Olasılıksal Robotik

Dr. Öğr. Üyesi Erkan Uslu

5

Parametrik Olmayan Filtreler

- **Kalman ve türevleri parametrik ve fonksiyonel olarak tanımlanabilir posterior olasılık dağılımı (gauss dağılımı) varsayımına sahiptir**
- **Parametrik olmayan filtreler ise posterior olasılık dağılımını sonlu sayıda değer ile temsil eder**
  - Durum uzayını ayrıştırarak → Histogram filtresi
  - Rastgele örnek üreterek → Parçacık filtresi
- **Parametrik olmayan filtreler kompleks multimodal dağılımları temsil edebilir.**

Parametrik Olmayan Filtreler

- Parametrik olmayan filtrelerin temsil kabiliyeti parametre sayısı ile doğru orantılıdır
- Posterior olasılık düşük bir karmaşıklığa sahipse (durum uzayı bir durum çevresinde odaklı ve düşük belirsizliğe sahipse) parametre sayısı az olur
- Kompleks posterior olasılıklar için ise parametre sayısı yüksek olur

Histogram Filtresi

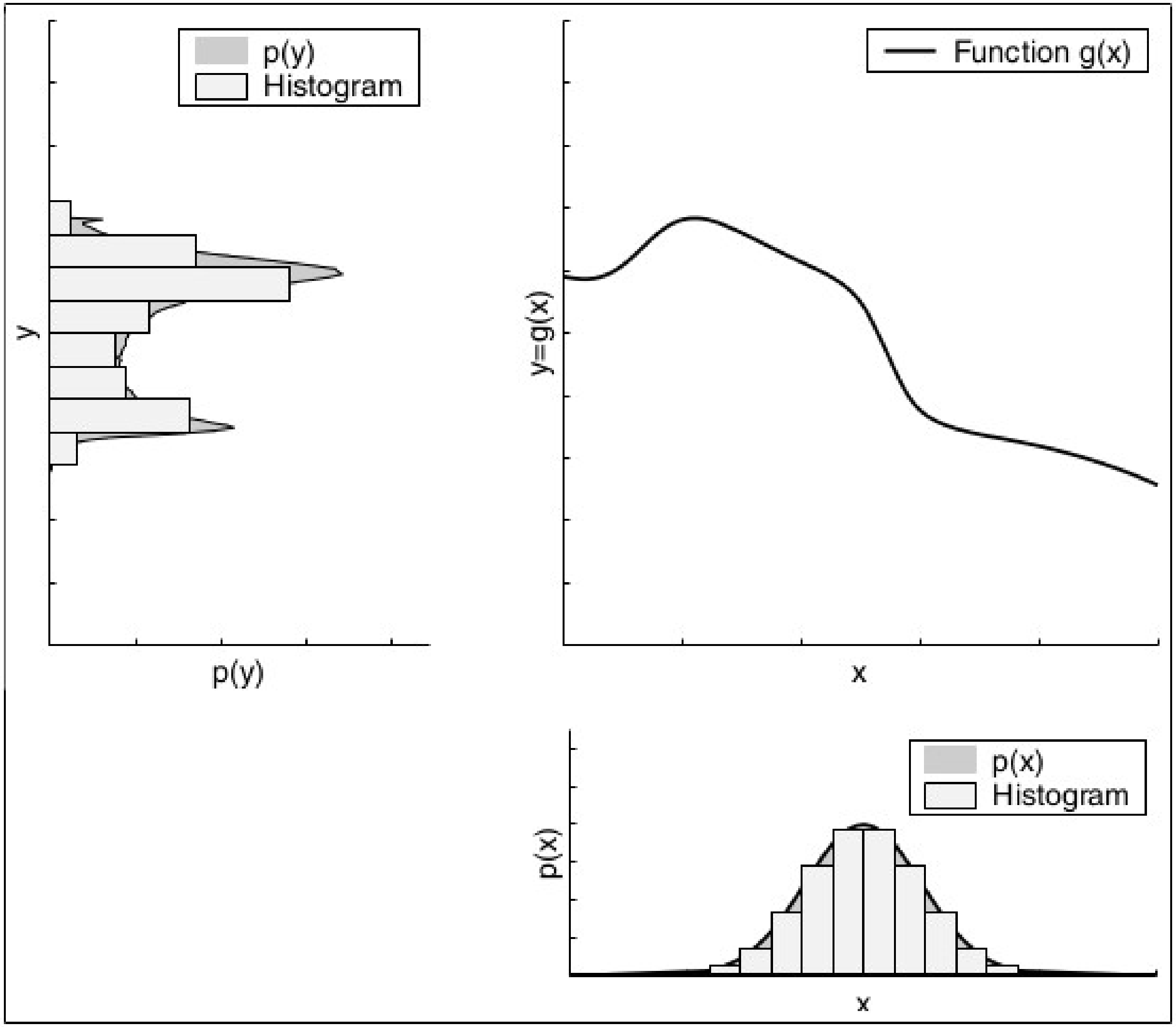
- **Durum uzayı sonlu sayıda bölgeye ayrıştırılır**
- **Durum uzayının yapısına göre**
  - Sürekli durum uzayı → Histogram filtresi
  - Sonlu durum uzayı → Discrete Bayes filtresi

Discrete Bayes Filtresi

Algorithm 1: Ayrık (Discrete) Bayes Filtresi

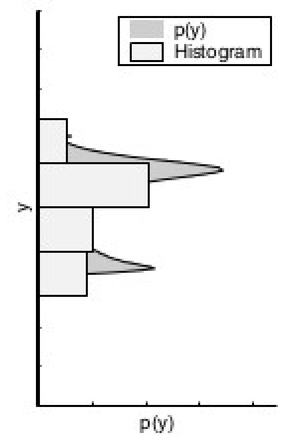
input : $p_{k,t-1}, u_t, z_t$
output: $p_{k,t})$
forall k **do**
    $\overline{p}_{k,t} = \sum_i p(X_t = x_k | X_{t-1} = x_i, u_t) p_{i,t-1}$;
    $p_{k,t} = \eta p(z_t | X_t = x_t) \overline{p}_{k,t}$;
end

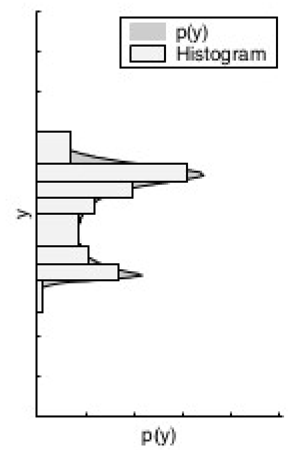
Histogram Filtresi



Histogram Filtresi

- **Ayrıştırma yöntemleri**
- **Statik ayrıştırma:**
  - 1B uzay için eşit aralık binler
  - 2B uzay için eşit aralık ızgaralar (grids)
- **Dinamik ayrıştırma:**
  - Yoğunluk ağaçları (density trees)
- **Hibrid ayrıştırma:**
  - Seçici güncelleme (selective updating)

