National Cheng Kung University

Mechanical Engineering

Stochastic Dynamic Data - Analysis and Processing

Gaussian & Uniform Distribution

Author:
Zhao Kai-Wen

Supervisor:
Chang Ren-Jung

October 18, 2012

Contents

1 Introduction

The assignment would introduce Gaussian and uniform distribution. Scale and shift their range or parameters and verify the results.

2 Generating Gaussian Sequence

Gaussian distribution is one of the most well-known probabilty distribution in various fields, such as signal processing, statistics, and so on. The standard form is showed below and where standard deviation $\sigma_z = 1$ and mean $\mu_z = 0$

$$f(z) = \frac{1}{\sqrt{2\pi}} e^{\frac{-z^2}{2}}$$

2.1 Create sequence

In $Matlab$, creating a series based on Gaussian distribution is quite facile. We call the function named $normrnd()$ with two arguments, mean and deviation.

$$R = normrnd(0, 1)$$

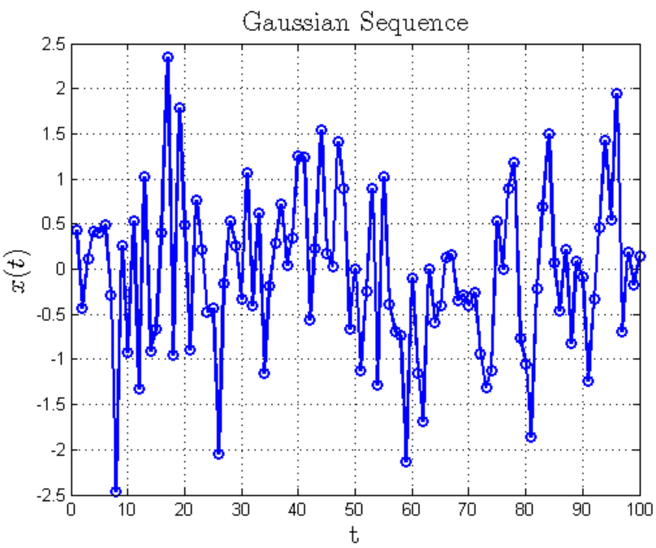


Figure 1: Gaussian Sequence

2.2 Verifiying the distribution

We try to verify these data and comfirm they follow the normal distribtion which I set the mean $\mu_x = 0$ and deviation $\sigma_x = 1$. We can get the result below.

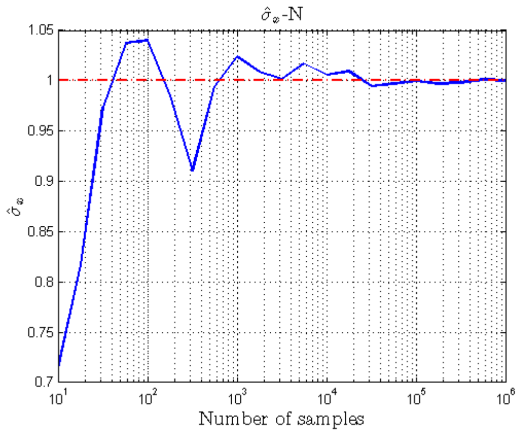


Figure 2: Mean value

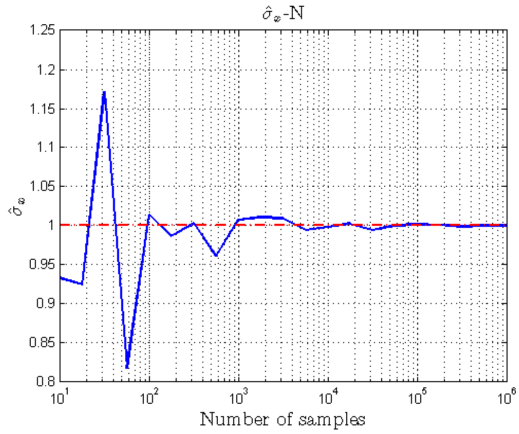


Figure 3: Standard deviation

The estimated mean value $\hat{\mu_x}$ approaches 0 when Nsurpass 10^5 and the estimated deviation $\hat{\sigma_x}$ converges to the value 1 when N grows larger than 10^5. We use the N value to reconstruct the histogram, can it looks really like bell curve.

