçekli temsil))
- Matematiksel Model (Sistemi göstermek için sembolik notasyonlar ve matematiksel eşitlikler kullanılır)

(#) Benzetim Modelleri 3 ana grupta toplanabilir;
- Statik veya Dinamik
- Belirli veya Olasılıklı
- Kesikli veya Sürekli

**Statik Benzetim Modeli;** Sistemin belirli bir andaki gösterimidir. Monte-Carlo benzetim modelleri bu türe uygun modellerdir. Bu modeller, kesikli ve sürekli sistemlerin tanımlarına benzer şekilde tanımlanabilir. (Zamana bağlı olmadan çözüm yapıyorsak)

**Dinamik Benzetim Modeli;** Sistemin çalışma zamanına göre (bir aralık veya tüm çalışma zamanı dikkate alınarak) yapılan modeldir.

(Deterministik)
Belirli Benzetim Modeli; Rassal değişken içermeyen benzetim modelidir. Bu modellerde verilen GİRDİ seti için ÇIKTI seti vardır.
(Stokastik) Olasılıklı Benzetim Modeli; Bir veya birden fazla rassal değişken içeren benzetim modelidir. Stokastik (olasılıklı) benzetim modeli kullanılarak elde edilen çıktı rassal olup modelin karakteristiklerinin tahminidir.
Banka örneğinde, varışlar arası zaman aralığı ve servis zamanları rassal değişkenlerdir.
∘ Rastgele Model; İçerisinde hem kesikli, hem sürekli benzetim bulunan sistemler hibrit çalışan sistemlerdir.

## ZAMAN DİLİMLEME

Dinamik benzetimin temeli sistemin durum değişmelerinin zaman boyunca modellenmesidir. Benzetimde zaman akışının nasıl ele alınabileceğini göz önüne almak önemlidir.
Benzetimde zaman akışını kontrol etmenin en basit yolu eşit zaman aralıklarında ilerlemektir. (zaman dilimleme)
dt zaman dilimi uzunluğu için (t ile (t+dt)) aralığında ortaya çıkan değişimlere ilişkin model (t+dt) anında güncellenir.
dt büyük seçilirse; ortaya çıkan durum değişimlerinin bazılarının benzetimini yapmak olanaksız olur.
dt küçük seçilirse; model gereksiz yere sıkça incelenir ve bu aşırı bilgisayar çalışmalarına yol açar.

ÖR: A ve B gibi iki makinenin bulunduğu bir atölyeyi ele alalım.

| İş numarası | Yığın büyüklüğü | Beklenen sipariş günü |
|---|---|---|
| 1 | 200 | 1 |
| 2 | 400 | 8 |
| 3 | 100 | 14 |
| 4 | 200 | 18 |

İş süreleri; Makine A : (Yığın büyüklüğü/50)+1 gün
Makine B : (Yığın büyüklüğü/100)+3 gün

Her iş önce makine A'da yığın olarak bitirildikten sonra makine B'de yığın olarak başlar ve tamamlanır. (Varsayım)
Atölye şekilde görülen dört siparişi kabul ederse son yığın ne zaman tamamlanacaktır?

Beklenen iş süreleri;

| İş numarası | Makine A | Makine B |
|---|---|---|
| 1 | 5 | 5 |
| 2 | 9 | 7 |
| 3 | 3 | 4 |
| 4 | 5 | 5 |

Atölye zaman dilimi benzetimi;

| | Kuyruktaki işler | | İşlem gören işler | |
|---|---|---|---|---|
| Gün | A Mak | B Mak | A mak | B mak |
| 1 | - | - | 1 | - |
| 2 | - | - | 1 | - |
| 3 | - | - | 1 | - |
| 4 | - | - | 1 | - |
| 5 | - | - | 1 | - |
| 6 | - | - | - | 1 |
| 7 | - | - | - | 1 |
| 8 | - | - | 2 | 1 |
| 9 | - | - | 2 | 1 |
| 10 | - | - | 2 | 1 |
| 11 | - | - | 2 | - |
| 12 | - | - | 2 | - |
| 13 | - | - | 2 | - |
| 14 | 3 | - | 2 | - |
| 15 | 3 | - | 2 | - |
| 16 | 3 | - | 2 | - |
| 17 | - | | 3 | 2 |
| 18 | 4 | | 3 | 2 |
| 19 | 4 | | 3 | 2 |
| 20 | - | 3 | 4 | 2 |
| 21 | - | 3 | 4 | 2 |
| 22 | - | 3 | 4 | 2 |
| 23 | - | 3 | 4 | 2 |
| 24 | - | - | 4 | 3 |
| 25 | - | 4 | - | 3 |
| 26 | - | 4 | - | 3 |
| 27 | - | 4 | - | 3 |
| 28 | - | - | - | 4 |
| 29 | - | - | - | 4 |
| 30 | - | - | - | 4 |
| 31 | - | - | - | 4 |
| 32 | - | - | - | 4 |

## SONRAKİ OLAY TEKNİĞİ

Bu yaklaşımda, model yalnız bir durum değişmesinin olacağı bilindiğinde yüklenir ve güncellenir. Bu durum değişimleri genellikle olaylar olarak adlandırılır ve zaman olaylar olaya aktarıldığı için "sonraki olay" tekniği olarak adlandırılır.

Tablodaki olaylar;

- bir iş gelir.
- makine A işe başlar.
- makine A işi bitirir.
- makine B işe başlar.
- makine B işi bitirir.

Sonraki Olay +

Atölye sonraki-olay benzetimi

| İş No. | Geliş zamanı | Makine A Başlama | Makine A Bitiş | Makine B Başlama | Makine B Bitiş |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 5 | 6 | 10 |
| 2 | 8 | 8 | 16 | 17 | 23 |
| 3 | 14 | 17 | 19 | 24 | 27 |
| 4 | 16 | 20 | 24 | 28 | 32 |

Zaman Dilimleme Mi Sonraki Olay Mı?

Sonraki olay tekniği zaman dilimleme yaklaşımına göre iki avantaja sahiptir;

1) zaman artımı, yüksek yada düşük faaliyet dönemlerini otomatik olarak ayarlar, böylece yararsız ve gereksiz modelin durumunun kontrollerinden kaçınılmış olur.

2) önemli olayların benzetimde ne zaman olduğunu açıklıkla ortaya koyar.

Stokastik mi Deterministik mi?

Bir sistem; eğer davranışı tümüyle tahmin edilebilir ise; deterministik
eğer davranışı tümüyle tahmin edilemiyor ise; stokastiktir

! Deterministik benzetim modeli hiç bir stokastik eleman içermez !
(Bu yüzden deterministik modelle zaman dilimleme kullanılabilir) ?

! Stokastik benzetim modellerinde olasılık dağılımları kullanılır !

Sürekli ve Kesikli Değişme

Bir benzetim modelinde bulunan değişkenlerin değerlerinin dört farklı yolla değişeceği düşünülebilir.

→ Her bir zaman noktasında sürekli
→ Sürekli fakat yalnız kesikli zaman noktalarında
→ Herhangi bir zaman noktasında kesikli
→ Kesikli ve yalnız zamanın kesikli noktalarında

Kesikli Benzetim Modeli

Kesikli sistemlerde, durum değişkenleri zaman içinde yalnızca kesikli noktalarda değişir. Örnek; banka

Kesikli bir benzetim modeli her zaman kesikli bir sistemin benzetimi için kullanılmaz. Belirli bir sistem için kesikli veya sürekli modelin kullanılacağına dair karar çalışmanın amacına bağlıdır. Örneğin; çevre yolunda trafik akışının modellenmesi; arabaların hareketi ve özellikleri önemli ise kesikli bir modeldir. Arabaların hareketi bir bütün olarak dikkate alınıyorsa, trafik akışı; sürekli bir model olarak diferansiyel eşitlikler ile tanımlanabilir.

# Sayısal bilgisayarlar yalnız kesikli değerlerle işlem yaparlar. #

## Sürekli Benzetim Modeli

Sürekli sistemlerde, durum değişkenleri zaman boyunca sürekli olarak değişir. Uçak örneğinde, durum değişkenleri hız ve pozisyon sürekli olarak değişir.

## Kesikli-Sürekli Benzetim Modeli

Gerçek hayatta karşılaşılan bazı sistemler ne tam olarak sürekli ne de tam olarak kesiklidir. Bu nedenle iki modeli birlikte kurma ihtiyacı ortaya çıkar. Bu durumda düzenlenen benzetime "kesikli-sürekli bileşik benzetim modeli" adı verilir.

Kesikli ve sürekli olarak değişen durum değişkenleri arasındaki etkileşimin üç temel türü Pritsker, Pritsker ve Pegden tarafından şu şekilde açıklanmıştır.

Benzetim Modeli;

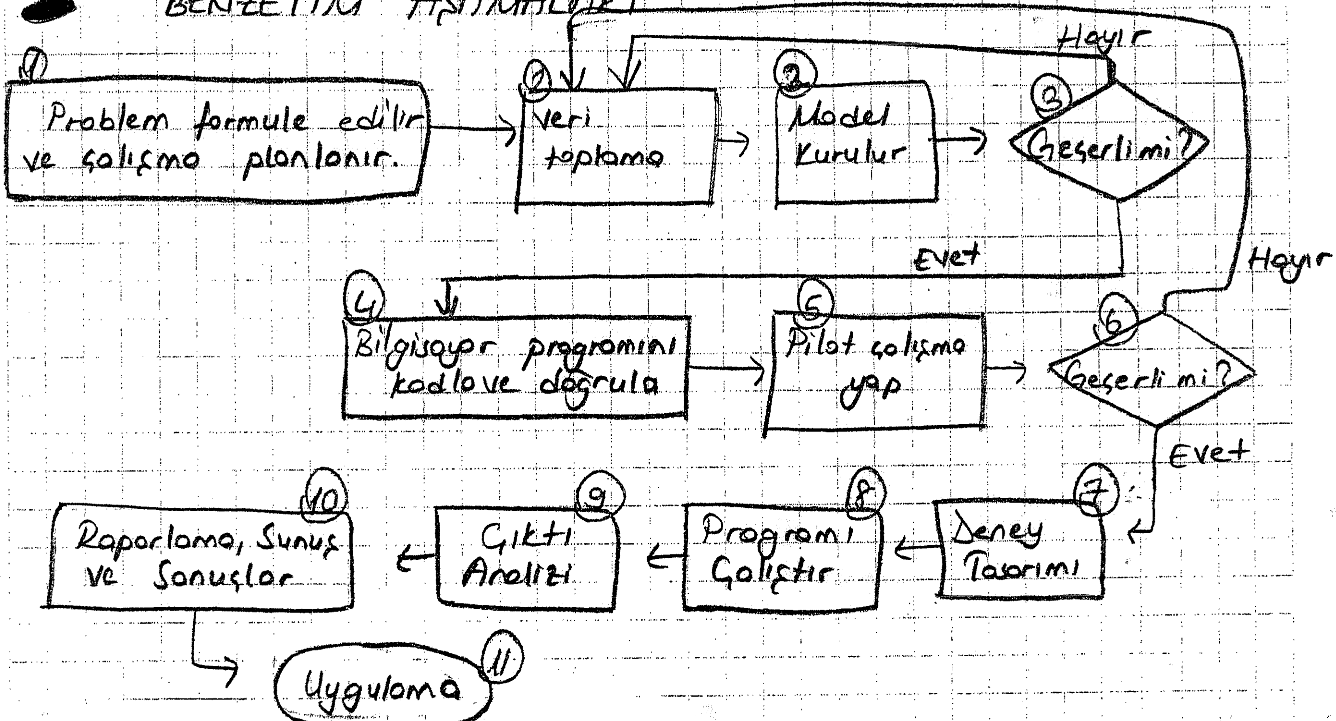
→ Kesikli bir olay, sürekli durum değişkenin değerinde kesikli bir değişikliğe sebep olabilir.

→ Kesikli bir olay, sürekli durum değişkeninin değişim bağlantısını (fonksiyonunu) belirli bir zamanda değiştirir.

→ Tetikleme noktasına (başlama veya limit değerine, yani bir üretim prosesinde sürekli bir üretim yapılırken saat 12.00'de öğle paydosu olması gibi) gelen sürekli durum değişkeni kesikli bir olayın olmasına ve programlanmasına sebep olabilir.

## BENZETİM AŞAMALARI

Sınavda sordu (10 puan)




(2) Kurulan model sistemi tanımlayacak yeterli detaya sahip olmalıdır. Ancak sistem elemanlarıyla model elemanları arasında bire bir eşleme gerekli değildir. Çok detaylı bir modelin programlanması ve çalıştırılması çok pahalı olabilir.

200

(5) Bilgisayar programını kodla ve doğrula

KOD

Genel Amaçlı Diller — Örnek; Java, C++, Visual Basic

Özel Amaçlı Simülasyon Dilleri / Programları — Örnek; SIMAN, ARENA, EXTEND

(6) Geçerli mi?

Pilot deneylerle, bir girdi parametresinde küçük değişiklikler yapılarak modelin çıktısının duyarlılığı test edilir.

(7) Deney Tasarımı

Model kurulduktan sonra, alternatif senaryolar detaylı olarak belirlenir.

(8) Programı Çalıştır - Deneyler

Deneylerin, oluşturulan deney tasarımına uygun olarak bilgisayar ortamında koşturulması çıktıların elde edilmesidir.

(9) Çıktı Analizi

8. adımda yapılan deneylerden elde edilen çıktıların istatistiksel analizi yapılır. Amaç; sistem için performans ölçüsünün güven aralığını oluşturmak.

(10) Raporlar - Sonuçlar

Modelin çalıştırılması ve sonuçlarının elde edilmesinden sonra, toplanan bilgilerin ve varılan sonuçların karar vericiye sunulması

(11) Uygulama

Başarılı? Başarısız?

Son Sayfaya Bak!

3. Slayt

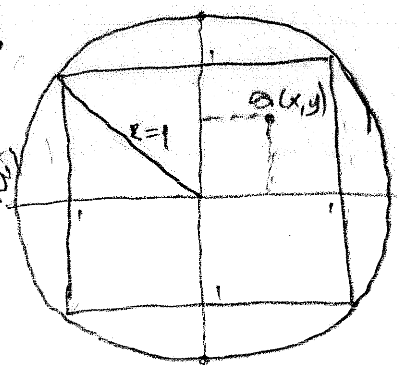


## Sistemin performans ölçütleri

- **MONTE CARLO Benzetim Modeli**
  - → Olasılık teorisi üzerine kuruludur.

~~Örnek 1~~;

direkt yarıçap çizilmiş




Yanda verilen şekildeki gibi bir karenin içine teğet olarak yerleştirilmiş bir çember düşünelim. Çemberin yarıçapı = 1 birim

Karenin bir kenarı = 2 birim

Q(x,y) → karenin içinde bir nokta.

$x^2+y^2 \leq 1$ ise Q noktası çemberin içinde deriz, Aksi halde nokta çemberin dışında demektir.

$$\text{Çember alanı} = \pi r^2$$

$$\text{Karenin alanı} = 2r \times 2r = 4r^2$$

Q noktasının çemberin içinde kalma ihtimali;

$$\frac{\text{çemberin alanı}}{\text{karenin alanı}} = \frac{\pi}{4} \text{ olur.}$$

→ Karenin içi n adet rastgele noktalarla doldurulsun. Toplamda n tane nokta olsun, bunlardan m tanesi çemberin içinde kalacaktır.

$$\frac{\text{çemberin içinde kalan noktalar}}{\text{karenin içinde kalan noktalar}} = \frac{m}{n}$$

Diğer denklemle bu denklemi birleştirirsek;

$$\frac{m}{n} = \frac{\pi}{4} \qquad \boxed{\pi = \frac{4m}{n}} \text{ olarak bulunur.}$$

matlab kodu

```
function pi = montecarlo(n)
 cember = 0;
 sayac = 0;
 for i = 1:n
     x = rand;
     y = rand;
     if ((x^2 + y^2 <= 1)
         cember = cember + 1
     end
       sayac = sayac + 1
     end
 pi = 4*cember/sayac
```

cember; çemberin içinde kalanların sayısı
n: nokta sayısı
pi: pi sayısı

```
>> pi = montecarlo(1000000)
   pi = 3.1415
```

1

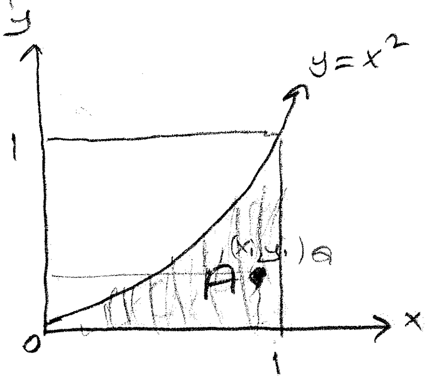
# BMÜ-421 Benzetim ve Modelleme MATLAB SIMULINK

İlhan AYDIN

## SIMULINK ORTAMI

- Simulink bize karmaşık sistemleri tasarlama ve simülasyon yapma olanağı vermektedir.
- Mühendislik sistemlerinde simülasyonun önemi gün geçtikçe artmaktadır.
- Sistemlerin tasarımında büyük oranda bilgisayar simülasyonlarından faydalanmakta, mümkün olduğunda tasarımın test aşamaları da bilgisayarlar yardımıyla yapılmaktadır.
- Günümüzde mühendislik alanında en çok kullanılan programlardan birisi MATLAB'dir.
- Simulink, MATLAB ile birlikte bütünleşik olarak çalışan bir simülasyon ortamıdır.
- Sürekli zamanlı ve ayrık zamanlı sistemleri ,veya her ikisini de içeren hibrit sistemleri desteklemektedir.
- İçinde birçok alt sistemi blok olarak barındırdığından sürükle-bırak yöntemiyle birçok sistemi bir-kaç dakikada kurarak simule edebilir, değişik durumlardaki cevabını test edebilirsiniz.
- Bunun için Simulink bizlere zengin bir blok kütüphanesi sunmaktadır.

