5. SCHIMBAREA RATEI DE EŞANTIONARE. APLICAȚII ALE CIRCUITELOR MULTIRATĂ

5.1. Introducere

În multe aplicaţii este necesară schimbarea frecvenţei de eşantionare a semnalului. Fie secvența numerică $x[n]$, obținută prin eşantionarea semnalului continuu $x_c(t)$:

$$x[n] = x_c(nT) \tag{5.1}$$

unde T este *perioada de eşantionare*.

Schimbarea ratei de eşantionare pentru secvența discretă $x[n]$ este echivalentă cu obținerea unei secvențe care să conțină eşantioanele semnalului continuu obținute cu o perioadă $T' \neq T$:

$$x'[n] = x_c(nT') \tag{5.2}$$

Deoarece avem la dispoziție secvența discretă $x[n]$ şi nu semnalul continuu $x_c(t)$, prezintă interes acele metode de schimbare a ratei de eşantionare care operează numai asupra semnalului discret. Acestea sunt metodele de eşantionare multirată care realizează decimarea (reducerea ratei de eşantionare) sau interpolarea (creşterea ratei de eşantionare) cu ajutorul circuitelor de decimare, respectiv expandare şi a filtrelor adevate. Deoarece creşterea sau scăderea ratei de eşantionare se face în aceste circuite cu factori întregi, obținerea unei modificări fracționare se poate realiza prin cascadarea circuitelor de decimare cu cele de interpolare sau invers.

5.2. Decimarea

Decimarea reprezintă operația de *reducere a ratei de eşantionare* a unui semnal discret cu un factor întreg *M*:

$$x_d[n] = x[nM] = x_c(nMT) \tag{5.3}$$

Fie F_M frecvența maximă a semnalului continuu $x_c(t)$. Condiția ca semnalul $x_c(t)$ să poată fi reconstituit din eşantioanele semnalului decimat $x_d[n]$ este ca prin reducerea ratei de eşantionare să fie îndeplinită *condiția Nyquist*:

$$F_S' \geq 2F_M \tag{5.4}$$

unde $F_S' = \frac{1}{T'} = \frac{1}{MT} = \frac{F_s}{M}$ este rata de eşantionare redusă cu factorul *M*.

Dacă nu se respectă condițiile de mai sus în urma decimării apare fenomenul de aliere (suprapunerea spectrelor pentru semnalul decimat). Pentru a evita alierea se introduce înaintea circuitului elementar de decimare un filtru trece-jos ca în figură:

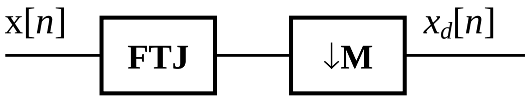


Figura 5.1. Circuitul complet de decimare cu factorul *M*.

Frecvența (normată) de tăiere a filtrului este:

$$\omega_t = \frac{\pi}{M}. \tag{5.5}$$

E1. ***Exercițiu***:

a) Să se genereze următorul semnal:

$$x[n] = \sin(2\pi f_1 n) + \sin(2\pi f_2 n), \qquad n = 0 : N-1 \tag{5.6}$$

unde f_1 şi f_2 sunt frecvențele normate corespunzătoare frecvențelor $F_1 = 1000Hz$, $F_2 = 3500Hz$, frecvența de eşantionare fiind $F_s = 20kHz$. Numărul de eşantioane este $N = 64$.

b) Să se realizeze decimarea acestui semnal cu $M=2$ şi $M=4$. Se vor reprezenta spectrul semnalului iniţial şi spectrul semnalului decimat.