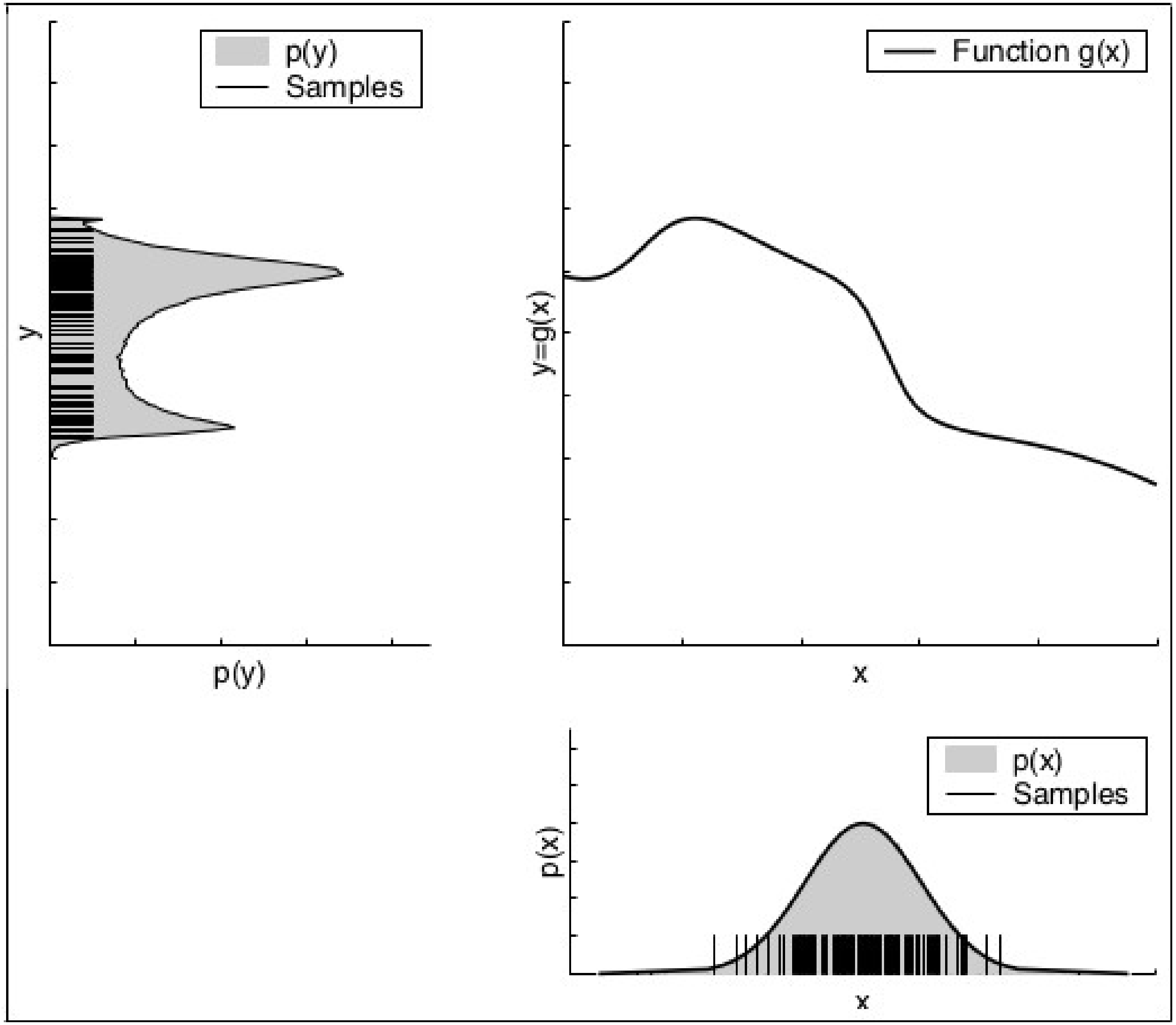




Parçacık Filtresi – Particle Filter

- Posterior olasılık rastgele (random) olarak üretilmiş örnekler ile temsil edilir
- Karmaşık olasılık dağılımları temsil edilebilir
- Random değişkenlerin doğrusal olmayan dönüşümlerinde de iyi temsil kabiliyetine sahiptir

Parçacık Filtresi – Particle Filter



Parçacık Filtresi – Particle Filter

Algorithm 1: Parçacık (Particle) Filtresi

input : χ_{t-1}, u_t, z_t
output: χ_t

$\overline{\chi}_t = \chi_t = \emptyset$;
for $m \leftarrow 1$ ***to*** M **do**
  sample $x_t^{[m]} \sim p\left(x_t | u_t, x_{t-1}^{[m]}\right)$;
  $w_t^{[m]} = p\left(z_t | x_t^{[m]}\right)$;
  $\overline{\chi}_t = \overline{\chi}_t \cup \left\langle x_t^{[m]}, w_t^{[m]} \right\rangle$;
end
for $m \leftarrow 1$ ***to*** M **do**
  sample $x_t^{[i]} \sim w_t^{[i]}$;
  $\chi_t = \chi_t \cup x_t^{[i]}$;
end

Parçacık Filtresi – Particle Filter

Algorithm 1: Parçacık (Particle) Filtresi

input : χ_{t-1}, u_t, z_t
output: χ_t

$\overline{\chi}_t = \chi_t = \emptyset$;
for $m \leftarrow 1$ ***to*** M **do**
  sample $x_t^{[m]} \sim p\left(x_t | u_t, x_{t-1}^{[m]}\right)$;
  $w_t^{[m]} = p\left(z_t | x_t^{[m]}\right)$;
  $\overline{\chi}_t = \overline{\chi}_t \cup \left\langle x_t^{[m]}, w_t^{[m]} \right\rangle$;
end
for $m \leftarrow 1$ ***to*** M **do**
  sample $x_t^{[i]} \sim w_t^{[i]}$;
  $\chi_t = \chi_t \cup x_t^{[i]}$;
end