Örnek 2:




Şekilde görülen $y=x^2$ eğrisi ile x ekseni arasında kalan taralı alanı bulmak için monte carlo metodunu kullanınız.

Eğer dikdörtgen içerisinde rastgele noktalar $(x_i, y_i)$ işaretleyip bu noktaların eğrinin altında olup olmadıklarını belirler ve bunu toplam nokta sayısına oranlarsak A alanını 1 karesine olan oranını yaklaşık olarak elde edebiliriz.

$a(x_i, y_i)$

$$\frac{\text{eğrinin altında olma oranı}}{\text{karenin altında olma oranı}} = \frac{\int_0^1 x^2 \, dx}{1 \cdot 1} = \frac{\left.\frac{x^3}{3}\right|_0^1}{1} = \frac{1}{3}$$

Toplam nokta $= n$ olsun.
Eğrinin altında kalan nokta sayısı $= m$

$$\frac{m}{n} = \text{noktanın eğri altında kalma olasılığı}$$

iki denklemi eşlersek;

$$\frac{m}{n} = \frac{1}{3}$$

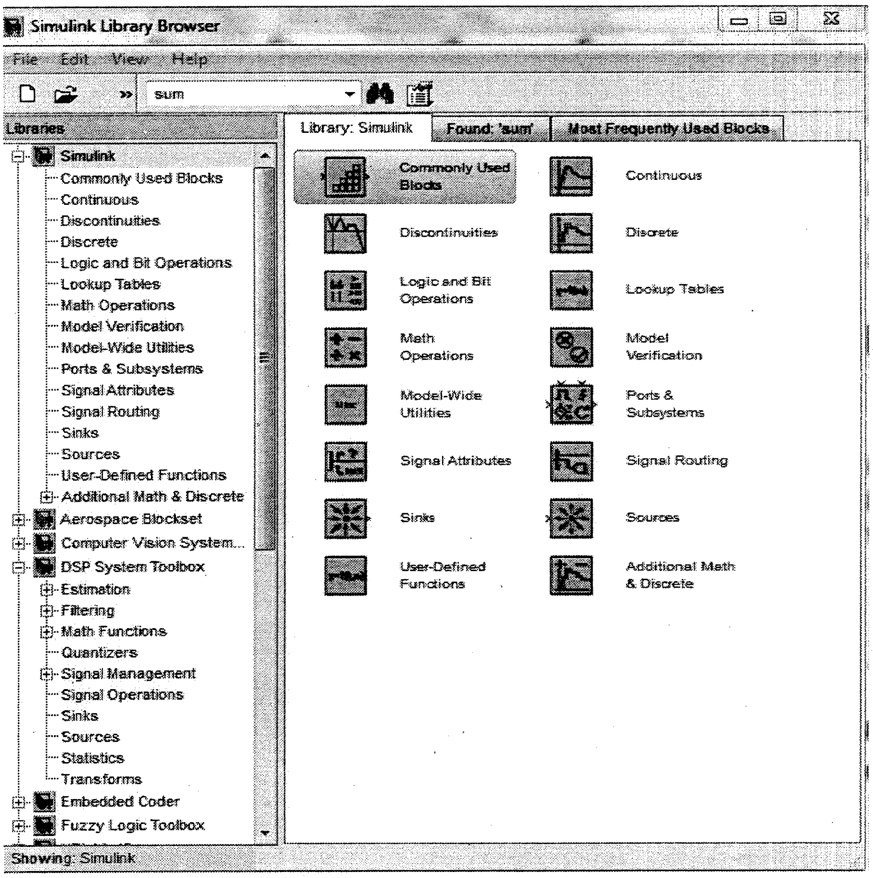
**matlab kodu**

```
function egriAlan = xkare(n)
    egri=0;
    sayac=0;
    for i=0:n
        x=rand;
        y=rand;
        if (y<x*x)
            egri=egri+1
        end
        sayac=sayac+1
    end
egriAlan= eğri/sayac;
```

# SIMULINK ORTAMI

- **Simulink Kütüphanesi:**
- Simulink'i çalıştırdığınızda karşınıza Simulink kütüphanesi gelecektir.
- Simulasyon yaparken kullanacağımız bloklar kategorilere ayrılmış biçimde burada bulunmaktadır.
- **Blok Diyagramlar:**Her bir blok sürekli zamanda ya da ayrık zamanda çıkış veren temel bir dinamik sistemi ifade eder.
- **Bloklar:**Bloklar Simulink'in nasıl simule edileceğini bildiği temel dinamik sistemleri temsil eder.
- **Durumlar:**Bloklar durumlara sahip olabilirler. Simulink İntegral alıcı (integrator) bloğu duruma sahip bloklara bir örnektir.
- **Değiştirilebilir Parametreler:**
- Birçok blok parametresi değiştirilebilirdir. Örneğin Kazanç bloğunun kazanç parametresi değiştirilebilir parametredir.
- **Veri Tipleri:** SIMULINK'te desteklenen int8,double ve boolean gibi herhangi bir dahili veri tipini kullanabilir.

# SIMULINK ORTAMI



Örnek 3 : 0 ile 100 arasında bulunan sayılar içinden rastgele seçilen bir sayının 11'e tam bölünebilme olasılığını monte carlo yöntemiyle çözünüz.

Analitik çözüm :

0 ile 100 arasında 11 ile bölünen sayıları bulup bunları tüm sayılara oranlarsak sorumuzun cevabını bulabiliriz.

11, 22, 33, 44, 55, 66, 77, 88, 99 olmak üzere toplam 9 adet.

Tüm sayılar → 100 adet

$P = \frac{9}{100} = 0.09$ olmaktadır.

monte Carlo benzetimi ile çözüm ;

monte carlo benzetimi 0 ile 100 arasında rastgele n adet sayı seçmemizi ister. Seçeceğimiz n adet sayıdan m tanesinin 11 ile tam bölündüğünü farzedelim.

Olasılık değeri $= \frac{m}{n}$ 'dir.

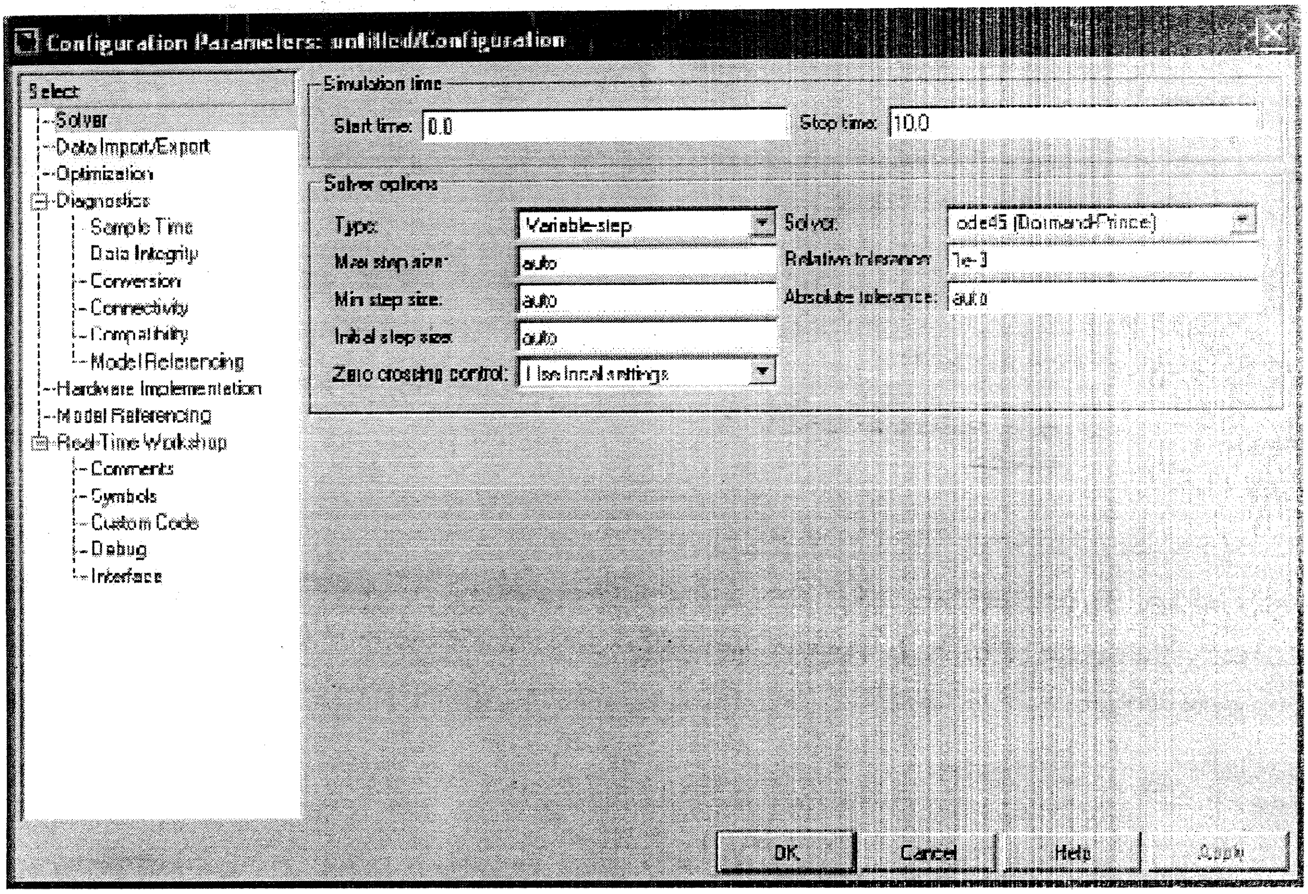
rogramını yazarken ; rastgele seçilecek her sayı değeri için n artarken ;
1 ile bölünmesi durumunda m arttırılmalıdır.

matlab kodu

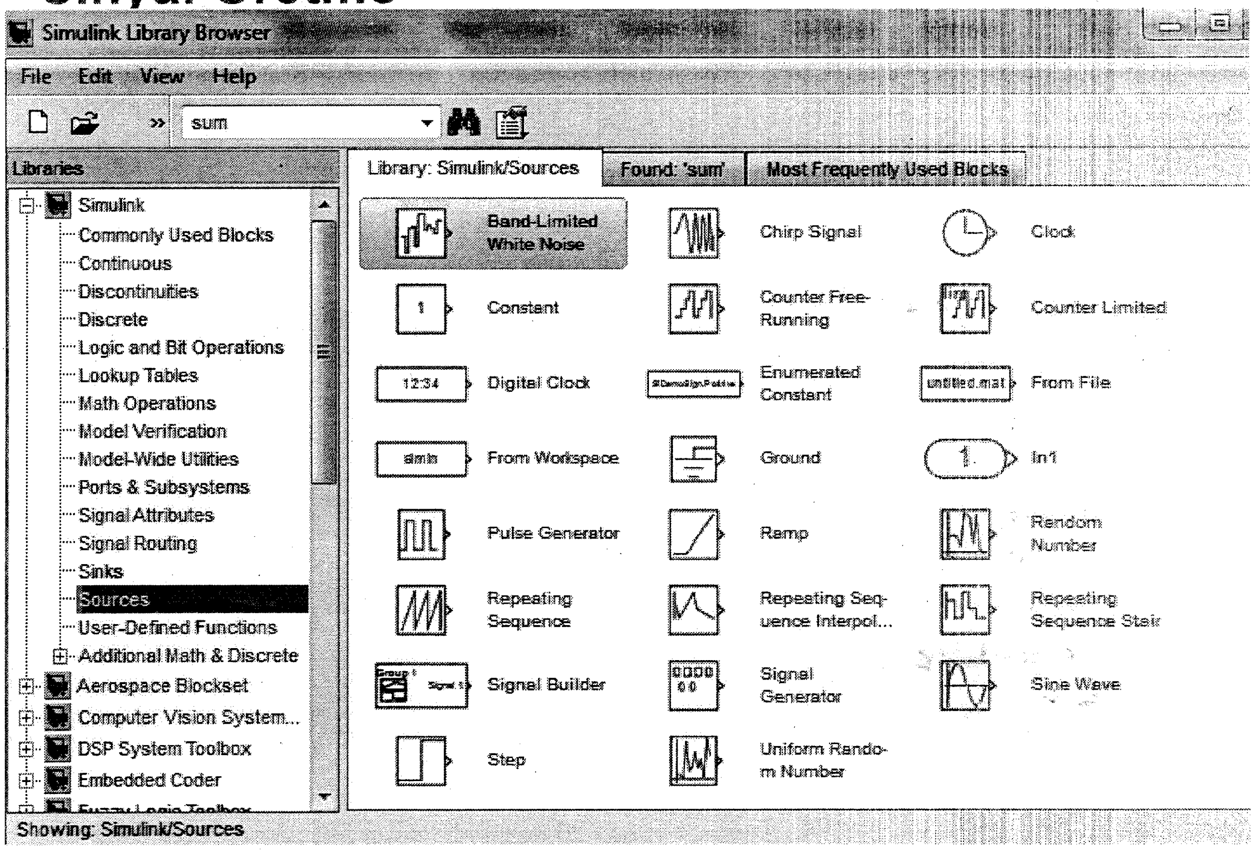
b : bolunen sayıların sayısı
s : sayac, toplam seçilen sayı
n : uretilecek sayı miktarı

```
function sonuc = bolunebilme (n)
    b=0;
    s=0;

    for i=0:n
     sayi= round (rand * 99) +1;
       sonuc = mod (sayi,11);
          if (sonuc ==0)
             b=b+1;
          end
             s=s+1
          end
   sonuc = b/s;
```

# Simülasyon Zamanı Ayarlama



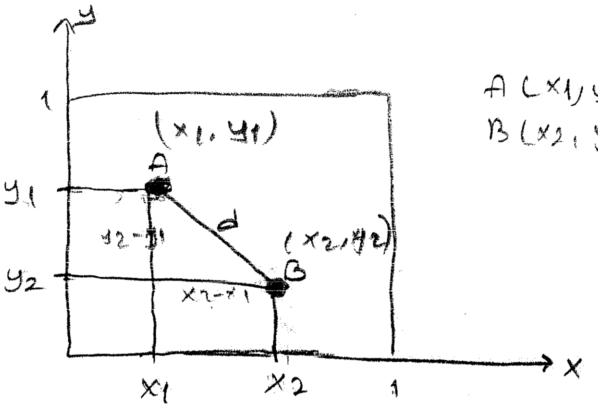

# Sinyal Üretme



## ÖRNEK 4:

Kenarları birim uzunlukta olan bir kare düşününüz. Bu kare içinde rassal seçilen A ve B noktaları olsun. A ve B arası d uzunluğundadır. d'nin 0.8'den küçük olma olasılığı nedir?

Monte Carlo tekniğiyle rassal olarak 1000 adet A ve B noktaları üreterek d'nin 0.8'den küçük olma olasılığını bulunuz. Kullanacağınız yaklaşımı açıklayarak, akış ~~şeması~~ çiziniz.



