

# SFSK Modem

## Cap.1 Modelarea si Analiza folosind MATLAB script File

### Sectiunea Parametri

Parametrii modemului sint setati cu ajutorul functiei Matlab `sys=param()`, care returneaza o structura cu cimpurile de ma jos:

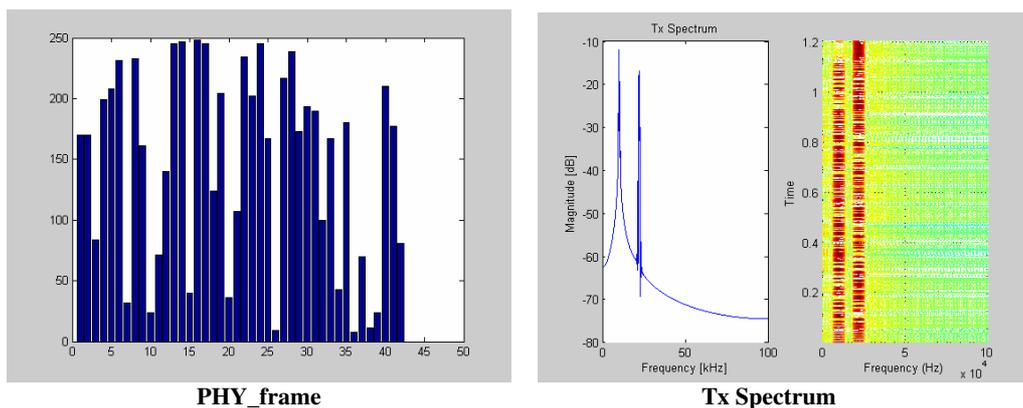
```
f: [10000 22000 200000]
time: [1x84 double]
N: 84
pw: [0 0]
speed: 2400
SNR: {[ -7] [1x107 double]}
narrow: {[1x86 double] [5] [15500] [1x101 double]}
impulse: {[1x81 double] [5] [10] [1x101 double] [1000]}
```

Se recomanda vizualizarea in prealabil a fisierului `param.m` si setarea valorilor dorite pentru Modem si anume:

- Frecventa retelei electrice
- Viteza de transmisie
- Frecventele de semnalizare
- Frecventa de esantionare utilizata in system
- Puterile de emisie pe cei 2 biti
- Raportul semnal zgomot la testarea pe un canal AWGN
- Vectorul SNR la evaluarea BER pe un canal AWGN
- Parametrii zgomotului de tip impuls si a celui armonic de banda ingusta.

### Sectiunea Tx

Implementarea partii de emisie ce include partea generare a cadrului de pe nivelul fizic, partea de modulare prin decizie si partea de filtrare este realizata in scriptul Matlab `S_FSK_Tx`. Rezultatele grafice ale scriptului sint redade mai jos. Acestea se refera la vizualizarea cadrului de octeti si a semnalului modulat in domeniul frecventa:

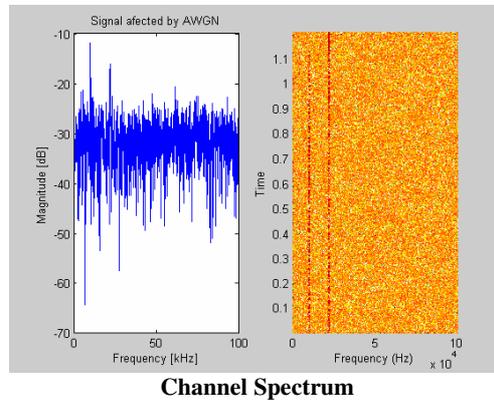


In acest exemplu am considerat o viteza de 300 bps, frecvente de semnalizare de 10 si respectiv 22 kHz, iar frecventa de esantionare 200 kHz. S-au utilizat amplitudini unitare pentru semnale de semnalizare rezultind o putere a semnalului modulat si filtrat  $\approx -3.8$  dB