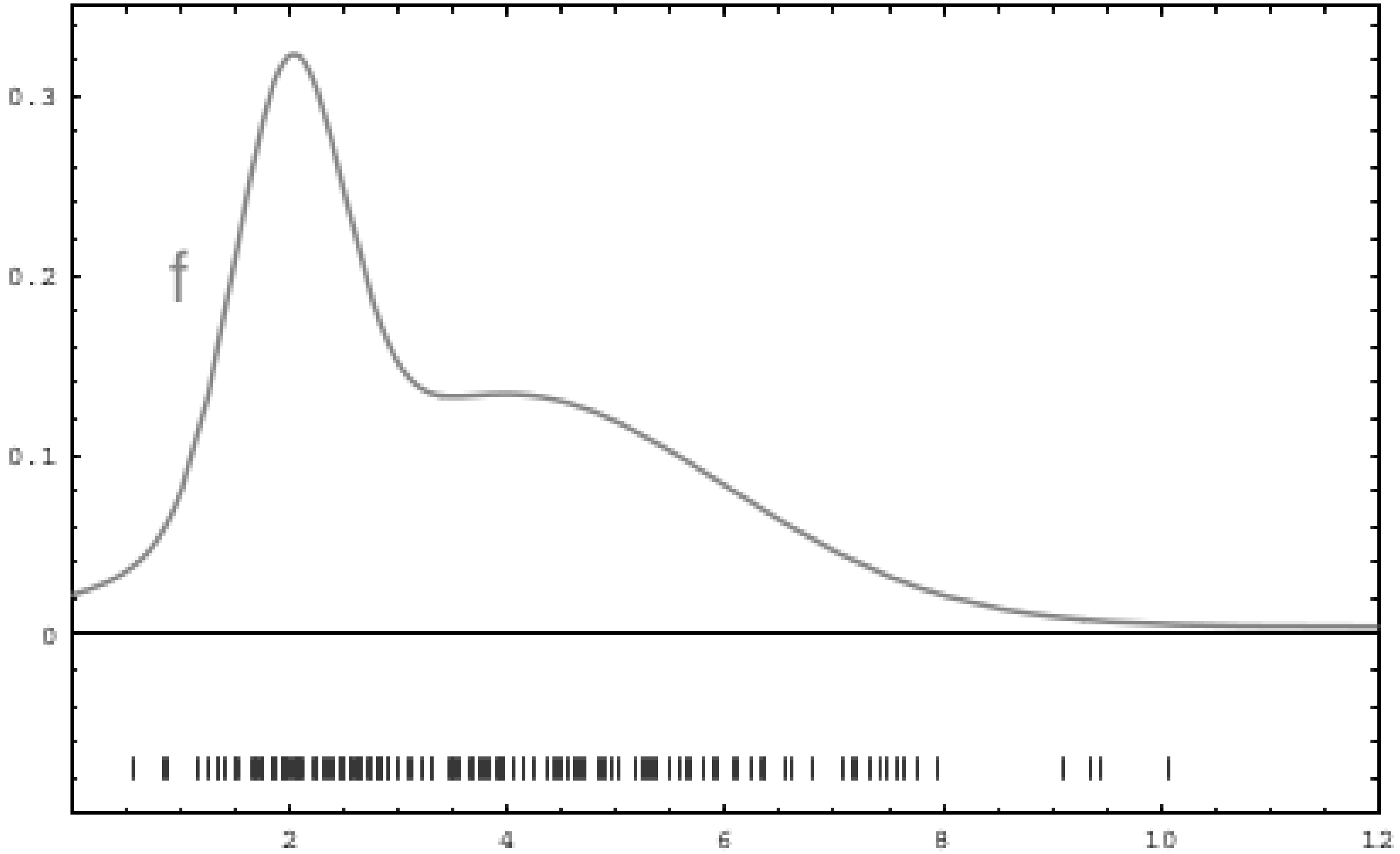
Resampling
Importance sampling

Resampling – Importance Sampling

$$
\begin{aligned}
E_f\left[x\right] &= \int f(x)x dx \\
&= \int \frac{f(x)}{g(x)} g(x) x dx \\
&= \int \omega(x) g(x) x dx \\
&= E_g\left[\omega(x) x\right]
\end{aligned}
$$