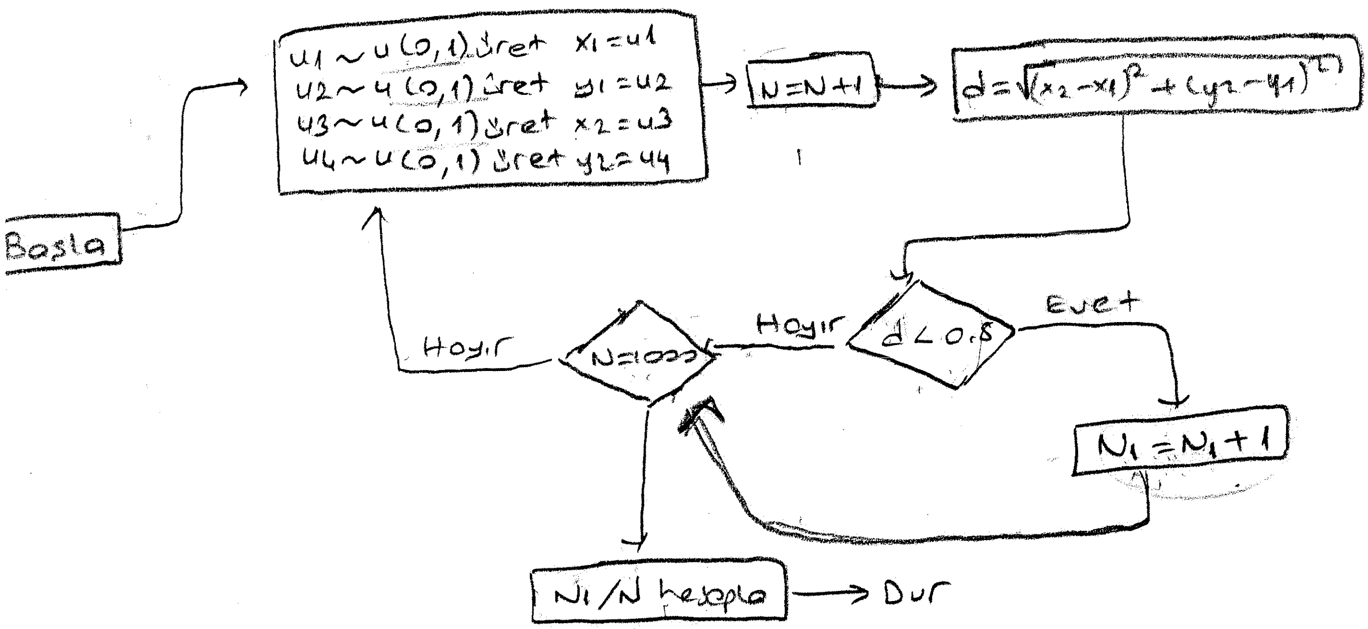

A $(x_1, y_1)$

B $(x_2, y_2)$

$$d = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$$

$N_1$: d'nin 0.8'den küçük old. sayaç

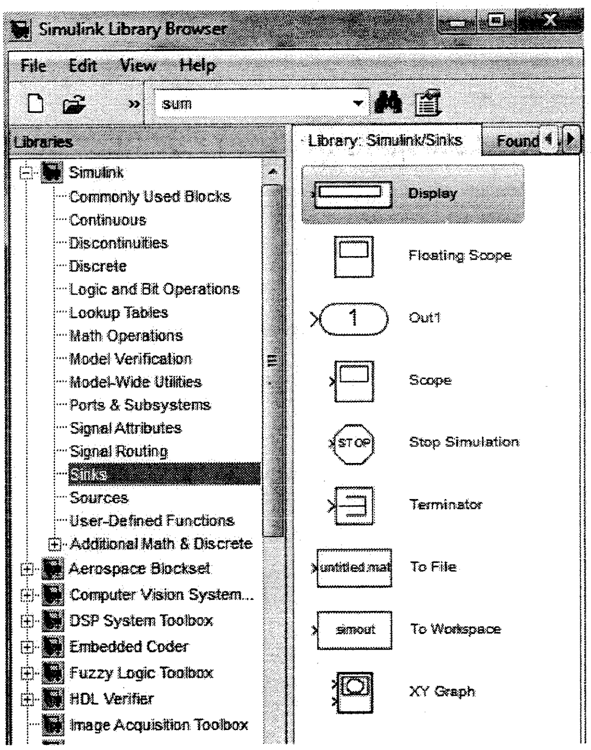
N: 1000 deneme

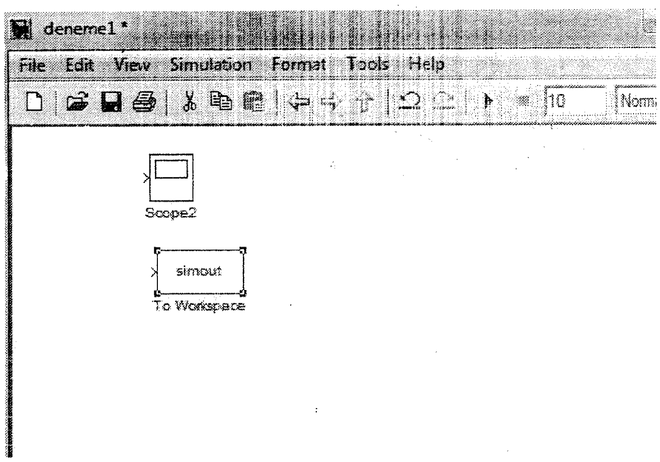



### matlab kodu

```
function sonuc=hesaplama(n)
   x1=rnd;  x2=rnd;  y1=rnd;  y2=rnd;
   N=0;
   N1=0;
   for i=0:n

        d = sqrt(((x2-x1)^2)+((y2-y1)^2))
       if (d<0.8)
           N1 = N1 + 1
       end
   N=N+1
   end

 sonuc= N1/N;
```

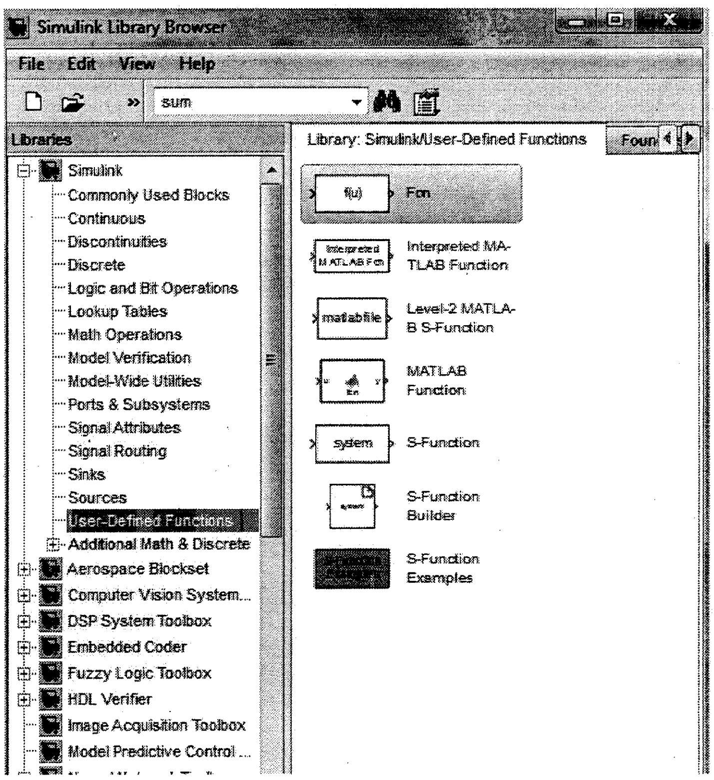
# Simülasyon Sonuçlarının Elde Edilmesi




Herhangi bir simgeyi seçip çalışma ortamına sürükleyip bırakarak ekleyebilirsiniz.

# Kullanıcı Tanımlı Fonksiyonlar




Tanımladığınız .m file dosyalarını MATLAB Function ile ekleyebilirsiniz.

Örnek 5:

| Aylar | Gönderilen miktar |
|---|---|
| 1 | 200 |
| 2 | 400 |
| 3 | 300 |
| 4 | 300 |
| 5 | 400 |
| 6 | 100 |
| 7 | 400 |
| 8 | 300 |
| 9 | 400 |
| 10 | 400 |
| 11 | 500 |
| 12 | 300 |

1. yıl

| Aylar | Gönderilen miktar |
|---|---|
| 1 | 300 |
| 2 | 200 |
| 3 | 400 |
| 4 | 300 |
| 5 | 400 |
| 6 | 500 |
| 7 | 100 |
| 8 | 400 |
| 9 | 400 |
| 10 | 200 |
| 11 | 400 |
| 12 | 500 |

2. yıl

| Aylar | Gönderilen miktar |
|---|---|
| 1 | 200 |
| 2 | 300 |
| 3 | 400 |
| 4 | 300 |
| 5 | 200 |
| 6 | 500 |
| 7 | 300 |
| 8 | 200 |
| 9 | 200 |
| 10 | 100 |
| 11 | 300 |
| 12 | 200 |

3. yıl

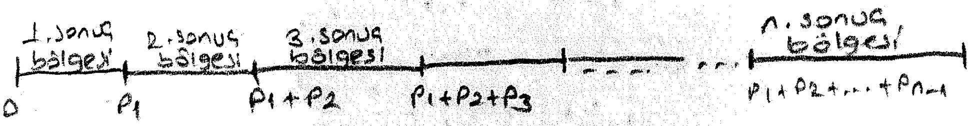
4. sınıfta olan bir üni öğrencisine ailesinin 3 yıl içinde aylık gönderdiği para miktarları yukarıda gösterilmiştir. Monte carlo yöntemiyle öğrenciye 4. yılında gönderilecek parayı tespit ediniz.

Çözüm:

1.) İlk 3 yıl içinde gönderilmiş miktarların olasılıklarını buluyoruz.

| Gönderilen miktar | Frekans | Olasılık |
|---|---|---|
| 100 | 3 | 0.083333 |
| 200 | 8 | 0.222222 |
| 300 | 10 | 0.27778 |
| 400 | 11 | 0.30555 |
| 500 | 4 | 0.111111 |
| | 36 | 1 |

2.) Bulduğumuz olasılık değerlerini Monte Carlo benzetiminde kullanmak için aşağıdaki yapıya benzer bir yapı elde etmemiz gerekiyor.




Belirtilen bir gelişigüzel sayı, hangi sonuç bölgesine düşerse, olayda o sonuç meydana gelmiştir.

$0 < q < P_1$ ise 1. sonuç bölgesi

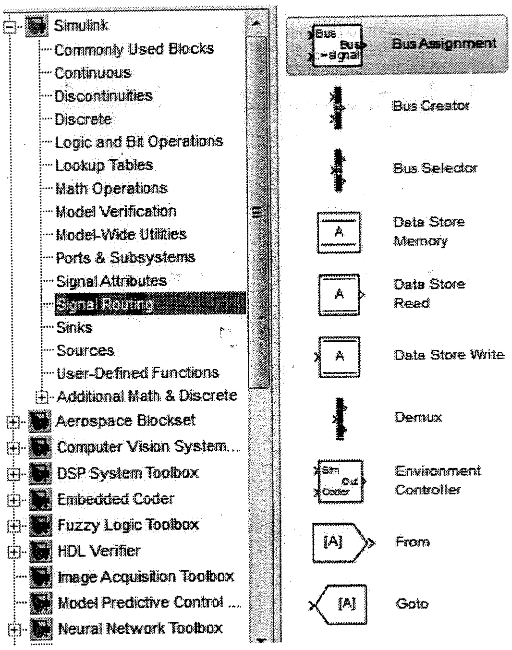
$P_1 \le q < P_1+P_2$ ise 2. sonuç bölgesi

$\vdots$

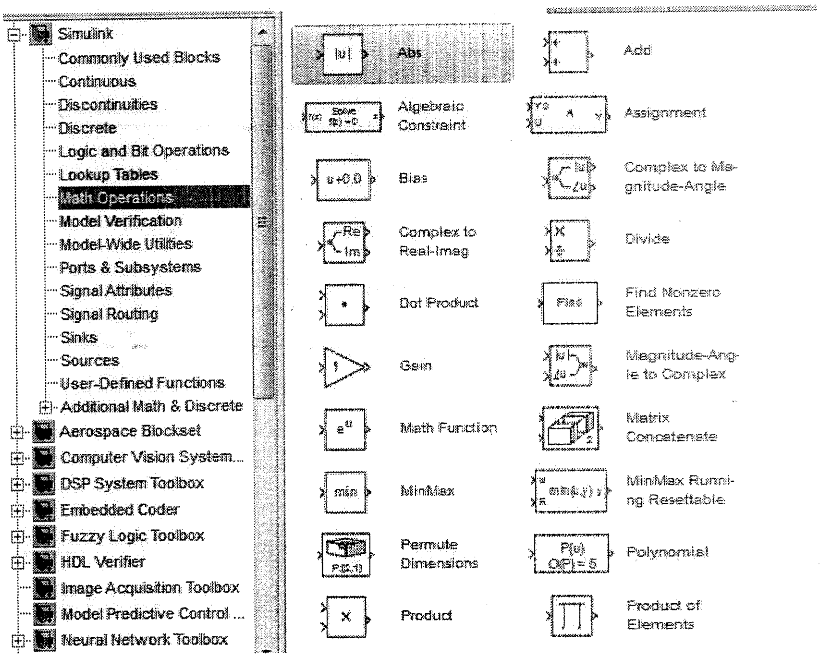
$P_1+P_2+\dots+P_{n-1} \le q < 1$ ise n. sonuç bölgesi

Şimdi örneğimize bu mantığı uygulayalım. (Kümülatif olasılık demektir bu)

# Diğer Fonksiyonlar



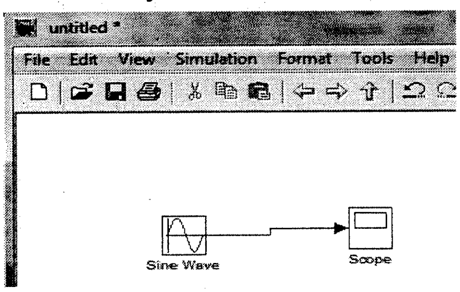

Sinyal yönlendirme

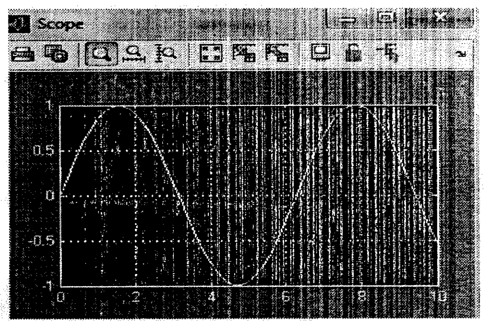


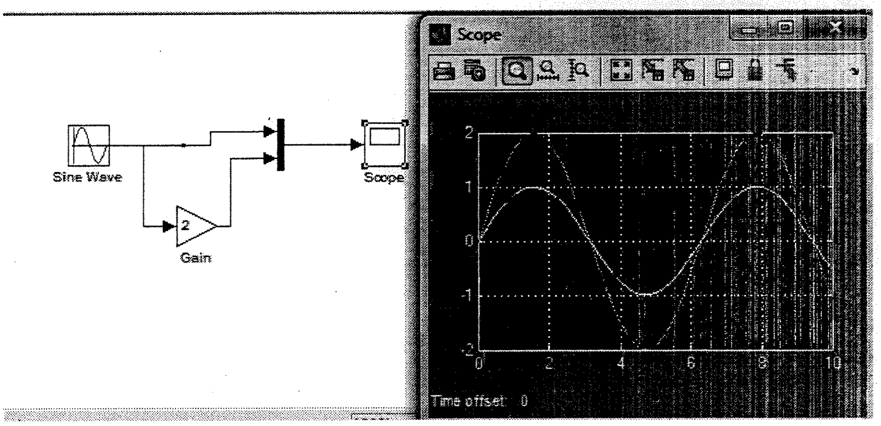

Matematiksel işlemler

# Blokların Bağlanması

- Bloklar sürükle-bırak şeklinde çalışma ortamına eklenir.
- Blokları bağlamak için kontrol tuşu basılı iken seçili bloktan diğerine tıklamak yeterlidir.



| Kümülatif Olasılık | miktar |
|---|---|
| 0 | 100 |
| 0,0833333 | 200 |
| 0,3055556 | 300 |
| 0,5833333 | 400 |
| 0,888889 | 500 |

0.000 – 0.083 Arası → 100
0.083 – 0.305 Arası → 200
0.305 – 0.583 Arası → 300
0.583 – 0.888 Arası → 400
0.888 – 1.000 Arası → 500

Her bir olasılık değerini bir öncekiyle toplayarak kümülatif olasılığı buluruz.

...) monte carlo benzetimine uygun hale geldi örneğimiz. Çeşitli platformlar-a rastgele sayılar üreterek deneyebiliriz. Excel ortamında yaparsak;
-SAYIURET() metodu ile 0-1 aralığında sayılar üretiriz. Sayıuret() metodu excel-de varolan bir metoddur.
Kümülatif hesaplamaları DUSEYARA() metoduyla yapıyoruz.

Ürettiğimiz rastgele değerlere göre (12 değer) tam olarak doğru bir sonuç elde edemedik. Fakat örnek sayısını 200 yaptığımızda doğru sonuca yaklaşmış oluruz.
Daha sağlıklı sonuçlar için haydi örnek sayısını arttırmaya! :)

matlab

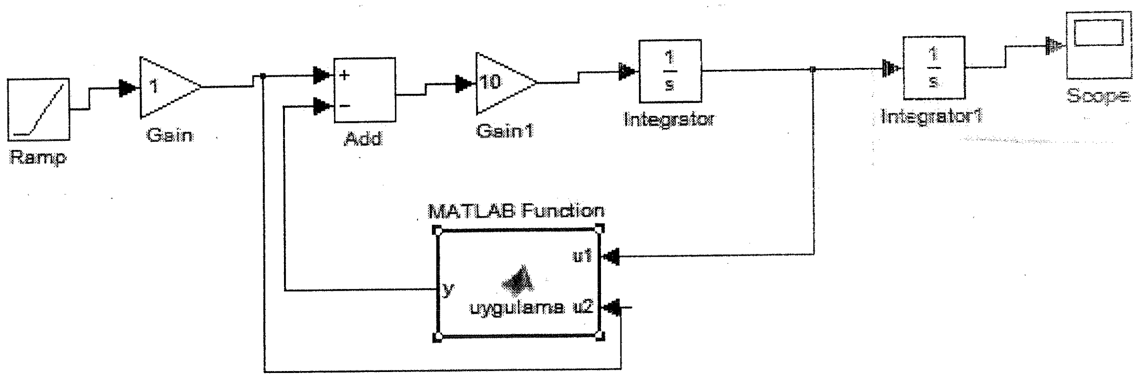
```
function para = hesaplama(n)
    f1=0; f2=0;

    for i = 1:n
        if( 0.000 < f < 0.083)
            f1 = f1+1;
        end
        if( 0.083 < f < 0.305)
            f2=f2+1;
        ;
        end
    end
para = f1 + f2 + f3 + f4 + f5
      *100
```

f=rand

rand;

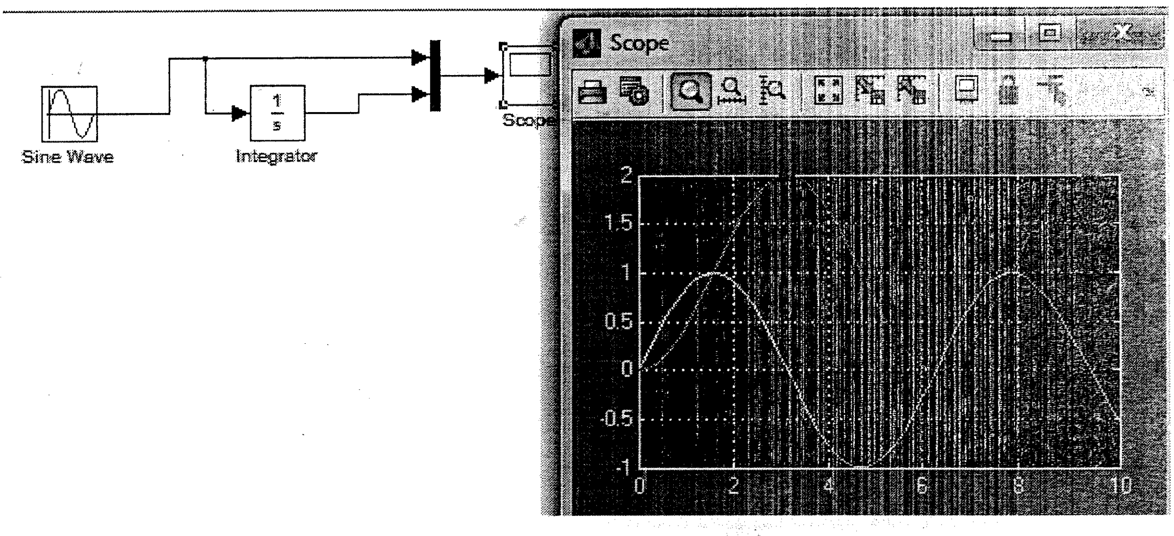
round(rand*99)+1

# Kullanıcı Tanımlı Fonksiyonlar




```
function y = uygulama(u1,u2)
St1=1;
c=1;
if u2>St1 || u2<-St1
   y=c*u1;
else
   y=u2;
end
```