**Observatie:** Se recomanda utilizarea vitezei de transmisie in concordanta cu frecventele de sistem pentru a nu produce un volum mare de date de procesat pentru MATLAB

## Sectiunea Ch

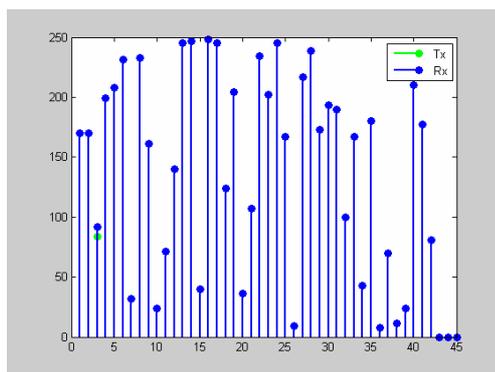
Implementarea si modelarea unui canal AWGN se face in scriptul S\_FSK\_Ch. Puterea de zgomot este evaluata pe baza SNR fixat in fisierul de configurare si a puterii de semnal estimata mai sus. Rezultatul trecerii semnalului printr-un canal AWGN cu SNR=-7 dB este redat mai jos:



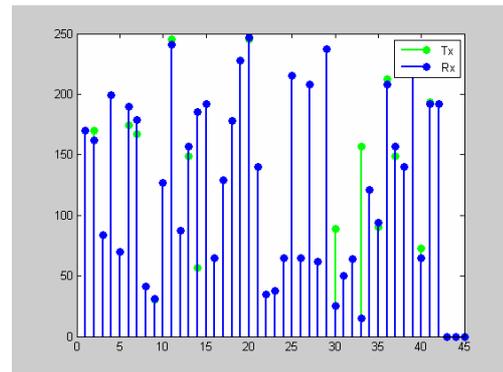
Channel Spectrum

## Sectiunea Rx

Demodularea s-a realizat cu filtre trece banda centrate pe cele 2 frecvente de semnalizare. Rezultatul demodularii este redat mai jos pentru 2 valori SNR, -7 si respectiv -12 dB. Rezultind astfel o ratele de eroare de 0.027778 si respectiv 0.05



SNR=-7 dB, BER=0.027778



SNR=-12 dB, BER=0.005

## Sectiunea Performante

Pentru evaluarea performantelor unui astfel de sistem s-au realizat trei functii care implementeaza cele trei entitati (Tx+Ch+Rx). Functiile sint redade mai jos:

```
function [signal_SFSK,bitsTx]=Tx()  
% SFSK Transmitter Module  
sys=param();  
F_mark=sys.f(1);  
F_space=sys.f(2);  
Fs=sys.f(3);  
Am=10^(sys.pw(1)/20);  
As=10^(sys.pw(2)/20);  
speed=sys.speed;  
% Physical Frame parameters settings  
sizeP_sdu=38;  
preamble=uint8([hex2dec('AA') hex2dec('AA')]);  
StartSubFrDelimiter=uint8([hex2dec('54') hex2dec('C7')]);  
pause=uint8(zeros(1,3));  
N=sys.N;  
% Generate one frame
```

```

PHY_frame.Preamble=preamble;
PHY_frame.StartSubFrameDelimiter=StartSubFrDelimiter;
PHY_frame.Pause=pause;
PHY_frame.P_sdu=uint8(randsrc(1,sizeP_sdu,[0,255]));
% Convert frame-byte to frame bits
bitsTx=dec2bin([PHY_frame.Preamble,PHY_frame.StartSubFrameDelimiter, ...
PHY_frame.P_sdu,PHY_frame.Pause],8);
% Modulation process
signal_SFSK=zeros(360*sys.N,1);
k=0;
for i=1:45
    for j=1:8
        if bitsTx(i,j)=='1'
            signal_SFSK(k*N+1:(k+1)*N)=Am*sin(2*pi*sys.f(1)*sys.time);
        else
            signal_SFSK(k*N+1:(k+1)*N)=As*sin(2*pi*sys.f(2)*sys.time);
        end
        k=k+1;
    end
end
nmTx=cheb1ord(2*[F_mark-2*speed, F_mark+2*speed]/Fs,...
2*[F_mark-4*speed, F_mark+4*speed]/Fs, 1, 30);
nsTx=cheb1ord(2*[F_space-2*speed, F_space+2*speed]/Fs,...
2*[F_space-4*speed, F_space+4*speed]/Fs,1,30);
[bmTx,amTx]=cheby1(nmTx,1,2*[F_mark-2*speed, F_mark+2*speed]/Fs);
[bsTx,asTx]=cheby1(nsTx,1,2*[F_space-2*speed, F_space+2*speed]/Fs);
signal_SFSK=filter(bmTx,amTx, signal_SFSK)+...
filter(bsTx, asTx, signal_SFSK);