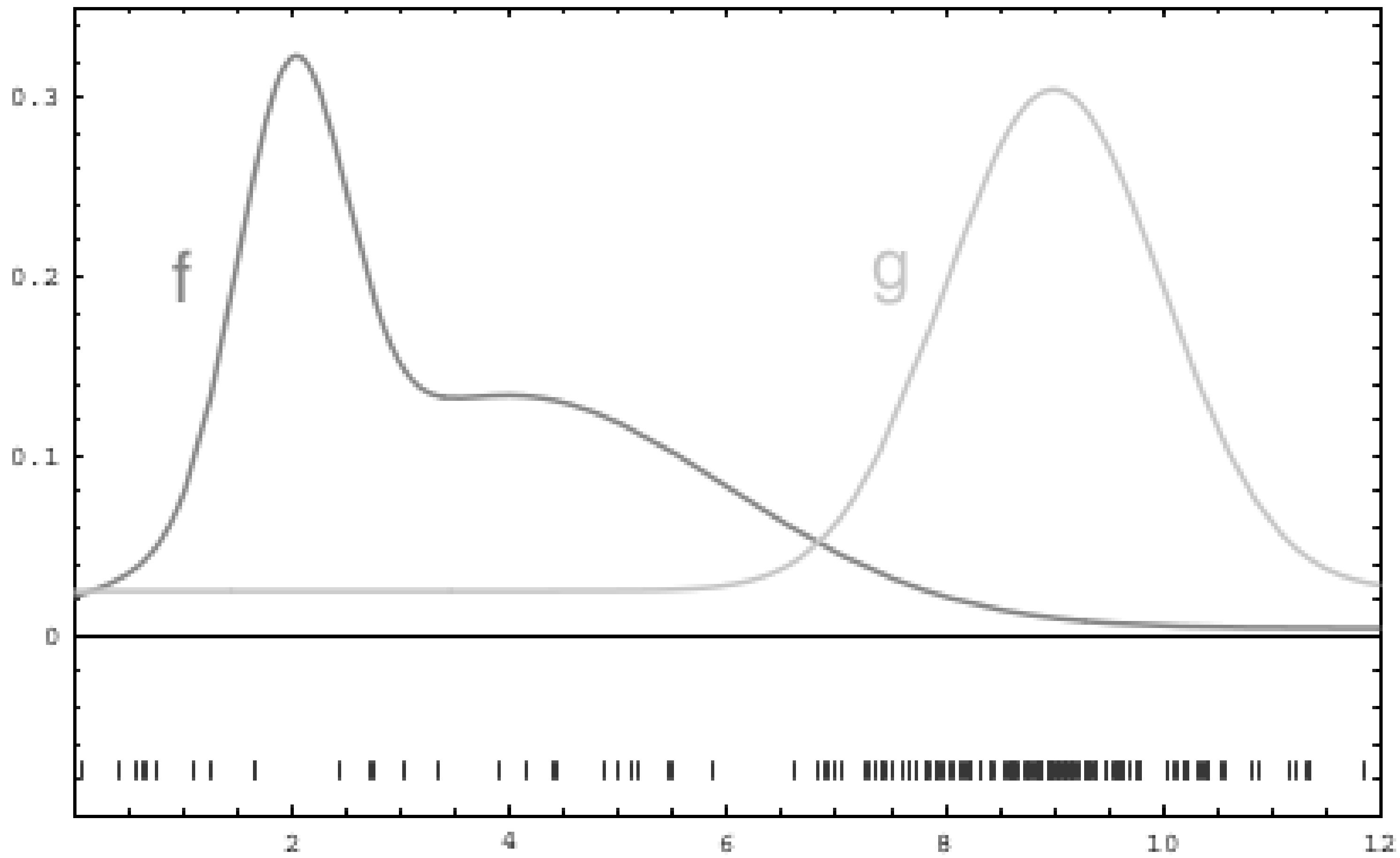
Resampling – Importance Sampling

- Elde etmek istenen temsil



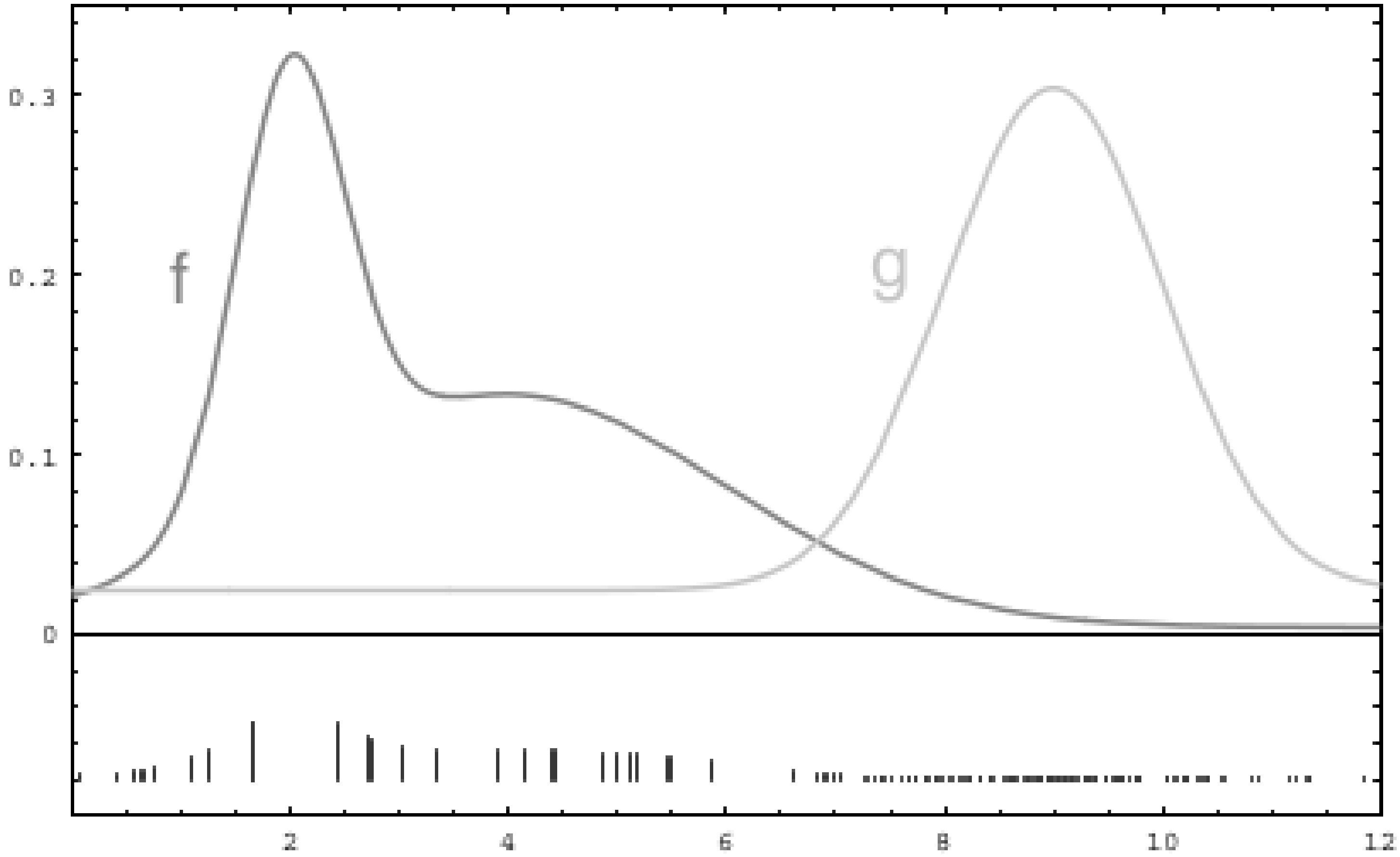
Resampling – Importance Sampling

- Mevcut örnekler



Resampling – Importance Sampling

- Elde edilmek istenen temsile göre ağırlıklandırma



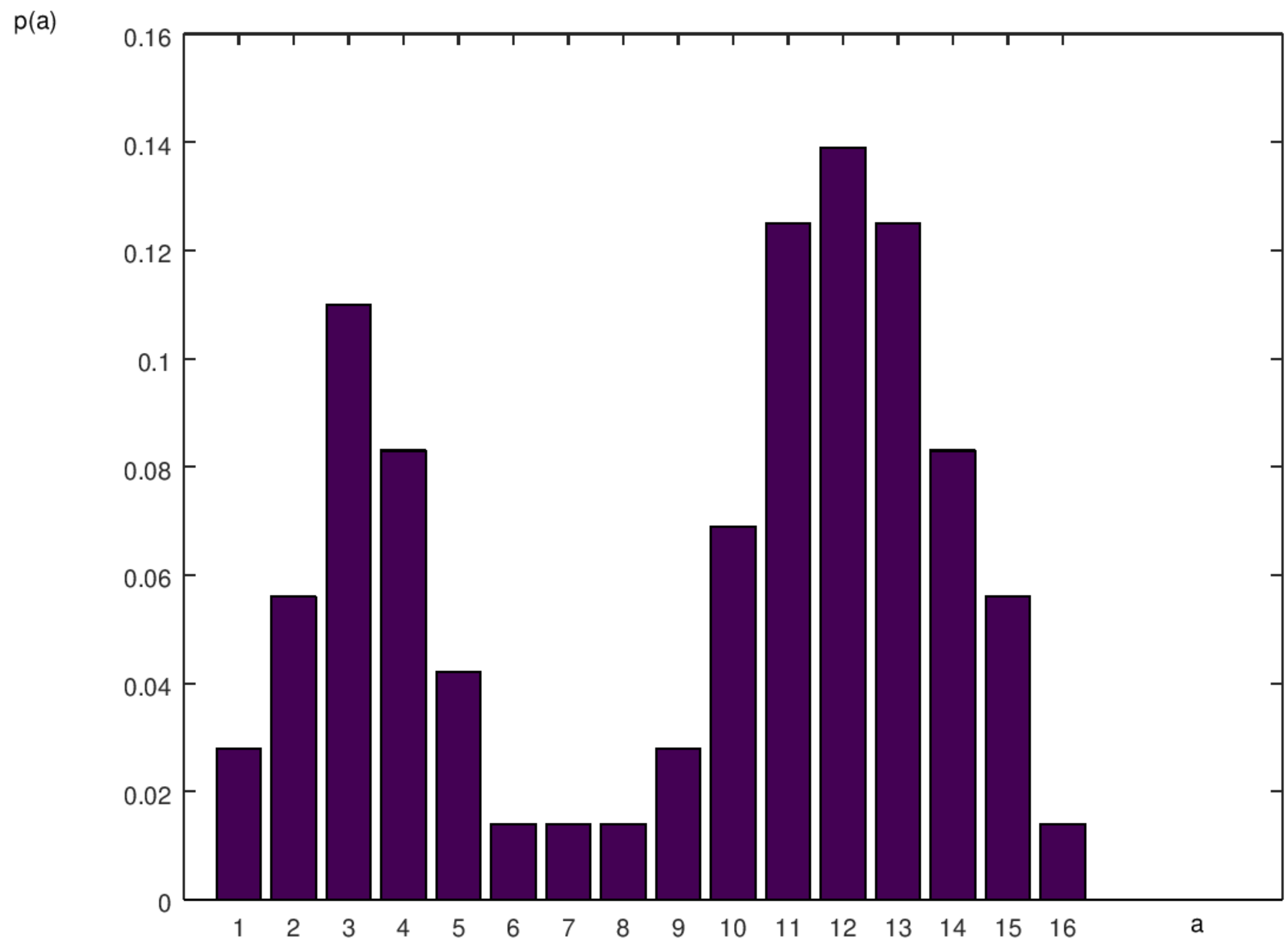
Ters Dönüşüm ile Örnekleme

- P 1B olasılık dağılımına ilişkin PDF üzerinden integral alınarak CDF elde edilir
- Uniform dağılıma uygun bir x random sayısı üretilir
- $CDF^{-1}(x)$ P olasılık dağılımına göre üretilmiş bir random örnek olarak elde edilir

Ters Dönüşüm ile Örnekleme (PDF)

- p(a)=

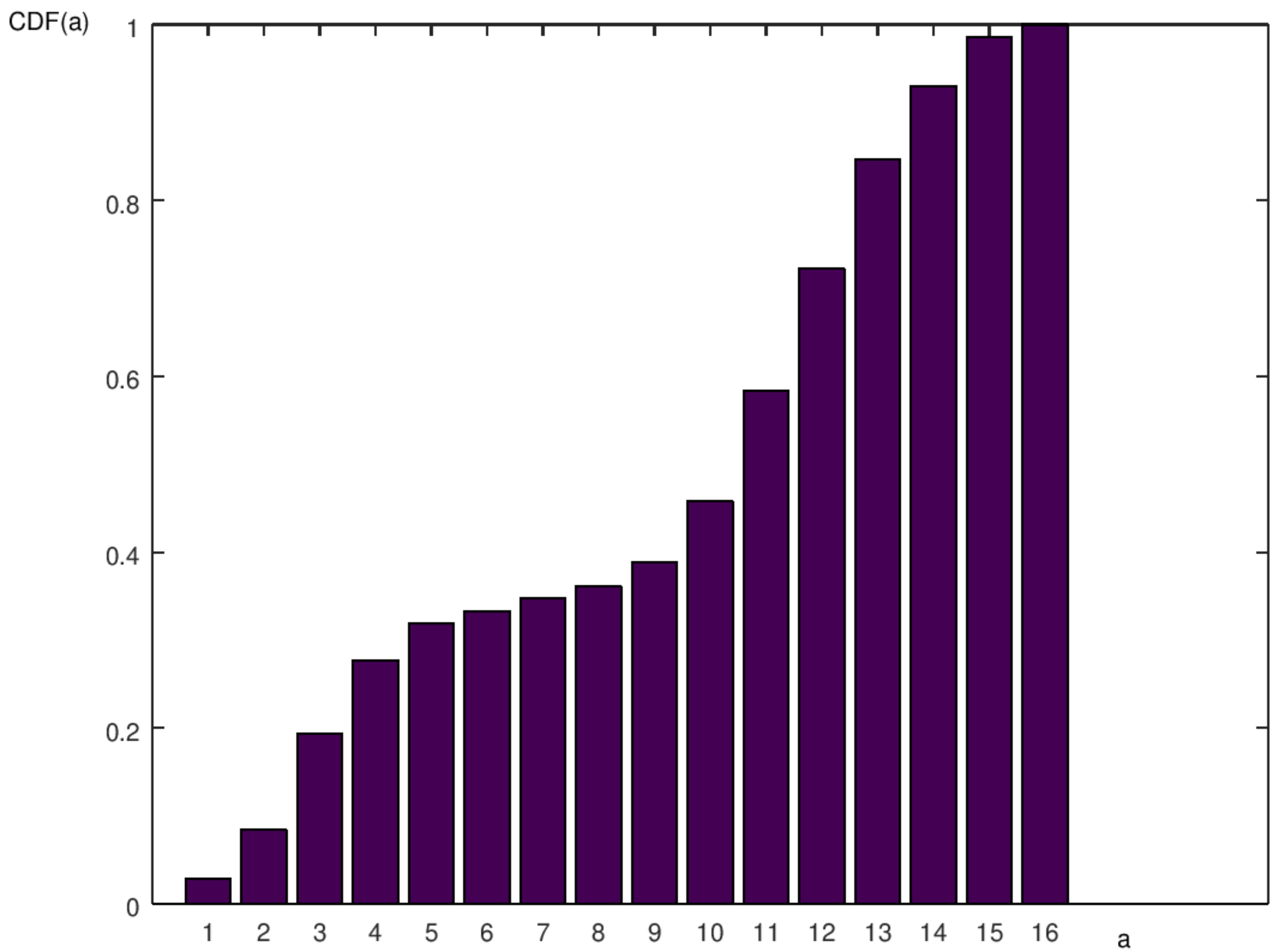
  [0.028, 0.056, 0.110, 0.083, 0.042, 0.014, 0.014, 0.014, 0.028, 0.069, 0.125, 0.139, 0.125, 0.083, 0.056, 0.014]



