DBBT
Digital Broadcasting & Broadband Technologies

Co-funded by the Erasmus+ Programme of the European Union

Procesiranje signala

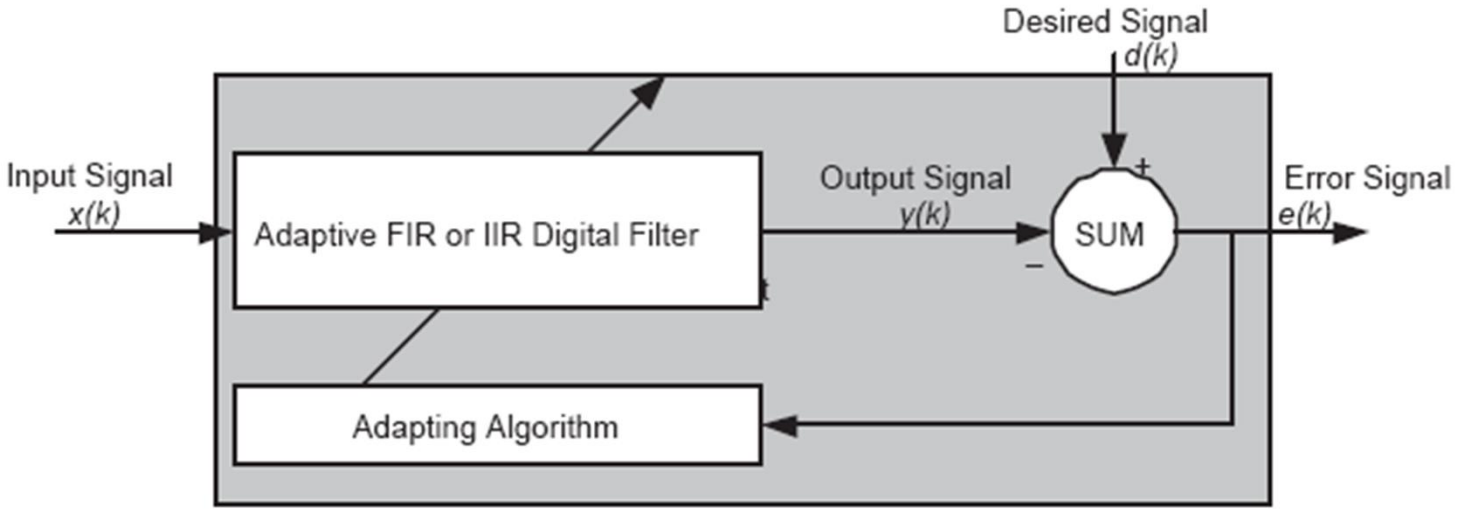
Profesor dr Miroslav Lutovac

"This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein"

- Adaptivni filtri i njihova primena
- Identifikacija sistema
- Identifikacija inverznog sistema
- Poništavanje šuma
- Predikcija

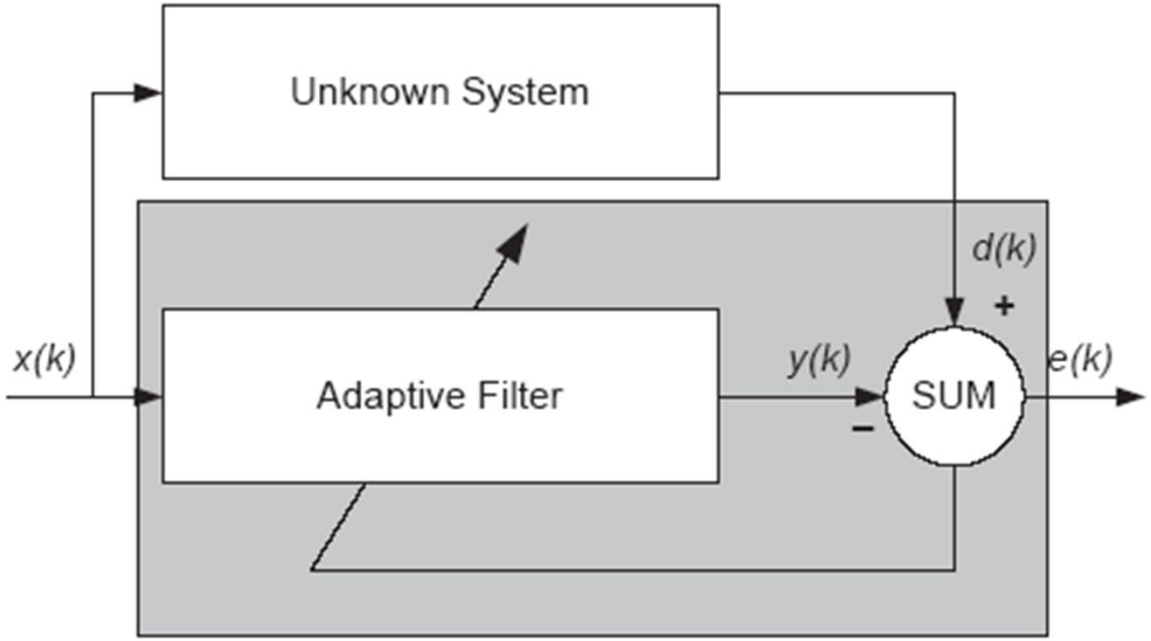
Šta je adaptivni filtar?

- **Adaptivni filtar** je filtar čiji koeficijenti se podešavaju (menjaju) tokom rada filtra tako da se procesiranjem adaptivnim filtrom ostvari željeni rezultat obrade
- Primer: Signal se propušta kroz adaptivni filtar čiji koeficijenti se menjaju sve dok filtrirani signal ne postane približno jednak zadatom signalu



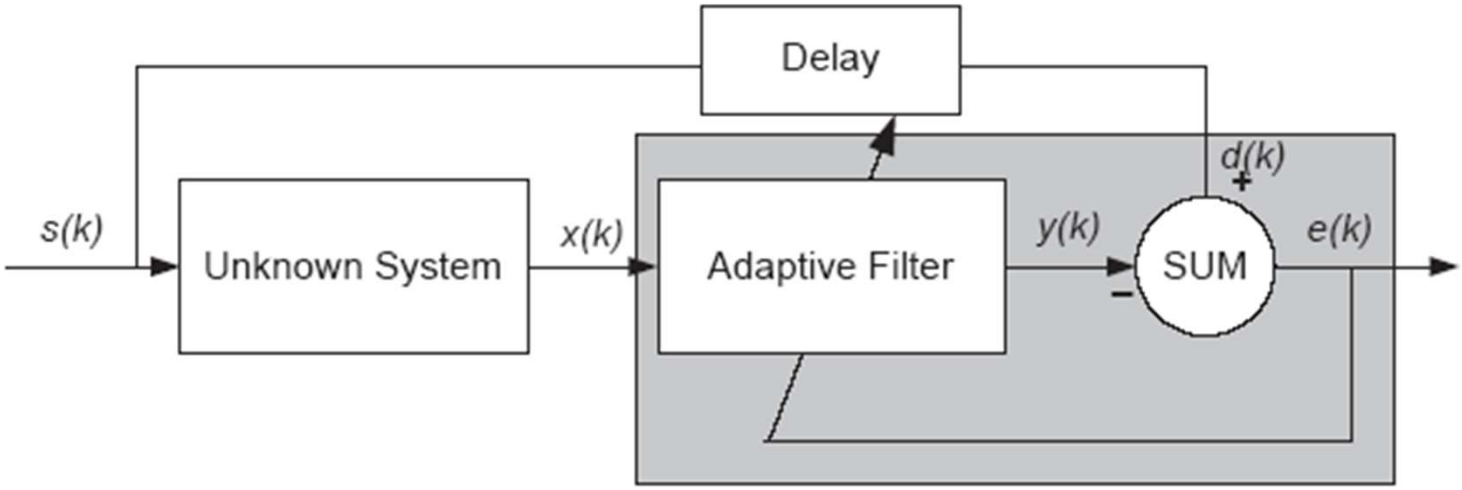
Identifikacija sistema

- Adaptivni filtar se koristi da se odrede karakteristike nepoznatog sistema (telekomunikacionog kanala)
- Na ulaz adaptivnog filtra se dovodi signal koji je identičan pobudi nepoznatog sistema
- Kada je razlika izlaza nepoznatog sistema i adaptivnog filtra veoma mala, karakteristika filtra je približno ista kao i nepoznatog sistema



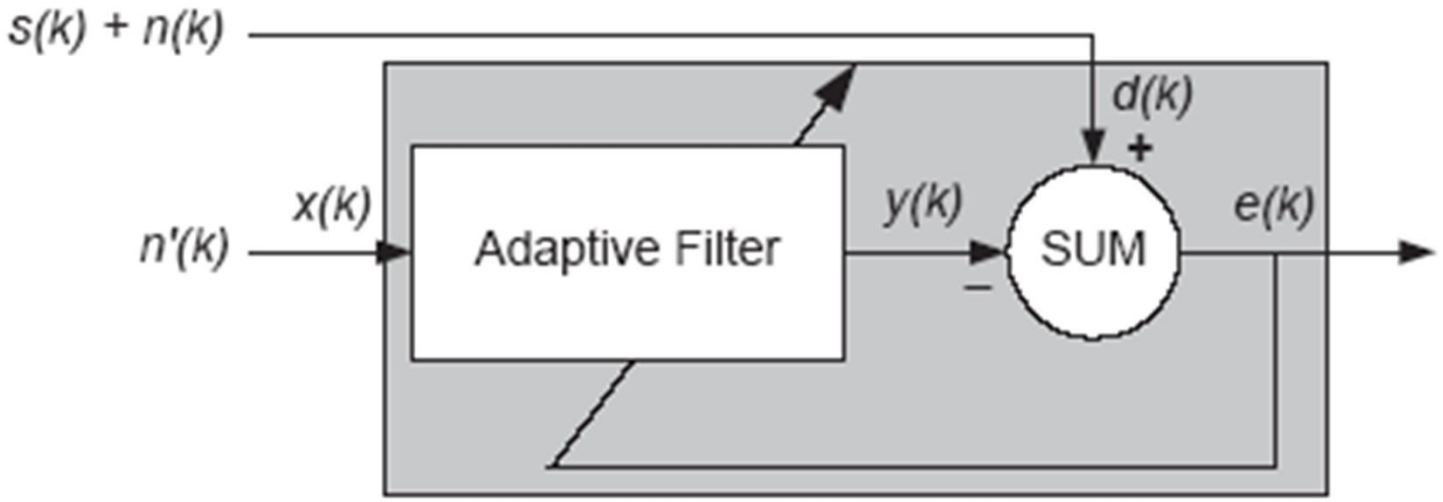
Identifikacija inverzne karakteristike sistema

- Izlaz nepoznatog sistema se veže kaskadno sa adaptivnim filtrom, a na ulaz kašnjenja se dovodi signal identičan ulazu u nepoznati sistem
- Kada je razlika mala, tada je frekvencijska karakteristika adaptivnog filtra jednaka recipročnoj karakteristici nepoznatog sistema



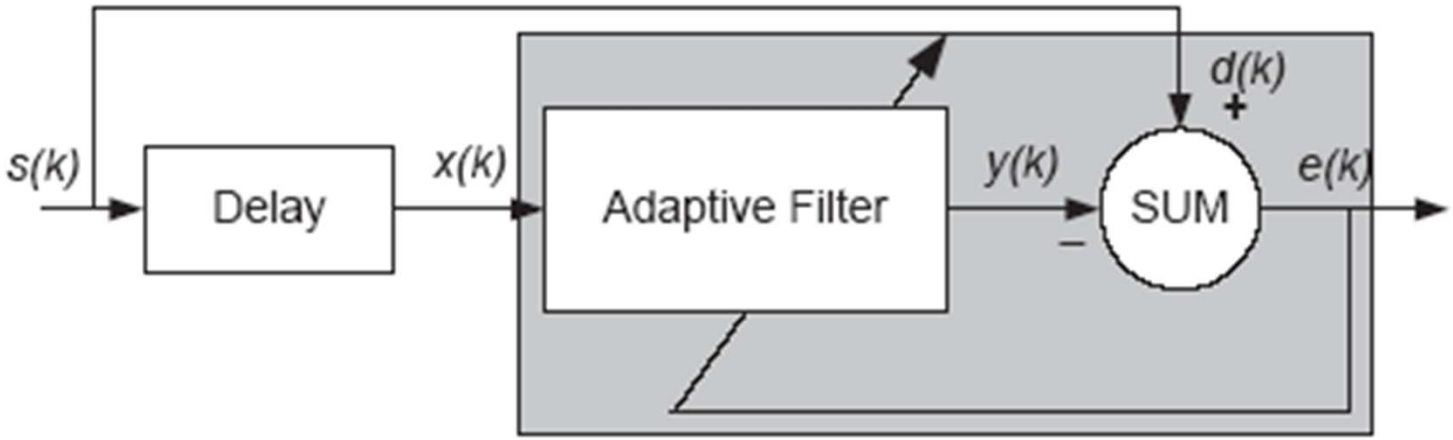
Poništavanje šuma i interferencije

- Iz signala $s[k]+n[k]$ treba da se eliminiše smetnja, $s[k]$ je informacioni signal, $n[k]$ je signal smetnje
- Signal $n'[k]$, koji je korelisan sa $n[k]$, dovodi se na adaptivni filtar koji podešava koeficijente sve dok signal greške ne postane približno jednak informacionom signalu



Predikcija periodičnog signala

- Neka je signal $s[k]$ periodičan u nekom vremenskom intervalu i stacionaran ili sporo promenljiv
- Adaptivni filtar treba da predvidi buduće vrednosti signala na osnovu prethodnih vrednosti
- Može da se koristi i za eliminaciju periodičnog signala iz slučajnog signala



Algoritmi za adaptivne filtre

- LMS (Least Mean Squares)
- RMS (Recursive Least Squares)
- Adaptivni filtri u frekvencijskom domenu
- Lattice adaptivni filtri

Least Mean Squares (LMS) Based FIR Adaptive Filters

- **Adaptive filter method – adapting algorithm used to generate filter coefficients during adaptation**
- adaptfilt.adjlms Adjoint LMS FIR adaptive filter algorithm
- adaptfilt.blms Block LMS FIR adaptive filter algorithm
- adaptfilt.blmsfft FFT-based Block LMS FIR adaptive filter algorithm
- adaptfilt.dlms Delayed LMS FIR adaptive filter algorithm
- adaptfilt.filtxlms Filtered-x LMS FIR adaptive filter algorithm
- **adaptfilt.lms** LMS FIR adaptive filter algorithm
- **adaptfilt.nlms** Normalized LMS FIR adaptive filter algorithm
- adaptfilt.sd Sign-data LMS FIR adaptive filter algorithm
- adaptfilt.se Sign-error LMS FIR adaptive filter algorithm
- adaptfilt.ss Sign-sign LMS FIR adaptive filter algorithm

sistem_identification.m

```
% x je slucajni signal kao informacioni signal
nsamp = 500;
x = randn(1,nsamp);

% Nepoznati sistem koji treba da se identifikuje
% generisan je kao FIR filtar sa koeficijentima b
b = fir1(31,0.5);

% y je signal koji je primljen kao izlaz nepoznatog sistema b
% a na pobudu poznatim signalom x
xf = filter(b,1,x);

% signal [uma koji je dodat u prenosu korisnog signala
n = 0.1*randn(1,nsamp);

% signal koji je primljen = informacioni + sum
d = xf+n;
```

Informacioni signal

Informacioni signal na izlazu nepoznatog sistema

Informacioni signal i šum

sistem_identification.m (2)

```
% LMS step size
mu = 0.008;

% Izabran je LMS algoritam sa tezinskim koeficijentom mu
% pocetna stanja su izabrana da su 0
ha = adaptfilt.lms(32,mu);

% procesiranje adaptivnim filtrom
[y,e] = filter(ha,x,d);
```

Korak adaptacije

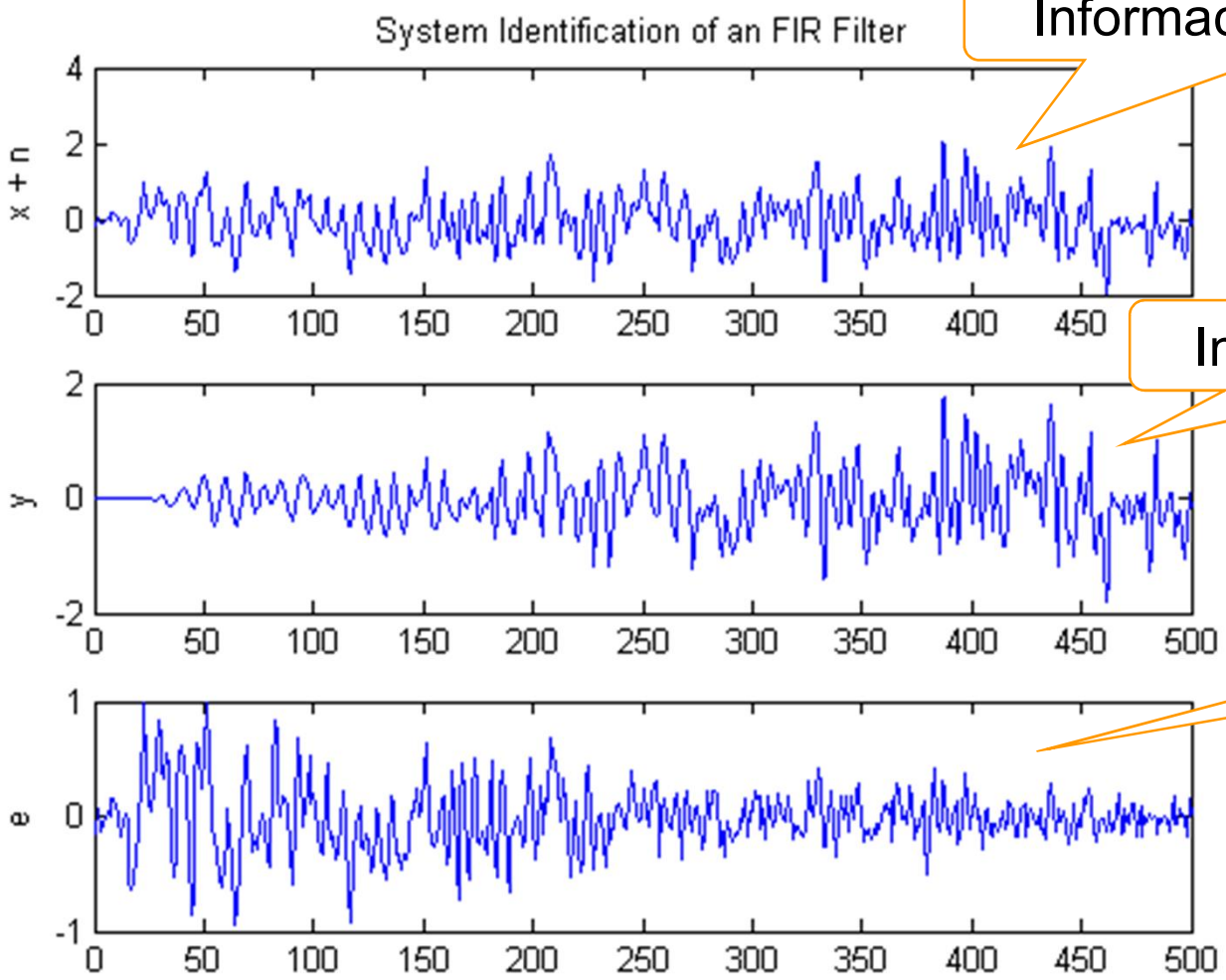
LMS algoritam

Procesiranje

sistem_identification.m (3)

```
subplot(3,1,1);
plot(1:500,d);
ylabel('x + n');
title('System Identification of an FIR Filter');
subplot(3,1,2);
plot(1:500,y);
ylabel('y');
subplot(3,1,3);
plot(1:500,e);
ylabel('e');
```

sistem_identification.m (4)



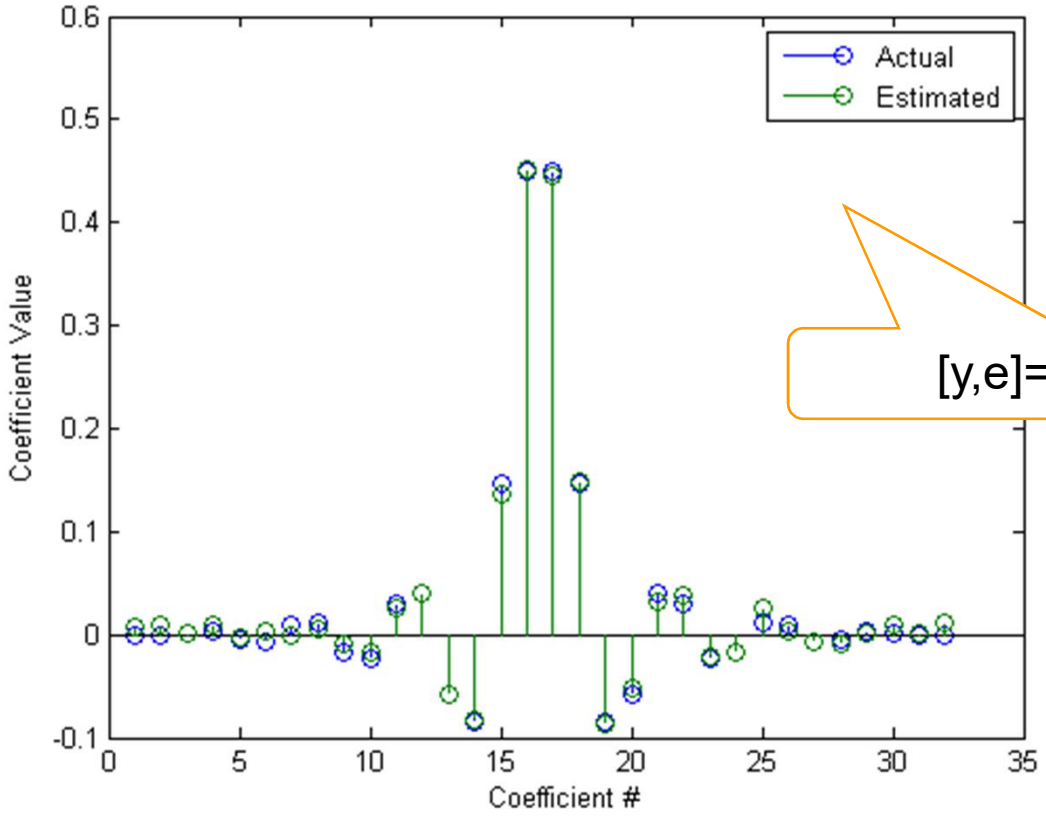
Informacioni signal i šum

Informacioni signal

Greška

sistem_identification.m (5)

```
stem([b.',ha.coefficients.']);
legend('Actual','Estimated');
xlabel('Coefficient #'); ylabel('Coefficient Value');
```

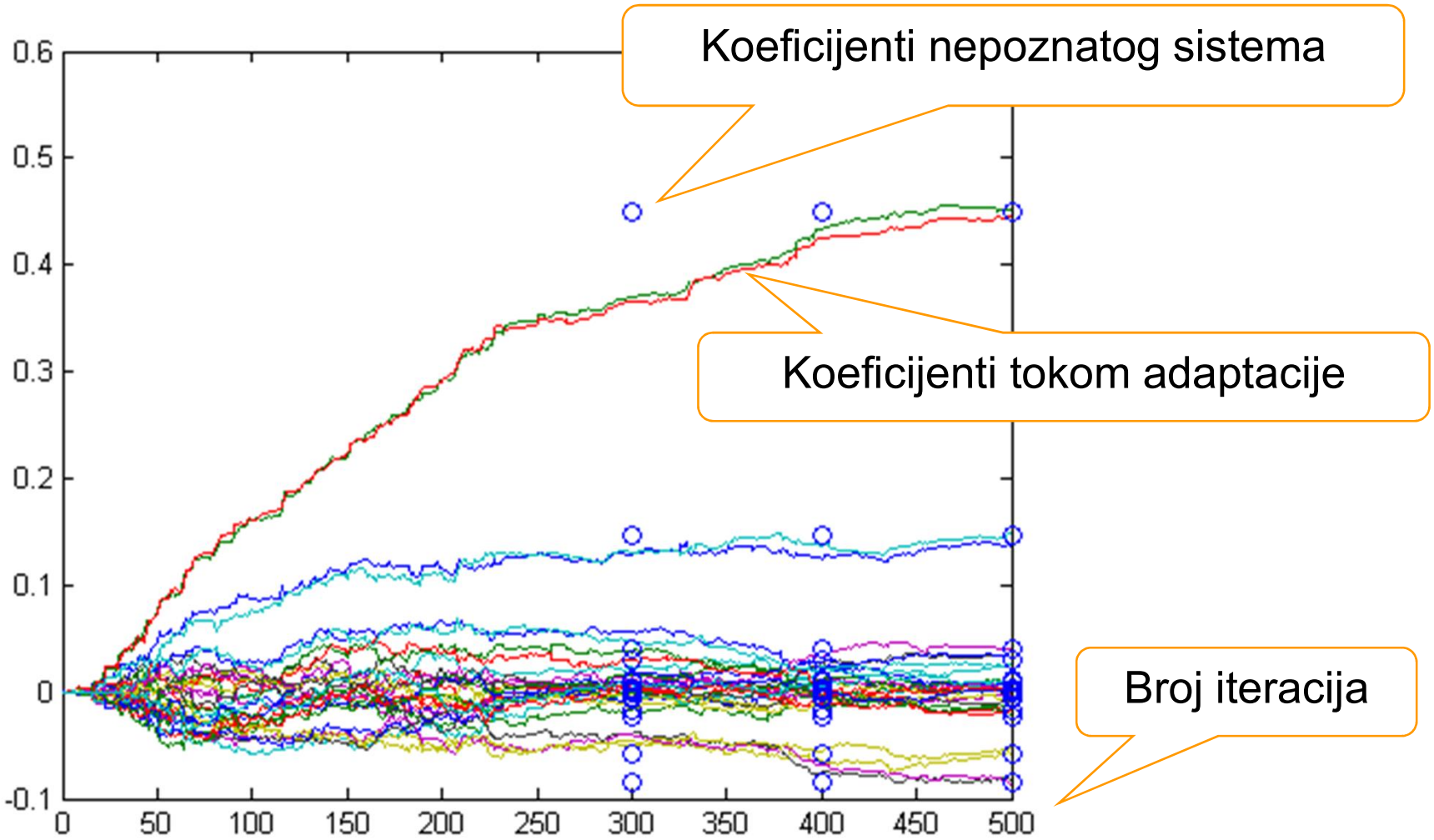


[y,e]=filter(ha,x,d)

sistem_identification.m (6)

```
c0=[];
reset(ha);                  % Start from zero again
ha.PersistentMemory = true; % Don't reset
for ind = 1:nsamp
    [y1,e1] = filter(ha,x(ind),d(ind));
    c0 = [c0;ha.coefficients];
end
plot(c0)
```

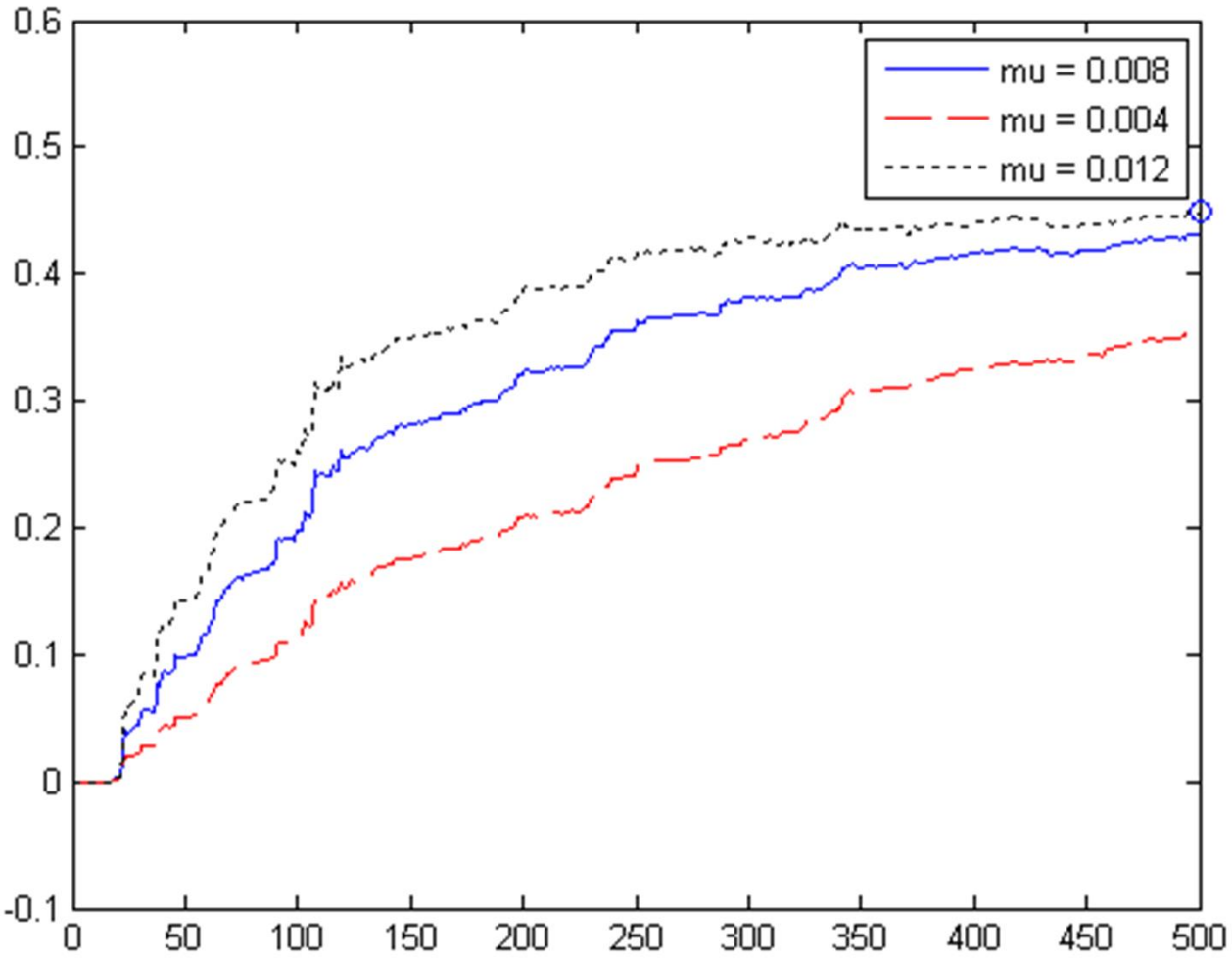
sistem_identification.m (7)



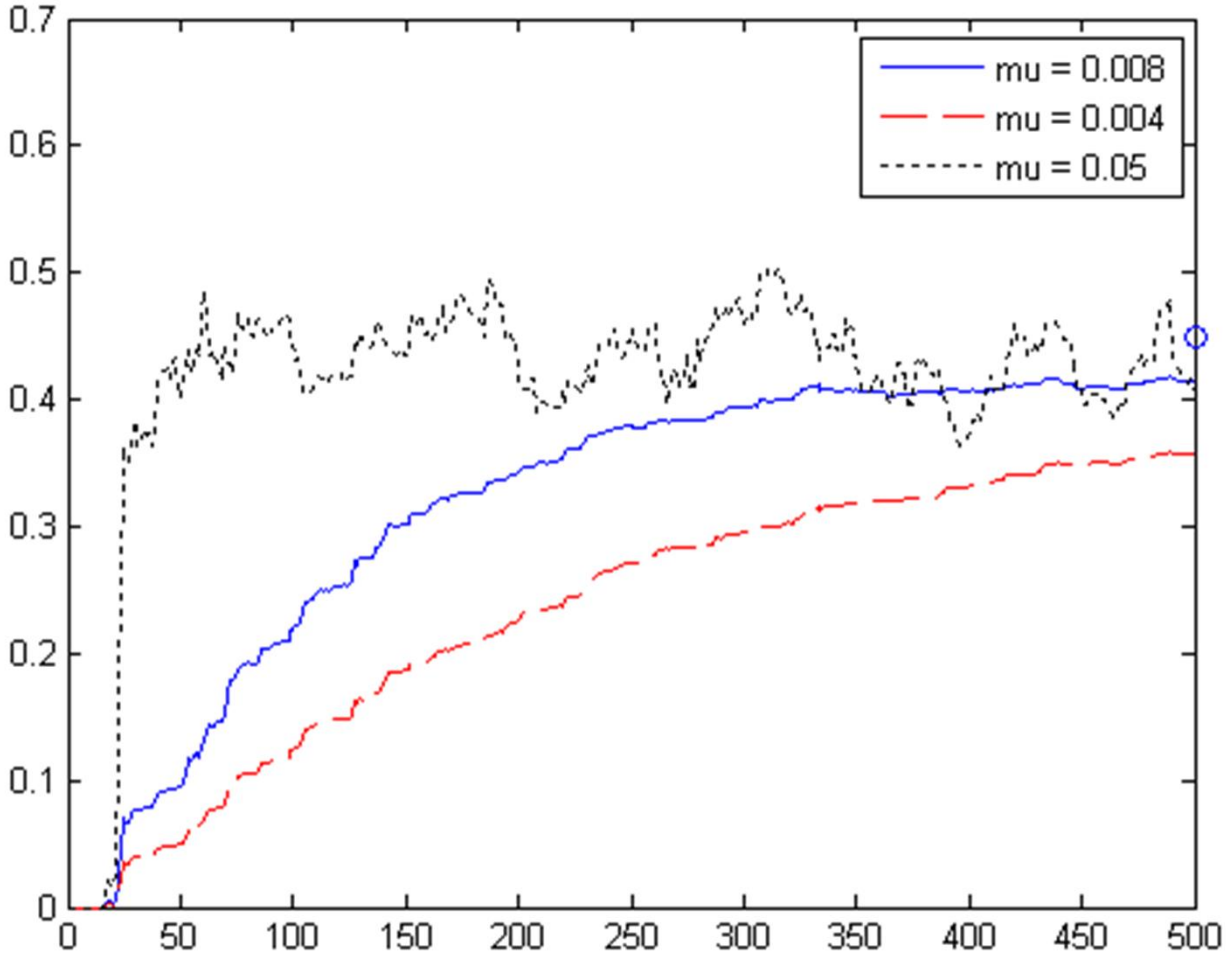
sistem_ident_2.m

```
mu = 0.004;
ha = adaptfilt.lms(32,mu);  [y,e] = filter(ha,x,d);
c1=[]; reset(ha); ha.PersistentMemory = true;
for ind = 1:nsamp
    [y1,e1] = filter(ha,x(ind),d(ind));
    c1 = [c1;ha.coefficients];
end

mu = 0.012;
ha = adaptfilt.lms(32,mu);  [y,e] = filter(ha,x,d);
c2=[]; reset(ha); ha.PersistentMemory = true;
for ind = 1:nsamp
    [y1,e1] = filter(ha,x(ind),d(ind));
    c2 = [c2;ha.coefficients];
end
```

sistem_ident_2.m (2)