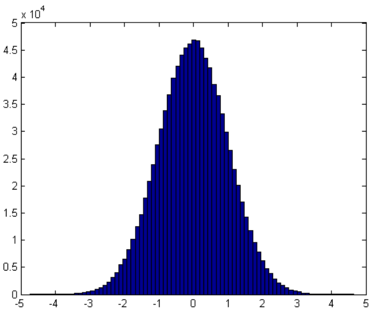


Figure 4: Bell-shaped Curve

To remind that we possess the whole series of data from normal distribution, so I select the formula of Standard Deviation rather than Sample Deviation. Even when N goes extremely large they have no difference, I think there are different statistical pictures in some sense. And in Matlab, the function called $std(x, 1)$ rather than $std(x)$.

$$\sigma_x = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

2.3 Standardizing Normal Distribution

However, we often face the problems with specific deviation and average. In that case, we could do linear transformation by

$$x = \sigma z + \mu$$

So, we rearrange the standard form of distribution to a widely-used form. We can assign specific μ_x and σ_x.

$$f(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-1}{2}} (\frac{x-\mu}{\sigma})^2$$

2.4 Verifying the Result

I assign $\sigma_x = 2$ and $\mu_x = 3$, and confirm these parameters are followed my assignment.

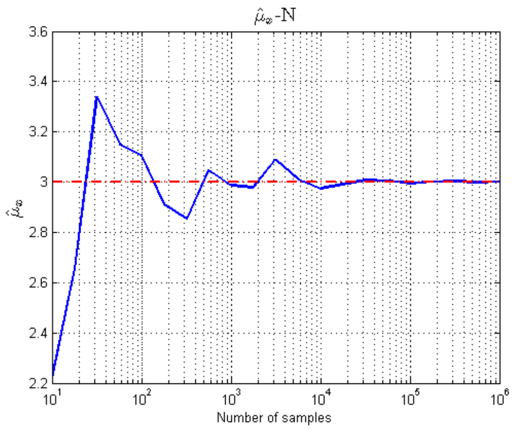


Figure 5: Estimated mean : $\hat{\mu_x} = 3$

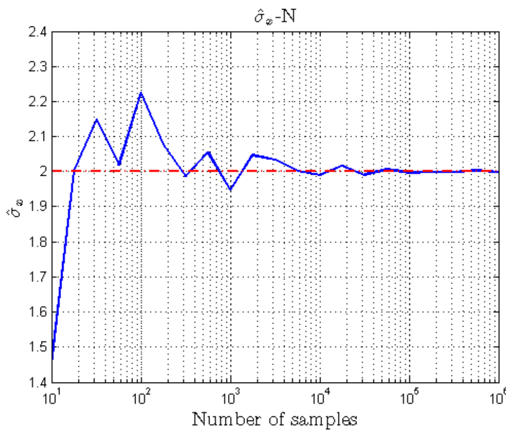
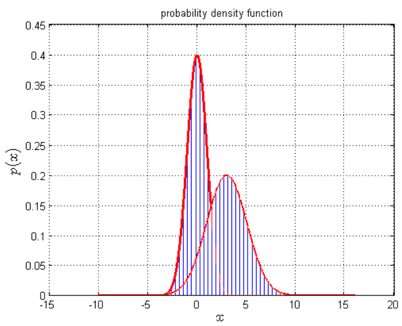


Figure 6: Estimated deviation : $\hat{\sigma_x} = 2$

Clearly, these graphs demonstrate the distribution follow the Gaussian form and mean and deviation are assigned. As the left diagram indicates, the higer one is standard form. When we change μ_x, it shifts right 3 units and enlarge σ_x boarden the distribution.



