Miloš Stanisavljević

Šum vakuumske diode kao generator slučajnih brojeva

Šum vakuumske diode korišćen je za generisanje slučajnih brojeva. Softverskom obradom šuma poboljšane su osobine ulaznog signala i obezbeđen veći opseg random brojeva. Tako je dobijen random generator koji ima slične osobine kao random generatori kod kojih se pseudoslučajni brojevi generišu isključivo softverskim putem. Međutim, ovaj generator ima znatno duži period generisane sekvence brojeva, pa je zato pogodan za korišćenje u Monte Karlo simulacijama.

Uvod

Spontane električne fluktuacije koje su prisutne u električnim i elektronskim kolima nazivaju se šum. Šum obično negativno utiče na osobine kola. Međutim, može biti i koristan, tj. može se koristiti kao generator slučajnih fluktuacija, koje se upotrebljevaju pri različitim električnim merenjima. Takođe se može koristiti i kao generator slučajnih brojeva. Na izlazu pojačavača sa velikim pojačanjem, kome na ulaz nije doveden nikakav signal, moći će se npr. registrovati fluktuacije napona kao na slici 1.

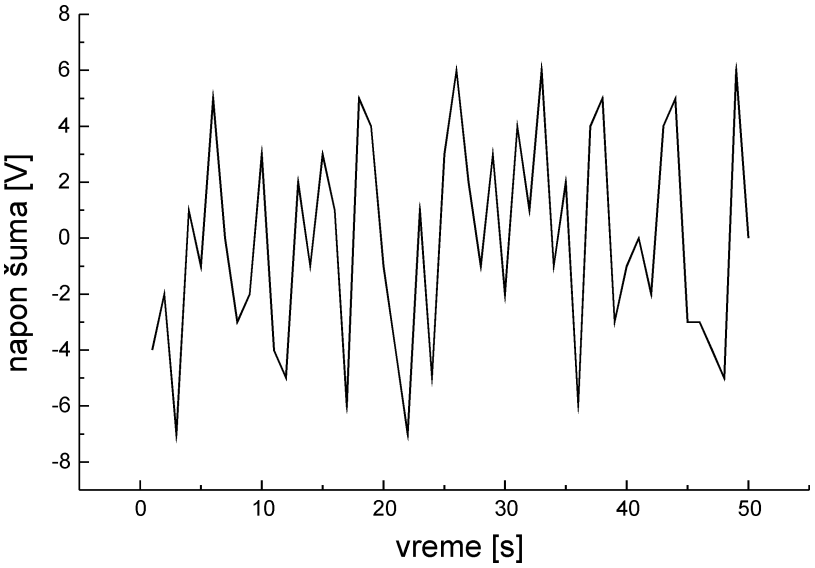
S obzirom na izvor nastajanja, šum se deli u dve grupe (Zovko-Cimlar 1987):

a) šum koji nastaje *izvan mernog sistema* (kosmički, atmosferski, elektrostatički, od aparata i uređaja itd.) i
b) šum koji se pojavljuje *unutar sistema* (termički šum, šum u vakuumskim elemenata, šum poluprovodničkih komponenti, kontaktni šum i sl.).

U odnosu na fizički proces u kome nastaje šum može biti:

a) *Džonsonov* (Johnson) ili *termalni* šum – izazvan nagomilavanjem slobodnih elektrona, koji ubrzani termičkim putem stvaraju fluktuirajuće struje u otporničkim elementima.
b) š*um sačme* (inače najčešća vrsta šuma) – nastaje u fluktuacijama anodne struje u vakuumskim elementima ili usled spontanog kre-

Miloš Stanisavljević (1979), Smederevo, Ivana Gundulića 7/81, učenik 4. razreda Gimnazije u Smederevu.



Slika 1.
Grafik zavisnosti napona šuma od vremena.