Întâi să realizăm reducerea frecvenţei de eşantionare cu un decimator elementar care reţine eşantioanele multiplii de *M*:

$$xd[n]=x[nM]$$

```
xd = x(1:M:N);
X = abs(fftshift(fft(x,256)));
Xd = abs(fftshift(fft(xd,256)));
f = linspace(-0.5,0.5,256);
figure(1)
subplot(211),plot(f,X),grid,title('Spectrul lui x')
subplot(212),plot(f,Xd),grid,title('Spectrul lui xd')
```

Cum se modifică spectrul prin decimare? Completaţi tabelul:

	$F_s' = \frac{F_s}{M}$	Frecvenţe normate		Frecvenţe nenormate	
		f_1'	f_2'	F_1' (Hz)	F_2' (Hz)
$M=2$					
$M=4$					

Pentru a evita fenomenul de aliere (observat pentru frecvenţa F_2' dacă $M=4$) semnalul de intrare trebuie filtrat cu un filtru trece jos cu frecvenţa de tăiere $\omega_t = \pi / M$.

c) Obţineţi coeficienţii filtrului antialiere de ordin 30 cu funcţia **`fir1`**. Filtraţi semnalul de intrare cu filtrul proiectat.

Semnalul filtrat se obţine astfel:

```
xf = filter(h,1,x);
```

unde **`h`** sunt coeficienţii funcţiei pondere ai filtrului FIR proiectat.

d) Decimaţi semnalul filtrat cu $M=2$ şi $M=4$. Reprezentaţi spectrul semnalului filtrat şi spectrul semnalului filtrat şi decimat. Comentaţi diferenţele faţă de rezultatele obţinute anterior.

Funcția Matlab **`decimate`** implementează un decimator, adică filtrează trece jos semnalul dat, după care îl decimează cu rata specificată.

***Sintaxe*:**

`y = decimate(x,D)`
- vectorul `x` conține valorile eşantioanelor secvenței de intrare $x[n]$ iar `D` este factorul de decimare. Pentru evitarea alierii se foloseşte în mod implicit un filtru trece jos de tip Cebîşev I, de ordinul 8. Va rezulta vectorul `y` ce conține valorile eşantioanelor semnalului de ieşire $y[m]$.
- lungimea vectorului `x` trebuie să fie de cel puțin 3 ori mai mare decât ordinul filtrului folosit pentru a evita alierea. În această sintaxă se foloseşte în mod implicit un filtru de ordinul 8 deci lungimea minimă a vectorului x este 25.

`y = decimate(x,D,n)`
- aceleaşi considerente ca în sintaxa precedentă cu deosebirea că se va folosi pentru evitarea alierii un filtru trece jos de tip Cebîşev I de ordinul `n.` În acest caz vectorul `x` trebuie să aibă lungimea mai mare ca 3`n`. Nu este recomandat să se aleagă un ordin mai mare decât 13 datorită instabilității numerice (MATLAB avertizează în acest caz).

`y = decimate(x,D,'fir')`
- aceleaşi considerente ca în prima sintaxă cu deosebirea că se va folosi pentru evitarea alierii un filtru cu răspuns finit la impuls (*RFI*) de lungime 30.

`y = decimate(x,D,n,'fir')`
- aceleaşi considerente ca în a doua sintaxă cu deosebirea că se va folosi pentru evitarea alierii un filtru cu răspuns finit la impuls (*RFI*) de lungime `n`.

5.3. Interpolarea

Interpolarea reprezintă operația de *creştere a ratei de eşantionare* a unui semnal discret cu un factor întreg *L*. Aceasta constă întâi în expandarea semnalului discret inițial urmată de filtrarea trece jos.

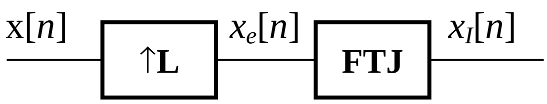


Figura 5.2. Circuitul complet de interpolare cu factorul *L*.

Expandarea semnalului iniţial se realizează prin introducerea a $L-1$ zerouri între două eşantioane succesive ale lui $x[n]$:

$$x_e[n]=\begin{cases} x[n/L], & n=0,\ \pm L,\ \pm 2L,\dots \\ 0 & \text{în} \quad \text{rest} \end{cases}, \tag{5.7}$$

