⊛ Benzetim = Gerçek hayattaki bir sistemin veya sürecin çalışmasının genellikle bilgisayar üzerinde taklit edilmesidir. Genel anlamda benzetim zaman içinde sistemin işleyişinin taklitidir.

→ Benzetimin gerçekleştirilebilmesi için karmaşıklık ve rassallık belirli düzeyde olmalıdır. Karmaşıklık arttıkça performans düzeyinden rassallık azalır.

→ Bir benzetim modeli; gerçek sistem üzerinde yapılacak değişikliklerin etkilerini ve yeni kurulacak bir sistemin performansını tahmin etmek için analiz ve tasarım amacı olarak kullanılır.

⊛ Model = Genelde matematiksel bir ifadedir. Bir sistemin karakterize edilmesidir. Bir sistemin gösterimi olarak tanımlanabilir.

⊛ Fiziksel Model = Bir matematiksel sistemin analog bir devre olarak elde edilmesidir.

⇒ Benzetim ve Modelleme için;

⊛ Amaçları = Değerlendirme, Karşılaştırma, Tahmin, Duyarlılık Analizi, Optimizasyon, Darboğaz Analizi..

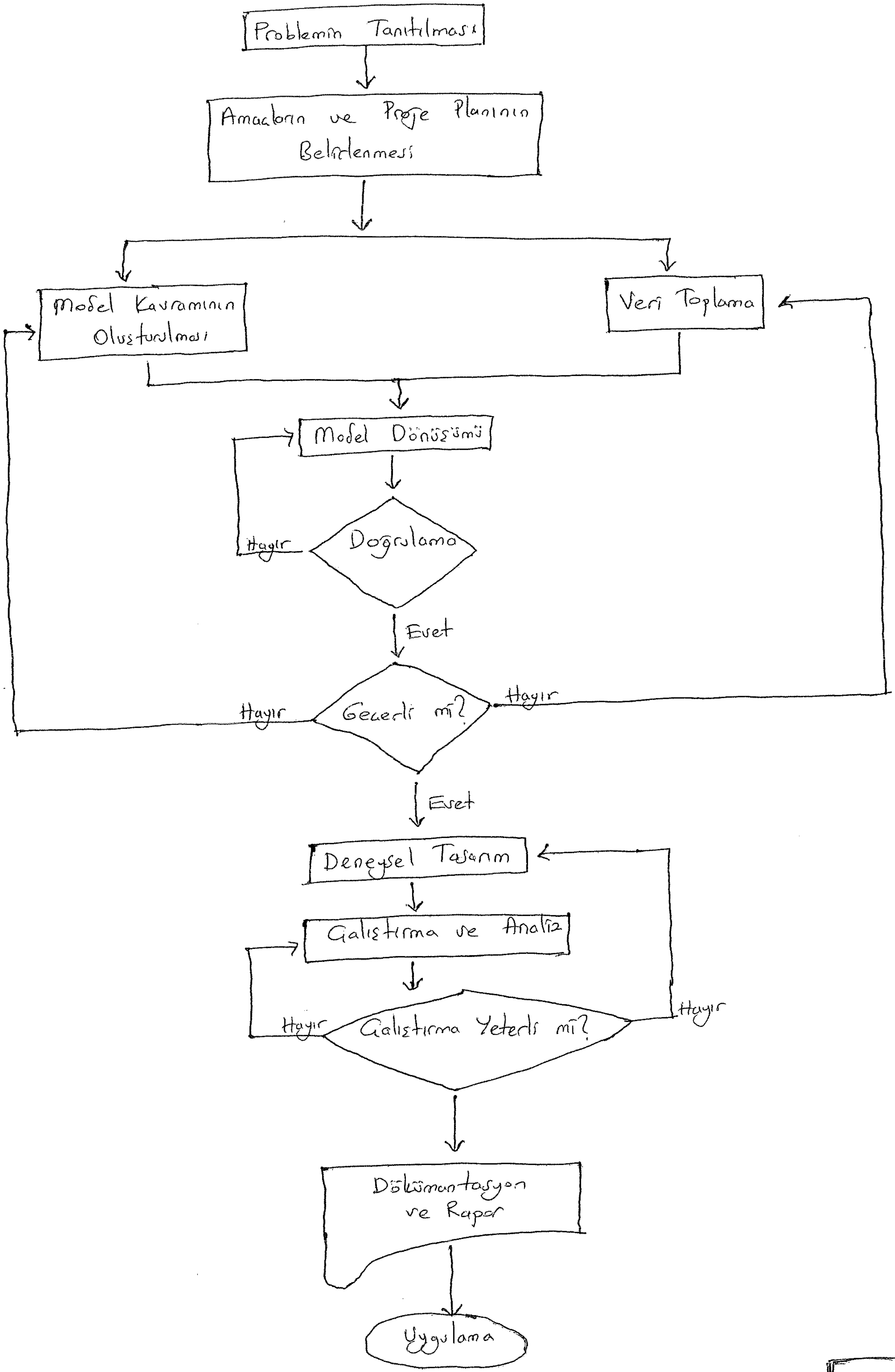
⊛ Ne zaman kullanılır? = ⊛ Üzerinde çalışılacak sistem deney yapmaya uygun değilse,
⊛ Sistem henüz tasarım aşamasında ise,
⊛ Problemin analitik çözümü mümkün değilse,
⊛ Sistemin davranış analizi yapılacaksa vb..

⊛ Avantajları = ⊛ Yeni bir sistem analizine yardımcı olmak için kullanılır.
⊛ Analitik modellerde çözüme ulaşmak için basitleştirici kabuller gereklidir, benzetimde bunlara ihtiyaç duyulmaz.
⊛ Analitik modellerde kısıtlı sayıda performans ölçütü sağlanabilir. Benzetim modelleri ile akla gelebilen herhangi bir performans ölçütü tahmin edilebilir.
⊛ Bazı durumlarda benzetim bir çözümün elde edilmesi için tek araçtır.

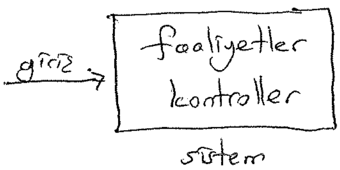
⊛ Dezavantajları = ⊛ Benzetim modellerinin kurulması ve geçerliliğinin araştırılması zaman alıcıdır.
⊛ Bu nedenle bilgisayarlarda benzetim modellerinin koşum maliyeti yüksek olabilir.
⊛ Benzetim modellerinin birden fazla n kez çalıştırılması maliyeti etkiler.
⊛ Analitik yöntemlerin yeterli olduğu durumlarda kullanmak faydalı değildir.
⊛ Benzetim modelleri girdi/çıktı sistemleri olarak bilinir. Benzetim modelleri sistemleri matematiksel sistemler gibi çözülmezler, araştırılırlar.
⊛ Belirli koşullar altında sistemin davranışı incelendiğinden analitik çözümler gibi en iyi sonucu üretmezler.

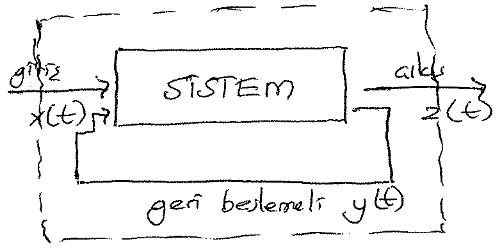
⊛ Uygulama Alanları = ⊛ Bilgisayar Sistemleri ⇒ Donanım, Yazılım, Bilgisayar Ağları, VTY, OPNET..
⊛ Üretim Sistemleri ⇒ Malzeme Taşıma Sistemleri, Montaj Hatları...
⊛ İşletme ⇒ Ücretlendirme Politikası...
⊛ Kamu Hizmeti ⇒ Askeri silahlar ve kullanımları, Nüfus Tahmini, Sağlık Hizmetleri
⊛ Ekoloji ve Çevre ⇒ Atık kontrolü, Hava kirliliği, Güneş enerjisi sistemleri
⊛ Sosyoloji ⇒ Eğitim alanında, Sosyal sistem analizi
⊛ Biyoloji ⇒ Biyomedikal..

Benzetim Aşamaları =



⊛ Sistem = Bir amacı gerçekleştirmek için aralarında düzenli bir etkileşim ve bağımlılığın bulunduğu nesneler topluluğudur. Geri besleme sistemi dinamikleştirir.





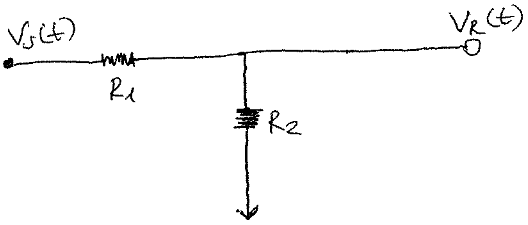
→ $x(t)$ sistemin giriş vektörü, $z(t)$ sistemin çıkışı...

→ Eğer sistem sadece giriş vektörüne bağlıysa 0. dereceden bir sistemdir.

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

<u>NOT</u> ⇒ Eğer sistemdeki giriş sinyali dinamik olarak çıkışa bağlı ise sistem bir hafızaya sahiptir denir.

Ör



→ Tek girişli elektrik devresi sistemi (dinamik değil)

→ $x(t)$ [giriş] ⇒ $V_s(t)$ ve $z(t)$ [çıkış] ⇒ $V_R(t)$

→ Çıkış ve durum denklemlerini tanımlamak için durum değişkenleri tanımlamalıdır. Durum değişkeni akım ise;

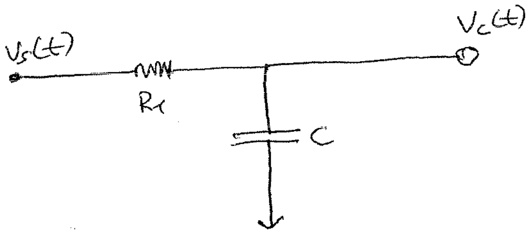
Çözüm

$$I = \frac{V_s(t)}{R_{eş}} = \frac{x(t)}{\frac{R_1 \cdot R_2}{R_1+R_2}}$$

$$V_R(t) = R_2 \cdot I$$

$$V_R(t) = \cancel{R_2} \cdot \frac{x(t) \cdot (R_1+R_2)}{R_1 \cdot \cancel{R_2}}$$ bulunur.

Ör



→ Kondansatör enerji depolayan bir eleman olduğu için sistem dinamiktir.

Çözüm

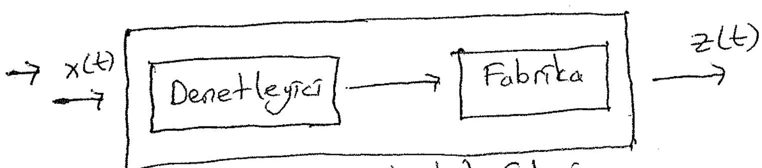
$$x(t) = V_s(t)$$
$$z(t) = V_c(t)$$

$$\Rightarrow \quad V_s = V_c + RC \cdot \frac{dV_c}{dt}$$

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

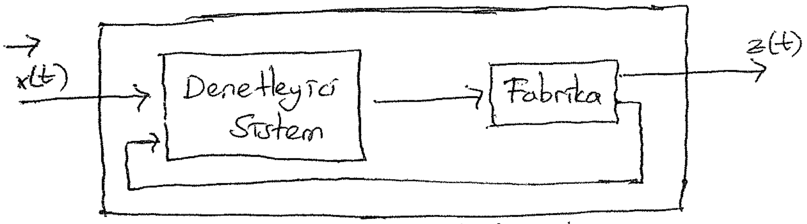
$$y'(t) = \frac{1}{RC} \cdot (x(t) - y(t))$$

⊛ Kontrol Sistemleri =



Açık Çevrim Kontrol Sistemi

⇒ Gelen giriş sinyallerine göre istenilen çıkışları elde eden sistemdir.

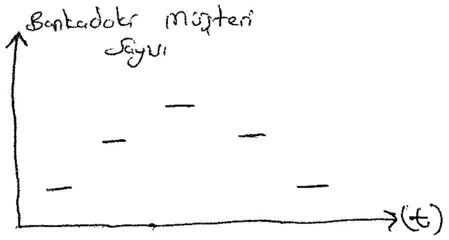


Kapalı Çevrim Kontrol Sistemi

⇒ Eğer çevresel etkiler sistem üzerinde etkiliyse, geri beslemeli sistem gerekir. Bu durumda kapalı çevrim kontrol sistemi kullanılır.

→ Sistemler yapısına göre ikiye ayrılır;

⊛ Ayrık Zamanlı Sistemler = Durum değişkenlerinin sadece ayrık zamanda değiştiği sistemdir.



→ Örneğin; banka müşteri gelince aktif olur.

⊛ Sürekli Zamanlı Sistemler = Zaman üzerinde durum değişkenlerinin sürekli değiştiği sistemdir.

→ Bu tür sistemde $t_k = t_0 + h_k$ for $k = 0, 1, 2 \ldots$ ile ifade edilir. t_k; t_0 başlangıç zamanı ile başlar ve sistem sinyali h birim zamanına kadar değişmez.