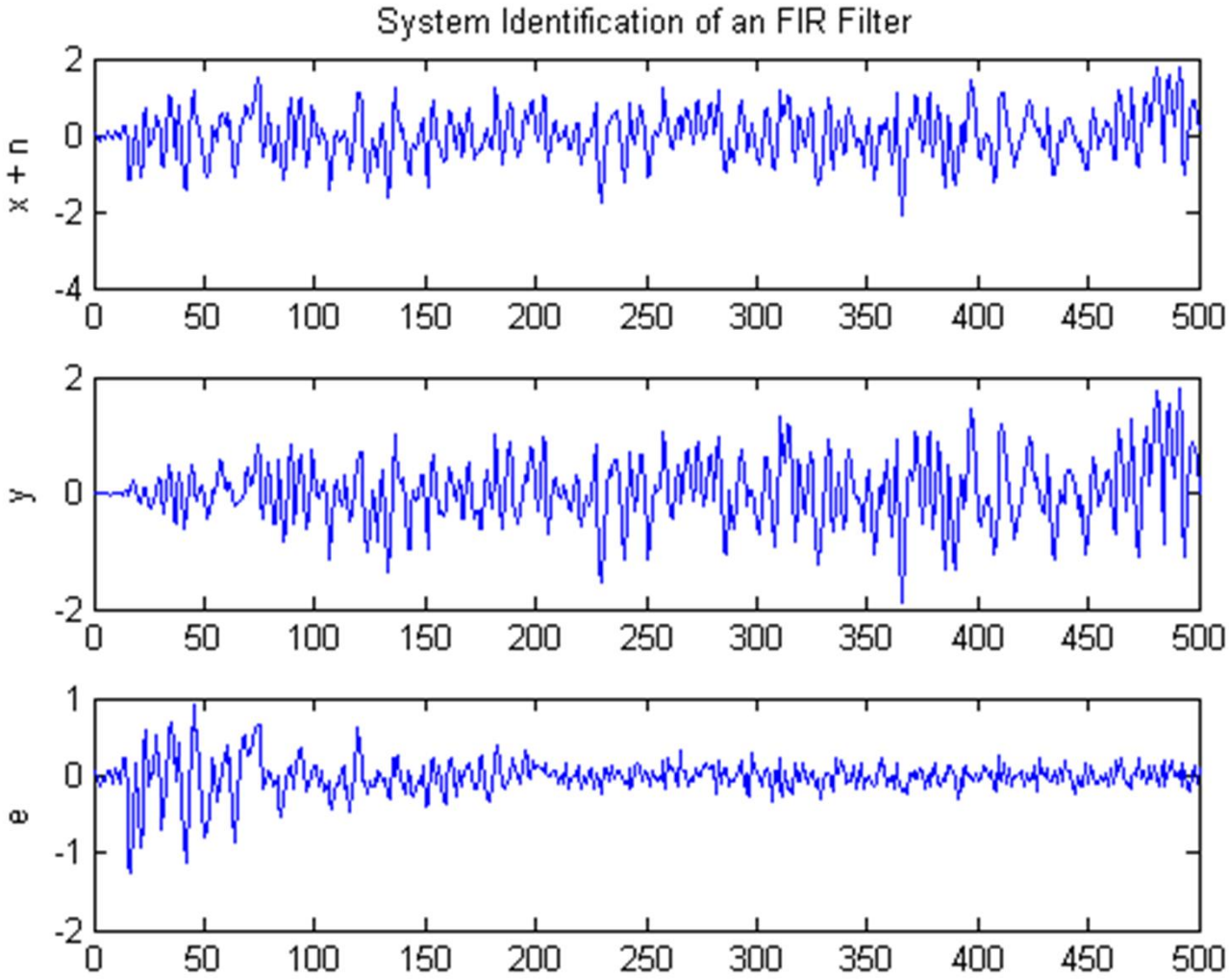
sistem_ident_3.m



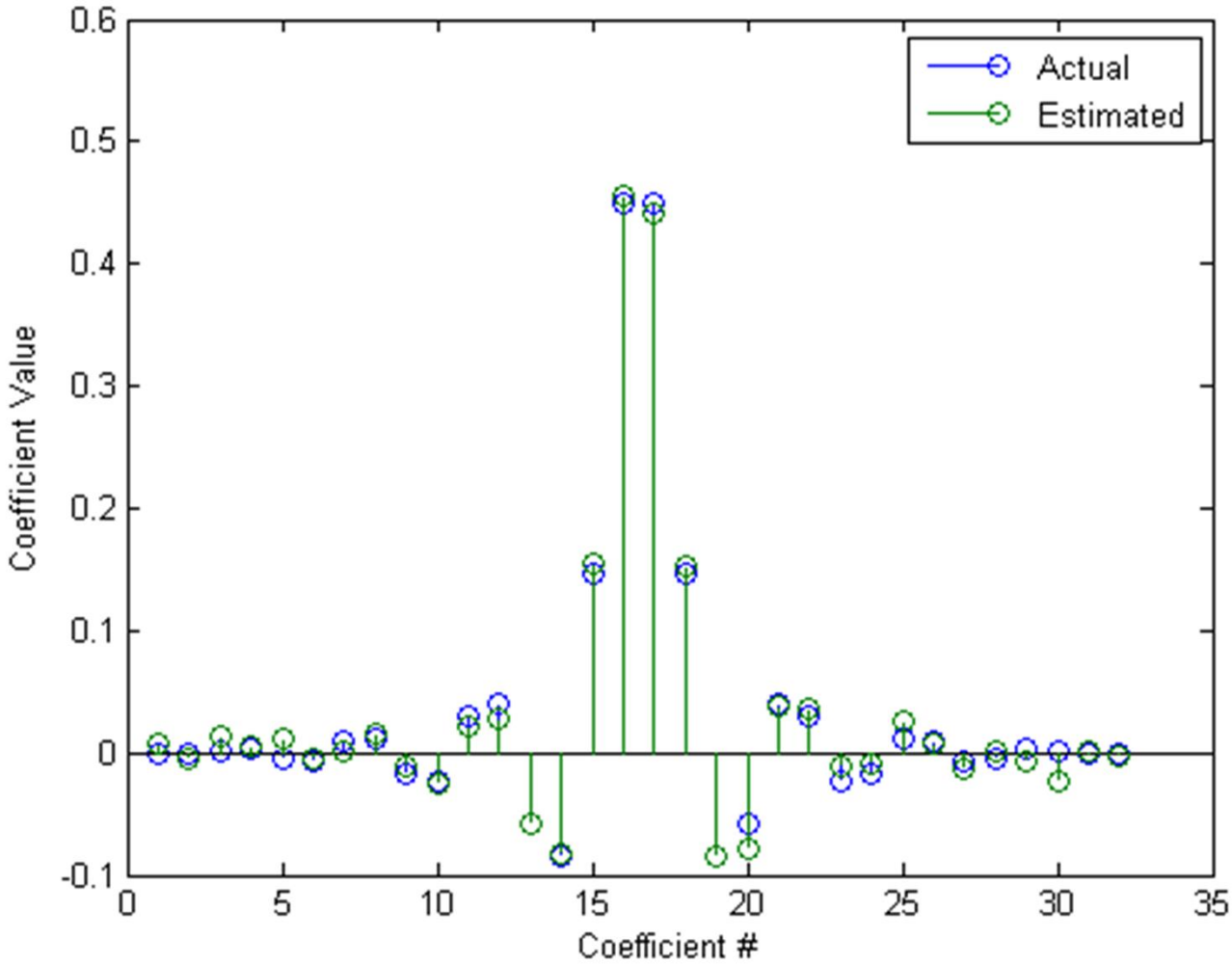
sistem_ident_nlms.m

```
mu = 0.4;

. . .

ha = adaptfilt.nlms(32,mu);
```

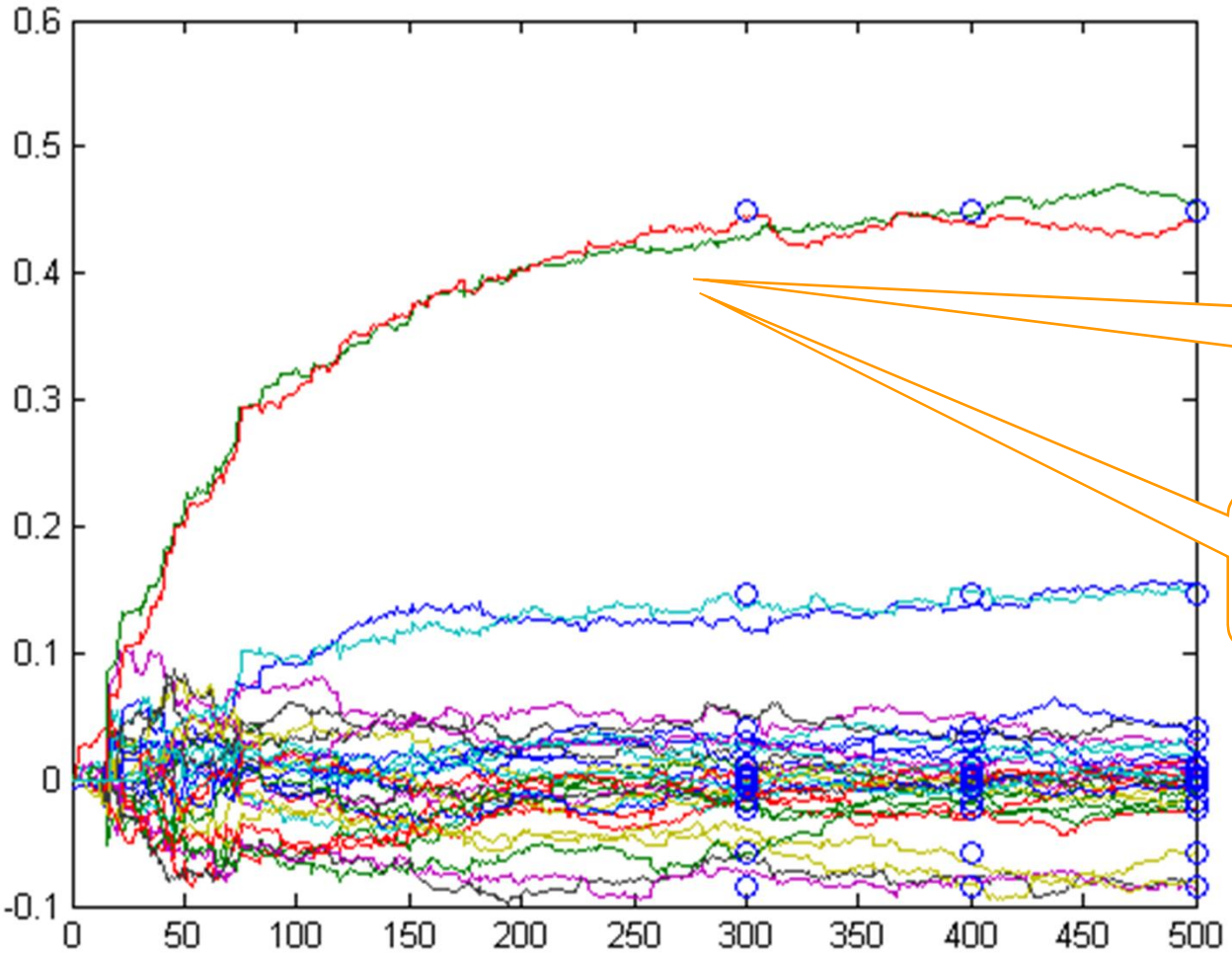
sistem_ident_nlms.m (2)



sistem_ident_nlms.m (3)



sistem_ident_nlms.m (4)



NLMS
mu = 0.4

Brže konvergira

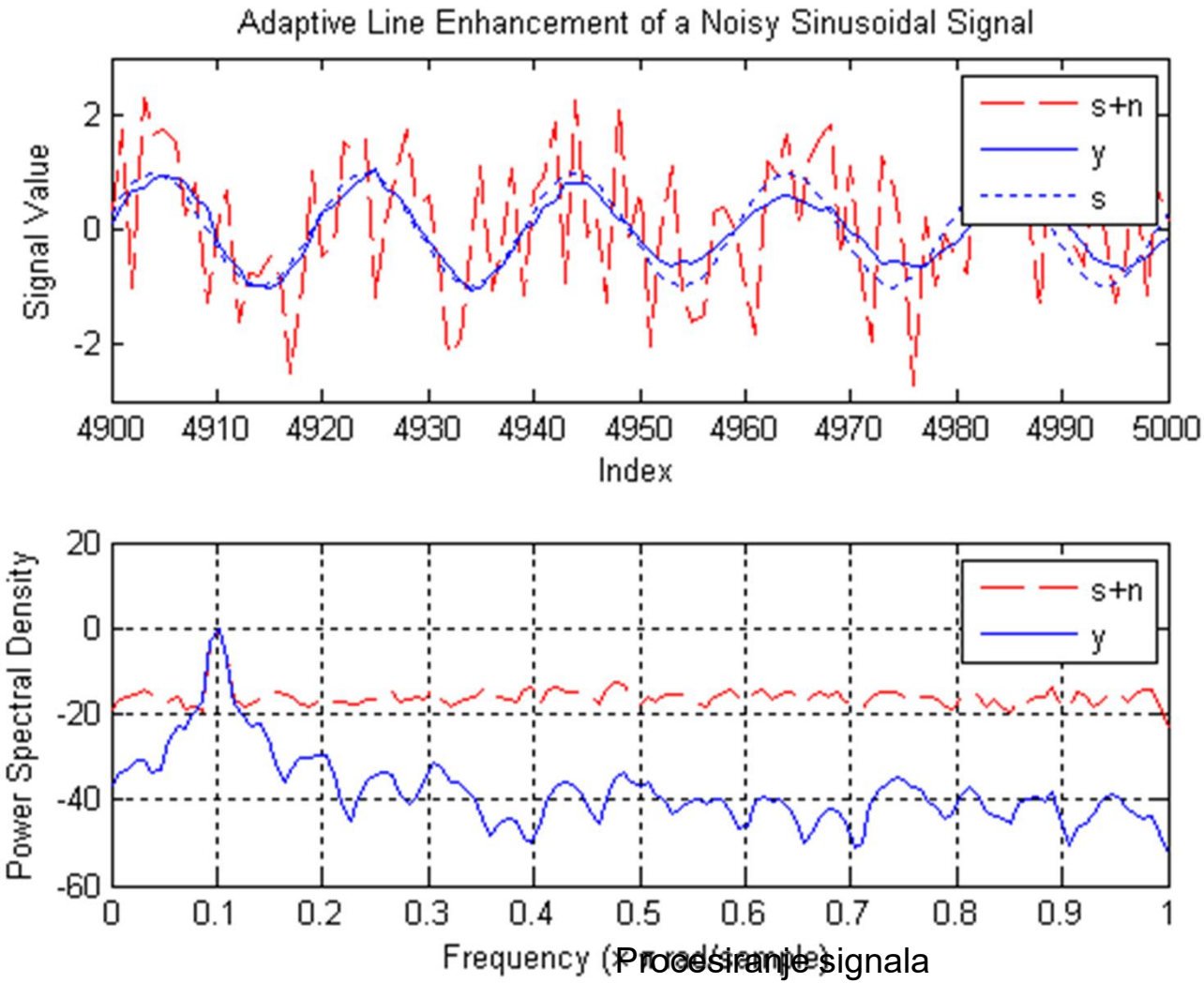
noise_cancellation.m

```
d=1; % Number of samples of delay
ntr=5000; % Number of iterations
v=sin(2*pi*0.05*[1:ntr+d]); % Sinusoidal signal
n=randn(1,ntr+d); % Noise signal
x=v(1:ntr)+n(1:ntr); % Input signal(delayed desired)
ds = v(1+d:ntr+d)+n(1+d:ntr+d);% Desired signal
mu = 0.0001;                % Sign-data step size.
ha = adaptfilt.sd(32,mu);
[y,e] = filter(ha,x,ds);
k = 1:ntr;
plot(k,ds,'r--',k,y,'b-',k,v(1+d:ntr+d),':');
axis([ntr-100 ntr -3 3]);
xlabel('Index'); ylabel('Signal Value');
```

noise_cancellation.m (2)

```
[pxx,om] = pwelch(x(ntr-1000:ntr));
pyy = pwelch(y(ntr-1000:ntr));
subplot(2,1,2);
w = om/pi;
plot(w,10*log10(pxx/max(pxx)),'r--',...
     w,10*log10(pyy/max(pyy)),'b-');
axis([0 1 -60 20]);
legend('s+n','y');
xlabel('Frequency (\times \pi rad/sample)');
ylabel('Power Spectral Density'); grid on;
```

noise_cancellation.m (3)



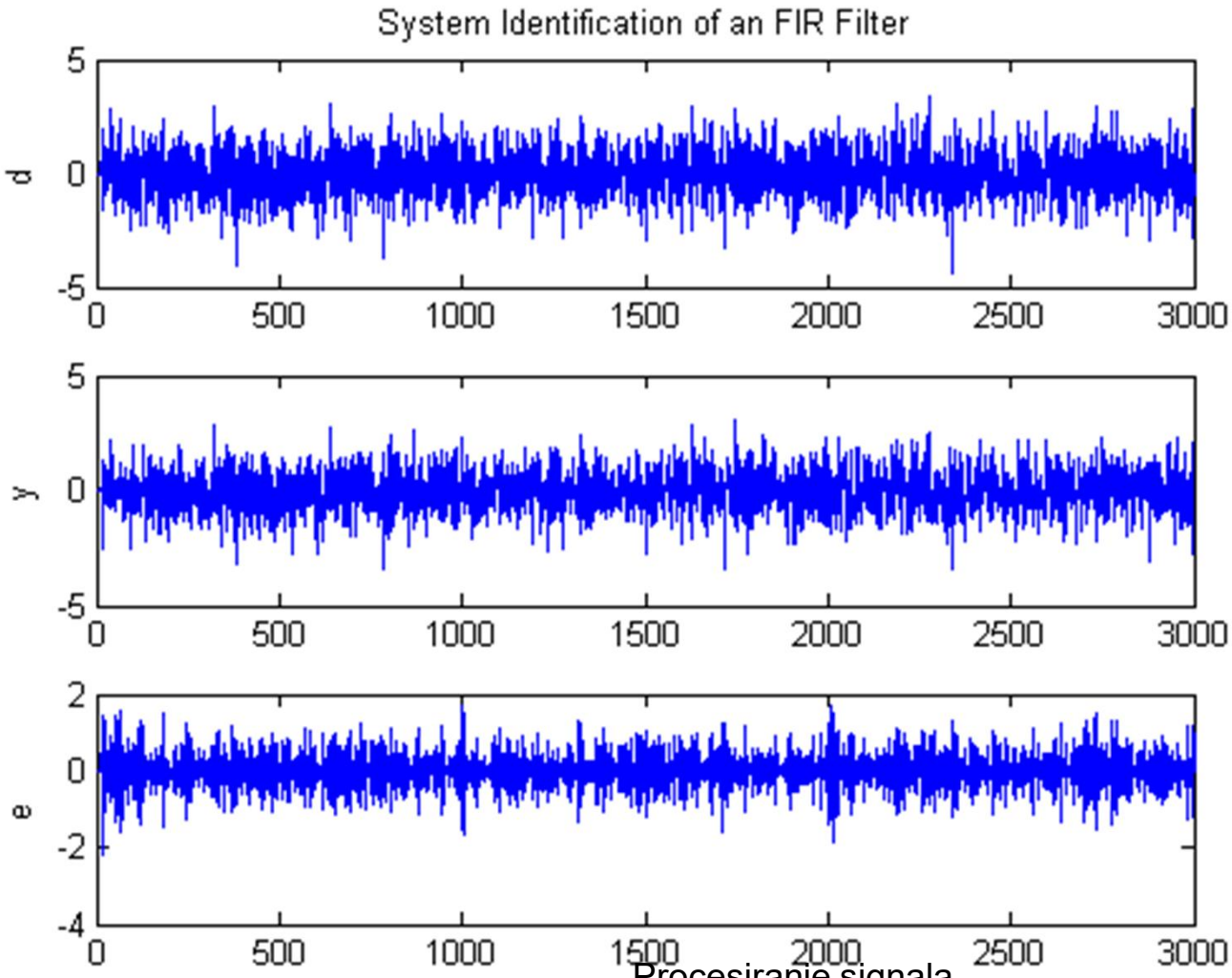
sistem_inv_ident.m

```
x = randn(1,3000);
delay = zeros(1,12);
d = [delay x(1:2988)];
ufilt = fir1(12,0.55,'low');
xdata = filter(ufilt,1,x);
p0 = 2*eye(13);
lambda = 0.99;
ha = adaptfilt.rls(13,lambda,p0);
[y,e] = filter(ha,xdata,d);
k = 1:length(d);
subplot(3,1,1);
plot(k,d);
ylabel('d');
title('System Identification of an FIR Filter');
subplot(3,1,2); plot(k,y);ylabel('y');
subplot(3,1,3); plot(k,e);ylabel('e');
```

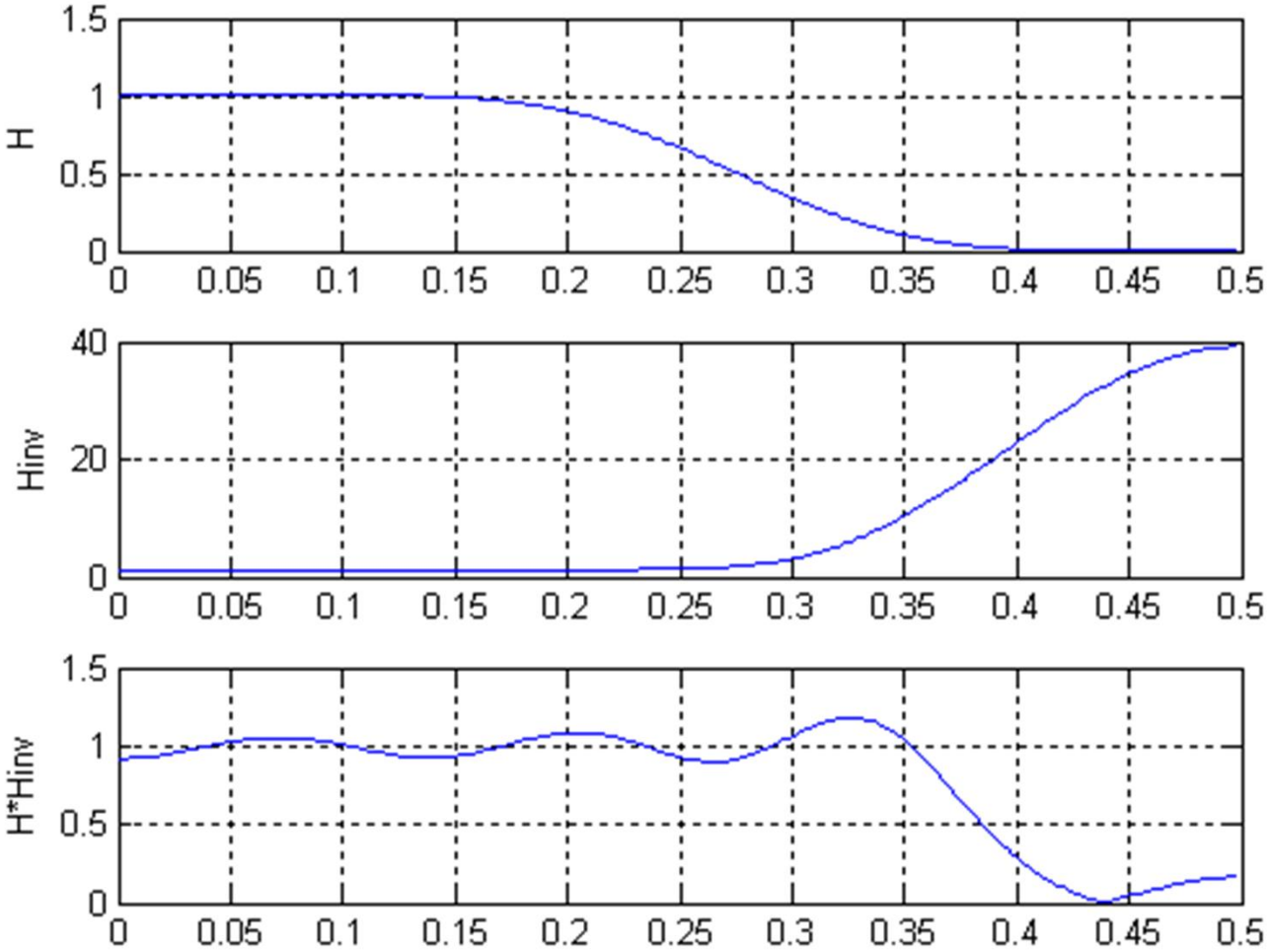
sistem_inv_ident.m (2)

```
figure
subplot(3,1,1)
[h,w] = freqz(ufilt,1,256);
plot(w/2/pi,abs(h))
ylabel('H'); grid on;

invb = ha.Coefficients;
subplot(3,1,2)
[h,w] = freqz(invb,1,256);
plot(w/2/pi,abs(h))
ylabel('Hinv'); grid on;

subplot(3,1,3)
[h,w] = freqz(conv(ufilt,invb),1,256);
plot(w/2/pi,abs(h))
ylabel('H*Hinv'); grid on;
```

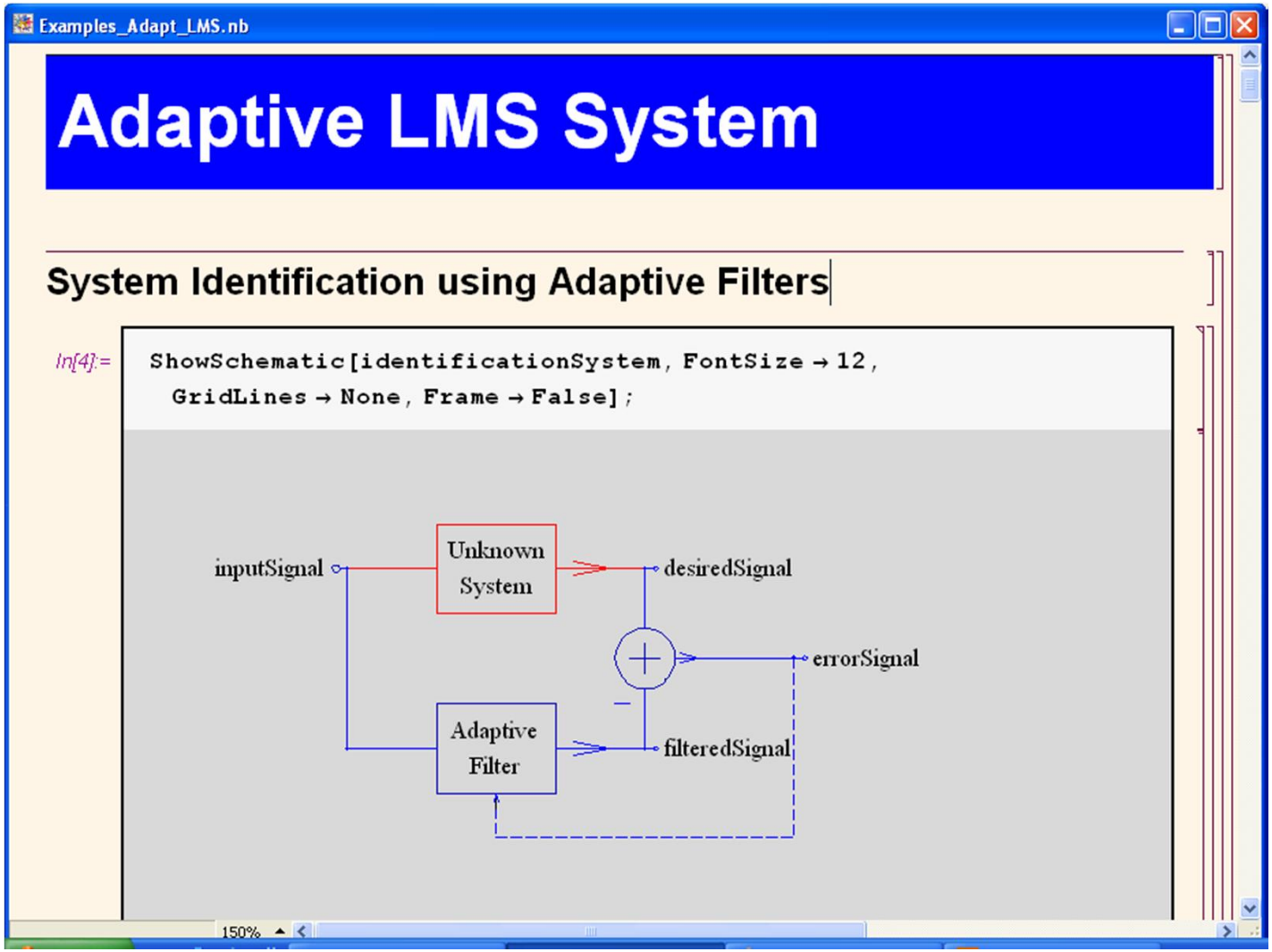
sistem_inv_ident.m (3)



sistem_inv_ident.m (4)



Adaptive LMS System



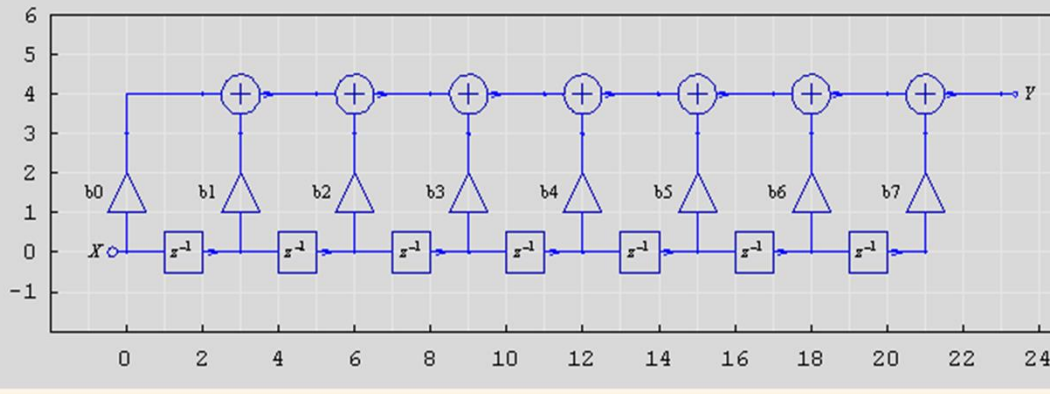
Draw Schematic

Examples_Adapt_LMS.nb

Example Unknown System

In[5]:=
```
{unknownSystemSchematic, inpCoords, outCoords} =
  DirectFormFIRFilterSchematic[{b0, b1, b2, b3, b4, b5, b6, b7}];
```

In[6]:=
```
unknownSystem = Join[
   unknownSystemSchematic,
   {{"Input", {0, 0}, X}, {"Output", {23, 4}, Y}}];
ShowSchematic[%, FontSize → 7, PlotRange → {{-2, 25}, {-2, 6}}];
```



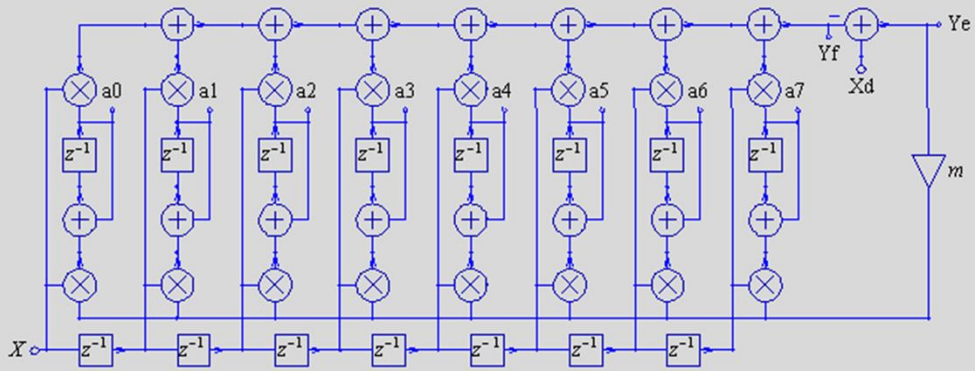
150%

Adaptive System

Examples_Adapt_LMS.nb *

The Adaptive System

In[59]:=
```
ShowSchematic[adaptiveSystem, FontSize → 9, GridLines → None,
  Frame → False];
```



In[60]:=
```
DiscreteSystemImplementationSummary[adaptiveSystem]
```

```
Input: {Y[{25, 9}], Y[{0, 0}]}

Parameter: {m}

Output: {Y[{27, 10}], Y[{24, 10}], Y[{1, 7}], Y[{4, 7}], Y[{7, 7}],
```

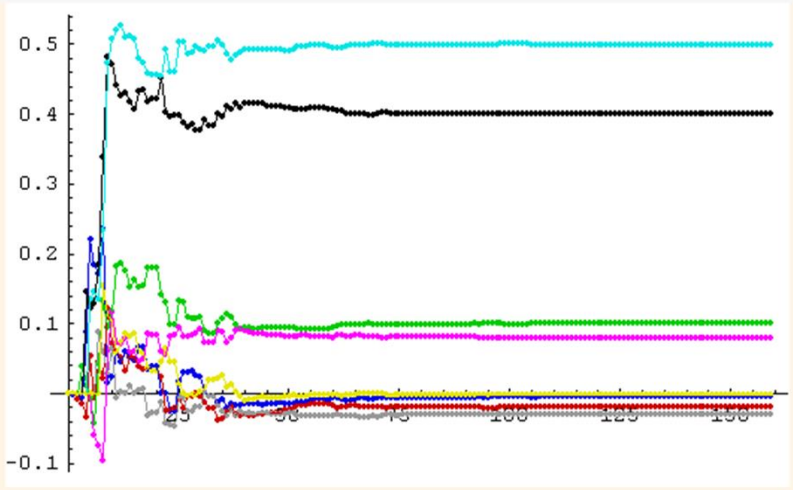
150%

Finding Adapted Coefficients

Examples_Adapt_LMS.nb

In[61]:=

```
coefSeq2 = MultiplexSequence[a0Seq2, a1Seq2, a2Seq2, a3Seq2,
    a4Seq2, a5Seq2, a6Seq2, a7Seq2];
SequencePlot[coefSeq2, PlotJoined → True, StemPlot → False];
```



The values of the coefficients of the adaptive system converge to

In[63]:=

```
coefValues2 = coefSeq2 // Last
```

Out[63]=

```
{-0.00500032, -0.0199994, 0.0999997,
 0.4, 0.5, 0.0800001, -0.03, -0.00200037}
```

150%

Symbolic Response of the Unknown System

- desiredSignalSymbolic = DiscreteSystemSimulation[unknownSystem,inputSignal]
- {{-0.0026 b0},
  {-0.1111 b0 - 0.0026 b1},
  {0.0751 b0 - 0.1111 b1 - 0.0026 b2},
  {0.05 b0 + 0.0751 b1 - 0.1111 b2 - 0.0026 b3},
  {-0.0517 b0 + 0.05 b1 + 0.0751 b2 - 0.1111 b3 - 0.0026 b4},
  … }=
  {{0.000013}, {0.0006075}, {0.0015865}, {-0.013902}, ...}

Nonlinear Systems

Example_NonlinearSystem.nb

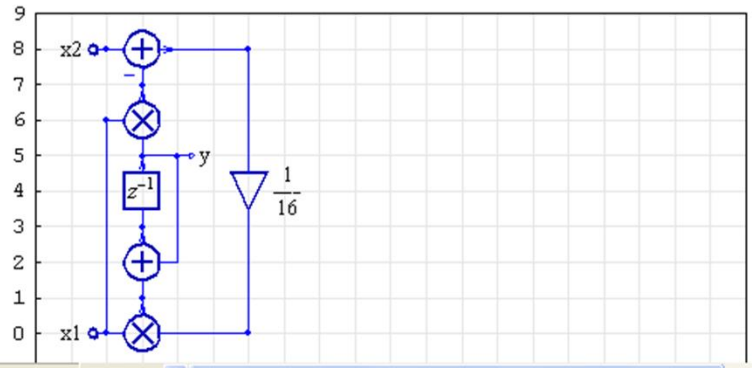
Solving Nonlinear Systems

The term ***solve*** means that *SchematicSolver* can find the closed-form expression of the output signal for a known stimulus given by a closed-form expression.