După expandare, filtrul trece jos are rolul de a elimina spectrele imagine care apar în domeniul de frecvenţe normate $\omega\in[-\pi,\pi]$, în urma creşterii frecvenţei de eşantionare de L ori. Frecvenţa normată de tăiere a filtrului este:

$$\omega_t=\frac{\pi}{L}. \tag{5.8}$$

De asemenea câştigul filtrului trebuie să fie egal cu L astfel încât să fie îndeplinită relaţia între eşantioanele semnalelor $x[n]$ şi $x_i[n]$:

$$x_i[m]=x[m/L],\ \text{pentru}\ m=0,\ \pm L\dots \tag{5.9}$$

E2. ***Exerciţiu*:**

a) Se va genera un semnal sinusoidal:

$$x[n]=\sin(0.4\pi n) \qquad n=0:N-1 \tag{5.10}$$

unde $N=64$.

b) Se va realiza mărirea ratei de eşantionare prin expandare cu $L=3$.

$$xe[n]=\begin{cases} x[n/L] & \text{pentru} \quad n=kL,\quad k\in\mathbf{Z} \\ 0 & \text{în} \quad \text{rest} \end{cases}$$

```
xe = zeros(1,L*N);
xe(1:L:L*(N-1)) = x;
```

În figura 1 se va reprezenta o porţiune din cele două semnale:

```
figure(1)
subplot(311),stem(0:10,x(1:11)),grid;
subplot(312),stem(0:L*10,xe(1:L*10+1)),grid;
```

În figura 2 se vor reprezenta spectrele celor două semnale.

```
X = abs(fftshift(fft(x,256)));
Xe = abs(fftshift(fft(xe,256)));
f = linspace(-0.5,0.5,256);
```

```
figure(2)
subplot(311),plot(f,X),grid,title('Spectrul lui x')
subplot(312),plot(f,Xe),grid,title('Spectrul lui xe')
```

Cum se modifică spectrul prin expandare? Explicaţi apariţia componentelor spectrale suplimentare în urma expandării.

c) Structura completă a circuitului de interpolare se obţine prin adăugarea după expandare a unui filtru trece jos cu frecvenţa de tăiere $\omega_t = \pi / L$ şi cu câştigul egal cu L, care să elimine spectrele imagine. Efectul acestei filtrări în domeniul timp este de a reface eşantioanele semnalului în punctele unde s-au introdus zerouri prin obţinerea unui semnal numeric cu spectrul identic cu cel al semnalului analogic.

Proiectaţi un filtru trece jos de ordin 30 cu specificaţiile de mai sus şi filtraţi semnalul expandat.

d) Se va reprezenta semnalul **xi** obţinut în urma filtrării în aceeaşi figură cu semnalele **x** şi **xe** şi spectrul acestui semnal în figura 2 (împreună cu celelalte spectre):

```
figure(1),
subplot(313),stem(0:L*10,xi(16:L*10+16)),grid;
Xi = abs(fftshift(fft(xi,265)));
figure(2)
subplot(313),plot(f,Xi),grid,title('Spectrul lui x')
```

Reluaţi exerciţiul pentru $L = 6$.

Funcţia Matlab **`interp`** implementează un interpolator, adică expandează semnalul dat cu rata specificată, după care îl filtrează trece jos.

***Sintaxe*:**

`y = interp(x,U)`

- vectorul `x` conţine valorile eşantioanelor secvenţei de intrare $x[n]$ iar `U` este factorul de interpolare. Va rezulta vectorul `y` ce conţine valorile eşantioanelor semnalului de ieşire $y[m]$.
- se foloseşte în mod implicit un filtru anti-imagine de lungime 4 cu frecvenţa de tăiere normată 0,5. Lungimea vectorului `x` trebuie să fie de cel puţin 2(lungimea filtrului)+1. În această sintaxă se foloseşte în mod implicit un filtru de lungime 4 deci lungimea minimă a vectorului `x` este 9.

`y = interp(x,U,l,ft)`

- aceleaşi considerente ca în prima sintaxă cu deosebirea că se va folosi un filtru anti-imagine de lungime l şi având frecvenţa de tăiere normată `ft`. În acest caz vectorul `x` trebuie să aibă lungimea de cel puţin `2l+1`.

`[y,b] = interp(x,U,l,ft)`

- `y`, `x`, `U`, `l`, `ft` au aceleaşi semnificaţii ca în sintaxa precedentă; se va returna în plus vectorul `b` ce va conţine coeficienţii filtrului anti-imagine.