# Örnek: Basit bir model oluşturmak



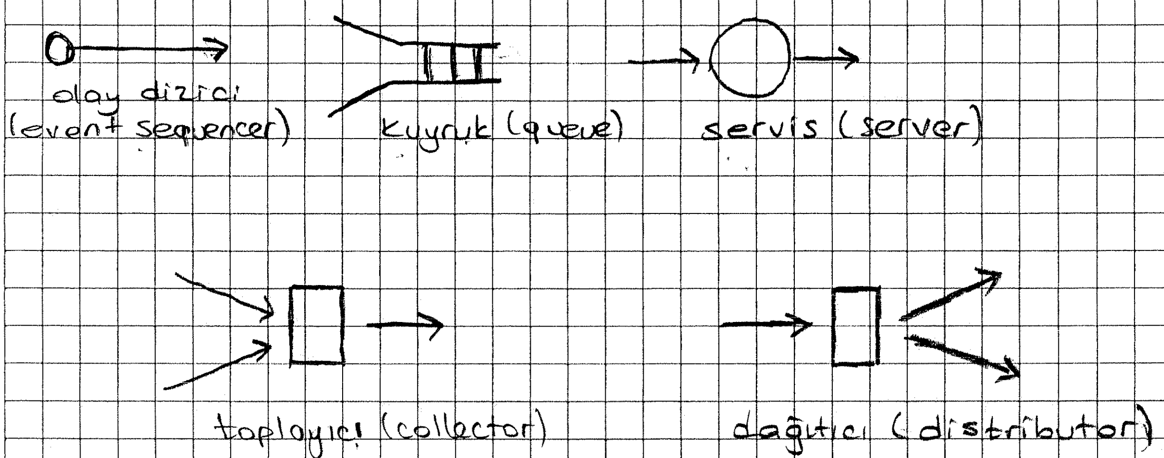
MERVE TELGEKEN

## BENZETİM VE MODELLEME (6. hafta)

### OLAY TABANLI MODELLEME

- Zaman sürümlü modeller, düzenli zaman aralıklarında senkron bir tarzda ilerleyen sinyallere sahip sistemleri karakterize eder.
- Olay zamanlı modeller asenkron olup, düzensiz ve çoğunlukla rastgele aralıklarla oluşan çok basit sinyallere sahiptir.
- Olayların ne olduğu değil ne zaman olduğu önemlidir.
- Sayısal bilgisayar sistemleri olay-tabanlı sistemler için tipik bir örnektir. Örneğin, sayısal iletişim sisteminde rastgele fakat istatistiksel bir şekilde dağılım ile mesaj varışları modellenir.

Benzetim Diyagramları




- Olay dizici : Belirli zamanlarda varan olayları kabul eder (müşteri)
- Kuyruk : Eğer servis dolu ise müşteriler kuyrukta bekler. (FIFO)
- Servis : Kuyruktan ayrılan müşteriler serviste işlem görür.
- Toplayıcı : İki veya daha fazla kuyruğu birleştirir.
- Dağıtıcı : Kuyrukta bekleyenleri servislere ya da diğer kuyruklara yönlendirir.

KUYRUK SİSTEMİ

| Sistem | Servisler | Müşteriler |
|---|---|---|
| • Banka | Vezneler | Müşteriler |
| • Hastane | Doktorlar, Hemşireler, Yataklar | Hastalar |
| • Bilgisayar Sistemi | MİB, Giriş çıkış aygıtları | İşler |
| • Montaj Hattı | İşçiler, makinalar | Üretilen birimler |
| • Havaalanı | Pist, güvenlik birimleri | Uçaklar, yolcular |

Bir Kuyruk Sisteminin Bileşenleri

1.) Varış prosesi

2.) Servis Prosesi

3.) Kuyruk Disiplini

4.) Sistemde izin verilen müşteri sayısı (max 100 tane)

5.) Müşterilerin geldiği yığının genişliği

1.) Varış Prosesi :

Müşterilerin sisteme geliş modelini tanımlar.

E(a) : Varışlararası ortalama zaman

$\lambda$ : Müşterilerin varış oranı.

$E(a) = 1/\lambda$ $\quad 1/5$

örneğin; Bir dakikada 5 varış olan bir sistemde varışlar arası zaman aralığı ortalaması

$E(a) = 1/\lambda = 1/5 = 0.20$ dakika.

• Deterministik Varış prosesi

• Rassal varış prosesi
→ Poisson Dağılımı

$$P(X=k) = \frac{(\lambda t)^k \cdot e^{-\lambda t}}{k!}$$

KeskinColor

ÖRNEK : Bilgisayar Donanım Problemi

müşteriler, Poison dağılımına göre varış yapmaktadır.
Salı 8:00 - 9:00 = 6 müşteri (ortalama) ise;

8:00 - 8:30 saatleri arasında varış yapma olasılığı = ?

Çözüm:

$\lambda = 6$ müşteri varışı / saat

$t = 30$ dk $= 0.5$ saat

$\lambda t = 0.5 \times 6 = 3$ müşteri

$\lambda$: müşterilerin varış oranı

$t$:

$$P(x=k) = \frac{(\lambda t)^k \cdot e^{-\lambda t}}{k!}$$

$P(x=0) = 3^0 \cdot e^{-3} / 0! = 0.049787$

$P(x=1) = 3^1 \cdot e^{-3} / 1! = 0.149361$

$P(x=2) = 3^2 \cdot e^{-3} / 2! = 0.224042$

ÖRNEK: Bir şehirde ender rastlanan bir hastalıktan, bir hafta içinde ortalama ölen kişi sayısı 4'tür. Belli bir hafta içinde bu hastalıktan,

a.) Hiç kimsenin ölmemesi
b.) En az 2 kişinin ölmemesi
c.) 3 kişinin ölmesi
olasılıklarını hesaplayınız.

Çözüm: $x$: bir haftada bu hastalıktan ölenlerin sayısı

$$P(x=k) = \frac{(\lambda t)^k \cdot e^{-\lambda t}}{k!}$$

a.) $P(x=0) = \dfrac{e^{-4} \cdot 4^0}{0!} = 0.0183$

b.) $P(x \geq 2) = 1 - P(x=1) - P(x=0)$

c.) $P(x=3) = \dfrac{e^{-4} \cdot 4^3}{3!} = 0.195$

## 2.) Servis Prosesi

Servis sayısı ve servis zamanı dağılımı ile karakterize edilir.

s : müşterinin servis zamanı
E(s) : Bir müşterinin servis zamanı ortalaması
$\mu$ : servis oranı (Birim zamanda servis gören müşteri sayısı)

$$\mu = 1 / E(s)$$

1 servis 2 dk
x 1 dk
$\frac{1}{2}$

ÖRNEK: Ortalama servis zamanı 2 dakika ise,

Servis oranı $= \mu = 1 / E(s) = 1/2 = 0.5$ servis/dakika

## 3.) Kuyruk Disiplini

Kuyruk sistemlerinde en önemli parametre trafik yoğunluğudur.

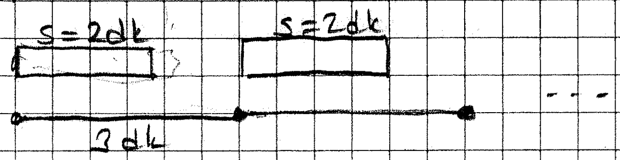
$$p = (\text{varış oranı}) / [(\text{servis oranı}) * c]$$

c → servis sayısı

$p < 1$ ise servis $(1-p)$ oranında boştur.
$p < 1$ ise sistemde sürekli artan bir kuyruk oluşur.
$p = 1$ ise servis %100 doludur ve kuyruk yoktur.

ÖRNEK: 3 dakikada bir servisin olduğu bir sistemde servis zamanı 2 dakika olsun. Doluluk oranını bulunuz.

ÇÖZÜM:




$p = E(s) / E(a) = 2/3 = 0.667$ (doluluk oranı)

$= 1 - p = 1 - 0.667 = 0.333$ (servisin boş kalma oranı)

- Analitik ve benzetim modelinde $p < 1$ olduğu kabul edilir.

Kuyruk Modelleri

FIFO : İlk giren ilk çıkar prensibi

LIFO : Son giren ilk çıkar prensibi

Öncelik (Priority) : Müşterilerin önemine göre servis

Aksi belirtilmedikçe FIFO kullanılır.

Kendall'in kuyruk sistemi modellerini sınıflandırmak için kriterleri:

1 • Varış prosesi
2 • Servis prosesi
3 • Servis sayısı
4 • Paralel servis sayısı
5 • Sisteme izin verilen müşteri sayısı
6 • müşterilerin geldiği yığının genişliği

Kuyruk teorisi kapsamında performans kriterleri arasında karşılıklı ilişkileri "Little" formülleriyle çözümlemek mümkündür.

Keskin Color

Dağılım olarak üstel dağılım mevcut ise;

$$f(x) = \mu \cdot e^{-\mu x}$$

$\mu$ : ortalama servis hızı (birim zamanda hizmet sunulabilen ortalama müşteri sayısı)

$1/\mu$ : ortalama servis zamanı

* "t" süresinde hizmetin tamamlanma olasılığı ;

$$\boxed{P(x \leq t) = 1 - e^{-\mu t}}$$

ÖRNEK: Bilgisayar Donanım Problemi

Servis süresi = 4 dk ; üssel dağılım

Servis zamanının 3 dakikadan kısa olması olasılığı = ?

Çözüm:

Ortalama servis zamanı $= 1/\mu = 4$ dk

Ortalama servis hızı $= \mu = 1 \times 60/4$ müşteri/saat

$= 15$ müşteri/saat

Bir hizmetin 3 dakikadan kısa verilme olasılığı = ?

$3\text{dk} = 3/60 = 0.05$ saat

$$P(x < 0.05) = 1 - e^{-15 \times 0.05}$$

$$= 0.52763$$

Keskin Color

MERVE TELÇEKEN

# BENZETİM VE MODELLEME (7. hafta)

## KESİKLİ OLAY BENZETİMİ

Sistemin zamana göre benzetimidir. Zaman içinde kesikli noktalarda bir olay ortaya çıkar ve sistemin durumunu değiştirir.

M/M/1 Modelinde

- Varışlar arası zaman rassal değişkendir.
- Servis zamanları rassal değişkendir.
- İş veya müşteri servisleri bittiği an sistemden çıkar.
- Bir servis tamamlandığında en yakın müşteri servise alınır.

Kesikli olay benzetiminde kullanılacak kavramlar:

1) Sistem : Bir veya daha fazla amacı gerçekleştirmek için çalışma zamanı boyunca etkileşimli nesnelerin toplamıdır.

2) Model : Bir sistemin gösterimidir.

3) Sistem Durumu : Herhangi bir zamanda sistemi tanımlamak için kullanılan değişkenler setidir.

4) Nesne : Sistemin 'bir' bileşenidir. (Bir müşteri, bir servis...)

5.) Özellik : Verilen nesnelerin özellikleri (Atölye'de bir işin uğrayacağı makinaların sırası.)

6) Olay : Bir sistemin durumunu değiştiren bir oluş. (müşteri varışı)

7) Faaliyet : Belirli bir zaman içinde tamamlanan iş veya işlem.

Müşterinin kuyrukta Bekleme zamanı = Müşterinin Servise Alınma zamanı − Müşterinin Sisteme varış zamanı

Müşterinin bekleme zamanı = müşterinin kuyrukta bekleme zamanı + müşterinin servis zamanı

Keskin Color

## Kesikli Olay Benzetiminde Zaman İlerletme ;

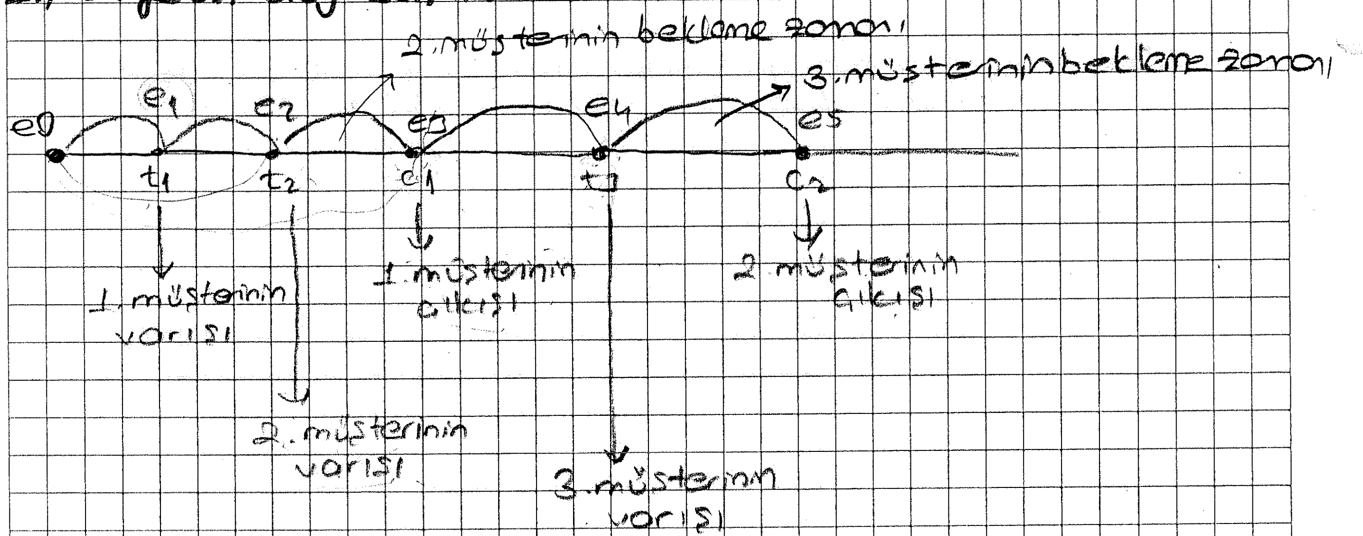
• Kesikli Olay benzetim modelinin yapısı gereği, her adımda benzetim zamanının bilinmesi gerekir. Bu noktada benzetim saatinin bir değerden bir değere artmasını sağlayacak bir işlem gerekir. Benzetim zamanını veren değişken 'Benzetim saati' olarak bilinir.

Benzetim Saatinin İlerlemesinde iki yaklaşım kullanılmaktadır.

1) En yakın olay zamanı

2) Sabit artışlar

### 1) En yakın olay zamanı ;




$t_2 < c_1$ olduğunda $e_2 = t_2$ olarak ilerletilir.

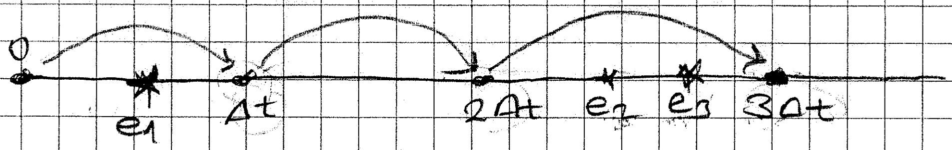
$c_1 < t_2$ olsaydı $e_2 = c_1$ olarak ilerletilir.

2.) Sabit artışlar ile zaman ilerletme

Bu yaklaşımda, benzetim saati önceden belirlenen bir $\Delta t$ zamanı kadar arttırılır.

Her artış sonrasında bir olayın ortaya çıkıp çıkmadığına bakılır.

Bu aralıkta bir veya birden fazla olay ortaya çıkmış ise bu olaylar aralığın sonunda olmuş gibi dikkate alınır ve sistemin durumu güncelleştirilir.




$\Delta t$ zamanında, $e_1$ olayı

$3\Delta t$ zamanında, $e_2$, $e_3$ olayı olmuştur.

Dezavantaj !!

? $e_2$'mi $e_3$'mü önce oldu. $\Delta t$ çok küçük alınarak bu sorun çözülebilir fakat bu şekilde de çalışma zamanı artacaktır.

Bu nedenle kesikli olay benzetiminde bu yaklaşım genellikle kullanılmaz.