Draw Schematic

Show Schematic

```
In[3]:= ShowSchematic[nonlinearSystem, PlotRange → {{-2, 18}, {-1, 9}}];
```



150%

Response - unit step input

Example_NonlinearSystem.nb

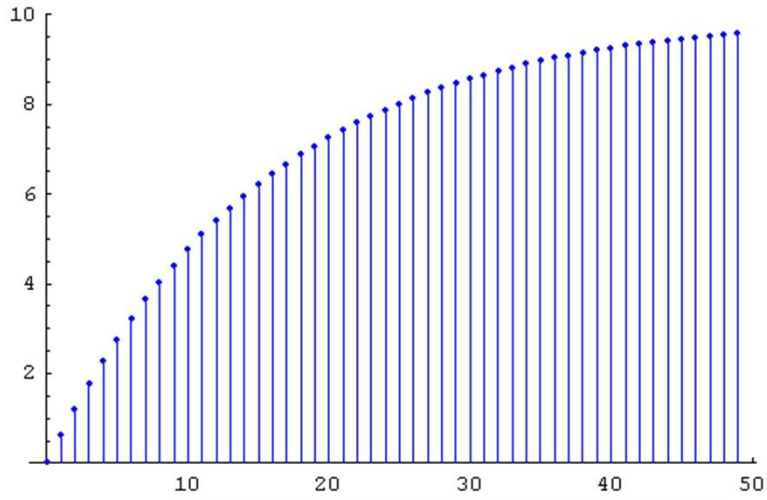
In[8]:=
```
inpSeq = MultiplexSequence[inpSeq1, inpSeq2];
```

DiscreteSystemSimulation finds the system output:

In[9]:=
```
outSeq = DiscreteSystemSimulation[nonlinearSystem, inpSeq];
```

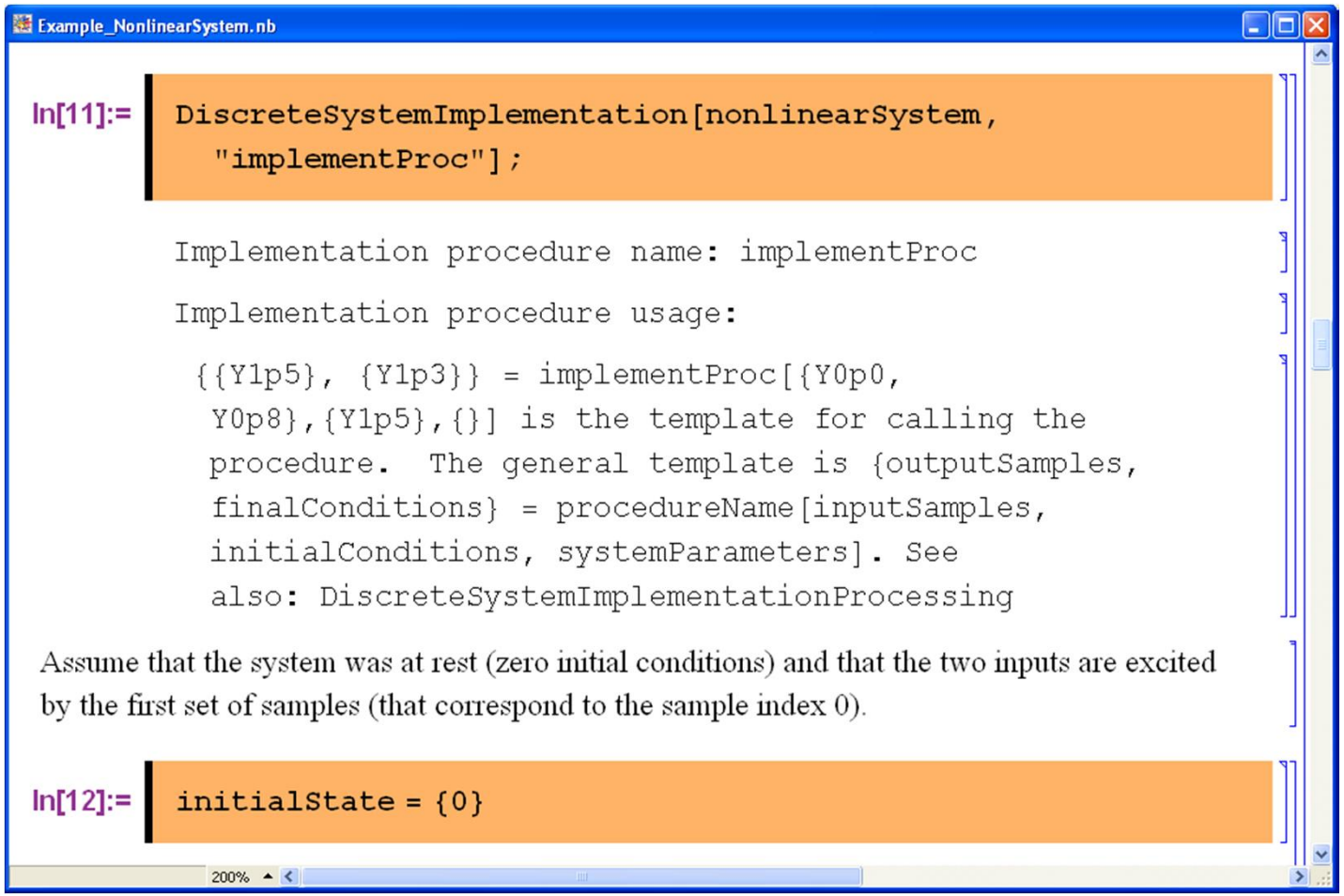
SequencePlot plots outSeq.

In[10]:=
```
SequencePlot[outSeq, PlotRange → {0, gain}];
```

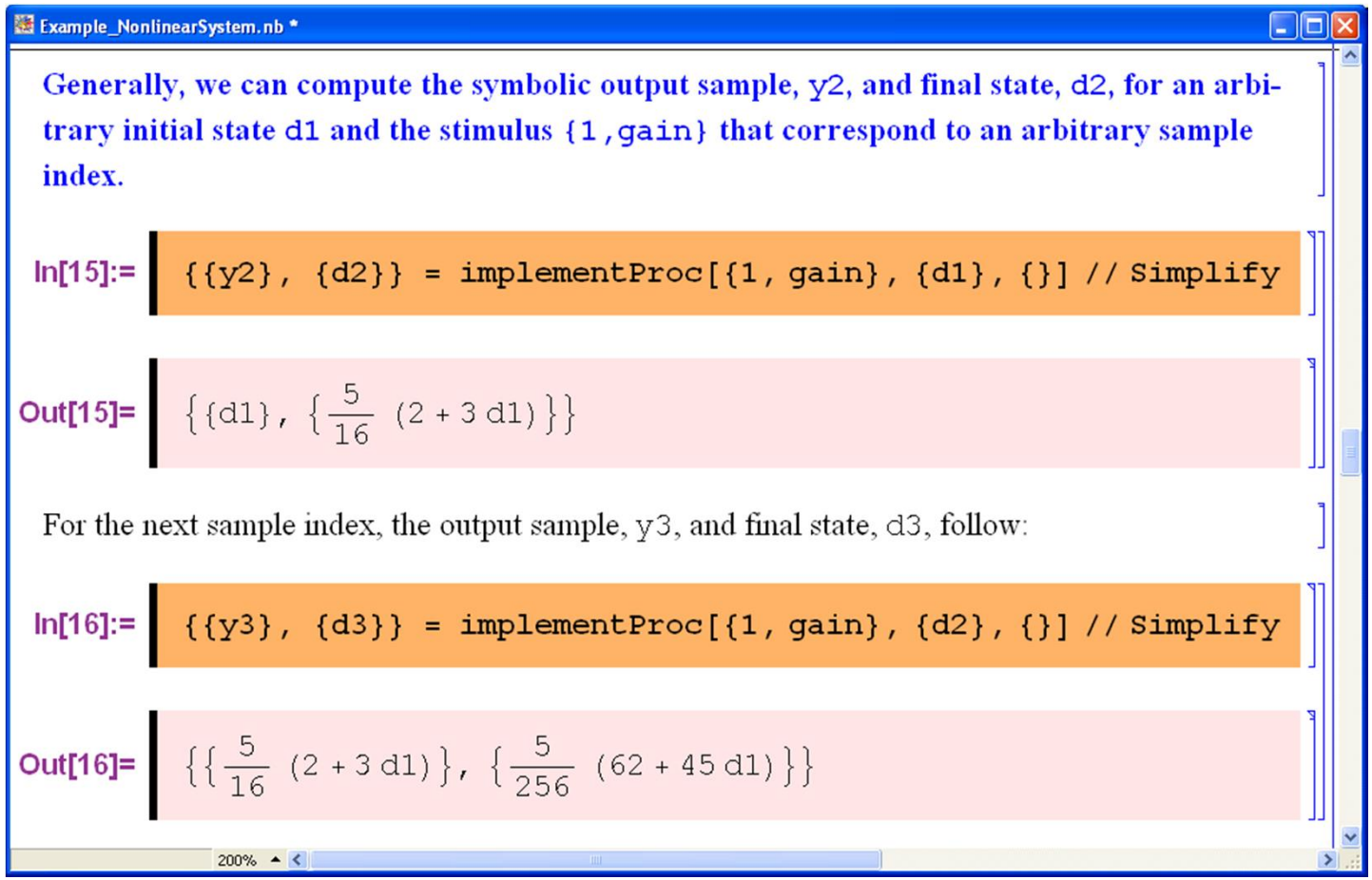


150%

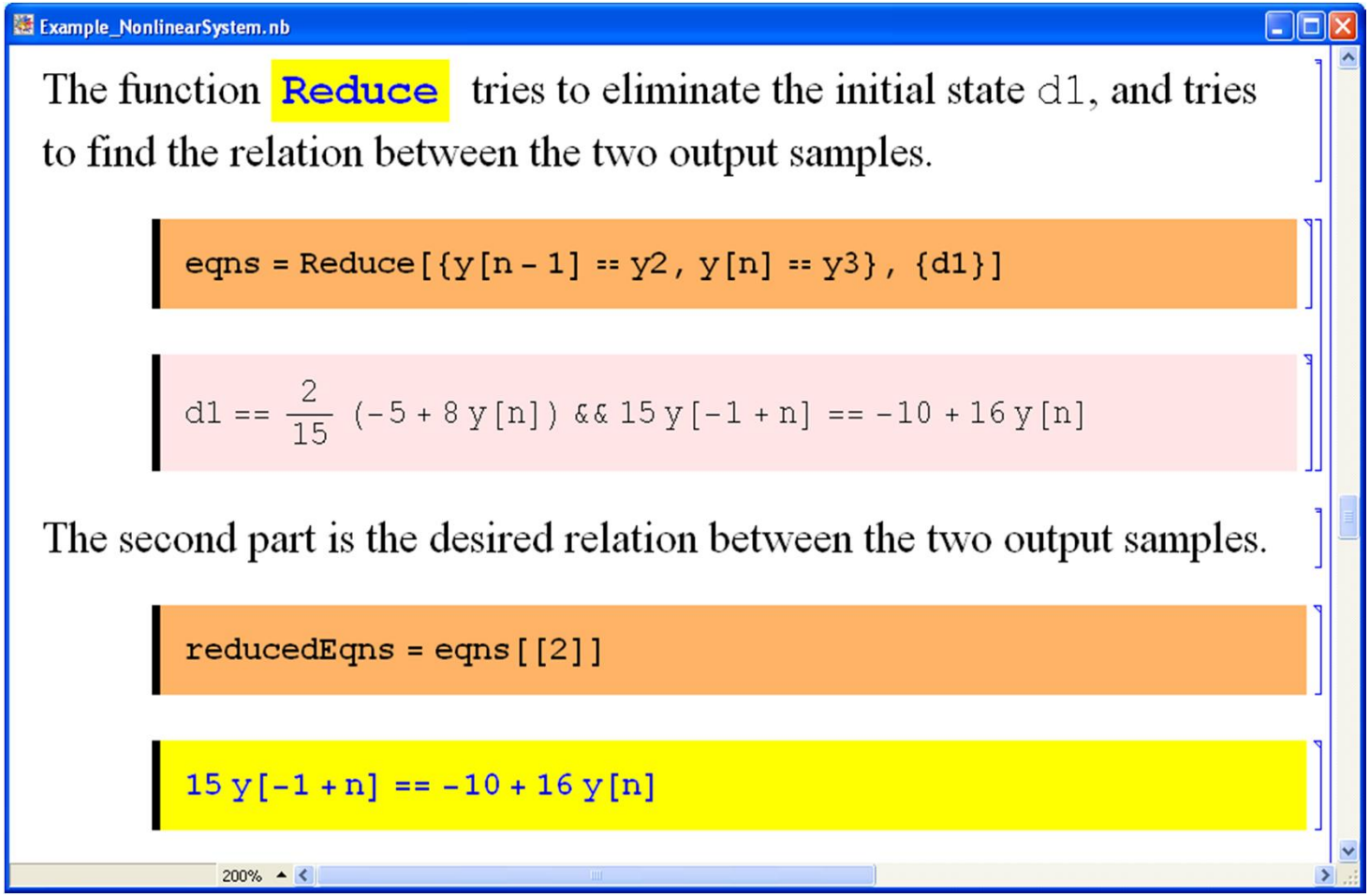
Can we find a closed-form expression in terms of the sample index?



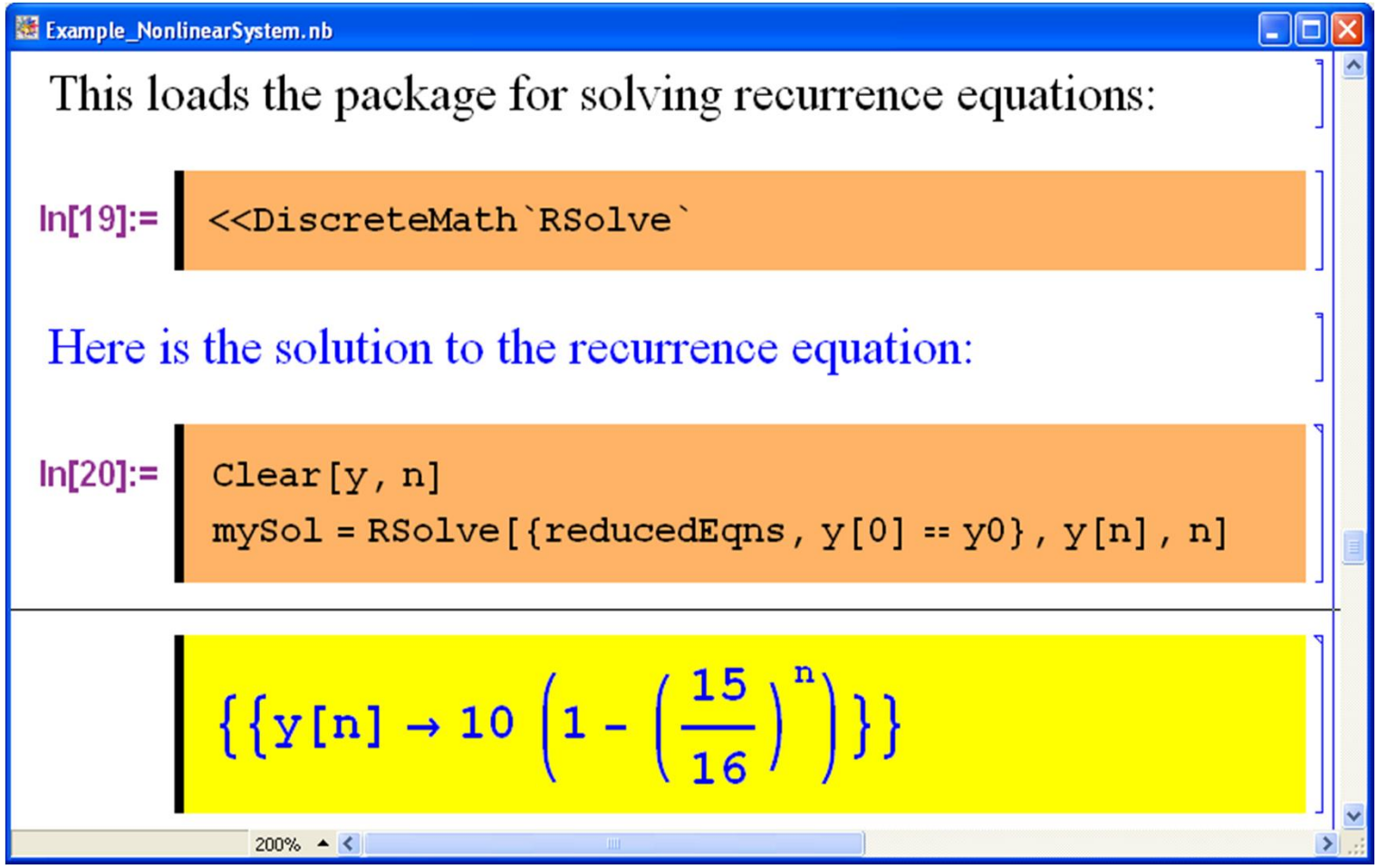
Compute symbolic output sample



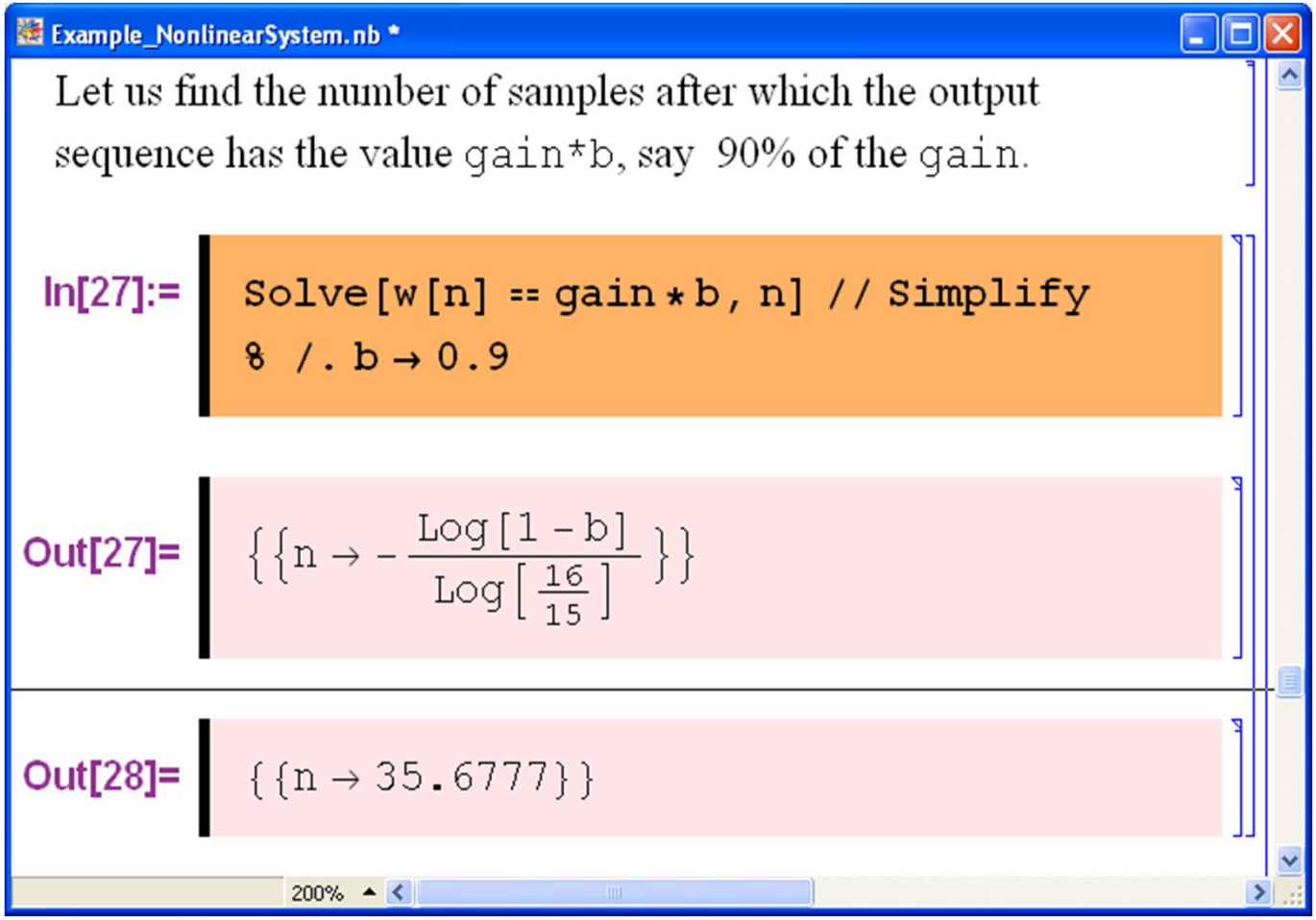
Find relation between the two output samples



Solve the recurrence equation



Optimization



- Promena učestanosti odabiranja
- Smanjenje i povećanje učestanosti odabiranja
- Decimacija
- Interpolacija
- Polifazna dekompozicija
- Projektovanje višestepenih sistema

Zašto primena različitih frekvencija odabiranja?

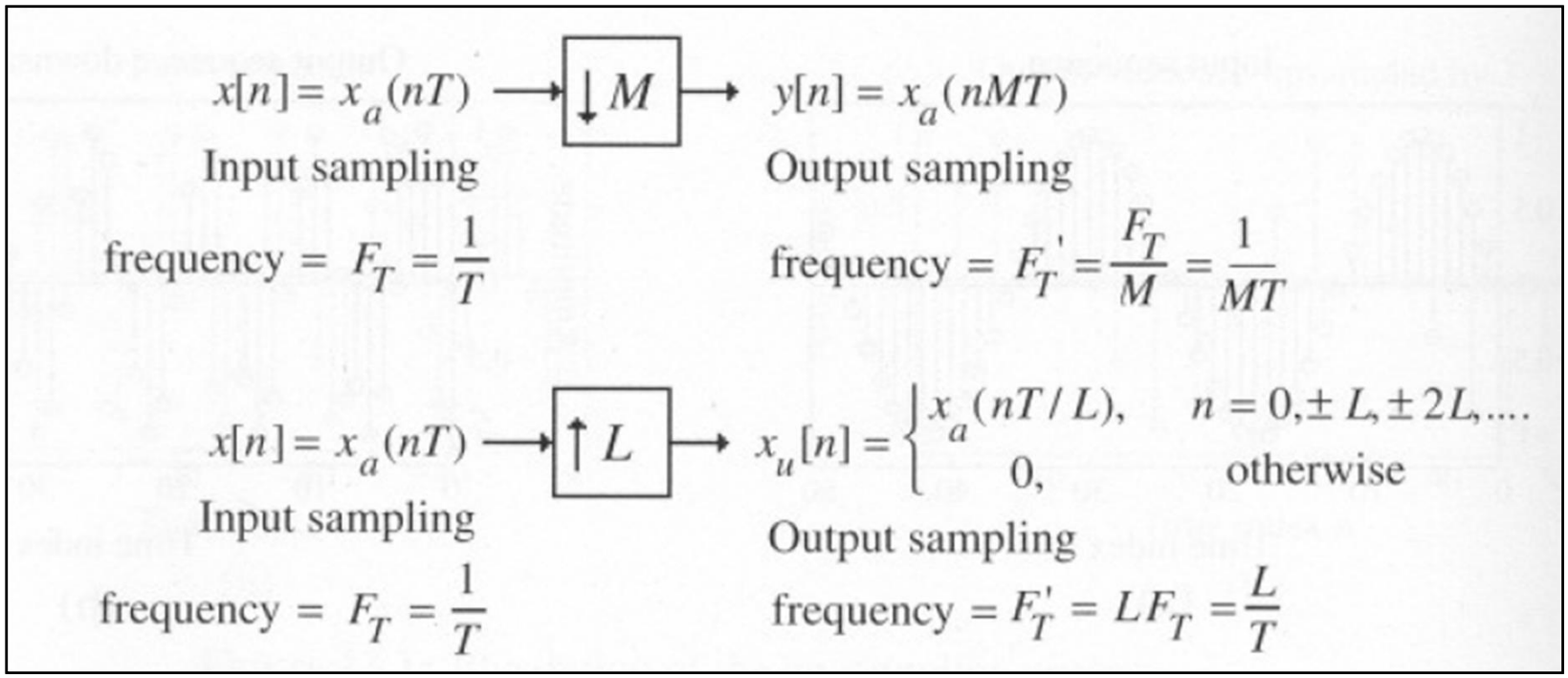
Prednosti:

- Manje računskih operacija
- Manje memorije za koeficijente i smeštanje prethodnih stanja
- Filtrima se snižava red
- Manji su efekti konačne dužine reči
- Manja je osetljivost na tačnost koeficijenata filtra

Primeri *multirate* sistema

- Sistemi sa više različitih učestanosti odabiranja se nazivaju *multirate* sistemi
- Digitalni audio signal:
  (1) prenos na 32 kHz, (2) zapis na CD-u na 44.1 kHz, (3) digitalne audio trake na 48 kHz
- Digitalni TV signal – različiti standardi
- Digitalni signal može da se pretvori u analogni i da se ponovo generiše digitalni za drugu učestanost odabiranja
- Digitalni signal sa jednom učestanošću odabiranja direktno se pretvori u drugi digitalni signal

Osnovne operacije za promenu učestanosti odabiranja



Down-sampler

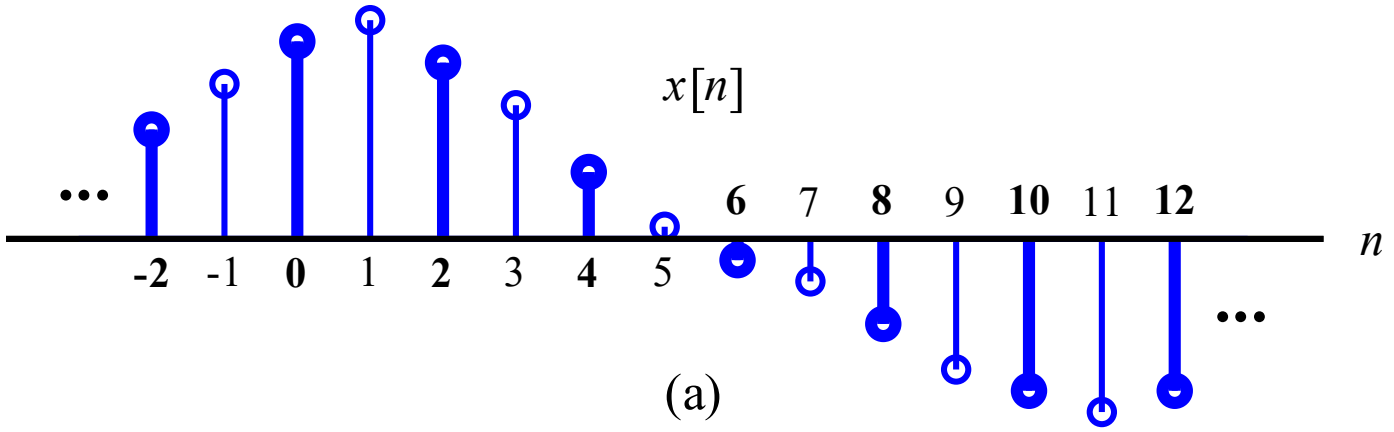
- **Down-sampler** ↓M – vrši se redukcija učestanosti odabiranja sa celobrojnim faktorom $\boldsymbol{M}$

$$x[n] = x_{\mathrm{a}}(t)\big|_{t=nT} = x_{\mathrm{a}}(nT)$$

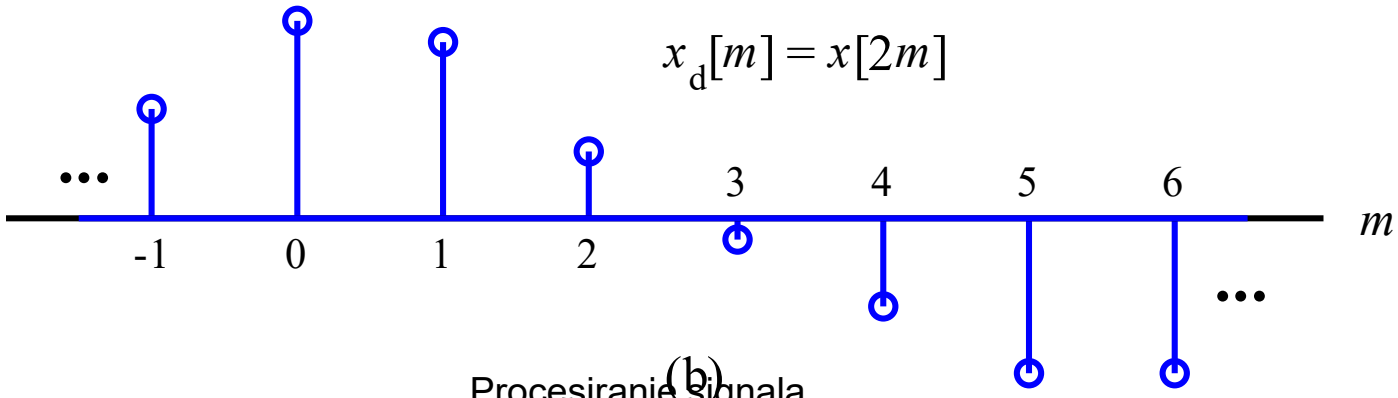
$$x_{\mathrm{d}}[n] = x[Mn] = x_{\mathrm{a}}(t)\big|_{t=nMT} = x_{\mathrm{a}}(t)\big|_{t=nT'} = x_{\mathrm{a}}(nT')$$

$$T' = MT \qquad F_T^{'} = F_T / M$$

Down-sampling za M=2



(a)



(b)

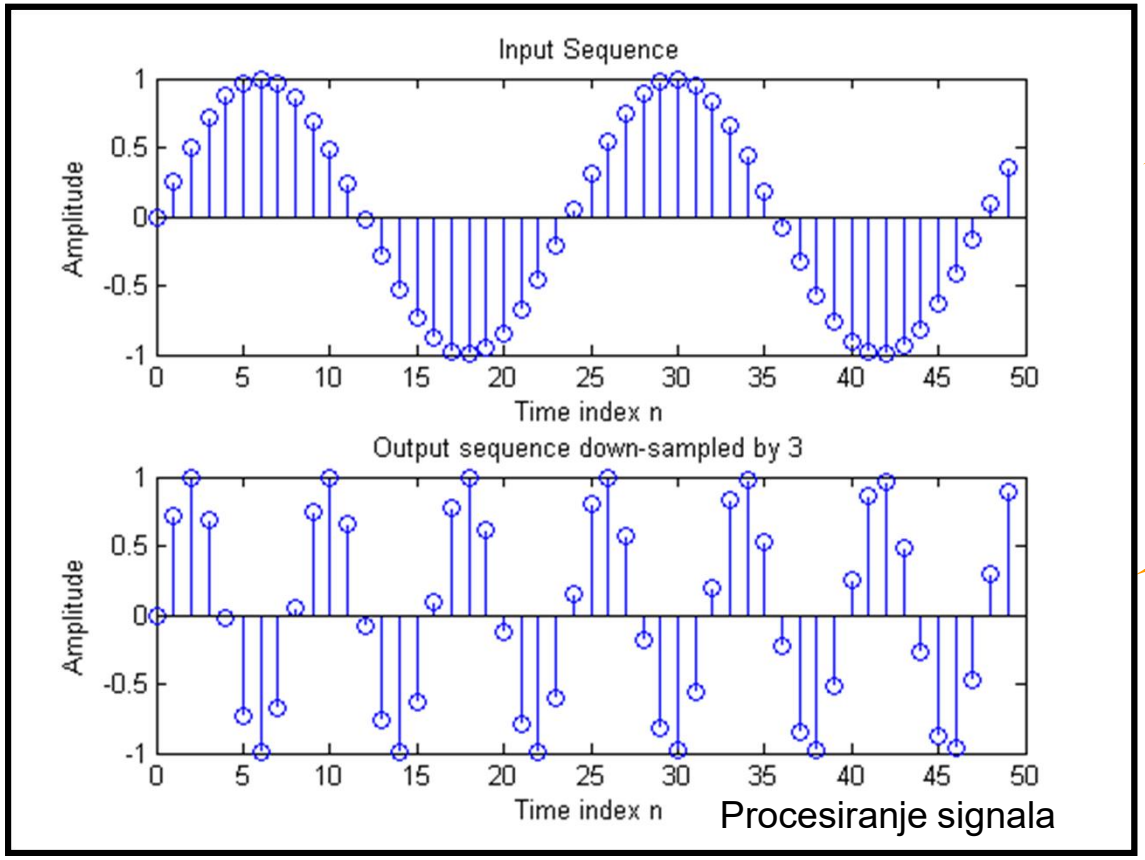
Program 13_2

Inicijalizacija

Down-sampling

```
N = 50;
M = 3;
fo = 0.042;

n = 0 : N-1;
m = 0 : N*M-1;
x = sin(2*pi*fo*m);
y = x([1 : M : length(x)]);
subplot(2,1,1)
stem(n, x(1:N));
title('Input Sequence');
xlabel('Time index n');ylabel('Amplitude');
subplot(2,1,2)
stem(n, y);
title(['Output sequence down-sampled by ',num2str(M)]);
ylabel('Amplitude');xlabel('Time index n');
```

