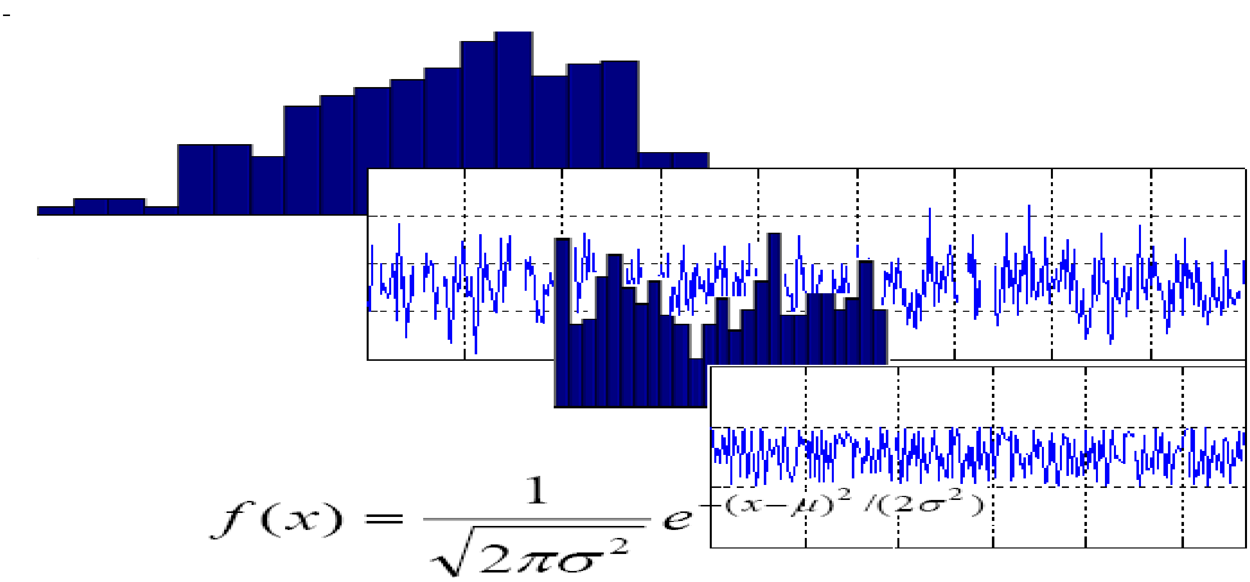


5 სტატისტიკური გაზომვები

შესავალი

ექსპერიმენტის ჩატარებისას შეგროვილ მონაცემთა ანალიზი მნიშვნელოვანი პუნქტია საინჟინრო ექსპერიმენტის შესაფასებლად. ანალიზი მოიცავს როგორც მარტივ გამოთვლებს აგრეთვე მონაცემთა საშუალო მნიშვნელობის, სტანდარტული გადახრისა და დისპერსიის გამოთვლას. ეს სიდიდეები სტატისტიკური სიდიდეებია, რადგან მათ აქვთ სტატისტიკური თვისებები. მაგალითად $\sin 60^0$ ზუსტი რიცხვია. ის არ შეიცვლება, რამდენჯერაც არ უნდა გამოვითვალოთ, მაგრამ მანძილი, რომელსაც მანქანა გაივლის 1 გალონი საწვავის გამოყენებით, სტატისტიკური სიდიდეა, რადგან იგი გამოკიდებულია სხვადასხვა პარამეტრებზე, როგორიცაა ტემპერატურა, სიჩქარე, რომელსაც მანქანა ავითარებს, გზის ტიპი, რომელზეც მოძრაობს. შეგვიძლია შევაფასოთ სტატისტიკური მონაცემების მახასიათებლები, ასევე შეგვიძლია კომპიუტერის საშუალებით ვაწარმოოთ რიცხვითი მწკრივები (შემთხვევითი სიდიდეები) განსაზღვრული მახასიათებლებით. ამ თავში ვნახავთ, როგორ გამოვიყენოთ მონაცემთა ანალიზის ფუნქციები MATLAB–ში და როგორ შევქმნათ შემთხვევით სიდიდეთა განაწილება, რომელსაც ექნება განსაზღვრული სტატისტიკური მახასიათებლები.

1.1 ფუნქციები მონაცემთა ანალიზისათვის

MATLAB შეიცავს ფუცქციათა რიგს მონაცემთა სტატისტიკური ანალიზისათვის. ჯერ განვიხილოთ რამდენიმე მარტივი ფუნქცია.

ფუნქციათა შემდეგი ჯგუფი ხშირად გამოიყენება მონაცემთა ანალიზისათვის.

4.1.1 მაქსიმუმი და მინიმუმი

მონაცემთა ისეთი პარამეტრების შესაფაებლად, როგორიცაა მაქსიმუმი და მინიმუმი, გვაქვს ფუნქციები:

max (x)	თუ x ვექტორია, ეს ფუნქცია გვაძლევს მის ელემენტებს შორის უდიდესის მნიშვნელობას, თუ მატრიცაა, მაშინ – სტრიქონ – ვექტორს, რომლის ელემენტებიც თითოეული სვეტის ელემენტთა შორის უდიდესის ტოლია
max (x,y)	გვაძლევს იგივე ზომის მატრიცას, როგორიცაა x და y, რომლის ყოველი ელემენტი x და y შესაბამის ელემენტებს შორის უდიდესის ტოლია
[y,I] = max (x)	ასეთი ფორმით ბრძანება აგბს y ვექტორს x მატრიცის ყოველი სვეტის მაქსიმუმის მნიშვნელობებით, ხოლო ამ ელემენტების შესაბამის ინდექსებს გვაძლევს i ვექტორის სახით. თუ ორო მაქსიმუმი ერთმანეთს დაემთხვა, პირველი მნიშვნელობის ინდექსს გვაძლევს
min (x)	თუ x ვექტორია, ეს ფუნქცია გვაძლევს x უმცირეს მნიშვნელობას, თუ x მატრიცაა მაშინ – სტრიქონ – ვექტორს, რომლის ელემენტებიც თითოეული სვეტის ელემენტთა შორის უმცირესის ტოლია
min (x, y)	გვაძლევს იგივე ზომის მატრიცას, როგორიცაა x და y, რომლის ყოველი ელემენტი x და y შესაბამის ელემენტებს შორის უმცირესის ტოლია
[y, I] = min (x)	ასეთი ფორმით ბრძანება აგბს y ვექტორს x მატრიცის ყოველი სვეტის მინიმუმის მნიშვნელობებით, ხოლო ამ ელემენტების შესაბამის ინდექსებს გვაძლევს i ვექტორის სახით. თუ ორო მინიმუმი ერთმანეთს დაემთხვა, პირველი მნიშვნელობის ინდექსს გვაძლევს

საშუალო და მედიანა. რიცხვით სიდიდეთა ჯგუფის საშუალო მათი არითმეტიკული საშუალოა. საშუალოს აღნიშნავენ ბერძნული სიმბოლოთი μ(მიუ)

$$\mu = \frac{\sum_{k=1}^{N} x_k}{N}$$

სადაც

$$\sum_{k=1}^{N} x_k = x_1 + x_2 + x_3 + \cdots + x_n$$

მედიანა არის ჯგუფის საშუალედო მნიშვნელობა, თუ ჯგფის წევრებს დავალაგებთ ზრდის მიხედვით. თუ x ვექრორი შეიცავს ეელემენტთა კენტ რაოდენობას N, ჯერ დავალაგებთ x ელემენტებს ზრდის მიხედვით, მედიანა იქნება იმ ელემენტის ტოლი, რომლის ინდექსია **ceil**(N/2) (თუ N=5, მედიანა იქნება მესამე ელემენტის ტოლი), თუ N ლუწია, მაშინ მედიანა იქნება (სრდის მიხედვით დალაგების შემდეგ) შუა ორი ელემენტის საშუალო მნიშვნელობა.

MATLAB საშუალოს და მედიანას გამოითვლის ფუნქციით:

mean (x)	თუ x ვექტორია, ეს ფუნქცია გამოითვლის x ელემენტების საშუალო მნიშვნელობას. თუ x მატრიცაა, ეს ფუნქცია გვაძლევს ვექტორს, რომლის ელემენტებია შესაბამისი სვეტების ელემენტთა საშუალო მნიშვნელობა
median (x)	თუ x ვექტორია, ეს ფუნქცია გამოითვლის x ელემენტების მედიანას. თუ x მატრიცაა, ეს ფუნქცია გვაძლევს ვექტორს, რომლის ელემენტებია შესაბამისი სვეტების ელემენტთა მედიანა

4.1.2 ჯამი და ნამრავლი

MATLAB აქვს ფუნქციები, რომლებიც გამოითვლის მატრიცის ელემენტების ჯამს და ნამრავლს და აგრეათვე ისეთი ფუნქციები, რომებიც იძლევა მატრიცის ელემენტების კუმულაციურ ჯამსა და ნამრავლს

sum (x)	თუ x ვექტორია, ეს ფუნქცია გამოითვლის x ელემენტების ჯამს. თუ x მატრიცაა, ეს ფუნქცია გვაძლევს ვექტორს, რომლის ელემენტებია შესაბამისი სვეტების ელემენტთა ჯამს
prod (x)	თუ x ვექტორია, ეს ფუნქცია გამოითვლის x ელემენტების ნამრავლს. თუ x მატრიცაა, ეს ფუნქცია გვაძლევს ვექტორს, რომლის ელემენტებია შესაბამისი სვეტების ელემენტთა ნამრავლი
cumsum (x)	თუ x ვექტორია, ეს ფუნქცია გვაძლევს იგივე სიგრძის ვექტორს, რომლის ელემენტები x ელემენტების კუმულაციურ ჯამს წარმოადგენ, თუ x მატრიცაა, ეს ფუნქცია იძლევა იგივე ზომის მატრიცას, რომლის ელემენტებიც კუმულაციურ ჯამს წაროადგენს სვეტების მიმართ
cumprod (x)	თუ x ვექტორია, ეს ფუნქცია გვაძლევს იგივე სიგრძის ვექტორს, რომლის ელემენტები x ელემენტების კუმულაციურ ნამრავლს წარმოადგენ, თუ x მატრიცაა, ეს ფუნქცია იძლევა იგივე ზომის მატრიცას, რომლის ელემენტებიც კუმულაციურ ნამრავლს წაროადგენს სვეტების მიმართ

sort (x)	თუ x ვექტორია, დაალაგებს ელემენტებს ზრდის მიხედვით. თუ x მატრიცაა, ეს ფუნქცია თითოეულ სვეტს დაალაგებს ზრდის მიხედვით
[y, i]=sort(x)	ასეთი ფორმით ეს ფუნქცია y ვექტორის სახით მოგვცემს x ვექტორის მნიშვნელობებს დალაგებულს ზრდის მიხდვით ხოლო i ვექტორში მოგვცემს შესაბამისი ინდექსების მნიშვნელობას.

სავარჯიშო

მოცემულია მატრიცები:

w = [0 3 –2 7];
x = [3 –1 5 7];

$$y = \begin{bmatrix} 1 & 3 & 7 \\ 2 & 8 & 4 \\ 6 & -1 & -2 \end{bmatrix}$$

გამოითვალეთ შემდეგი სიდიდეები და შეამოწმეთ MATLAB საშუალებით

1. max (w)
2. min (y)
3. min (w,x)
4. [z, i] = max (y)
5. mean (y)
6. median (w)
7. cumprod (y)
8. sum (x)
9. sort (2*w + x)
10. sort (y)

4.1.3 დისპერსია და სტანდარტული გადახრა

მონაცემთა ჯგუფისათვის ერთ-ერთი მნიშვნელოვანი სტატისტიკური მახასიათებელია დისპერსია. ვიდრე განვიხილავდეთ დისპერსიის მათემათიკურ შინაარსს, განვიხილოთ ასეთი მაგალითი: გვაქვს მონაცემთა ორი ჯგუფს data1 და data2 (ნახ. 5.1) ნახაზი აგებულია subplot ბრძანებით, რომელსაც მოგვიანებით გავეცნობით. თუ შევეცდებით გავატაროთ მრუდი საშუალო მნიშვნელობებზე, ეს მრუდი ორივე მათგანისთვის შეესაბამება მნიშვნელობას 3.0. ამიტომ შეგვიძლია ვივარაუდოთ, რომ ორივე ჯგუფის საშუალო მნიშვნელობა 3.0 ტოლია, თუმცა მონაცემებს განსხვავებული მახასიათებლები გააჩნიათ. data2 მონაცემები უფრო მკვეთრად გადაიხრება საშუალო მნიშვნელობიდან. ე. ი. ცვლილების, გადახრის

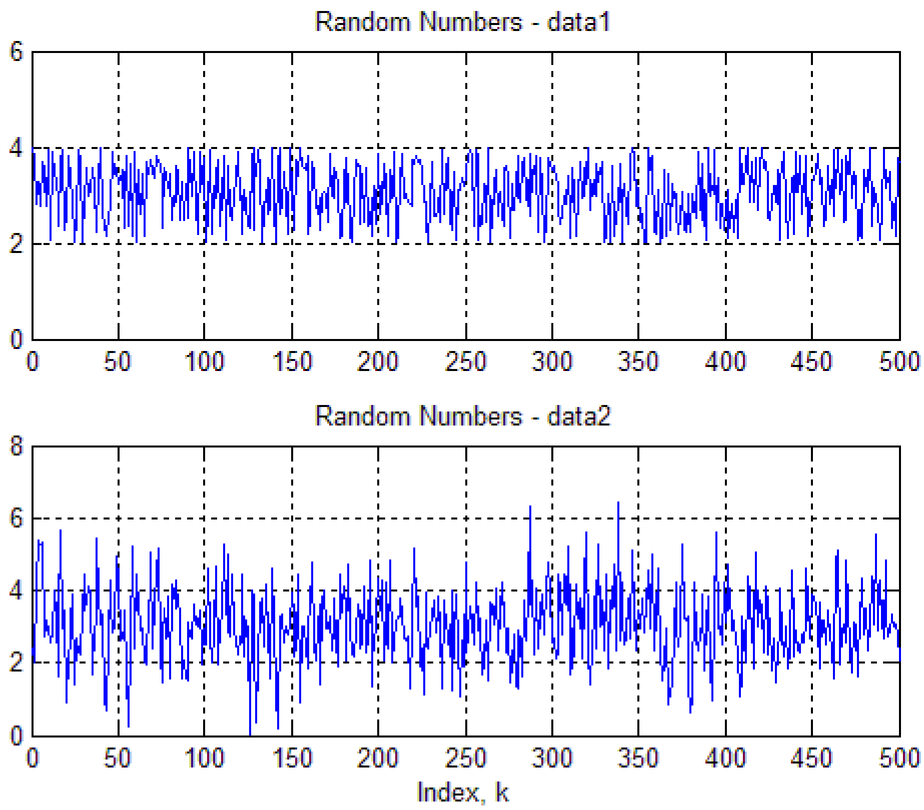
მნიშვნელობები data2 უფრო დიდია, ვიდრე data1 –თვის. რაც უფრო დიდია გადახრა, უფრო მეტად ფლუქტუირებენ მონაცემები საშუალო მნიშვნელობის მიმართ.