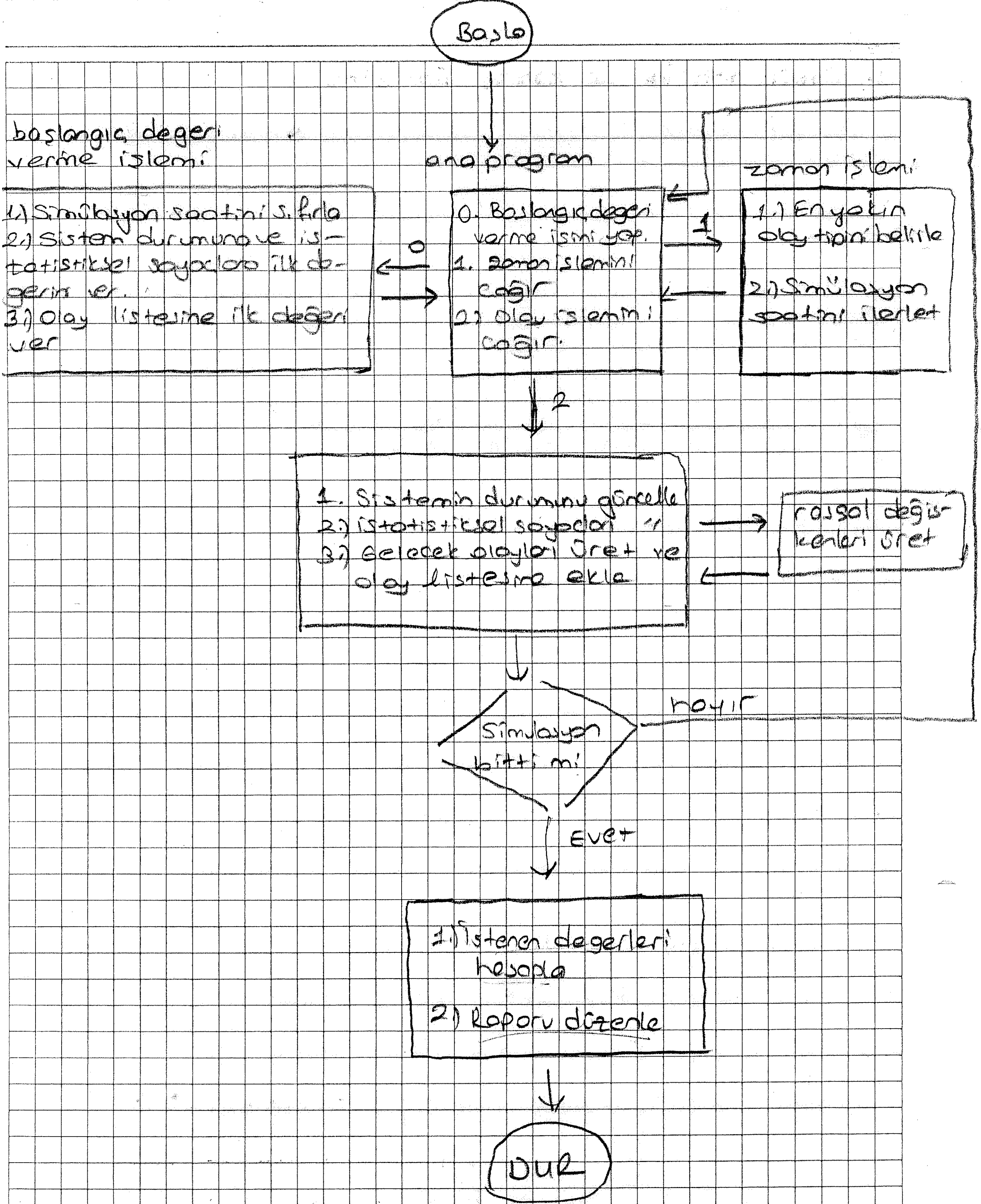
Kesikli olay Benzetimi Bileşenleri

- Sistem durumu
- Benzetim Saati
- Olay listesi
- İstatistiksel sayaçlar
- İlk değer verme işlemi
- Zaman işlemi
- Olay işlemi
- Rapor Üreteci

SİBİZORO

Keskin Color

# Kesikli Olay Benzetimi Aşamaları #



# Kesikli Olay Benzetiminde Kullanılan İstatistiksel Sayaçlar

3 tip istatistik vardır.

1) Değişkenlerin Gözlemlenmesine Dayalı İstatistik (Kesikli zaman ist.)

2) Zamana Göre Ortalama

3) Zamana göre Ortalama Değerlerin bir aralık boyunca değişmesi

## 1) Değişkenlerin Gözlemlenmesine Dayalı İstatistik

Örneğin; bir kuyruk sisteminde ortalama bekleme zamanı. Kuyruk sisteminde kuyrukta ortalama bekleme zamanının ve varyansının hesaplanması isteniyor olsun.

i. müşterinin bekleme zamanını hesapla → ($d_i$)

n müşteri için toplam bekleme zamanını bul → $\sum_{i=1}^{n} d_i$

Bekleme zamanının kareleri toplamını bul → $\sum_{i=1}^{n} d_i^2$

* müşterilerin ortalama Bekleme zamanı

$$\bar{d} = \frac{\sum_{i=1}^{n} d_i}{n}$$

* Müşterilerin Bekleme zamanı Varyansı

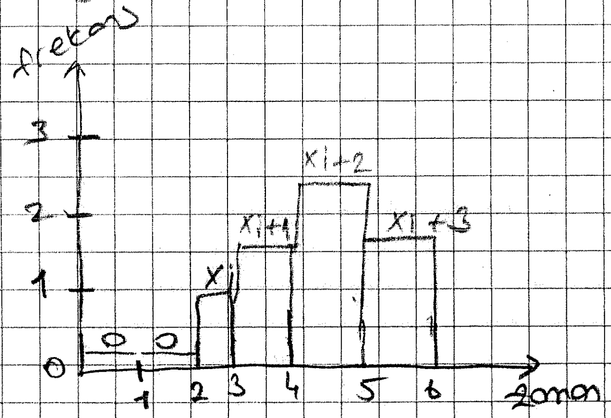
$$var(d) = \frac{\sum_{i=1}^{n} (d_i)^2 - n\bar{d}^2}{n-1}$$

Keskin Color

## 2.) Zamana Göre Ortalama

Durum değişkenleri değerlerini belirli zamanlarda atlamalarla değiştirir.

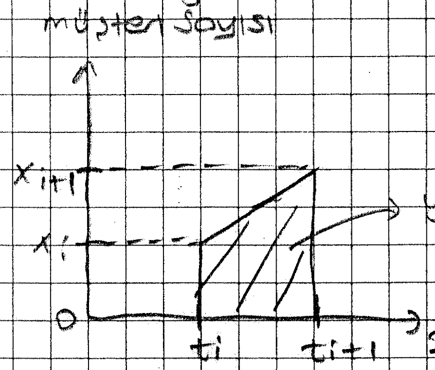
Örneğin;




Bir kuyrukta zaman periyodu süresince bekleyen müşteri sayısı

- Servis dolu ise 1, servis boş ise 0'dır.

## 3.) Zamana göre Ortalama Değerler Bir aralık boyunca değişebilir




yamuğun alanı → ortalama müşteri sayısını verir.

## M/M/1 Özellikleri

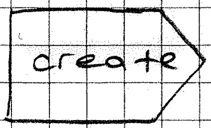
M.Ü

- Hizmet süresi üssel dağılımdır.
- Gelişler poisson dağılımdır.
- Kuyruk sonsuz uzunlukta
- Sistem gelişleri sonsuzdur.
- Tekli hizmet sunucu vardır.

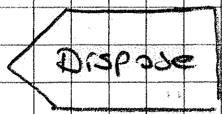
MERVE TELCEKEN

# BENZETİM VE MODELLEME (8. hafta)

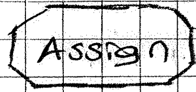
## ARENA:



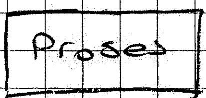

Varlıklar için başlangıç noktası tasarlar.



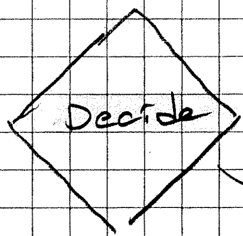

Varlıklar için son noktayı tasarlar.



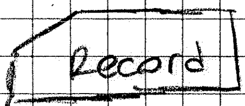

Değişkenlere, varlık tiplerine değer atanması için kullanılır.




İşlemlerimizin yapıldığı yerdir.
Zaman ayarları buradan yapılır.




Karar verme prosesidir.
Bir veya birden fazla durum olabilir.
Şart cümlecikleri buraya yazılır.

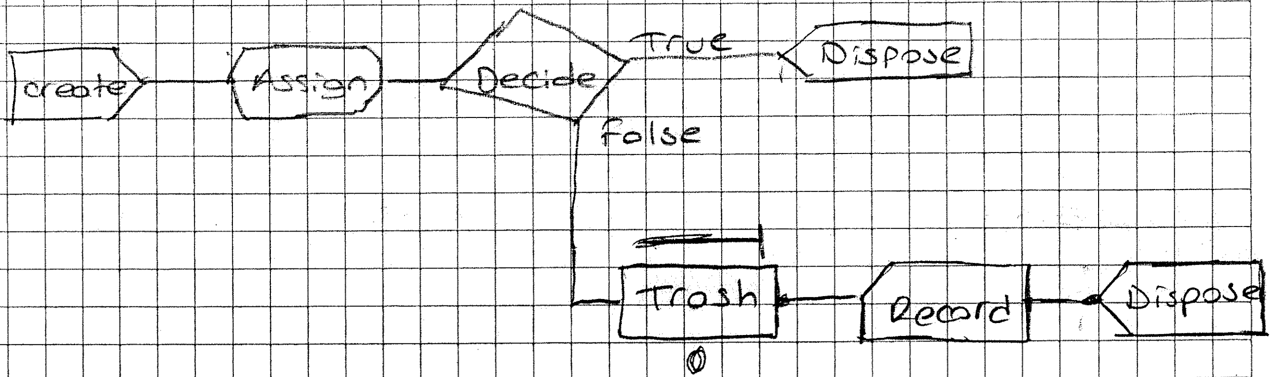



İstatistikleri biriktirmekte kullanılır. 'Count' tutulabilir.

**Diğer modüller:**

- Batch
- Hold
- Route
- Access
- Separate
- Match
- Station
- Convey
- Exit
- Request
- Transport
- Free

ÖRNEK: Bir erkek kuaföründe traş kuyruğunun simülasyonu yapılmıştır. Kuaföre gelen müşteriler sıraya girer. Müşteri sırası FIFO mantığına göre çalışmaktadır. Bir müşteri kuaföre girdiğinde eğer traş kuyruğu 3 kişi ise kuafördan çıkmaktadır. Traş kuyruğu 3 kişiden az ise müşteri kuyruğa girerek traş almaktadır.



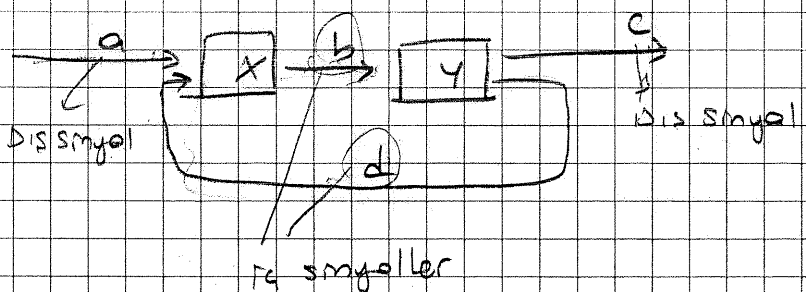
MERVE TELÇEKEN

# BENZETİM VE MODELLEME (9. hafta)

## HARİCİ SİNYALLER VE OLAYLAR

İç ve Dış sinyaller:

- Dış sinyaller, port olarak isimlendirilen giriş / çıkış'tan gelen veya giden sinyallerdir.
- İç sinyaller, sadece sistem modüllerini bağlar ve tamamen sistem içinden beslenirler.

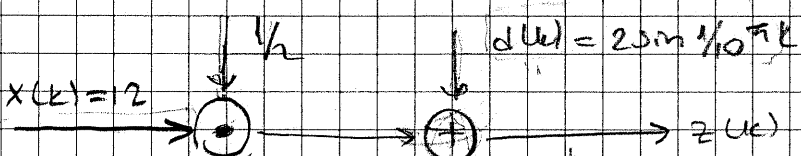



Bozuk Sinyaller:

- Genellikle böyle modeller olasılıksaldır. Bunlar özel bir örnek için tam olmayan olası yanlış sonuçlar verir, fakat bütün sistem için çok iyi sonuçlar verir.

ÖRNEK: $x(k) = 12$, transfer font $= 1/2$

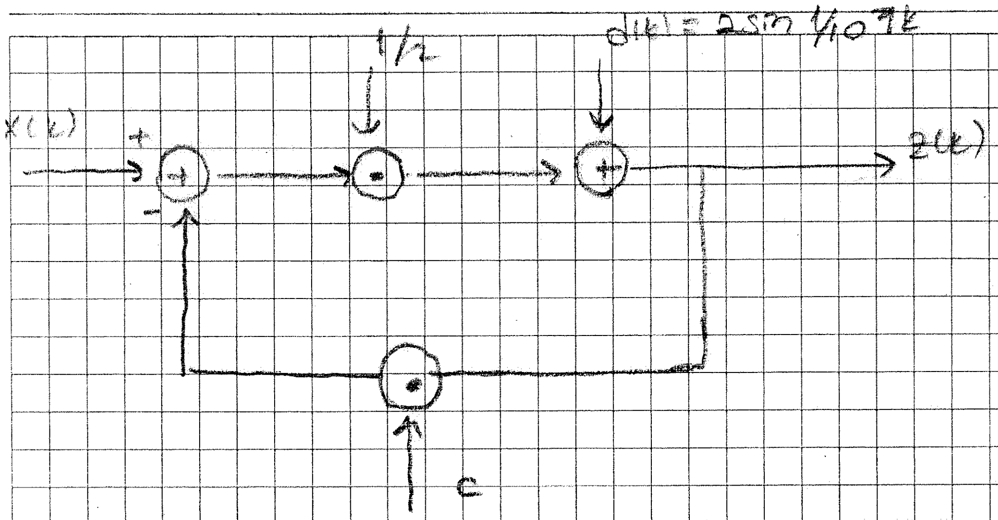
Deterministik bozukluk sinyali $d(k) = 2\sin(1/10\,\pi k)$




Eğer sistemde bozukluk yoksa $z(k) = 6$'dır.
Bozukluk $\pm 2$'lik bir değişim uygular.

$z(k) = 6 + 2\sin(1/10\,\pi k)$ sistemin çıkışı olur.

- Geri Besleme yapılarak bozukluğun etkisi azaltılabilir.

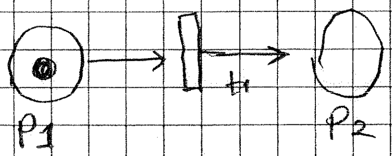



Temel fikr d(k)'yı azaltmak için c'yi bulmaktır.

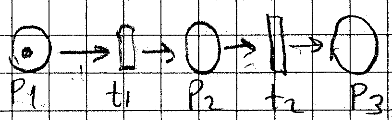
$$z(k) = d(k) + \frac{1}{2} \cdot \left[ 12 - c \cdot z(k) \right]$$

# PETRI AĞLARI

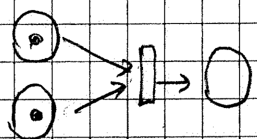
Çok işlemcili makineler aynı anda birden fazla işin paralel olarak işlemesini sağlar. Burada eş zamanlı modellemeye ihtiyaç duyulur. Bu yapılar petri ağlarıdır.



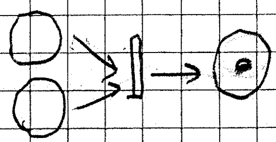

Sıralı Geçiş



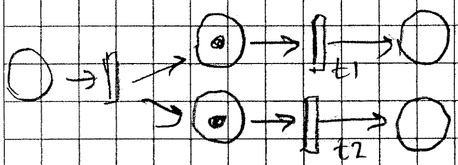

Senkronize Geçiş



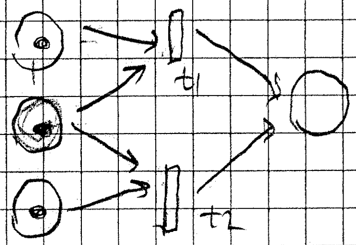
Birleştirme



Eş zamanlılık




Çakışma




ÖRNEK: Mesaj merkezli bir ağ olsun.

| Şehir Merkezi | Düğüm | Yerel Durum |
|---|---|---|
| Charlotte | 1 | $x_1$ |
| Kalamazo | 2 | $x_2$ |
| Paris | 3 | $x_3$ |
| London | 4 | $x_4$ |
| Standish | 5 | $x_5$ |

Örneğin x vektörü $[\overset{x_1}{2}, \overset{x_2}{0}, \overset{x_3}{1}, \overset{x_4}{3}, \overset{x_5}{0}]$ olduğunda charlott ilet mek için 2, Paris 1, Londra 3, Kalamazo ve Standish 0 mesaja sahiptir.

DURUM MAKİNALARI

ÖRNEK: Düzenli, bir saatin vuruşlarında A, B, C, D harfleriyle ardışık olarak çalışan bir sayısal ardışıl sistem düşünelim..

Başlangıç durumu $y(0) = A$

$d(k) = 1$ → Bulunduğu durumda kalır.

$d(k) = 2$ → Durum artan sırada listelenir.
(A, B, C, D, A, B, C, D ...)

$d(k) = 3$ → Durum azalan sırada listelenir.
(A, D, C, B, A, D, C, B, ...)

- Saat vuruşu ile değişebildiğinden senkrondur.
- Eğer olaylar düzensiz zamanlarda oluşuyorsa yani olaylar arası zaman değişken veya rastgele ise sistem asenkrondur.

Asenkron'da önceki durum bilinmez.

**ÖRNEK:** $Pr[D=1] = 0.1$, $Pr[D=2] = 0.3$, $Pr[D=3] = 0.6$

```
r = 10 * RND;
if r<1 then d=1;
if 1≤r<4 then d=2;
if r≥4 then d=3;
```

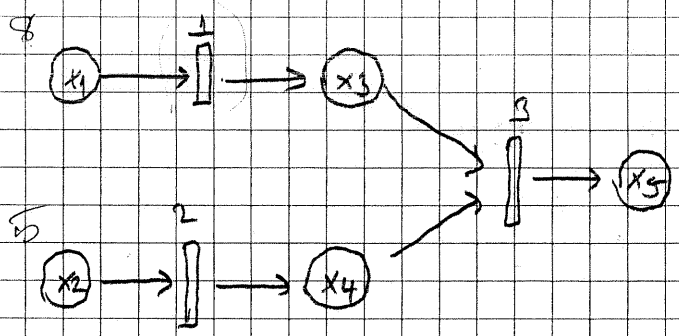
**ÖRNEK!** Şekilde gösterilen bir petri ağı düşünelim;

1. yer için 8 mesaj,

2. yer için 5 mesaj,

Diğerleri 0,

Bu modelde petri ağı modeli kullanarak geçici ve uzun vadeli bir davranış simüle ediniz.