3 Generate Uniform Sequence

The uniform distribution is determined by two variables which are upper bound and lower bound. As we assign the range, we could generate the distribution. The probibility density function is written as

$$f(x) = \begin{cases} \frac{1}{b-a} & for\ a \le x \le b \\ 0 & otherwise \end{cases}$$

In the standard form, a is 0 and b is 1.

3.1 Create Sequence

Apply the Matlab function called

$$r = rand(n)$$

That we generate uniform distributed random numbers between 0 and 1 Two

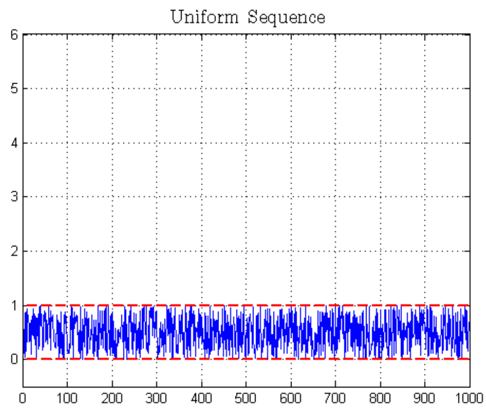


Figure 7: Uniform Sequences are confined between 0 and 1

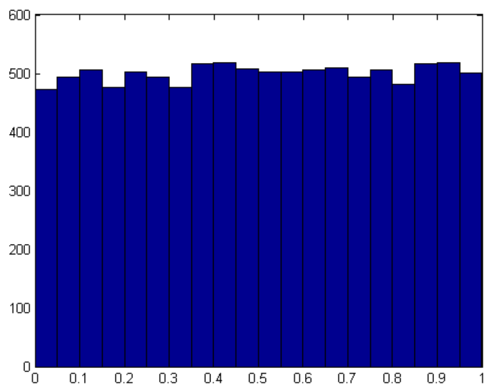


Figure 8: Histogram

graphs show that numbers distributed between 0 and 1 and there are no values outside the range.

3.2 Standardizing Uniform Distribution

When we need different range of uniform distribution, simple linear transformation can do use a fever. We set standard uniform distribution $U(0, 1)$ and the specific range is $[a, b]$. The distribution is $U(a, b)$.
First we shift $U(1, 0)$ to the new lower bound and then scale the range. Then we can get

$$U(a, b) = a + (b - a) \times U(1, 0)$$

3.3 Verifying the Result

We set the range between 1 and 5 which means $a = 1$ and $b = 5$.

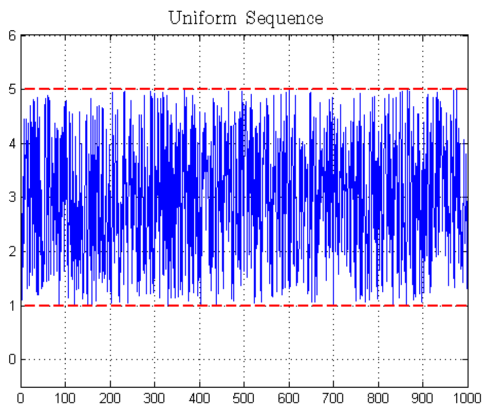


Figure 9: Uniform Sequences are confined between 1 and 5

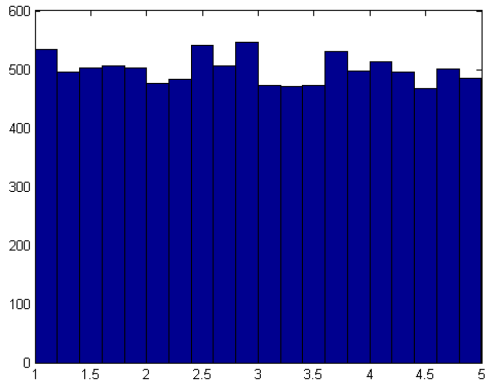


Figure 10: Histogram

It is quite easy to confirm the result by visualizing the time sequence. Obviously, the band of dsitributed numbers boraden and the range becomes $[1, 5]$ due to our linear transformation. The histogram still keeps uniform and each numbers in the range appear equalvalently.