```

```

function out=Ch(in,channel_type,varargin)
sys=param();
N=sys.N;
switch channel_type
    case 'awgn'
        SNR=varargin{1};
        Pin=10*log10(var(in));
        Npower=Pin-SNR;
        noise_awgn=10^(Npower/20)*randn(360*sys.N,1);
        out=in+noise_awgn;
    case 'narrow'
        [F_noise,am]=varargin{1:2};
        noise_narrow=zeros(360*sys.N,1);
        t=(0:360*sys.N)/sys.f(3);
        noise_narrow=am*sin(2*pi*F_noise*t(1:end-1));
        out=in+noise_narrow';
    case 'impulse'
        [F_imp,duty,A]=varargin{1:3};
        noise_imp=zeros(360*sys.N,1);
        t=(0:360*sys.N)/sys.f(3);
        noise_imp=A*square(2*pi*F_imp*t(1:end-1),duty);
        out=in+noise_imp';
end

```

```

function bitsRx=Rx(in)
sys=param();
% Design Filters for half-channels
WsmRx=2*[sys.f(1)-4*sys.speed, sys.f(1)+4*sys.speed]/sys.f(3);
WpmRx=2*[sys.f(1)-2*sys.speed, sys.f(1)+2*sys.speed]/sys.f(3);
nmRx=cheb1ord(WpmRx,WsmRx,1,20);
[bmRx,amRx]=cheby1(nmRx,1,WpmRx);

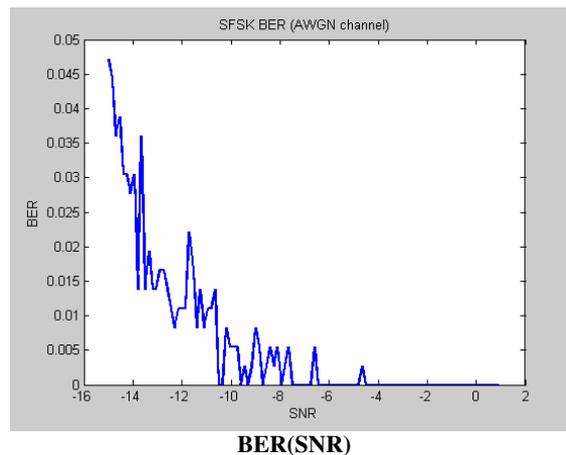
```

```

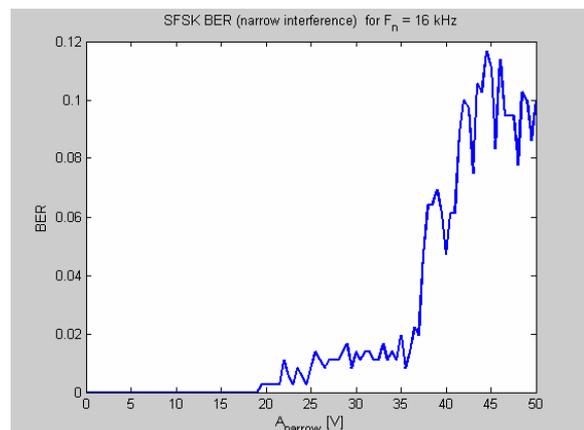
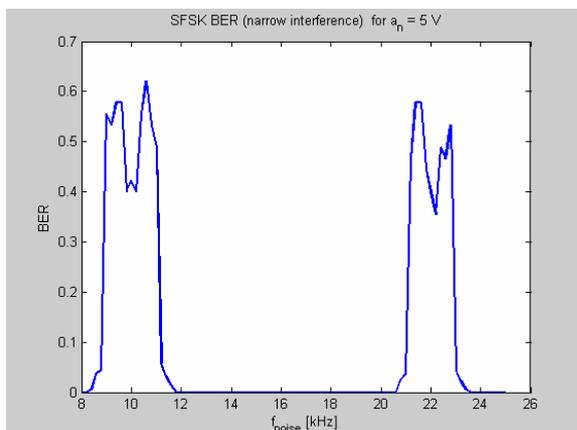
WssRx=2*[sys.f(2)-4*sys.speed, sys.f(2)+4*sys.speed]/sys.f(3);
WpsRx=2*[sys.f(2)-2*sys.speed, sys.f(2)+2*sys.speed]/sys.f(3);
nsRx=cheb1ord(WpsRx,WssRx,1,30);
[bsRx,asRx]=cheby1(nsRx,1,WpsRx);
% S-FSK demodulation
bitsRx=cell(1,1);
SFSK_reshaped=reshape(in,sys.N,360);
for i=1:360
    buffer=SFSK_reshaped(:,i);
    signal_ch_mark=filter(bmRx,amRx,buffer);
    signal_ch_space=filter(bsRx,asRx,buffer);
    d_mark=var(signal_ch_mark);
    d_space=var(signal_ch_space);
    if d_mark>d_space
        bitsRx{1}(i)='1';
    else
        bitsRx{1}(i)='0';
    end
end
bitsRx=reshape(bitsRx{1},8,45)';