Ters Dönüşüm ile Örnekleme (CDF)

- p(a)=

  [0.028, 0.056, 0.110, 0.083, 0.042, 0.014, 0.014, 0.014, 0.028, 0.069, 0.125, 0.139, 0.125, 0.083, 0.056, 0.014]



Ters Dönüşüm ile Örnekleme

```
a=[0.028, 0.056, 0.110, 0.083, 0.042, 0.014, 0.014, 0.014, 0.028, 0.069, 0.125, 0.139,
0.125, 0.083, 0.056, 0.014];

bar(a);

b(1)=a(1);
for i=2:16
  b(i)=b(i-1)+a(i);
endfor

figure
bar(b)
```

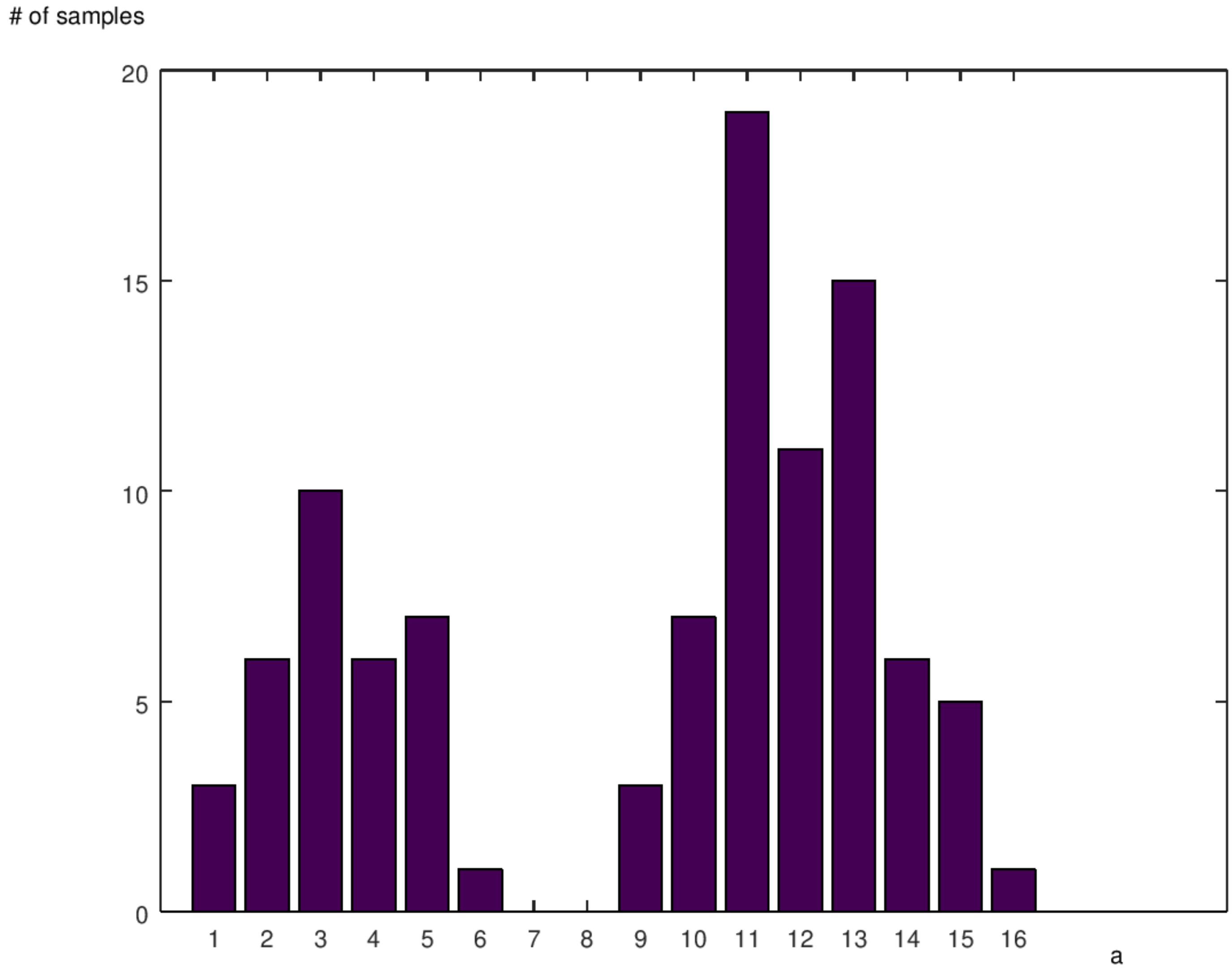
Ters Dönüşüm ile Örnekleme

```
for i=1:100
 c(i)=rand;
endfor

for i=1:100
 j=1;
 while(c(i)>b(j) && j<=16)
  j=j+1;
 endwhile
 s(i)=j;
endfor
```