5.4. Modificarea fracţionară a ratei de eşantionare

În multe aplicaţii practice ale procesării semnalelor digitale apare problema schimbării frecvenţei de eşantionare a semnalului prin creşterea sau scăderea acesteia. Conversia cu factorul raţional L/M a ratei de eşantionare se poate realiza interpolând mai întâi semnalul cu un factor L şi apoi decimând ieşirea interpolatorului cu factorul M. Schema corespunzătoare constă din interconectarea în cascadă a unui interpolator cu un decimator :

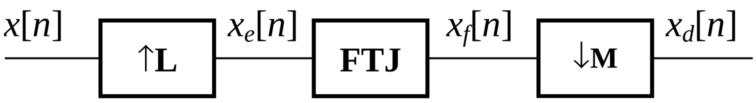


Figura 5.3. Circuitul de modificare fracţionară a ratei de eşantionare.

Este important să se realizeze mai întâi interpolarea şi după aceea decimarea pentru a prezerva caracteristicile spectrale dorite ale lui $x[n]$. Filtrul trece jos din Figura 5.3. încorporează operaţiile de filtrare pentru interpolare şi decimare. Expresia lui $H(e^{j\omega})$ este:

$$H(e^{j\omega}) = \begin{cases} L, & 0 \le |\omega| \le \min\left(\dfrac{\pi}{M}, \dfrac{\pi}{L}\right) \\ & 0, \ \text{in rest} \end{cases} \tag{5.11}$$

Pentru o bună înţelegere se recomandă parcurgerea teoriei din cadrul sistemelor cu eşantionare multirată referitoare la conversia ratei de eşantionare printr-un factor raţional L/M.

E3. ***Exerciţiu*:**

Fie un semnal sinusoidal de frecvenţă $F_1 = 8kHz$, eşantionat cu frecvenţa $F_s = 32kHz$. Lungimea semnalului este $N = 256$. Se doreşte modificarea frecvenţei de eşantionare la:

1. $F_{s1} = 48kHz$.
2. $F_{s2} = 20kHz$.
3. $F_{s3} = 12kHz$.

a) Stabiliţi factorii de interpolare *L*, respectiv de decimare *M* pentru fiecare caz.
b) Proiectaţi filtrul trece jos din compunerea circuitului de modificare fracţionară a ratei de eşantionare.
c) Efectuaţi schimbarea ratei de eşantionare pentru semnalul dat.
d) Reprezentaţi spectrele semnalelor $x[n]$, $x_e[n]$, $x_f[n]$ şi $x_d[n]$.

Funcţia Matlab **`resample`** implementează modificarea fracţionară a ratei de eşantionare pentru un semnal dat.

***Sintaxe*:**

`y = resample(x,U,D)`

- vectorul `x` conţine valorile eşantioanelor secvenţei de intrare $x[n]$, `U` este factorul de interpolare iar `D` este factorul de decimare astfel încât se va realiza conversia ratei de eşantionare prin factorul raţional U/D. Va rezulta vectorul `y` ce conţine valorile eşantioanelor semnalului de ieşire $y[m]$;
- în mod implicit se foloseşte un filtru trece jos cu răspuns finit la impuls, proiectat cu ajutorul procedurii `fir1` folosind o fereastră Kaiser cu parametrul `beta = 5`.

`[y,b] = resample(x,U,D)`

- se returnează în plus faţă de prima sintaxă vectorul `b` ce conţine coeficienţii filtrului folosit în mod implicit.

`y = resample(x,U,D,b)`

- `y`, `x`, `U` şi `D` au aceleaşi semnificaţii ca în prima sintaxă ;
- vectorul `b` va conţine coeficienţii filtrului pe care dorim să-l folosim în locul filtrului folosit implicit în prima sintaxă.

5.5. Aplicații ale circuitelor multirată

5.5.1. Filtrarea FIR prin decimare

În acest capitol se demonstrează utilitatea mai multor etaje de decimare pentru obținerea unei filtrări cu bandă foarte îngustă şi volum redus de calcule.