Program 13_2



Ulazna sekvenca

Sekvenca posle down-sampling

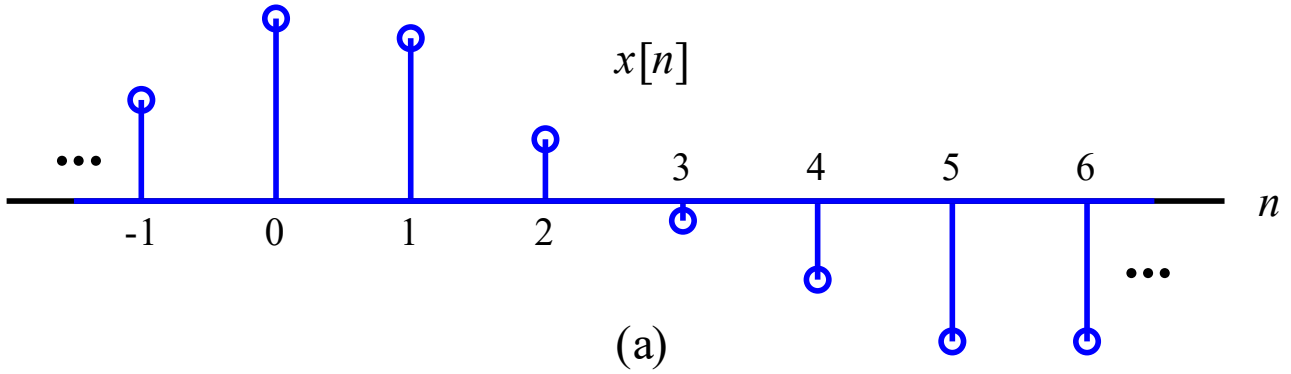
Up-sampler

- **Up-sampler** ↑L – vrši se povećanje učestanosti odabiranja sa celobrojnim faktorom L

$$x_{\mathrm{u}}[n] = \begin{cases} x[n/L], & n = 0, \pm L, \pm 2L, \cdots \\ 0, & \text{drugde} \end{cases}$$

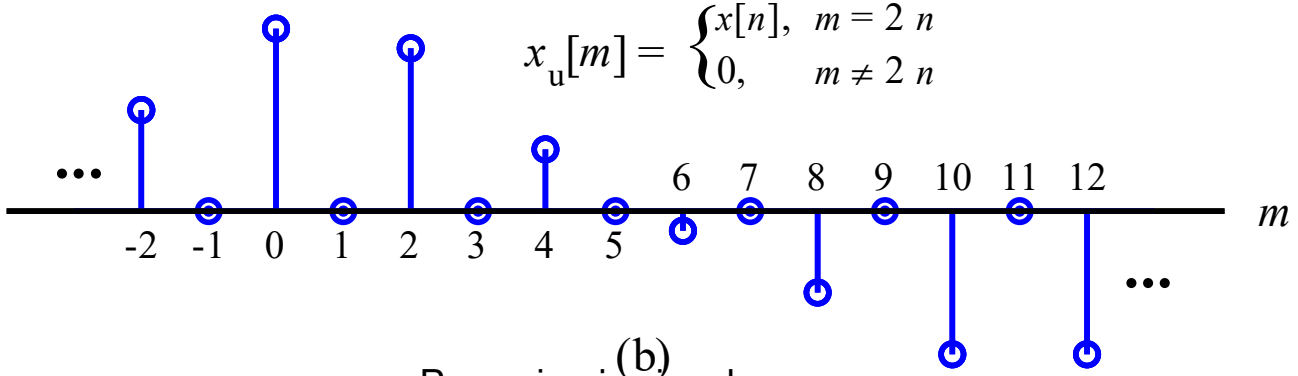
$$T' = T / L \qquad F_T^{'} = L\, F_T$$

Up-sampling za L=2



(a)

$$x_u[m] = \begin{cases} x[n], & m = 2n \\ 0, & m \neq 2n \end{cases}$$



(b)

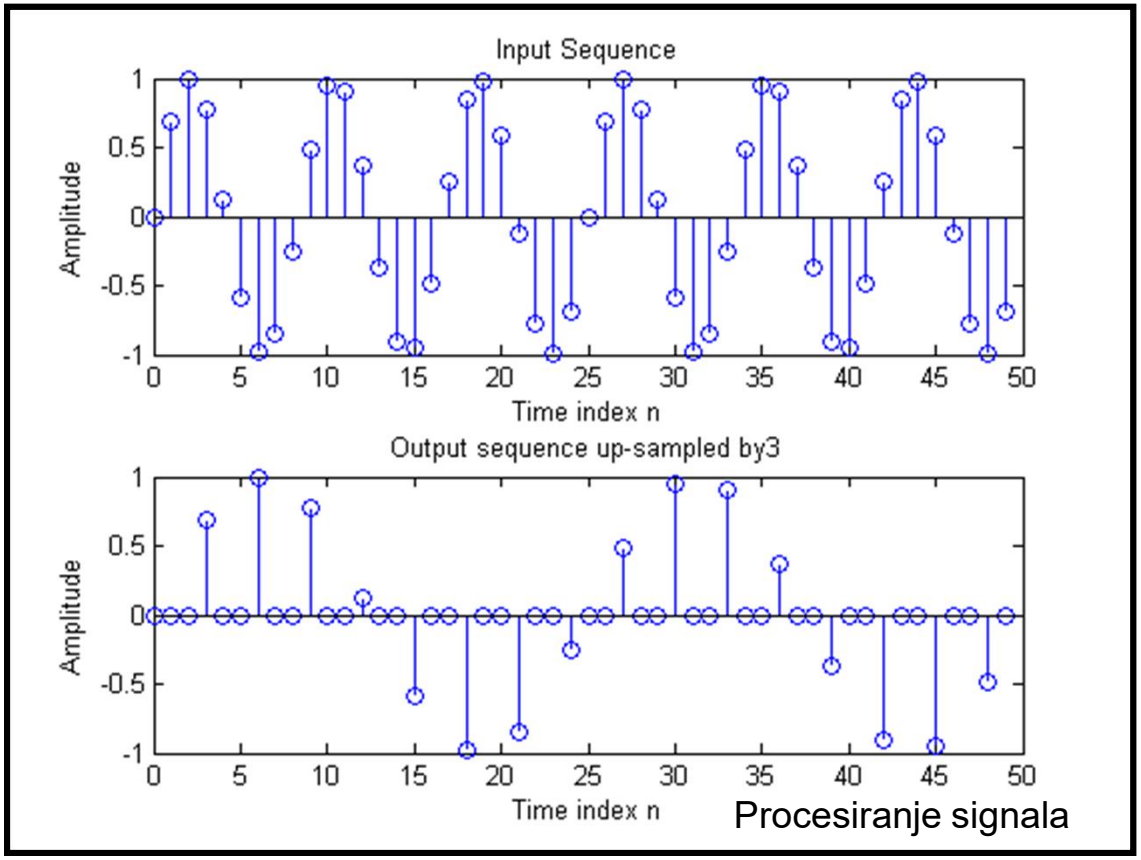
Program 13_1

Inicijalizacija

Up-sampling

```
N = 50;
fo = 0.12;
L = 3;

n = 0:N-1;
x = sin(2*pi*fo*n);
y = zeros(1, L*length(x));
y([1: L: length(y)]) = x;
subplot(2,1,1)
stem(n,x);
title('Input Sequence');
xlabel('Time index n'); ylabel('Amplitude');
subplot(2,1,2)
stem(n,y(1:length(x)));
title(['Output sequence up-sampled by', num2str(L)]);
xlabel('Time index n'); ylabel('Amplitude');
```

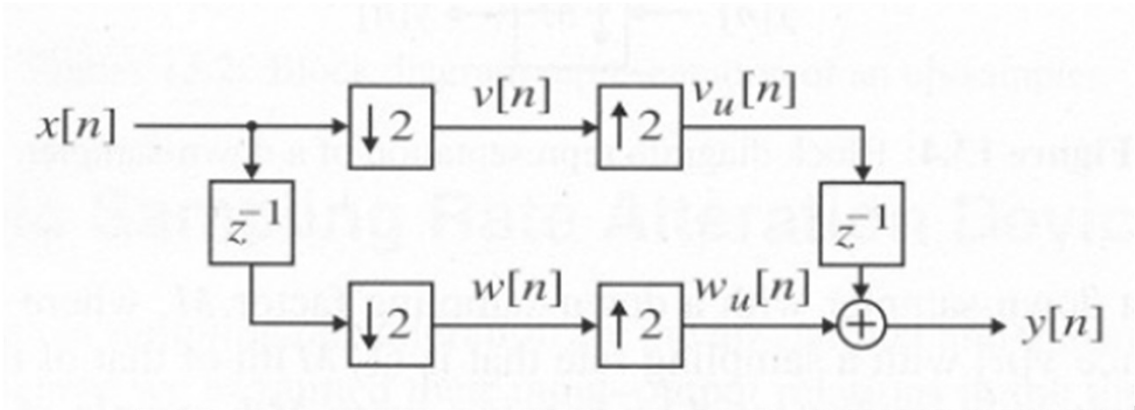
Program 13_1



Ulazna sekvenca

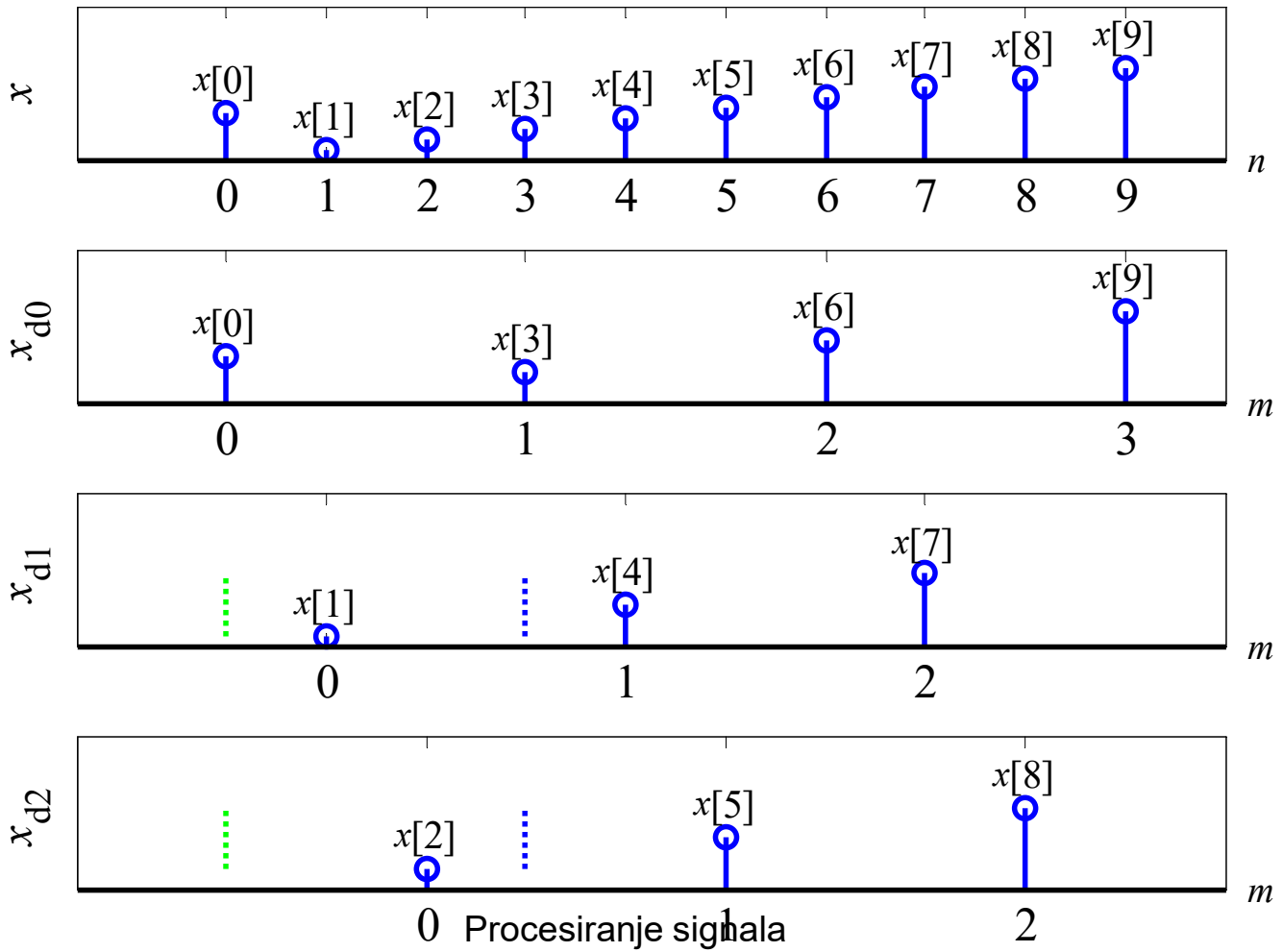
Sekvenca posle up-sampling

Perfektna rekonstrukcija

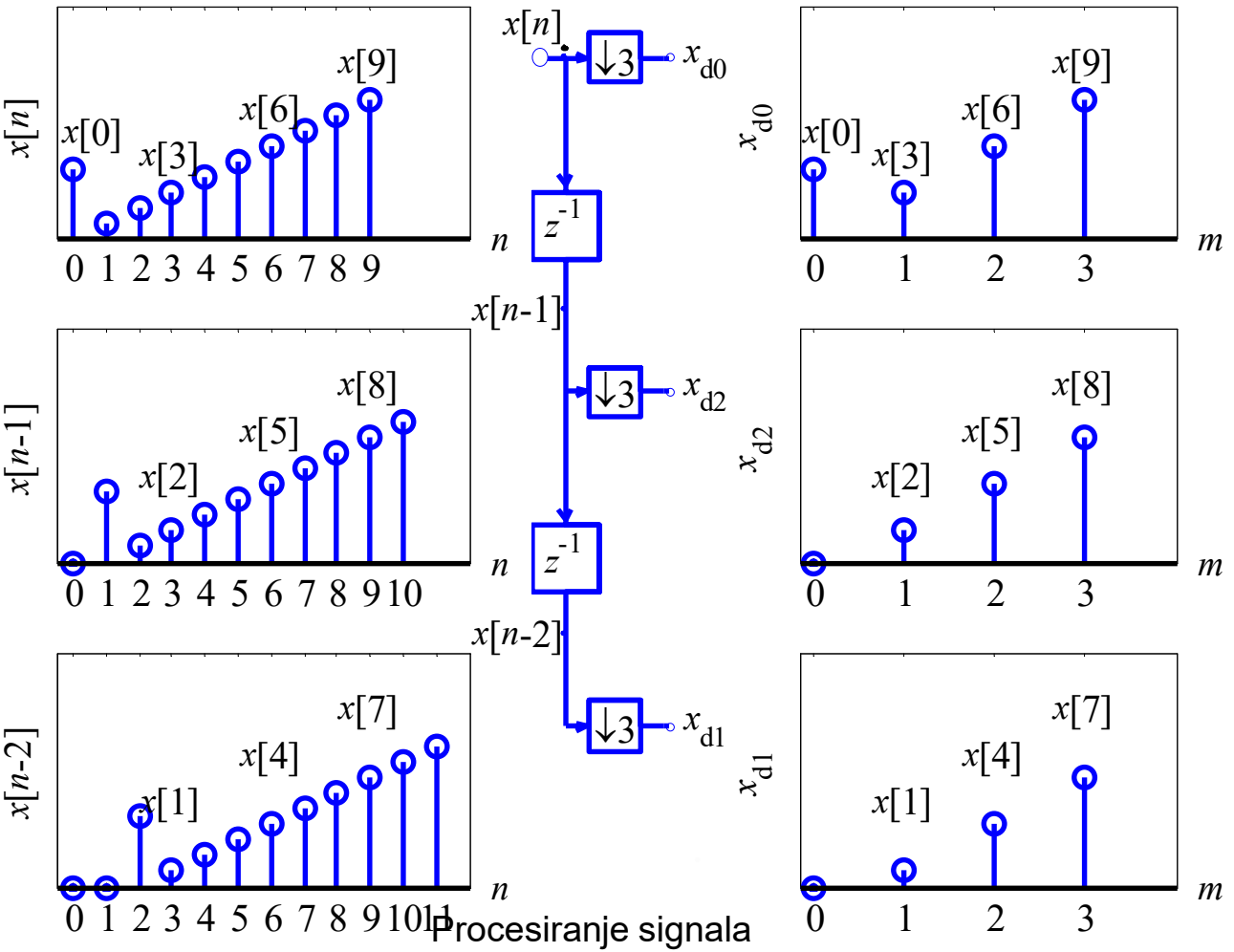


n :	0	1	2	3	4	5	6	7	8
$x[n]$:	$x[0]$	$x[1]$	$x[2]$	$x[3]$	$x[4]$	$x[5]$	$x[6]$	$x[7]$	$x[8]$
$v[n]$:	$x[0]$	$x[2]$	$x[4]$	$x[6]$	$x[8]$	$x[10]$	$x[12]$	$x[14]$	$x[16]$
$w[n]$:	$x[-1]$	$x[1]$	$x[3]$	$x[5]$	$x[7]$	$x[9]$	$x[11]$	$x[13]$	$x[15]$
$v_u[n]$:	$x[0]$	0	$x[2]$	0	$x[4]$	0	$x[6]$	0	$x[8]$
$w_u[n]$:	$x[-1]$	0	$x[1]$	0	$x[3]$	0	$x[5]$	0	$x[7]$
$v_u[n-1]$:	0	$x[0]$	0	$x[2]$	0	$x[4]$	0	$x[6]$	0
$y[n]$:	$x[-1]$	$x[0]$	$x[1]$	$x[2]$	$x[3]$	$x[4]$	$x[5]$	$x[6]$	$x[7]$

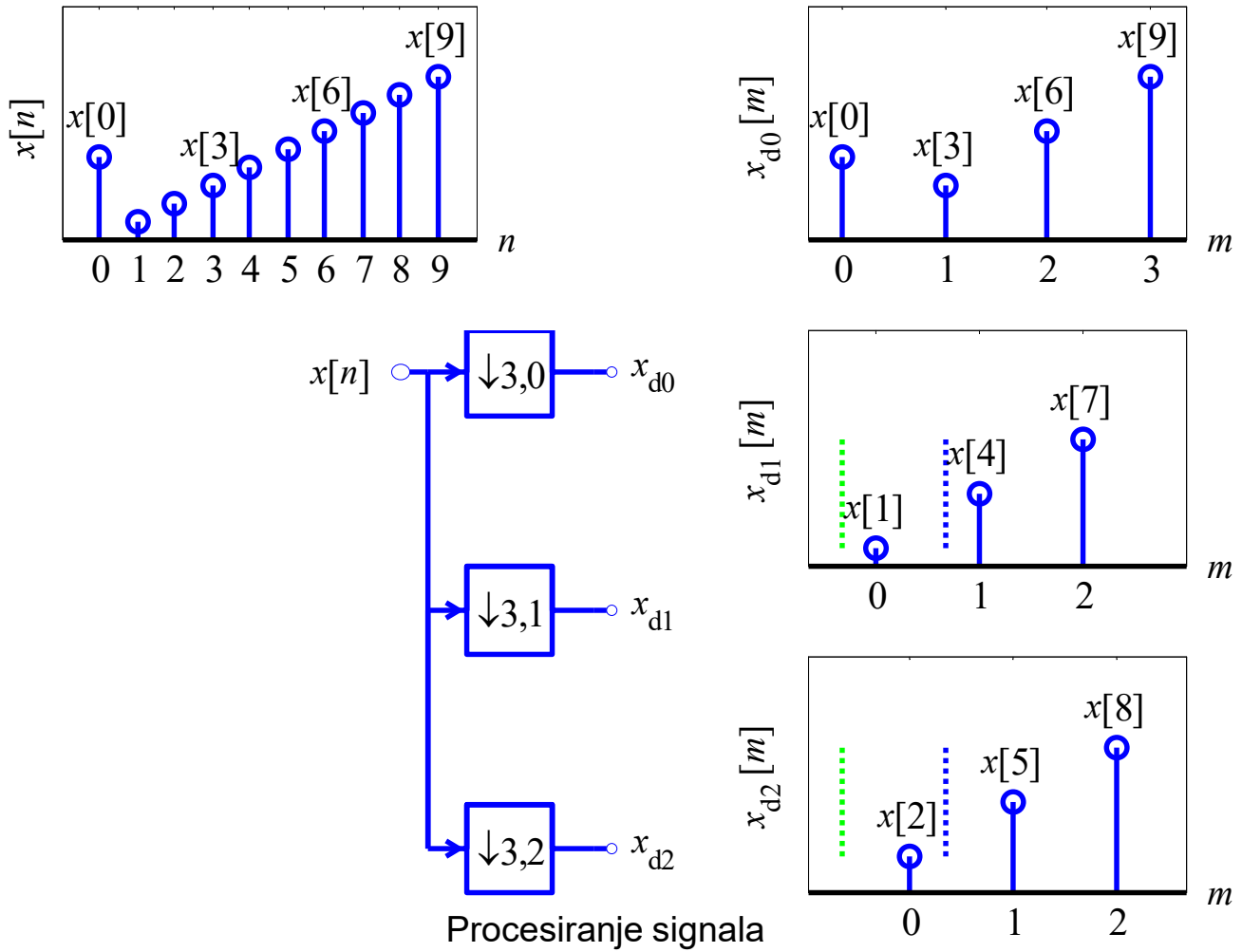
Down-sampling sa ofsetom



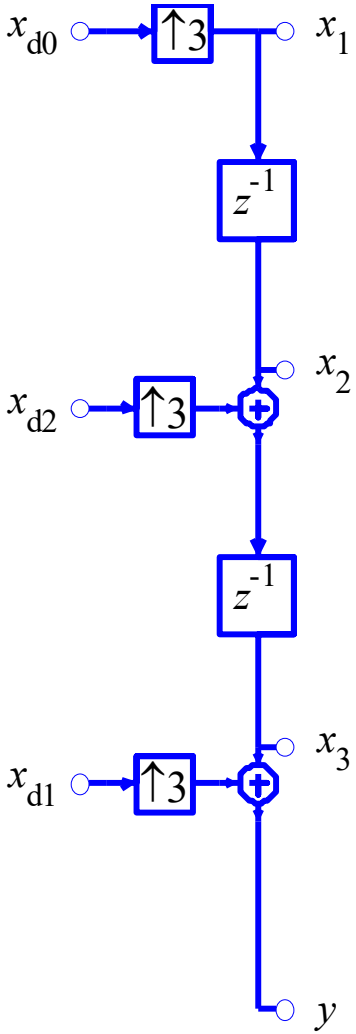
Down-sampling sa kašnjenjem

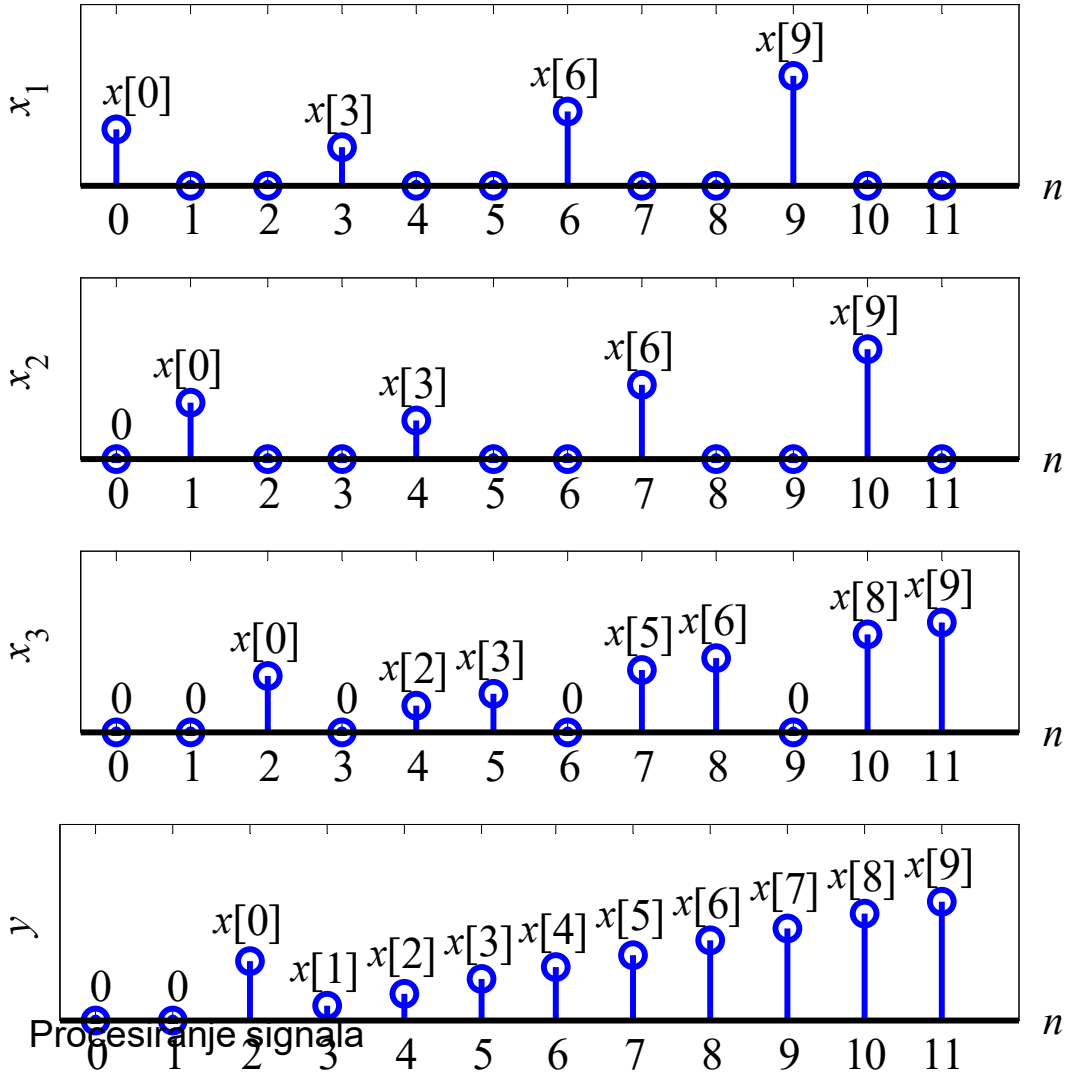


Realizacija sa ofsetom

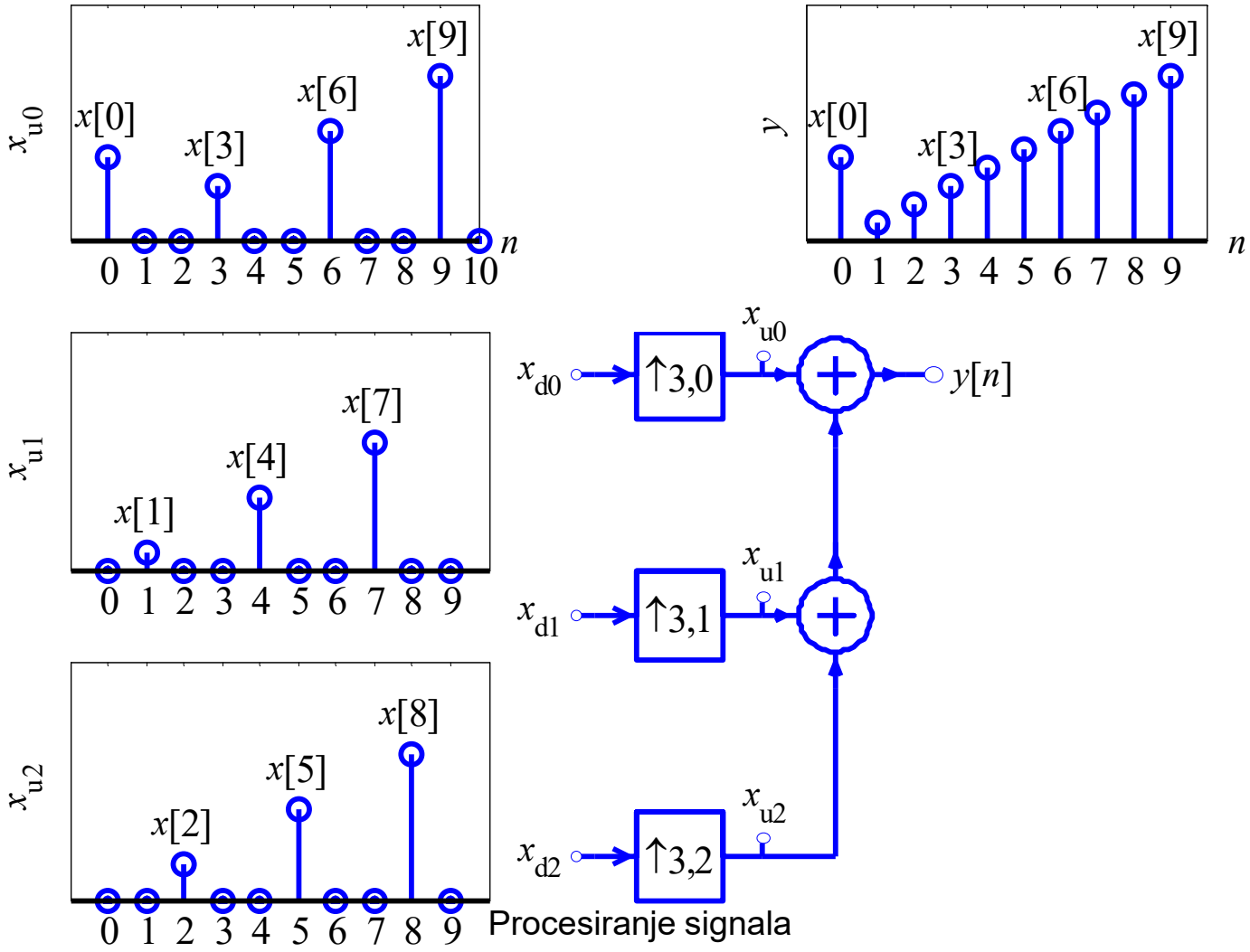


Up-sampling sa kašnjenjem





Rekonstrukcija sa ofsetom



Aliasing – folding
za pobude
periodičnim sekvencama

$M = 5$	$x[n]$ pre down-sampling	$x_d[n]$ posle down-sampling	$y[m]$ equivalent
	$x_1[n] = \sin(2\pi\ 0.04\ n + \varphi)$	$x_{d1}[m] = \sin(2\pi\ 0.2\ m + \varphi)$	$\sin(2\pi\ 0.2\ m + \varphi)$
	$x_2[n] = -\sin(2\pi\ 0.16\ n - \varphi)$	$x_{d2}[m] = -\sin(2\pi\ 0.8\ m - \varphi)$	$\sin(2\pi\ 0.2\ m + \varphi)$
	$x_3[n] = \sin(2\pi\ 0.24\ n + \varphi)$	$x_{d3}[m] = \sin(2\pi\ 1.2\ m + \varphi)$	$\sin(2\pi\ 0.2\ m + \varphi)$

$$\{x[n]\} = \{\sin(2\pi f n + \varphi)\}$$

$$0 < f < 1/2M$$

Karakteristike u frekvencijskom domenu za L=2

$$x_{\mathrm{u}}[n]=\begin{cases} x[n/2], & n=0,\pm 2,\pm 4,\cdots \\ 0, & \text{drugde} \end{cases}$$

$$X_{\mathrm{u}}(z)=\sum_{n=-\infty}^{n=\infty} x_{\mathrm{u}}[n/2]z^{-n}=\sum_{\substack{n=-\infty \\ n \text{ even}}}^{n=\infty} x_{\mathrm{u}}[n/2]z^{-n}$$

$$X_{\mathrm{u}}(z)=\sum_{m=-\infty}^{m=\infty} x_{\mathrm{u}}[m]z^{-2m}=X(z^2)$$