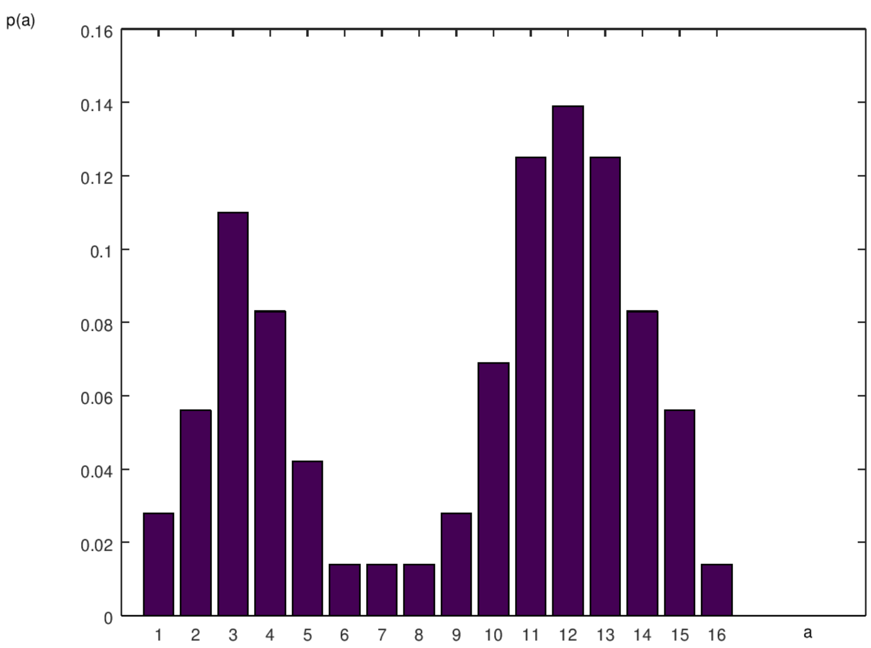
Ters Dönüşüm ile Örnekleme

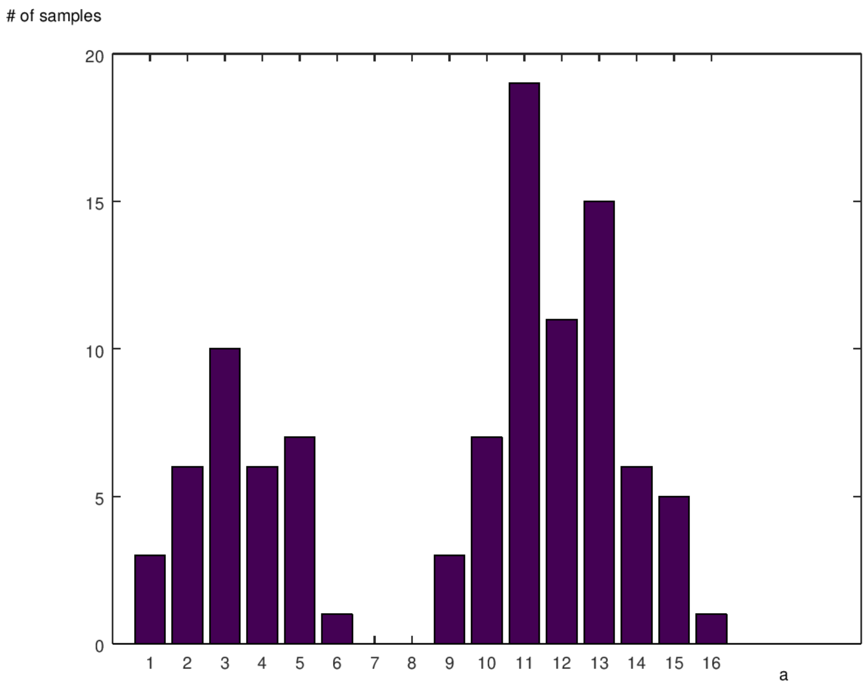
```
for i=1:16
  bin(i)=sum(s==i);
endfor

figure
bar(bin)
```

Ters Dönüşüm ile Örnekleme



Gerçek Dağılım – Üretilen Örnekler

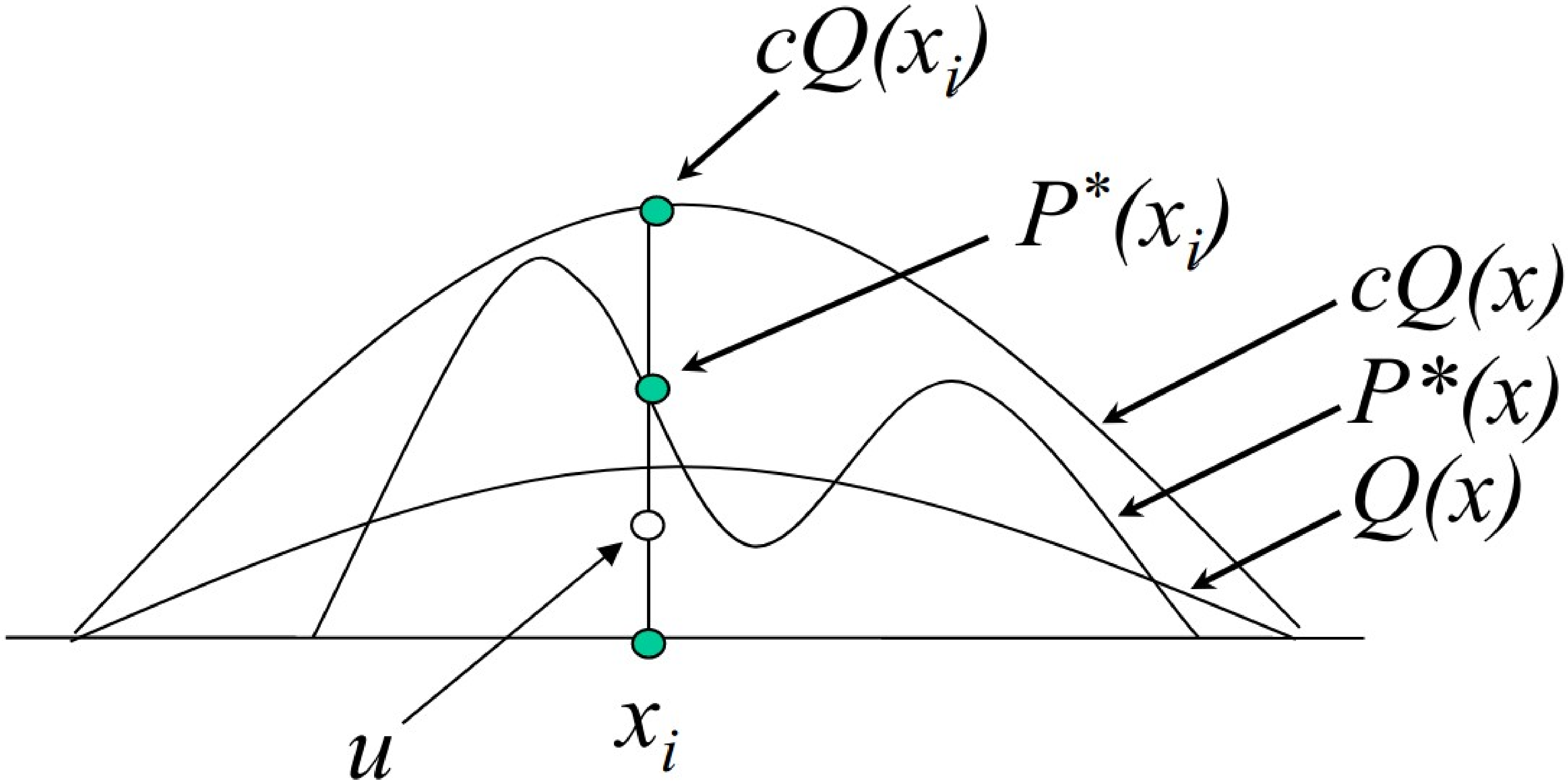




Rejection Sampling

- Bilinen bir Q dağılımı kullanarak P dağılımı için örnekleme yapılmak istenirse
- c bir sabit olmak üzere $cQ \geq P$ olacak şekilde üstten limitleyen bir dağılım kullanılır
- Q’dan bir x örneklemesi yapılır
- [0,cQ] aralığından u uniform bir örnek üretilir
- $u \leq P(x)$ ise örnekleme kabul edilir değilse reddedilir

Rejection Sampling



Normal Dağılım için Örnekleme – Box Muller Metodu

$$
\begin{aligned}
u_1, u_2 &\sim Unif(0,1) \\
R &= \sqrt{-2\log(u_1)} \\
\theta &= 2\pi u_2 \\
(x, y) &= (R\cos\theta, R\sin\theta)
\end{aligned}
$$

