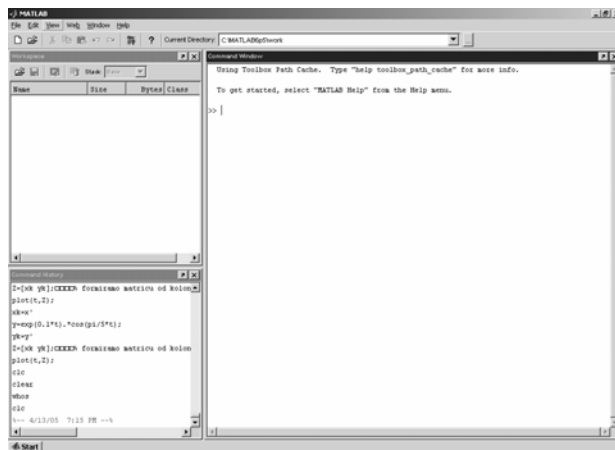


### V 1.1. Kratak uvod u Matlab i analizu osnovnih osobina signala i sistema

Programski paket Matlab omogućava interaktivan i programski rad. U interaktivnom radu naredbe se zadaju u promptu komandnog prozora Matlaba i interpretiraju odmah po zadavanju. Prethodno otkucane naredbe ostaju u istoriji i mogu se vratiti pritiskom na kursorski taster  $\uparrow$  (gore), editovati i ponovo pokrenuti. Programi se takođe pozivaju iz komandnog prompta navođenjem imena i interpretiraju liniju po liniju.



Sl. 1. Matlab okruženje

#### (1) Osnovne strukture podataka

Razmatranje struktura podataka u Matlabu izvešćemo u interaktivnom radu, uz pripadajuća objašnjenja, a zatim kreirati programe, sve u cilju upoznavanja sa Matlab okruženjem i realizacijom zadataka u vezi analize osnovnih osobina diskretnih i analognih signala i sistema.

Sve promenljive u Matlabu su matrice:

```
>> a=1
```

```
a =
    1
```

```
>> whos
```

Name	Size	Bytes	Class
a	1x1	8	double array

U programskom prostoru Matlabu je definisana promenljiva  $a$  kao matrica dimenzija  $1 \times 1$  brojeva dvostruke preciznosti. Naredba *whos* izlistava sve definisane promenljive u Matlabovom radnom okruženju.

Stringovi su takođe matrice – vrste karaktera:

```
>> b='string - vektor karakter vrednosti'
```

```
b =
string - vektor karakter vrednosti
```

```
>> whos
```

Name	Size	Bytes	Class
a	1x1	8	double array
b	1x34	68	char array

Pristup elementima matrice:

```
>> b(2)
ans =
t

>> b(10:15)
ans =
vektor

>> A=[1 2;3 4]
A =
     1     2
     3     4

>> A(2,1)
ans =
     3

>> A(1,2)
ans =
     2

>> A(2,:)
ans =
     3     4

>> A(:,1)
ans =
     1
     3
```

Nema posebnog definisanja tipa promenljive; promenljive mogu narednom dodelom da promene i tip i sadržaj, a matrice se automatski proširuju:

```
>> A(3,4)=9

A =
     1     2     0     0
     3     4     0     0
     0     0     0     9

>> whos A

Name      Size      Bytes  Class
A         3x4         96   double array

>> A = 'string'

A =
string

>> whos A

Name      Size      Bytes  Class
A         1x6         12   char array
```

Komentari u Matlabu počinju sa %, sve nakon procentnog znaka se ignoriše:

```
>> c=[b ' i moze se jednostavno prosirivati'] % kreiranje slozenog stringa

c =
string = vektor karakter vrednosti i moze se jednostavno prosirivati
```

Promenljive se mogu izbrisati naredbom `clear`, npr. `clear a b` briše promenljive `a` i `b` iz Matlabovog radnog prostora. Samo `clear` briše sve trenutno definisane promenljive.