Figure 1.
Noise voltage vs. time.

tanja individualnih nosilaca naelektrisanja na granicama P-N spojeva kod poluprovodnika

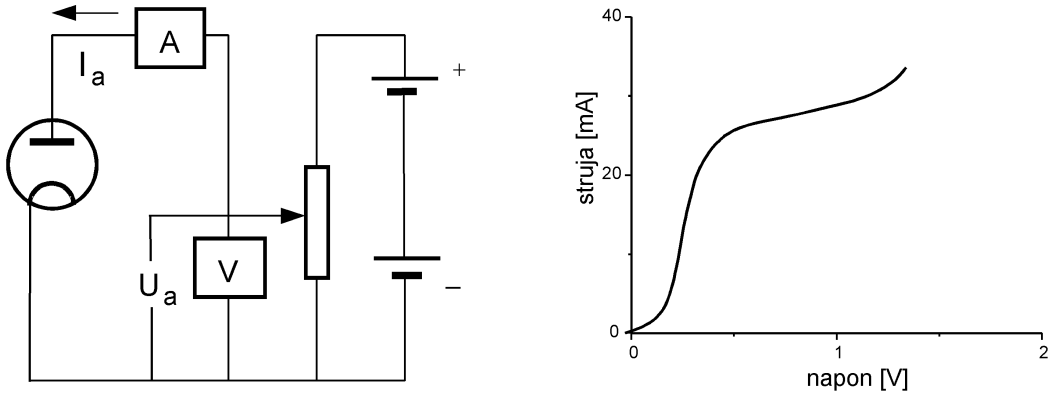
c) *generaciono-rekombinantni* šum – nastaje u poluprovodničkim elementima fluktuacijama u rekombinaciji strujnih nosilaca
d) 1/*f* – *fliker šum* – nastaje zbog varijacija u struji katode u vakuumskim cevima ili fluktuacijama nosilaca naelektrisanja koji uleću u provodnu zonu
e) *fotonski šum* – nastaje zbog fluktuacija odnosno kvantne prirode elektromagnetnih talasa kojima se pobuđuju fotodetektorski elementi (McCluney 1994).

Dioda

Vakuumska dioda je najprostiji elektronski element koja sadrži samo dve elektrode, katodu i anodu. Uloga katode je da emituje elektrone, a anode da ih prima. Na slici 2a možemo videti prosto kolo kojim se može ustanoviti zavisnost anodne struje I_a, od napona U_a, koja je prikazana na slici 2b (Raković 1987). Pored vakuumske postoji i poluprovodnička dioda koja ima znatno složeniju strukturu.

Uprkos prostoj strukturi cev dioda ima široku primenu, ali i tu negativnu osobinu da ima veliki šum. Međutim, zbog relativno velike vrednosti tog šuma ona se može iskoristiti kao generator slučajnih brojeva koji neće koristiti softversko generisanje, već podatke dobijene očitavanjem vrednosti struje diode, tzv. 'hardversko generisanje'.

Dioda koja radi u području zasićenja (vrednost napona između 0.5 i 0.8 V, a anodna struja vrlo slabo raste) ponaša se unutar velikog frekven-



Slika 2.
a) Kolo sa diodom
b) Karakteristika diode – zavisnost struje od napona diode.

Figure 2.
a) Diode circuit
b) Voltage characteristic of diode.

cijskog područja kao strujni izvor šuma s jednim beskonačno velikim unutrašnjim otporom (Raković 1987). U jednom relativno uskom frekventnom području koje se posmatra f, srednja vrednost kvadrata struje šuma (odnosno fluktuacije anodne struje pri efektu sačme) data je Šotkijevom (Schottky) formulom:

$$i_S^2 = 2\,q\,I_a\,\Delta f, \tag{1}$$

gde je q elementarno naelektrisanje elektrona (q = 1.60210^{-19}C), a I_a je struja zasićenja koja teče kroz diodu (Zovko-Cimlar 1987).