Normal Dağılım için Örnekleme – Central Limit Theorem

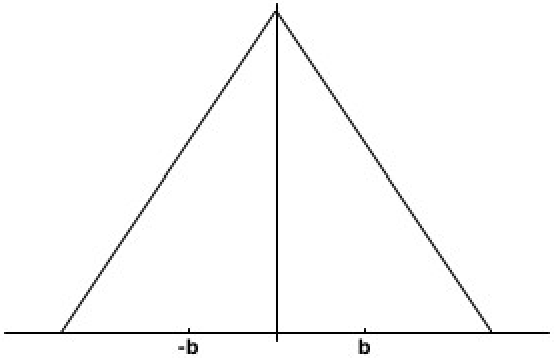
- Central Limit Theorem : bağımsız rassal değişkenler toplandığında bunların normalize edilmiş toplamları normal dağılımı yakınsar
- Probabilistic robotics: b^2 varyanslı, 0 ortalamalı normal dağılımdan örnekleme için

$$\frac{1}{2} \cdot \sum_{i=1}^{12} rand(-b, b)$$

Üçgen Dağılım Örnekleme

- Üçgen dağılım normal dağılımın basitlerştirilmiş bir versiyonu olarak düşünülebilir
- Probabilistic robotics: b^2 varyanslı, 0 ortalamalı üçgen dağılımdan örnekleme için

$$\frac{\sqrt{6}}{2} \cdot [rand(-b, b) + rand(-b, b)]$$



Örnek

- Mobil robot 2B uzayda sadece belirli bir doğrultuda uygulanan kontrol işareti L kadar hareket edebilmektedir. Robot bulunduğu konumun x eksenine izdüşümünü normal dağılıma uygun bir hata terimi ile ölçebilmektedir.
- Robota ilişkin durum tanımı aşağıda verilmiştir.

$$\mathbf{x} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} \qquad \begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x + L\cos\theta \\ y + L\sin\theta \\ \theta \end{pmatrix} \qquad \mathbf{z} = \mathbf{x}^{'}$$

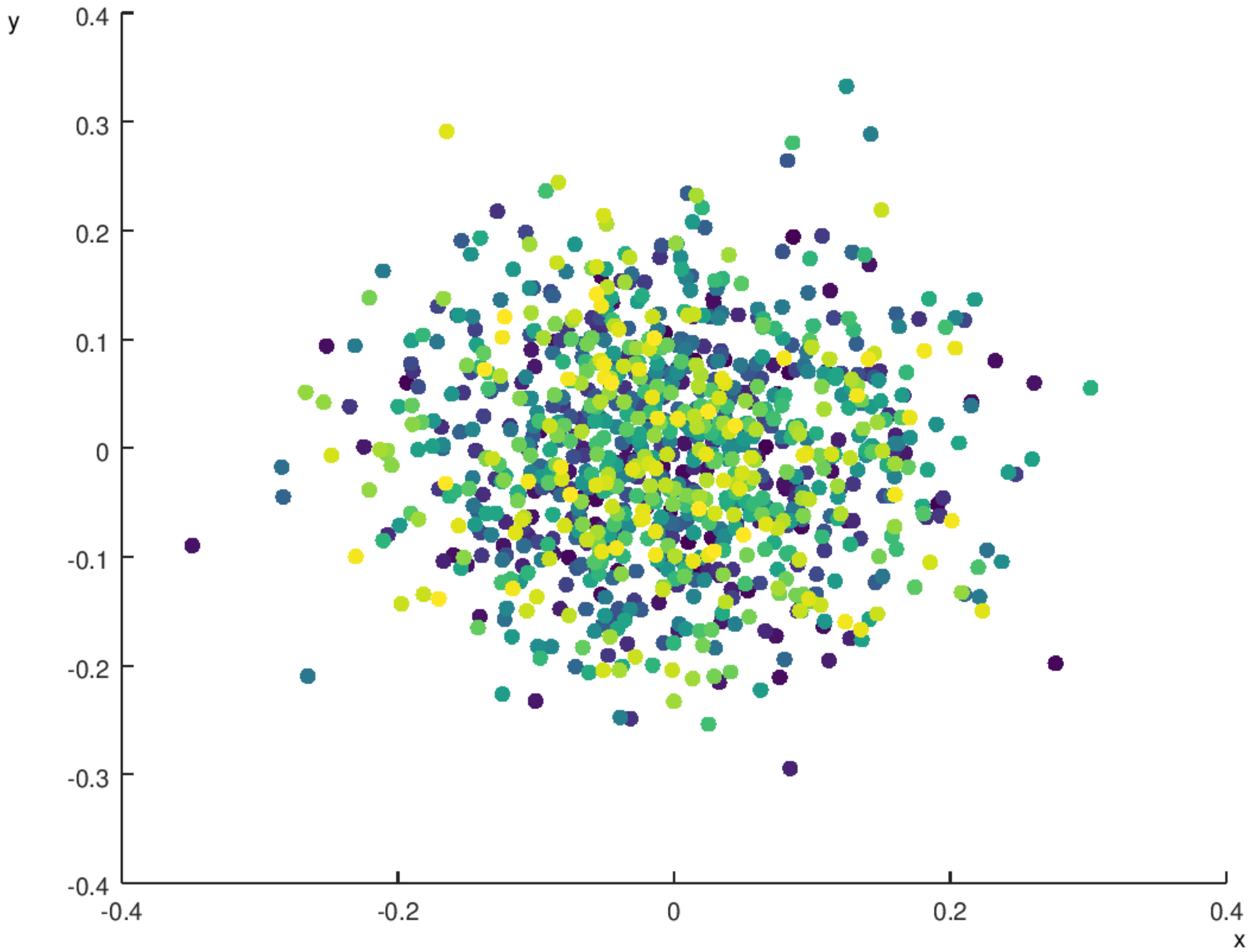
Örnek

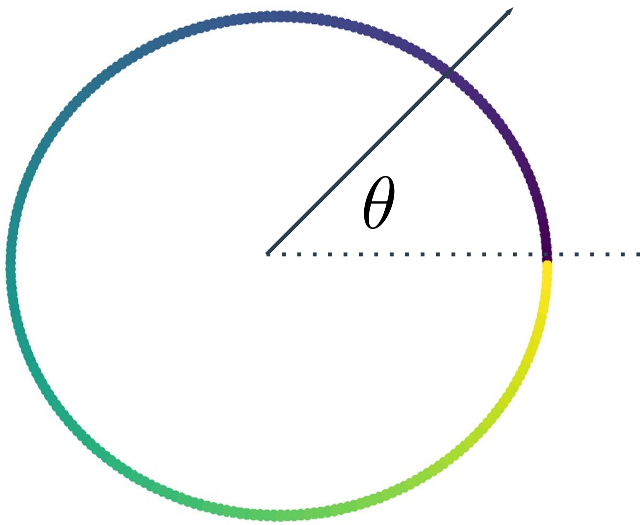
- Robotun başlangıç durum inancı aşağıdaki gibidir

$$\mu_{\mathbf{0}} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \boldsymbol{\Sigma}_{\mathbf{0}} = \begin{pmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 10000 \end{pmatrix}$$