$x(0) = [8, 5, 0, 0, 0]$

**Çözüm:**

1.) Tüm geçişler belirlenip, işaretlenmeli

$S = \emptyset$

If $x_1 > 0$ then $S = S \cup [1]$

If $x_2 > 0$ then $S = S \cup [2]$

If $x_3 > 0$ and $x_4 > 0$ then $S \cup [3]$

2.) Geçişlerden biri rastgele seçilmeli

'S' stringinden rastgele bir karakter döndüren algoritma:

```
function RAND(s)
    r = RAND;
    for i = 1 to LEN(S)
        if r < i/LEN(S) then
            y = VAL(MID(S, i))
            goto [I]
            end if
        next i
        y = 0
[I] RAND = y;
return
```

MID(S, i) → S'den tek bir karakter döndürür.

VAL → Bu değeri sayıya dönüştürür.

LEN → karakter sayısını gösterir.

y = RAND(S) ile rastgele geçiş sağlanır.

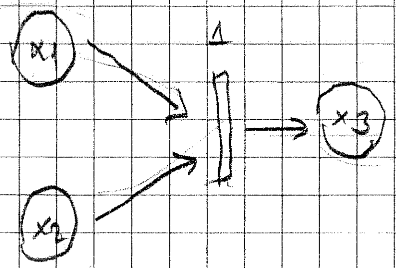
Keskin Color

```
n=20
k=0
read   x1, x2, x3, x4, x5
print  k, x1, x2, x3, x4, x5

for k=1 to n
      S=∅
      if x1>0 then S=S∪{1}
      if x2>0 then S=S∪{2}
      if x3>0 and x4>0 then S=S∪{3}
      y=RAND(S)
      if y=0 then "deadlock"
      if y=1 then x1=x1-1 : x3=x3+1
      if y=2 then x2=x2-1 : x4=x4+1
      if y=3 then x3=x3-1 : x4=x4-1 : x5=x5+1
      print k, x1, x2, x3, x4, x5
next k;
```

| k | x1 | x2 | x3 | x4 | x5 |
|---|---|---|---|---|---|
| 0 | 8 | 5 | 0 | 0 | 0 |
| 1 | 8 | 4 | 0 | 1 | 0 |
| 2 | 7 | 4 | 1 | 1 | 0 |
| 3 | 7 | 4 | 0 | 0 | 1 |
| 4 | 7 | 3 | 0 | 1 | 1 |
| 5 | 7 | 2 | 0 | 2 | 1 |
| 6 | 7 | 1 | 0 | 3 | 1 |

Keskin Color

## PETRİ AĞLARI İLE AND KAPISI



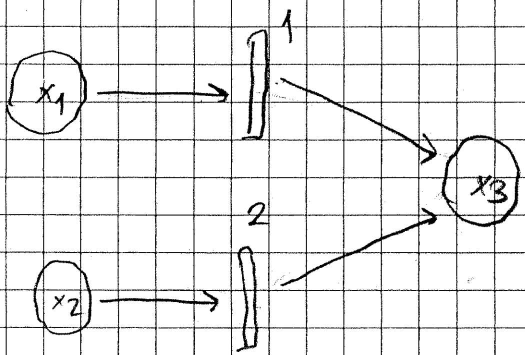

if $x_1 > 0$ and $x_2 > 0$ then $y = 1$

$S = \{1\}$

if $y = 1$ then

$x_1 = x_1 - 1$ ; $x_2 = x_2 - 1$ ; $x_3 = x_3 + 1$

## PETRİ AĞLARI İLE OR KAPISI




if $x_1 > 0$ then $y = 1$

if $x_2 > 0$ then $y = 2$

$S = \{1, 2\}$

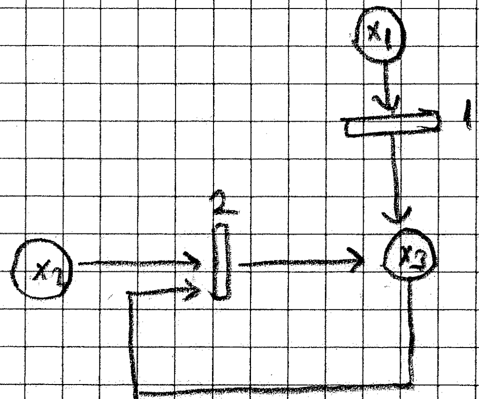
if $y = 1$ then

$x_1 = x_1 - 1$ ; $x_3 = x_3 + 1$

if $y = 2$ then

$x_2 = x_2 - 1$ ; $x_3 = x_3 + 1$

## PETRİ AĞLARI İLE HAFIZA YAPISI




if $x_1 > 0$ then $y = 1$

$S = \{1\}$;

if $y = 1$ then $x_1 = x_1 - 1$ ; $x_3 = x_3 + 1$

if $x_2 > 0$ and $x_3 > 0$ then

$y = 2$ $S = \{2\}$

if $y = 2$ then $x_2 = x_2 - 1$

KeshinColor

Keskin Color

* Başlangıç Değer Problemi =

...ullanılan değişkenlerin 1. türevi cinsinden eşitliğin elde edilmeye
...alışılması sağlanır.
...eğişken ismi verirken de; başlangıç değerleri verilen ifadelere ve
...ır. ..

Euler Yöntemi = 1. dereceden türev kullanılarak çözüm yapılır.

$$\boxed{\frac{x(t+h)-x(t)}{h} = f(x,t)} \rightarrow \text{1. derece türev gösterimi}$$

$x(t+h) = x(t) \cdot f(x,t) \cdot h$ (Sürekli zaman)

$x(k+1) = x(k) + h \cdot f(x(k), t_k)$ (Ayrık zaman)

Euler ile doğru çözüm elde edebilmek için; h uzunluğu azaltılabi-
H uzunluğunu azalmanın 2 olumsuz yönü var;
) Daha fazla hesaplama yapmak gerekir
) Veri gösteriminde bir sonraki makine sınırlamalarından dolayı h
... küçük görünebilir

$t0 = 1$
$x0 = 3$

$x_{gercek} = \frac{6}{5-3t^2}$ , $x(k+1) = x(k) + h \cdot x^2 t$ olan için Euler Matlab kodu;

```
t0=1; x0=3;
x=[x0]; t=[t0];
xg=(6/(5-3*t^2));
h= 0.05;
for k=1 to 5;  --> for k=1:5;
xyeni = x0 + h* x^2 *t0;
t0 = t0 +h;
x = [x xyeni];
t = [t t0];
x0= xyeni;
end
plot (t, xg, 'r')
hold on
plot (t, x)
```

**Sözde Kod:**

```
t=1 x=3 h=0.05
print t, x, h
for k=1 to 5
x = x + h.x^2 t
t = t+h
print t,x next k
```

**Kontrol Break Yapısı:**

— Her hesaplama sonucunda bir prosedürün sonuçlarının çıktısını almak yerine, hesapların periyodik sonuç çıktılarının alınması daha yararlı. (İç içe döngüyle oluşur bu yapı)

```
t=1 x=3 print t, x
for i=1 to n
  for j=1 to m
    x= x + h x^2 t
    t= t + h
  next j print t, x next i
```

→ Hesaplamaları iç döngü boyunca yapıp yazdırır. Yeni değer için dış döngüyle —1—
...ken n x n defa hesaplama yapılır.

Taylor yöntemi ...
rev için değil, 2 veya daha fazla derece türevde kullanılır.

$$\boxed{x(t+h) = x(t) + h.\,f(x,t) + \underbrace{\frac{h^2}{2}\,f^{\cdot}(x,t)}}$$

2. derece için h'ın integrali
x'in türevi

. derece türev;

$$\boxed{\ddot{x} = \frac{d}{df}\,.\,f(t,x) = \frac{df}{dt} + \frac{df}{dx}\cdot\frac{dx}{dt} = \frac{df}{dt} + \frac{df}{dx}\cdot f(x,t)}$$

. Euler'e göre daha doğru sonuç verir.

Runge-Kutta yöntemi =

Genellikle 4. derece denklemlerde uygulanan yöntemdir. 4. derece denklemler için Taylor karışık olabilir. Bunun yerine runge-kutta kullanılır.

Klasik bir Runge-Kutta Algoritması;

$$k_1 = f(t_k, x(k))$$

$$k_2 = f\left(t_k + \frac{1}{2}h,\ x(k) + \frac{1}{2}hk_1\right)$$

$$k_3 = f\left(t_k + \frac{1}{2}h,\ x(k) + \frac{1}{2}hk_2\right)$$

$$k_4 = f(t_k + h,\ x(k) + h.k_3)$$

$$t_{k+1} = t_k + h$$

$$x(k+1) = x(k) + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

Yüksek Dereceden Sistemler;

İki tane birinci dereceden denklem sistemini şöyle ifade ederiz;

$$\boxed{\begin{aligned}\dot{x} &= f(t,x,y)\\ \dot{y} &= g(t,x,y)\end{aligned}}$$

$x$ değişkenini $k_1, k_2, k_3, k_4$

$y$ değişkenini $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ ile ilişkilendirerek Runge-Kutta uygulayalım.

$$k_1 = f(t, x, y)$$

$$\lambda_1 = g(t, x, y)$$

$$k_2 = f\left(t + \frac{h}{2},\ x + \frac{h}{2}k_1,\ y + \frac{h}{2}\lambda_1\right)$$

$$\lambda_2 = g\left(t + \frac{h}{2},\ x + \frac{h}{2}k_1,\ y + \frac{h}{2}\lambda_1\right)$$

$$k_3 = f\left(t + \frac{h}{2},\ x + \frac{h}{2}k_2,\ y + \frac{h}{2}\lambda_2\right)$$

$$\lambda_3 = f\left(t + \frac{h}{2},\ x + \frac{h}{2}k_2,\ y + \frac{h}{2}\lambda_2\right)$$

$$k_4 = f(t + h,\ x + hk_3,\ y + h\lambda_3)$$

$$\lambda_4 = g(t + h,\ x + \lambda k_3,\ y + h\lambda_3)$$

$$x_{k+1} = x_k + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$y_{k+1} = y_k + \frac{h}{6}(\lambda_1 + 2\lambda_2 + 2\lambda_3 + \lambda_4)$$

dışından daha önceki yöntemlerle çözülebilir.

$x(t)$ ve $y(t)$ sırasıyla av ve avcı popülasyonunun büyüklükleri olsun.

Av-avcı popülasyonunun büyüme oranı aşağıdaki gibidir;

$$\boxed{\dot{x} = \alpha_1 x\left(1 - \frac{y}{\beta_1}\right)} \qquad \boxed{\dot{y} = \alpha_2 y\left(-1 + \frac{x}{\beta_2}\right)}$$

$\alpha_1, \alpha_2, \beta_1, \beta_2$ her popülasyon için pozitif kopozite sabitleridir.

Lotka-Volterra modelinin benzetimi için Euler yöntemi uygulanabilir.

Euler hatırlayalım;

$$\dot{x} = \frac{x(t+h) - x(t)}{h}$$

$$x(t+h) = \dot{x}h + x(t)$$

$$x(k+1) = x(k) + h.\dot{x}(k)$$

→ $\dot{x} = \alpha_1 x\left(1 - \frac{y}{\beta_1}\right)$ için euler uygularsak;

$$x(k+1) = x + h.\alpha_1 x\left(1 - \frac{y}{\beta_1}\right)$$

$$\boxed{x(k+1) = x\left(1 + \alpha_1 * h - \alpha_1 h y/\beta_1\right)}$$

→ $\dot{y} = \alpha_2 y\left(-1 + \frac{x}{\beta_2}\right)$ için euler uygularsak;

$$y(k+1) = y + h.\alpha_2 y\left(-1 + \frac{x}{\beta_2}\right)$$

$$\boxed{y(k+1) = y\left(1 - \alpha_2 * h + \alpha_2 h x/\beta_2\right)}$$

Lojistik Sistem

$$\boxed{\dot{x} = \lambda . x\left(1 - \frac{xt}{xm}\right)}$$

Malthousian

$$\boxed{x(t) = x_0 . e^{\lambda t}}$$

→ Lojistik Lotka'dan farkı;
- belli bir noktaya kadar artması
- başlangıç şartlarından bağımsız olması

→ Lojistik denklemde $x(t)$ küçükse; Malthusiandır.

→ $[0, t_n]$ aralığında $m \times n$ hesaplama yapılır.
$h$ → adım sayısı 0.02 ise;

$$\boxed{m = (t_n - t_0)/n h}$$

└→ örnek sayısı

Sözde Kod;

```
read t, x, y
print t, x, y
for i=1 to n
  x1 = x(1+α1h-α1hy/β1)
  y1 = y(1-α2h+α2h.x/β2)
  x = x1   y = y1
  t = t+h
  print t, x, y  next i
```

Kontrol Break kullanarak Sözde Kod

```
read t, x, y  print t, x, y
for k=1 to n
  for i = 1 to m
    x1 = x(1+α1h-α1hy/β1)
    y1 = y(1-α2h-α2hx/β2)
    x = x1   y = y1   t = t+h
  next i
  print t, x, y
next k
```

latlab Kodu :  ( 10 )  ( 25 )  y(0) = 5

```
function [t, x, y] = lotkavolterra (h, x0, y0, n)
  x0 = 10;    y0 = 5;
  x = [x0];
  y = [y0];
  t = 0;
  for i = 1 : n
     for j = 1 : 100
         xyeni = x0 * (1 + 3*h*y0/10);
         yyeni = y0 * (1 - 1,2*h + 1,2*h*x0/25);
     x0 = xyeni;
     y0 = yyeni;
     t = t + h;
    x = [x ; xyeni];
    y = [y ; yyeni];
    end
 end
 plot (x)
 hold on
 plot (y, 'r-')
```

→ $h = 0.001$ alalım.
[0,5] aralığında 5 birim vardır ve herbir zamanda 10 örnek yeterlidir.
Dolayısıyla toplam örnek = 50'dir.

$$m = (5-0) / 50 * 0,001$$

m = 100 güncelleme olur

Toplam güncelleme = 5000 olur.

$$(k+1) = x + 3x\left(1 - \frac{y}{10}\right) \cdot h \qquad y(k+1) = y + 1,2y\left(-1 + \frac{x}{25}\right) h$$

$$\text{yeni} = x\left(1 + 3h - 3hy/10\right) \qquad y_{yeni} = y\left(1 - 1,2yh + 1,2hx/25\right)$$

(*) Lotka Volterra Matlab kodunda; genelde kontrol break kullan.

Girişi Olan — Girişi Olmayan (OTONOM)

**Girişi Olan:**

**Zaman Odaklı** — Girişler eş zamanlıdır. Deterministik zaman fonksiyonu ise zaman odaklı.

**Olay Tabanlı** — Girişler eş zamanlı değildir.

**Girişi Olmayan (OTONOM):**

**Kararlı** — Kısa bir geçiş evresinden sonra çıktı 0'a yaklaşır.

**Kararsız** — Sınırlama olmaksızın doğal tepki artar.

**Marjinal** — Tepki periyodik ve sınırlı.

---

**MALTHUSIAN MODEL ;**

$$\boxed{x(t) = x_0 \cdot e^{\lambda t}}$$

$$\boxed{\dot{x} = \lambda \cdot x\left(1 - \frac{x_t}{x_m}\right)}$$

→ Sistemin taşıdığı maksimum popülasyon büyüklüğü $x_m$ taşıma kapasitesidir. $x(t)/x(m)$ doluluk oranı

↓

Lojistik Denklem denir. $x(t)$ küçükse $\left(1 - \frac{x_t}{x_m} \cong 1\right)$ olduğundan Malthusian

## * MONTE CARLO BENZETİMİ

- Olasılık teorisi üzerine kurulu bir sistemdir.

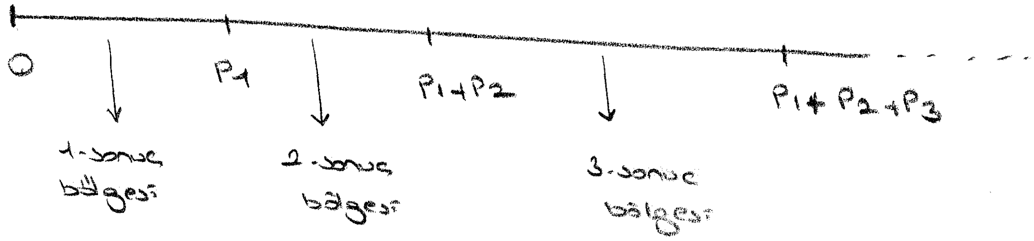
İstatistiksel veya matematiksel tekniklerle bir deneyi, rassal sayıları defalarca kullanarak simülasyon edilip çözmek esastır.

+ Genel olarak monte carlo matlab kod;

```
function sonuc=montecarlo(n)
  cember=0;
  sayac=0;
for i:1 to n
  x=rand ;
  y=rand ;
  if ( x^2 + y^2 < 0.8)
   cember=cember+1;
end
  sayac =sayac+1;
end
sonuc=cember/sayac;
```

- Function tanımlama
- Sayaçları tanımlama
- Döngü (for)
- Rastgele sayıları üretme
- Hesaplama + karşılaştırma
- Sayaçları arttırma
- Sonuç

1 = 1. olasılık
2 = 2. olasılık
3 = 3. olasılık




) Monte Carlo benzetiminde zaman faktörü önemli değildir. Zamana bağlı olmadığından; statik benzetim modelidir.

---

RASTGELE SAYI ÜRETEÇLERİ — Doğa ve mühendislik sistemleri kesin olarak tahmin edilemezler.

sistemler genellikle görültü içerirler. Bir sistemi gerçekçi olarak modelleyebilmek için rastgeleliğin bir derecesi modele eklenmelidir. Bunun için rastgele sayı üreteçleri kullanılır.