მათემატიკურად დისპერსიას აღნიშნავენ ბერძნული სიმბოლოთი σ^2(სიგმა), მონაცემთა მწკრივის (რომელიც x ვექტორის ელემენტებს წარმოადგენს) დისპერსია გამოითვლება შემდეგი ფორმულით:

$$\sigma^2 = \frac{\sum_{k=1}^{N}(x_k - \mu)^2}{N-1}$$

თუ კარგად დააკვირდებით გამოსახულება საკმაოდ მარტივია. $x_k - \mu$ წარმოადგენს ვექტორის k ური ელემენტის გადახრას საშუალოდან, ეს მნიშვნელობა კვადრატშია აყვანილი, რომ ჯამის ყველა შესაკრები დადებითი სიდიდე იყოს, შემდეგ აღებულია ჯამი საშუალოდან თითოეული ელემენტის გადახრისა. საბოლოოდ მიღებული ჯამი გაყოფილი N-1 – ელემენტთა რაოდენობაზე. ეს განტოლება გამოითვლის დისპერსიას. კვადრატული ფესვის მნიშვნელობას დისპერსიიდან, უწოდებენ საშუალო კვადრატულ ან სტანდარტულ გადახრას:

$$\sigma = \sqrt{\sigma^2}$$



MATLAB აქვს ფუნქცია სტანდარტული გადახრის გამოსათვლელად. თუ დისპერსია გვაინტერესებს, უბრალოდ, სტანდარტულ გადახრას კვადრატში ავიყვანთ.

std (x)	თუ x ვექტორია, ეს ფუნქცია გამოითვლის x მნიშვნელობათა სტანდარტულ გადახრას. თუ x მატრიცაა, მოგვცემს

სტრიქონ ვექტორს, რომლის ელემენტებია შესაბამისი სვეტის ელემენტთა სტანდარტული გადახრაა.

სავარჯიშო

განსაზღვრეთ შემდეგ ფუნფციათა მნიშვნელობები, თუ მოცემულია მატრიცები:

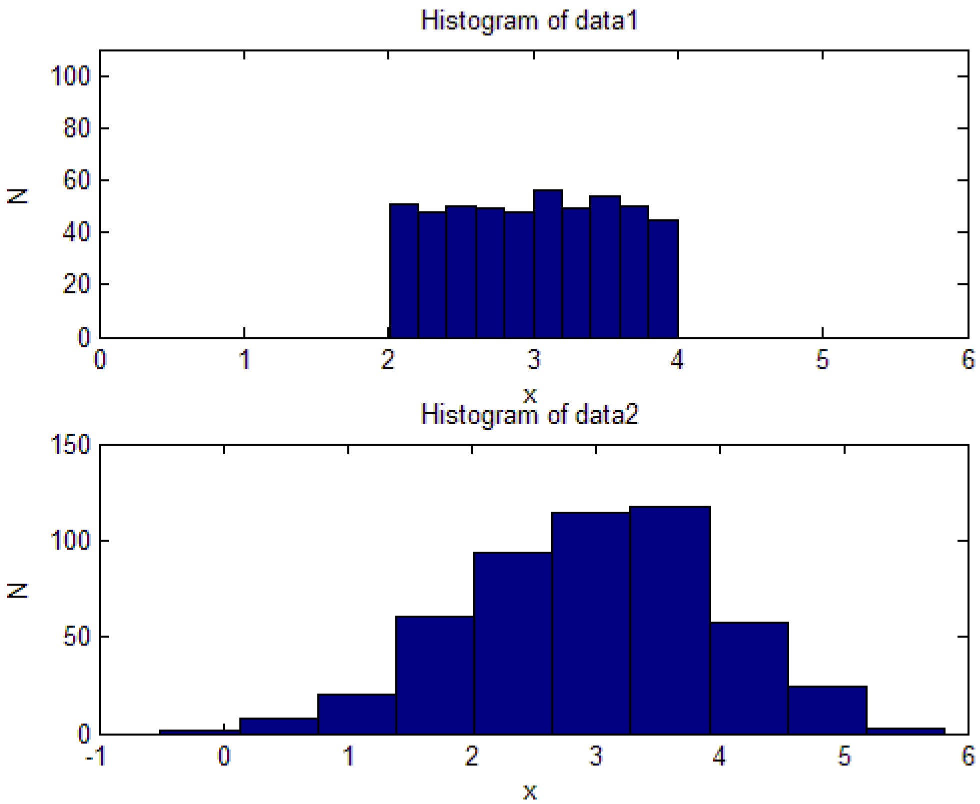
w = [0 3 –2 7];
x = [3 –1 5 7];

$$y = \begin{bmatrix} 1 & 3 & 7 \\ 2 & 8 & 4 \\ 6 & -1 & -2 \end{bmatrix}$$

1. std (w)
2. std (x)^2
3. std (y(:2))
4. std (y)
5. std (y).^2

4.1.4 ჰისტოგრამა

MATLAB –ის გრაფიკული შესაძლებლობები განხილული იქნება მე-7 თავში. აქ განვიხილავთ ჰისტოგრამას, რომელიც სპეციალური გრაფიკული საშუალებაა სტატისტიკური მონაცემების ვიზუალიზციისათვის. ჰისტოგრამა გვიჩვენებს მონაცემთა განაწილების სურათს. MATLAB –ში ჰისტოგრამა გამოითვლის იმ სიდიდეთა რაოდენობას, რომელნიც ხვდებიან 10 ზოლში, როლებადაც დაყოფილია მონაცემთა უმცირეს და უდიდეს მნიშვნელობას შორის ინტერვალი. მაგალოთად, თუ ავაგებთ data1 და data2 მონაცემების ჰისტოგრამას, მივიღებთ ნახ, 5.2.



ნახ. 5.2   ჰისტოგრამა 10 სვეტით

შევნიშნავთ, რომ ჰისტოგრამა განსხვავებულ ინფორმაციას გვაძლევს. ჰისტოგრამა გვაძლევს არა მარტო მნიშვნელობათა არეს, არამედ ინფორმაციას იმის თაობაზე, თუ როგორ არის ეს მონაცემები განაწილებული. მაგალითად data1 მონაცემები უჩვენებს რომ მონაცემები თითქმის თანაბრადაა განაწილებული უმცირეს (2) და უდიდეს (4) სიდიდეს შორის (ასეთი ტიპის განაწილებას თანაბარი ეწოდება). data2 მონაცემები არ არის განაწილებული თანაბრად, მონაცემთა უმრავლესობა კონცენტრირებულია საშუალო მნიშვნელობის მახლობლად (ასეთი ტიპის განაწილებას გაუსისებური, ან ნორმალური განაწილება ეწოდება).

MATLAB ბრძანება ჰისტოგრამის ასაგებად ამგვარია:

hist (x)

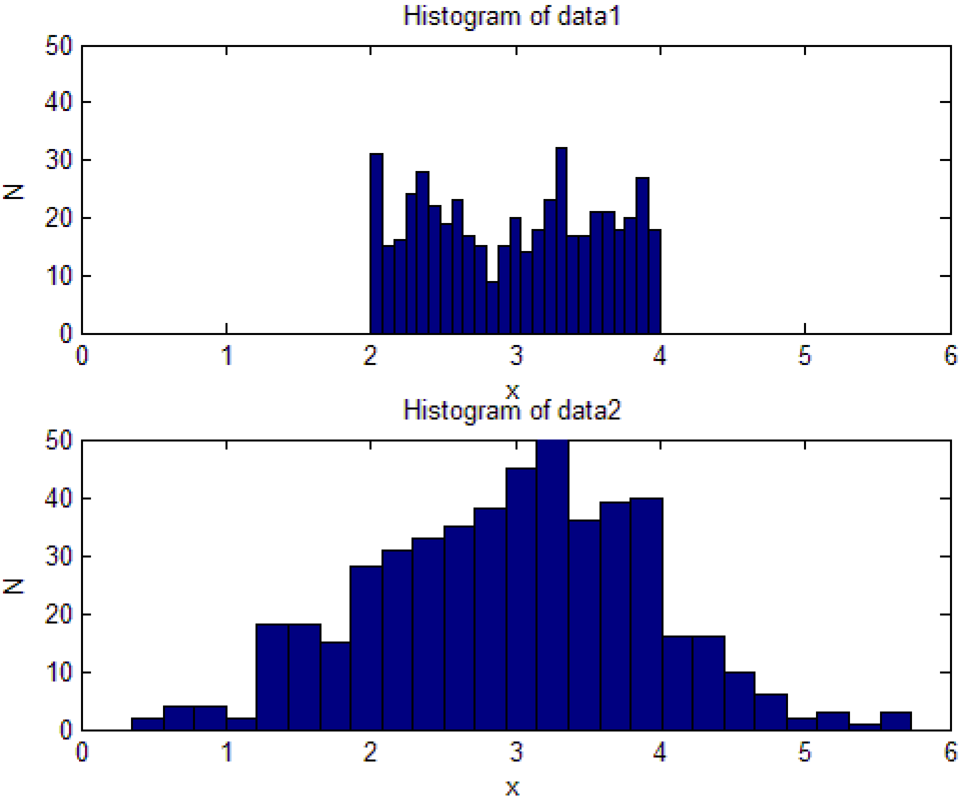
სადაც x ვექტორია იმ მონაცემებით, რომლის ჰისტოგრამაც უნდა ავაგოთ. ეს ბრძანება პირველ რიგში დაალაგებს x ვექტორის ელემენტებს ზრდის მიხედვით, შემდეგ ინტერვალს x-ის მინიმალურ და მაქსიმალურ მნიშვნელობებს შორის გაყოფს 10 ტოლ ნაწილად და დაითვლის თითოეულ სექტორში რამდენი მონაცემი მოხვდა.

თუ მოცემული გვაქვს მონაცემები მატრიცის სახით და გვსურს ავაგოთ მისი მეორე სვეტის მნიშვნელობათა ჰისტოგრამა, ასე უნდა მივუთითოთ:

hist (data (: , 2))

ბრძანება **hist** ასევე საშუალებას იძლევა შევარჩიოთ ჰისტოგრამის სვეტების რაოდენობა. თუ გვსურს გავზარდოთ ჰისტოგრამის გარჩევა ისე, რომ იგი შეიცავდეს 25 სვეტს ნაცვლად ათისა, მივმართავთ შემდეგ ბრძანებას:

hist (x, 25)



ნახ. 5.3 ჰისტოგრამა 25 სვეტით

შესაბამისი ჰისტოგრამა ნაჩვენებია ნახ. 5.3. (5.2 და 5.3 ნახაზები აგებულია MATLAB ბრძანებებით **hist** და **subplot,** რომელთაც მოგვიანებით შევისწავლით).
ინფორმაცია, რომლის მიხედვითაც ჰისტოგრამა აიგება, შეგვიძლია ვექტორის სახითაც ჩავწეროთ:

```
[n, x] = hist (data1);
[n, x] = hist (data1,25);
```

ეს ბრძანებები ჰისტოგრამას კი არ ააგებენ, არამედ შექმნიან ორ ვექტორს – n და x. n ვექტორი შეიცავს დათვლებს (თითოეულ სვეტში მოხვედრილი მონაცემების რაოდენობა) 10 სვეტისათვის, x ვექტორი კი - თითოეული სვეტის შუაწერტილის მნიშვნელობას. მეორე ბრძანება იგივეა, რაც პირველი, მაგრამ 25 სვეტისათვის.
ეს ვექტორები დაგვჭირდება **bar** გრაფიკების ასაგებად, რომელსაც მე-7 თავში განვიხილავთ.