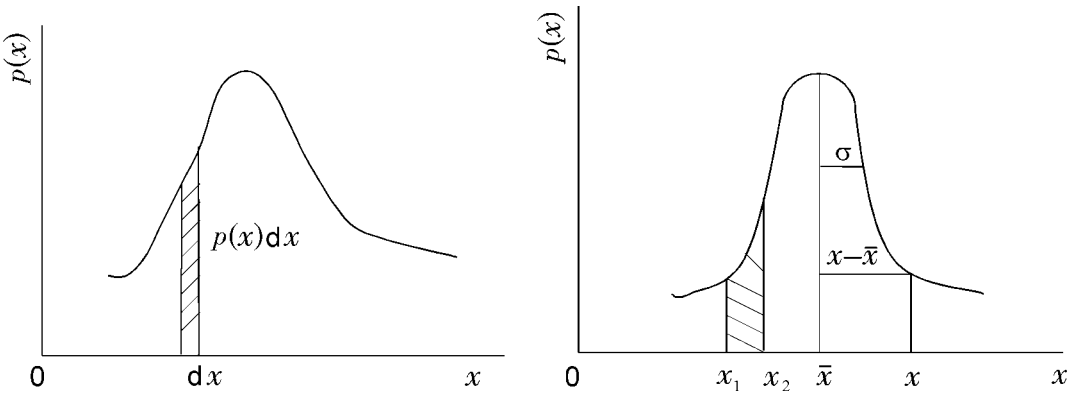
Funkcije raspodele

Pri posmatranju i analizi slučajnih procesa opravdano je pretpostaviti da promenljiva X može da ima bilo kakvu vrednost. Na primer, u jednom eksperimentu posle niza merenja promenljive X dobijene su sledeće vrednosti x_1, x_2, x_3, … x_i, kojima odgovaraju verovatnoće pojavljivanja $P(x_1)$, $P(x_2)$, $P(x_3)$, … $P(x_i)$. U kontinuiranim slučajnim procesima, kao što su fluktuacije napona šuma, slučajna promenljiva X može imati kontinuirani niz slučajnih vrednosti. U tom slučaju upotrebljava se izraz – funkcija gustine raspodele verovatnoće $p(x)$ kao prvi izvod funkcije raspodele:

$$p(x) = \lim_{\Delta x \to 0} \frac{P\,(x \le X \le x + \Delta x)}{\Delta x} \tag{2}$$

Dati izraz se normira na sledeći način:

$$\int_{-\infty}^{\infty} p(x)\,dx = 1 \tag{3}$$



Slika 3.
a) Funkcija gustine verovatnoće
b) Kriva Gausove raspodele.

Figure 3.
a) Density of probability function.
b) Gaussian.

Prema tome, verovatnoća da slučajna promenljiva X uzima vrednosti u nekoj okolini tačke x je proporcionalna gustini verovatnoće u tački x (Merkle i Vasić 1995). Funkcija gustine verovatnoće je prikazana na slici 3, gde je $p(x)$ je funkcija gustine verovatnoće, a $p(x)\,\mathrm{d}x$ je gustina verovatnoće (šrafirana oblast na grafiku); dx predstavlja širinu šrafirane oblasti na slici i po definiciji je infinitezimalna vrednost. Primena funkcije gustine raspodele dovode do raznih tipova raspodela. Najpoznatije su: Gausova (normalna), binomna, Poasonova i uniformna.

Generisanje slučajnih brojeva

Svi softverski generatori pseudoslučajnih brojeva zasnovani su na izvesnim matematičkim algoritmima koji su periodični i sastoje se iz nizova koji se ponavljaju. Kao takvi ovi brojevi su pseudoslučajni tj., odlikuju se izvesnom slučajnošću, ali su uvek u manjoj ili većoj meri peroidični.

Za različite primene ovih brojeva imamo nekoliko kriterijuma valjanosti generatora.

Jedan od zahteva je različitost sekvenci brojeva koje random generator generiše. To znači da bilo koje dve n-torke koje random generator generiše moraju biti nezavisne jedne od drugih. Naravno, ovaj uslov je limitiran konačnošću memorije računara, kod softverskog generisanja, dok kod 'hardverskog' generatora (koji koristi šum) takvih ograničenja nema.

Drugi zahtev je da random generator ima dovoljno dug period (idealno bi bilo kada se sekvenca generatora ne bi uopšte ponavljala, što u praksi znači da se sekvenca random generatora sme ponoviti samo posle generisanja veoma velikog broja random brojeva) Možemo uzeti da je ovaj uslov ispunjen ako je period duži od broja koraka u simulaciji (npr. Monte Karlo).

Treći i četvrti kriterijum se odnose na distribucije slučajnih brojeva i njihovih razlika. Najčešće se koriste random generatori sa pretpostavlje-

nom raspodelom $U(0,1)$ (uniformna raspodela na intervalu od 0 do 1, raspodela koja za sve frekvencije pojavljivanja neke promenljive X daje istu vrednost, odnosno gde su sve vrednosti promenljive X jednako verovatne). Sličnost sa njom je upravo pomenuti treći kriterijum. Distribucija razlika slučajnih brojeva mora imati oblik:

$$f(z) = f(x-y) = \int_0^1 f(x)\, f(x-z)\, dx = z\,|1-z\,|\, U(-1,1) \tag{4}$$

Osim ovih osnovnih kriterijuma valjanosti svaka konkretna primena može zahtevati i dodatne, npr. odgovarajuća raspodela razlike razlika.