Rastgele sayı üreteçlerinden istenen özellikler;

1) Rassallık
2) Büyük Periyod
3) Yeniden üretilebilirlik
4) Hesaplama etkinliği

Bu üreteçler 2'ye ayrılır → Tek Düze
→ Tek Düze Olmayan

Tek Düze Dağılımlı Sayı Üreteçleri
Orta kare yöntemi
Doğrusal Eşlik yöntemi (LCG)

Tek Düze Olmayan
Ters Dönüşüm Metodu
Konvolüsyon metodu
Ret Metodu

Orta Kare Yöntemi: Genellikle tek sayı seçilir, kaç haneli sayı üretilecek o kadar haneli başlangıç değeri seçilir.

Dezavantajları;
Üretilen sayılar rastgele olmayabilir (aynı sayılar olabilir)
Dizinin tekrar sayısının ne kadar olacağı önceden kestirilemez
Dizi dejenere olmuş olabilir.

* Doğrusal [illegible]

- Deterministiktir.
- Maksimum üretilen rastgele sayı sayısı = mod kadardır.

$$\boxed{\begin{array}{l} z_{k+1} = (a \cdot z_k + c) \bmod m \\ u_0 = \dfrac{z_0}{m} \rightarrow 1.\ \text{rassal sayı} \end{array}}$$

LCG'nin periyodik olması için (tam periyoda sahip olması için);

- a ile c asal olmalı.
- m sayısının bölünebildiği bütün asal sayılara $(a-1)$'de bölünmeli.
- Eğer m 4'e bölünüyorsa $(a-1)$'de 4'e bölünmeli.

} Hull–Dobel Teoremi

LCG'lerin donanımsal olarak ifade edilmesi;

- Shiftregisterlar ile gösterilir. m (mod) → 2'nin kuvveti olarak yazıldığında kuvvet ne kadarsa shiftregister o kadar bit olur.

<u>Mesela</u>; $mod = m = 16 \rightarrow 2^4 \Rightarrow$ Shiftregister 4 bit

$a=5$
$c=3$
$z_5 = 5$ olsun

$$z_6 = 5.5 + 3 = 28 \rightarrow [\cancel{1}\,1100]$$
$$28 \bmod (16) = 12 \rightarrow [1100] \text{ olur.}$$

<u>LCG Matlab Kod</u>

$a=5$
$c=3$
$m=16$
$z_0=7$ için

$$z_{k+1} = (5z_0 + 3) \bmod 16$$
$$u = \frac{z}{m} \rightarrow \text{rassal sayı}$$

```
a=5; c=3; z0=7; m=16;
z=[z0]; u0 = z0/m; u=[u0];
   for i=1:n
 zyeni = mod((5*z0+3),16);
 uyeni = zyeni/m;
 z0=zyeni;
 u0=uyeni;
 z=[z zyeni]; u=[u uyeni]; end
 plot(u);
```

- Belli bir sayıda aralık vardır. Bu aralıkların hepsinde eşit miktarda sayı atılıp üretilmediği test edilir. Aralıkların arası mesafe eşit.

Beklenen ve gözlenen frekans değerlerine bakılır.

Frekans Dağılım Tablosu kullanılır.

n = üretilen toplam sayı

m = aralık sayısı

$$\boxed{e_k = \frac{n}{m}} \rightarrow \text{beklenen frekans}$$

$f_k \rightarrow$ gözlenen frekans

$$\boxed{X^2 = \frac{m}{n} \cdot \sum_{k=1}^{m} (f_k - e_k)^2}$$

→ değeri hesaplanır. $\boxed{v = m-1}$ ⇒ bağımsızlık derecesi

Tablodan gidip v değerindeki güven aralığında yazan değere bakarız. Mesela $v=3$'de güven aralığı %95 ise $1-0.95 = 0.05$ olan yerdeki değere bakarız. Bu değer = $X_c^2$ dir.

$X_c^2 > X^2$ ise uniform dağılım vardır.

*** Ters Dönüşüm Metodu:**

Verilen sistemi u'ya eşitleyip ($u = f(x)$ gibi) denklemlerin integralini alıyoruz. En son sınırları yazarken; $f^{-1}(u) =$ diye belirtip sınırları ve ifadeleri u cinsinden yazıyoruz.

Algoritması:
1) $u \sim U(0,1)$
2) sınırların yazılması
3) return

*) $f(x)$ fonksiyonu için $[a,b]$ aralığında beklenen frekans;

$$\boxed{\text{Beklenen frekans} = n \cdot \int_a^b f(x)\,dx}$$

n = üretilen sayı sayısı

- Bağımsız, rastgele üretilen $(x_1, x_2 \ldots x_n)$ değişkenlerinin toplamı x'e eşittir.

- Her $x_i$ için $i=1,2,\ldots n$ yoğunluk fonksiyonu $f_i(x)$ aynı ise, x'in yoğunluk fonksiyonu olan $f(x)$, herbir n tabanlı yoğunluk fonksiyonları için konvolüsyondur.

Yani;

$$X = \sum_{i=1}^{n} x_i \quad \text{ise} \quad f(x) = \underbrace{f_1(x) \otimes}_{\text{Konvolüsyon}} f_2(x) \otimes \ldots \otimes f_n(x)$$

- Konvolüsyon metodu şöyle gösterilir;

$$\boxed{f_1(x) \otimes f_2(x) = \int_{-\infty}^{\infty} f_1(\lambda) \cdot f_2(x-\lambda)\, d\lambda}$$

- Rastgele değişken kendini, $X = \sum_{k=1}^{n} x_k$ n tane aynı şekilde dağılmış değişkenine ekleyerek bulan.

Konvolüsyon metodu için özel bir durum; m-Earlang dağıtımıdır.

m-Earlang Dağıtımı

- n adet aynı şekilde dağılmış rastgele değişkenin toplamı olarak tanımlanır. Bu dağıtımın ortalamı $\mu$ ile gösterilir (Rastgele üretilen sayıların toplamının ortalaması)

$$\boxed{\mu = \frac{m}{\lambda}} \qquad \boxed{m = \sum_{k=1}^{m} x_k}$$

$\lambda$: exponansiyel dağıtımın ortalamasının matematiksel karşıtıdır.

> m-earlang dağıtımı için yoğunluk fonksiyonu;

$$\boxed{f(x) = \frac{\lambda^m \cdot x^{m-1} \cdot e^{-\lambda x}}{(m-1)!}, \quad x > 0}$$

> Rastgele bir m-earlang değişkeni oluşturma algoritması

```
x=0
for k= 1 to m
    x = x - μ ln(RND)/m
next k
print x
```

Ör 1/ $\frac{dx}{dt} = \cos 2t + \sin 2t \qquad x(0) = 2$

$t0 = 0$

$x0 = 0$

---

Ör 2/ $\dot{x} = 2x^2 + \cos t \qquad x(0) = 4 \qquad h = 0.1$ alarak $[0-10]$ aralığında

$f(x,t) = 2x^2 + \cos t$

$t0 = 0$

$x0 = 4$

$x(t+h) = x(t) + h.\, f(x,t)$

$x(k+1) = x(k) + 0.1.\, f(x(k), t_k)$

$\frac{dx}{dt} = 2x^2 + \cos t$

$dx = 2x^2 dt + dt\cos t$

$\int \frac{dx}{2x^2 dt} = dt\cos t$

$\int_4^x \frac{dx}{2x^2} = \int_0^t dt\cos t$

$-\frac{2}{x}\Big|_4^x = \sin t \Big|_0^t$

$-\frac{2}{x} + \frac{1}{2} = \sin t$

$-\frac{2}{x} + 1 = 2\sin t$

$\boxed{x = \frac{-2}{2\sin t - 1}}$ → gerçek çözüm

---

$\int \cos cx\, dx = \frac{1}{c}\sin cx \qquad \int \sin cx\, dx = -\frac{1}{c}\cos x$

$\int \cos x\, dx = \sin x \qquad \boxed{\int \frac{dx}{2x} = \frac{1}{2}\ln x}$

$\int \sin x\, dx = -\cos x$

$\sin 0 = 0$

$\cos 0 = 1$

- Net bir yoğunluk fonksiyonunun bulunamadığı durumlar için kullanılır. Sistem için geçmiş kayıtlar tespit edilir ve bu değer ile aynı istatistiklere sahip rastgele değişkenler oluşturulur.

Süreç 2 aşamalıdır.

↳ Birinci aşama: Sürekli yoğunluk fonksiyonunu elde etmek.

$$f(x) = \begin{cases} 0, & x < x_1 \\ \dfrac{i-1}{n-1} \times \dfrac{x - x_i}{(n-1).(x_{i+1} - x_i)}, & x_i \leq x < x_{i+1}, \quad i = 1, \dots n-1 \\ 1, & x \geq x_n \end{cases}$$

↳ İkinci aşama: $F^{-1}(x)$ bulmaktır.

$F(x)$ lineer olduğundan $\underline{X = F^{-1}(RND)}$

Ör/ Bilinmeyen bir süreçten alınan rastgele değişkenler kümesi $\{1, 2, 4, 5, 7, 7, 8\}$ olsun.

Çözüm:

$x < x_1$ için $F(x) = 0$ olduğundan $x < 1$ için $F(x) = 0$ $\longrightarrow x < 1$

$1 \leq x < 2$ için $F(x) = \dfrac{1-1}{7-1} * \dfrac{(x-1)}{(7-1).(2-1)} = \dfrac{1}{6}(x-1)$ $\longrightarrow 1 \leq x < 2$

$2 \leq x < 4$ için $F(x) = \dfrac{1}{12}x$ $\longrightarrow 2 \leq x < 4$

⋮

$\frac{1}{6}+$

$F^{-1}(x)$'lere bakalım

$$F^{-1}(x) = \begin{cases} \text{tanımsız}, & x < 1 \\ 6x+1, & 0 \leq x < \frac{1}{6} \\ 12x, & \frac{1}{6} \leq x < \frac{1}{3} \\ \vdots \end{cases}$$

$60x^3(1-x)^2 = f(x)$

$180x^2(1-x)^2 + 60x^3 \cdot -2(1-x)$

$-120x^3(1-x) + 180x^2(1-x^2) = 0$

$\cancel{180}^{3}x^2(1-x^2) = \cancel{120}^{2}x^3(1-x)$

$3-3x^2 = 2-2x$

$3x^2-2x-1=0 \qquad x=1 \qquad x=-\frac{1}{3}$

$3x \quad +1$

$x \quad -1$

$180x^2(1-x^2)-120x^3(1-x)$

$180x^2(1-x).(1+x)-120x^3(1-x)$

$x^2(1-x)\,[180.(1+x)-120x]=0$

$x=0$

$x=1$

$180+180x-120x=0$

$x=\frac{1}{3}$

~~1) f(x)~~

1) $f'(x)=0$

2) $f(x)$

$c=\int f(x)dx$

3) $r(x)=\frac{f(x)}{c} \longrightarrow 0<x<1$

4) $\int_0^x r(x)dx$

$$f(x)=\begin{cases} x & 0\le x\le 1 \\ 2-x & 1<x<2 \\ 0 & dd \end{cases}$$

$f(x)=x$

$c=\int_0^2 f(x)dx \Rightarrow c=2$

$r(x)=\frac{f(x)}{2}=\frac{x}{2} \rightarrow$

$\int_0^x \frac{x}{2}dx$

$\frac{1}{2}\cdot\frac{x^2}{2}$

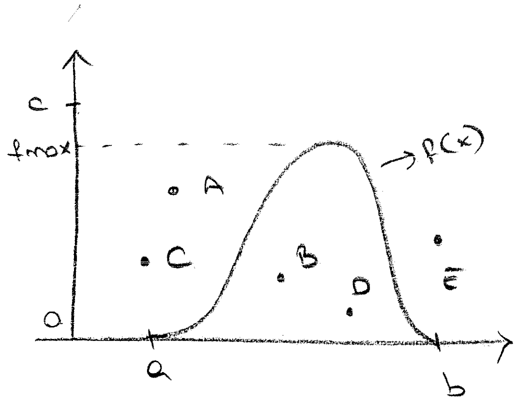
$fx=1$

$\frac{x}{2}=u \qquad x=2u$

. Sürekli ve sınırlı olan bir f(x) olasılık yoğunluk fonksiyonundan rassal değişken üretmek için kullanılan yöntemdir.

Sürekli bir x rassal değişkeni için;

$\boxed{0 \le f(x) \le f_{max}}$ $\boxed{a \le x \le b}$ dir.




— Amacımız verilen fonksiyonun altında kalan noktaları bulmak.

— $\boxed{c = f_{max}}$ diyebiliriz.

Algoritması
for i=1 to n
[1] x = a + (b−a) * RND → x ekseninde 0 ile b aralığında sayı üretme
y = c * RND → y ekseninde fmax'a kadar
if (y>f(x)) then goto [1]
print x,y
next i

Reddetme Tekniğinin Adımları

) Öncelikle bir t fonksiyonunun tanımlanması gerekir.

Her x için $t(x) \geq f(x)$ olmalıdır.

$$\boxed{c = \int t(x)dx \geq \int f(x)dx = 1} \rightarrow \boxed{c = \int t(x)dx}$$

) $\boxed{r(x) = \frac{t(x)}{c}}$ bir olasılık yoğunluk fonksiyonudur.

) r(x) olasılık yoğunluk fonksiyonundan y rassal değişkenlerini şöyle üretebiliriz;

$u_1 \sim u(0,1)$ ; y = x [1]

) $u_2 \sim u(0,1)$ üret (y'den bağımsız)

) $u_2 \le \frac{f(y)}{t(y)}$ ise x=y and return

değilse goto [1]

) t(x) = a olsun

$$\left.\begin{array}{l} t(x)=a \\ c = \int_a^b t(x)dx = a.(b-a) \end{array}\right\} \quad \begin{array}{l} t(x) = a \\ c = a.(b-a) \end{array} \Rightarrow r(x) = \frac{t(x)}{c} = \frac{1}{b-a}$$

$$r(x) = \begin{cases} \frac{1}{b-a} & a \le x \le b \\ 0 & dd \end{cases}$$

5) Ters dönüşüm kullanarak $r(x)$ fonksiyonundan (a, b) aralığında
eşitken üretilebilir.

$$R(x) = \int_a^y r(x)\,dx = u$$

$$= \int_a^y \frac{dx}{b-a} = \frac{x-a}{b-a} = u \Rightarrow \boxed{y = u.(b-a)+a}$$

Algoritma

1) $u_1 \sim U(0,1)$ üret $y = a + (b-a).u$ → x noktaları

2) $u_2 \sim U(0,1)$ üret → y noktası

3) $u_2 \leq \frac{f(y)}{t(y)}$ ise $x=y$ Return → elde edilen fonksiyon verilen fonksiyon altında kalması durumunda $x=y$

değilse Goto 1. adım

12/

$$f(x) = \begin{cases} 60x^3(1-x)^2 & 0<x<1 \\ 0 & dd \end{cases}$$

Çözüm

Önce bir $t$ fonksiyonu olmalı. $c = \int t(x)dx$ olduğundan c noktasını bilmemiz lazım. $c = f_{max}$ olduğundan ve bir fonksiyonun 1. türevini sıfır yapan nokta maksimum nokta olduğundan $f^{1}(x) = 0$ hesaplarız

$$f'(x) = 60x^2(1-x).(3-5x) = 0 \qquad \left.\begin{matrix} x=0 \\ x=1 \\ x=0.6 \end{matrix}\right\} \text{Çözüm kümesi}$$

değerlerinin hepsini $f(x)$ fonksiyonunda yazarsak max değeri $x=0.6$ için
dır. $\boxed{f(0.6) = 2.0736}$

O halde $c = 2.0736$'dır

$$t(x) = \begin{cases} 2.0736 & 0<x<1 \\ 0 & dd \end{cases}$$

$$c = \int_0^1 2.0736\,dx \Rightarrow \boxed{c = 2.0736}$$

$r(x)$ olasılık yoğunluk fonksiyonu gereklidir

$$r(x) = \frac{t(x)}{c} = \frac{2.0736}{2.0736} = 1 \Rightarrow r(x) = \begin{cases} 1 & 0<x<1 \\ 0 & dd \end{cases}$$

) Ters dönüşüm ile r(x)'den sayılar üretme;

$$R(x) = \int_0^x r(x)dx = \int_0^x 1dx = \boxed{x = u}$$

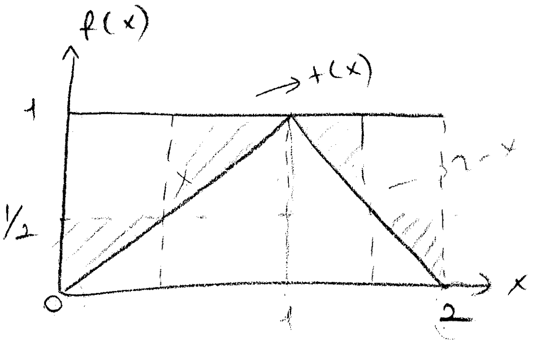
Algoritma

1) $u_1 \sim U(0,1)$ üret $y = x = u_1$

2) $u_2 \sim U(0,1)$ üret

3) $u_2 \leq \frac{60y^3(1-y)^2}{2.0736}$ ise $x=y$ return

Değilse 1. adıma git




$$f(x) = \begin{cases} x & 0 \leq x \leq 1 \\ 2-x & 1 < x \leq 2 \\ 0 & dd \end{cases}$$ ?

Reddetme tekniğini uygulayalım.