პრობლემა: საკომუნიკაციო სიგნალი

დავუშვათ გვინდა შევქმნათ სისტემა, რომელიც გამოიცნობს წარმოთქმულ სიტყვას შემდეგი ათი სიტყვიდან: “ნული”, “ერთი”, “ორი”, “სამი”. . . “ცხრა”, რომლებიც ციფრული სიგნალის სახითაა წარმოდგენილი. პირველი, რაც უნდა გავაკეთოთ არის გავაანალოზოთ შესაბამისი სიგნალის მნიშვნელობები. ვნახით, არსებობს თუ არა რაიმე სტატისტიკური შეფასება, რომელიც მათ გამორჩევაში დაგვეხმარება. მონაცემთა ანალიზის MATLAB ფუნქციები მარტივად გამოითვლის სტატისტიკურ მახასიათებლებს,

შემდეგ შეგვიძლია დავბეჭდოთ მათი მნიშვნელობები, მოვძებნოთ ის პარამეტრები, რომლებიც დასმული ამოცანის გადაწყვეტაში დაგვეხმარება. მაგალითად შესაძლოა ერთი შედეგის საშუალებით 10 დან 3 მონაცემი გამოვარჩიოთ, შემდეგ კი მეორე შეფასების საშუალებით ამ სამს შორის ჩვენთვის საჭირო სიგნალი გამოვარჩიოთ.

ვისარგებლოთ რეალური სიგნალით, გამოვთვალოთ სტატისტიკური მახასიათებლები, რომლებიც შემდგომში გამოიყენება სიგნალის გამოცნობის სრულ ალგორითმში. თითოეული სიგნალის შესაბამისი მონაცემთა ფაილი შეიცავს 1000 სიდიდეს. დაწერეთ MATLAB პროგრამა, რომელიც წაიკითხავს ASCII მონაცემთა ფაილს ‘‘nuli.dat’’ და გამოითვლის შემდეგ ინფორმაციას: საშუალო, სტანდარტული გადახრა, დისპერსია, საშუალო სიმძლავრე, საშუალო ამპლიტუდა და ნულოვანი გადაკვეთა.

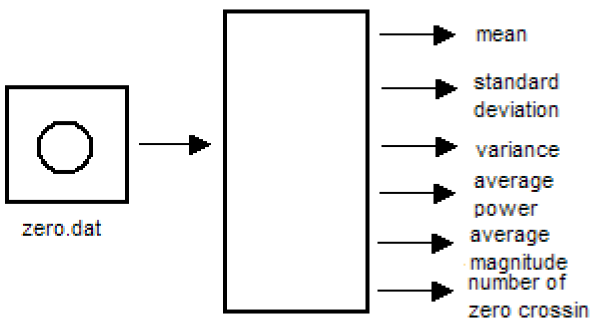
ზემოთ უკვე განვიხილეთ საშუალო, სტანდარტული გადახრა და დისპერსია. საშუალო სიმძლავრე არის საშუალო კვადრატული სიდიდე და უფრო დაწვრილებით შემდეგში იქნება განხილული. ნულოვანი გადაკვეთა გვიჩვენებს თუ მონაცემთა მწკრივში რამდენჯერ შეიცვალა უარყოფითი სიდიდე დადებითით და პირიქით.

1. ამოცანის დასმა

გამოითვალეთ საკომუნიკაციო სიგნალის სტატისტიკური მახასიათებლები: საშუალო, სტანდარტული გადახრა, დისპერსია, საშუალო სიმძლავრე, საშუალო ამპლიტუდა და ნულოვანი გადაკვეთის წერტილების რაოდენობა.

2. INPUT/OUTPUT აღწერა

ნახ 5.4 წარმოადგენს დიაგრამას. საწყისი მნიშვნელობა ეს არის ფაილი ხმოვანი სიგნალით, საბოლოო მნიშვნელობა კი ფაილის მონაცემთა სტატიტიკური მახასიათებლები.



ნახ. 5.4 INPUT/OUTPUT დიაგრამა

3. სახელდახელო ამოხსნა

დავუშვათ ფაილი შეიცავს შემდეგ მონაცემებს:

[2.5 8.2 –1.1 –0.2 1.5]

კალკულატორის საშუალებით შეგვიძლია გამოვითვალოთ:

საშუალო = $(2.5 + 8.2 + -1.1 + -0.2 + 1.5)/5 = 2.18$
დისპერსია = $[(2.5 - \mu)^2 + (8.2 - \mu)^2 + (-1.1 - \mu)^2 + (-0.2 - \mu)^2 + (1.5 - \mu)^2]/4$ $=13.307$

სტანდარტული გადახრა = $\sqrt{13.307} = 3.648$

საშუალო სიმძლავრე = $(2.5^2 + 8.2^2 + (-1.1)^2 + (-0.2)^2 + 1.5^2)/5$
საშუალო ამპლიტუდა = $(|2.5| + |8.2| + |-1.1| + |-0.2| + |1.5|)/5 = 2.7$
ნულოვანი გადაკვეთის წერტილები = 2

4. MATLAB ამოხსნა

```
%
%    This program komputes a number of statistics
%    for an utterance stored in a data file
%
load one.dat;
x = one;
fprintf ('Digit Statistics \n\n')
fprintf ('mean: %f \n', mean (x))
fprintf ('standard deviation: %f \n', std(x))
fprintf ('variance: %f \n', std(x)^2)
fprintf ('average power: %f \n', mean(x.^2))
fprintf ('average magnitude: %f \n', mean(abs(x)))
crossings = 0;
for j = 1:length(x)-1
    if x(j)*x(j+1) < 0
        crossings =crossings + 1;
    end
end
fprintf ('zero crossings: %f \n', crossings)
```

5. შემოწმება

პროგრამა გავუშვით MATLAB–ში და მონაცემთა ფაილისათვის “one,dat” გამოვითვალეთ შემდეგი სტატისტიკური მნიშნმელობები:

```
Digit Statistics

mean: 0.000494
standard deviation: 0.218396
variance: 0.047697
average power: 0.047696
average magnitude: 0.099959
zero crossings: 424.000000
```

1.2 შემთხვევითი რიცხვები

შემთხვევითი რიცხვები არ მოიცემა განტოლებით. მათი მნიშვნელობა შემთხვევაზეა დამოკიდებული. წინასწარ შეუძლებელია იმის განსაზღვრა, თუ რა მნიშვნელობას მიიღებს შემთხვევითი სიდიდე. ის რიცხვითი მნიშვნელობა, რომელიც შეუძლია მიიღოს შემთხვევილ სიდიდეს, არის მისი შესაძლო მნიშვნელობა. შემთხვევითი სიდიდის შესასწავლად პირველ რიგში უნდა ვიცოდეთ ის რიცხვითი მნიშვნელობები, რომელთა მიღებაც მას შეუძლია. მაგრამ მარტო შესაძლო რიცხვით მნიშვნელობათა ცოდნა არ არის საკმარისი შემთხვევითი სიდიდის შესასწავლად. საჭიროა აგრეთვე ვიცოდეთ, თუ რა სიხშირით ღებულობს იგი ამა თუ იმ შესაძლო მნიშვნელობას. ამას კი განსაზღვრავს მისი განაწილების კანონი, რომელიც მოიცემა განაწილების ფუნქციით. მრავალ საინჟინრო ამოცანაში გვჭირდება შემთხვევითი სიდიდეების გამოყენება. ზოგ შემთხვევაში ისინი მოდელირებისთვისაა საჭირო. მოდელი შეაძლოა რამდენჯერმე გამოიცადოს და მოხდეს მიღებული შედეგის ანალიზი, ყოველი ხელახალი ტესტირება შეესაბამება ექპერიმენტის განმეორებას შეცვლილი პარამეტრებით. შემთხვევითი რიცხვები გვჭირდება ხმაურის მოდელირებისათვის. მაგალითად რადიოს მოსმენისას ხშირად გვესმის ხმაური, რომელიც სიგნალს ერთვის და ამახინჯებს. შეიძლება ჩვენც გარკვეული ექსპერიმენტის მოდელირებისთვის სიგნალს დავურთოთ ხმაური უფრო რეალური სურათის შესაქმნელად.

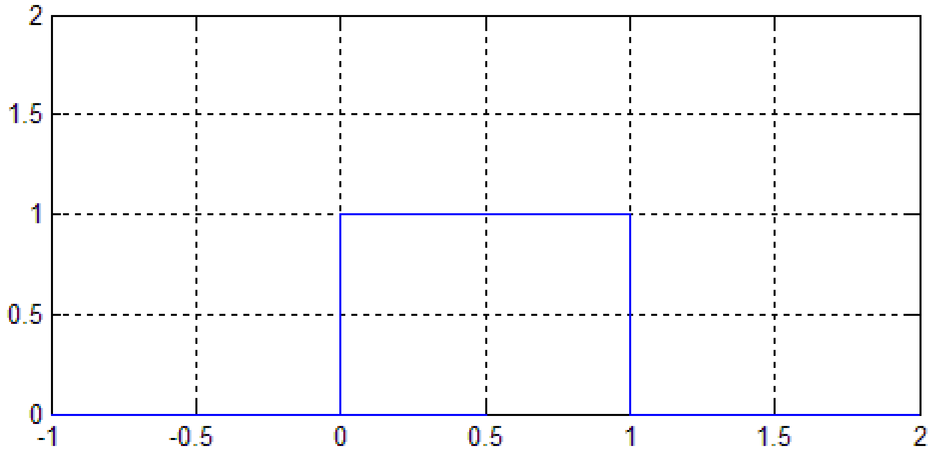
4.2.1 ფუნქციები შემთხვევითი რიცხვებისათვის

ფუნქცია rand MATLAB- ში ქმნის შემთხვევით რიცხვებს ინტერვალში [0, 1], ასე მიღებულ შემთხვევით სიდიდეთა განაწილება თანაბარია. **seed** მნიშვნელობა გამოიყენება ინიციალიზაციისათვის.

rand (n)	ეს ფუნქცია გვაძლევს n სტრიქონიან, n სვეტიან მატრიცას, რომლის ელემენტებიც 0 და 1 შორის მოთავსებული შემთხვევითი რიცხვებია. მიღებული შერჩევის განაწილება თანაბარია
rand (m,n)	გვაძლევს m სტრიქონიან n სვეტიან მატრიცას, რომლის ელემენტებიც 0 და 1 შორის მოთავსებული შემთხვევითი რიცხვებია. მიღებული შერჩევის განაწილება თანაბარია
randn (n)	ეს ფუნქცია გვაძლევს n სტრიქონიან, n სვეტიან მატრიცას, რომლის ელემენტებიც ქმნიან შერჩევას ნორმალური განაწილებით, რომლის საშუალოა 0, ხოლო დისპერსია და სტანდარტული გადახრა 1
randn (m,n)	ეს ფუნქცია გვაძლევს m სტრიქონიან, n სვეტიან მატრიცას, რომლის ელემენტებიც ქმნიან შერჩევას ნორმალური განაწილებით, რომლის საშუალოა 0, ხოლი დისპერსია და სტანდარტული გადახრა 1
rand('seed',s)	განსაზღვრავს seed მნიშვნელობას
rand('seed',0)	დააბრუნებს seed მნიშვნელობას საწყის მდგომარეობაში

4.2.2 შემთხვევით სიდიდეთა თანაბარი განაწილება

შემთხვევითი სიდიდეები ხასიათდება სიმკვრივის ფუნქციით, რომელიც განაწილების ფუნქციის წარმოებულს წარმოადგენს. ეს ფუნქცია ჰგავს ჰისტოგრამას, გვიჩვენებს შემთხვევით სიდიდეთა ინტერვალს და განსაზღვრავს ცალკეული სიდიდეების ხდომილობის ალბათობას. MATLAB ფუნქცია **rand** გვაძლევს შემთხვევით სიდიდეთა ერთობლიობას ისე, რომ თითოეულის მნიშვნელობა თანაბრადალბათურია ინტერვალში [0,1]. მისი სიმკვრივის ფუნქცია მოცემული 5.5 ნახაზზე. სიმკვრივის ფუნქცია უჩვენებს შემთხვევით სიდიდეთა ზედა და ქვედა ზღვარს (0 და 1) x ღერძზე. ფუნქციის (y ღერძი) მართკუთხა ფორმა მიუთითებს, რომ 0 სა და 1 შორის ყველა სიდიდის ხდომილობის ალბათობა ერთნაირია. სიმკვრივის ფუნქციით შემოსაზღვრული ფართი ერთის ტოლია. ამიტომ თუ მისი სიგანე 1-ის ტოლია, სიმაღლეც ერთის ტოლი უნდა იყოს.



ნახ. 5.5 თანაბარი განაწილების სიმკვრივის ფუნქცია

შემდეგი ბრძანებები ქმნის თანაბრად განაწილებული 10 სიდიდის ერთობლიობას, რომელთა მნიშვნელობები 0 და 1 შორისაა:

rand (‘seed’,0) განსაზღვრავს seed საწყის მდგომარეობას
rand(10,1) აწარმოებს 10 სიდიდეს

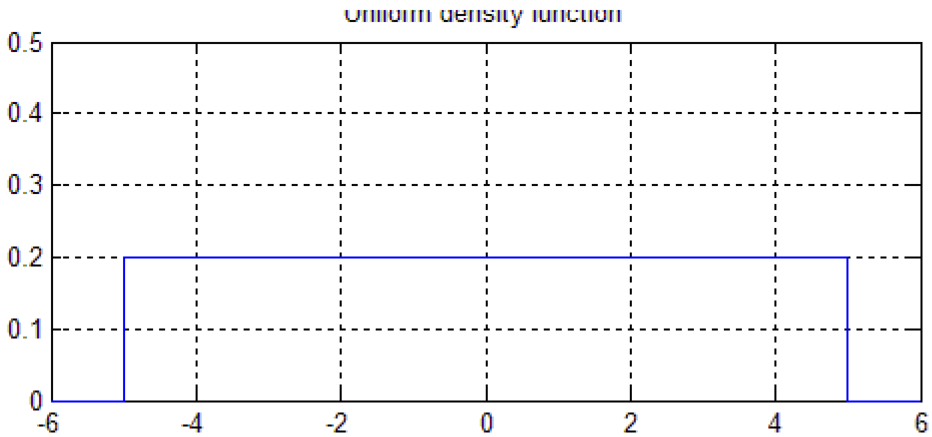
rand ფუნქცია აწარმოებს შემთხვევით სიდიდეთა ერთიდაიგივე მიმდევრობას ერთიდაიგივე ‘seed’ შემთხვევაში. ეს ბრძანება მოგვცემს სვეტ ვექტორს:

0.2190
0.0470
0.6789
0.6793
0.9347
0.3835
0.5194
0.8310
0.0346
0.0535

ხშირად საჭიროა შემთხვევითი სიდიდეები არა მხოლოდ [0 1] ინტერვალში. მაგალითად 5.6 თანაბარი განაწილების ფუნქციაა –5-სა და5 შორის.