→ $t_{k+1} - t_k = h$ ⇒ Örnekleme oranının adım boyutu

Ör: 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;

$$x(t) = \cos(\pi t) \qquad t_k = 3 + \frac{1}{2}k$$

$$x(t_k) = \cos(\pi(3 + 1/2\,k)) = -\cos(1/2\,\pi k)$$

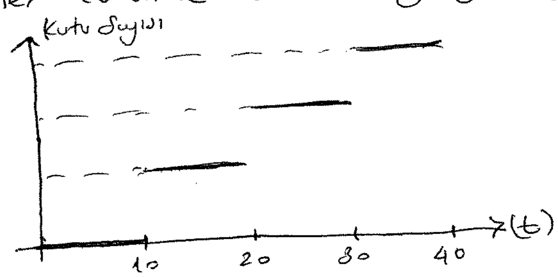
$$x(k) = \begin{cases} 0, & k = \text{tek} \\ (-1)^{\frac{k+2}{2}}, & k = \text{çift} \end{cases}$$

Ör: Her 10 sn'de bir kutunun ulaştığı bir fabrika taşıyıcı sistemi düşünelim. Kutuların ağırlıkları sırasıyla 5, 10 ve 15 kg'dır. Fakat 5 ve 15 kg'lık kutuların gelme olasılığı 10 kg'ın gelme olasılığının iki katı fazladır. Bu sistem nasıl modellenir? Benzetimi nasıl yapılır?

Çözüm

w	olasılık	
5	0,4	→ 0,4
10	0,2	→ 0,4 − 0,6
15	0,4	→ 0,6 − 1
	+ 1	

⊛ Her 10 sn'de bir kutu geldiğine göre;



⊛ Sözde Kodu = for k=1:n

```
r = 10 * rnd
if r<4 then w(k)=5          (0,1,2,3)
if 4≤r<6     w(k)=10        (4,5)
if r≥6 then  w(k)=15        (6,7,8,9)
next k
```

Olay: Sistemin durumunu değiştiren ve anlık olarak ortaya çıkan durumlardır.

Örneğin; bankaya müşteri gelmesi, ayrılması

- DİNAMİK SİSTEMLER -

Dinamik model zamana bağlı modeldir. zamanla ilgili değişken tutar.

ÖRNEK: Ayrık zamanlı sistem örnekleri;

- Banka ⟶ sistem

  müşteri ⟶ varlık

  hesap kontrolü ⟶ özellik

  mevduat açma ⟶ faaliyet

  varış, çıkış ⟶ olay

  mesgul vezne sayısı, müşteri sayısı, varış zamanı } Sistem durum değişkeni

- Hızlı tren sistemi için;

  yolcular ⟶ varlık

  trene binip inilen yer ⟶ özellik

  seyahat ⟶ faaliyet

  trenin istasyona varışı ⟶ olay

  gidilecek şehre varış ⟶ olay

  Her istasyonda bekleyen yolcu sayısı ⟶ durum değişkeni

  toplam yolcu sayısı ⟶ Durum değişkeni

- Haberleşme örneği;

  mesaj ⟶ varlık

  mesajın uzunluğu ⟶ özellik

  mesaj gönderme ⟶ faaliyet

  iletildi raporu ⟶ olay

  karakter uzunluğu ⟶ özellik

- başlangıç değer problemi

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

- Birinci dereceden bir başlangıç değer problemi için;

$$\underbrace{\frac{dx}{dt} = f(t,x)}_{\text{dif. denklem}} \qquad \underbrace{x(t_0) = x_0}_{\text{başlangıç koşulu}} \quad \text{ise}$$

$x(t) = [x_1(t), x_2(t), \dots x_n(t)]$ → sistem durum vektörü

$x(0) = [x_1(0), x_2(0), \dots x_n(0)]$ → ilgili başlangıç durumları

Örnek: $\alpha'' + 2\beta\alpha' + \beta^2\alpha = \cos t$ Başlangıç durumları

$\beta' + \alpha\beta = 4$ $\alpha(0) = 2$, $\alpha'(0) = -1$, $\beta(0) = 1$

- Eğer elimizde yüksek mertebeden bir diferansiyel denklem mevcut ise bu dif. denklemi daha düşük derecede bir denkleme dönüştürüp sayısal çözümü o şekilde yapmamız gerekir.
- Örnekte iki dinamik durum değişkeni ve birinci ve ikinci dereceden diferansiyel denklemler olduğu için; üçüncü dereceden bir sistemdir. Bu yüzden 3 tane 1. dereceden diferansiyel denklem sistemi olarak tekrar tanımlamak mümkündür.

$x_1(t) = \alpha(t)$, $x_2(t) = \alpha'(t)$, $x_3(t) = \beta(t)$ ⇒ tanımlanan bu denklemleri denklem sisteminde yerine yazalım,

$$x_2' + 2x_2 . x_3 + x_3^2 . x_1 = \cos t$$

$$x_3' + x_1 . x_3 = 4$$

- x_1, x_2 ve x_3'ün türevlerini bulalım.

$x_1' = x_2$ $x_1(0) = 2$

$x_2' = -2x_2x_3 - x_3^2 x_1 + \cos t$ $x_2(0) = -1$

$x_3' = -x_1 . x_3 + 4$ $x_3(0) = 1$

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

$f = [x_2, \cos t - 2x_2x_3 - x_3^2 x_1, -x_1 . x_3 + 4]$

$x_0 = [2, -1, 1]$

$t_0 = 0$

- f fonksiyonunda x_1, x_2 ve x_3 değerlerinin türevlerini gösteriyoruz

-Euler yöntemi

Euler yöntemi türev tanımını kullanır. $\lim_{x \to 0} \frac{x(t+h) - x(t)}{h} = f(x,t)$

$f(x,t) = \frac{dx}{dt}$

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

ÖRNEK: $x' = x^2 t$, $x(1) = 3$ denkleminin gerçek ve Euler çözümünü bulalım.

- Önce gerçek çözümü yapalım;

$$\frac{dx}{dt} = x^2 t \Rightarrow \frac{dx}{x^2} = t.dt \qquad x(\underset{t_0}{1}) = \underset{x_0}{3}$$

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

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

- Keyfi verilmiş $h = 0.05$ adım uzunluğu ile Euler çözümünü yapalım;

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

ayrık zamanlı Euler formülümüz $\Rightarrow x(k+1) = x(k) + h.f[t(k), x(k)]$

$t_{k+1} = t_k + 1/20$

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

Euler sözde kodu

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

Euler matlab kodu

```
to=1 ; xo=3;
x=[xo];
t=[to];
xg=[6/(5-3*to^2)]      % gerçek çözüm (mutlak çözüm)
h=0,05                 % adım uzunluğu
for i=1:5
 xyeni=xo+h*xo^2*to;
 to=to+h;
 xg=[xg 6/(5-3*to^2)];
 x=[x xyeni];
 t=[t to];
 xo=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 hesaplamak için daha fazla hesaplama gerekecektir.
  2) Veri gösteriminde bir sonraki makine sınırlamalarından dolayı h çok fazla küçük olabilir.

- Taylor yöntemi

Euler yönteminin özel bir durumu olarak düşünülebilir.

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

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

ayrıklaştırırsak;

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

ÖRNEK: $x' = x^2t$ ve $x(1) = 3$ sistemine Taylor tekniğini uygulayalım;

$$f(t,x) = x' = x^2t \quad , \quad \frac{\partial f}{\partial t} = x^2 \quad , \quad \frac{\partial f}{\partial x} = 2xt$$

$$\begin{aligned} x(t+h) &= x(t) + h.f(t,x) + \frac{1}{2}h^2\left[\frac{\partial f}{\partial t} + \frac{\partial f}{\partial x} f(t,x)\right] \\ &= x(t) + h.x^2(t).t + \frac{1}{2}h^2\left[x^2(t) + 2x(t).t.x^2(t).t\right] \\ &= x(t) + h.x^2(t).t + \frac{1}{2}h^2\left[x^2(t) + 2x^3(t).t^2\right] \end{aligned}$$

- ayrıklaştırırsak;

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

NOT: Taylor yöntemi Euler yöntemine göre daha doğru sonuçlar elde eder.

- Runge Kutta

$$x(t+h) = x(t) + hx'(t) + \frac{1}{2}h^2x''(t) + \frac{1}{6}.h^3x^{(3)}(t) + \frac{1}{24}.h^4x^{(4)}(t)$$

$K_1 = f(t_k, x(k))$

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

$K_2 = f(t_k + \frac{1}{2}h, x(k) + \frac{1}{2}hK_1)$

- $x(k+1) = x(k) + \frac{1}{6}h(K_1 + 2K_2 + 2K_3 + K_4)$

$K_3 = f(t_k + \frac{1}{2}h, x(k) + \frac{1}{2}hK_2)$

$K_4 = f(t_k + h, x(k) + hK_3)$

ÖRNEK: $x' = x^2 t$ ve $x(1) = 3$ sistemine Runge Kutta tekniğini uygulayınız.

$t_0 = 1$, $x_0 = 3$

- Sözde kod

```
input t,x
print t,x
for k=1 to n
k1 = t x^2
k2 = (t + 1/2 h) (x + 1/2 h k1)^2
k3 = (t + 1/2 h) (x + 1/2 h k2)^2
k4 = (t + h) (x + h k3)^2
x = x + 1/6 h (k1 + 2k2 + 2k3 + k4)
t = t + h
print t,x
next k
```

- Matlab kodu

```
to=1;
xo=3;
x=[xo];
t=[to];
xg=[6/(5-3*to^2)]
h=0.05;
for k=1:5;
  K1=to*xo^2;
  K2=(to+1/2h)*[xo+(1/2)*h*K1]^2;
  K3=(to+1/2*h)*[xo+(1/2)*h*K2]^2;
  K4=(to+h)*[xo+h*K3]^2;
  xyeni=xo+(1/6)*h*(K1+2K2+2K3+K4);
  to=to+h
  xg=[xg 6/(5-3*to^2)];
  x=[x xyeni];
  t=[t to];
  xo=xyeni;
  end
  plot (t,x,'r-d')
  hold on
  plot (t,xg,'k-s')
```

- Yüksek mertebeden sistemler

İki tane birinci dereceden denklem sistemi şöyle tanımlanabilir;

$x' = f(t,x,y)$ (k_1,k_2,k_3,k_4)

$y' = g(t,x,y)$ $(\lambda_1,\lambda_2,\lambda_3,\lambda_4)$

$x(t_0) = x_0$

$y(t_0) = y_0$

Ayrık çözüm denklemi şöyle elde edilir;

$K_1 = f(tk, x(k), y(k))$

$\lambda_1 = g(tk, x(k), y(k))$,

$K_2 = f(tk + 1/2 h, x(k) + \frac{1}{2} h K_1, y(k) + \frac{1}{2} h \lambda_1)$;

$\lambda_2 = g(tk + 1/2, x(k) + 1/2 h K_1, y(k) + \frac{1}{2} h \lambda_1)$;

$K_3 = f(tk + 1/2 h, x(k) + 1/2 h K_2, y(k) + \frac{1}{2} h \lambda_2)$,

$\lambda_3 = g(tk + \frac{1}{2} h, x(k) + \frac{1}{2} K_2, y(k) + \frac{1}{2} h \lambda_2)$,

$K_4 = f(tk + h, x(k) + h K_3, y(k) + h \lambda_3)$,

$\lambda_4 = g(tk + h, x(k) + h K_3, y(k) + h \lambda_3)$,

$x(k+1) = x(k) + 1/6 h (K_1 + 2K_2 + 2K_3 + K_4)$

$y(k+1) = y(k) + 1/6 h (\lambda_1 + 2\lambda_2 + 2\lambda_3 + \lambda_4)$

ÖRNEK: $x'' + 3xx' = u(t)$ $u(t)=t$, $t \geq 0$ $x(0)=2$ $x'(0)=1$

- İkinci dereceden bir diferansiyel denklem olduğu için iki durum değişkeni olmalıdır. Bunlardan birini $x(t)$ çıkışı olarak, diğerini $y(t)=x'(t)$ olarak tanımlarız. Böylece;

$x'(t)=y$, $y'=t-3xy$, $x(0)=2$, $y(0)=1$ 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;

```
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
```

- Otonom dinamik sistemler

Herhangi bir girişi olmadan kendi kendine çalışan sistemlerdir.

1) Kararlı: Kısa bir geçiş evresinden sonra çıkış 0'a yaklaşır.

2) Kararsız: Sınırlama olmaksızın doğal tepki artar.

3) Marjinal: Tepki periyodik ve sınırlıdır.

- Populasyon

$x(t) \Rightarrow t$ zamanındaki populasyon büyüklüğü

$\lambda \Rightarrow$ sabit orantı değeri

$x' = \lambda x$ $t=0$ zamanındaki başlangıç büyüklüğünü x_0 olarak kabul edersek $t \geq 0$ için;

- $x(t) = x_0 . e^{\lambda t}$ 'dir. Populasyon sürekli artar, Mathasion modelidir.

- $X_m \Rightarrow$ populasyonun maksimum taşıma kapasitesi

$\frac{X}{X_m} \Rightarrow$ sistemin doluluk oranı

- $x' = \lambda . x \left(1 - \frac{x}{x_m}\right) \Rightarrow$ lojistik denklem

- $x(t) = \frac{x_m}{1-(1-(x_m/x_0)).e^{-\lambda t}} \Rightarrow$ analitik çözüm

- Yiyerek ve yenilmeyerek hayatta kalan av ve bir avcının var olduğu bir ortam tipik durumlardan biridir.

$x(t) \Rightarrow$ av populasyonunun büyüklüğü , $y(t) \Rightarrow$ avcı populasyonunun büyüklüğü

av-avcı etkileşim sayısı $\Rightarrow x(t).y(t)$

- av-avcı populasyonunun büyüme oranı;

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

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

Yukarıdaki denklemlere Lotka-Voltera denklemleri denir.

- Lotka-Voltera modelinin benzetimi için; Euler yöntemi uygulanabilir.