Up-sampler: karakteristike u frekvencijskom domenu

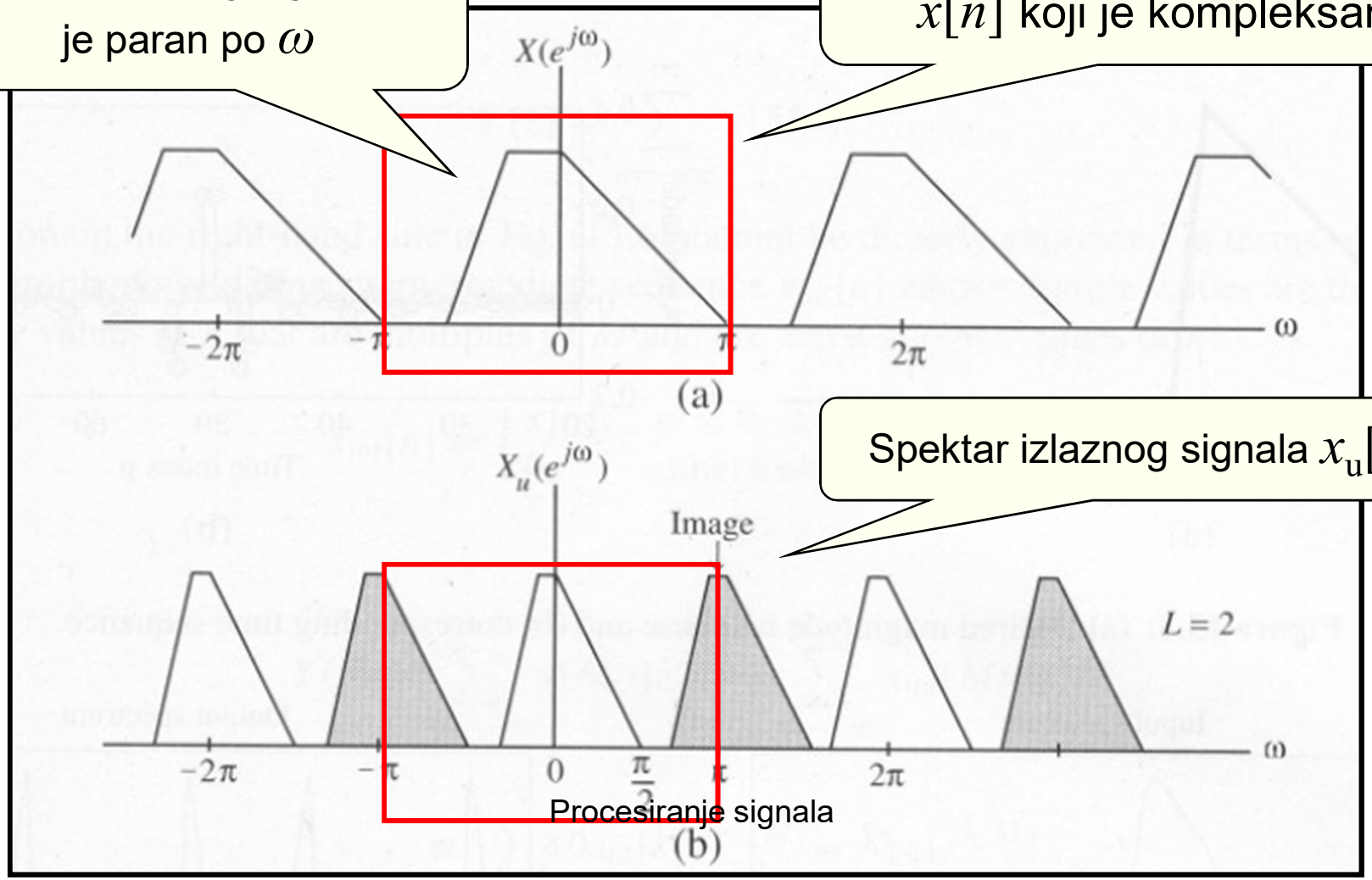
$$x_{\mathrm{u}}[n] = \begin{cases} x[n/L], & n = 0, \pm L, \pm 2L, \cdots \\ 0, & \text{drugde} \end{cases}$$

$$X_{\mathrm{u}}(z) = X(z^{L})$$

Up-sampling u frekvencijskom domenu

DTFT realnog signala je paran po ω

Spektar ulaznog signala $x[n]$ koji je kompleksan



Spektar izlaznog signala $x_u[n]$

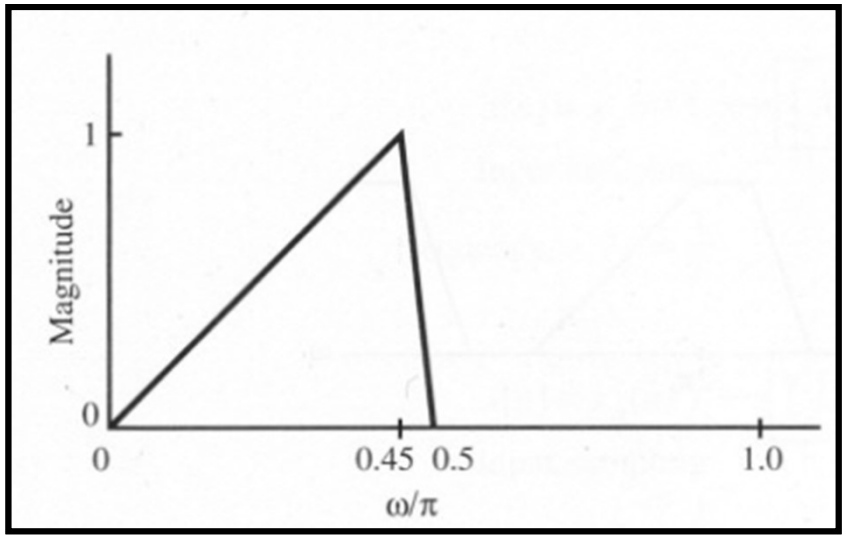
Program 13_3

```
freq = [0 0.45 0.5 1];
mag = [0 1 0 0];
x = fir2(99, freq, mag);
[Xz, w] = freqz(x, 1, 512);
plot(w/pi, abs(Xz)); grid
xlabel('\omega/\pi'); ylabel('Magnitude');
title('Input spectrum');

L = 5;
disp(['Up-sampling factor = ' num2str(L)])
y = zeros(1, L*length(x));
y([1: L: length(y)]) = x;
[Yz, w] = freqz(y, 1, 512);
plot(w/pi, abs(Yz)); grid
xlabel('\omega/\pi'); ylabel('Magnitude');
title('Output spectrum');
```

Ulazni signal

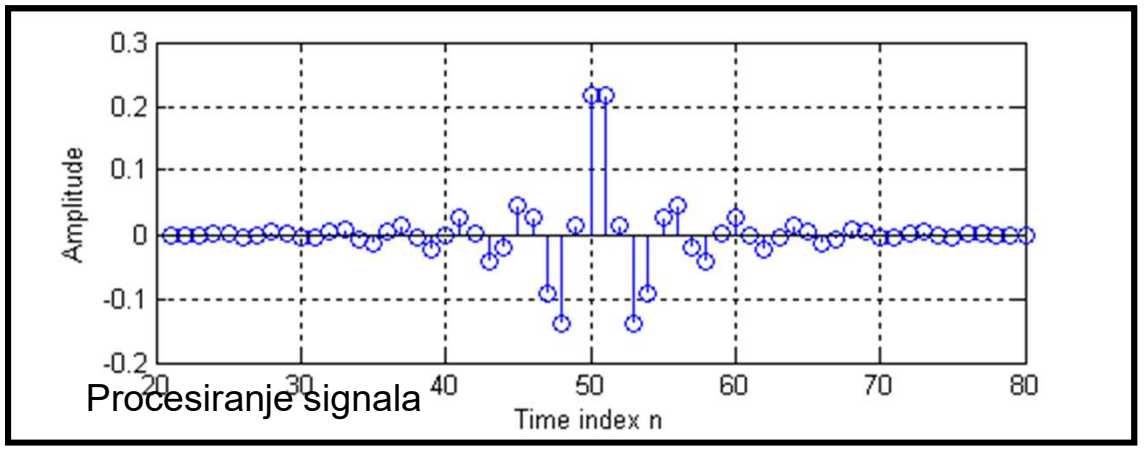
Spektar

Program 13_3

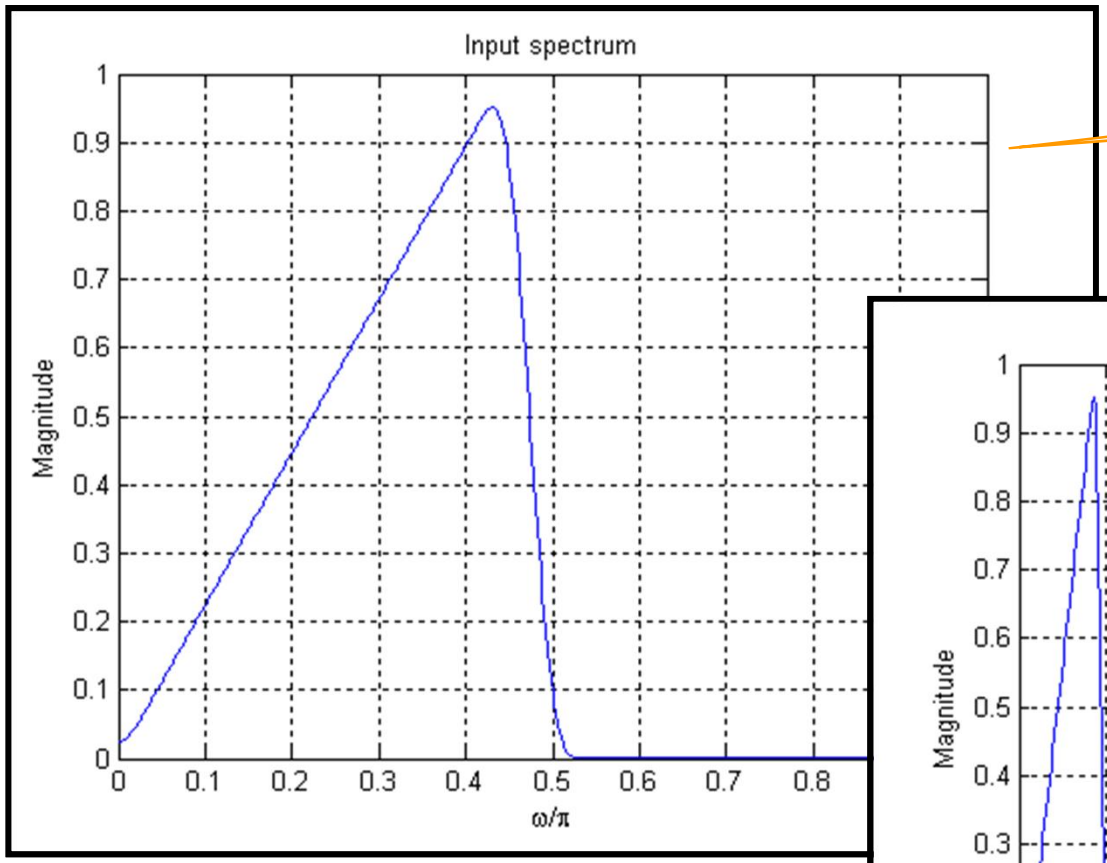


Generisati signal sa spektrom

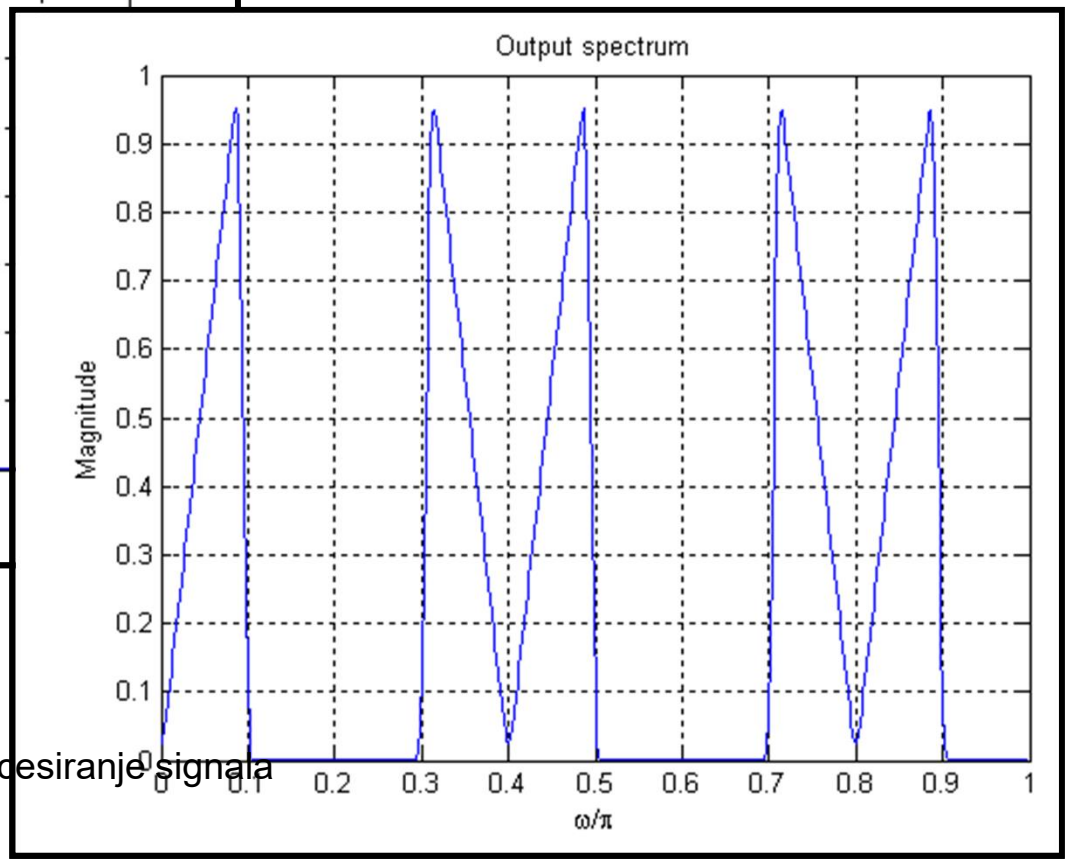
Signal koji ima definisani spektar



Program 13_3



Spektar ulaznog signala



Spektar izlaznog signala

Down-sampler: karakteristike u frekvencijskom domenu (1)

$$Y(z) = \sum_{n=-\infty}^{n=\infty} x[Mn]\, z^{-n}$$

$$x_{\text{int}}[n] = \begin{cases} x[n], & n = 0, \pm M, \pm 2M, \cdots \\ 0, & \text{drugde} \end{cases}$$

$$Y(z) = \sum_{n=-\infty}^{n=\infty} x_{\text{int}}[Mn]\, z^{-n} = \sum_{k=-\infty}^{k=\infty} x_{\text{int}}[k]\, z^{-k/M} = X_{\text{int}}(z^{1/M})$$

Down-sampler: karakteristike u frekvencijskom domenu (2)

$$x_{\text{int}}[n] = c[n]\, x[n]$$

$$Y(z) = X_{\text{int}}(z^{1/M})$$

$$c[n] = \begin{cases} 1, & n = 0, \pm M, \pm 2M, \cdots \\ 0, & \text{drugde} \end{cases}$$

$$c[n] = \frac{1}{M} \sum_{k=0}^{M-1} W_M^{kn}, \quad W_M = \mathrm{e}^{-j2\pi/M}$$

$$X_{\text{int}}(z) = \frac{1}{M} \sum_{k=0}^{M-1} X(z\, W_M^{-k})$$

$$Y(z) = \frac{1}{M} \sum_{k=0}^{M-1} X(z^{1/M} W_M^{-k})$$

Down-sampler: karakteristike u frekvencijskom domenu (3)

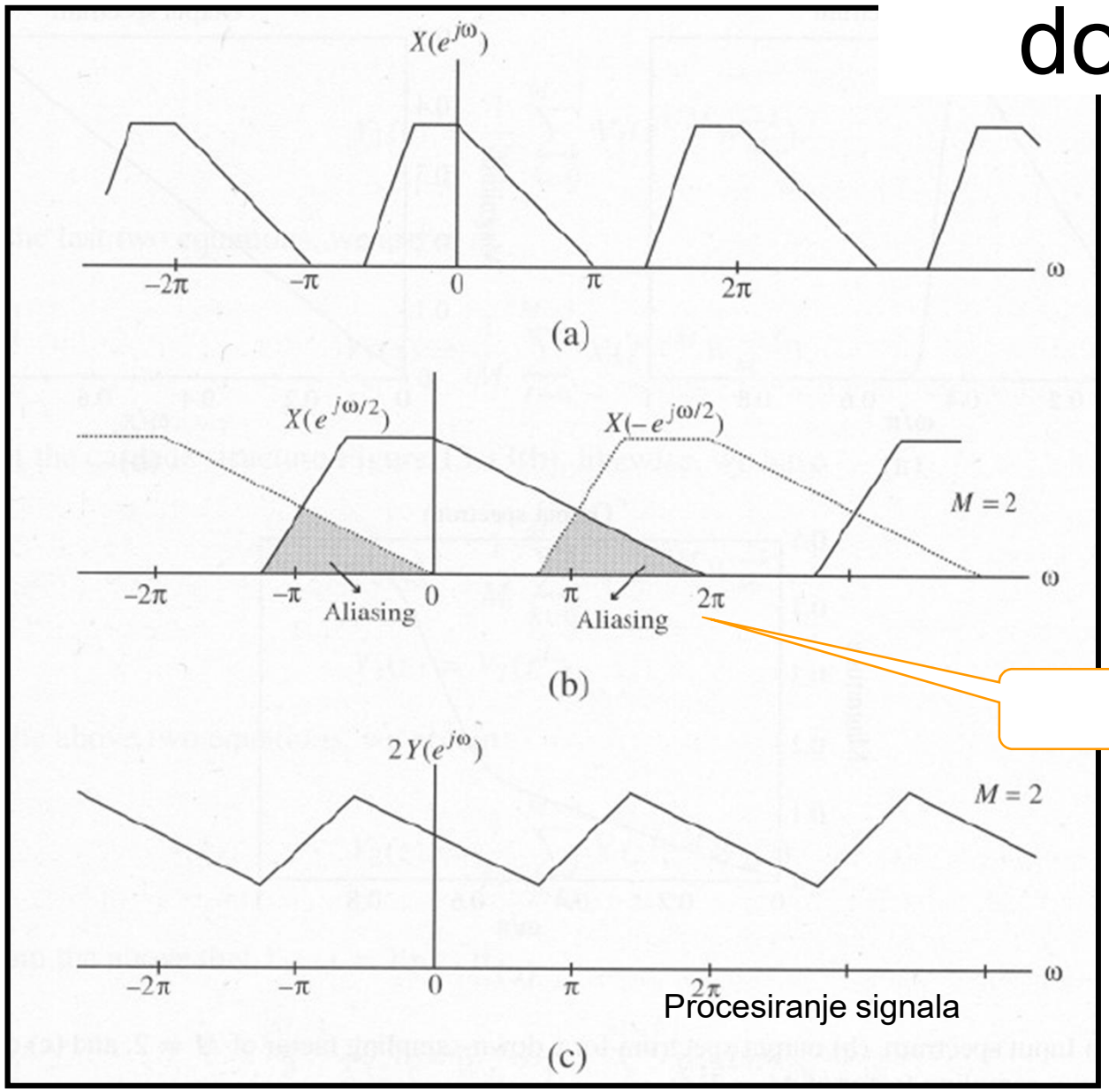
$$Y(z) = \frac{1}{M}\sum_{k=0}^{M-1} X(z^{1/M} W_M^{-k})$$

$$Y(\mathrm{e}^{j\omega}) = \frac{1}{2}\left(X(\mathrm{e}^{j\omega/2}) + X(-\mathrm{e}^{j\omega/2})\right)$$

$$X(-\mathrm{e}^{j\omega/2}) = X(\mathrm{e}^{j(\omega-2\pi)/2})$$

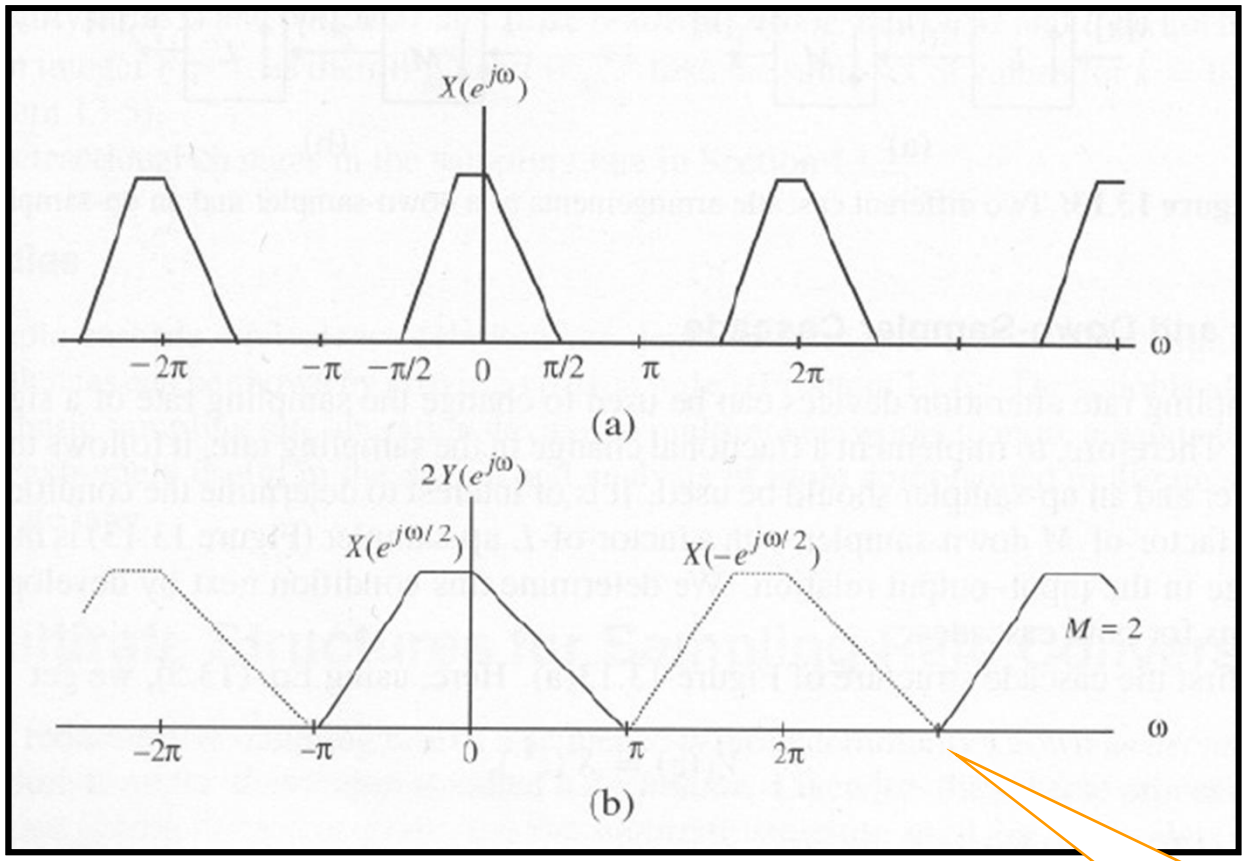
Neka je $X(\mathrm{e}^{j\omega/2})$ realna funkcija sa asimetričnim spektrom

Frekvencijski domen (4)



Aliasing

Frekvencijski domen (5)



Bez efekta aliasing

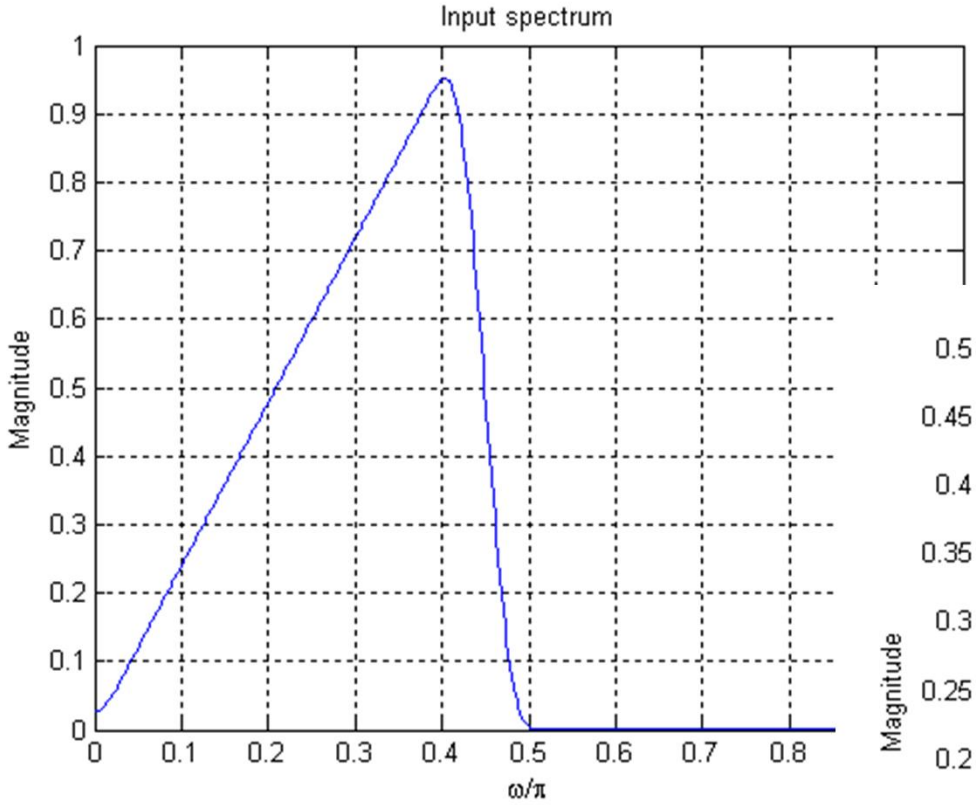
Program 13_4

```
freq = [0 0.42 0.48 1];
mag = [0 1 0 0];
x = fir2(101, freq, mag);
[Xz, w] = freqz(x, 1, 512);
plot(w/pi, abs(Xz)); grid
xlabel('\omega/\pi'); ylabel('Magnitude');
title('Input spectrum');

M = 2;
y = x([1: M: length(x)]);
[Yz, w] = freqz(y, 1, 512);
plot(w/pi, abs(Yz)); grid
xlabel('\omega/\pi'); ylabel('Magnitude');
title(['Output spectrum, M = ' num2str(M)]);
```

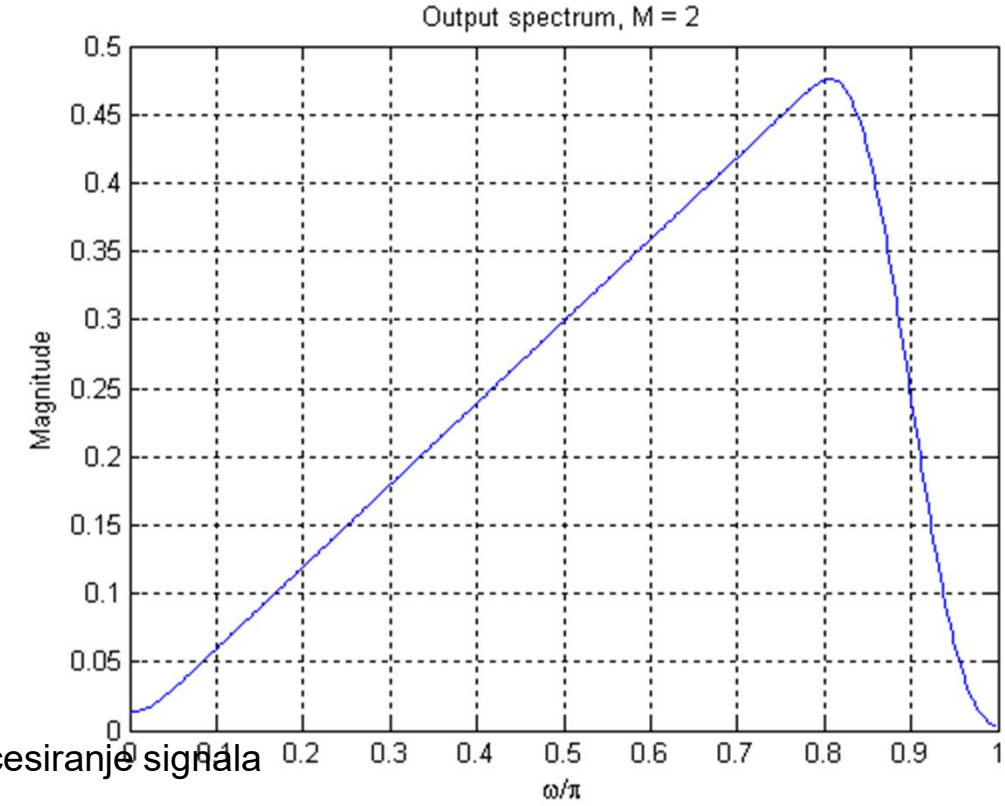
Ulazni signal

Spektar

Program 13_4

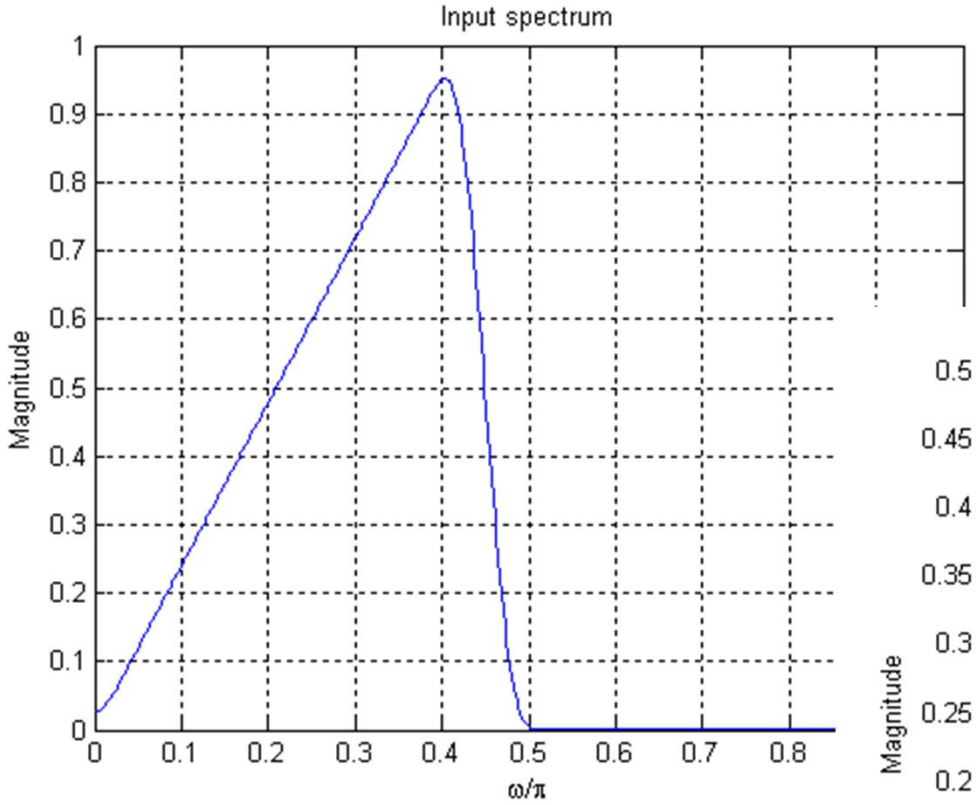


Spektar ulaznog signala

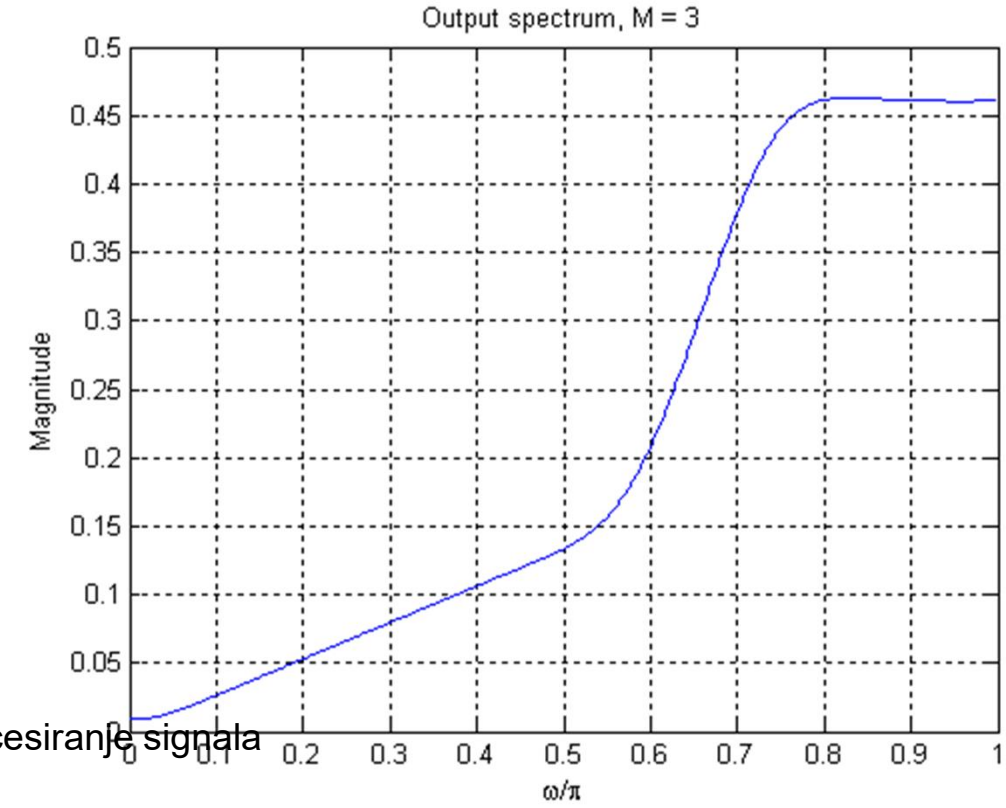


Spektar izlaznog signala
M=2

Program 13_4

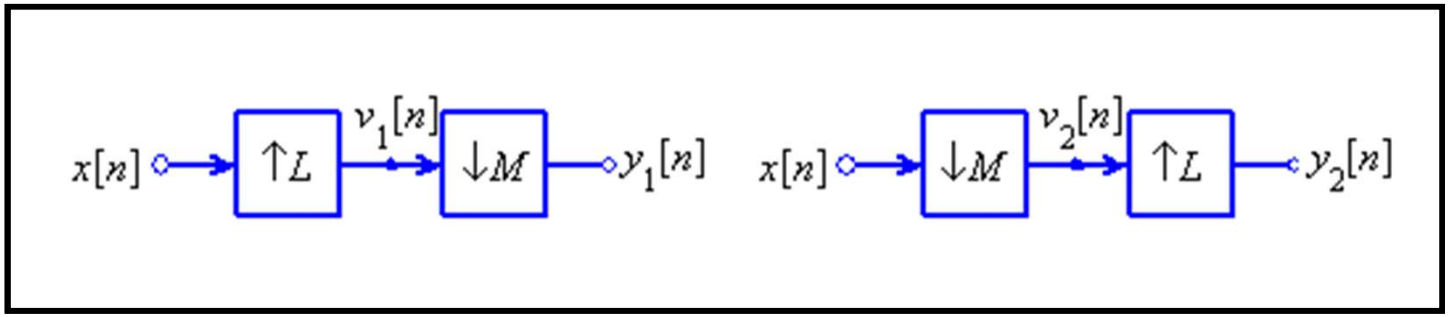


Spektar ulaznog signala



Spektar izlaznog signala
M=3

Kaskadna veza



$$V_1(z) = X(z^L)$$

$$V_2(z) = \frac{1}{M}\sum_{k=0}^{M-1} X(z^{1/M} W_M^{-k})$$

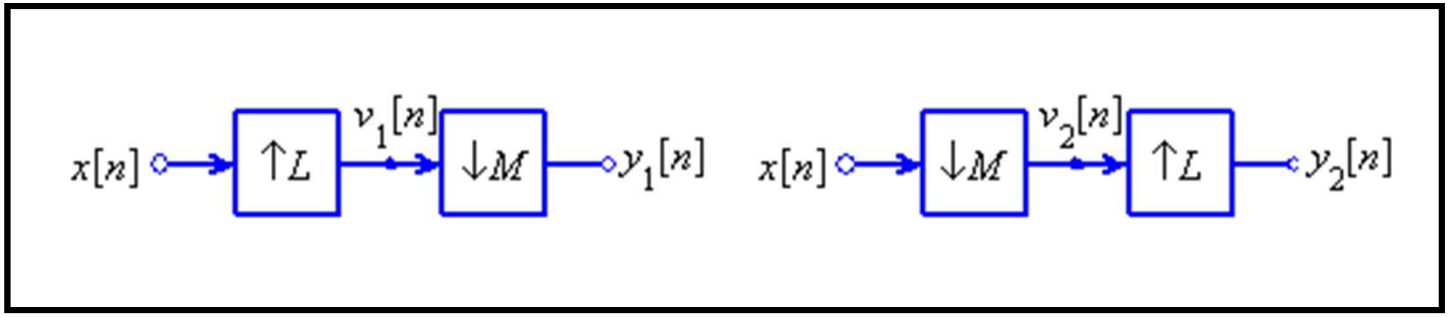
$$Y_1(z) = \frac{1}{M}\sum_{k=0}^{M-1} V_1(z^{1/M} W_M^{-k})$$