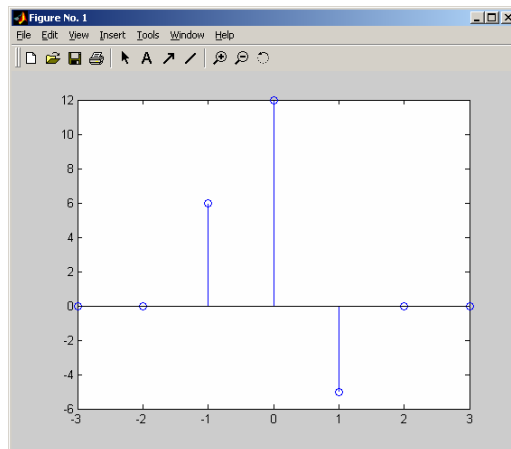
**(2) Grafički prikaz diskretnih i analognih signala**

Neka je zadat signal  $x(n) = \begin{cases} 0, & n < -1 \\ 6, & n = -1 \\ 12, & n = 0 \\ -5, & n = 1 \\ 0, & n > 1 \end{cases}$

Za crtanje relevantnih odbiraka ovog signala potrebno je definisati vektore  $n$  i  $x$  i iskoristiti naredbu *stem*:

```
>> n=[-3 -2 -1 0 1 2 3]; % znak ; na kraju komande ukida Matlabov odziv
>> x=[0 0 6 12 -5 0 0];
>> stem(n,x)
```

Po zadavanju komande *stem*, otvara se grafički prozor Figure No. 1, kao na slici 2.



Sl. 2. Grafički prozor dobijen izvršenjem naredbe *stem*

Neka je zadat signal  $x(n) = \exp(0.1n)$  i potrebno ga je nacrtati u opsegu  $n \in [-10, 10]$ . Način da se automatski generiše vektor sa početnom vrednošću, korakom i završnom vrednošću je:

$$\text{vektor} = \text{početna\_vrednost} : \text{korak} : \text{završna\_vrednost}$$

Korak može da se izostavi, podrazumevani korak je 1.

```
>> n=-10:10
```

```
n =
```

```
Columns 1 through 11
```

```
-10    -9    -8    -7    -6    -5    -4    -3    -2    -1     0
```

```
Columns 12 through 21
```

```
1      2      3      4      5      6      7      8      9     10
```

Matlab ugrađenim komandama podržava matrične argumente:

```
>> x=exp(0.1*n)
```

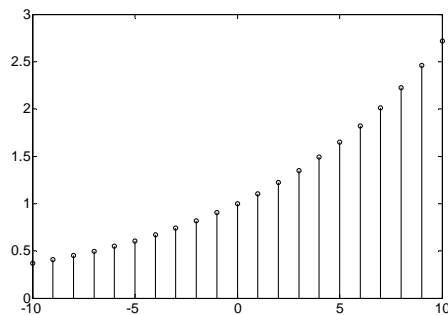
```
x =
Columns 1 through 7
    0.3679    0.4066    0.4493    0.4966    0.5488    0.6065    0.6703

Columns 8 through 14
    0.7408    0.8187    0.9048    1.0000    1.1052    1.2214    1.3499

Columns 15 through 21
    1.4918    1.6487    1.8221    2.0138    2.2255    2.4596    2.7183
```

Sada možemo iscrtati signal:

```
>> stem(n,x)
```



Sl. 4. Grafički prikaz odbiraka signala  $x(n) = \exp(0.1n)$

Ako pokušamo da definišemo složeniji signal  $x(n) = \exp(0.1n)\sin(n\pi/5)$ :

```
>> x=exp(0.1*n)*sin(n*pi/5)
??? Error using ==> *
Inner matrix dimensions must agree.
```

dobijamo poruku o grešci pri množenju. Setimo se, sve operacije koje Matlab izvodi su matrične, pa su matrice - rezultati funkcija *exp* i *sin* obe dimenzija 1x21, pa nisu kompatibilne za matrično množenje. Nama je, pak, potrebno množenje element po element, što se označava dodavanjem interpunkcijske tačke ispred operacije:

```
>> x=exp(0.1*n).*sin(n*pi/5)
```

```
x =

Columns 1 through 7
    0.0000    0.2390    0.4273    0.4723    0.3226   -0.0000   -0.3940

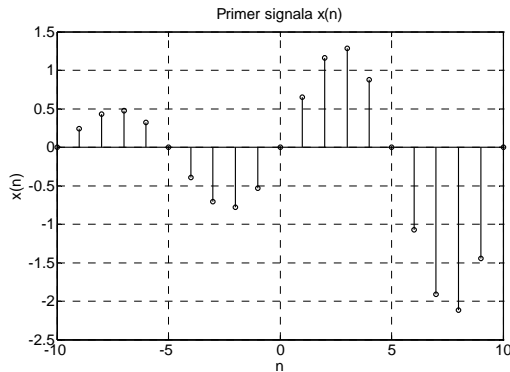
Columns 8 through 14
   -0.7046   -0.7787   -0.5319         0    0.6496    1.1616    1.2838

Columns 15 through 21
    0.8769    0.0000   -1.0710   -1.9152   -2.1166   -1.4457   -0.0000
```

“Kandidati za tačnicu” su operacije množenja \*, deljenja / i stepenovanja ^.