```
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
```

NOT: Lotka-Voltera sisteminde, sonuçların $[t_0, t_n]$ zaman aralığında ve h aralığında 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.

ÖRNEK: $x' = 3x\,(1 - 1/10\,y)$, $x(0) = 10$

$y' = 1{,}2\,y\,(-1 + 1/25\,x)$, $y(0) = 5$ verilen Lotka-Voltera sistemini $[0,5]$ aralığında çözelim.

- $h = 0{,}001$ alalım. Grafik 5 birim zaman kadar olacak ve her bir aralıkta 10 örnek yeterlidir. Dolayısıyla 50 örnek iyidir;

$n = (5-0)/(50 * 0{,}001) = 100$ güncelleme olur.

```
function [t,x,y]
  x=[x0]
  y=[y0]
  t=0;
  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
      plot (x)
      hold on
      plot (y,'r-')
```

Benzetim modelleri 3 ana grupta toplanır;

- Statik veya dinamik
- Belirli veya Olasılıklı
- Kesikli veya Sürekli

Statik benzetim modeli: Sistemin belirli bir anındaki gösterimidir. Monte-Carlo modelleri bu türe uygun modellerdir.

Dinamik benzetim modeli: Sistemin bir aralığı veya tüm çalışma zamanı dikkate alınarak yapılan modellemedir. Örneğin; bir banka için kurulan bir benzetim modeli 8 saatlik bir çalışma zamanı dikkate alınarak çalıştırılır.

Belirli Benzetim modeli: Rassal değişken içermeyen benzetim modelidir. Bu modellerde verilen girdi seti için bir çıktı seti vardır.

Olasılıklı benzetim modeli: Bir veya birden fazla rassal değişken içeren benzetim modelidir. Banka örneğinde, varışlar arası zaman aralığı ve servis zamanları rassal değişkenlerdir.

ÖRNEK:

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

A ve B gibi iki makinenin bulunduğu bir atölyeyi ele alalım. Her bir iş önce makine A'da yığın olarak bitirildikten sonra makine B'de yığın olarak başlar ve tamamlanır. Son yığın ne zaman tamamlanır? (Makine A: Yığın büyüklüğü / 50 + 1) gün, Makine B: (Yığın büyüklüğü / 100 + 3) gün

		Makine A		Makine B	
İş. No	Geliş zamanı	Başlama	Bitiş	Başlama	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

- 200 için;

Makine A: $200/50+1 = 5$ gün

Makine B: $200/100+3 = 5$ gün

- **Stokastik & deterministik:** Bir sistem, eğer davranışı tümüyle tahmin edilebilir ise deterministikdir. Eğer bir sistemin davranışı bütünüyle tahmin edilemiyorsa stokastikdir.

ÖRNEK: (Deterministik benzetim): Büyük AL banka soymak için çetesini oluşturmaya karar verir. 6 ay içinde onun için çalışacak 50 çete üyesine sahip olması gerektiğini anlar.

- Ekibe haftalık olarak, ideal çete büyüklüğü (50) ve çetedeki adam sayısı arasındaki farkın dörtte birine eşit oranda adam katılacaktır.
- Aynasızlar her hafta Büyük AL'in adamlarının %5'ini yakalar ve her birini en az 12 ay cezaya çarptırır.
- Hapistekilerin %10'u her hafta firar eder ve Büyük AL'in çetesine katılır.

Bu şartlar altında 10 hafta sonra Büyük AL'in çetesinin büyüklüğü ne kadar olacaktır?

- 10 haftalık süre için benzetim sonuçları aşağıdaki tablodadır.
- Büyük AL on hafta içinde 50 kişilik çete hedefine ulaşamayacaktır.

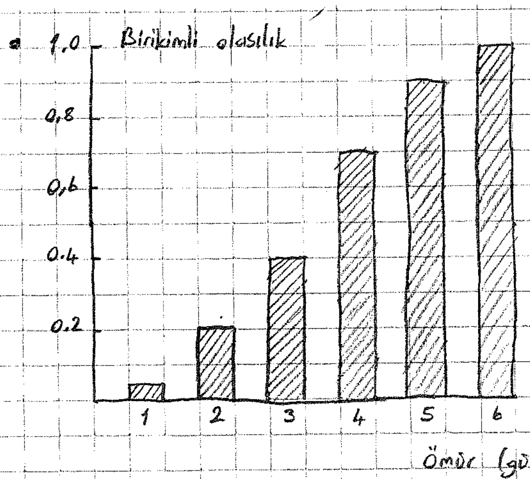
Hafta	Toplama Oranı	Yakalanma oranı	Kaçış oranı	Kadesteki sayı	Çete büyüklüğü
1	12,5 ((50-0)/4)	0.00	0.00	0.00	12.50
2	9,38 ((50-12,5)/4)	0,63 (T.0,05)	0,00	0.63	21,25
3	7,19 ((50-21,25)/4)	1,06 (21.25x0,05)	0.06	1.63	27.64
4	5,64	1,37 (27.64x0.05)	0.16	2.83	31,87
5	4,53	1,59	0.28	4.14	35.09
6	3,73	1,75	0.41	5.48	37.48
7	3.13	1.87	0.55	6.81	39.28
8	2,68	1.96	0.68	8.09	40.68
9	2.33	2.03	0.81	9.32	41.78
10	2.05	2.09	0.93	10.48	42.68

ÖRNEK (stokastik benzetim):

tamirden sonraki gün	Arıza olasılığı
1	0.05
2	0.15
3	0.20
4	0.30
5	0.20
6	0.10
>6	0.00

tablo en son tamir gördükten sonra izleyen günlerde bir disk biriminin tekrar arızalanma olasılığını göstermektedir.

- Yani ünitelerin %5 i tamirden 1 gün sonra arızalanır %15 i 2 gün sonra vs.



- Kümülatif olasılık

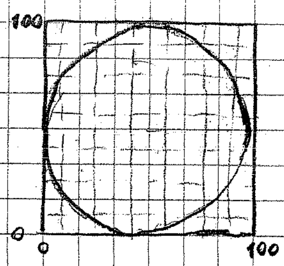
1. gün 0,05

2. gün 0,05+0.15

3. gün 0,05+0.15+0.20

- Monte Carlo benzetim modeli: Statik bir benzetim modelidir. Olasılık teorisi üstüne kurulu bir sistemdir. Diğer analitik yaklaşımların mümkün olmadığı fonksiyonların integralinin sayısal olarak elde edilmesinin bir yoludur. Monte Carlo metoduyla istatiktiksel ve matematiksel tekniklerle bir deneyi veya çözülmesi gereken fiziksel bir olayı tesadüfi sayılarla defalarca kullanarak benzetimi yapılıp çözmek esastır. İlk defa II. Dünya savaşı sırasında atom bombasının geliştirilmesiyle ilgili problemlere uygulanmıştır.

ÖRNEK:



• Şekildeki küçük karelerin sayılması π değerinin hesaplanmasına olanak verir.

$\pi = \frac{4m}{n}$ $m \rightarrow$ çemberin içindeki kare sayısı, $n \rightarrow$ toplam küçük kare sayısı

- Subjektif olasılıklar

• Önsel sav: Olabilecek bütün sonuçlar hakkında tam bilgiye sahip olma durumu (zar problemi) $\frac{1}{6}$

• Göreli sıklık savı: Çıktıları üreten süreci anlamadığımızda fakat onların göreli sıklıklarını hesaplamak için yeterli veriye sahip olduğumuz durum.

• Öznel bakış: Önsel veya göreli sıklık yaklaşımının ikisinin birden uzantısı olarak bakılan durum.

Yazı gelme olasılığı gibi tura gelme olasılığının da 0.5 olması.

Monte carlo benzetimi ortalama metodu:

ÖRNEK: $\int_a^b g(x)dx$ integralini çözelim;

• G(x) fonksiyonu, analitik çözümü olmayan bir fonksiyon olsun

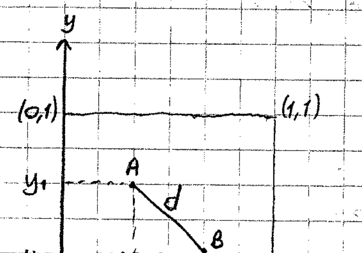
• Monte carlo ile çözelim;

$$E(y) = y_{ort} = \frac{\sum_{i=1}^{n} y_i}{n} = (b-a) \cdot \frac{\sum_{i=1}^{n} g(X_i)}{n}$$

Burada $X_1, X_2 \ldots X_n$ rassal değişkenlerdir.

ÖRNEK: Kenarları birim uzunlukta bir kare düşünelim. 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?

Açıklama; Monte Carlo tekniğiyle rassal olarak 1000 adet A ve B noktaları üreterek d'nin 0.8'den küçük olma olasılığını ve kullanacağınız yaklaşımı açıklayarak akış şemasını çiziniz.



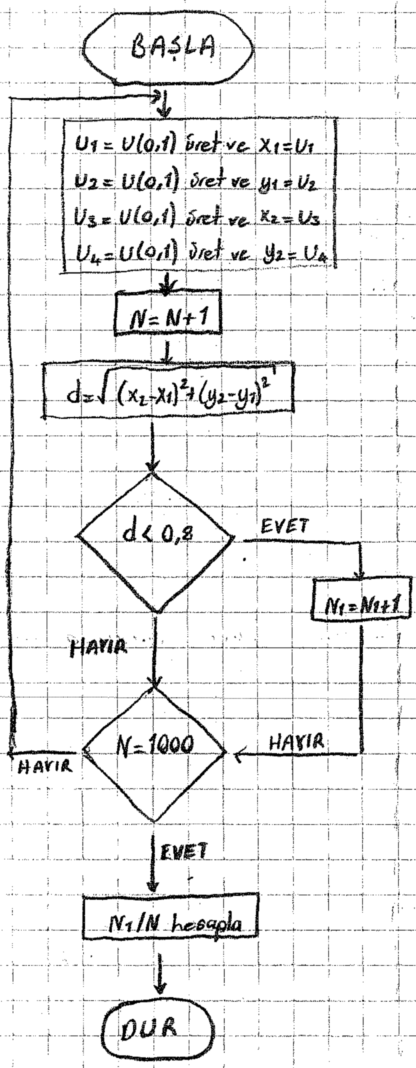
$A = (x_1, y_1)$

$B = (x_2, y_2)$

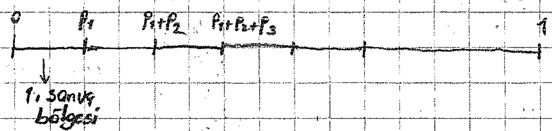
$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

N_1 = d'nin 0.8 den küçük olduğu sayaç

N = 1000 deneme



ÖRNEK: Gelişigüzel sayı eksenine n-tane sonuç bölgesinin yerleştirilmesi



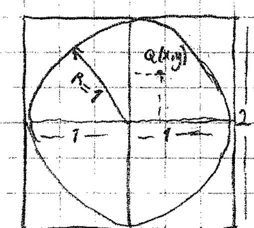
- Gelişigüzel sayıların P_1 olasılıkla belirlenen miktarını 1. sonuç P_2 olasılıkla belirlenen miktarını 2. sonuç, P_n olasılıkla belirlenen miktarını da n. sonuç için ayırmış olduk. Böylece belirtilen bir gelişigüzel sayı hangi sonuç bölgesine düşerse, olayda o sonuç meydana gelmiştir. Dolayısıyla olasılık dağılımı şöyle olur;

$0 < Q < P_1$ ise 1. sonuç

$P_1 <= Q < P_1+P_2$ ise 2. sonuç

$P_1+P_2+ \ldots P_{n-1} <= Q < 1$ ise n. sonuç

ÖRNEK: Karenin bir kenarı 2 birim ve çemberin yarıçapı $R=1$ birim olsun. Koordinatları (x,y) olan herhangi bir Q noktası seçilsin. Q noktası $x^2+y^2 \leq 1$ şeklinde seçilmişse, Q noktası çemberin içinde, aksi halde nokta çemberin dışındadır. Çemberin içinde kalma ihtimali;



Çember alanı $= \pi r^2$, karenin alanı $= 2r \times 2r = 4r^2$

$$P(x^2+y^2<1) = \frac{\text{çemberin alanı}}{\text{karenin alanı}} = \frac{\pi}{4}$$

$$P(x^2+y^2<1) = \frac{\text{çemberin içinde kalan noktalar}}{\text{karenin içinde kalan noktalar}} = \frac{m}{n}$$

iki denklemi birleştirirsek;

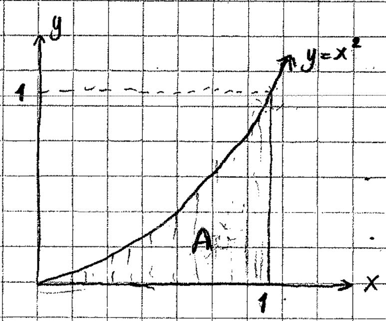
$\pi \approx \frac{4m}{n}$ şeklinde olur.

matlab kodu;

```
function pi=montecarlo1(n)

% pi : pi sayısı
% n: nokta sayısı
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;
```

$\int_0^1 x^2 dx \quad \frac{\frac{1}{3}}{1} = \frac{1}{3}$

ÖRNEK:



Şekilde görülen $y=x^2$ ile x ekseni arasında kalan taralı alanı bulmak için Monte Carlo benzetimi kullanılabilir.

- Eğer Eğrinin altındaki noktaları toplam nokta sayısına oranlarsak, A alanının R karesine olan oranını yaklaşık olarak elde edebiliriz.