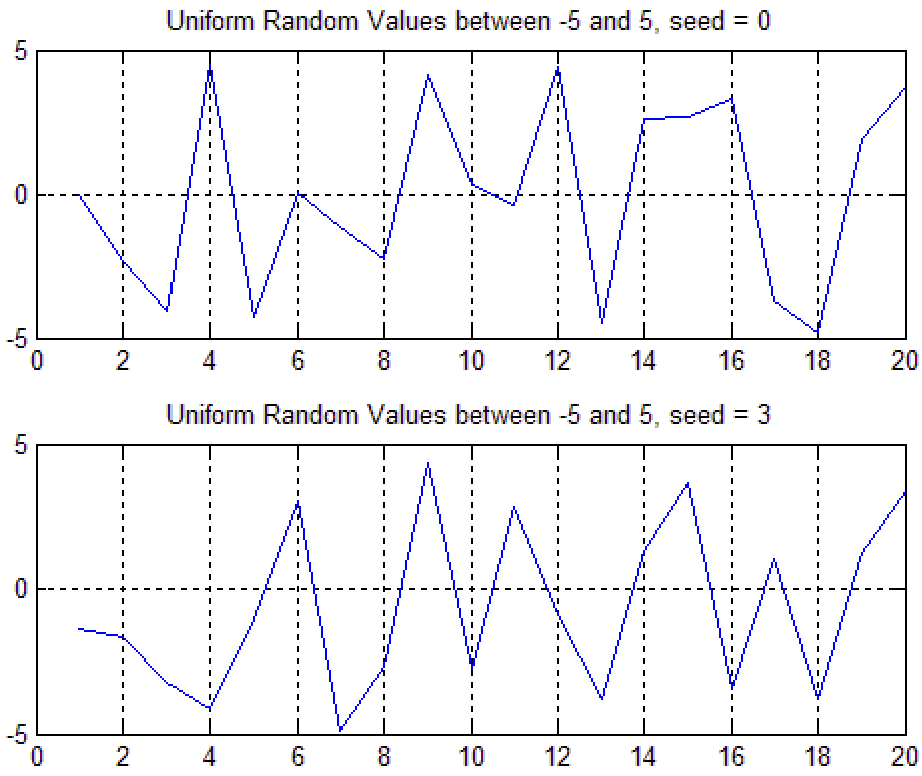
დავუშვათ გვაქვს r შემთხვევითი სიდიდე, რომელიც ვაწარმოეთ თანაბარი განაწილების განერატორით [0 1] შორის. შეგვიძლია ვაწარმოოთ შემთხვევითი სიდიდე, თანაბარი განაწილებით, რომლის მნიშვნელიბაც იქნება სხვა საზღვრებში. პირველ რიგში r გავამრავლებთ სიმკვრივის ფუნქციის სიგანეზე. სიგანე გამოითვლება ზედა და ქვედა საზღვრებს შორის სხვაობით. მიღებულ შედგეს დავუმატებთ ქვედა საზღვრის მნიშვნელობას, რომ მივიღოთ საჭირო ინტერვალი. მაგალითად, დავუშვათ გვსურს მივიღოთ შემთხვევითი სიდიდეთა ერთობლიობა ინტერვალში –5, 5. პირველ რიგში ვაწარმოებთ შემთხვევით სიდიდეებს 0-სა და 1 შორის, შემდეგ მათ გავამრავლებთ 10-ზე (5 – (-5)). შემდეგ დავუმატებთ –5 და მივიღებთ შემთხვევით სიდიდეთა ერთობლიობას, რომელიც განაწილებულია თანაბრად ინტერვალში [–5 5]. მაშასადამე, თუ გვსურს [0 1] ინტერვალში თანაბრად განაწილებული სიდიდეები გადავიყვანოთ ისეთ სიდიდეებში, რომელნიც თანაბრად იქნებიან განაწილებულნი ინტერვალში [a b], უნდა ვისარგებლოთ ფორმულით:

$$x = (b - a) * r + a$$



ნახ. 5.6 სიმკვრივის ფუნქცია თანაბარი განაწილებისათვის ინტერვალში –5 და 5

თუ შემთხვევით სიდიდეთა ორი მიმდევრობა ელემენტთა ერთიდაიგივე რაოდენობით და ერთნაირი საზღვრებით ვაწარმოეთ seed სხვადასხვა მნიშვნელობით, შედეგიც სხვადასხვა იქნება. ზოგჯერ საინჟინრო პროგრამაში შესაძლოა დაგვჭირდეს შემთხვევით სიდიდეთა ზუსტად ერთნაირი მიმდევრობის წარმოება სხვადასხვა შემთხვევაში, ამის საშუალებას მოგვცემს seed ერთიდაიგივე მნიშვნელობით სარგებლობა.



ნახ. 5.7 შემთხვევით სიდიდეთა მიმდევრობა ინტერვალში –5 და 5 **seed** სხვადასხვა მნიშვნელობით

სავარჯიშო

მიეცით MATLAB–ს ბრძანება, რომ შექმნათ შემთხვევით რიცხვთა 10 წევრიანი მიმდევრობა შემდეგი მახასიათებლებით:

1. მიმდევრობა თანაბარი განაწილებით ინტერვალში 0 – 10.0.
2. მიმდევრობა თანაბარი განაწილებით ინტერვალში -1 – 1.
3. მიმდევრობა თანაბარი განაწილებით ინტერვალში -20 – 10.
4. მიმდევრობა თანაბარი განაწილებით ინტერვალში 4.5 – 5.
5. მიმდევრობა თანაბარი განაწილებით ინტერვალში π - -π.

შემთხვევით სიდიდეთა მიმდევრობას აგრეთვე უწოდებენ შემთხვევით სიგნალს. ხშირად გვაინტერესებს სიგნალის საშუალო მნიშვნელობა, რომელიც ადვილად გამოითვლება **mean** ფუნქციის საშუალებით. თეორიულად თანაბარი განაწილების საშუალო მნიშვნელობა ტოლია განაწილების ფუნქციის შუაწრტილისა ანუ;

$$\mu = \frac{upper\_bound + lower\_bound}{2}$$

ასევე საინტერესოა სიგნალის დისპერსიის გამოთვლა. ეს დიდიდე შეიძლება გამოვითვალოთ როგორც კვადრატული ფესვი **std** ფუნქციიდან. ალბათობის თეორია უჩვენებს, რომ დისპერსია შემდეგნაირადაც შეგვიძლია გამოვთვალოთ:

$$\sigma^2 = \frac{\sum_{k=1}^{N} x_k^2}{N} - \mu^2 \qquad (5.6)$$

თანაბარი განაწილების გადახრა ასევე ალბათობის თეორიის გამოყენებით სხვაგვარადაც შეგვიძლია გამოვთვალოთ:

$$\sigma^2 = \frac{(upper\_bound - lower\_bound)^2}{12} \qquad (5.7)$$

რადგან ზედა დაქვედა საზღვრებს შორის სხვაობა განაწილების ფუნქციის სიგანეს უდრის:

$$\sigma^2 = \frac{width^2}{12} \qquad (5.8)$$

შევნიშნავთ, რომ 5.6 განტოლება ვრცელდება შემთხვევით სიდიდეთა ყველა სახის მიმდევრობაზე, მაშინ როცა 5.8 – მხოლოდ თანაბარი განაწილებისთვისა ვარგისი. ასევე მნიშვნელოვანია გვახსოვდეს, რომ როცა საქმე გვაქვს რეალურ მონაცემებთან, საშუალო და დისპერსია თეორიულ მნიშვნელობებთან ყოველთვის როდია კარგ თანხვდომაში. თუმცა მოცემულ ინტრვალში მიმდევრობის რაც უფრო მეტი რაოდენობის წევრებს ავიღებთ, გამოთვლილი და თეორიული მნიშვნელობები მით უფრო დაუახლოვდებიან ერთმანეთს. ამის საილუსტრაციოდ შევქმნათ MATLAB საშუალებით შემთხვევით რიცხვთა მიმდევრობა თანაბარი განაწილებით ინტერვალში [-5 5]. თეორიული საშუალო 0 –ის ტოლია, ხოლო თეორიული დისპერსია – 100/12 = 8.333. ცხრილში მოყვანილია მიმდევრობის წევრთა რაოდენობა და MATLAB ფუნქციებით გამოთვლილი საშუალო და დისპერსიის შესაბამისი მნიშვნელობები.

მიმდევრობის წევრთა რაოდენობა	საშუალო	გადახრა
10	0.1286	9.1868
100	0.3421	8.0959
1000	0.0036	7.8525
5000	-0.0022	8.2091

რაც მეტია მოცემულ ინტერვალში შემთხვევით სიდიდეთა რაოდენობა, თეორიული და გამოთვლილი მნიშვნელობები მით მეტად უახლოვდება ერთმანეთს.

თუ **seed** სხვადასხვა მნიშვნელობას ავიღებთ შემთხვევითი სიდიდეთა გენერირებისას, ანდა თუ **rand** ბრძანებას გავიმეორებთ ზედიზედ seed მნიშვნელობის შეუცვლელად, მივიღებთ შემთხვევით სიდიდეთა განსხვავებულ მიმდევრობას.

სავარჯიშო

გამოთვალეთ წინასწარგანსაზღვრული მახასიათებლების მქონე შემთხვევით რიცხვთა მიმდევრობის საშუალო და გადახრა. MATLAB საშუალებით აწარმოეთ განსაზღვრული

მახასიათებლების მქონე 1000 შემთხვევითი რიცხვი. გამოთვალეთ მიღებული განაწილების თეორიული საშუალო და დისპერსია და შეადარეთ იგი შესაბამის თეორიულ მნიშვნელობებს.

1. თანაბარი განაწილება ინტერვალში 0 და 10.
2. თანაბარი განაწილება ინტერვალში –1 და +1.
3. თანაბარი განაწილება ინტერვალში -20 და -10.
4. თანაბარი განაწილება ინტერვალში 4.5 და 5.
5. თანაბარი განაწილება ინტერვალში π და -π.

4.2.3 ნორმალური ანუ გაუსის განაწილება

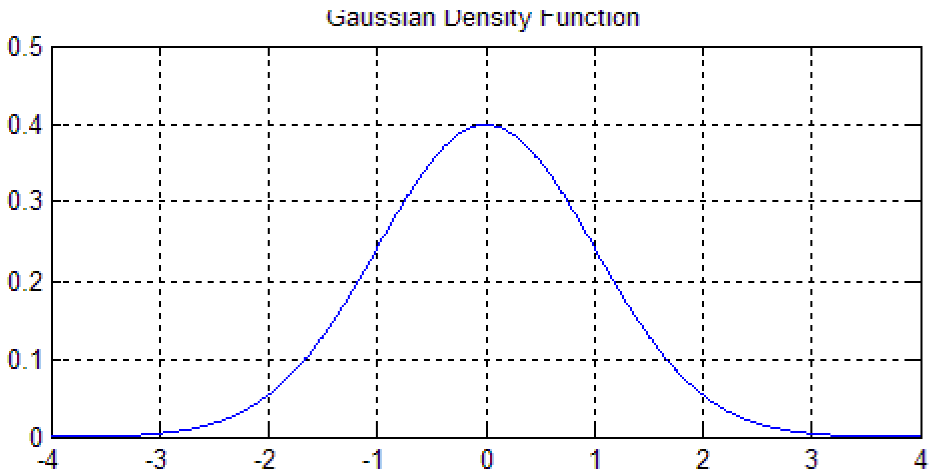
თანაბარი განაწილების დროს ყველა სიდიდის ხდომილობის ალბათობა ერთნაირია. ზოგჯერ გვჭირება შევქმნათ ისეთი განაწილება, სადაც სიდიდეთა რომელიღაც მნიშვნელობები უფრო ხშირად გვხვდება სხვა მნიშვნელობებთან შედარებით. მაგალითად, დავუშვათ შემთხვევით სიდიეთა მიმდევრობა წარმოადგენს გარემოს ტემპერატურის ანათვლებს დროის რომელიმე ინტერვალში. ვნახავთ, რომ ტემპერატურის მნიშვნელობები განსხვავდებიან, მაგრამ არ არიან ერთნაირად ალბათურნი. მაგალითად ტემპერატურა იცვლება მხოლოდ რამდენიმე გრადუსუს ფარგლებში, თუმცა შესაძლოა დიდი ცვლილებები აღინიშნოს ქარის ამოვარდნის, მოღრუბლულობის ანდა დღე – ღამის მონაცვლეობის გამო.