Slikama se mogu dodati oznake po x osi, y osi, naslov, staviti mreža:

```
>> stem(n,x)
>> xlabel('n');
>> ylabel('x(n)');
>> title('Primer signala x(n)');
>> grid
```

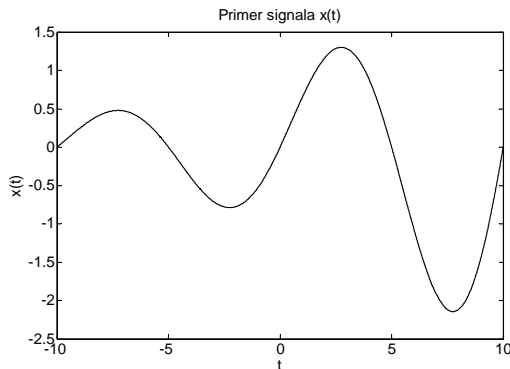


Sl. 5. Grafički prikaz odbiraka signala  $x(n) = \exp(0.1n) \sin(n\pi/5)$

Analogne signale možemo iscrtavati tako što napravimo finu podelu vremenske ose i koristimo umesto *stem* naredbu *plot*:

```
>> t=-10:0.01:10;
>> x=exp(0.1*t).*sin(pi/5*t);
>> plot(t,x);
>> xlabel('t');
>> ylabel('x(t)');
>> title('Primer signala x(t)');
```

Rezultat izvršavanja navedenog niza Matlab instrukcija je grafik predstavljen slikom 6.

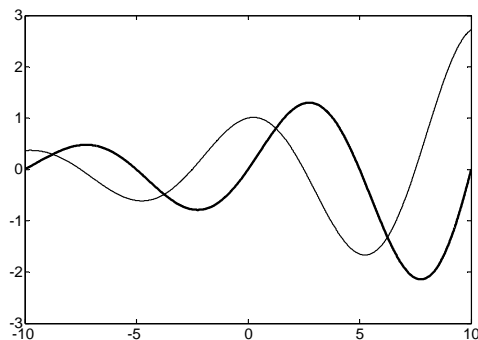


Sl. 6. Grafički prikaz signala  $x(t) = \exp(0.1t) \sin(t\pi/5)$

Matlab ima veoma dobro urađen help sistem (Help meni u Matlab komandnom prozoru vodi u html sistem help-a). Međutim, i u komandnom promptu, korišćenjem naredbe *help* moguće je dobiti informacije o svim grupama Matlab-ovih funkcija i struktura. Ako je poznato ime grupe funkcija, npr. `matlab\elfun` – elementarne matematike funkcije, ili ime funkcije za koje nam treba pomoć, oni se

moгу navesti i dobiti u komandnom prozoru tekstualna pomoć vezana za datu grupu, odnosno funkciju. Npr. ako želimo da prethodnu signal iscrtamo u crvenoj boji, potrebno nam je objašnjenje za način zadavanja boja i tipa linije u naredbi `plot`, pa kucamo `help plot` i dobijamo opis načina korišćenja i parametre te naredbe. Naredba `plot(t,x,'r')` će ostvariti željenu promenu boje. Takođe, može se u helpu naredbe `plot` videti i način korišćenja kada se želi iscrtati više signala na istom grafiku. Tada se koristi proširena forma naredbe `plot` gde se sukcesivno navode vektori  $x$  i  $y$  argumenata za sve signale koji se crtaju, sa opcionim navođenjem stringa formata za boje i tip linija pojedinačnih signala. Npr. ako se, pored signala  $x(t)$  želi prikazati i signal  $y(t)=\exp(0.1t)\cos(\pi/5)$ , sledeći Matlab kod će dati željeni dijagram, predstavljen na slici 7.

```
t=-10:0.01:10;
x=exp(0.1*t).*sin(pi/5*t);
y=exp(0.1*t).*cos(pi/5*t);
plot(t,x,'b',t,y,'g');
```