```
function   egrialan=montecarlo2(n)
 % egrialan:
 % n : nokta sayısı
  egri=0;
  sayac=0;
  for i=1:n
       x=rand;
       y=rand;
       if (y<=x^2)
            egri=egri+1;
       end
       sayac=sayac+1;
  end
  egrialan=egri/sayac;
```

ÖRNEK: 0 ile 100 arasında bulunan sayılar içinden rastgele seçilen bir sayının 11'e tam bölünebilme olasılığı nedir?

• analitik çözüm; bu sayılar; 11, 22, 33, 44, 55 ... 99 toplam 9 adet

tüm sayılarda 100 adet, olasılık değerimiz $P = 9/100 = 0{,}09$

• Monte Carlo benzetimi 0 ile 100 arasında rastgele n adet sayı seçmemizi ister.

Seçeceğimiz n adet sayıdan m tanesinin 11'e tam bölündüğünü farz edelim.

• Bu durumda olasılık değerimiz m/n olur.

• Programlama esnasında rastgele seçilecek her sayı için n artarken, m değeri seçilen sayının 11'e tam bölünmesi durumda arttırılmalıdır.

```
%
function sonuc = montecarlo3(n)
% bol : 11'e bölünebilen sayısı
% n : nokta sayısı
bol=0;
sayac=0;
for i=1:n
    x = round ((rand*99)+1);
    sonuc= mod (x,11);
    if (sonuc==0)
        bol=bol+1;
    end
    sayac=sayac+1;
end
sonuc= bol/sayac;
```

NOT: 100 örnekli montecarlo'da sonuc = 0,15

1.000.000 örneklide 0,0905 olur.

ÖRNEK:

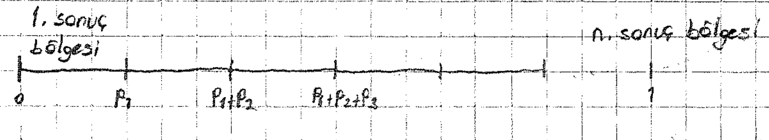
Gönderilen miktar	Frekans	Olasılık
100	3	0,083
200	8	0,222
300	10	0,277
400	11	0,305
500	4	0,111
	36	1

$\frac{3}{36}$

Daha önceki 3 yılı gönderilen para miktarını gösteren grafiklerden Frekans ve olasılıkları bulduk

3 yılda 4 kez 500 lira gönderilmiş

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



• Böylece belirtilen bir gelişigüzel sayı hangi sonuç bölgesine düşerse, olayda o sonuç meydana gelmiştir. Bu durumda olasılık dağılımı aşağıdaki gibi olur.

- $0 < Q < P_1$ ise 1. sonuç

  $P_1 \leq Q < P_1 + P_2$ ise 2. sonuç

  $P_1 + P_2 + \dots P_{n-1} \leq Q < 1$ ise n. sonuç $\Rightarrow$ Buradaki mantığı örneğe uygularsak (kümülatif)

Kümülatif olasılık	Miktar
0	100
0,0833	200
0,305	300
0,583	400
0,888	500

0 – 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

– Benzetim dili kullanmanın faydaları; Benzetim dilleri kullanılarak programlama zamanı azaltılır. Modelin programlanmasında gerekli özelliklerin birçoğu benzetim dilinde mevcuttur. Benzetim modelleri benzetim dili ile kodlandığında değiştirilmesi kolaydır. Benzetim dili kullanıldığında, programlama hatasını bulmak daha kolaydır. Benzetim dili, programın çalışması sırasında dinamik depolama özelliğine sahiptir. Fakat Genel amaçlı dillerde çok iyi yazılmış bir programın çalışma zamanı, benzetim dili kullanılarak yazılmış programın çalışma zamanından daha az olabilir. Genel amaçlı diller programlamada büyük esneklik sağlar.

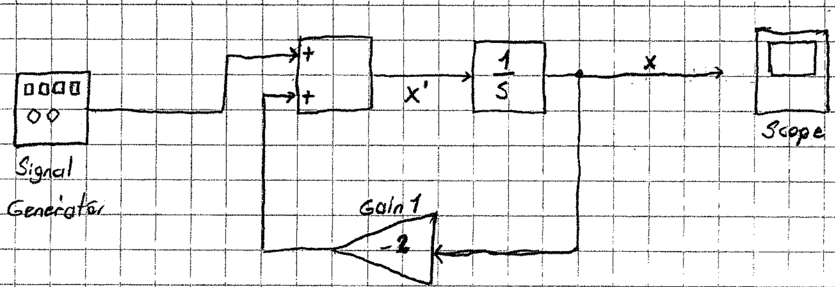
– SİMULİNK –

Simulink karmaşık sistemleri tasarlama ve simülasyon yapma olanağı verir.

Diferansiyel denklemin modellenmesi:

ÖRNEK: $x'(t) = -2x(t) + u(t)$ şeklinde bir denklem verilsin.

İntegral alıcı girişin integralini alır ve x değişkenini üretir.



- Aynı işlemi transfer fonksiyonu kullanarak da yapabilirdik.

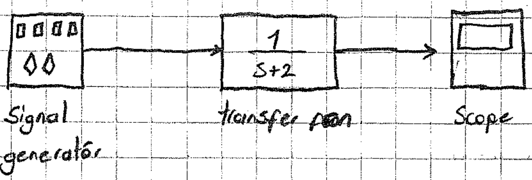
  Model transfer Fcn bloğunu kullanır U girişini alarak x çıkışını verir.

  Dolayısıyla blok x/U işlemini uygular.

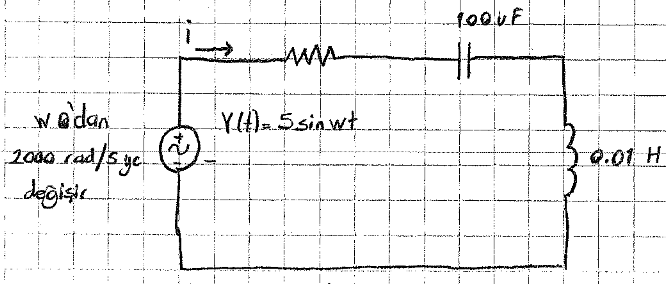
- Yukarıdaki denklemde x' yerine sx yazarsak

$$sX = -2x + U$$

$$X = U/(s+2) \Rightarrow X/U = 1/s+2 \text{ olur.}$$



- Aşağıdaki elektrik devresini simulinkte modelleyelim ve frekans değişimine göre akımı çizdirelim.



Akım sinyalinin genliği frekans w = 1000 rad/s olduğunda maksimum olur.

Akımın ne olacağını gözlemleyelim

- Devre nasıl modellenecek;

$$V = iR + L\frac{di}{dt} + \frac{1}{C}\int i\,dt$$

- Zamana göre dif denklem alınırsa; Laplace uygulanırsa;

$$\frac{1}{L}\frac{dV}{dt} = \frac{di}{dt}\frac{R}{L} + \frac{d^2i}{dt^2} + \frac{i}{LC} \qquad \frac{sV}{L} = \frac{R}{L}sI + s^2I + \frac{I}{LC}$$

$$\frac{sV}{L} = I\left[s^2 + \frac{R}{L}s + \frac{1}{LC}\right]$$

- Böylece akım gerilimden elde edilebilir

$$I = V\left[\frac{s(1/L)}{s^2 + \frac{R}{L}s + \frac{1}{LC}}\right]$$

$$\frac{s(1/L)}{s^2 + \frac{R}{L}s + \frac{1}{LC}} \Rightarrow \frac{s(100)}{s^2 + 1000s + 1\times10^6}$$

– Rastgele sayı üreteçleri –

• Rastgele sayı ve değişken üretimi

1) ORTA KARE YÖNTEMİ:

- m basamaklı tek sayı seçilir.
- sayının karesi alınır (m^2)
- Bulunan sayının ortasından m basamaklı sayıyı al

Bu işlem istenilen sayıda rassal sayı üretilene kadar devam eder.

ÖRNEK: $X_0 = 5497$ orta kare yöntemiyle yapalım.

$X_0^2 = 30.217.009 \Rightarrow X_1 = 2170 \Rightarrow U_1 = 0,2170$ X_1'i 10.000'e böldük

$X_1^2 = (2170)^2 = 4.708.900 \Rightarrow X_2 = 7089 \Rightarrow U_2 = 0,7089$

$X_2^2 = (7089)^2 = 50.253.921 \Rightarrow X_3 = 2539 \Rightarrow U_3 = 0,2539$

2) LCG (Lineer Eşleşiksel Üreteçler):

Z_0: çekirdek, $Z_{k+1} = (aZ_k + c) \pmod{m}$; $U_k = \frac{Z_k}{m}$

a: çarpan, c: artım, m: genlik

ÖRNEK: $a=5$, $c=3$, $m=16$ ve $Z_0=7$ değerleri ile LCG kullanılarak oluşturulan sayı dizisini belirleyelim

$U_0 = \frac{Z_0}{m} \Rightarrow U_0 = \frac{7}{16} \approx 0,437$

$Z_{k+1} = (aZ_k + c) \pmod{m}$

$Z_1 = (5.7+3) \pmod{16}$ $Z_1 = 6 \Rightarrow U_1 = \frac{Z_1}{m} = \frac{6}{16} = 0,375$

$Z_2 = (5.6+3) \pmod{16}$ $Z_2 = 1 \Rightarrow U_2 = \frac{Z_2}{m} = \frac{1}{16} = 0,0625$

- m=16 olduğu için bu örnekte maksimum 16 tane rastgele sayı üretilir.
- m adet tekrar için m adet farklı sayının oluştuğu durumda seçilen LCG'nin tam periyoda sahip olduğu söylenir.
- 16 rastgele farklı elemandan sonra yine ilk eleman gelir.

Hull-Dobel Teoremi:

I. a ve c asal olmalı

II. m sayısının bölünebildiği bütün asal sayılara a-1'de bölünebilmelidir.

III. Eğer m dörde bölünüyorsa a-1 de 4'e bölünebilir.

- Önceki örnekte; 5 ve 3 asal olduğu için şart (I) ✓

m=16 olduğundan 16 sadece 2 asal sayısına bölünür ve a-1=5-1=4'de 2 ye bölünür (şart II) ✓

16 dörde bölünmekte ve a-1'de dörde bölünmektedir (şart III) ✓

- Bütün şartlar sağlandığı için tam periyoda sahiptir.

Üreteçlerin İstatiktiksel Özellikleri

Donanımsal hesaplanabildiği için üretilen sayının karmaşıklığı ve olabildiğince uzun bir periyoda sahip olması istenen bir durumdur. Ayrıca üretilen sayının ve üreteçlerin şu özelliklere sahip olması beklenir.

1) Üreteç tekdüze (Uniform) olmalı: Herhangi bir L uzunluk aralığında oluşan sayıların miktarı, diğer bir L uzunluk aralığında oluşan miktara yakın olmalı.

2) Dizi bağımsız olmalı: Özellikle, herhangi bir sayı bir sonrakine etkisini göstermemelidir.

- Üreteçlerin uniform olup olmadığını test etmek için Ki-kare testi yapılır.
- Ki-kare testi: beklenen frekans değerler ile gözlenen frekans değerlerinin karşılaştırılıp, aradaki uyuma bakılmasıdır.

Uniform Testi (Ki-kare testi):

$$X^2 = \frac{m}{n}\sum_{k=1}^{m}\left(f_k - \frac{n}{m}\right)^2 \qquad v = m-1 \text{ bağımsızlık derecesi}$$

[0,1] aralığında 100 sayı üretilmiş $f_1=21$, $f_2=31$, $f_3=26$, $f_4=22$

$0{,}0 \leq X < 0{,}25$
$0{,}25 \leq X < 0{,}50$
$0{,}50 \leq X < 0{,}75$
$0{,}75 \leq X < 1{,}00$

$n=100$, $m=4$ sınıf var

$$X^2 = \frac{4}{100}\cdot\left[(21-25)^2+(31-25)^2+(26-25)^2+(22-25)^2\right] = 2{,}48$$

X^2 değeri $\alpha = \%95$ $X_c^2 = 7{,}81$

$X^2 < X_c^2$ olduğundan Uniformdur denir.

$2{,}48 < 7{,}81$

TEK DÜZE OLMAYAN RASTGELE DEĞİŞKENLERİN ÜRETİMİ

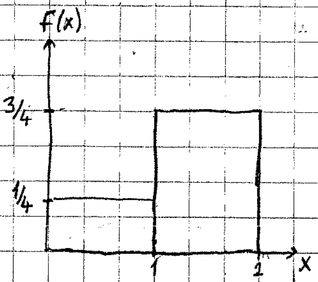
1) TERS DÖNÜŞÜM METODU: $F(x)$ olasılık yoğunluk fonksiyonunun verildiğini kabul edelim. Amaç $f(x)$'den bir rassal değişken üretmektir. Algoritması;

a) $U \leftarrow U(0,1)$ aralığında rasgele sayı üret

b) $X = f^{-1}(U)$ den X sayısını üret

c) Return

ÖRNEK:



$$f(x) = \begin{cases} \frac{1}{4} & 0 \leq x < 1 \\ \frac{3}{4} & 1 \leq x < 2 \end{cases}$$

(1)

- $U = f(x) = \int_0^x \frac{1}{4}\,dt \quad 0 \leq x \leq 1$

$$U = \frac{1}{4}x \Rightarrow x = 4U$$

- 1. eşitsizlikte X yerine $4U$ yazarsak;

$$0 \leq 4U < 1 \Rightarrow 0 \leq U < \frac{1}{4}$$

yani $\begin{cases} x = 0 & U = 0 \\ x = 1 & U = \frac{1}{4} \end{cases}$

(2)

- $U = \int_0^1 \frac{1}{4}\,dt + \int_1^x \frac{3}{4}\,dt \Rightarrow U = \frac{3}{4}x - \frac{2}{3}$

$U = \frac{3}{4}x + \frac{2}{3}$ burdan $\begin{cases} x = 1 & U = \frac{1}{4} \\ x = 2 & U = 1 \end{cases}$ dir.

- $X = \frac{4}{3}U + \frac{2}{3} \qquad \frac{1}{4} \leq U < 1$

- genel eşitsizliği $\Rightarrow$

$$f^{-1}(U) = \begin{cases} 4U & 0 \leq U < \frac{1}{4} \\ \frac{4}{3}U + \frac{2}{3} & \frac{1}{4} \leq U < 1 \end{cases}$$

- Algoritması:

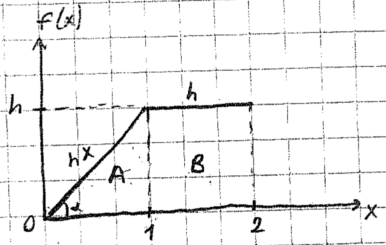
1) $U \rightarrow U(0,1)$

2) if $U < \frac{1}{4} \Rightarrow X = 4U$

3) if $U \geq \frac{1}{4} \Rightarrow X = \frac{4}{3}U + \frac{2}{3}$

4) RETURN

ÖRNEK: Aşağıda verilen olasılık yoğunluk fonksiyonuna uygun rassal değişken üreten algoritmayı ters dönüşüm tekniğiyle çıkarınız.



$\tan\alpha = \frac{h}{1} = h$

$f_1(x) = hx \qquad f_2(x) = h$

$$f(x) = \begin{cases} hx & 0 \leq x < 1 \\ h & 1 \leq x \leq 2 \end{cases}$$

- $f(x)$'in altındaki toplam alanın 1 olması gerekir;

$A + B = 1 \qquad \frac{h}{2} + h = 1 \Rightarrow h = \frac{2}{3}$

- $f(x)$'in yeni hali;

$$f(x) = \begin{cases} \frac{2}{3}x & 0 \leq x \leq 1 \\ \frac{2}{3} & 1 \leq x \leq 2 \end{cases}$$

- $\int_0^x \frac{2}{3} t\,dt = \frac{1}{3}x^2 \qquad 0 \leq x \leq 1$

- $\int_0^1 \frac{2}{3} t\,dt + \int_1^x \frac{2}{3} t\,dt = \frac{1}{3} + \frac{2}{3}(x-1) \qquad 1 \leq x \leq 2$

- $X = f^{-1}(U) \qquad U = \frac{1}{3}x^2 \qquad X = \sqrt{3U} \qquad 0 \leq U \leq \frac{1}{3}$

- $U = \frac{1}{3} + \frac{2}{3}(x-1) \qquad X = \frac{3}{2}U + \frac{1}{2} \qquad \frac{1}{3} \leq U \leq 1$

- $$X = f^{-1}(U) = \begin{cases} \sqrt{3U} & 0 \leq U \leq \frac{1}{3} \\ \frac{3}{2}U + \frac{1}{2} & \frac{1}{3} \leq U \leq 1 \end{cases}$$

- Algoritması;

1) $U \sim U(0,1)$
2) if $U < \frac{1}{3} \Rightarrow X = \sqrt{3U}$
3) if $U > \frac{1}{3} \Rightarrow X = \frac{3}{2}U + \frac{1}{2}$
4) RETURN

ÖRNEK: $f(x) = \begin{cases} x, & 0 \le x \le 1 \\ 2-x, & 1 < x \le 2 \\ 0, & \text{diğer durumlar} \end{cases}$ ters dönüşümünü yapalım.

- $U = f_1(x) = \int_0^x x\,dx = \dfrac{x^2}{2} \qquad 0 \le x \le 1$

- $U = f_2(x) = \int_0^1 x\,dx + \int_1^x (2-x)\,dx = \dfrac{4x - x^2 - 2}{2} \Longrightarrow \dfrac{1-(2-x)^2}{2} \qquad 1 < x \le 2$

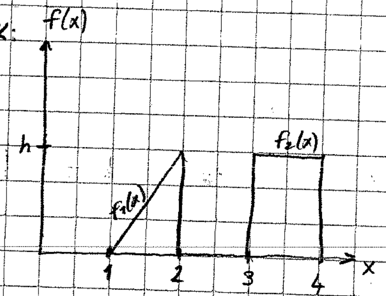
x^2 1 olmamalı

$x = \sqrt{2U} \qquad 0 \le U \le \dfrac{1}{2}$

$x = 2 - \sqrt{2.(1-U)} \qquad \dfrac{1}{2} \le U \le 1$

$f^{-1}(x) = \begin{cases} \sqrt{2U}, & 0 \le U \le \frac{1}{2} \\ 2 - \sqrt{2(1-U)}, & \frac{1}{2} \le U \le 1 \end{cases}$

ÖRNEK:



şekildeki f(x) fonksiyonundan ters dönüşüm tekniği ile rassal değişken üreten algoritmayı bulunuz.

- Alanlar toplamını 1'e eşitleyerek h'i bulalım $\Rightarrow \dfrac{h}{2} + h = 1 \Rightarrow h = \dfrac{2}{3}$

$f_1(x) = m.(x - x_1) \Rightarrow \dfrac{2}{3}.(x-1)$

- yeni eşitsizlik şöyle olur: $f(x) = \begin{cases} \frac{2}{3}(x-1) & 1 \le x \le 2 \\ \frac{2}{3} & 3 \le x \le 4 \end{cases}$

- $\int_1^x \dfrac{2}{3}(x-1)\,dx \Rightarrow U = \dfrac{1}{3}(x^2 - 2x + 1) \qquad 1 \le x \le 2 \qquad 0 \le U_1 \le \dfrac{1}{3}$

- $\int_1^2 \dfrac{2}{3}(x-1) + \int_3^x \dfrac{2}{3}\,dx \Rightarrow U = \dfrac{1}{3} + \dfrac{2}{3}(x-3) \qquad 3 \le x \le 4 \qquad \dfrac{1}{3} \le U_2 \le 1$

- $f^{-1}(U) = \begin{cases} \sqrt{3U} + 1 & 0 \le U_1 \le \frac{1}{3} \\ \frac{3}{2}U - \frac{1}{2} + 3 & \frac{1}{3} \le U_2 \le 1 \end{cases}$

Algoritması;

1) $U \sim U(0,1)$

2) if $0 \le U < \dfrac{1}{3} \Rightarrow x = \sqrt{3U} + 1$

3) if $\dfrac{1}{3} \le U \le 1 \Rightarrow x = \dfrac{3}{2}U - \dfrac{1}{2} + 3$

4) RETURN

• Deneysel Sürekli Dağılımdan Rassal değişken üretimi

Eğer elimizde veriyi modelleyecek bir fonksiyon yoksa, sadece veriler varsa verilerin deneysel dağılımını kullanmak gerekir. Bu dağılım aslında parçalı lineer bir fonksiyon gibi düşünülerek yapılır. Eldeki deneysel veriler artan sırada sıralanır. Daha sonra her deneysel iki nokta arasındaki eğim bulunur ve kümülatif olasılıklarda kullanılarak değişken üretimi sağlanır.

ÖRNEK: Deneysel veriler; 2,76 1,38 0,8 1,45 1,24

• $X_1 < X_2 < X_3 < X_4 < X_5$ küçükten büyüğe sıralayalım

• $0,8 < 1,24 < 1,45 < 1,83 < 2,76$

• 5 sayı olduğu için her bir aralığın olasılığı $\Rightarrow \frac{1}{5} = 0,2$ olur. $\left(\frac{1}{n}\right)$

• Eğim $= a_i = \dfrac{X_i - X_{i-1}}{1/n}$ • $X = f^{-1}(U) = X_{i-1} + a_i\left(U - \frac{(i-1)}{n}\right)$ • Kümülatif Olasılık $\left(\frac{i}{n}\right)$

Aralık	Olasılık (1/n)	Kümülatif olasılık (i/n)	Eğim (a_i)
$0,0 < X < 0,8$	0,2	0,2	4
$0,8 \leq X < 1,24$	0,2	0,4	2,2
$1,24 < X \leq 1,45$	0,2	0,6	1,05
$1,45 < X \leq 1,84$	0,2	0,8	1,90
$1,83 < X \leq 2,76$	0,2	1,0	4,65

$a_1 = \dfrac{0,8 - 0,0}{0,2} = 4$ $\quad a_2 = \dfrac{1,24 - 0,8}{0,2} = 2,2$

• $\dfrac{i-1}{n} < U < \dfrac{i}{n}$

ÖRNEK: 100 makinenin tamir zamanları veri olarak toplanmıştır. Bu veriler farklı aralıklarda gözlemlerin sayısı olarak kaydedilmiş ve şu şekilde verilmiştir.

i	Aralık (saat)	Frekans	İlgili Olasılık	Kümülatif Olasılık (c_i)	Eğim
1	$0,25 \leq X < 0,5$	31	0,31	0,31	0,81
2	$0,5 \leq X < 1,0$	10	0,10	0,41	5,0
3	$1,0 \leq X < 1,5$	25	0,25	0,66	2,0
4	$1,5 \leq X < 2,0$	34	0,34	1,0	1,47

$a_1 = \dfrac{X_i - X_{i-1}}{C_i - C_{i-1}} = \dfrac{0,5 - 0,25}{0,31} =$

$a_2 = \dfrac{1 - 0,5}{0,10} = 5$

• $U = 0,83$'e karşılık gelenler şöyle hesaplanır;

$\dfrac{i-1}{n} < U < \dfrac{i}{n} \longrightarrow$ Kümülatif olasılıklar

$0,66 < 0,83 < 1 \longrightarrow$ 4. durumu alırız

Eğim $= 1,47$, ilgili olasılık $= 0,34$

• Ayrık deneysel verilerden Rastgele değişken üretimi

Bütün ayrık dağılımlar ters dönüşüm tekniğiyle üretilebilir. Ayrık verilerden rassal değişken üretilirken genellikle look-up table dediğimiz tablolardan faydalanılır. Burada yine olasılık yoğunluktan faydalanılır.

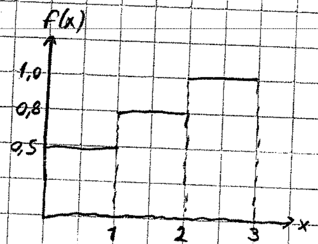
ÖRNEK: Herhangi bir günün sonunda kargoya verilen gönderilerin sayısı 0,1,2 olmaktadır. 0 olma olasılığı %50, 1 olma olasılığı %30, 2 olma olasılığı ise %20'dir. Buna göre rassal değişken üreten bir denklem geliştirelim.

x	P(x)	Kümülatif olasılık
0	0,5	0,5
1	0,3	0,8
2	0,2	1,0

$$f(x) = \begin{cases} 0, & x < 0 \\ 0{,}5, & 0 \leq x < 1 \\ 0{,}8, & 1 \leq x < 2 \\ 1{,}0, & 2 \leq x \end{cases}$$

ÖRN; U = 0,73 için 0,5 olur.

P(0) = 0,5
P(1) = 0,3
P(2) = 0,2



$$X = \begin{cases} 0, & U \leq 0{,}5 \\ 1, & 0{,}5 < U \leq 0{,}8 \\ 2, & 0{,}8 < U \leq 1{,}0 \end{cases}$$

— Sorular —

1) Benzetim modellerini sınıflandırarak açıklayınız?

- Statik veya dinamik
- Belirli veya Olasılıklı
- Kesikli veya Sürekli

- Statik benzetim modeli: Sistemin belirli bir anındaki gösterimidir.
- Dinamik benzetim modeli: Sistemin çalışma zamanına göre yapılan modellemedir.
- Belirli benzetim modeli: Rassal değişken içermeyen benzetim modelidir.
- Olasılıklı benzetim modeli: Bir veya birden fazla rassal değişken içeren benzetim modelidir.
- Kesikli: Kesikli zamanda değişen benzetim modeli
- Sürekli: Zaman üzerinde sürekli değişen benzetim modelleridir.

2) Monte Carlo benzetimi nedir?

Montecarlo metodunda istatikstiksel ve matematiksel tekniklerle bir deneyi veya çözülmesi gereken bir fiziksel olayı tesadüfi sayıları defalarca kullanarak simülasyon edilip çözmek esastır.

3) Sinx'in matlab kodunu yazınız?