შემთხვევით სიდიდდეთა ასეთი მიმდევრობის მოდელირება შესაძლებელია გაუსის შემთხვევითი ცვლადით (ნორმალური). მისი სიმკვრივის ფუნქცია;

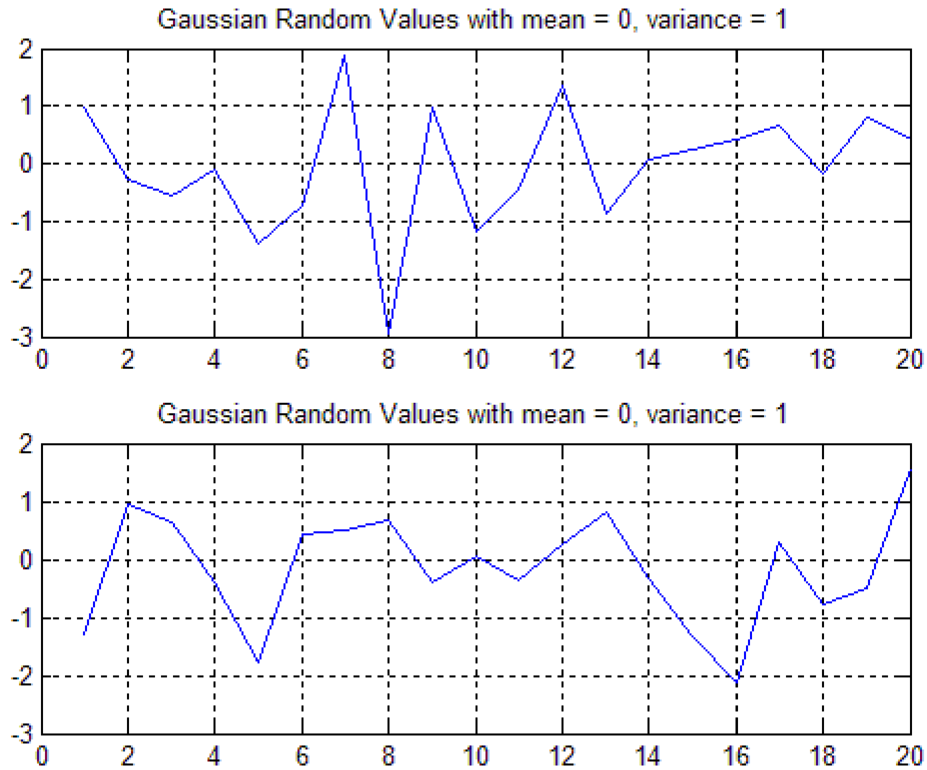
$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/(2\sigma^2)}$$

სადაც μ საშუალოა, ხოლო σ^2 გადახრა. მაგალითად სიმკვრივის ფუნქცია გაუსის განაწილებისათვის , სადაც μ=0 და σ^2 = 1 ნაჩვენებია ნახ. 5.8.

ასეთი განაწილების საშუალო მნიშვნელობა შეესაბამება სიმკვრივის ფუნქციის მაქსიმუმის წერტილის x კოორდინატს. სიმკვრივის ფუნქციის გრაფიკის მიხედვით შეამჩნევთ, რომ საშუალო მნიშვნელობების მახლობელი სიდიდეების გენერირების ალბათობა უფრო დიდია. შევნიშნავთ, რომ თანაბარ განაწილებას ახასიათებს მნიშვნელობათა ზედა და ქვედა საზღვარი, ხოლო გაუსის განაწილებას ასეთი საზღვრები არ გააჩნია. გაუსის შემთხვევოთი სიდიდეების უმრავლესობა საშუალოდან მცირედ განსხვავებულ მნიშვნელობებს იღებს, თუმცა ზოგიერთი მათგანი შესაძლოა საკმარისად დაშორდეს საშუალო მნიშვნელობას. 5.9 ნახაზი წარმოადგენს ორი განსხვავებული ნორმალური განაწილებას საშუალოთი 0 და გადახრით 1. შესაბამისი განაწილების სიმკვრივის ფუნქციის გარაფიკი მოცემულია ნახ. 5.8.

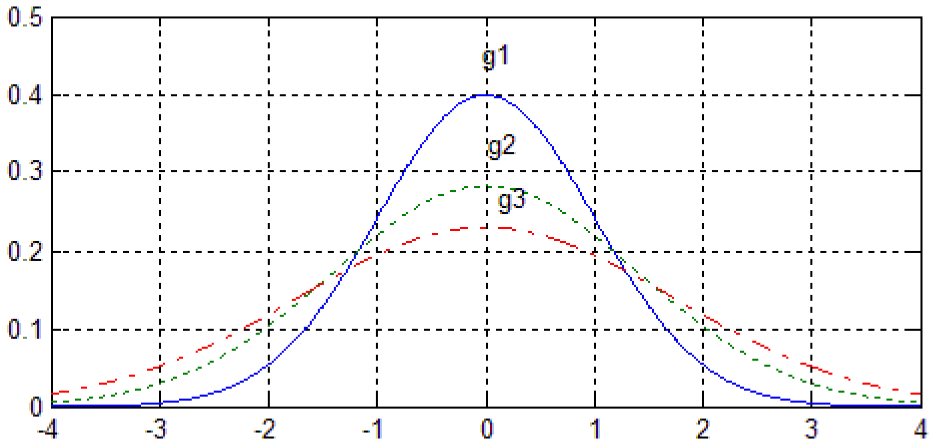


ნახ. 5.8 ნორმალური განაწილების სიმკვრივის ფუნქცია $\mu = 0$, $\sigma^2 = 1$.



ნახ. 5.9 შემთხვევით სიდიდეთა მიმდევრობა გაუსის განაწილებით. $\mu = 0$, $\sigma^2 = 1$.

ნახ 5.10 წარმოდგენილია გაუსის განაწილების სიმკვრივის სამი სხვადასხვა ფუნქცია. სამივე მათგანის საშუალო მნიშვნელობაა 5, მაგრამ მათი სტანდარტული გადახრა და დისპერსია სხვადასხვაა. g1 განაწილებას აქვს ყველაზე მცირე დისპერსია, g3 – ყველაზე დიდი.



ნახ. 5.10 გაუსის სამი სხვადასხვა განაწილების სიმკვრივის ფუნქცია

სტატისტიკიდან ცნობილია, რომ გაუსის განაწილებისათვის სიდიდეთა 68% ღებულობს მნიშვნელობებს საშუალოდან ±σ ინტერვალში, 95% ±2σ ინტერვალში, ხოლო 99% - ±3σ ინტერვალში.

MATLAB ფუნქცია **randn** ქმნის შემთხვევით სიდიდეთა მიმდევრობას საშუალოთი 0 და დისპერსიით 1 (გაუსის, ანუ ნორმალური განაწილებით). იმისათვის რომ შევქმნათ განაწილება განსხვავებული პარამეტრებით, ეს სიდიდე უნდა გავამრავლოთ შესაბამის სტანდარტულ გადახრაზე და დავუმატოთ საშუალო მნიშვნელობა. ამგვარად, თუ r შემთხვევითი სიდიდეა საშუალოთი 0 და სტანდარტული გადახრით 1, შემდეგი განტოლება აწარმოებს შემთხვევით სიდიდეს x საშუალოთი b და სტანდარტული გადახრით a:

$$x = a \cdot r + b$$

შემდეგი ბრძანებები გვაძლევს შემთხვევით სიდიდეთა მიმდევრობას გაუსის ანუ ნორმალური განაწილებით, რომლის საშუალოა 5 და დისპერსია 2:

```
randn (‘seed’, 0)
s = sqrt(2)*randn(10,1) + 5
```

მივიღებთ ვექტორს s:

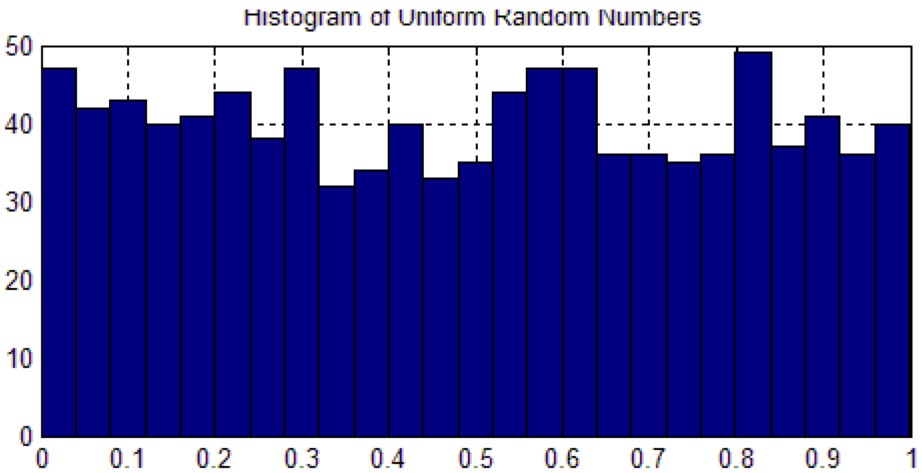
```
s =

   6.6475
   5.8865
   5.1062
   5.4972
   4.0150
   7.3987
   5.0835
   7.5414
   5.3734
   6.2327
```

4.2.4 სიმკვრივის ფუნქცია

განყოფილებაში 5.1 განვიხილეთ ბრძანება **hist**, რომელიც გვაძლევს ჰისტოგრამას. ეს ბრძანება შეიძლება გამოვიყენოთ სიგნალის სიმკვრივის ფუნქციის შესაფასებლად. თუ გვაქვს 1000 შემთხვევითი სიდიდე და ავაგებთ მათ ჰისტოგრამას, ფაქტიურად ჩვენ ვაგებთ მათი განაწილების სიმკვრივის ფუნქციის გრაფიკს. მაგალითად MATLAB საშუალებით შევქმენით თანაბრად განაწილებული 1000 რიცხვის მიმდევრობა 0 და 1 შორის, ჩავწერეთ ისინი სვეტ ვექტორში u_values. შეგვიძლია ვისარგებლოთ **hist** ბრძანებით იმისათვის, რომ ავაგოთ სათანადო განაწილების ფუნქცია 25 სვეტით:

```
u_values = rand(1000,1);
hist(u_values,25)
```

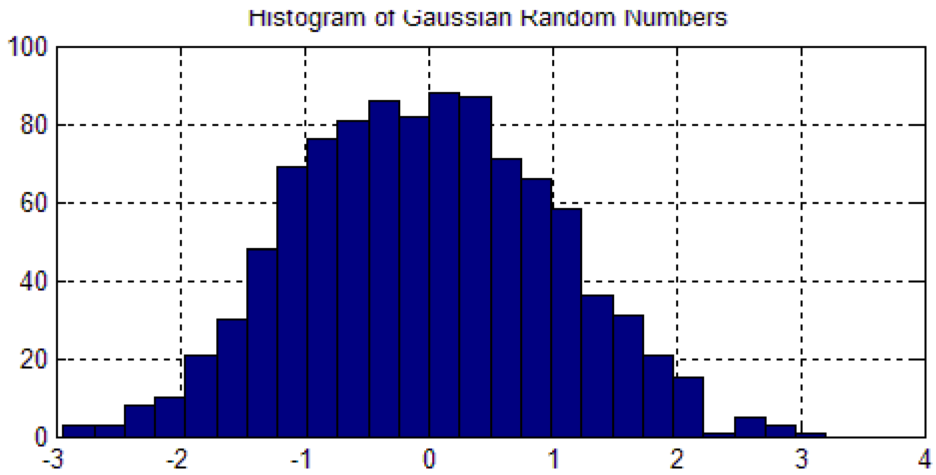
მივიღებთ გრაფიკს 5.11. როგორც მოსალოდნელი იყო, სიდიდეები განაწილებული არიან 0 და 1 შორის და განაწილება შედარებით თნაბარია. ახლა გავიმეოროთ იგივე ნორმალური განაწილებისათვის საშუალოთი 0 და გადახრით 1.



ნახ. 5.11 თანაბარი განაწილების ჰისტოგრამა

```
g_values = randn(1000,1);
hist(u_values,25)
```

მივიღებთ გრაფიკს 5.12. როგორც მოსალოდნელი იყო, განაწილების გრაფიკის პიკია 0, საშუალო მნიშვნელობა და მონაცემთა უმრავლესობის მნიშვნელობები მოთავსებულია ინტერვალში $\pm 2\sigma$.



ნახ. 5.12 ნორმალური განაწილების ჰისტოგრამა

სავარჯიშო

MATLAB საშუალებით აწარმოეთ 1000 შემთხვევითი სიდიდე განსაზღვრული მახასიათებლებით. გამოთვალეთ მისი საშუალო და სტანდარტული გადახრა. გამოთვლილი და მოცემული მნიშვნელობები ერთმანეთთან ახლოს უნდა იყოს. ააგეთ მათთვის ჰისტოგრამა 25 სვეტით:

1. შემთხვევით სიდიდეთა გაუსის განაწილება საშუალოთი 1 და სტანდარტული გადახრით 0.5
2. შემთხვევით სიდიდეთა გაუსის განაწილება საშუალოთი –5.5 და სტანდარტული გადახრით 0.25
3. შემთხვევით სიდიდეთა გაუსის განაწილება საშუალოთი –5.5 და სტანდარტული გადახრით 1.25
4. შემთხვევით სიდიდეთა გაუსის განაწილება საშუალოთი π და სტანდარტული გადახრით π/8.