Sl. 7. Istovremeni grafički prikaz signala  $x(t)$  i  $y(t)$

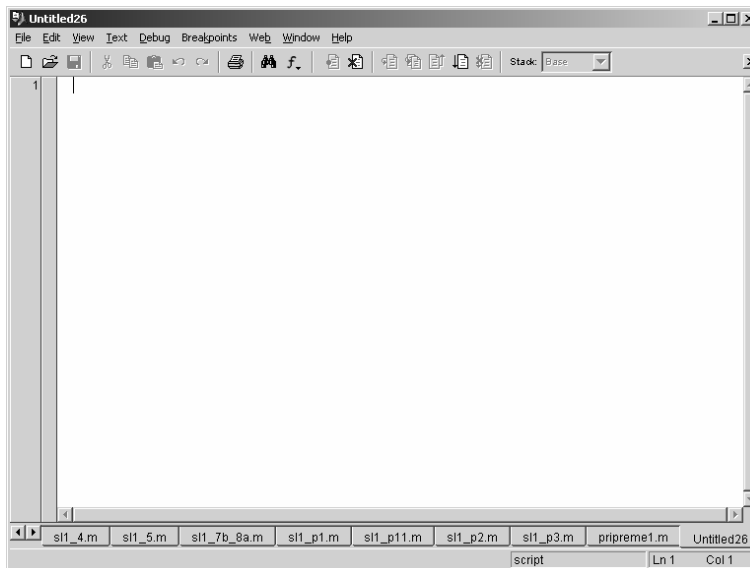
### (3) Matlab programi – skriptovi

Matlab skriptovi su standardne ASCII datoteke sa ekstenzijom ".m". Naredbe u programu Matlab sukcesivno izvršava do završetka programa, odnosno do prve greške u programu. Iako bilo koji editor teksta može da napravi ASCII datoteku za program, Matlab programski paket sadrži i editor teksta, koji podržava integrisani razvoj Matlab programa (pokretanje Matlab programa iz editora, debugging, breakpoints itd). Izborom stavke menija: File/New/M-file u Matlab-ovom komandnom prozoru, pokreće se Matlab-ov programski editor (sl. 8), u kojem možemo kreirati novi program ili modifikovati već postojeće programe.

Radi demonstracije osnovnih naredbi za ulaz i izlaz, ukucajmo u editoru jednostavan program za unos broja i godine indeksa, njihovo sabiranje i ispis:

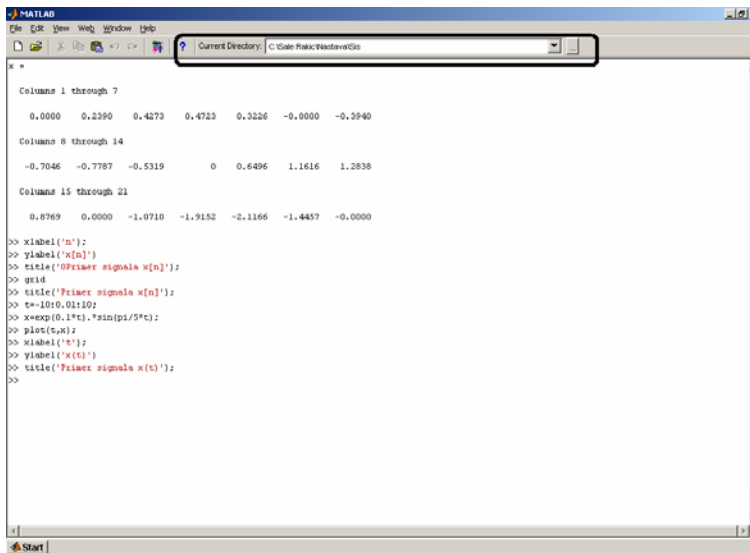
```
b=input('Unesite broj indeksa:');
g=input('Unesite godinu indeksa:');
disp(['Uneti parametri: b=' num2str(b) ', g=' num2str(g)]);
disp(['Zbir=' num2str(b+g)]);
```

Preporučljivo je programe snimati i čuvati u ličnom folderu, a ne u ponuđenom Matlab-ovom work folderu (izborom opcije menija editora File/Save As). Ime programa treba da bude jednosložno i u skladu sa Windows konvencijama za imenovanje datoteka, da bi mogli navođenjem imena programa u Matlabovom komandnom prozoru da isti pokrenemo. Snimimo program u lični folder (npr. c:\Sale Rakic\Nastava\Sis), pod imenom primer1 (eksenziju .m dodaje sam editor).



Sl. 8. Matlab-ov programski editor

Sada treba preći u Matlab-ov komandni prozor i za radni folder proglasiti lični folder u kome je program, kao što je prikazano na slici 9.