```
function sonuc = montecarlo (n)

    x_max = 3,14159;
    y_max = 1;
    hitcount = 0;
    for i=1: iter
        x=rand*x_max;
        y=rand*y_max;
        if (y< sin(x))
                hitcount = hitcount + 1;
        end
    end
    sonuc = (hitcount / iter)*x_max*y_max;
```

4) Ters dönüşüm metodunu uygulayınız?

$$f(x) = \begin{cases} \frac{1}{3}, & 0 \leq x < 2 \\ \frac{1}{24}, & 2 < x \leq 10 \\ 0, & \text{diğer durumlar} \end{cases}$$

$$U = f(x) = \int_0^x \frac{1}{3}\,dt \Rightarrow U = \frac{x}{3}, \quad x = 3U$$

$$U = f(x) = \int_0^2 \frac{1}{3}\,dt + \int_2^x \frac{1}{24}\,dt \Rightarrow U = \frac{2}{3} + \frac{x-2}{24} \quad U = \frac{x+14}{24} \quad x = 24U - 14$$

$$f^{-1}(x) = \begin{cases} 3U, & 0 \leq U \leq \frac{2}{3} \\ 24U-14, & \frac{2}{3} < U \leq 1 \\ 0, & \text{diğer durumlar} \end{cases}$$

1) $U \rightarrow U(0,1)$

2) if $U \leq \frac{2}{3} \Rightarrow x = 3U$

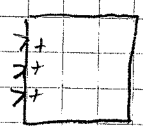
3) if $U \geq \frac{2}{3} \Rightarrow x = 24U - 14$

4) RETURN

5) Para atıldığında yazı gelirse bir sonraki aşamada tura veya yazı gelme şansı %50 dir. Eğer tura gelirse sonraki aşamada tura gelme şansı %75 dir. MonteCarlo ile yapınız

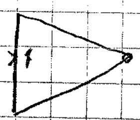
```
function t=montecarlo (n)
olasilik=0,5;
yazi=0;
tura=0;
for i=1:N
  if(rand<olasilik)

    tura= tura+1;
    olasilik=0.75;
  else
    yazi=yazi+1;
    olasilik=0,5;
  end
  end
  t=tura/(yazi+tura);
```

6)

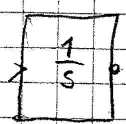


A

Gelen sinyalleri toplar

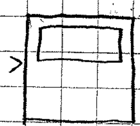


Çarpan bloğu

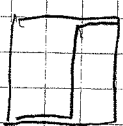


C

integral alıcı



Sinyallerin gösterim aracı



Birim basamak fonksiyonu

- Olay Tabanlı Modelleme -

Zaman sürümlü modeller: Düzenli zaman aralıklarında senkron bir tarzda ilerleyen sinyallere sahip sistemleri karakterize eder.

Olay sürümlü modeller: asenkron olup, düzensiz ve çoğunlukla rastgele aralıklarla oluşan çok basit sinyallere sahiptir. Olayların ne olduğu değil ne zaman oluştuğu önemlidir. Mesajın sistemde olması lojik 1, olmaması lojik 0 ile verilir. Mesajlar FIFO sistemine göre kuyrukta bekler. Banka örneği.

Bir kuyruk sisteminin bileşenleri: Varış prosesi, Servis prosesi, Kuyruk disiplini, Sistemde izin verilen müşteri sayısı, Müşterinin geldiği yığının genişliği

- $E(a) = \frac{1}{\lambda}$ ⇒ Bir dakikada 5 varış olan bir sistemde varışlar arası zaman aralığı ortalaması $E(a) = \frac{1}{5} = 0{,}20$ dk

- Poison dağılıma bağlı varışlar;

t süresinde k varışın olma olasılığı $P(x=k) = \frac{(\lambda t)^k e^{-\lambda t}}{k!}$

λ: Birim zamanda ortalama varış hızı

t: zaman

e: 2,7182

k!: k.(k-1)..

ÖRNEK: Müşteriler Poison dağılımına uygun varış yapmaktadır. Salı 8:00-9:00 = 6 müşteri (ortalama) ise 8:00-8:30 saatleri arasında varış yapma olasılığı nedir?

$\lambda = 6$ müşteri varışı / saat , $t = 30\text{dk} = 0{,}5$ saat , $\lambda t = 3$ müşteri

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

$$P(x=0) = 3^0 e^{-3}/0! = e^{-3} = 0{,}049$$

$$P(x=1) = 3 \cdot e^{-3}/1! = 3e^{-3} = 0{,}149$$

$$P(x=2) = 3^2 \cdot e^{-3}/2! = 9e^{-3}/2 = 0{,}224$$

$$P(x=3) = 3^3 \cdot e^{-3}/3! = 27e^{-3}/6 = 0{,}224$$

$$P(x=4) = 3^4 \cdot e^{-3}/4! = 81e^{-3}/24 = 0{,}168$$

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

a) Hiç kimsenin ölmemesi

b) En az iki kişinin ölmesi

c) 3 kişinin ölmesi olasılıklarını hesaplayınız.

- $P(X=k) = \dfrac{(\lambda t)^k e^{-\lambda t}}{k!}$ X: Bir haftada bu hastalıktan ölenlerin sayısı

a) $P(X=0) = e^{-4} \cdot 4^0 / 0! = 0{,}018$

b) $P(X \geq 2) = 1 - P(X<2) = 1 - (P(X=0) + P(X=1)) = 1 - \left(\dfrac{e^{-4} \cdot 4^0}{0!} + \dfrac{e^{-4} \cdot 4^1}{1!}\right) = 0{,}908$

c) $P(X=3) = e^{-4} \cdot 4^3 / 3! = 0{,}195$

2. Servis prosesi:

- $\mu = 1/E(s)$: Servis oranı (Birim zamanda servis gören müşteri sayısı)

Ortalama servis zamanı 2 dakika ise, servis oranı $\mu = 1/2 = 0{,}5$ servis/dakika

- Trafik yoğunluğu ρ = (varış oranı) / [(servis oranı) * c] , c = servis sayısı

$\rho = \lambda / (\mu * c) = E(s) / (E(a) * c)$

trafik yoğunluğu (ρ)

$\rho < 1$ ise servis $(1-\rho)$ oranında boştur.

$\rho = 1$ ise servis %100 doludur ve kuyruk yoktur.

$\rho > 1$ ise sistemde sürekli artan bir kuyruk oluşur.

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

$\rho = E(s)/E(a) = 2/3 = 0{,}667$ (doluluk oranı)

$= (1-\rho) = 0{,}333$ (Servisin boş kalma oranı)

Analitik ve benzetim modelinde $\rho < 1$ kabul edilir.

Üssel hizmet dağılımının süreye bağlılığı:

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

μ = 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: servis süresi = 4 dk, üssel dağılım, servis zamanının < 3 dk'dan kısa olma olasılığı?

- ortalama servis zamanı = $1/\mu = 4$ dk

ortalama servis hızı = $\mu = 60/4 = 15$ müşteri / saat

- Bir hizmetin 3 dk'dan kısa verilme olasılığı;

3 dk'yı saate çevirelim; $3/60 = 0{,}05$ saat

$P(X < 0.05) = 1 - e^{(-15 \times 0{,}05)} = 1 - e^{(-0{,}75)} = 0{,}52763$

Kuyruk Modeli Notasyonu: (Kendall kuyruk sistemi)

- 1 / 2 / 3 / 4 / 5 / 6

1: varış prosesi, 2: Servis prosesi, 3: Servis sayısı, 4: Paralel servis sayısı, 5: Sistemde izin verilen müşteri sayısı, 6: Müşterinin geldiği yığının genişliği

- 1 ve 2 için;

M: Üstel dağılıma sahip servis yada varışlar arası zaman

D: Sabit servis yada varışlar arası zaman

E_k: k- Erlang dağılmış servis yada varışlar arası zaman

G: Genel dağılım

- 4 için;

FIFO: İlk giren ilk çıkar

SIRO: Rassal sırada servis

PRI: Öncelikli servis

GD: Genel kuyruk disiplini

- M / D / 3 / PRI / 50 / ∞
  - ∞ → Yığın genişliği
  - 50 → Sistemde 50 müşteri sınırı
  - PRI → Öncelikli servis
  - 3 → 3 servis
  - D → Sabit servis süresi
  - M → Varışlar arası zaman üstel dağılım

- Formüller -

• Varışlar		• Hizmet	
- Varış hızı	λ	- Hizmet hızı	μ
- t sürede k varışının olma olasılığı	$\frac{(\lambda t)^k e^{-\lambda t}}{k!}$	- t sürede k hizmetin verilme olasılığı	$\frac{(\mu t)^k e^{-\mu t}}{k!}$
- Varışlar arasındaki ortalama zaman	$1/\lambda$	- Ortalama hizmet zamanı	$1/\mu$
- Herhangi bir varışın t süresinde gerçekleşme olasılığı	$1-e^{-\lambda t}$	- Hizmetin "t" süresinde tamamlanma olasılığı	$1-e^{-\mu t}$
- Müteakip varışın "t" süresinde oluşmama olasılığı	$e^{-\lambda t}$	- Servis süresinin "t" süresinden büyük olma olasılığı	$e^{-\mu t}$

- M / M / 1 kuyruk sistemi

- Gelişler Poison dağılımdadır, Hizmet süresi üssel dağılım sergiler, tekli hizmet sunucu vardır, Kuyruk potansiyel olarak sonsuz uzunluktadır, gelen müşteri sayısı sonsuzdur.

Performans ölçütleri;

- $P_0 = 1-(\lambda/\mu)$ ⇒ Sistemde müşteri olmama olasılığı
- $P_n = [1-(\lambda/\mu)](\lambda/\mu)^n$ ⇒ Sistemde n müşteri olma olasılığı
- $L = \lambda/(\mu-\lambda)$ ⇒ sistemdeki ortalama müşteri sayısı
- $L_q = \lambda^2/[\mu(\mu-\lambda)]$ ⇒ Sıradaki ortalama müşteri sayısı
- $W = 1/(\mu-\lambda)$ ⇒ Sistemde bir müşteri tarafından harcanan ortalama zaman
- $W_q = \lambda/[\mu\cdot(\mu-\lambda)]$ ⇒ Sırada bir müşteri tarafından harcanan ortalama zaman
- $P_W = \lambda/\mu$ ⇒ Varış yapan müşterinin hizmet almak için bekleme olasılığı
- $\rho = \lambda/\mu$ ⇒ Hizmet hattının kullanım oranı
- $P(x>t) = e^{-(\mu-\lambda)t}$ ⇒ Bir müşterinin sistemde "t" süresinden fazla bekleme olasılığı

ÖRNEK: Müşteriler, 12 dakikada bir ortalama hızda ve poison dağılıma uygun olarak varış yapmaktadır. Servis hızı ortalama 8 dk /müşteri. şirket yönetimi bu hizmet için performans düzeyinin belirlenmesini istemektedir.

- Veriler;

$\lambda = 1/12$ müşteri/dk $= 60/12 = 5$ müşteri/saat

$\mu = 1/8$ müşteri/dk $= 60/8 = 7{,}5$ müşteri/saat

- performans hesaplamaları;

$P_0 = 1-(\lambda/\mu) = 1-(5/7{,}5) = 0{,}333$

$P_n = [1-(\lambda/\mu)](\lambda/\mu) = (0{,}333).(0{,}666)\hat{}\,n$

$L = \lambda/(\mu-\lambda) = 2$

$L_q = \lambda\hat{}2 / [\mu(\mu-\lambda)] = 1{,}333$

$W = 1/(\mu-\lambda) = 0{,}4$ saat $= 24$ dk

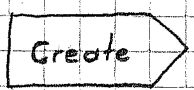
$W_q = \lambda/[\mu\cdot(\mu-\lambda)] = 0{,}266$ saat $= 16$ dk

$P_W = \lambda/\mu = 0{,}666$

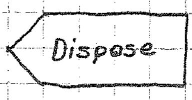
$\rho = \lambda/\mu = 0{,}666$

– ARENA MODÜLLERİ –

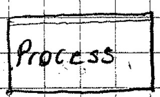
Create: Bu modül bir simülasyon modelinde varlıklar için başlangıç noktası tasarlar.



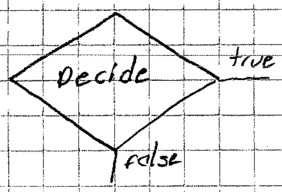
Dispose: Bu modül simülasyon modelinde varlıklar için son noktayı tasarlar. Varlıklar çıkarılmadan önce varlık istatistikleri kayıt edilebilir. Ayrıca modülün içinde sistemden ayrılan varlık sayıları da gösterilir.



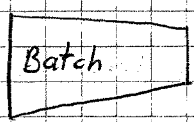
Process: Bu modül simülasyonda ana proses metodunu tasarlar. Kaynak kısıtlarını tutmak ve bırakmak için seçenekler kullanılır. Proses zamanı, varlıkları ayırt eder ve değer eklenen, değer eklenmeyen, taşıma, bekleme ve diğerlerini dikkate alabilir. İşlem önceliğini dikkate alır ve buna göre işlemleri kuyrukta bekletir. Simge üzerindeki çizgi kuyruğu ifade eder.



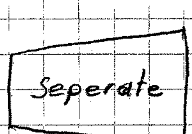
Decide: Bu modül sistemde karar verme prosesi için izin verir. Karar alınmasında bir veya daha fazla koşula yada bir veya daha fazla olasılığa dayanarak seçeneği içerir.



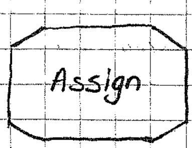
Batch: Bu modül, simülasyon modeli içinde gruplama mekanizmasını tasarlar. Batch'lar sürekli yada geçici olarak gruplanabilir. Gerekli varlık sayısı birikene kadar Batch modülünde varlık gelişleri bir kuyruğa yerleştirilir.



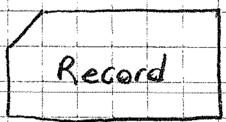
Separate: Bu modül çoklu varlıkların içine gelen bir varlığı kopyalamakta yada önceden oluşturulan bir varlık yığınını bölmekte kullanılabilir.



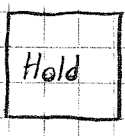
Assign: Bu modül, değişkenlere varlık özelliklerine, varlık tiplerine, varlık resimlerine yada diğer sistem değişkenlerine yeni değer atanması için kullanılır. Tek bir assign modülle çoklu atamalar yapılabilir.



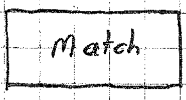
Record: simülasyon modelinde istatistikleri biriktirmekte kullanılır. Modül içinde çıkışlar arası zaman, varlık istatistikleri (zaman, maliyet vb.), genel gözlemler ve ara istatistikleri içerir.



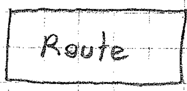
Hold: Bu modülde eğer varlık bir sinyal için tutuluyorsa, sinyal modülü varlığı sonraki modüle geçmek için izin vermede kullanılır. Eğer varlık verilmiş bir halin doğru olması için tutuluyorsa varlık hal doğru oluncaya kadar modülde kalacak



Match: Match modülü farklı kuyruklarda bekleyen varlıkları belli sayılarda gruplar, bir araya getirir. Ayrıca match komutu kullanılmadan önce, kuyruklarda beklemekte olan varlıkların en az bir ortak özellikleri olmalıdır.



Route: Belirtilen bir istasyona bir varlığı transfer eder veya istasyona ziyaret sırasında, sonraki istasyona gezen biçimi tanımlamak için kullanılır.



Station: gezen birimin gideceği yerleri tanımlamak için kullanılır.

Station

Access: Varlığın bir istasyondan diğerine hareketi için konveyörün bir yada daha fazla hücresine yer eder.

Access

Convey: Bulunduğu istasyondan belirtilen varış istasyonuna taşır.

Convey

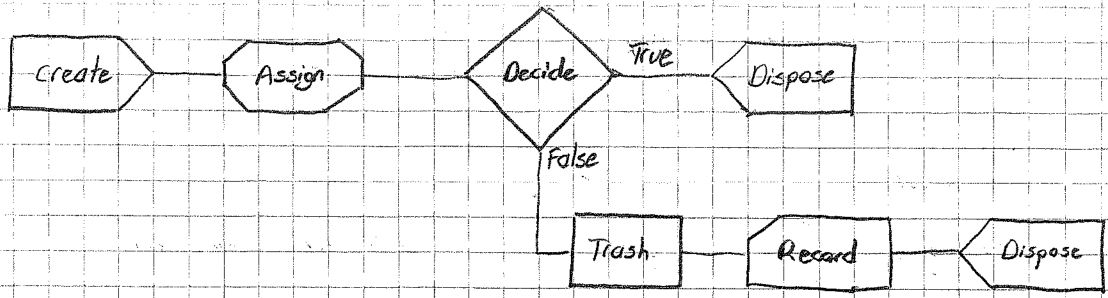
Exit: Access modülü ile Conveyöre alınan gezen birimi herhangi bir işlem için konveyörden almaya yarar.

Request: İstek modülü, bir varlığa bir taşıyıcı ünitesini tayin eder ve varlığın yerine üniteye hareket eder.

Transport: Bu modül yine gezen birimin taşınmasında kullanılır. Bu modülde taşıyıcı sınıflaması vardır.

Free: Bu modül varlığın en son pay edilmiş taşıyıcısını salmak için kullanılır.

Örnek: Bir erkek kuaföründe traş kuyruğu simülasyonu yapılmıştır. Müşteri sırası FIFO mantığıyla çalışmaktadır. Bir müşteri kuaföre girdiğinde traş kuyruğu 3 kişiyse kuaförden çıkmaktadır. Traş kuyruğu 3 kişiden az ise müşteri kuyruğa girerek traş olmaktadır.

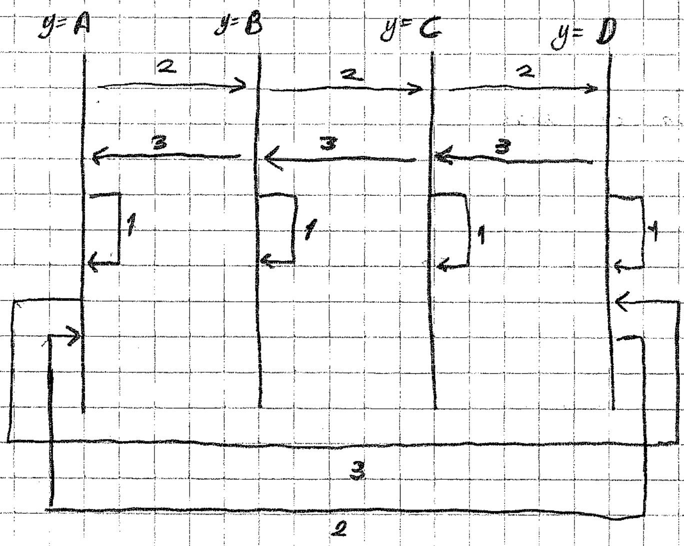


– Durum Makinaları –

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

Geçiş tablosu;

	$y(k)=A$	$y(k)=B$	$y(k)=C$	$y(k)=D$
$d(k)=1$	A	B	C	D
$d(k)=2$	B	C	D	A
$d(k)=3$	D	A	B	C



– Durum diyagramı –

• kodu;