Za testiranje hipoteze o raspodeli koristi se χ^2 funkcija. Ona pokazuje koliko je odstupanje dobijene funkcije, odnosno histograma od stvarne raspodele. Ovo praktično znači da neki random generator možemo smatrati dovoljno dobrim na nivou poverenja $1-\alpha$ ako χ^2 leži izmeću kvantila raspodele $\varepsilon_{\alpha/2}$ i $\varepsilon_{1-\alpha/2}$. Ne primer random generator možemo smatrati dobrim na nivou poverenja od 95% ako χ^2 leži između kvantila raspodele $\chi^2_{0.975}$ i $\chi^2_{0.025}$ (Merkle i Vasić 1995).

Metod

Opis korišćene aparature

1. Vakuumska dioda
2. Multimetar VOLTCRAFT HC-505ODB
3. Unimer velike tačnosti sa mogućnošću povezivanja na računar (FLUKE 45)
4. Računar PC486DX4 / 100 MHz ili brži, sa 16Mb RAM-a i I/O karticom (programom F45COM i softverskim paketima ORIGIN, MATLAB, STATISTICA, BP i QBASIC)
5. Protoboard
6. Nekoliko otpornika, otpora u rasponu od 5-50 (tolerancije 2-5)
7. Nekoliko provodnika
8. PIN fotodida FD5 N-tipa

Postupak rada i način merenja

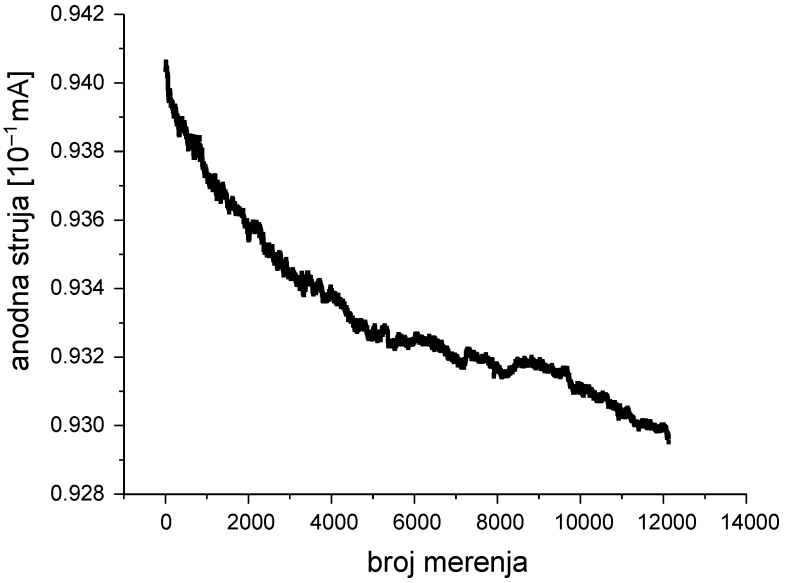
Kao izvor šuma uzeta vakumska dioda, jer je poznato da je upravo kod nje odnos signal/šum najmanji.

Za merenje je sklopljena aparatura kao na slici 2, a na mestu ampermetra bio je priključen unimer (FLUKE 45). Dioda je priključena na izvor napona od 0.7 V (jer se za tu vrednost napona vakumska dioda nalazi u stanju zasićenja i daje najveći šum). Kao izvor je korišćen generator napona. Merena je anodna struja u mA sa tačnošću od 0.0001 mA (100 nA).

Greška je takođe bila 100 nA, što je suviše mala vrednost, u odnosu na izmerenu vrednost napona, da bi se uzela u razmatranje. Vrednosti su očitavane 2.5 puta u sekundi. Uz pomoć odgovarajućeg programa F45COM vrednosti su snimane u fajl. Izvršeno je nekoliko serija od po 12000 merenja. Vrednosti su zatim prenete u ORIGIN gde su dalje obrađivane.