პრობლემა: თვითმფრინავის ფრენის მოდელირება

კომპიუტერული მოდელირება გულისხმობს პროგრამულად შეიქმნას სიტუაცია, რომელიც რეალური პროცესების იმიტაციას წარმოადგენს. მოდელირების ნიმუშია მაგალითად კომპიუტერული თამაშები. კომპიუტერული თამაშის დროს თქვენი მოქმედების მიხედვით პროგრამა ირჩევს შესაბამის პასუხს. ზოგიერთი ანიმაციური თამაში იყენებს კომპიუტერულ გრაფიკას იმისათვის, რომ უპასუხოს მოთამაშის რეაქციას, კლავიატურით თუ მაუსით მანიპულირებას. მოდელირების უფრო სრულყოფილ პროგრამებში, როგორიცაა ფრენის სიმულატორი, კომპიუტერი არა მარტო პასუხობს სათანადოდ მომხმარებლის მიერ მიწოდებულ ბრძანებებს, არამედ სიტუაციის შესაბამისად აწარმოებს ისეთ სიდიდეებს, როგორიცაა ტემპერატურა, ქარის სიჩქარე და თვითმფრინავის მდებარეობა. სიმულატორი, ამასთანავე, ახდენს მოულოდნელი მოვლენების მოდელირებას, რომელთაც შესაძლოა ადგილი ჰქონდეს თვითმფრინავის ფრენისას. იმისათვის, რომ პროგრამის მიერ ასახული სიტუაცია, რაც შეძლება ახლოს იყოს რეალურთან, გენერირებული მონაცემები შემთხვევით ხასიათს უნდა ატარებდეს. მონტე კარლოს მეთოდი იყენებს შემთხვევით რიცხვებს მოვლენების მოდელირებისათვის.

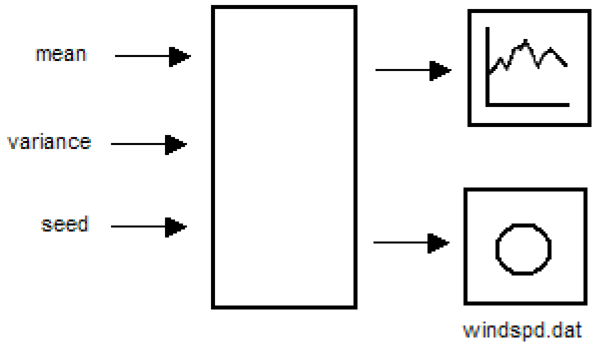
დაწერეთ პროგრამა რომელიც აწარმოებს შემთხვევვით სიდიდეთა მიმდევრობას, იმისათვის, რომ მოხდეს ქარის სიჩქარის მოდელირება, რომლის ხანგრძლივობა იქნება 1 სათი, მონაცემები განახლდება ყოველ 10 წუთში (მივიღებთ 361 მოვაცემს). რელური ექპერიმენტის მონაცემთა ანალიზის საფუძველზე დადგენილია, რომ ქარის სიჩქარის მოდელირებისათვის შეგვიძლია გამოვიყენოთ ნორმალური შემთხვევითი სიდიდე (გაუსის განაწილება. ჩავთვალოთ, რომ საშუალო და გადახრა მოცემული რეგიონისათვის წელიწადის განსაზღვრულ დროს გარკვეული, დადგენილი, ცნობილი სიდიდეებია და პროგრამაში ვიყენებთ როგორც მიწოდებულ, INPUT პარამეტრებს. ამასთან, შეფასებულია, რომ არსებობს 1% შანსი, ალბათობა იმისა, რომ თვითმფრინავი მოხვდეს მცირედ ქარიშხალში, ისე, რომ ქარის სიჩქარე გაიზარდოს 10 მილი/წმ –ით, ხოლო 0.01% ალბათობა იმისა, რომ მოხვდეს ძლიერ გრიგალში, რომელიც სიჩქარეს 50 მილი/წმ-ით გაზრდის. ააგეთ ქარის სიჩქარის გრაფიკი დროის მიმართ და შეინახეთ მონაცემები ASCII ფორმატით ფაილში სახელით windspd.dat.

1. **ამოცანის დასმა**

შევქმნათ ერთსაათიანი ქარის მოდელი სტატისატიკური პარამეტრების გათვალისწინებით.

2. **INPUT/OUTPUT აღწერა**

როგორც ჩანს 5.13 ნახაზიდან, პროგრამის საწყისი (INPUT) მნიშვნელობაა ამინდის სტატისტიკური მონაცემები თვითმფრინავის ფრენის გზაზე: ქარის სიჩქარის საშუალო მნიშვნელობა და სტანდარტული გადახრა ჩვეულებრივ პირობებში. შედეგად კი ვღებულობთ მონაცემთა ფაილს და გრაფიკს, რომელიც ასახავს ქარის სიჩქარის ცვლილებას.



nax. 5.13 I/O დიაგრამა

3. **სახელდახელო ამოხსნა**

პროგრამა იყენებს შემთხვევით სიდიდეთა სხვადასხვა მიმდევრობას გარკვეული საშუალოთი და დისპერსიით. მცირე ქარიშხლისა და გრიგალის ხდომილების ალბათობა მოცემულია პროცენტებში და წარმოადგენს შემთხვევით სიდიდეებს (განაწილება თანაბარია). გრიგალის მოდელირებისათვის ვაწარმოებთ შეთხვევით სიდიდეს 0 და 1 შორის ბრძანებით **rand** და ვუშვებთ, რომ მცირე ქარიშხალს ადგილი ექნება, თუ მიღებული სიდიდე მოთავსდება საზღვრებში [0.0, 0.01], ხოლო გრიგალს იმ შემთხვევაში ექნება ადგილი, თუ მიღებული სიდიდე მოთავსდება საზღვრებში [0.01 0.0101].

4. **MATLAB** ამოხსნა

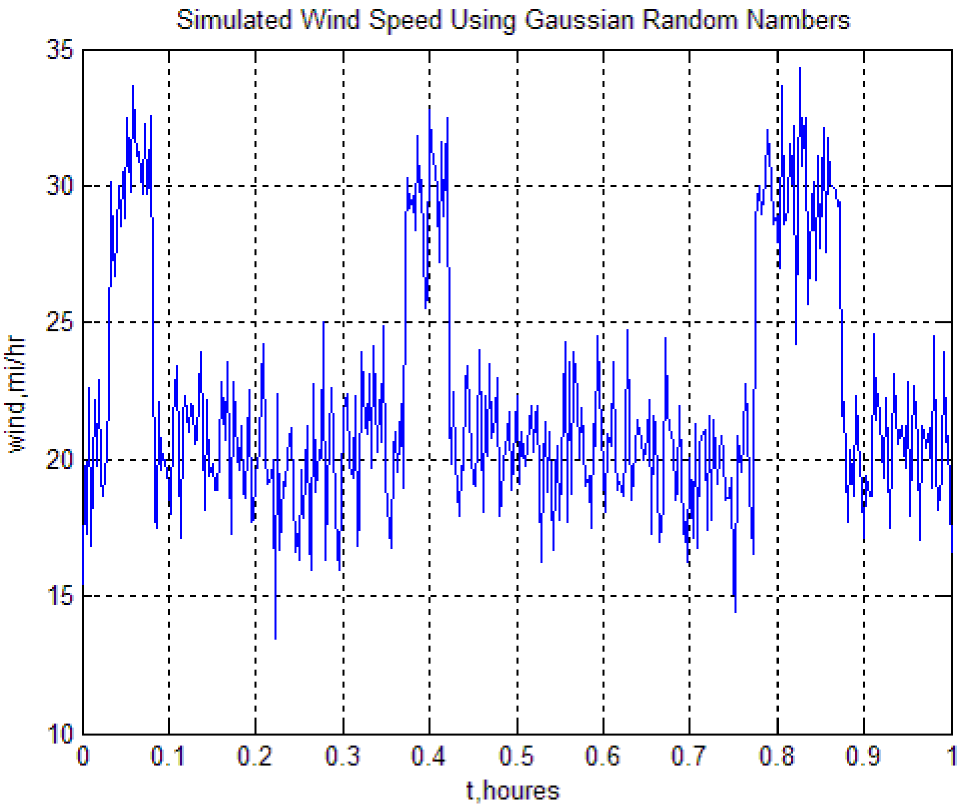
```
%        this program generates one hour of simulated wind
%        such that it uses uniform random numbers to generate
%            the initial wind and compares it to the wind generating
with using Gaussian nambers
%

mn_speed=input('Enter mean of wind speed  ');
var_speed=input('Enter variance of wind speed  ');
std_speed=sqrt(var_speed);
seed=input('Enter seed for random numbers  ');
%
%       generate simulated wind speed without storms
%
rand('seed',seed)
speed=std_speed*randn(1,361)+mn_speed;

%Add simulated storms and microbursts

t=[0:1:360]*(1/360);
k=1;
while k<=361
    random_x=rand(1);
    if random_x<=0.01
        end_storm=min(361,k+17);
        speed(k:end_storm)=speed(k:end_storm)+10;
        k=k+18;
    elseif 0.01<random_x&random_x<=0.0101
        end_micro=min(361,k+5);
        speed(k:end_micro)=speed20(k:end_micro)+50;
        k=k+6;
    else
        k=k+1;
    end
end
%
%                 Plot the data
%

    plot(t,speed),...
    title('Simulated Wind Speed Using Gaussian Random Nambers'),...
    xlabel('t,houres'),...
    ylabel('wind,mi/hr'),...
    grid

data(:,1) = t';
data(:,2) = speed';
save windspd.dat data  /ascii
```

5. შემოწმება

თუ პროგრამას გაშვების შემდეგ მივაწოდებთ მნიშვნელობებს:

```
Enter mean of wind speed  20
Enter variance of wind speed  5
Enter seed for random numbers  0
```

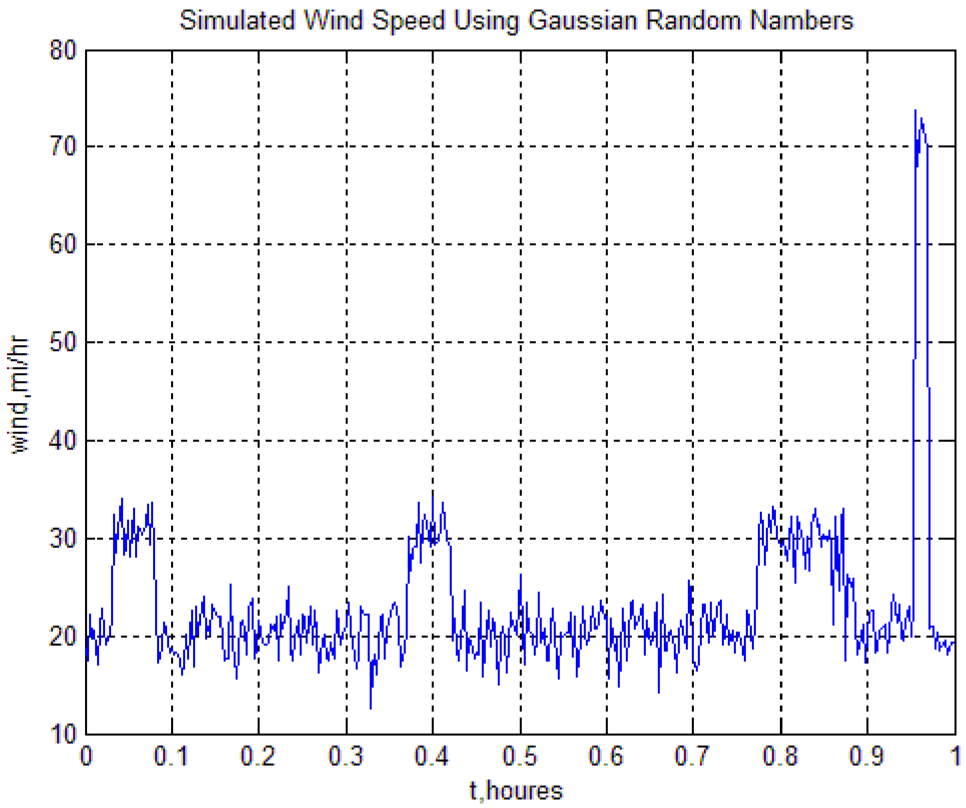
მივიღებთ სურათს ნახ. 5.14



ნახ. 5.15 მოდელირება სამი ქარიშხლით

ქარის უფრო რეალური სურათი რომ მივიღოთ, შემდეგი ცვლილებები შეგვიძლია შევიტანოთ პროგრამაში: გრიგალის განმავლობაში სიჩქარის ნაზრდი შეიძლება იყოს შემთხვევითი სიდიდე თანაბარი განაწილებით საშუალოთი 10 და დისპერსიის დიდი მნიშვნელობით. შეგვიძლია აგრეთვე დავუმატოთ ბრძანება, რომელიც განსაზღვრავს სიჩქარის მომატებას და კლებას არა მკვეთრად, არამედ თანდათანობით. შეგვიძლია გრიგალის ხანგრძლივობა ავიღოთ თანაბარი განაწილებით შემთხვევითი სიდიდე 1 წუთიდან 10 წუთამდე.

თუ პროგრამას გავუშვებთ რამდენიმეჯერ, **seed** სხვადასხვა მნიშვნელობით, გრაფიკზე გრიგალის შემთხვევასაც მივიღებთ. (ნახ. 5.15).



ნახ. 5.15 ქარის სიჩქარის მოდელირება (გრიგალის შემთხვევა)

1.3 თანაფარდობა სიგნალი/ხმაური

საცდელი საინჟინრო ამოცანების მოდელირებისას ხშირად საჭიროა ისეთი სიგნალის მოდელირება, რომელსაც ახლავს შემთხვევითი ხმაური. (ნახ. 5.16). ხმაური შემთხვევით სიდიდეთა ერთობლიობაა და მისი წილი მიღებულ საგნალში განისაზღვრება თანაფარდობით სიგნალი/ხმაური – SNR. იგი სიგნალის სიმძლავრის ერთეულებში გამოისახება. ჯერ განვიხილოთ რას წარმოადგენს სიგნალის სიმძლავრე და შემდეგ დავუბრუნდეთ SNR.