```
k=0
t=to
y=yo
print t,y
for k=1 to n
  t=t+γ
  d=INT(1+3*RND)
  call Φ(d,y,z)
  print t,y,z
next k
```

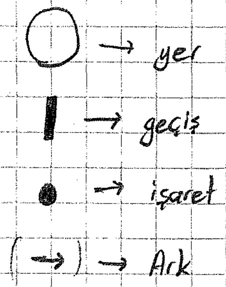
```
subroutine Φ (d,y,z)
  case y
   if y=A
      if d=1 then y1=A ; z1=A
      if d=2 then y1=B ; z1=B
      if d=3 then y1=D ; z1=D
   if y=B
      if d=1 then y1=B ; z1=B
      if d=2 then y1=C ; z1=C
      if d=3 then y1=A ; z1=A
   if y=C
      if d=1 then y1=C ; z1=C
      if d=2 then y1=D ; z1=D
      if d=3 then y1=B ; z1=B
   if y=D
      if d=1 then y1=D ; z1=D
      if d=2 then y1=A ; z1=A
      if d=3 then y1=C ; z1=C
  end case
      y=y1
      z=z1
return
```

• Asenkron durum makinası için sözde kod;

```
      t=to
      y=yo
      print t,y
[1]   t=t-ν*ln(RND)
      d=INT(1+3*RND)
      call Φ(d,y,z)
      print t,y,z
      if t<tmax then goto [1]
```

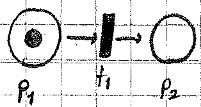
– Petri Ağları –

- Çok işlemcili makineler aynı anda birden fazla işin paralel olarak işlenmesini sağlar. Burada eş zamanlı modellemeye ihtiyaç duyulur. Bu yapılar Petri ağları olarak tanımlanır.



- Petri ağ iki düğümden oluşur; yerler ve geçişler.
- Arklar yalnızca bir yerden bir geçişe yada bir geçişten bir yere olabilir.
- Bir yerde sıfır yada daha fazla işaret bulunabilir.

- Giriş yerlerinin her birinde en az bir işaret varsa, geçiş düğümü iletime hazırdır.

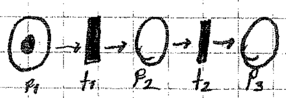


durum geçişi $(1,0) \Rightarrow (0,1)$

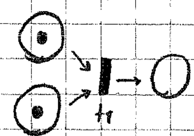
P_1: giriş yeri P_2: çıkış yeri

- Petri ağların özellikleri;

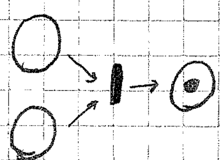
- Sıralı geçiş: t_1 iletiminden sonra t_2 iletimi gerçekleşir



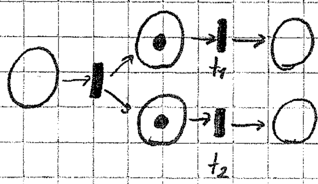
- Senkronize geçiş; giriş yerlerinin her birinde en az bir işaret olduğunda t_1 etkin olacaktır.



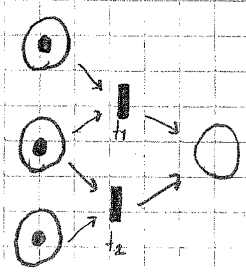
- Birleştirme; işaretler birçok yerden aynı geçiş birimine ulaştığında oluşur

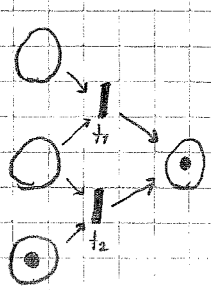


- Eş zamanlılık;



- Çakışma;





ÖRNEK: Şehirler birbirine mesaj iletmektedir;

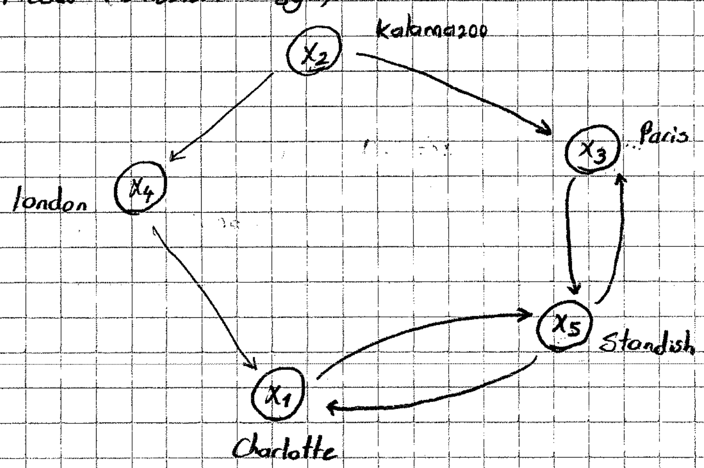
Şehir	Düğüm
charlotte	x_1
Kalamazoo	x_2
Paris	x_3
London	x_4
Standish	x_5

- X vektörü [2,0,1,3,0] olduğundan;

  Charlotte'ın iletmek için 2 mesajı var,

  Paris'in 1 mesajı vs.

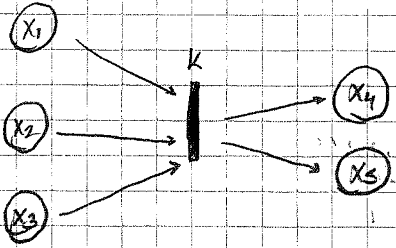
- Mesaj merkezlerinin ağı;



- Şimdi başlangıç durumu $X=[2,0,1,3,0]$ olan ve tüm geçişleri listeleyen bir tablo oluşturalım;

Geçiş	X (gelecek)	
Charlotte → Standish	[1,0,1,3,1]	⇒ Gönderenin bir azalır alanın bir artar.
Standish → charlotte	mümkün değil	
Standish → Paris	mümkün değil	
Standish → Kalamazoo	mümkün değil	
Paris → Standish	[2,0,0,3,1]	
Kalamazoo → Paris	mümkün değil	
Kalamazoo → london	mümkün değil	
London → Standish	[2,0,1,2,1]	⇒ Bu durumda aradaki düğüme bakılmaz
London → Charlotte	[3,0,1,2,0]	

- Olası geçişlerin kümesine eylem kümesi denir. S ile gösterilir.
- S kümesinden rastgele bir eleman seçilir. Bu geçiş k olsun ve herbir durum geçişinde k arttırılarak yeni durumlar tespit edilir. Bu güncellemeyle yeni bir durum oluşur bu olay simülasyon bitene kadar devam eder.



k, S'in bir elemanı ise;

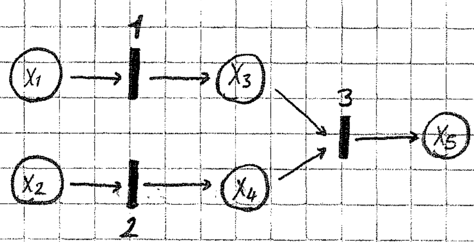
$$\left.\begin{aligned} X_1 &= X_1 - 1 \\ X_2 &= X_2 - 1 \\ X_3 &= X_3 - 1 \\ X_4 &= X_4 + 1 \\ X_5 &= X_5 + 1 \end{aligned}\right\} \text{Seçili düğümlerin güncellenmesi}$$

- Geçiş güncelleme kodu;