Obrada rezultata

Na slici 4 dat je grafik zavisnosti anodne struje od broja merenja

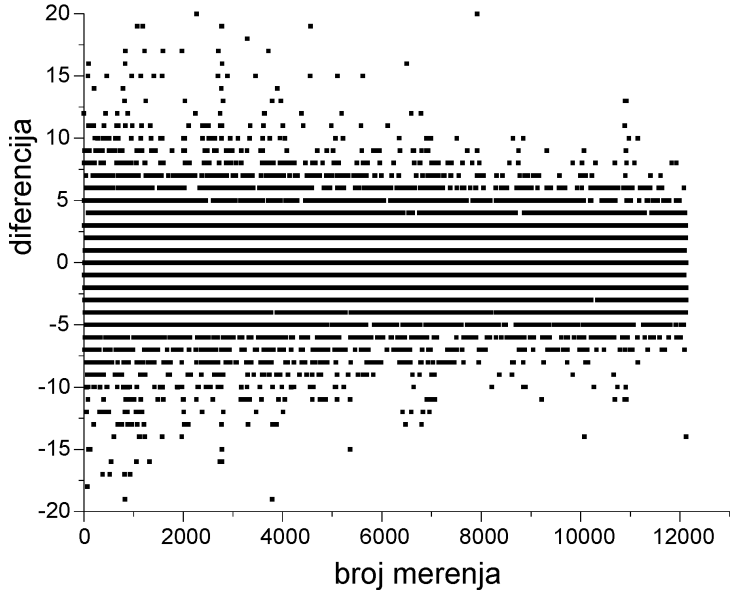


Slika 4.
Zavisnost anodne struje od broja merenja.

Figure 4.
Anode current vs. number of measurements.

Na slici 4 primećena je zanimljiva pojava opadanja izmerene vrednosti struje u toku dugog vremenskog intervala. Vrednost je opala nekih 0.5-0.8 za 1sat merenja. Uzrok ove pojave je povećanju anodne disicipacije, odnosno o konstantno zagrevanju anode, usled čega je došlo do intenzivnijeg sudaranja elektrona koji padaju na nju. Zato se i anodna struja vremenom smanjivala.

Vrednosti koje čine ovu krivu (i koje se stalno smanjuju) očigledno nemaju Gausovu raspodelu (kakvu bi trebalo da ima sačma šum), već im je raspodela potpuno nepravilna, te ju je nemoguće pretvoriti u uniformnu, kakvu imaju slučajni brojevi. Zato nisu posmatrane apsolutne vrednosti izmerene jačine struje, već diferencije, tj razlike svaka dva susedna člana početnog niza. Tako je dobijena Gausova raspodela koja je pogodna za pretvaranje u uniformnu. Međutim, ova raspodela ima isuviše veliki χ^2 (1976.1), tako da zaključujemo da funkcija iako liči, ipak nije Gausova raspodela. Dva glavna uzroka tako velikog χ^2 su:



Slika 5.
Diferencijacije u vremenu – grafik zavisnosti diferencija od broja merenja.

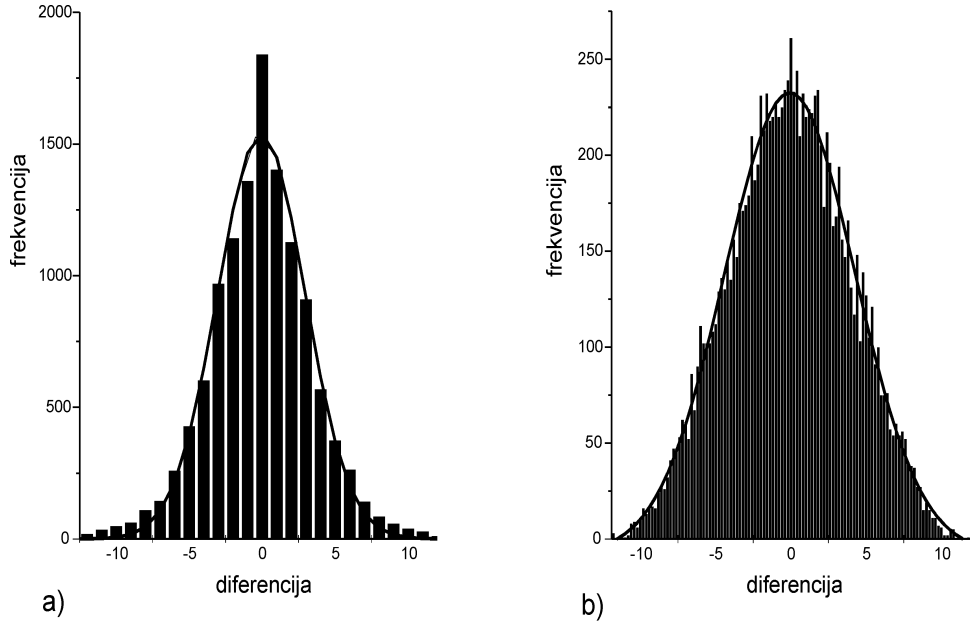
Figure 5.
Differences of values from the figure 4 – differences vs. number of measurements.