Sl. 9. Promena tekućeg radnog foldera Matlab okruženja

Navođenjem imena programa u komandnom promptu, nalaže se Matlabu da pokrene program. Korisniku programa se pruža mogućnost da unese podatke, a zatim izračunava i ispisuje zbir.

```
>> primer1
Unesite broj indeksa:306↵
Unesite godinu indeksa:2003↵
Uneti parametri: b=306, g=2003
Zbir=2309
```

**(4) Vremenske transformacije signala**

Neka je zadat signal  $x(n) = \begin{cases} 0, & n < -N \\ n + N/2, & -N \leq n < 0 \\ \sin(\Omega n), & 0 < n \leq N \\ 0, & n > N \end{cases}$ ,  $N = 15$ ,  $\Omega = \pi/2$ ,

i potrebno je napraviti Matlab program za iscrtavanje odbiraka tog signala.

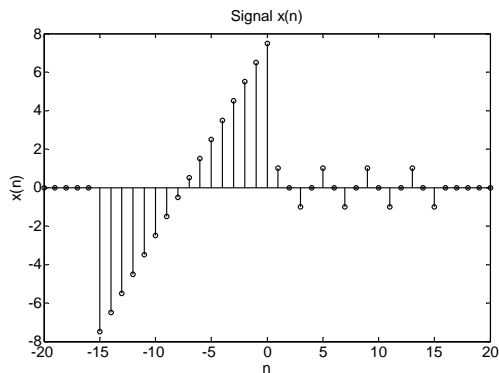
Jedan način da se zadatak reši ilustrovan je primerom 2.

For petljom sa odgovarajućom *if* strukturom ćemo generisati jedan po jedan odbirak  $xn$  (ime je izabrano da liči na  $x(n)$ ) signala u zavisnosti od oblasti u kojima su definisani analitički opisi signala, a zatim se generisani odbirak dodaje nizu  $x$  odbiraka signala.

```
% Primer 2

N=15;
omega=pi/2;
x=[];
for n=-N-5:N+5
    if n<-N
        xn=0;
    else
        if n<=0
            xn=n+N/2;
        elseif n <=N
            xn=sin(omega*n);
        else
            xn=0;
        end
    end
    x=[x xn];
end

figure(1);
stem(-N-5:N+5, x);
title('Signal x(n)'); xlabel('n'); ylabel('x(n)');
```



Sl. 10. Grafički prikaz diskretnog signala iz primera 2



Naredba *figure* otvara grafički prozor pod zadatim rednim brojem 1 i čini ga aktivnim, te naredba *stem* iscrtava odбирke signala u tom prozoru. Za neka buduća iscrtavanja, naredbom *figure(2)* bi se otvorio, pored postojećeg grafika 1, i grafik pod rednim brojem 2, pa bi naredna slika bila iscrtavana u njemu. Dobra je praksa koristiti naredbu *figure* pre naredbe iscrtavanja dijagrama, da bi pojedinačni grafici nekog programa uvek bili na fiksiranim brojevima grafičkih prozora.

Proširimo program primera 2, tako da se na dijagramu 2 iscrtava i signal pomeren u vremenu  $x(n+a)$ ,  $a = -5$ .

```
% dodatak primera 2 za pomeranje signala u vremenu x(n+a)

a=-5;
xp=[];
opseg_n=-N-5:N+5;           % opseg_n = [-20 -19 ... -1 0 1 ... 19 20]
duzina_n = length(opseg_n); % length vraća dužinu niza: duzina_n = 41
prvi_n = opseg_n(1);        % prvi element niza: prvi_n = -20
poslednji_n = opseg_n(duzina_n); % poslednji element niza:
                                % poslednji_n = opseg_n(41)=20

for n=opseg_n
    if (n+a >= prvi_n) && (n+a <= poslednji_n) % && je operator logičkog "i"
        (and)
            xnpa=x(n+a+N+6);
        else
            xnpa=0;
        end
        xp=[xp xnpa];
    end

figure(2);
stem(opseg_n, xp);
title('Vremenski pomeren x(n+a)');
xlabel('n');
ylabel('x(n+a)');
```

For petljom se generiše jedan po jedan odбирak vremenski pomerenog signala *xnpa* (ime je izabrano da liči na  $x(n+a)$ ) signala. Generisani odбирak se zatim dodaje nizu *xp* odбирaka signala.

Pošto se odbirci originalnog signala  $x$  pomeraju za 5 odбирaka u desno, a iscrtavamo sliku u istom opsegu  $[-20, 20]$ , početak signala  $x(n+a)$  treba popuniti nulama, pošto  $x(n)$  nismo definisali za odбирke  $n < -20$ .