Să presupunem că avem un semnal cu frecvența maximă $F_M = 4kHz$, eşantionat cu frecvența de eşantionare $F_s = 8kHz$. Dorim să filtrăm semnalul pentru obținerea componentelor situate în banda $0 - 75Hz$ cu o bandă de tranziție de la 75 la 80 *Hz*. Riplul maxim al filtrului în banda de trecere este 0.01 iar cel în banda de oprire 10^{-4}.

Pentru obținerea acestor specificații cu un singur filtru (fără a utiliza decimarea) putem folosi algoritmul Remez. Ordinul filtrului se determină cu funcția **remezord**. Sintaxa acestei funcții este:

```
remezord([F1 F2], [A1 A2], [R1 R2], Fs)
```

unde: F1 este frecvența maximă a benzii de trecere;
F2 este frecvența minimă a benzii de oprire;
A1 este amplificarea filtrului în banda de trecere;
A2 este amplificarea filtrului în banda de oprire;
R1 este riplul maxim al filtrului în banda de trecere;
R2 este riplul maxim al filtrului în banda de oprire;
Fs este frecvența de eşantionare la care lucrează filtrul.

E4. ***Exercițiu*:**
a) Să se determine ordinul filtrului cu specificațiile de mai sus.

Un semnal cu frecvența maximă de 80*Hz*, eşantionat cu $F_s = 8kHz$ este evident *supraeşantionat*. După cum s-a arătat anterior, prin decimare cu *M* frecvența de eşantionare se reduce de *M* ori.

b) Calculați cât trebuie ales *M* pentru a se obține frecvența minimă de eşantionare, care să îndeplinească condiția Nyquist pentru semnalul cu frecvența maximă de 80*Hz*.

Realizăm decimarea cu *M* în două etape, cu două circuite de decimare, primul cu $M_1 = 25$ şi al doilea cu $M_2 = 2$.

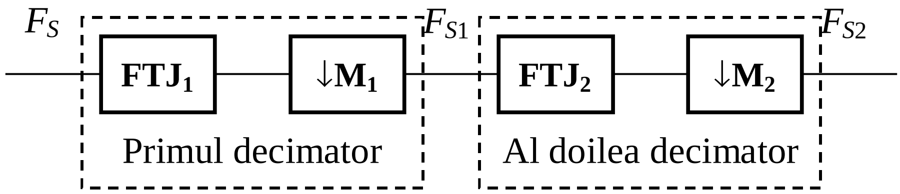


Figura 5.4. Filtrare cu două etaje de decimare

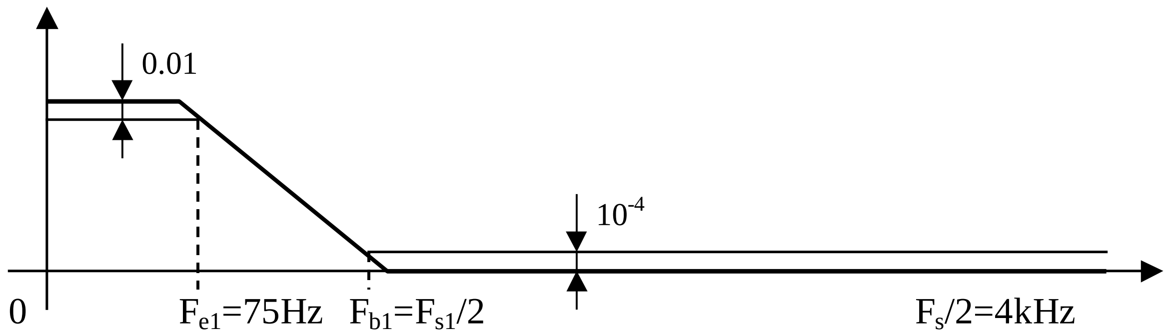
Pentru a evita fenomenul de aliere primul filtru trece jos trebuie să aibă frecvenţa minimă în banda de oprire:

$$F_{b1} = \frac{F_{s1}}{2} = \frac{F_s}{2M_1} \tag{5.12}$$

De asemenea pentru obţinerea componentelor situate în banda $0-75Hz$ trebuie ca frecvenţa maximă în banda de trecere să fie:

$$F_{e1} = 75Hz \tag{5.13}$$

Gabaritul filtrului este prezentat mai jos:

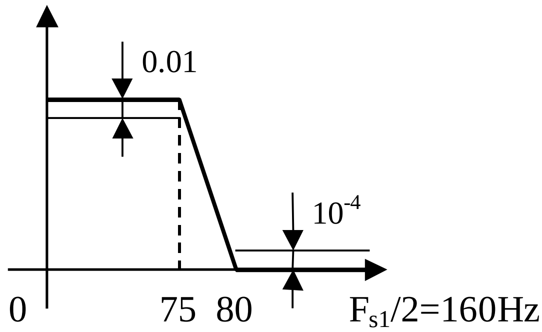


c) Calculaţi F_{s1} şi F_{b1}. Proiectaţi primul filtru folosind algoritmul Remez. Cât este ordinul primului filtru?

Atenţie!

Primul filtru lucrează tot la frecvenţa $F_s = 8kHz$. Numai după circuitul de decimare frecvenţa de eşantionare scade cu M_1. Ordinul filtrului este mai mic deoarece banda de tranziţie între F_{b1} şi F_{e1} este mai mare.

Pentru al doilea filtru trece jos condiţiile de proiectare sunt cele ale filtrului iniţial (bandă de tranziţie de la 75 la 80Hz), doar că frecvenţa lui de lucru este în acest caz F_{s1}. Aceasta conduce la o bandă de tranziţie în *frecvenţe normate* mai mare.



d) Proiectați al doilea filtru folosind algoritmul Remez. Cât este ordinul filtrului? Care este lungimea totală a celor două filtre? Reprezentați caracteristicile filtrelor proiectate.

5.5.2 Translația de frecvență prin decimare - interpolare

E5. ***Exercițiu*:**

Fie un semnal de bandă îngustă (de exemplu un semnal modulat în amplitudine):

$$x[n] = [1 + 0.8\cos(2\pi f_1 n)]\cos(2\pi f_0 n)$$

unde f_1 şi f_0 sunt frecvenţele normate pentru $F_1 = 600Hz$, $F_0 = 7000Hz$, frecvența de eşantionare $F_s = 20kHz$. Numărul de eşantioane este $N = 256$.

a) Determinați banda (în frecvențe normate) ocupată de semnal. Reprezentați spectrul semnalului.

```
N_fft = 512;
f = linspace(-0.5,0.5,N_fft);
X = abs(fftshift(fft(x,N_fft)));
subplot(411),plot(f,X),grid,title('Spectrul lui x')
```

b) Cât trebuie ales M astfel ca prin decimare cu un decimator elementar (fără filtru trece jos) să nu apară alierea şi banda semnalului decimat să ocupe tot domeniul de frecvență $[0, \pi)$? Deduceți o relație de calcul între M, F_s şi bandă.

c) Decimați semnalul cu M calculat anterior. Reprezentați spectrul semnalului decimat şi verificați că nu apare alierea.

```
xd=x(1:M:N);
Xd = abs(fftshift(fft(xd,N_fft)));
subplot(412),plot(f,Xd),grid,title('Spectrul lui xd')
```

d) Expandaţi semnalul cu acelaşi *M* şi reprezentaţi spectrul semnalului obţinut.

```
xe = zeros(1,N);
xe(1:M:N) = xd;
Xe = abs(fftshift(fft(xe,N_fft)));
subplot(413),plot(f,Xe),grid,title('Spectrul
semnalului xe')
```

e) Câte spectre imagine au apărut? Care este lărgimea benzii normate ocupate de un astfel de spectru faţă de banda semnalului iniţial?

Printr-o filtrare trece bandă convenabil aleasă se poate selecta oricare din spectrele imagine, rezultând o translaţie a spectrului iniţial.

f) Proiectaţi un filtru trece bandă care să reţină al doilea spectru al semnalului expandat. Ordinul filtrului trebuie ales suficient de mare ca să rejecteze frecvenţele spectrelor vecine. Reprezentaţi spectrul obţinut.

```
xf = filter(h,1,xe); % se filtreaza semnalul expandat
Xf = abs(fftshift(fft(xf,N_fft)));
subplot(414),plot(f,Xf),grid,title('Spectrul lui xf')
```