1) Neujednačen signal u toku vremena, što se uočava na slici 5, gde možemo primetiti da se u pojedinim opsezima neke vrednosti javljaju češće no u drugim.

Analizom podataka sa prethodnog grafika došlo se do zaključka da je standardna devijacija za većinu uzimanih pojedinačnih opsega znatno veća od srednje standardne devijacije za svih 12000 merenja, što je jasan uzrok velikog $\chi^{2.}$

2) Suviše širok interval diskretnih vrednosti (što se vidi na slici 6a, koja inače predstavlja fitovani, po Gausu, histogram raspodele za date vrednosti diferencija).

Zato je potrebno izvršiti softversku korekciju rezultata dobijenih merenjem. Ovo vrši program "Random generator" napisan u Pascal-u čiji se source nalazi u dodatku. Program radi po sledećem postupku: nalazi srednju standardnu devijaciju za svih 12000 diferencija; zatim nalazi standardnu devijaciju za grupe od po 10 diferencija i svaki elemenat pojedinačne grupe množi sa koeficijentom σ/σ_i gde je σ_i standardna devijacija *i*-te grupe. Pri tom se na svakih 100 koraka proverava srednja standardna devijacija σ. Na ovaj način se postiže ujednačavanje standardne devijacije i svake grupe, a samim tim nestaju nepravilnost koje su se mogle uočiti na slici 5. Takođe dobijen je histogram raspodele koji je imao mnogo uži interval diskretnih vrednosti (slika 6b), zato što sada nema više samo celobrojnih, već su vrednosti realne, što mnogo više odgovara i realnom izgledu signala (koji je po prirodi analogan). Zato je sada moguće dobiti random generator koji daje mnogo više različitih slučajnih brojeva.



Slika 6 (dole). Histogram raspodele diferencija pre (a) i nakon softverske obrade (b).

Figure 6 (bellow). Distribution of differences:
a) before software analysis
b) after software analysis.

Hi-kvadrat datog histograma je mnogo manji, nego hi-kvadrat histograma na slici 6a. Za gornji histogram broj stepeni slobode je 118, pa bi po Pirsonovoj statistici za nivo poverenja od 95% njegov χ^2 trebalo da leži između kvantila raspodele $\chi^2_{0.975}$=89.83 i $\chi^2_{0.025}$=149.96, a kako je χ^2 za dati histogram 103.38 to je ovaj uslov očigledno ispunjen, pa se može sa dovoljo sigurnosti zaključiti da je data raspodela zaista Gausova.

Dalje je potrebno Gausovu raspodelu pretvoriti u uniformnu, kakvog su oblika slučajni brojevi. Ovo takođe radi program *Random generator* (za source vidi prilog na kraju ovog rada) na sledeći način: prvo se određuje preciznost generatora (ulazni podatak *n*), tj. koliko će se različitih slučajnih brojeva u rasponu od 0 do 1 generisati; zatim se površina ispod Gausove krive na slici 10 deli na odgovarajući broj jednakih površina u zavisnosti od toga koliko se različitih slučajnih brojeva želi dobiti (program dozvoljava od 2 do 500). Drugim rečima, program računa poznatu eror funkciju (integralna funkcija koja računa sumu ispod krive funkcije do neke određene vrednosti) i određuje granice za intervale u koje 'upadaju' ulazni podaci, odnosno diferencije. Na ovaj način se Gausova raspodela preslikava u uniformnu.

Program učitava podatke dobijene iz programa F45COM iz ulazne datoteke, a smešta ih u izlaznu datoteku ili ih prosleđuje drugom programu, u zavisnosti od namene.

Napomena: Algoritam programa omogućava pretvaranje i iz bilo koje druge raspodele u uniformnu, tj. nezavisan je od oblika ulazne raspodele.