$$Y_2(z) = V_2(z^L)$$

$$Y_1(z) = \frac{1}{M}\sum_{k=0}^{M-1} X(z^{L/M} W_M^{-kL})$$

$$Y_2(z) = \frac{1}{M}\sum_{k=0}^{M-1} X(z^{L/M} W_M^{-k})$$

Kaskadna veza



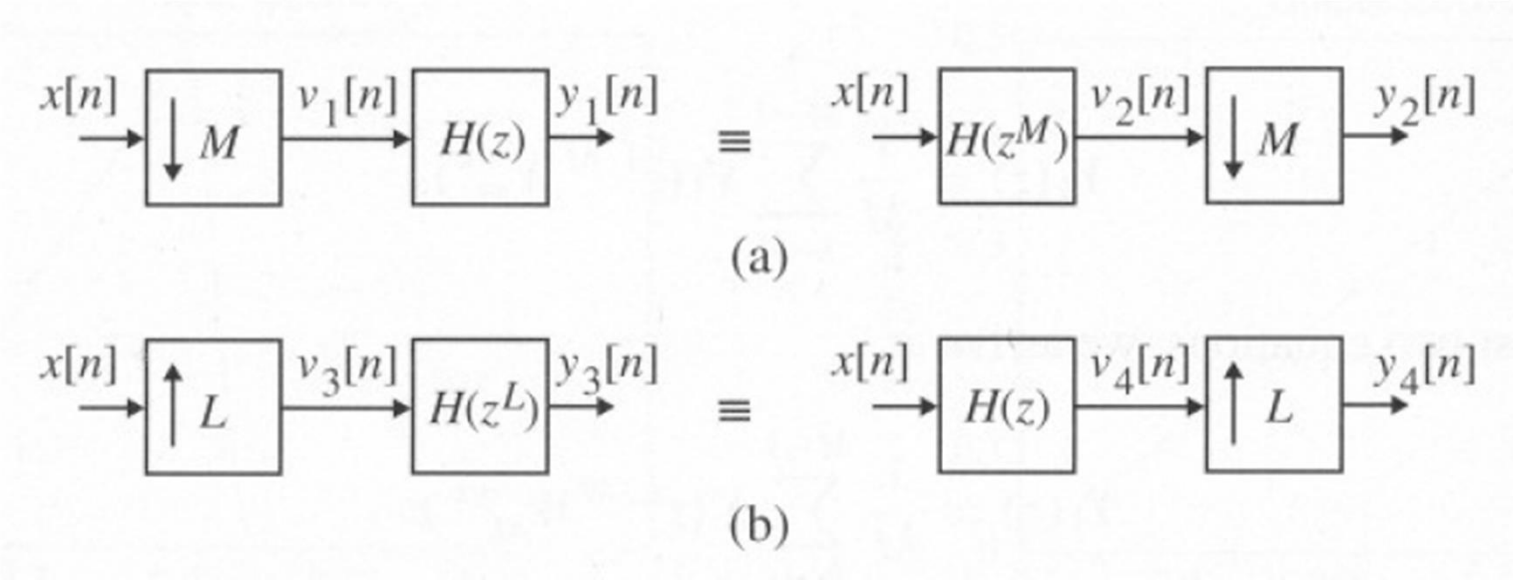
$$Y_1(z) = \frac{1}{M}\sum_{k=0}^{M-1} X(z^{L/M} W_M^{-kL})$$

$$Y_2(z) = \frac{1}{M}\sum_{k=0}^{M-1} X(z^{L/M} W_M^{-k})$$

$$Y_1(z) = Y_2(z) \quad \rightarrow \quad W_M^{-kL} = W_M^{-k}$$

M i *L* su celobrojni i nemaju zajednički činioc

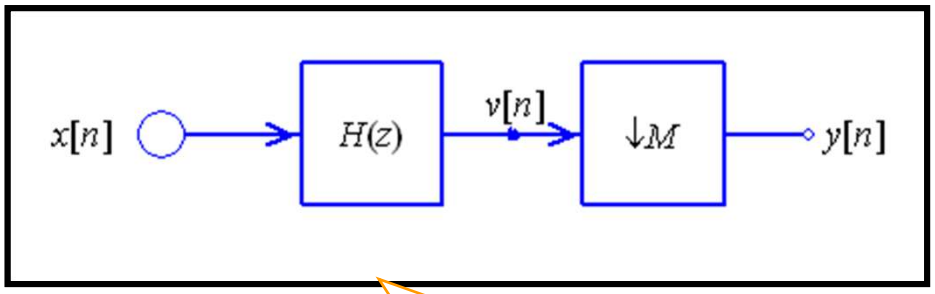
Ekvivalentne realizacije



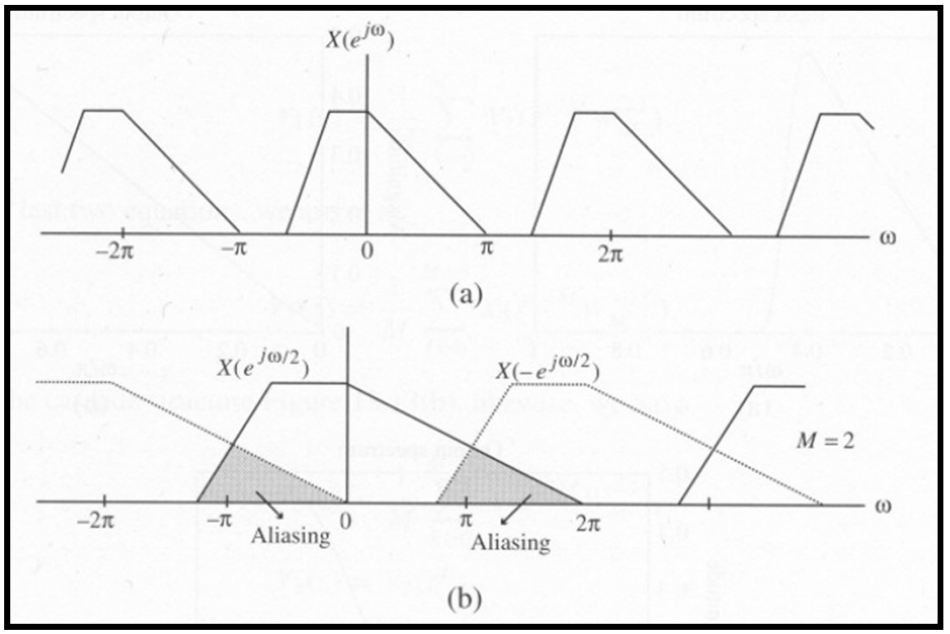
$$Y_1(z) = Y_2(z)$$

$$Y_3(z) = Y_4(z)$$

Decimacija

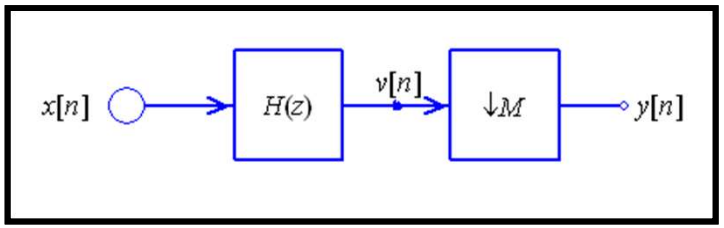


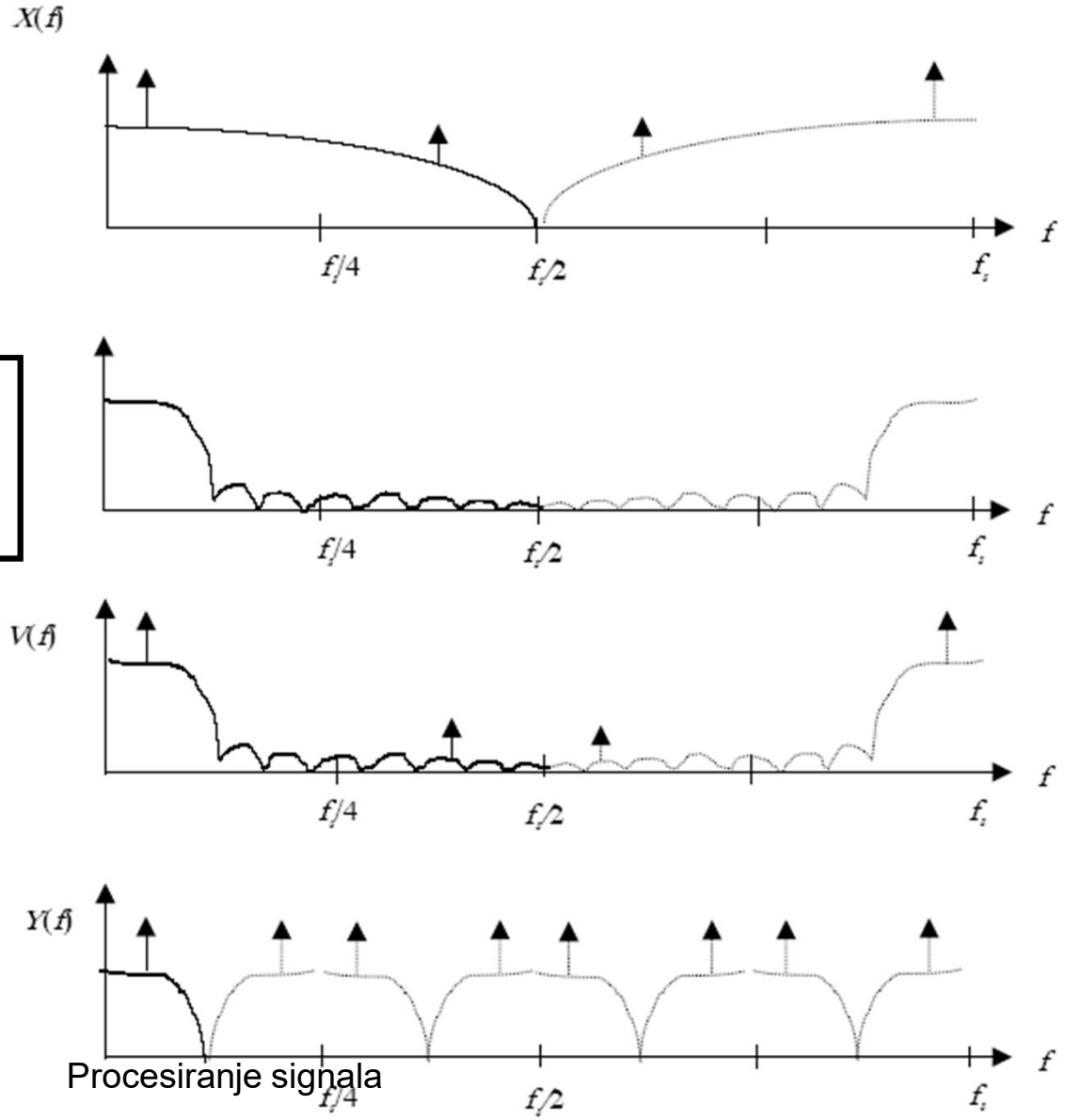
Filtar za decimaciju



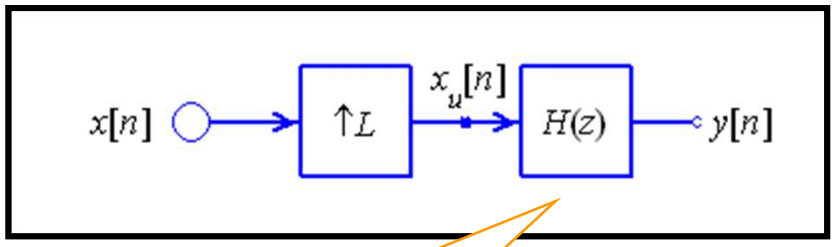
- Najviša učestanost u spektru signala mora da bude manja od polovine učestanosti odabiranja i tada nema aliasing-a
- Pre operacije down-sampling mora da se ograniči spektar signala na 1/(2*M*) da ne bi došlo do aliasing-a

Proces decimacije *M*=4

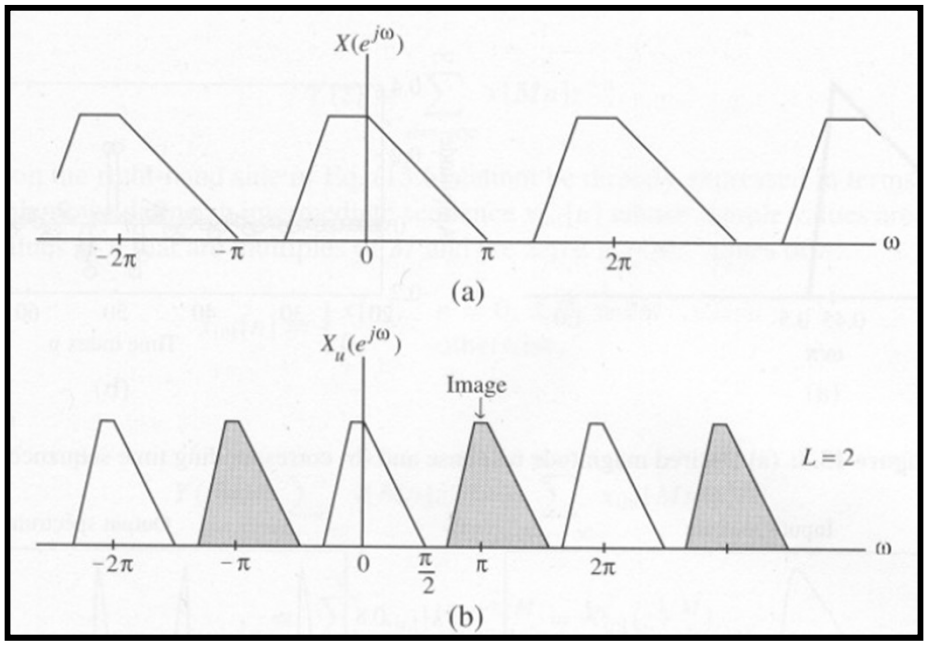




Interpolacija

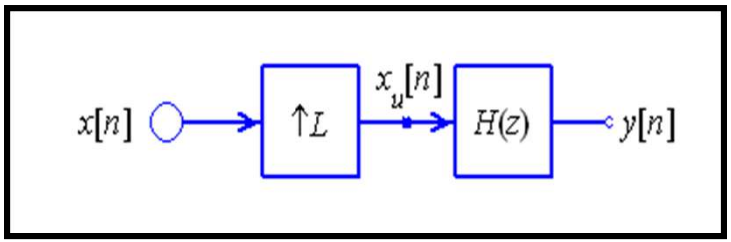


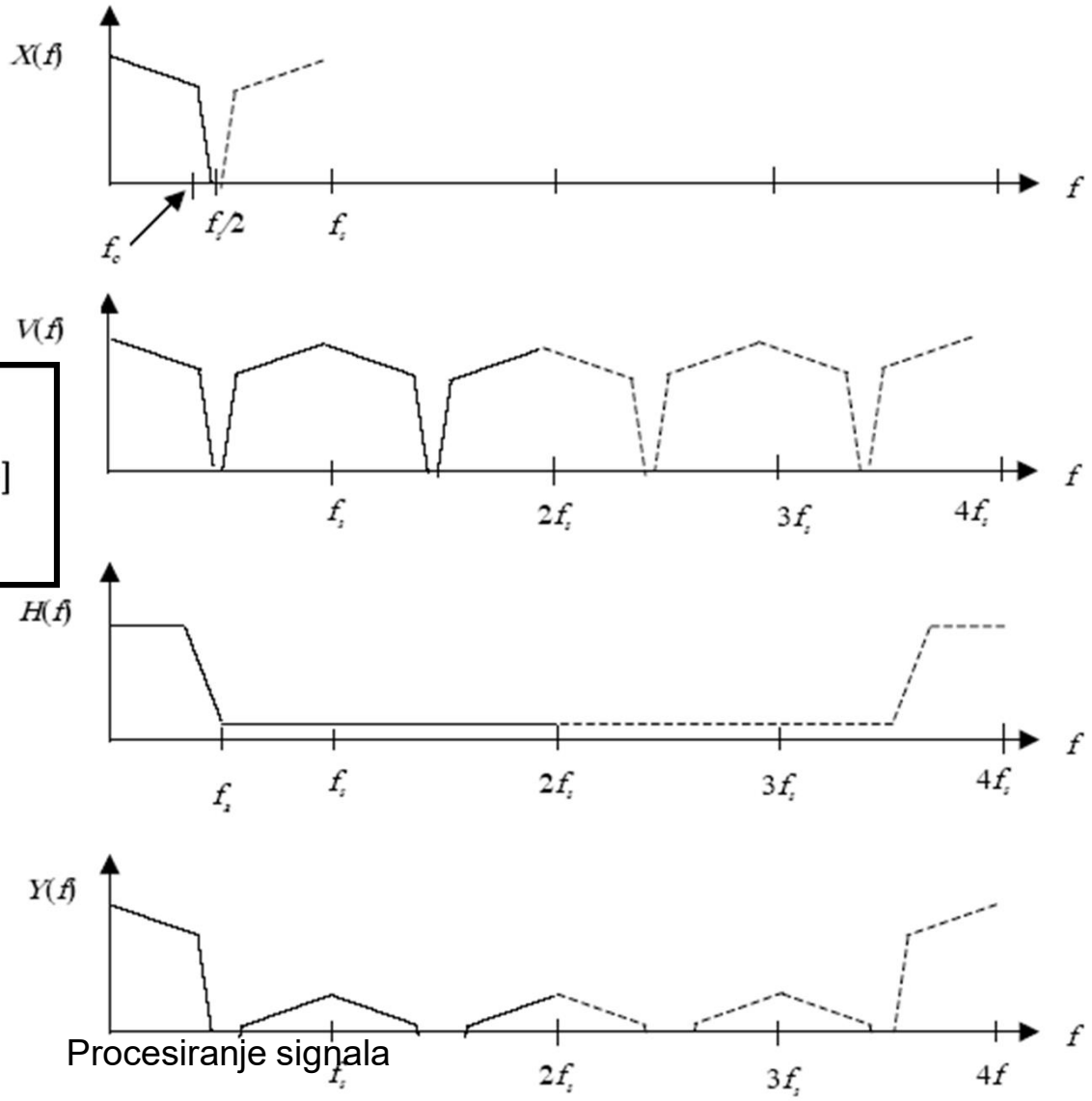
Interpolacioni filtar



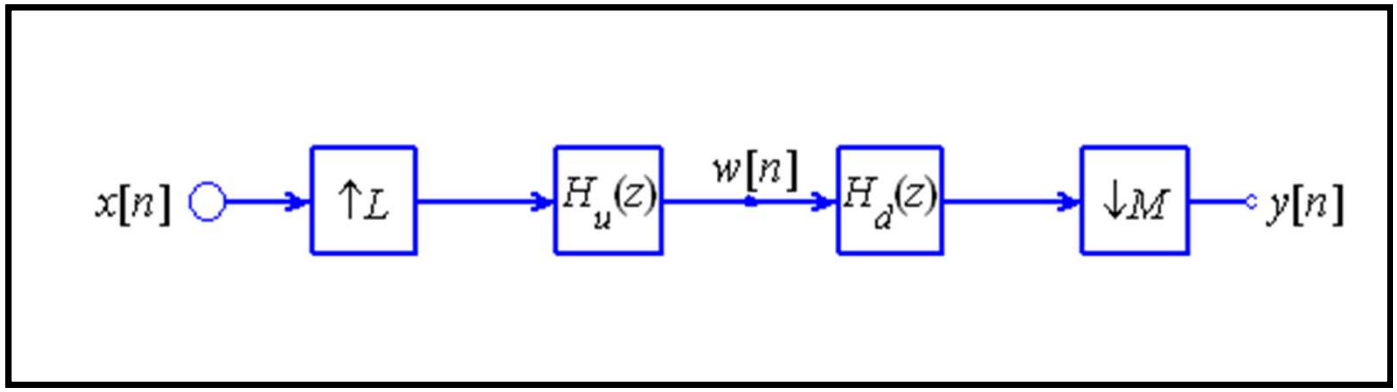
- Ubacivanje odbiraka nultih vrednosti sa up-sampler generiše spektralne komponente na učestanostima iznad 1/(2*L*)
- Posle operacije up-sampling mora da se ograniči spektar signala na 1/(2*L*) da bi se eliminisale nepoželjne komponente

Proces interpolacije L=4



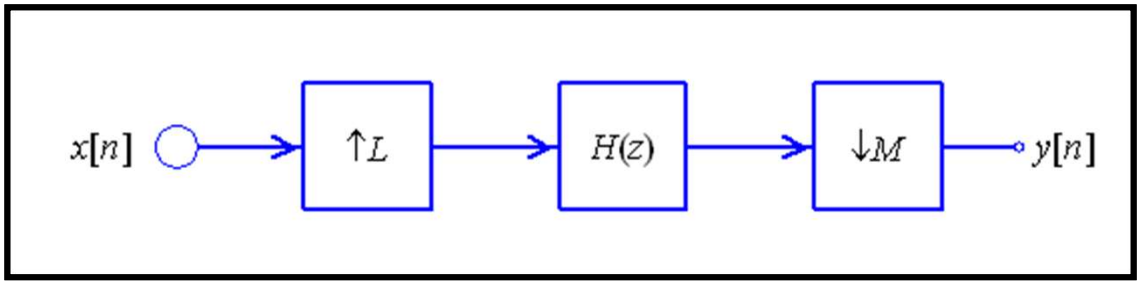


Promena učestanosti odabiranja sa faktorom *L*/*M*



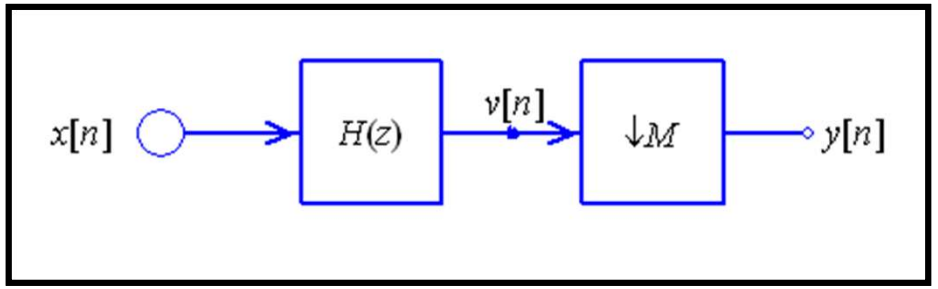
$$H(z) = H_u(z),\ L > M$$

$$H(z) = H_d(z),\ L < M$$



Ulazno-izlazne relacije decimatora

$$v[n] = \sum_{l=-\infty}^{\infty} h[n-l]\, x[l]$$



$$y[n] = v[Mn]$$

$$V(z) = H(z)\, X(z)$$

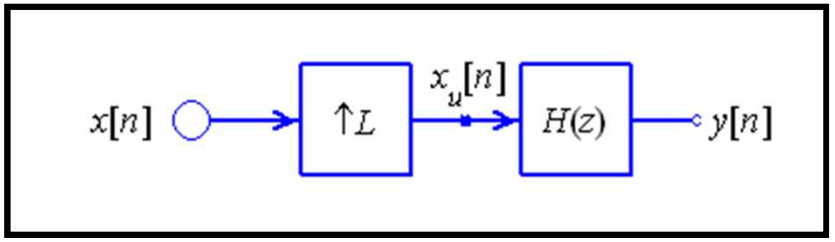
$$y[n] = \sum_{l=-\infty}^{\infty} h[Mn-l]\, x[l]$$

$$Y(z) = \frac{1}{M} \sum_{k=0}^{M-1} V(z^{1/M} W_M^{-k})$$

$$Y(z) = \frac{1}{M} \sum_{k=0}^{M-1} H(z^{1/M} W_M^{-k})\, X(z^{1/M} W_M^{-k})$$

Ulazno-izlazne relacije interpolatora

$$x_u[Lm] = x[m],\ m = 0, \pm 1, \pm 2$$

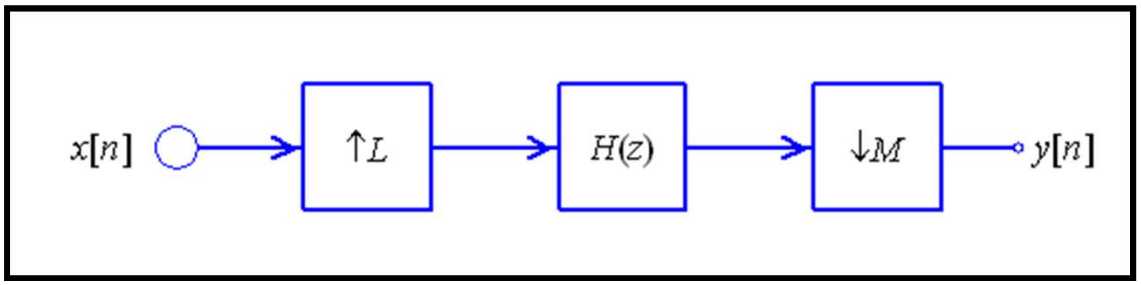


$$y[n] = \sum_{l=-\infty}^{\infty} h[n-l]\, x_u[l]$$

$$Y(z) = H(z)\, X(z^L)$$

$$y[n] = \sum_{m=-\infty}^{\infty} h[n-Lm]\, x[Lm]$$

Ulazno-izlazne relacije za promenu učestanosti odabiranja *L*/*M*

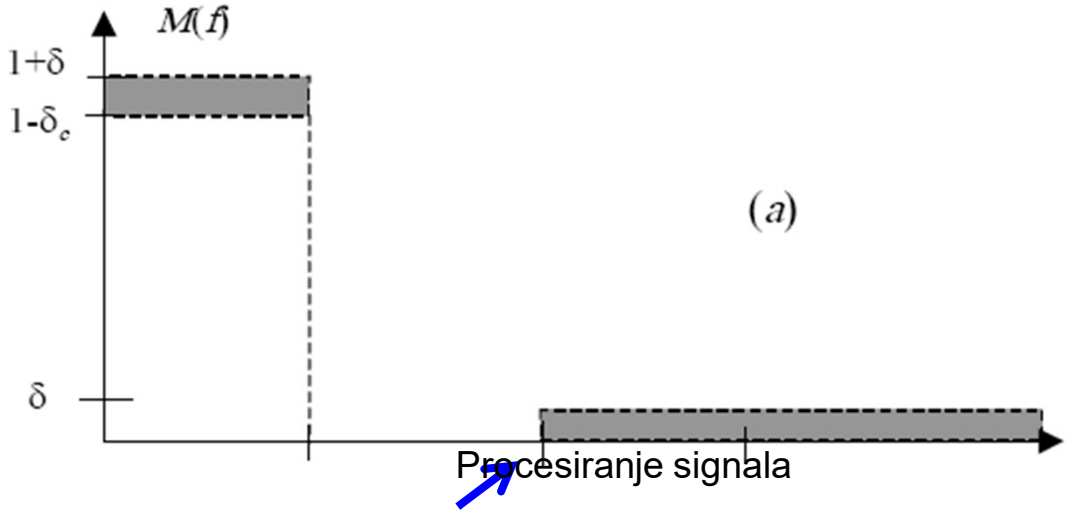


$$y[n] = \sum_{m=-\infty}^{\infty} h[Mn - Lm]\, x[m]$$

$$Y(z) = \frac{1}{M} \sum_{k=0}^{M-1} H(z^{1/M} W_M^{-k})\, X(z^{L/M} W_M^{-kL})$$

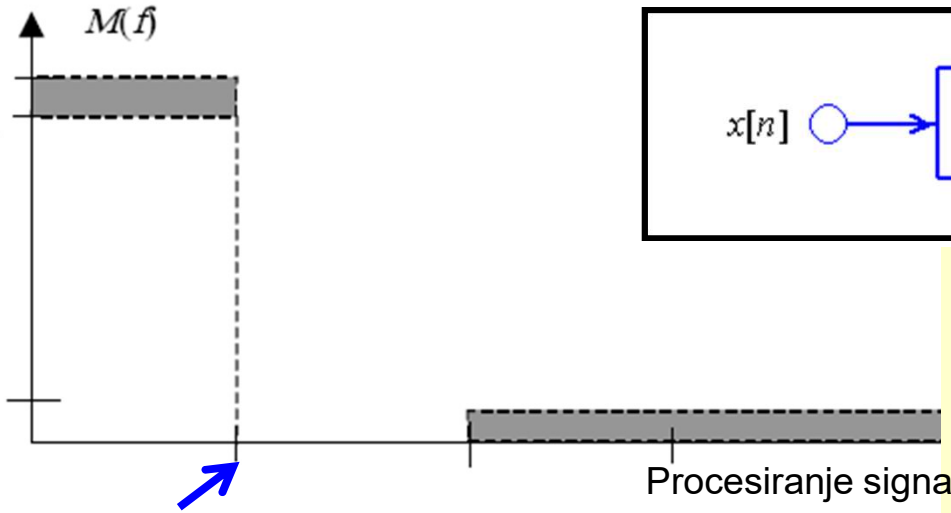
Interpolacioni filtar (1)

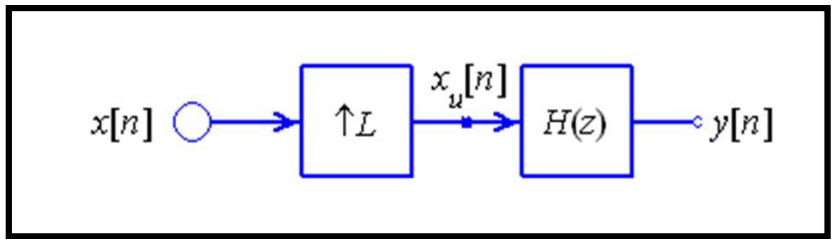
- Granica nepropusnog opsega mora da bude $1/(2L)$ u odnosu na učestanost odabiranja, $F_{\text{stop}} = F_{\text{T}}/(2L)$
- Granica propusnog opsega je manja od F_{stop} i što je moguće bliža do F_{stop}



Interpolacioni filtar (2)

- Neka je F_c najviša učestanost ulaznog signala koji treba da se zadrži posle interpolacije
- Granica propusnog opsega mora da bude $F_{\text{pass}} = F_c/L < F_T/(2L)$, F_T izlaznog signala

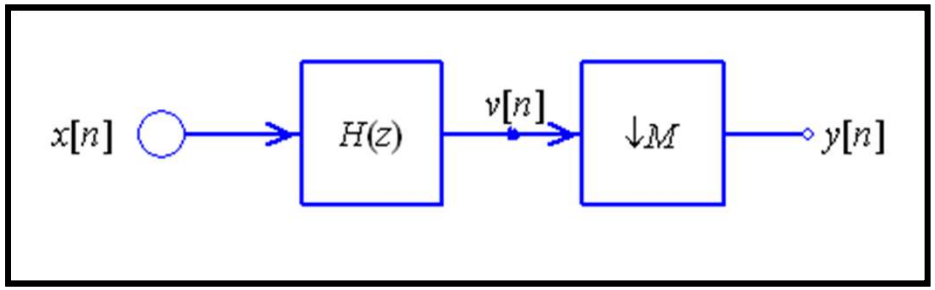


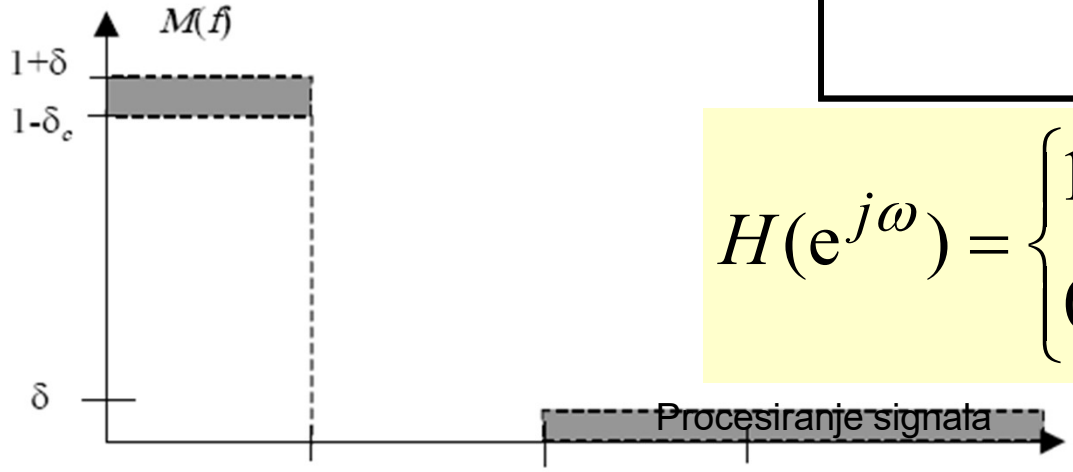


$$H(e^{j\omega}) = \begin{cases} L, |f| \le F_c / L \\ 0, |f| \ge F_T / 2L \end{cases}$$

Decimacioni filtar

- Neka je F_c najviša učestanost izlaznog signala koji treba da se zadrži posle decimacije
- Granica propusnog opsega mora da bude $F_{\text{pass}} = F_c/L < F_T/(2L)$, F_T ulaznog signala

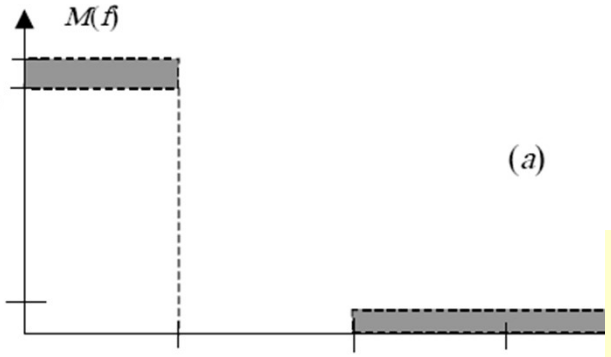


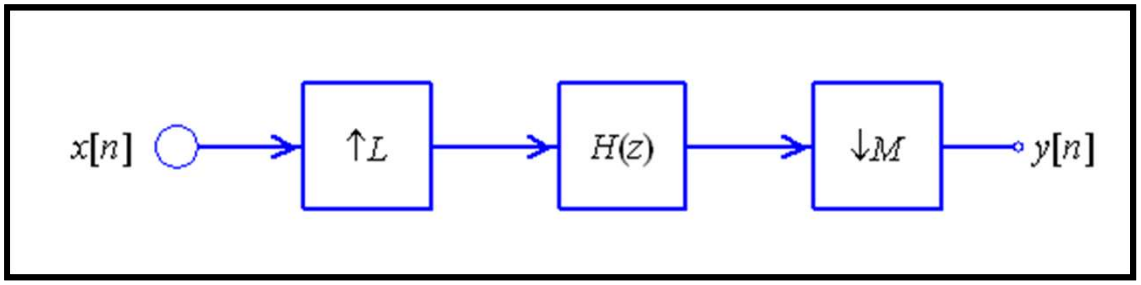


$$H(e^{j\omega}) = \begin{cases} 1, & |f| \le F_c / M \\ 0, & |f| \ge F_T / 2M \end{cases}$$