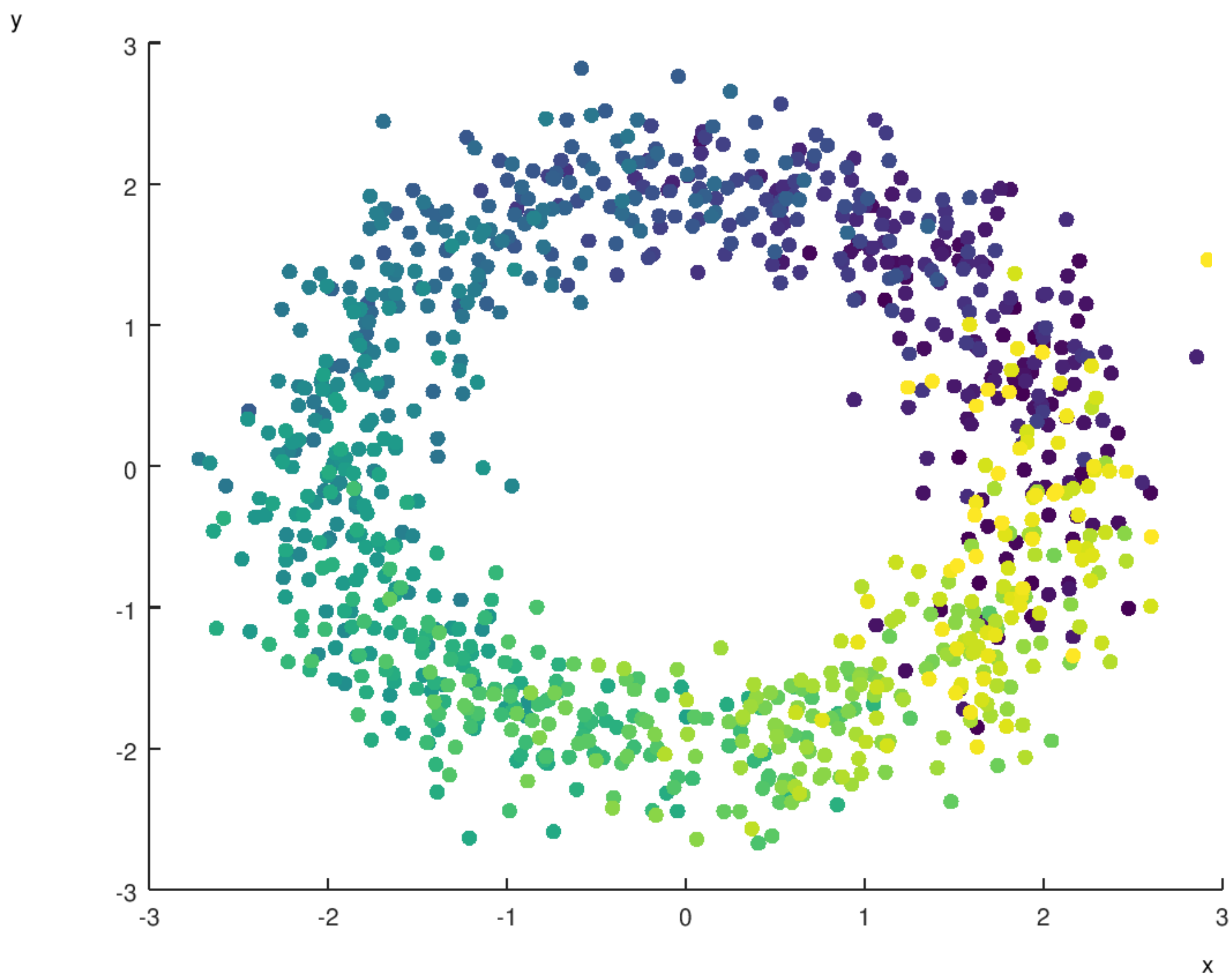
- Başlangıç durumuna göre 1000 adet parçacığın dağılımını sağlayın
- Robotun L=2 birim ilerleme komutunu yürütmesi sonrası parçacıkların dağılımını sağlayın
- Robotun z=1.5 birim ölçüm alması sonucu parçacık dağılımını sağlayın

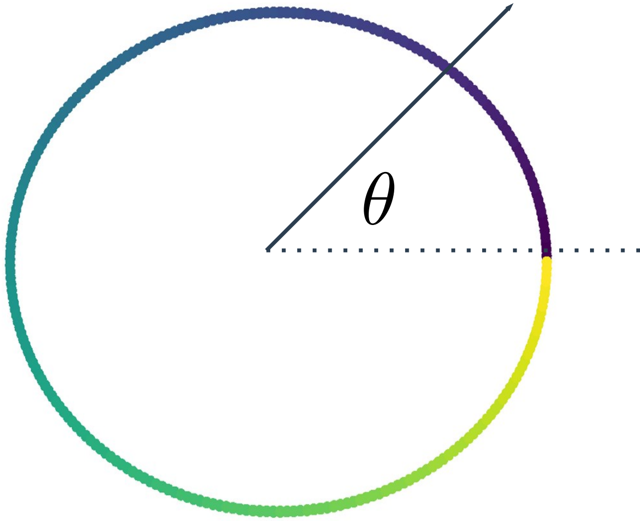
Örnek – Başlangıç konumu için parçacıklar



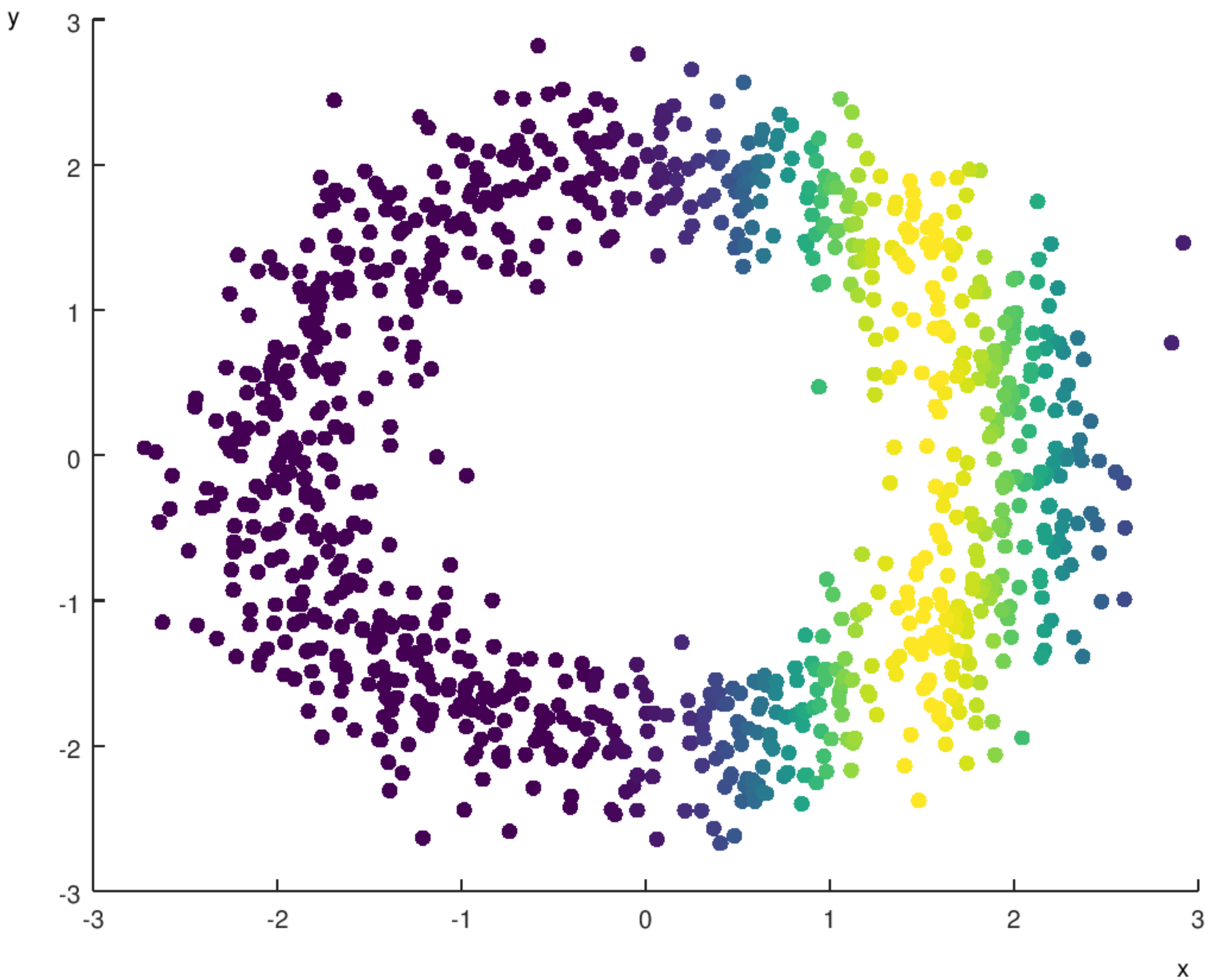


Örnek – L=2 ile hareket sonucu parçacıklar





Örnek – z=1.5 için parçacık ağırlıkları



Örnek – Resampling sonucu parçacıklar