4 Discussion

4.1 Generating Gaussian Distribution by PRNG

It's a much more tough and practical problem. Because the uniform probability density function is determined by a single variable. When we assign two variables(μ_x and σ_x), we would reshape the curve to bell-like, so called normal distribution. On the other hand, it is a real problem. Because our pseudo random number algorithm can only produce uniform, transformation from uniform sequence to Gaussian sequence is necessary.

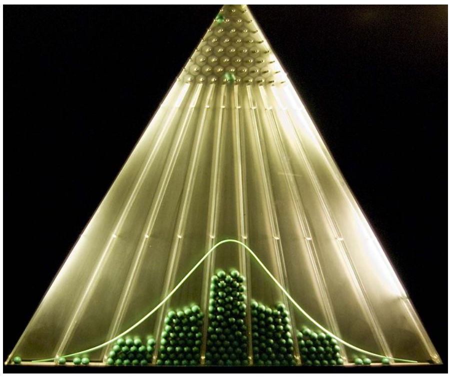
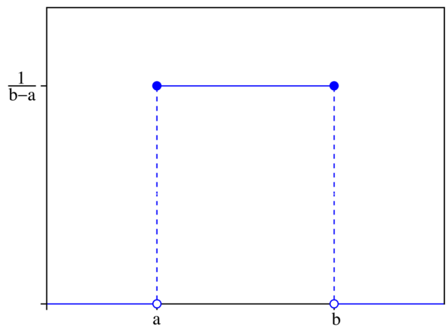


Figure 11: Convert uniform random variables to Gaussian Distribution

4.2 Generate sequence

4.2.1 Pseudo random number generator

The probibility density function of numbers generated by PRNG is a constant on theory. When N is large, the histogram tells us it could be right. We got the result from second assignment.



4.2.2 Distribution Transformation

There are several methods to convert random number to Gaussian sequence. We first convert variables of Gaussian distribution by shfting and scaling.

$$z = \frac{x - \mu}{\sigma}$$

So the Gaussian prabability densty function can be represented as single variable form

$$f(z) = \frac{1}{\sqrt{2\pi}} e^{\frac{-1}{2} z^2}$$

Intuitively, as we generate a set of pseudo random numbers z_0 and apply the formula, we could get an normal distributed sequence. There are kinds of methods can boost the computation efficiency and I choose **BoxMuller transform** to implement my program. It is a neat and fabulous formula.

4.2.3 Algorithm

The idea comes from *Laplace*. We can say the cumulative density function is I

$$I^2 = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-(z_1^2 + z_2^2)/2} dz_1 dz_2$$

Map it to polar coordinates.

$$I^2 = \int_0^{2\pi} \int_0^{\infty} r e^{-r^2/2} d\theta dr$$

I can not derive this rigorously , in fact, I can't do further more. But we can see the strong connection between BoxMuller transform and these formula.

$$z_1 = \sqrt{-\ln(u_1)} \cos(2\pi u_2) \tag{1}$$

$$z_2 = \sqrt{-\ln(u_2)} \sin(2\pi u_1) \tag{2}$$

Where u_1 and u_2 are independent random variables that are uniformly distributed in the interval $(0, 1]$and z_1,z_2 are independent random variables with a standard normal distribution. Then we do variables subsitution.

$$x = \mu + \sigma z$$

4.3 Verifiying the distribution

I use the formula and write the very teeny-tiny code. It is naive and less computational effieiecny but it works when we just do simulation.