```
Set initial place, token distribution
k=0
print k, state
for k=1 to n
        Establish all legal transitions;
            print k, state
next k
```

ÖRNEK: Aşağıdaki petri ağ'da; 1. yer için 8, 2. yer için 5 mesaj tanımlanmış diğer yerlerin mesajı 0 olarak verilmiş. Bu modelde petri ağ kullanarak geçici ve uzun vadeli bir davranış simule edelim.

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



- Bir petri ağ simülasyonu için 3 aşama vardır; İlk olarak tüm geçişler tespit edilir ve işaretlenir. Bunlardan biri rastgele seçilir ve yürütülür. 1. ve 2. geçiş paraleldir fakat hangisinin önce başlayacağını bilemeyiz ve birini ilk başlaması için seçmeliyiz. X_3 ve X_4 mesajları yerine ulaşanacaya kadar 3 geçişi aktif değildir. 3 geçişi mesajını X_5'e ulaştırır ancak bunlar olana kadar X_1 ve X_2 çoktan olmuş olacaktır.

1) tüm yasal geçişlerin belirlenmesi;

```
S = Ø                        ↗ geçişler
if x1>0 then  S=S∪{1}
if x2>0 then  S=S∪{2}
if x3>0 and  x4>0 then  S∪{3}
```

2) yasal geçişlerden birinin rastgele seçilmesi;

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

S: bir karakter stringidir

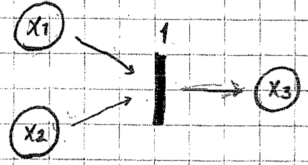
MID(S,i): S den tek bir karakter döndürür.

VAL: bu değeri sayıya dönüştürür.

LEN: karakter sayısını gösterir böylelikle y=RAND(S) ile rastgele geçiş sağlanır.

```
n=20
k=0
read  x1,x2,x3,x4,x5
print  k,x1,x2,x3,x4,x5
for k=1 to n
      S=Ø
      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
```

• AND kapısı

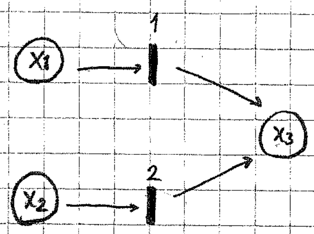


if $x_1>0$ and $x_2>0$ then $y=1$

$S=\{1\}$

if $y=1$ then $x_1=x_1-1$; $x_2=x_2-1$; $x_3=x_3+1$

• OR kapısı



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

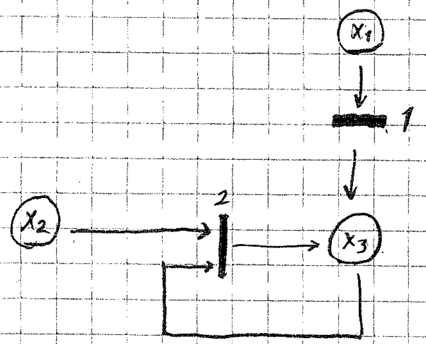
if $x_2>0$ then $y=2$

$S=\{1,2\}$

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

if $y=2$ then $x_2=x_2-1$; $x_3=x_3+1$

• Hafıza



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 $x_2=x_2-1$

ÖRNEK: Bir otomobil sistemi düşünelim. Bu sistemde araç bir anahtar ile çalıştırılır. Daha sonra başta çalışma ve hareket etme eylemleri gerçekleştirilir. Bu diziyi basit petri ağı kullanarak modelleyelim.

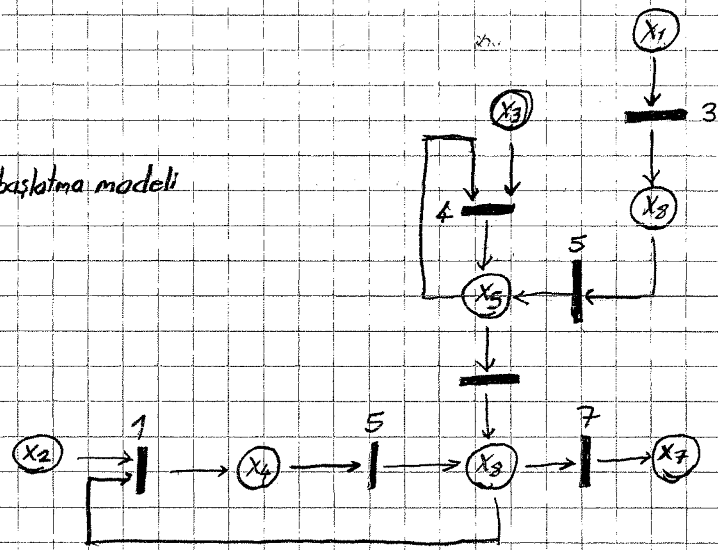
Yerler:
1: Anahtar ateşleme pozisyonunda
2: Tanktaki benzin
3: Akümülatörün şarjı
4: Benzin karbüratörde
5: Dağıtıcı güç alması
6: Motor çalışması
7: Arabanın hareketi
8: Arabanın çalıştırılması

Geçişler:
1: Karbüratöre gazın hareketi
2: Gazın silindire hareketi
3: Başlama pozisyonuna geçiş
4: Distribütöre akım geçişi
5: Kontakların kapatılması
6: Silindirde kıvılcım oluşması
7: Hızın azaltılması

• Başlangıç aşamasında ateşlemede bir anahtar olmalıdır. ($x_1 = 1$) Depoda biraz benzin olmalı ($x_2 = m > 0$) ve akümülatörde şarj olmalıdır. ($x_3 = n > 0$) Burada m ve n daha önceden verilir. Diğer bütün lokal durumlar sıfırdır. $X(0) = [1, m, n, 0, 0, 0, 0, 0]$

• İlk başlamadan sonra belirli bir zaman periyodunda benzin gücü ve bataryadan elektrik akımını korumak için iki hapısa elemanı kullanılır.

• Otomobil başlatma modeli



ÖRNEK: Sonsuz bir tampon ile bir üretici / tüketici sistem düşünelim.

• Üretici süreci bir nesne oluşturur, tüketici süreci ise nesneyi kullanır. Tampon tüketici ve üretici arasında bir arayüz olarak nesne için geçici bir alan tutar. Petri ağıyla modelleyelim.

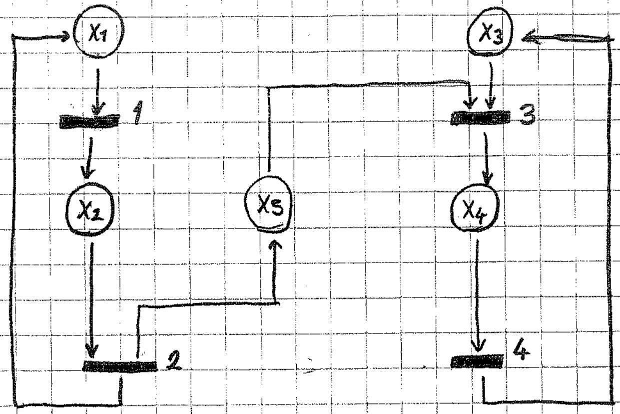
Yerler: X_1: Nesne üretmeye hazır olma
X_2: Nesne eklemeye hazır olma
X_3: Nesne silmeye hazır olma
X_4: Nesne tüketmeye hazır olma
X_5: Tampon nesneye sahip

Geçişler: 1: Nesne üretme
2: Nesne ekleme
3: Nesne silme
4: Nesne tüketme

protokol:
• Bir nesne üretmek
- Nesne üretmek
- Tampona nesne eklemek

• Bir nesne tüketmek
- Tampondan nesne silmek
- Nesne tüketmek

ÜRETİM TÜKETİM



- Bu modelin test edilmesinde "üretime hazır olma" ve "silme için hazır olma" yerleri için bir değerle başlatmak yeterlidir. $X = [x_1, x_2, x_3, x_4, x_5]$

  durum vektörü $X(0) = [1, 0, 1, 0, 0]$ olur.

- Üretici tüketici sisteminin benzetimi

```
read n
read x1, x2, x3, x4, x5
k=0
print k, x1, x2, x3, x4, x5
for k=1 to n
        S = Ø
        if x1>0 then S = S U {1}
        if x2>0 then S = S U {2}
        if x5>0 and x3>0 then S = S U {3}
        if x4>0 then S = S U {4}

        Y = RAND(S)

        if y=0 then print "deadlock!" : stop
        if y=1 then x1=x1-1 : x2=x2+1
        if y=2 then x2=x2-1 : x1=x1+1 : x5=x5+1
        if y=3 then x5=x5-1 : x3=x3-1 : x4=x4+1
        if y=4 then x4=x4-1 : x3=x3+1

          print k, x1, x2, x3, x4, x5

next k
```

Reddetme tekniği

ÖRN: $f(x) = \begin{cases} x & , 0 \le x \le 1 \\ 2-x & , 1 \le x \le 2 \\ 0 & \text{diğer durumlar} \end{cases}$ Reddetme tekniğiyle yapalım.

- fonksiyonların tek tek türevlerini alıyoruz.

$f_1(x) = x \Rightarrow f_1'(x) = 1$ 0 ile 1 aralığında olduğu için sıkıntı yok. $f''(a)$ yaparız

$f_2(x) = 2-x \Rightarrow f_2'(x) = -1$. -1 alınmaz çünkü aralık $1 \le x \le 2$

- şimdi $0 \le x \le 1$, $1 \le x \le 2$ aralıklarını birleştiriyoruz

$$t(x) = \begin{cases} 1 & 0 \le x \le 2 \\ 0, & \text{diğer durumlar} \end{cases}$$

1) $C = \int_0^2 t(x)\,dx = \int_0^2 1\,dx = \quad c = 2$

2) $\Rightarrow r(x) = \frac{t(x)}{c} = \frac{1}{2}$

$$r(x) = \begin{cases} \frac{1}{2} & 0 \le x \le 2 \\ 0 & \text{diğer durumlar} \end{cases}$$ > ters dönüşüm uyguluyoruz

$U = R(x) = \int_0^x \frac{1}{2}\,dx \quad U = \frac{1}{2}x \quad \boxed{X = 2U}$

Algoritma

1) $U_1 \sim U(0,1)$ üret $y = a + U_1(b-a)$

2) $U_2 \sim U(0,1)$ üret

3) $U_2 \le \frac{f(y)}{t(y)}$ ise, $x = y$

Return