Decimacioni-interpolacioni filtar

- Neka je F_c najviša učestanost izlaznog signala koji treba da se zadrži posle promene učestanosti odabiranja sa faktorom L/M, F_T je učestanost odabiranja filtra
- Granica propusnog opsega mora da zadovolji oba zahteva i za decimacioni i interpolacioni filtar





$$H(\mathrm{e}^{j\omega}) = \begin{cases} L, |f| \le \min(F_c / M, F_c / L) \\ 0, |f| \ge \min(F_T / 2M, F_T / 2L) \end{cases}$$

Efikasnost FIR i IIR filtara

- Decimacioni FIR filtar izračunava izlazne vrednosti samo one koji se propuštaju kroz down-sampler – ušteda je veća ukoliko je veći faktor decimacije
- Decimacioni IIR filtar izračunava izlazne vrednosti samo one koji se propuštaju kroz down-sampler ali sve vrednosti odbiraka koji se koriste u povratnim granama pa je i ušteda je manja u odnosu na FIR filtre

Primer složenosti decimatora

- R_M je broj množenja u sekundi
- F_T je učestanost odabiranja
- FIR, $H(z)$ dužine N: $R_M = N \times F_T$
- FIR $H(z)$ dužine N, down-sampler M: $\boldsymbol{R_M = F_T \times N/M}$
- IIR, $H(z)$ reda K: $R_M = (2K+1) \times F_T$
- IIR, $H(z)$ reda K, down-sampler M: $\boldsymbol{R_M = K \times F_T + (K+1) \times F_T/M = F_T\,((K+1)/M+K)}$

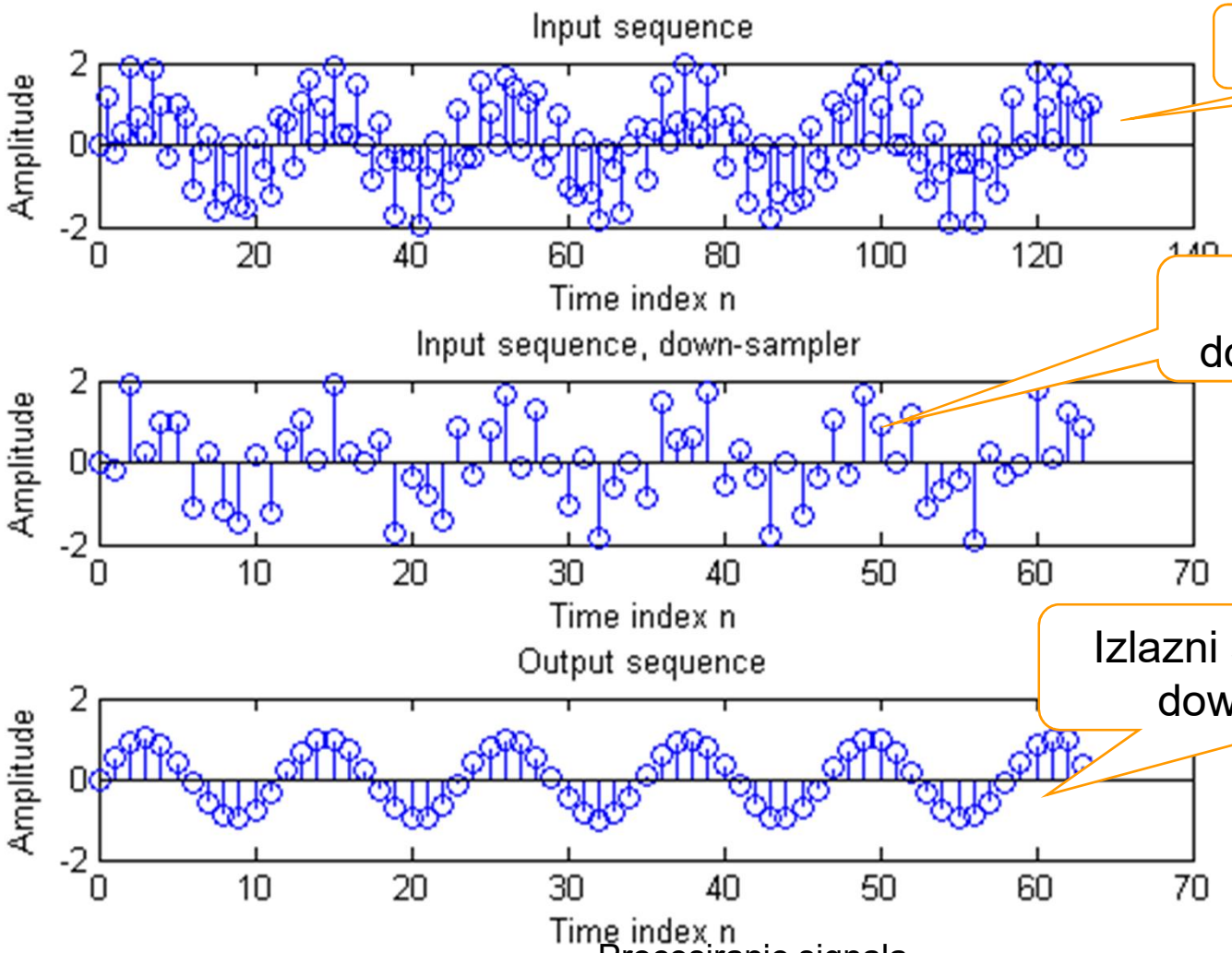
Program 13_5

```
N = 128;      M = 2;
f1 = 0.043;   f2 = 0.31;
n = 0:N-1;

x = sin(2*pi*f1*n) + sin(2*pi*f2*n);
y = decimate(x,M,'fir');
subplot(3,1,1); stem(n,x(1:N));
title('Input sequence');
xlabel('Time index n');ylabel('Amplitude');

subplot(3,1,2); stem(n(1:M:N)/M,x(1:M:N));
title('Input sequence, down-sampler ');
xlabel('Time index n');ylabel('Amplitude');

subplot(3,1,3)
m=0:N/M-1;  stem(m,y(1:N/M));
title('Output sequence');
xlabel('Time index n'); ylabel('Amplitude');
```

Ulazni signal

decimate

Vremenski domen: Program 13_5

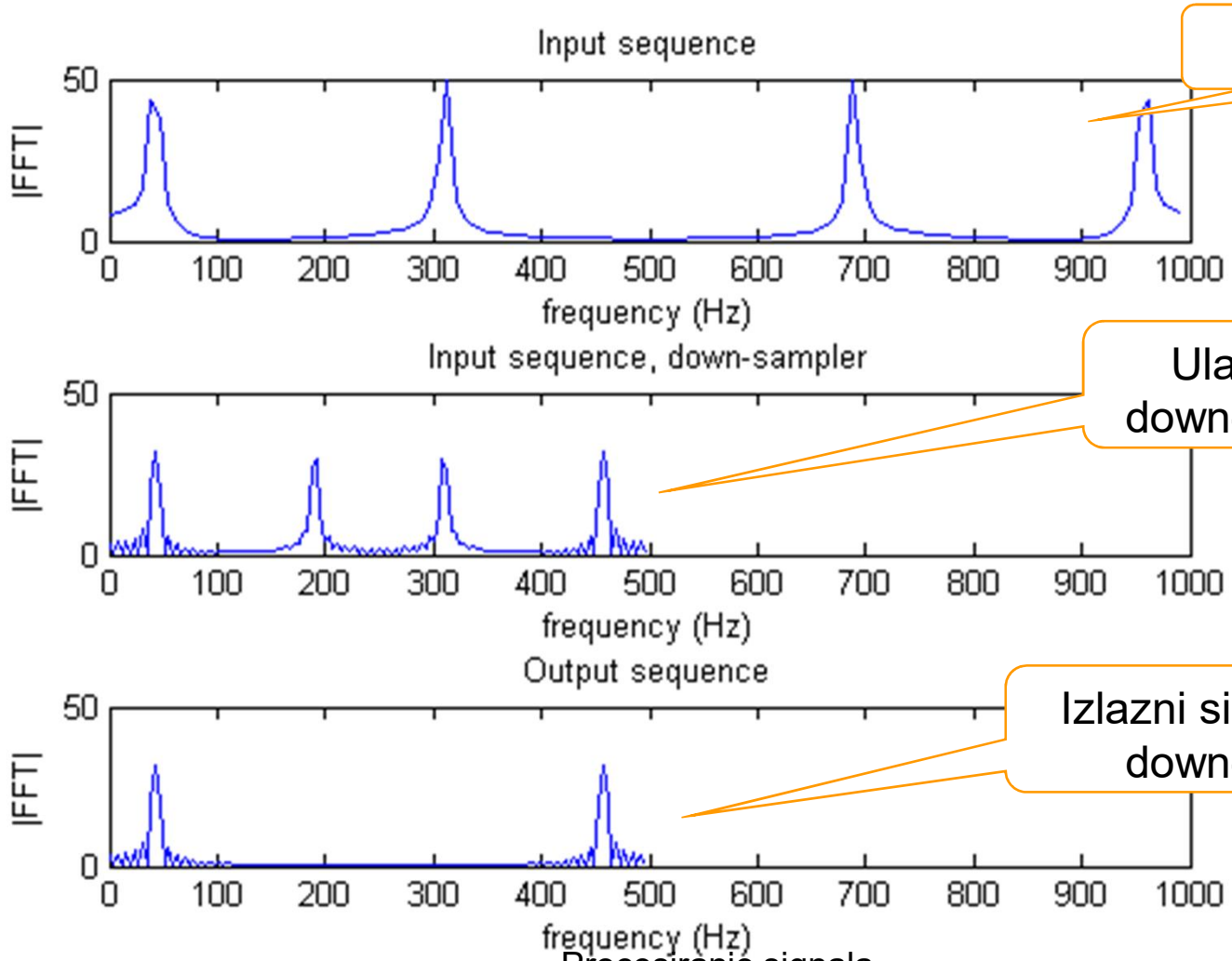


Ulazni signal

Ulazni signal i down-sample M=2

Izlazni signala, FIR filtar i down-sample M=2

Spektar signala: Program 13_5



Ulazni signal

Ulazni signal i down-sample *M*=2

Izlazni signala, FIR filtar i down-sample *M*=2

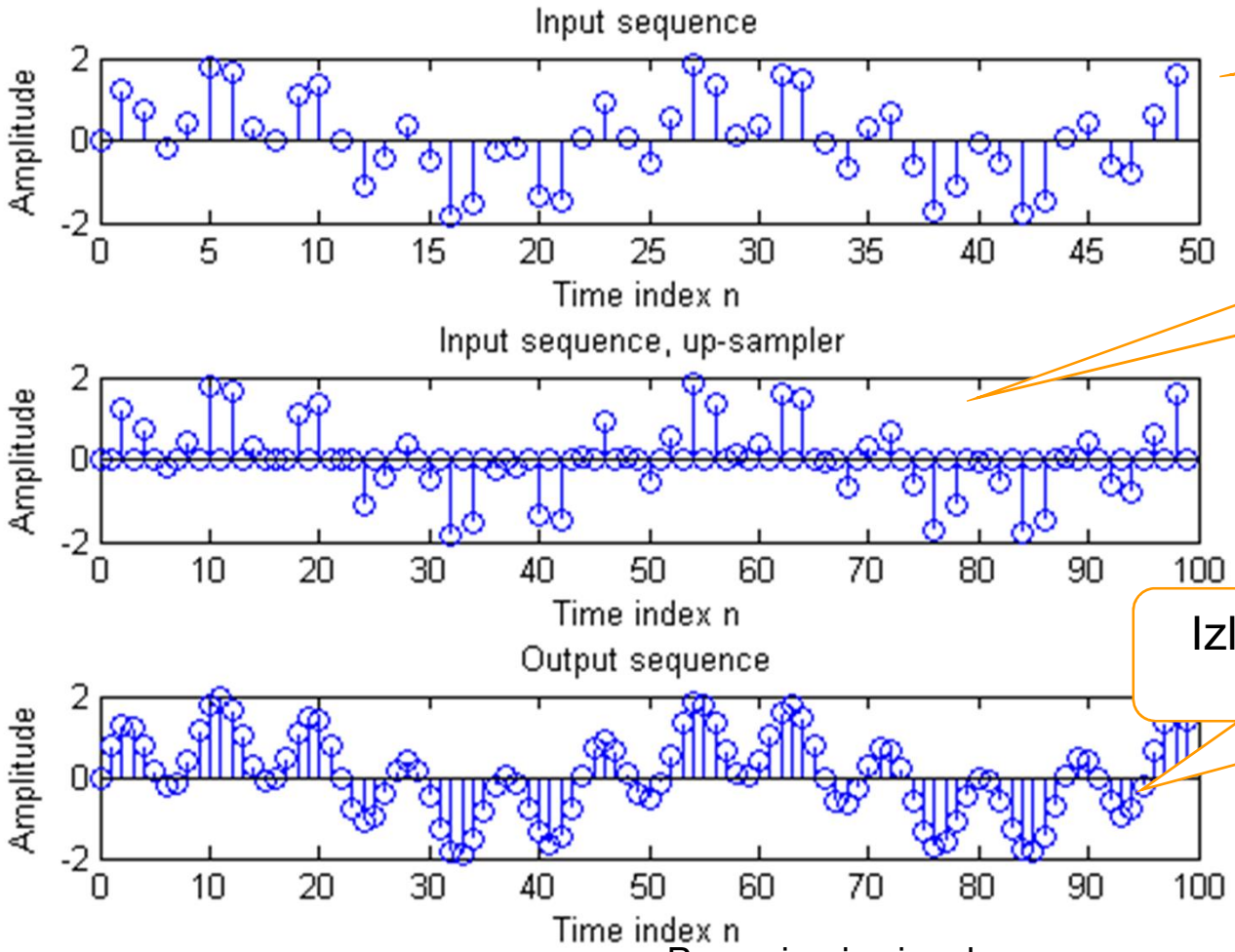
Program 13_6

```
N = 128;    L = 2;
f1 = 0.043; f2 = 0.23;
n = 0:N-1;

x = sin(2*pi*f1*n) + sin(2*pi*f2*n);
y = interp(x,L);
subplot(3,1,1), stem(n(1:50),x(1:50));
title('Input sequence');
xlabel('Time index n');ylabel('Amplitude');
subplot(3,1,3), m=0:N*L-1;
stem(m(1:L*50),y(1:L*50));
title('Output sequence');
xlabel('Time index n'); ylabel('Amplitude');
subplot(3,1,2)
x2 = zeros(size(y)); x2(1:L:L*N) = x;
stem(m(1:L*50),x2(1:L*50));
title('Input sequence, up-sampler ');
xlabel('Time index n');ylabel('Amplitude');
```

Ulazni signal

interp

Vremenski domen: Program 13_6

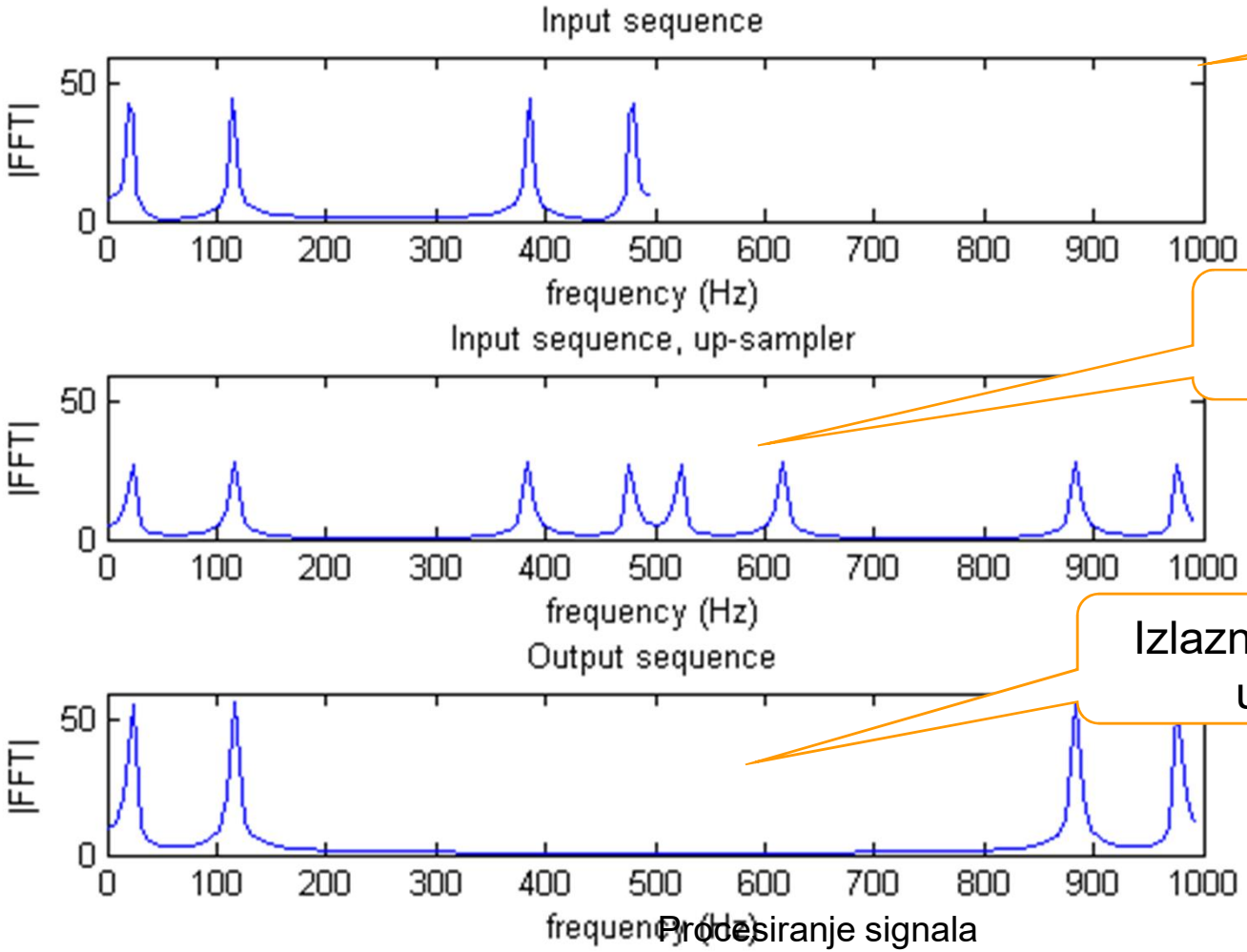


Ulazni signal

Ulazni signal i up-sample L=2

Izlazni signala, FIR filtar i up-sample L=2

Spektar signala: Program 13_6



Ulazni signal

Ulazni signal i up-sample L=2

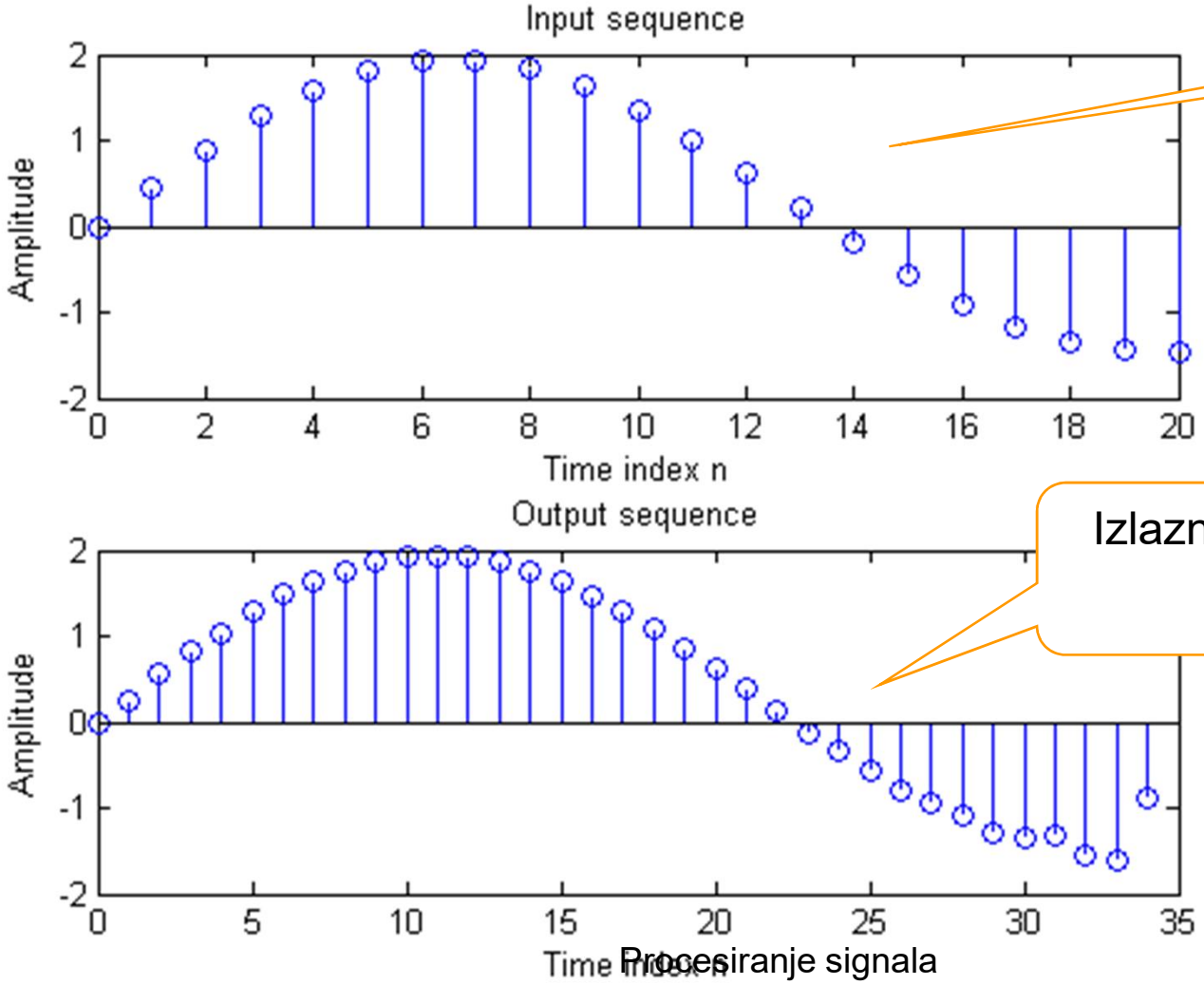
Izlazni signala, FIR filtar i up-sample L=2

Program 13_7

```
N = 21;
L = 5;
M = 3;
f1 = 0.043; f2 = 0.031;

n = 0:N-1;
x = sin(2*pi*f1*n) + sin(2*pi*f2*n);
y = resample(x,L,M);
subplot(2,1,1)
stem(n,x(1:N));
title('Input sequence');
xlabel('Time index n'); ylabel('Amplitude');
subplot(2,1,2)
m=0:N*L/M-1;
stem(m,y(uint8(1:N*L/M)));
title('Output sequence');
xlabel('Time index n'); ylabel('Amplitude');
```

resample

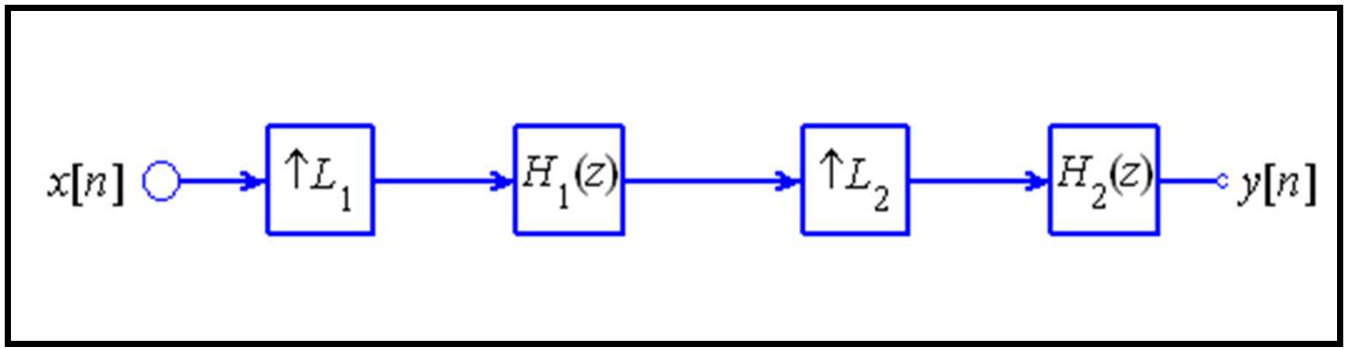
Vremenski domen: Program 13_7

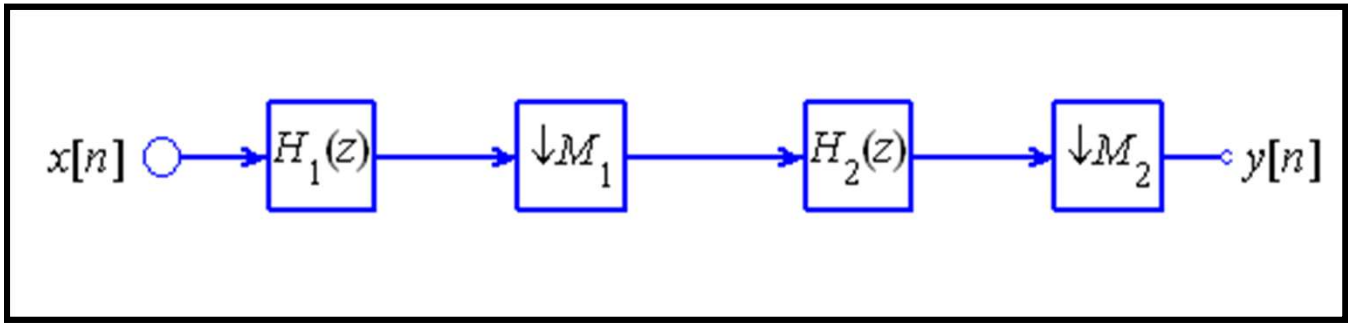


Ulazni signal

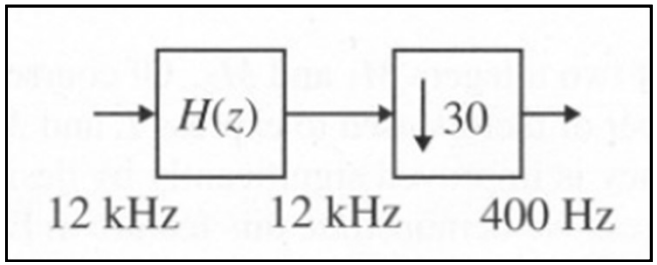
Izlazni signala, FIR filtar i
L=5, *M*=3

Višestepena realizacija



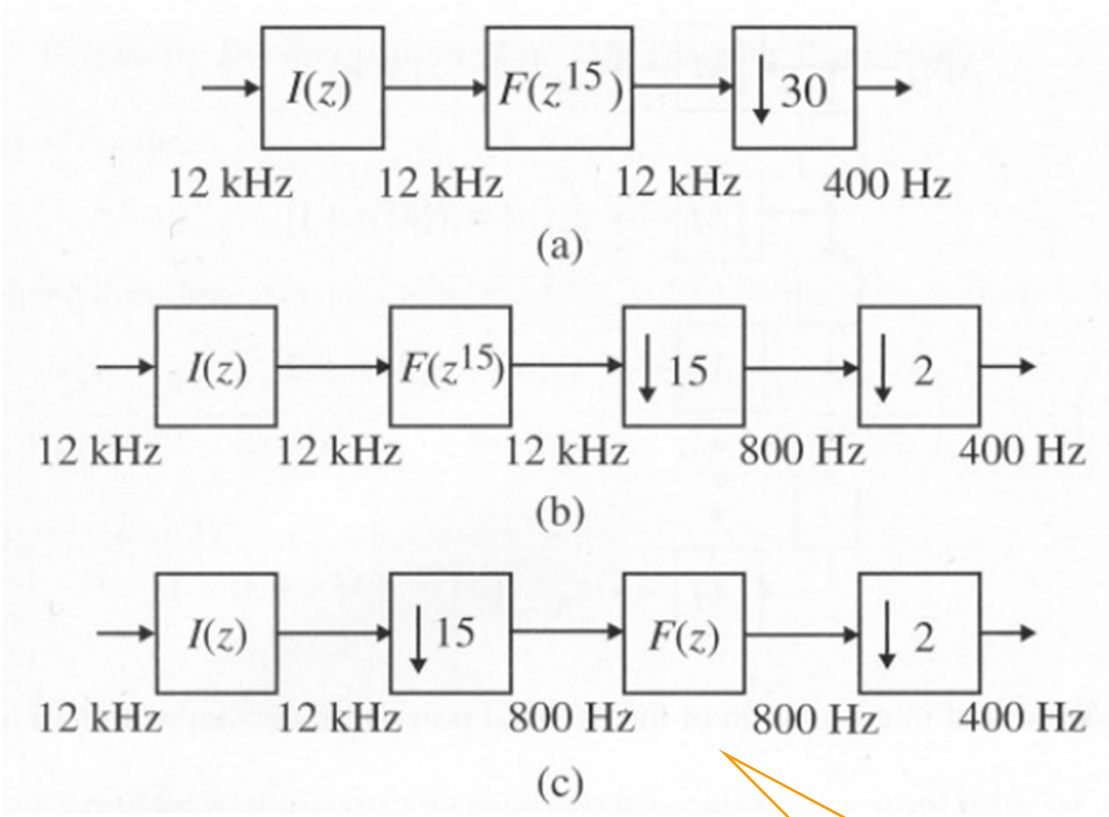


Primer višestepenog decimatora



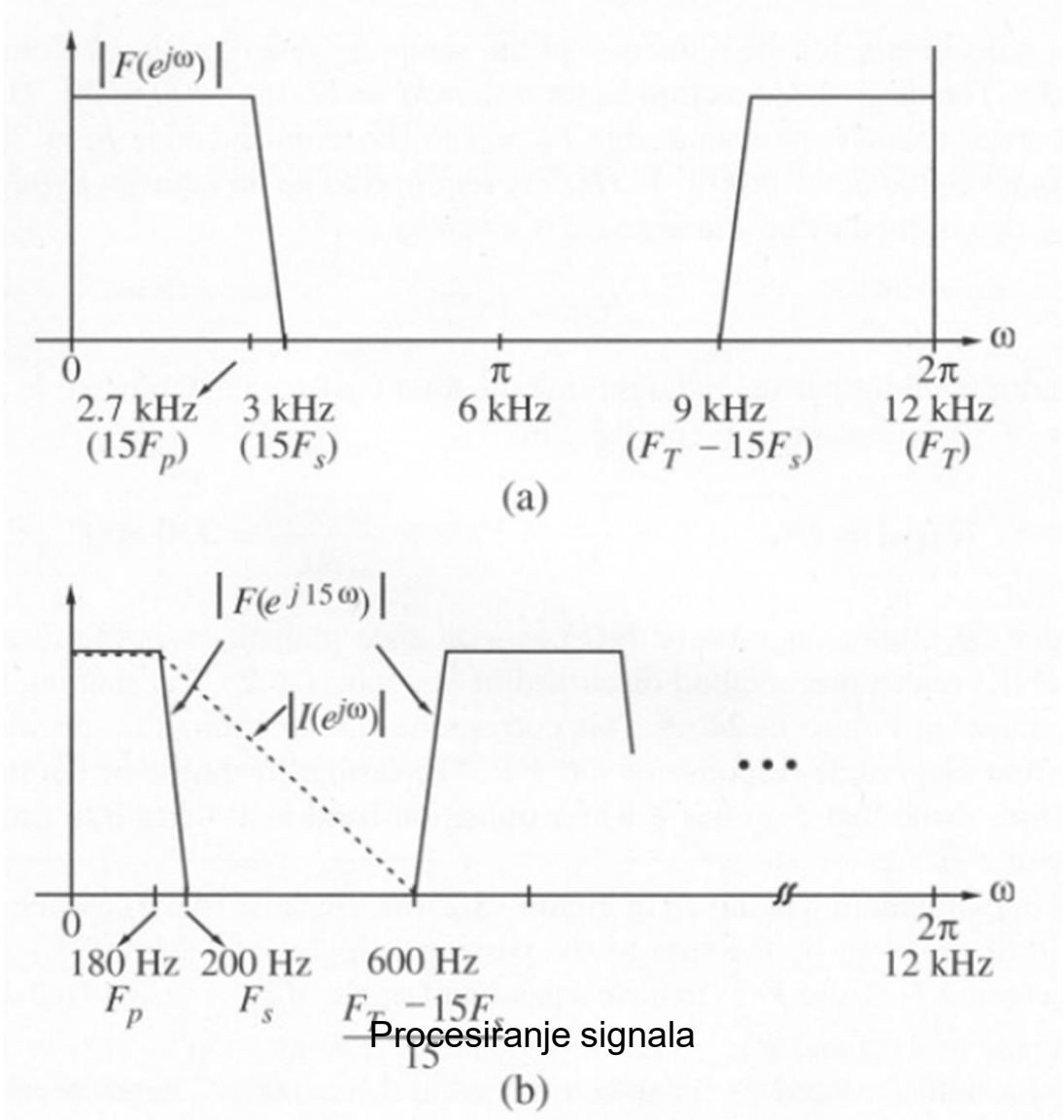
- Smanjiti učestanost odabiranja sa F_{T1}=12 kHz na F_{T2}=400 Hz, (M=30)
- Specifikacije: $F_{p,max}$=180 Hz
  δ_p=0.002, δ_s=0.001, F_s=F_{T2}/2=200 Hz
- **`remezord`** za FIR linearne faze, N=1827
- R_M=F_{T1}×N/M=12000×1828/30=730800

Višestepeni decimator (2)