) c noktası bulunmalı.

$$c = \int t(x)dx \rightarrow c = \int_0^2 1dx \rightarrow \boxed{c = 2}$$

r(x) olasılık yoğunluk fonksiyonu bulunmalı.

$$r(x) = \frac{t(x)}{c} = \frac{1}{2} \qquad r(x) = \begin{cases} 1/2 & 0 \leq x \leq 2 \\ 0 & dd \end{cases}$$

R(x) ile (yani r(x)'e ters dönüşüm uygulanması.)

$$R(x) = u = \int_0^x r(x)dx \rightarrow u = \frac{1}{2}\int_0^x dx \rightarrow u = \frac{x}{2} \rightarrow x = 2u$$

Algoritması

1) $u_1 \sim U(0,1)$ üret $y = x = 2u$

2) $u_2 \sim U(0,1)$ üret

3) $y \leq 1$ ise $u_2 \leq \frac{y}{1} \Rightarrow x = y$ $\quad \frac{f(x)}{t(x)}$

$y > 1$ ise $u_2 \leq \frac{2-y}{1} \Rightarrow x = y$ ve Return

4) Değilse Goto 1. adım

?

```
for i=1 to n
[1]  x = 2*RND
     y = RND
     if y > f(x) then
     goto [1]
     print x
     next i
```

## — BENZETİM ÖRNEK SORULAR —



**Ör/** $y' = -2x - y$ , $y(0) = -1$ $h = 0.1$ olduğuna $y(0.5)$ bulun.

→ $y(k+1) = y(k) + h \cdot f(y, x)$

$x(0) = 0$
$y(0) = -1$

$y(1) = y(0) + h \cdot -2(0-1)$

$y(1) = -1 + 0{,}2 = -0{,}8$

$y(2) = y(1) + h \cdot f(y, x)$

⋮

$y(5)$'e kadar gideriz...

Gerçek çözüm

$$\frac{dx}{dy} = -2x - y$$

$$-\frac{dx}{2x+y} = dy$$

Matlab kodu

```
x0=0;   y0=-1;
x=[x0];  y=[y0];
xg= [ .... ];
h=0.1;
for i = 1:5
  yyeni= y0 + h * (-2 * x - y);
  x0 = x0+h;
  x = [x  x0];
  y = [y  yyeni];
  y0 = yyeni;
end
plot (x,y)
```

**Ör/** $y' = (x-y)/2$ $y(0) = 1$ için $[0, 3]$ aralığında

$h = 1$
$h = 1{,}2$
$h = 1{,}4$
$h = 1{,}8$

} Taylor ile çöz.

* Ters dönüşüm → verilen fonksiyonun integralinin alınıp 0'ya eşitlenmesi.
- Alt sınırı verilen fonksiyonun sınırları arasındaki alt sınır olacak şekilde, üst sınırı $x - y$) falan olur.

* $f(x)$ fonksiyonu için $[a\ b]$ aralığında beklenen frekans sonuçları.

$$\boxed{\text{beklenen frekans} = n \cdot \int_a^b f(x)\,dx}$$

$n$ → Üretilen sayı sayısı

* Yoğunluk fonksiyonundan bahsedilen metodlar;

- Konvolüsyon metodu
- Keyfi rastgele değişkenlerin üretimi

:) Konvolüsyon metodunda; rastgele üretilen m adet $x_1, x_2 ... x_m$ toplamı verilmektedir. Her $x_i$ için yoğunluk fonksiyonu aynıysa bunların herbiri için yoğunluk fonksiyonunun konvolüsyonudur.

$$X = \sum_{i=1}^{m} x_i \quad ise \qquad f(x) = f(x_1) \circledast f(x_2) \circledast \cdots \circledast f_m(x)$$

konvolüsyon

- Konvolüsyon metodu için bir başka yöntem m-erlang dağılımıdır.

m tane $x_1, x_2, \ldots x_m$ için $x_1 + x_2 + \ldots + x_m = m$ olmak üzere

$\mu$ bu m tane sayının ortalaması olsun üzere

$$\boxed{\mu = \frac{m}{\lambda}}$$

Yoğunluk fonksiyonu $\longrightarrow$ $\boxed{f(x) = \dfrac{\lambda^m \cdot x^{m-1} \cdot e^{-\lambda x}}{(m-1)!}}$

:) Keyfi rastgele değişkenlerin üretiminde;

- Net bir yoğunluk fonksiyonu bulunmadığı durumlarda kullanılır
- Süreç 2 aşamalıdır.

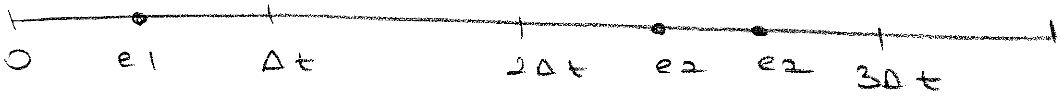
1) Birinci aşama; sürekli yoğunluk fonk. elde edilir

$$f(x) = \begin{cases} 0, & x < x_1 \\ \dfrac{i-1}{n-1} + \dfrac{x - x_i}{(n-1)\cdot(x_{i+1} - x_i)}, & x_i \leq x < x_{i+1} \end{cases}$$

2) İkinci aşama; $f^{-1}(x)$ bulunur. $n$ = Toplam üretilen rastgele sayı sayısı

# Sabit artışlar ile zaman ilerletme;

- Bu yaklaşımda; benzetim saati önceden belirlenen $\Delta t$ zaman kadar arttırılır.
- Her $\Delta t$ artış sonrası o aralıkta herhangi bir olayın olup olmadığı kontrol edilir. Eğer olay/olaylar olmuşsa, olayların aralığın sonunda olduğu kabul edilir ve sistem durumu güncellenir.




$0 - \Delta t$ aralığında, e1 olayı olmuş. $e_1 = \Delta t$'de olmuş gibi ele alınır.
$\Delta t - 2\Delta t$ aralığında; olay yok ama kontrol yapılır yinede
$2\Delta t - 3\Delta t$ aralığında; $e_2$ ve $e_3$'de $3\Delta t$'de olmuş kabul edilir. Ancak bu olaylardan hangisinin hangi sırada olduğu konusunda bir kural bulunmalıdır. Bu yaklaşımın 2 dezavantajı vardır;

1) Aynı anda oluşmuş gibi kabul edilen olaylardan hangisinin önce olduğuna karar vermede hata olur.
2) $\Delta t$ küçük alınarak hata azaltılabilir fakat bu seferde modelin çalışma zamanı artar.

→ Bu sebeple KOB'de bu yaklaşım sık kullanılmaz.

# ~~En yakın olay zamanı ile zaman ilerletme;~~

— KESİKLİ OLAY BENZETİMİ —

- Kesikli olay benzetimi; sistemin zamana göre benzetimidir. Zaman içinde kesikli noktalarda bir olay olur ve sistem durumunu değiştirir.

KOB'da kullanılan İstatistiksel Sayaçlar;
Zaman periyodu sürecinde bekleyen müşteri sayısı hesaplanır.

$$\boxed{\left(\text{Benzetim sonunda toplam zamana göre müşteri sayısı}\right) = \sum_i x_i (t_{i+1} - t_i)}$$

$t_i$ = i. müşteri geliş zamanı
$t_{i+1}$ = i+1. " " "

$$\boxed{\text{Kuyrukta bekleyen ortalama müşteri} = \frac{\sum_i x_i (t_{i+1} - t_i)}{\text{Toplam zaman}}}$$

Zamana göre ortalamanın alındığı bir başka performans ölçütü; servisin doluluk oranıdır.

$$\left.\begin{array}{l}\text{Servis doluysa} = 1 \\ \text{Servis boşsa} = 0\end{array}\right\} B(t)$$
↓
servisin durumu

$$\boxed{\text{Servisin doluluk oranı} = \frac{\sum_i B(t_i)(t_{i+1} - t_i)}{\text{Toplam zaman}}}$$

# — KESİKLİ OLAY BENZETİMİ — (8. HAFTA)

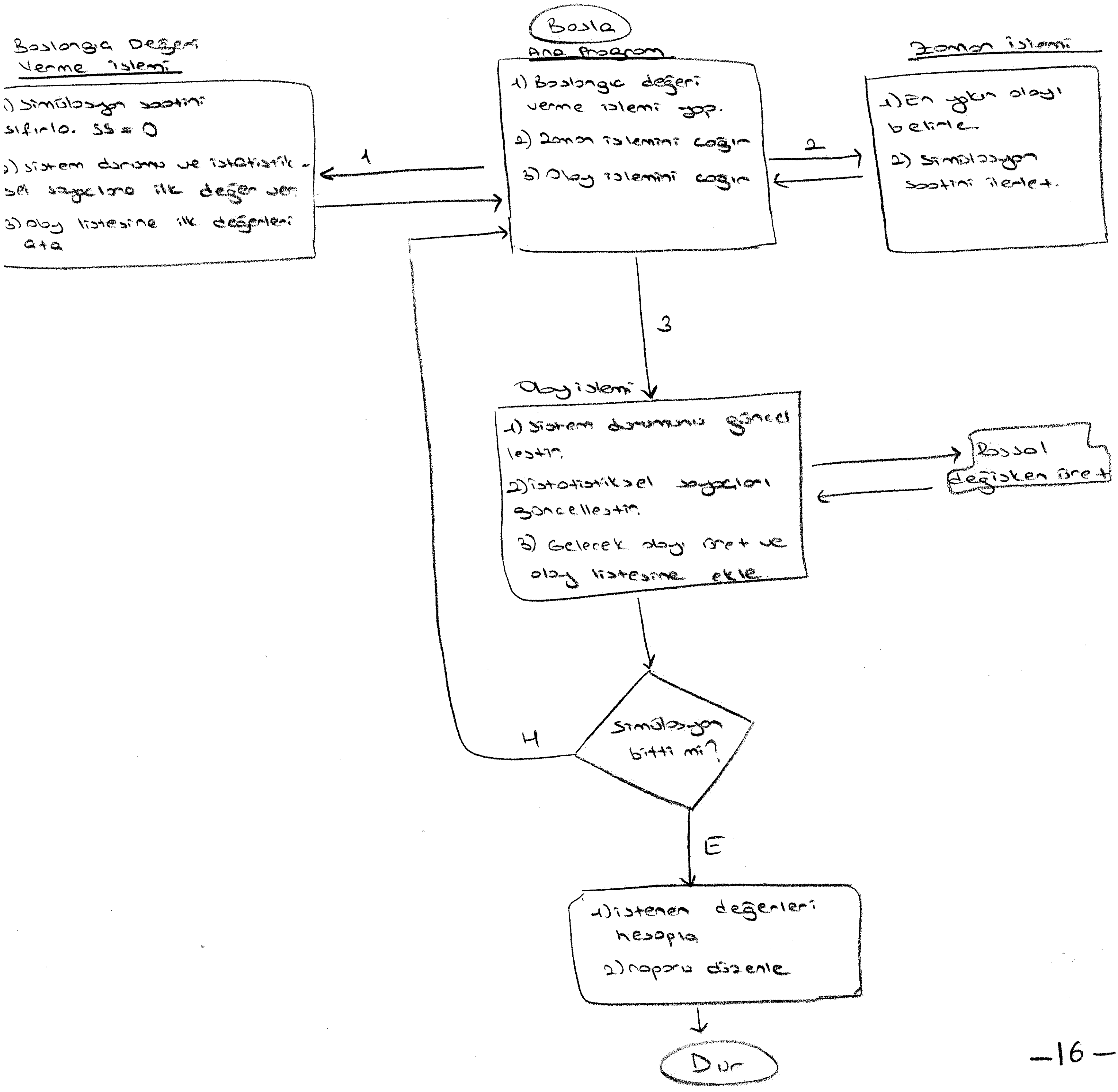
· En yakın olay zamanı
Zaman ilerletme } teknikleri kullanılır

Kesikli olay benzetimi modellerinde bulunan bileşenler;

- Sistem durumu
- Benzetim saati
- Olay listesi
- İstatiksel sayaçlar
- İlk değer verme işlemi
- Zaman işlemi
- Olay işlemi
- Rapor üreteci

· Ana programda yapılanlar;
Başlangıç değerler veriliyo
Zaman işlemi çağırıp en yakın olayı belirleme
Olay işlemlerini kontrol edip sistemin yeni durumunu güncelleme

· Kesikli olay benzetimi bileşenleri arasındaki mantıksal ilişki;



Toplam zaman

12/ El ile Benzetim Örneği;

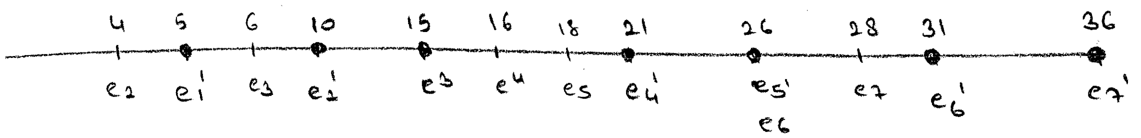
Aşağıda 7 iş ve herbirinin sisteme varışları verilmiştir.

| İş No | Varış Zamanı |
|---|---|
| 1 | 0 |
| 2 | 4 |
| 3 | 6 |
| 4 | 16 |
| 5 | 18 |
| 6 | 26 |
| 7 | 28 |

→ Bir makine ve bir iş için servis süresi 5 dk olsun diye varsayalım.

İşler tek bir kuyruk oluştursun.

(M|M|1 Model)




| İş | Varış zamanı | Kuyrukta bekleme | Servise başlama | Servis bitiş | Servis dolu zamanı |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 5 | 5 |
| 2 | 4 | 1 | 5 | 10 | 10 |
| 3 | 6 | 4 | 10 | 15 | 13 |
| 4 | 16 | 0 | 16 | 21 | 20 |
| 5 | 18 | 3 | 21 | 26 | 25 |
| 6 | 26 | 0 | 26 | 31 | 30 |
| 7 | 28 | 3 | 31 | 36 | 35 |

(Sistemde ortalama bekleme zamanı) = $\dfrac{\sum \text{kuyrukta bekleme} + \sum \text{servis zamanı}}{\text{Sisteme giren iş sayısı}} = \dfrac{11+35}{7}$

$= 6.571$ dk/müşteri

Sistemin dolu zaman yüzdesi = $\frac{\text{Servisin toplam dolu zamanı}}{\text{Sistemde geçen toplam zaman}}$ x100 ⇒ $\frac{35}{36}$ .100 = %97.22

→ (Servis bitiş zamanı)

→ KOB'un anlaşılmasının en iyi yolu; M/M/1 kuyruk modeli için ana ve alt programların akış şemalarının incelenmesidir.

— RASTGELE SÜREÇLER —

Dinamik sistemin simülasyonunu yaparken sadece sistemin kendisini modellemek önemli değil, daha önemlisi giriş sinyalini elde etmektir. Çünkü çoğunlukla bir sistemin giriş sinyali belli değildir.

Süreçler; sinyaller koleksiyonudur.

Bir sürekli zaman rastgele süreci $X(t)$ ile gösterilir. $X(t)$ = sinyaller dizisi

Ortalama – Otokorelasyon rastgele süreçler için önemlidir.

$$\boxed{\text{Ortalama} = \mu_{x(t)} = \frac{\sum x(t)}{\text{sinyal sayısı}}}$$

$$\boxed{\text{Moment} = \frac{\sum [x(t)]^2}{\text{sinyal sayısı}}}$$

$$\boxed{\text{Varyans} = \sigma^2 = \text{Moment} - (\text{Ortalama})^2}$$

$$\boxed{\text{Otokorelasyon} = \frac{\sum (\Pi\, x(t),\ x(t,r))}{\text{Toplam sinyal sayısı}}}$$

↓

sinyal ile kendisinin zaman sonraki çarpımının beklenen değeri

↓

Mesela; $2t+1$ sinyalinin zaman sonra beklenen değeri; $(2t + 2r + 1)$

→ Rastgele süreç; otokorelasyon ile sinyallerin dizisini oluşturma

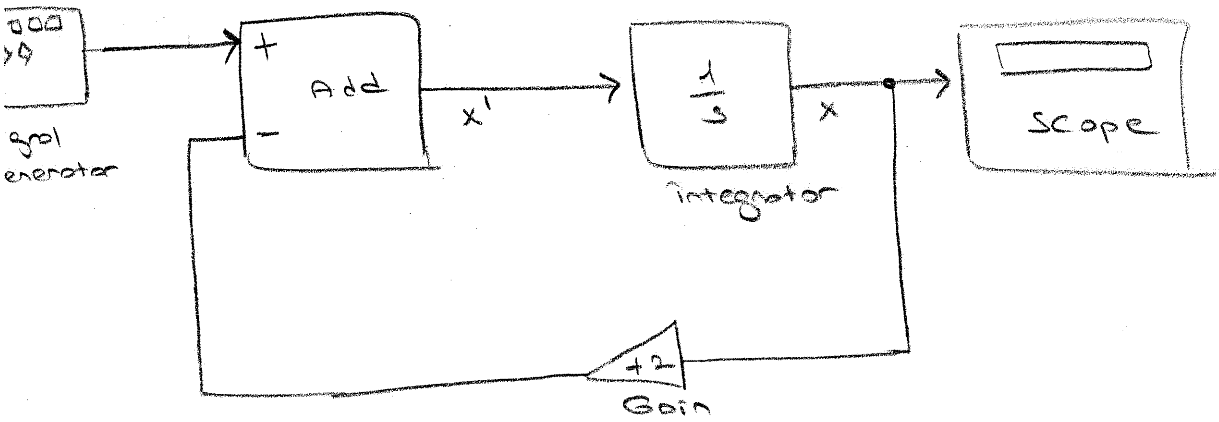
Rastgele sürece ait özellikler;
- Ortalama
- Standart sapma
- Eşitlik
- Varyans

→ 2 tür rastgele sinyal, dolayısıyla 2 tür rastgele süreç var.

Düzenli = Zaman senkrondur.
Herbir zaman unsurunda sinyal üretilir.
Eşit zaman aralıkları gözlenir.

Episodik = Olaylar düzenli oluşmaz.
Düzensiz asenkron zamanlarda oluşan olaylar
Olay oluştuğunda sinyal değerinde değişim olur.

SIMULINK ÖRNEK; $x(t) = -2x(t) + u(t)$ dif. denklemini modelle.




ÖRNEK/ Sinus ve cosinus dalgalarını gösterelim.