BoxMuller transform

```
function r = gaussian_dist( mu, sigma , n )
    z = sqrt( -2*log(rand(n,1) ) ).*cos(2*pi*rand(n,1) ) ;
    r = sigma*z + mu ;
end
}
```

I set mean value $\mu_x = 0$ and deviation $\sigma_x = 1$.

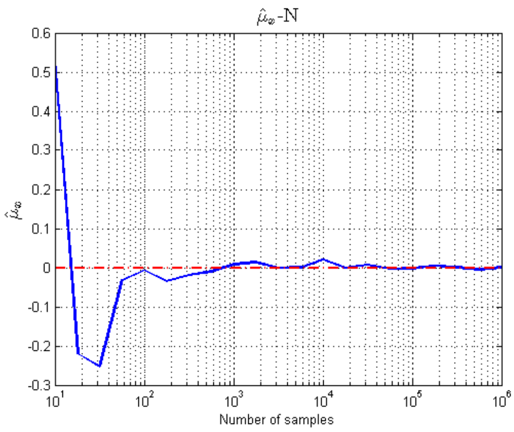


Figure 12: Mean value

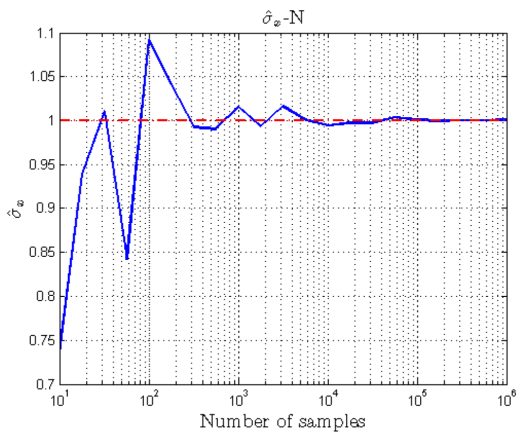


Figure 13: Standard deviation

The result looks good because it converges to assigned vaules when N goes bigger. And the histogram indicates the normal distribution. I could conclude that the algorithm truely generate the desired result.

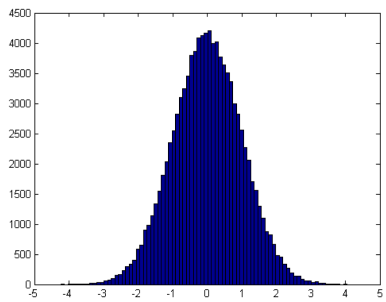


Figure 14: Bell-shaped curve

4.4 Normal Distribution Generator Algorithm

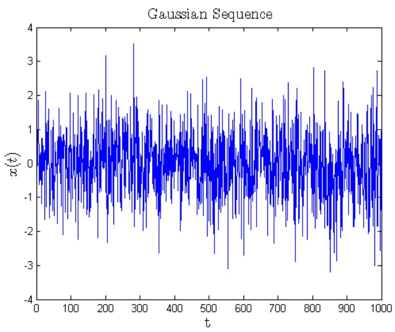
There are many algorithms are used to generate Gaussian distribution. I list two of them.

- BoxMuller transform
- Marsaglia polar method

I implmented BoxMuller transform which is proposed in 1958. Then, Marsaglia polar method was born in 1964, which is noted on the widly-known books in computer science *The Art of Computer Programming*, written by **Donald Knuth** . Nowadays, the algorithm we use to generate distribution is **Ziggurat algorithm**.

4.5 Gaussian White Process

White process is a focus in the course. A useful tool to model the process is Gaussian white process. They are used to simulate the real-world situations. These models are used so frequently that the term additive white Gaussian noise has a standard abbreviation: AWGN.



References

[1] MATLAB©: Users guide. Massachusetts: The Math Works Inc., 2009.

[2] Wikipedia : Ziggurat algorithm

[3] Wikipedia : BoxMuller transform

[4] Wikipedia : Marsaglia polar method