Analiza dobijenih podataka

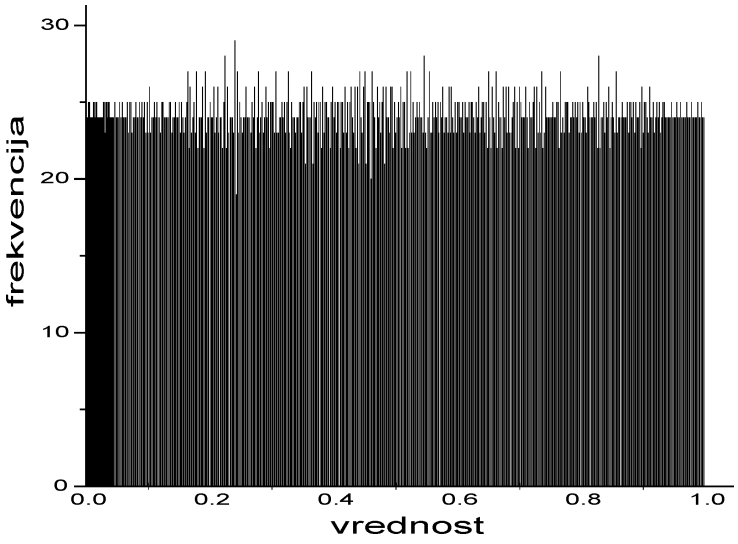
Izvršena je analiza dobijenih podataka za sledeće vrednosti ulaznog *n*: 2, 10, 100 i 500. Pritom su dobijeni sledeći rezultati (tabela 1):

Tabela 1. Dobijeni rzultati***

n	$\chi^2_{0.975}$	$\chi^2_{0.025}$	χ^2	τ_{12000} [s]	τ_1 [ms]
2	0.00	5.02	0.47	4.0	0.33
10	2.70	19.02	11.72	5.5	0.46
100	73.36	128.42	79.67	29.0	2.42
500	439.00	562.79	492.10	142.0	11.83

τ_{12000} – vreme potrebno za 12.000 iteracija
τ_1 – vreme po jednoj iteraciji

Vidimo da je u svim slučajevima χ^2 u odgovarajućim granicama za nivo poverenja od 95%, te je očigledno da je ovaj random generator dovoljno dobar za sve vrednosti n za koje radi (od 2 do 500). Histogram raspodele za poslednji slučaj, kada je n = 500 može se videti na slici 7.



Slika 7. Histogram raspodele dobijenih vrednosti po frekvencijama za n = 500.

Figure 7. Distribution of yielded values when number of measurements (n) is equal 500.

Pored ovih analiza izvešena je i uporedna analiza ovog i paskalovog random generatora za N = 10000 vrednosti i n = 50. Broj stepeni slobode je 49, pa imamo vrednosti $\chi^2_{0.975}$ = 31.5 i $\chi^2_{0.025}$ = 70.2. "Hardverski" generator ima χ^2 = 33.83, a paskalov generator χ^2 = 59. Očigledno oba generatora možemo uzeti kao dovoljno dobra na osnovu ovog testa. Što se

tiče histograma razlika random brojeva, tu je raspodela paskalovog generatora nešto pravilnija, ali je očigledno da i raspodela hardverskog generatora ima traženi oblik

$$f(z) = |1 - z|\, U(-1, 1)$$

na osnovu jednačine (4).

Što se periodičnosti tiče paskal generator ima period manji ili jednak 2^{39}, a u konkretnom slučaju oko 2^{24}, dok je period 'hardverskog' generatora znatno veći.

Zaključak

Na kraju možemo zaključiti da generator koji koristi softverski obrađeni šum vakuumske diode ima ukupno gledano, neznatno lošije osobine od paskalovog random generatora, ali da je pogodaniji za korišćenje u Monte Karlo simulacijama zbog većeg perioda.

Literatura

Lukatela, G. 1991. *Statistička teorija telekomunikacija i informacija*, 1. deo. Beograd: Građevinska knjiga.

McCluney, R. 1994. *Radiometry and photometry*. Boston: Artech House.

Merkle, M.J., Vasić, P.M. 1995. *Verovatnoća i statistika*. Beograd: ETF.

Raković, B. 1987. *Elektronika*, I. Beograd: Naučna knjiga.

Zovko-Cimlar, Branka. 1987. *Šum u radiokomunikacijama*. Zagreb: Školska knjiga.

Miloš Stanisavljević

The Vacuum Diode Noise as the Random Generator

A noise i.e. spontaneous current and voltage fluctuation in electrical circuits tends to be random. This enables noise to be used for generating random numbers. A vacuum diode was chosen as a noise source because it's signal to noise ratio is the lowest. In order to reach saturation level, where the noise is the highest, diode was connected to 0.7 V DC (Figure 2). The anode current was measured with an accuracy of 100 nA. It was sampled 2.5 times per second. Using the F45COM software package the data was recorded in a file. There has been a few series of measures. The differences between two successive measured values were taken as input

data for the software processing. A distribution similar to a Gaussian was obtained in that way (Figure 6a).