4.3.1 სიგნალის სიმძლავრე

სიმძლავრე – ეს არის სიგნალის ამპლიტუდის ზომა. რაც მეტია სიგნალის ამოლიტუდა, მით მეტია მისი სიმძლავრე. რამდენადაც ამპლიტუდა შეიძლება იყოს დადებითიც და უარყოფითიც, სიმძლავრე განისაზღვრება ამპლიტუდის კვადრატით, ისე რომ სიმძლავრის მნიშვნელობა ყოველთვის დადებითი სიდიდეა. x ვექტორით წარმოდგენილი სიგნალის სიმძლავრე შეიძლება შევაფასოთ სიგნალის მნიშვნელობათა კვადრატების საშუალო მნიშვნელობით:

$$power \approx \frac{\sum_{1}^{N} x_k^2}{N}$$

MATLAB –ში ეს სიდიდიე გამოითვლება **sum** ფუნქციის სასუალებით:

power = sum(x^2)/N;

შესაძლებელია აგრეთვე ვაჩვენოთ, რომ სიგნალის სიმძლავრე ტოლია დისპერსიისა და საშუალო მნიშვნელობის კვადრატის ჯამისა:

$$power = \sigma^2 + \mu^2$$

MATLAB საშუალებით ეს შეიძლება შემდეგი ბრძანებებით გამოითვალოს:

power = std(x)^2 + mean(x)^2;

თუ სიგნალი სინუსოიდაა, ადვილად შეიძლება ვაჩვენოთ, რომ მისი სიმძლავრე ტოლია მისივე ამპლიტუდის კვადრატის ნახევრის. მაგალითად სინუსოიდის 4sin2πt სიმძლავრე ტოლია 16/2, ანუ 8.

სავარჯიშო

მოცემული პარამეტრების საშუალებით აწარმოეთ 100 შემთხვევითი სიდიდე. გამოთვალეთ სიმძლავრე ორი სხვადასხვა გზით (მნიშვნელობათა კვადრატების სშუალო და საშუალოსა და დისპერსიის მნიშვნელობათა მიხედვით) და შეადარეთ ერთმანეთს.

1. თანაბარად განაწილებული შემთხვევითი სიდიდეები ინტერვალში 0 და 10.
2. თანაბარად განაწილებული შემთხვევითი სიდიდეები ინტერვალში -2 და 4.
3. თანაბარად განაწილებული შემთხვევითი სიდიდეები საშუალოთი 0 და გადახრით 1.0
4. თანაბარად განაწილებული შემთხვევითი სიდიდეები საშუალოთი –0.5 და დისპერსიით 4.0
5. შემთხვევოთი სიდიდეები ნორმალური განაწილებით საშუალოთი 0 და დისპერსიით 2.0
6. შემთხვევოთი სიდიდეები ნორმალური განაწილებით საშუალოთი 0 და დისპერსიით 0.5
7. შემთხვევოთი სიდიდეები ნორმალური განაწილებით საშუალოთი –2.5 და დისპერსიით 0.5

4.3.2 SNR გამოთვლა

თანაფარდობა სიგნალი – ხმაური არის სიგნალის სიმძლავრის თანაფარდობა ხმაურის სიმძლავრესთან. მაგალითად, SNR = 1, ნიშნავს, რომ სიგნალს სიმძლავრესა და ხმაურის სიმძლავრის თანაფარდობა = 1:1. თუ მოცემული გვაქვს სინუსოიდა ამპლიტუდით 3 და თანაბრი ხმაურით –1 და 1 შორის, შეგვიძლია გამოვთვალოთ SNR შესაბამის სიმძლავრეთა გამოთვლებზე დაყრდნობით:

$$SNR = \frac{9/2}{1/3} = 13.5$$

თუ მოცემული გვაქვს ძირითადი სინუსოიდური სიგნალი ამპლიტუდით A და თანაბრი ხმაური ინტერვალში [a b], MATLAB საშუალებით SNR ასე გამოითვლება

SNR = ((A^2)/2)/((b-a)^2/12

საილუსტრაციოდ დავუშვათ გვსურს შევქმნათ 201 ელემენტიანი ვექტორი სიგნალი, რომელიც შეიცავს 1 ჰერციან სინუსოიდას ხმაურით, რომლის საშუალოა 0 ისე, რომ SNR = 46. სინუსოიდას უნდა ჰქონდეს ამპლიტუდა 1.5 და ფაზური კუთხე 0.0, ათვლილი უნდა იყოს 100 ჰერცით (ეს ნიშნავს, ათვლა წარმოებს ყოველ 1/100 წამში):

$$\text{SNR} = \text{სიგნალის სიმძლავრე}/\text{ხმაურის სიმძლავრე} = ((1.5^2)/2)/\text{ხმაურის სიმძლავრე} = 46$$

აქედან მივიღებთ:

$$\text{ხმაურის სიმძლავრე} = (1.5^2)/2/46 = 0.024$$

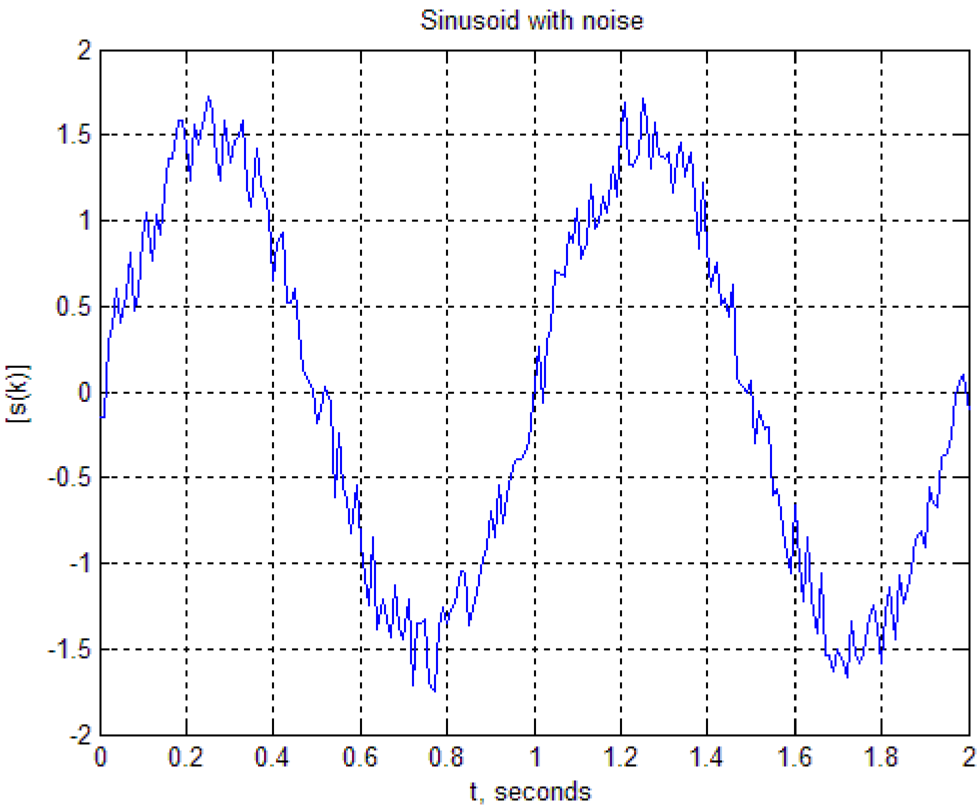
რადგანაც ხმაურის საშუალო = 0, ხმაურის სიმძლავრე ტოლია მისი დისპერსიისა. ე.ი ხმაურის გადახრა ტოლი ყოფილა 0.024, რადგან ხმაური თანაბარია და მისი საშუალო 0 – ის ტოლია, მისი მნიშვნელობები უნდა მდებარეობდეს [a –a] შორის. როგორც ვიცით, დისპერსია ტოლია $(2a)^2/12$, გვექნება:

$$0.024 = (2a)^2/12$$

ანუ a=0.27.

ახლა შეგვიძლია გამოთვლილი აღნიშნული პარამეტრებით შევქმნათ ხმაურის მნიშვნელობები და დავუმატოთ სინუსოიდურ სიგნალს:

```
%
%       Generate and plot sine plus noise.
%
rand('seed',0);
t=(0:0.01:2);
s=1.5*sin(2*pi*t) + (0.54*rand(1,201) -0.27);
plot(t,s),...
    title('Sinusoid with noise'),...
    xlabel('t, seconds'),...
    ylabel('[s(k)]'),...
    grid
```



ნახ. 5.16 სინუსოიდური სიგნალი ხმაურით.

სინუსოიდური სიგნალი თანაბარი ხმაურით ნაჩვენებია გრაფიკზე 5.16. შევნიშნავთ, რომ გრაფიკი მოიცავს სინუსოიდის ორ პეროოდს 2 წამის განმავლობაში, რაც შეესაბამება 1 ჰერციან სინუსოიდას, ესე იგი პერიოდი = 1 წამს.

4.3.3 არსებულ სიგნლზე ხმაურის დამატება

დავუშვათ გვაქვს სიგნალი, რომელიც ათვლილია და შენახულია მონაცემთა ფაილის სახით. თუ გვსურს დავუმატოთ ასეთ სიგნალს ხმაური, რომელიც მოგვცემს SNR განსაზღვრულ მნიშვნელობას, ჯერ უნდა შევაფასოთ სიგნალის სიმძლავრე, იმისათვის, რომ განვსაზღვროთ ხმაურის სიგნალის შესაბამისი სიმძლავრე. სიგნალის სიმძლავრის შესაფასებლად გამოვთვალოთ სიგნალის მნიშვნელობათა კვადრატების საშუალო, რომელიც MATLAB საშუალებით ადვილად გამოითვლება. ამის შემდეგ შეგვიძლია განვსაზღვროთ ხმაურის სიგნალის სიმძლავრე. ვიცით, რომ სიმძლავრე საშუალოსა და დისპერსიის ფუნქცია ამიტომ ერთ-ერთი მათგანი მაინც უნდა ვიცოდეთ, რომ მეორე განვსაზღვროთ. სასურველია, რომ ხმაურის საშუალო მნიშვნელობა 0 –ის ტოლი იყოს, ამ დაშვებას ხშირად მიმართავენ, თუ სხვა ინფორმაცია არ არსებობს. უკვე შეგვიძლია გამოვთვალოთ დისპერსია, მივიღოთ ვექტორი, რომელიც ხმაურის მნიშვნელობებს შეიცავს და დავუმატოთ იგი ძირითად სიგნალს.

სავარჯიშო

ააგეთ სიგნალი, რომელიც შედგება 5 ჰერციანი სინუსოიდის 100 წერტიისაგან ხმაურით, რომლის საშუალოა 0 და ხასიათდება პარამეტრებით:

5.1 თანაბარი ხმაური, SNR =5
5.1 თანაბარი ხმაური, SNR =1
5.1 თანაბარი ხმაური, SNR =0.2
5.1 ხმაური გაუსის განაწილებით, SNR =5
5.1 ხმაური გაუსის განაწილებით, SNR =1
5.1 ხმაური გაუსის განაწილებით, SNR =0.2

ამ თავში წარმოგიდგინეთ მონაცემთა სტატისტიკური ანალიზისათვის საჭირო მთელი რიგი ფუნქციებისა. განვიხილეთ აგრეთვე მაგალითი თუ როგორ შევქმნათ ზოგიერთი საინჟინრო პრობლემის მოდელირებისათვის საჭირო მონაცემები და მივცეთ მათ სტატისტიკური შეფასება. მაგალითებში განხილული იყო საკომუნიკაციო სიგნალის ანალიზი, მოდელირება მონტე-კარლოს მეთოდით და სინალი – ხმაური თანაფარდობის შეფასება.

ბრძანებები და ფუნქციები

cumprod	განსაზღვრავს კუმულაციურ ნამრავლს
cumsum	განსაზღვრავს კუმულაციურ ჯამს
hist	აგებს ჰისტოგრამას
max	განსაზღვრავს უდიდეს მნიშვნელობას
mean	განსაზღვრავს საუალო მნიშვნელობას
median	განსაზღვრავს მედიანას მნიშვნელობას
min	განსაზღვრავს უმცირეს მნიშვნელობას
prod	განსაზღვრავს სიდიდეთა ნამრავლს
rand	განსაზღვრავს შემთხვევით რიცხვებს
sort	დაალაგებს სიდიდეებს
std	გამოითვლის სტანდარტულ გადახრას
sum	განსაზღვრავს სიდიდეთა ჯამს

ამოცანები

ამოცანები 1-7 დაკავშირებულია საინჟინრო პრობლემებთან, რომლებიც ამ თავშია განხილული, ხოლო დანარჩენი ამოცანები სხვა საკითხებს ახება.

საკომუნიკაციო სიგნალი. ეს საკითხები დაკავშირებულია ამ თავში განხილულ ამოცანასთან საკომუნიკაციო სიგნალის შესახებ.