```

In continuare analizam performanta sistemului pe un canal AWGN. Scriptul Matlab care realizeaza aceasta este BER\_AWGN.m. Testul este realizat prin variatia SNR de la -15 dB la 1 dB cu un pas de variatie fixat. Pentru fiecare valoare SNR se face transmitia unui cadru si evaluarea BER. Ma jos se reda acest test pentru viteza de 300 bps.

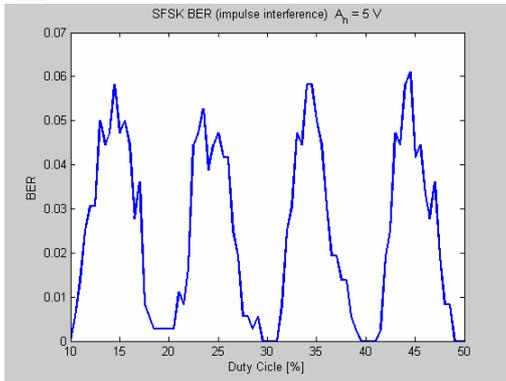


Scriptul Matlab BER\_Narrow.m testeaza imunitatea SFSK la zgomot de tip interferenta de banda ingusta. Testul include partea de evaluare a BER functie de amplitudinea interferentei pentru o valoare fixa de frecventa si respectiv partea de evaluare BER functie de frecventa interferentei pentru o valoare fixa a amplitudinii acesteea. Datele de test unui astfel de canal se afla in fisierul de configurare.