The acquired data was processed further to remove any unwanted properties of the input signal. The equalization of the signal level and narrowing of the discrete value interval to obtain real number instead of whole number values were achieved by equalizing the standard deviation during the entire duration of the input signal. The resulting Gaussian distribution (Figure 6b) was then converted to the uniform by using software procedure. The obtained data have been analyzed by Pearson's χ^2 statistics. All results were within the limit values of the distribution quintile at 95% level of confidence, according to criteria of uniformity and distribution of differences of successive values. The described random number generator was slightly less effective then algorithm used in the Pascal programming language according to comparative tests. Its advantage over only software method, however, is the ability to generate much longer sequences of pseudo random numbers. This makes it especially useful in Monte Carlo simulations where it is necessary for the sequence period to be longer than the number of iterations.

Prilog: source pascal programa *Random generator*

```
program Random_generator;

{$N+}

uses crt;
const stepsigme = 10;
      maxniza = 12200;
      maxstep = 6000;
      maxtacnost = 1000;
      proracun_sigme = 100;
      konstanta = 7.9;
type realniz = array [1..maxniza] of single;
     podelaniz = array [1..maxtacnost] of real;
var niz: realniz;
    podela_niz: podelaniz;
   brojac, i, j, k, l, m, n, count, frekvencija, max:
integer;
    sigma_srednje, srednje, sigma, con: real;
    sigma_pomocno, suma, maxelem, steper: real;
    datotekaI, datotekaO: text;

begin
   clrscr;
   brojac:=0;
   maxelem:=0;
   writeln ('Unesite koliko razlicitih rnd brojeva');
   write ('(od 2 do 500) zelite da se generise: ');
   readln (n);
   assign (datotekaI, 'c:\input.txt');
   assign (datotekaO, 'c:\output.txt');
   reset (datotekaI);
   rewrite (datotekaO);
   while not eof (datotekaI) do
   begin
      inc (brojac);
      read (datotekaI, niz[brojac]);
      if niz[brojac]=0 then
         niz[brojac]:=niz[brojac-1]/konstanta;
   end;
   for i:=1 to brojac-stepsigme+1 do
   begin
      if ((i mod proracun_sigme)=0) or (i=1) then
      begin
         srednje:=0;
         sigma_pomocno:=0;
         for j:=1 to brojac do
             srednje:=srednje+niz[j];
         for j:=1 to brojac do
                          sigma_pomocno:=sigma_po-
mocno+sqr(niz[j]-srednje/brojac);
         sigma_srednje:=sqrt(sigma_pomocno/brojac);
```

```
      end;
      srednje:=0;
      sigma_pomocno:=0;
      for k:=i to i+stepsigme-1 do
          srednje:=srednje+niz[k];
      for l:=i to i+stepsigme-1 do
            sigma_pomocno:=sigma_pomocno+sqr(niz[l]-
srednje/stepsigme);
      sigma:=sqrt(sigma_pomocno/stepsigme);
      for m:=i to i+stepsigme-1 do
          niz[m]:=niz[m]*(sigma_srednje/sigma);
                   if   abs(niz[i])=maxelem   then
maxelem:=abs(niz[i]);
   end;
   suma:=0;
   count:=0;
   steper:=1/n;
   max:=trunc(maxelem+1);
   for i:=1 to n*2*max do
   begin
      frekvencija:=0;
      for j:=1 to brojac do
          if (niz[j]ax+i/n) and
             (niz[j]-max+(i-1)/n) then
             inc(frekvencija);
      suma:=suma+frekvencija/brojac;
      if (suma=steper) and (suma-frekvencija/brojac
) then
      begin
         inc(count);
            if (suma-steper)(steper-suma+frekven-
cija/brojac) then
            podela_niz[count]:=-max+i/n
         else
            podela_niz[count]:=-max+(i-1)/n;
         steper:=(round(steper*n)+1)/n;
      end;
      if steper=1 then
      begin
         i:=n*2*max;
         inc(count);
         podela_niz[count]:=maxelem;
      end;
   end;
   for i:=1 to brojac do
       for j:=1 to count do
           if niz[i]a_niz[j] then
           begin
              writeln (datotekaO, (j-1)/count:8:6);
              j:=count;
           end;
end.
```