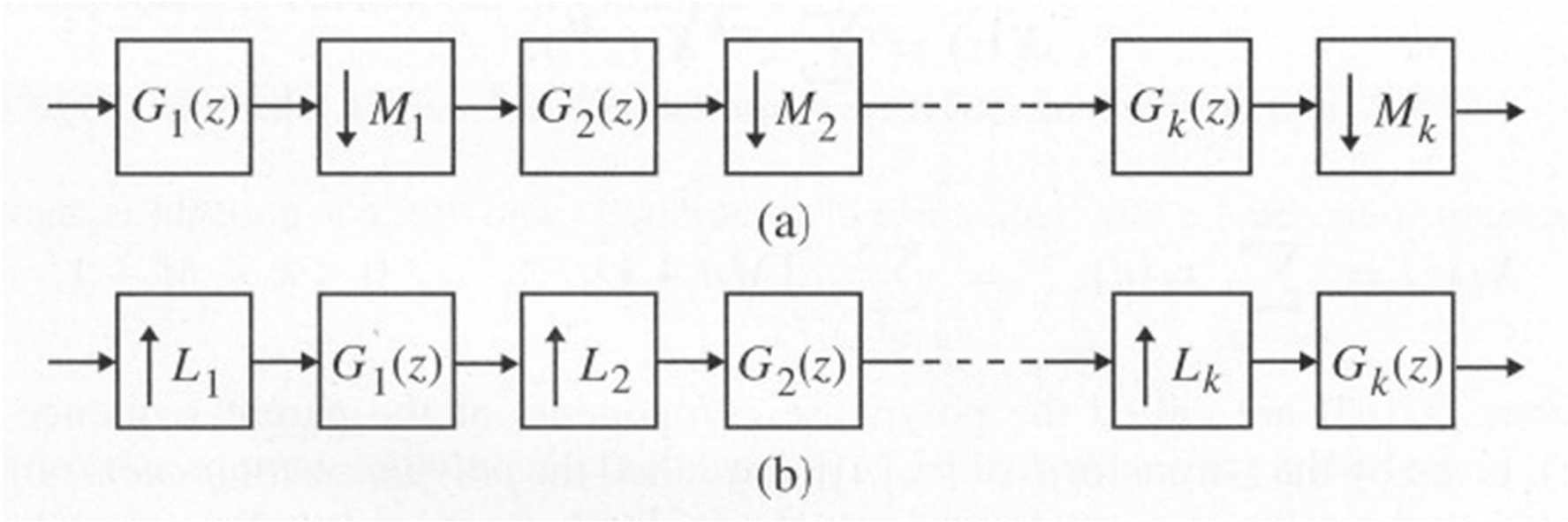


R_M=127600

Višestepeni decimator (3)



Višestepena decimacija - interpolacija

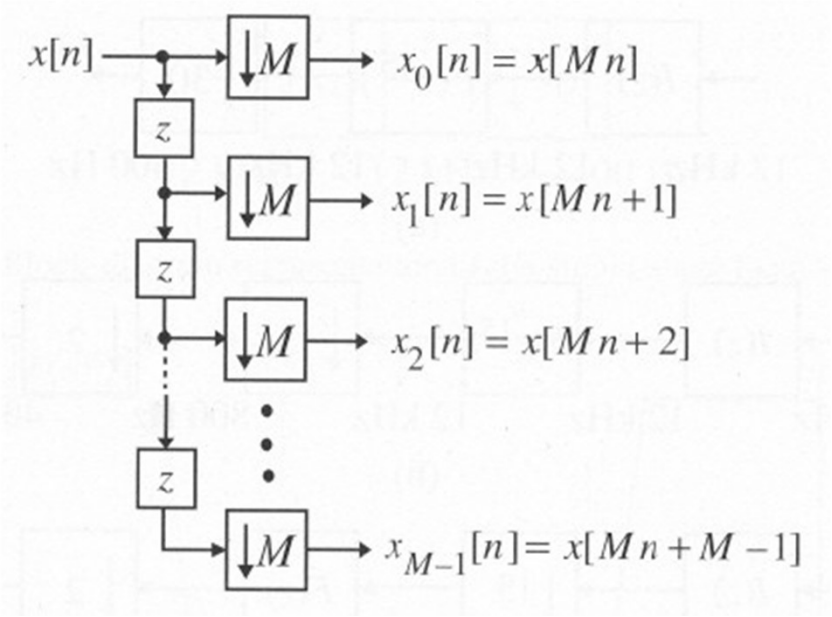


Polifazna dekompozicija

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]\, z^{-n}$$

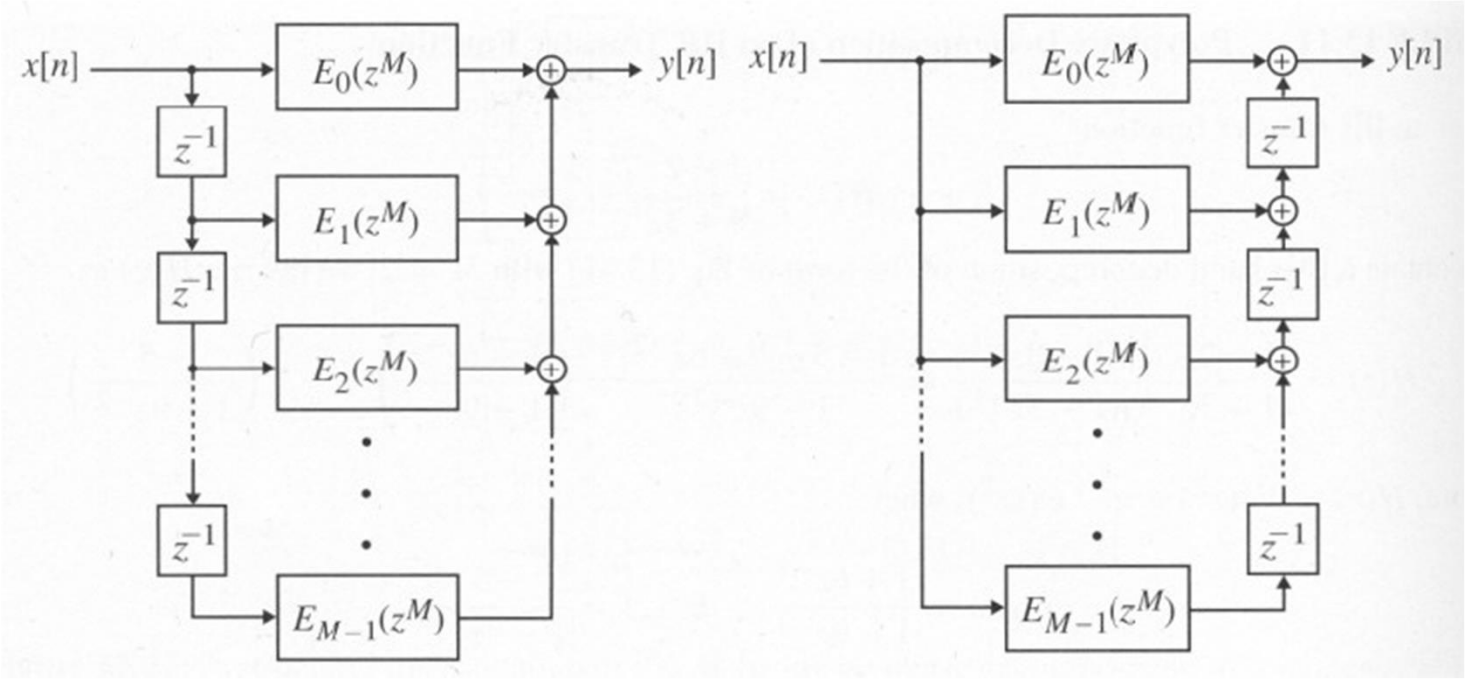
$$X(z) = \sum_{k=0}^{M-1} z^{-k} X_k(z^M)$$

$$X_k(z) = \sum_{n=-\infty}^{\infty} x_k[n]\, z^{-n}$$

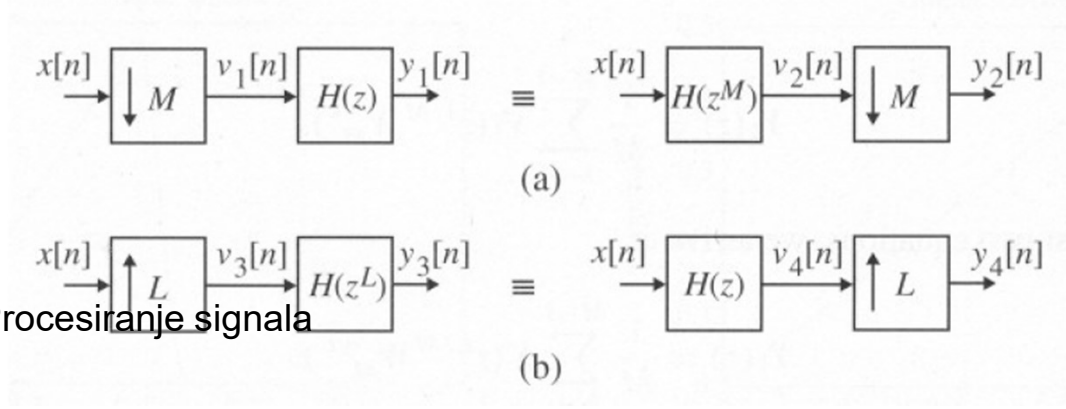


$$x_k[n] = x[Mn+k], \quad 0 \le k \le M-1$$

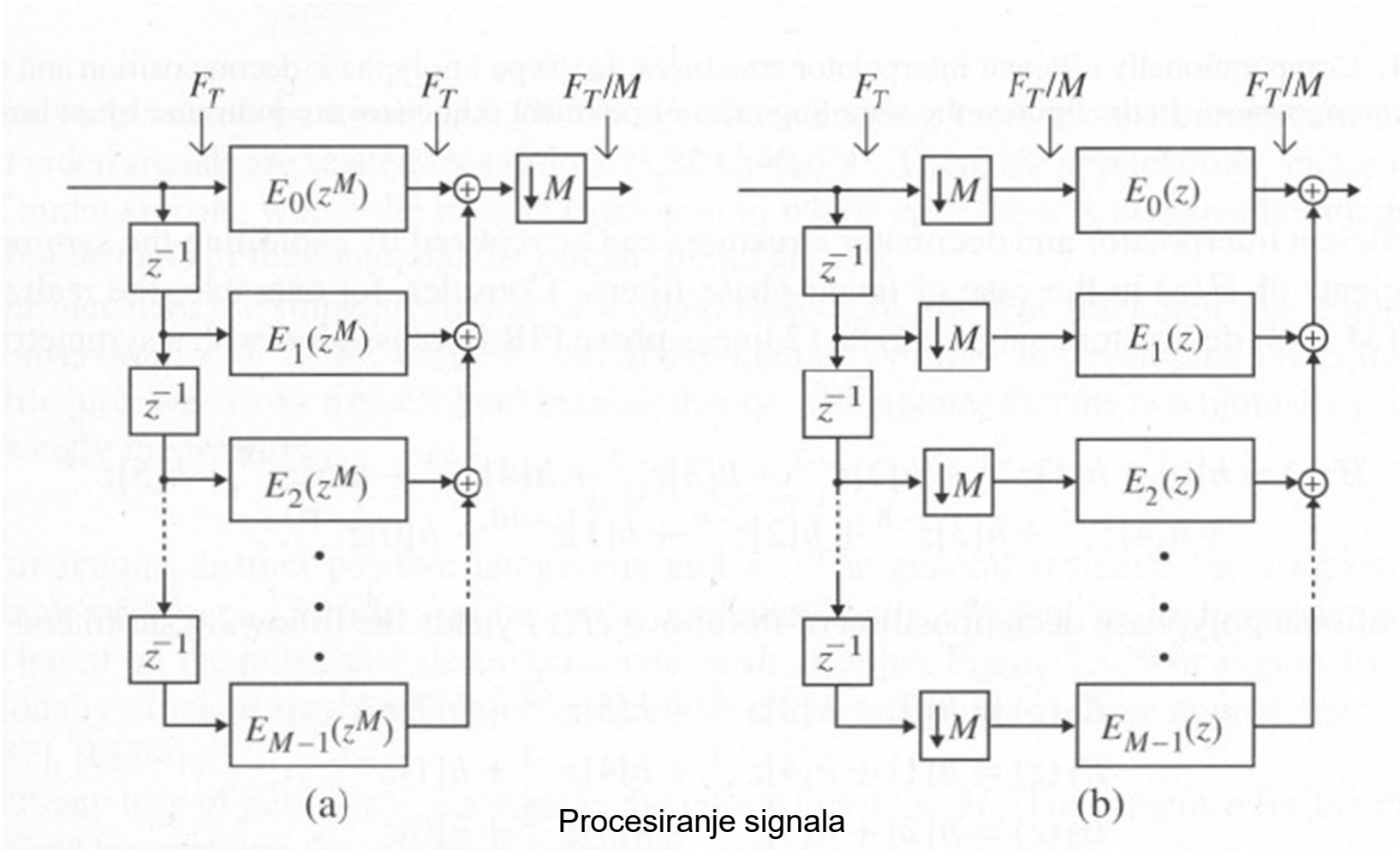
Direktna i transponovana FIR realizacija



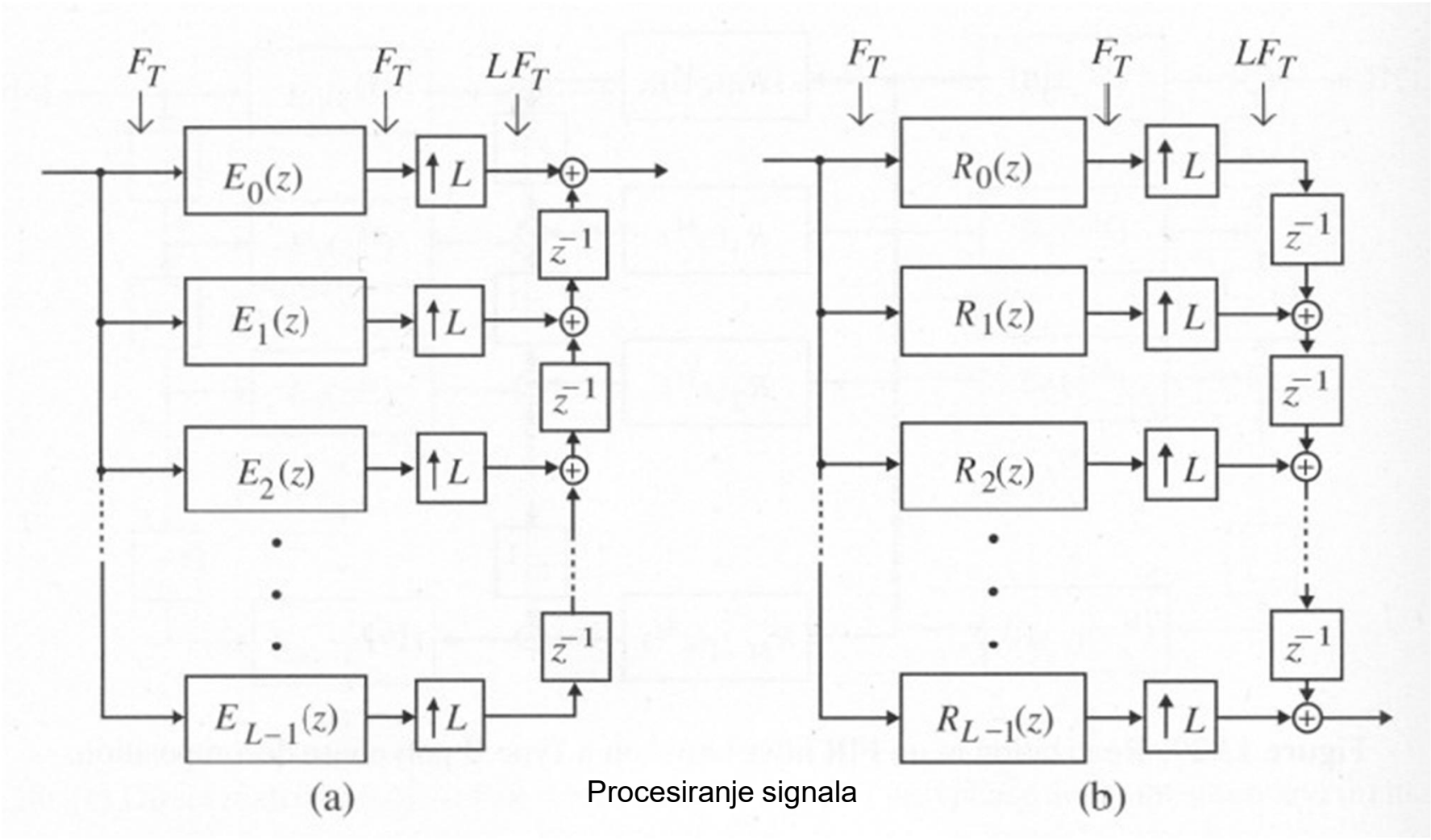
$$H(z) = \sum_{k=0}^{M-1} z^{-k} E_k(z^M)$$



Efikasne decimacione strukture



Efikasne interpolacione strukture

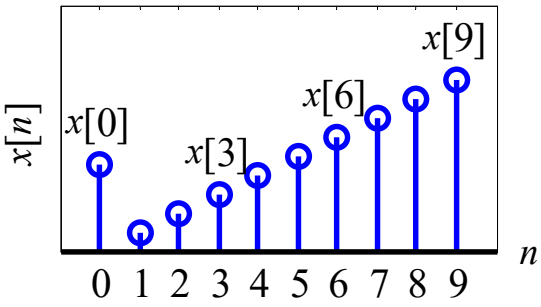


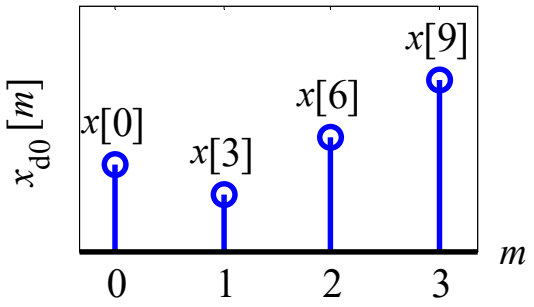
Efikasne FIR strukture

$$H(z) = h[0] + h[1]z^{-1} + h[2]z^{-2} + h[3]z^{-3} + h[4]z^{-4} + h[5]z^{-5} + h[5]z^{-6} + h[4]z^{-7} + h[3]z^{-8} + h[2]z^{-9} + h[1]z^{-10} + h[0]z^{-11}.$$

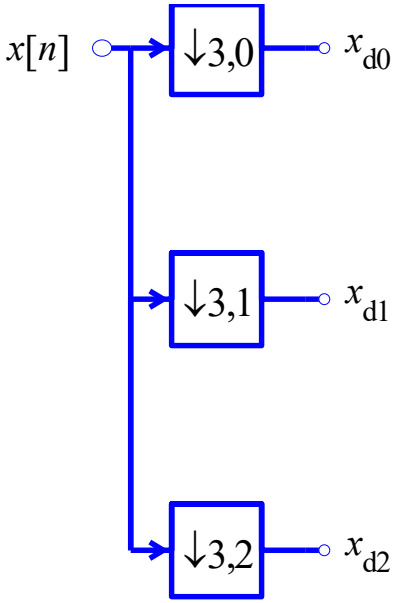
$$E_0(z) = h[0] + h[3]z^{-1} + h[5]z^{-2} + h[2]z^{-3},$$
$$E_1(z) = h[1] + h[4]z^{-1} + h[4]z^{-2} + h[1]z^{-3},$$
$$E_2(z) = h[2] + h[5]z^{-1} + h[3]z^{-2} + h[0]z^{-3}.$$

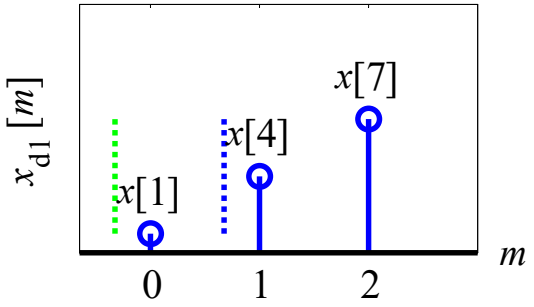
Down-sampling sa ofsetom

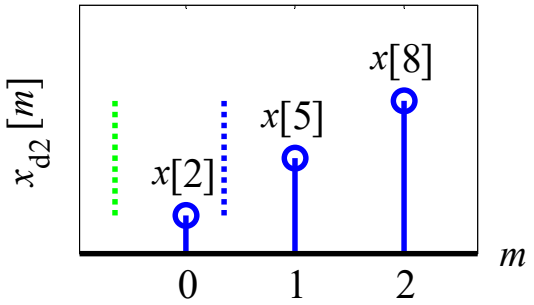




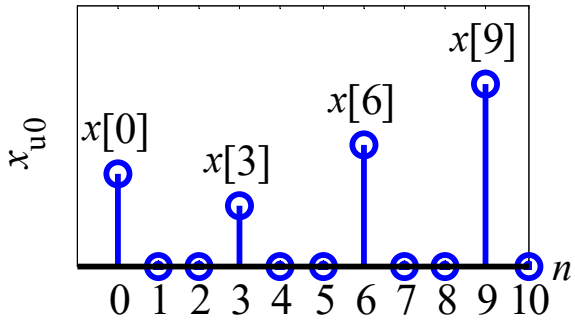

```
downsample(x,M,p)
```

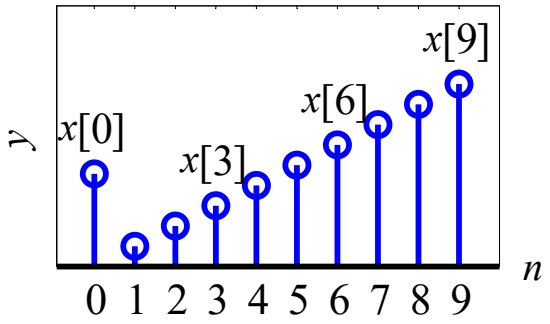


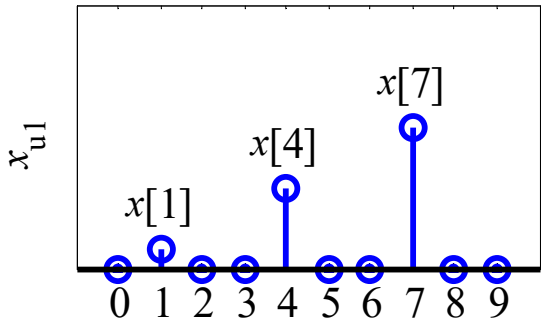


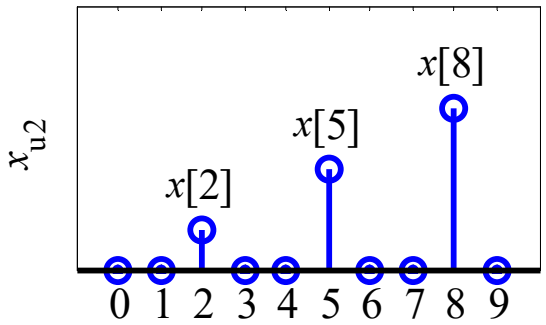


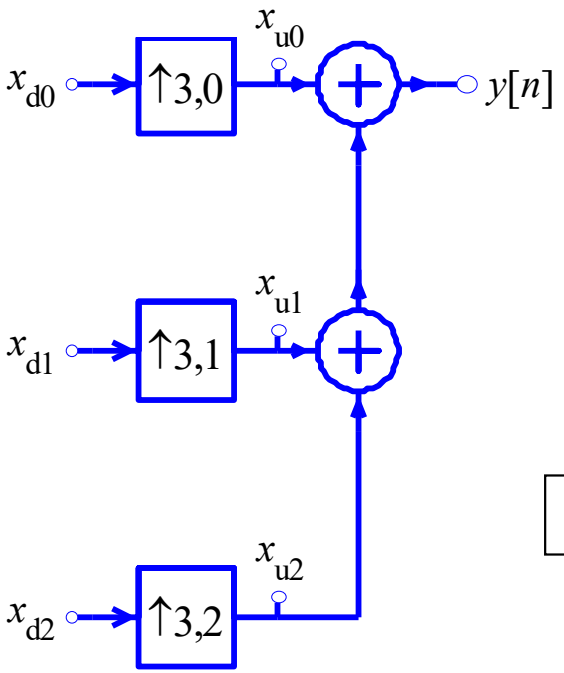
Up-sampling sa ofsetom







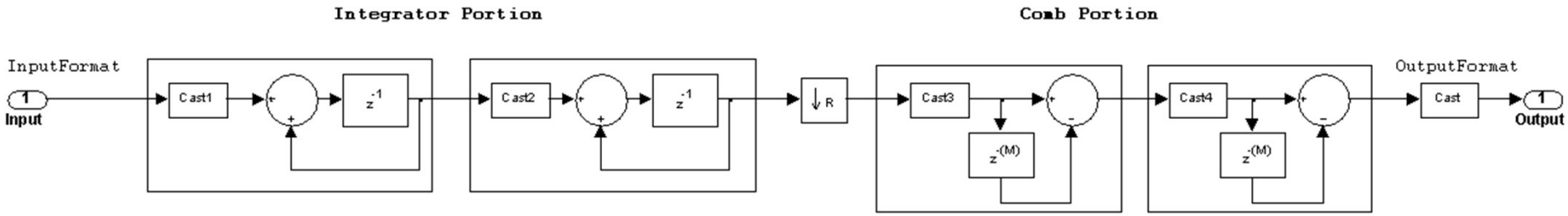




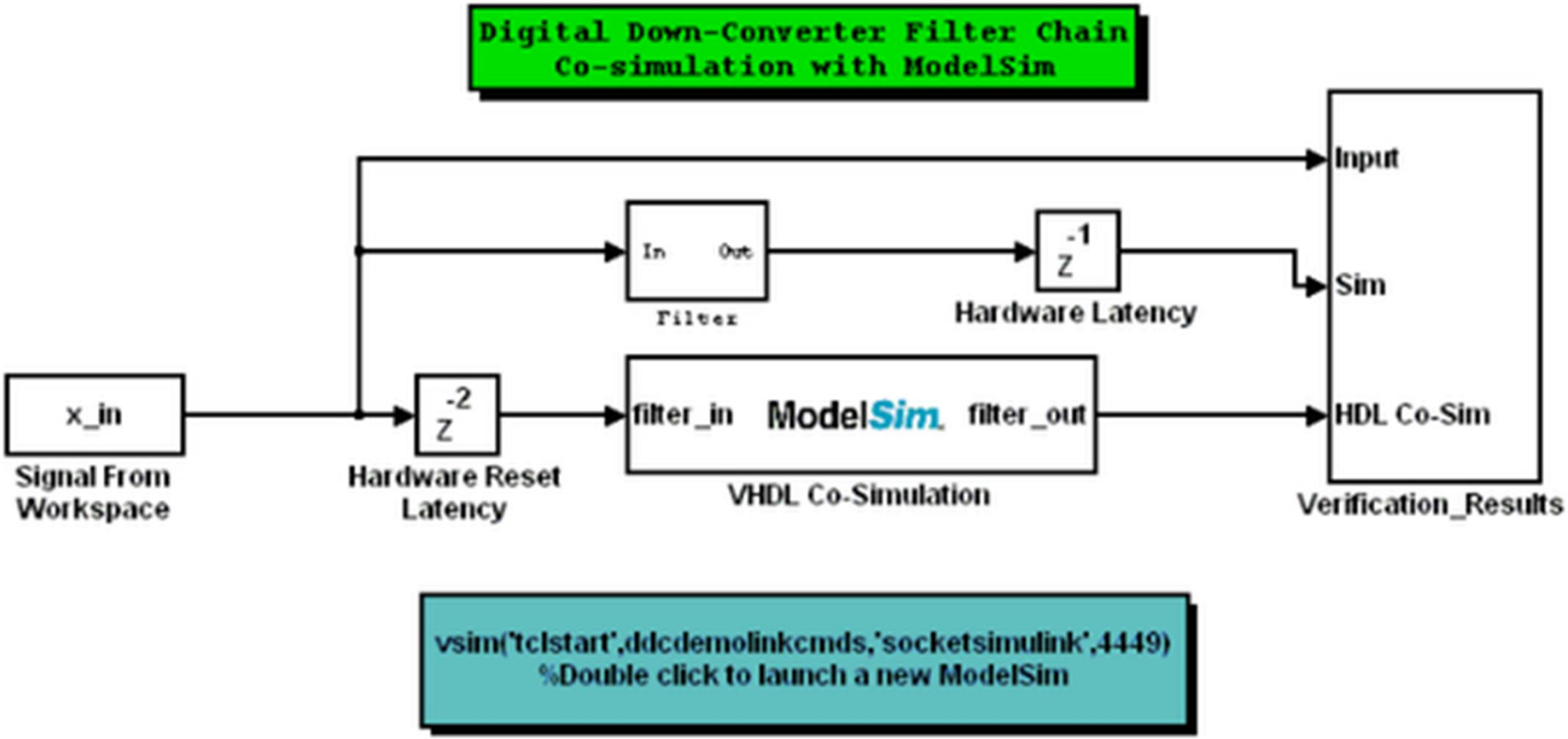

```
upsample(x,L,p)
```

Cascaded Integrator-Comb
CIC filtri

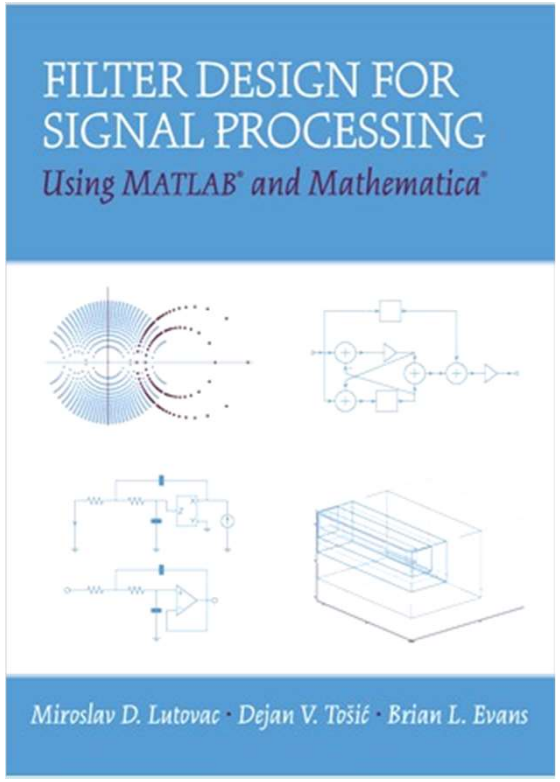
$$H(z) = H_I^N(z)\,H_C^N(z) = \frac{\left(1 - z^{-D}\right)^N}{\left(1 - z^{-1}\right)^N} = \left[\sum_{k=0}^{D-1} z^{-k}\right]^N$$



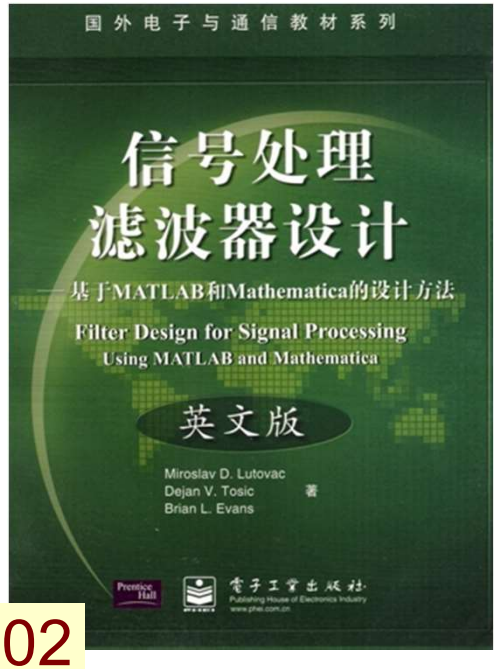
DDC – digital down converter



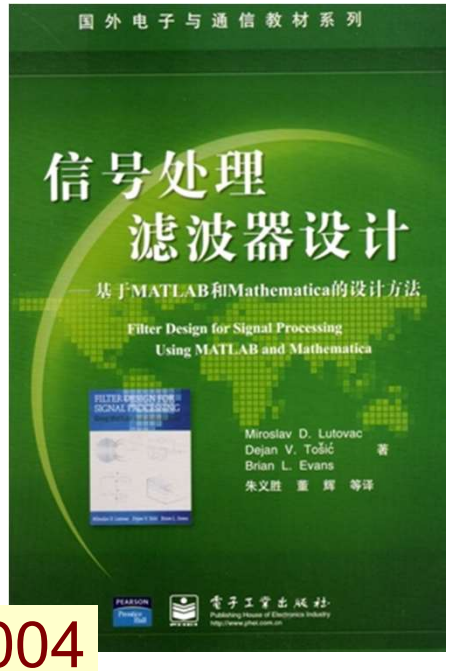
Further reading



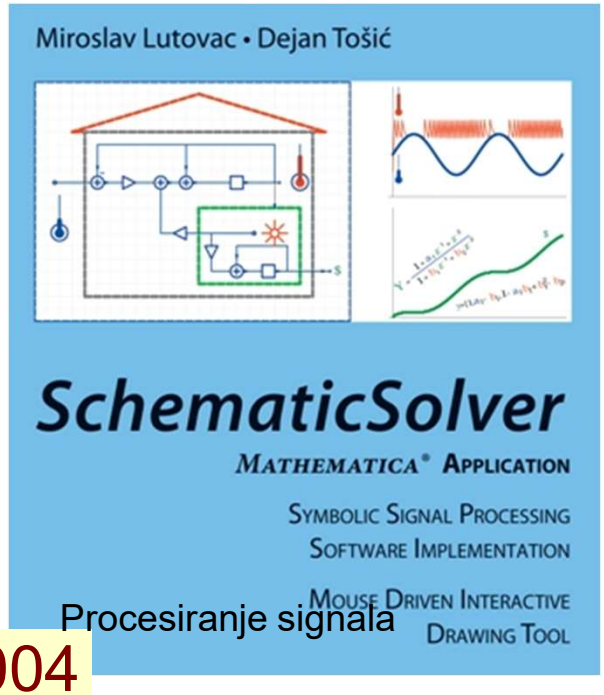
2001



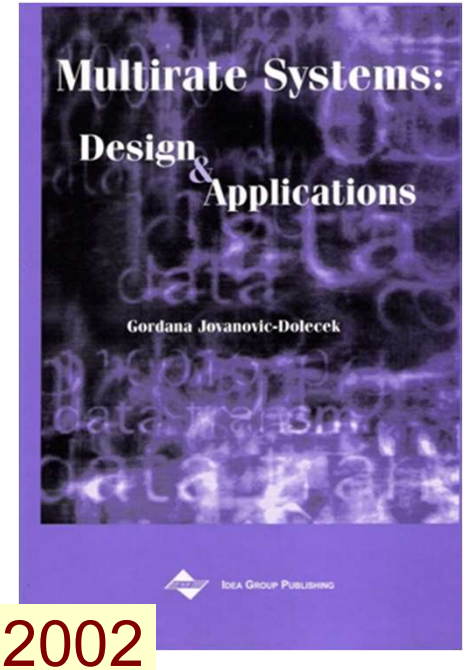
2002



2004



2004



2002

Procesiranje signala

Profesor dr Miroslav Lutovac
mlutovac@viser.edu.rs