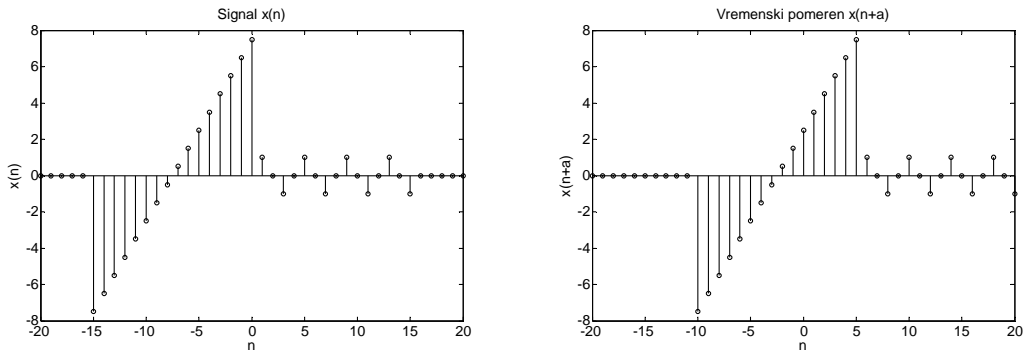
Slično bi vredelo za  $a > 0$  i pomeranje odбирaka u levo i „rupu” sa desne strane slike. Stoga se u for petlji nalazi if struktura, koja proverava da li imamo definisane odбирke  $x(n)$ , na osnovu kojih se generišu odbirci pomerenog signala, pa, ako ih ima, dodeljuje *xnpa*-u vrednost odбирka  $x(n+a)$ .

Pošto indeksi elemenata u Matlab nizu  $x$  imaju samo pozitivne vrednosti od 1 do 41, tj. sadržaj niza  $x$  je:

Elementa niza x	x(1)	x(2)	x(3)	...	x(41)
Oдбирak u elementu	$x(-20)$	$x(-19)$	$x(-18)$	...	$x(20)$

ostaje još da se da pomeraj  $N + 6 = 21$  u indeksu Matlab niza  $x$  u dodeli  $xnpa=x(n+a+N+6)$ , da bi se adresirao upravo odбирak  $x(n+a)$ .

Rezultat rada programa su dijagrami 1 i 2 – signal i vremenski pomeren signal, kao na slici 11.

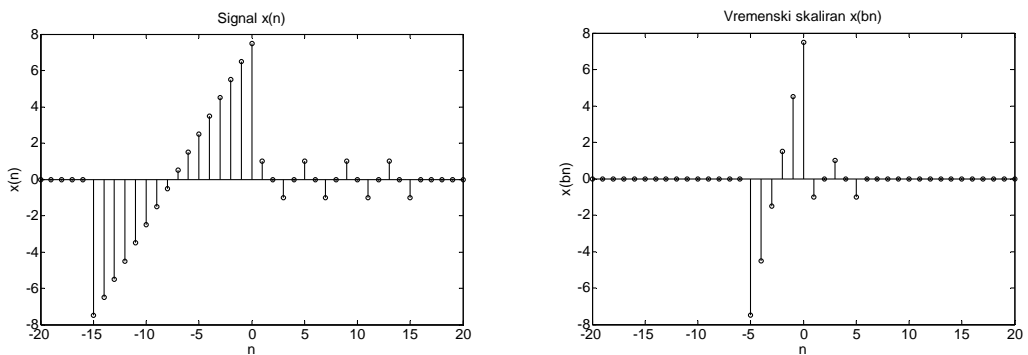


Sl. 11. Grafički prikaz originalnog  $x(n)$  i vremenski pomerenog signala  $x(n+a)$  iz primera 2

Proširimo, dalje, primer 2 iscrtavanjem odbiraka vremenski skaliranog signala  $x(bn)$ ,  $b = 3$ .

```
% skaliranje vremena x(b*n)
b=3;
xs=[];
for n=opseg_n
    if (b*n >= prvi_n) && (b*n <= poslednji_n)
        xbn=x(b*n+N/6);
    else
        xbn=0;
    end
    xs=[xs xbn];
end

figure(3);
stem(opseg_n, xs);
title('Vremenski skaliran x(bn)'); xlabel('n'); ylabel('x(bn)');
```



Sl. 12. Grafički prikaz originalnog  $x(n)$  i vremenski skaliranog signala  $x(bn)$  iz primera 2