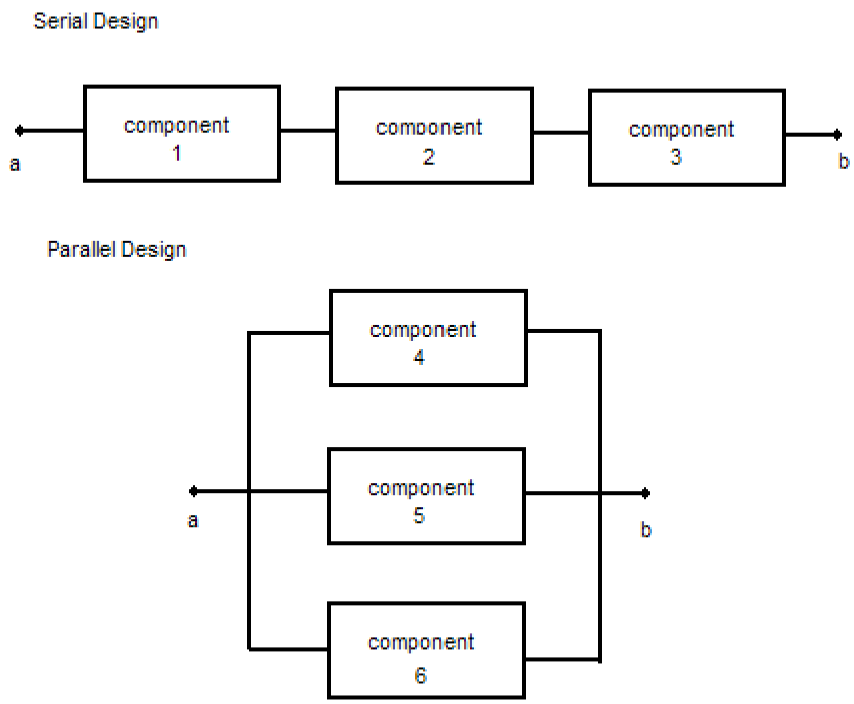
1. მიკროფონის საშუალებით ჩაწერეთ თქვენს მიერ წარმოთქმული სიტყვები – ნული, ერთი, ორი, სამი, . . . ცხრა ცალკე ფაილების სახით. გაუშვით პროგრამა, რომელიც დაწერეთ ამ თავში განხილული ამოცანისათვის საკომუნიკაციო საგნალის შესახებ თითოეული ფაილის მონაცემებისათვის და დაბეჭდეთ შესაბამისი ცხრილები.

2. გაუშვით იგივე პროგრამა სხვადასხვა ადამიანის მიერ წარმოქთმული ერთიდაიგივე სიტყვის შესაბამისი ფაილებისათვის და დაბეჭდეთ ცხრილი შესაბამის სტატიტიკურ მონაცემთა შესადარებლად.

თვითმფრინავის ფრენის მოდელირება. ქარის სიჩქარე. ეს პრობლემებიც ასევე ამთავში განხილულ პრობლემას უკავშირდება ქარის მოდელირებასთან დაკავშირებით.

3. ამ თავში ქარის მოდელირებასთან დაკავშირებით დაწერილი პროგრამა ისე შეცვალეთ, რომ მან ქარის სიჩქარის საწყისი მნიშვნელობებისათვის გამოიყენოს თანბრად განაწილებული შემთხვევითი სიდიდეები. შეადარეთ შედეგები შემთხვევით სიდიდეთა ორი სხვადასხვა ტიპის განაწილების შემთხვევაში.
4. იგივე პროგრამა შეცვალეთ ისე, რომ გრიგალის დროს სიჩქარის მატება იყოს თანაბარი შემთხვევითი სიდიდე საშუალოთი 10 მილი/საათში და სტანდარტული გადახრით 1.5 მილი/საათში.
5. იგივე პროგრამა შეცვალეთ ისე, რომ გრიგალის დროც სიჩქარის მატება იყოს თანაბარი შემთხვევითი სიდიდე, რომლის საშუალოა 10 მილი/საათში და გადახრა ორჯერ აგემატებოდეს გადახრას, რომელიც ახასიათებს ქარის სიჩქარეს გრიგალის გრეშე.
6. იგივე პროგრამა შეცვალეთ ისე, რომ საჭირო იყოს პროგრამისათვის გრიგალის და ქარიშხლის ხდომილებათა ალბათობის მითითება.
7. იგივე პროგრამა შეცვალეთ ისე, რომ გრიგალის ხანგრძლივობა იყოს შემთხვევითი სიდიდე ინტერვალში 1 – 10 წუთი.

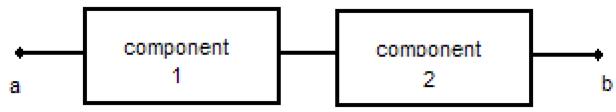
კომპონენტთა საიმედოობა. კომპონენტთა საიმედოობის ანალიზის განტოლებები გამომდინარეობს ალბათობის და სტატისტიკის თეორიიდან. საიმედოობა ასევე შეგვიძლია განვსაზღვროთ კომპიუტერული მოდელირების საფუძველზე, თუ თითოეული ელემენტის საიმედოობა ცნობილი სიდიდეა. განვიხილოთ დიაგრამა ნახ. 5.17. მიმდევრობითი შეერთების ამ სქემაში შეერთებულია 3 კომპონენტი. იმისათვის, რომ ინფორმაცია a წერტილიდან b –ში მივიდეს, სამივე კომპონენტი უნდა მუშაობდეს. პარალელური შეერთების შემთხვევაში სამიდან ერთი მაინც უნდა მუშაობდეს იგივე ინფორმაციის გადასაცემად. თუ ვიცით კომპონენტის საიმედოობა (რომელიც პროპორციულია იმ დროისა, როცა იგი სწორად მუშაობს), შეგვიძლია დავწეროთ განტოლება, რომელიც გამოითვლის მთელი სისტემის საიმედოობას. სისტემის საიმედოობა შეგვიძლია შევაფასოთ კომპიუტერული მოდელირების საფუძველზე. მაგალითად თუ მიმდევრობით ჩართული სისტემის ელემენტთა საიმედოობა 0.8 ტოლია (რაც ნიშნავს, რომ კომპონენტი დროის 80% სწორად მუშაობს), შეგვიძლია ვაწარმოოთ 3 შემთხვევითი სიდიდე 0 - 1 ინტერვალში. თუ სამივე მათგანი ნაკლებია ან ტოლია 0.8, მაშინ სისტემა იმუშავებს. თუ რომელიმე მათგანი მეტია 0.8, მაშინ კომპონენტი არ იმუშავებს (for one simulation). თუ ასეთ სიმულაციას 100 და 1000 ჯერ გავიმეორებთ, შევძლებთ გამოვთვალოთ წილი იმ დროისა, როცა სისტემა მუშაობს. იმისათვის, რომ შევაფასოთ პარალელური სისიტემის



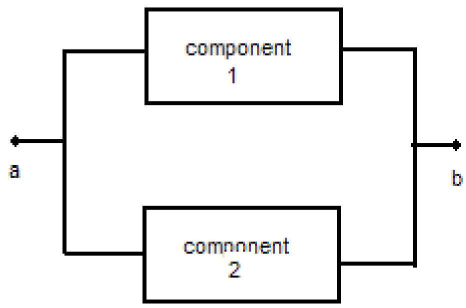
ნახ. 5.17

მუშაობის საიმედოობა, იმგვარადვე ვიქცევით. თუ ამ სამი რიცხვიდან ერთი მაინც ნაკლებია ან ტოლია 0.8, სისტემა იმუშავებს. თუ სამივე შემთხვევითი სიდიდე აჭარბებს 0.8, მაშინ სისტემა არ იმუშავებს.

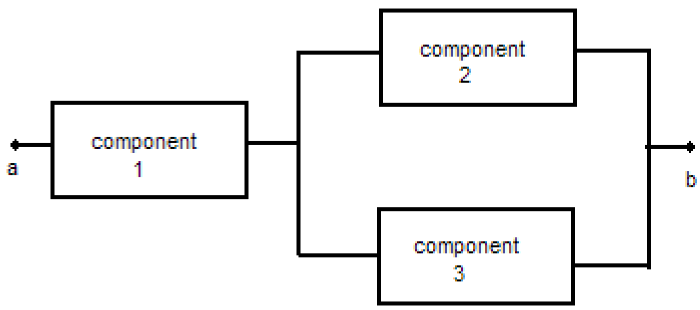
8. დაწერეთ პროგრამა მიმდევრობით შეერთებული სისტემის მოდელირებისათვის (ნახ. 5.17), ჩათვალეთ, რომ კომპონენტების საიმედოობაა (reliability) 0.8. გამოითვალეთ საიმედოობა 500 სიმულაციისათვის, 1000 სიმულაციისთვის
9. დაწერეთ პროგრამა პარალელურად შეერთებული სისტემის მოდელირებისათვის (ნახ. 5.17), ჩათვალეთ, რომ კომპონენტების საიმედოობაა (reliability) 0.8. გამოითვალეთ საიმედოობა 500 სიმულაციისათვის, 1000 სიმულაციისთვის. რომელი სისტემა უფრო საიმედოა?
10. დაწერეთ პროგრამა ნახ. 5.18 მოცემული სისტემის მოდელირებისათვის, თუ ერთი კომპონენტის საიმედოობაა 0.8, ხოლო მეორის – 0.92. დაბეჭდეთ საიმედოობა 5000 სიმულაციის შემთხვევაში.
11. დაწერეთ პროგრამა ნახ. 5.19 მოცემული სისტემის მოდელირებისათვის, თუ ერთი კომპონენტის საიმედოობაა 0.8, ხოლო მეორის – 0.92. დაბეჭდეთ საიმედოობა 5000 სიმულაციის შემთხვევაში. შეადარეთ პარალელური მიმდევრობით ჩართული სისტემის საიმედოობა.
12. დაწერეთ პროგრამა ნახ. 5.20 მოცემული სისტემის მოდელირებისათვის, თუ ერთი კომპონენტის საიმედოობაა 0.8, მეორის – 0.85, ხოლო მესამის – 0.95. დაბეჭდეთ საიმედოობა 5000 სიმულაციის შემთხვევაში.



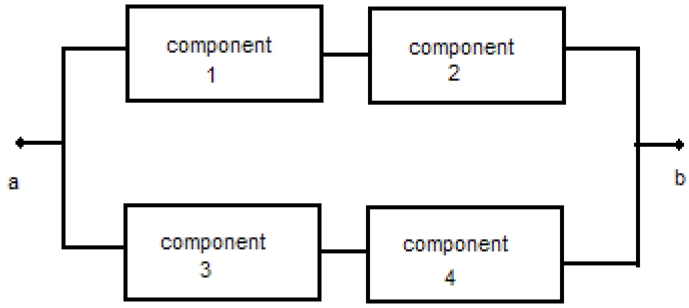
ნახ. 5.18



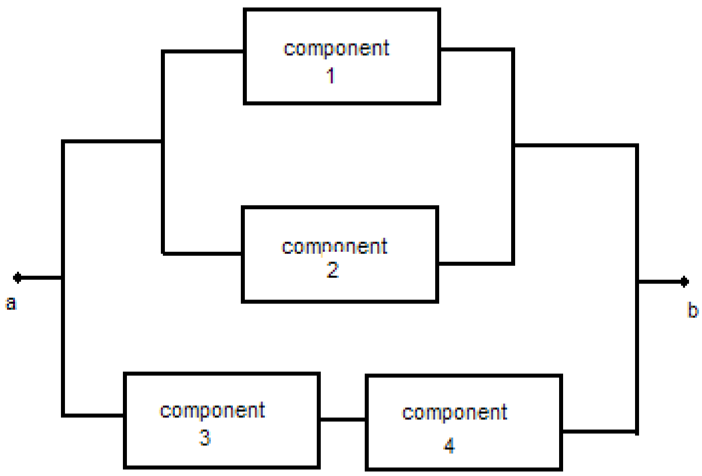
ნახ. 5.19



ნახ. 5.20



ნახ. 5.21



ნახ. 5.22

13. დაწერეთ პროგრამა ნახ. 5.21 მოცემული სისტემის მოდელირებისათვის, თუ 1და 2 კომპონენტის საიმედოობაა 0.8, მესამის – 0.95. დაბეჭდეთ საიმედოობა 5000 სიმულაციის შემთხვევაში.
14. დაწერეთ პროგრამა ნახ. 5.22 მოცემული სისტემის მოდელირებისათვის, კომპონენტების საიმედოობაა 0.95. დაბეჭდეთ საიმედოობა 5000 სიმულაციის შემთხვევაში.

ელექტროსადგურის მიერ გამომუშავებული ენერგია. 8 კვირის განმავლობაში გროვდებოდა მონაცემები ელექტროსადგურის მიერ გამომუშავებული ენერგიის შესახებ. მონაცემები

ჩაიწერა ფაილში plant.dat. თითოეული სტრიქონი შეიცავს მონაცემებს 1 კვირისათვის. 1, 2, 3 , ... , 7 დღეს გამომუშავებულ ენერგიას მეგავატებში.

15. დაწერეთ პროგრამა, რომელიც წაიკითხავს მონაცემებს ფაილიდან plant.dat, და მოგვცემს იმ დღეების რაოდენობას, როცა ელექტროსადგურმა გამოიმუშავა საშუალოზე მეტი ენერგია. უნდა მივიღოთ კვირის ნომერი და დღის ნომერი თითოეული ასეთი დღისათვის. ამასთან ერთად დაგვიბეჭდოს ამ რვა კვირის განმავლობაში საშუალოდ ერთ დღეში გამომუშავებული ელექტროენერგია მეგავატებში
16. დაწერეთ პროგრამა, რომელიც წაიკითხავს მონაცემებს ფაილიდან plant. dat, და დაბეჭდავს იმ კვირის და დღის ნომერს, როცა ყველაზე მეტი და ყველაზე მცირე ელექტროენერგია იქნა გამომუშავებული.
17. დაწერეთ პროგრამა, რომელიც წაიკითხავს მონაცემებს ფაილიდან plant. dat, და დაბეჭდავს ერთ დღეში გამომუშავებულ საშუალო ენერგიეს, ყოველი კვირისათვის ცალკ-ცალკე.