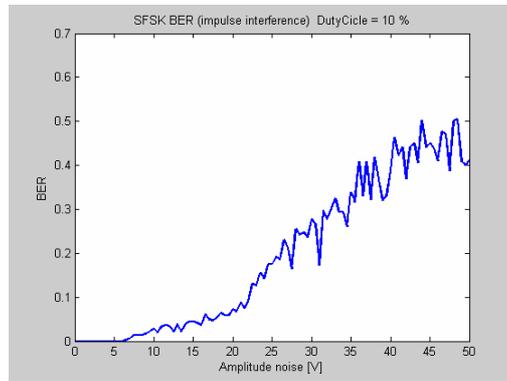


Mai jos se analizeaza performanta in raport cu zgomotul de tip impuls care poate sa apara pe canalul de comunicatie. Conform specificatiilor se iau in considerare doua frecvente de 100 Hz si respectiv 1000 Hz.

Se va evalua BER in functie de factorul de umplere si respectiv si respectiv amplitudinea impulsului.  
**F=1000 Hz**

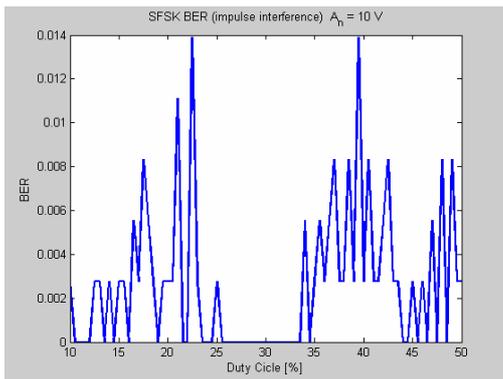


**F=1000 Hz, A=5 V**

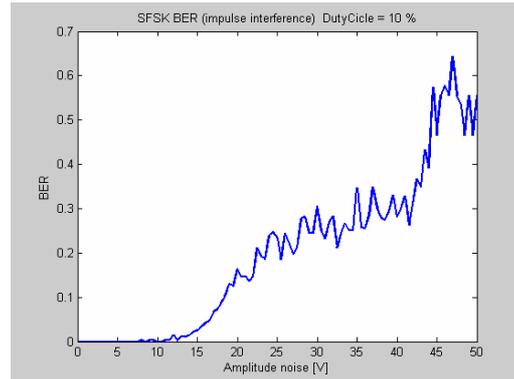


**F=1000 Hz, Duty Cycle=10 %**

**F=100 Hz**

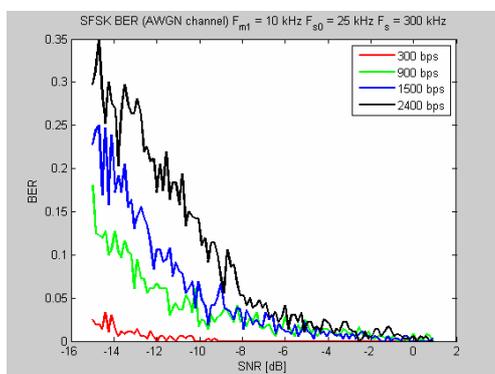


**F=100 Hz, A=10 V**

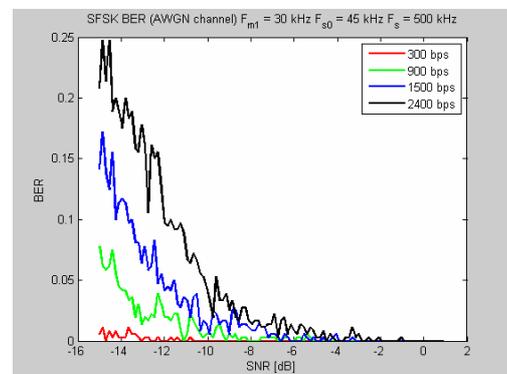


**F=100 Hz, Duty Cycle=10 %**

Prin rularea script-ului Matlab BER\_AWGN\_speed.m se obtinea reprezentarea BER functie de SNR pentru 4 valori reprezentative ale vitezei de transmisie. Deasemenea s-a refacut testul si pentru alte valori ale frecventelor de semnalizare si esantionare. Se observa o imbunatatire BER(SNR) la cresterea celor doua frecvente de semnalizare si a celei de esantionare.



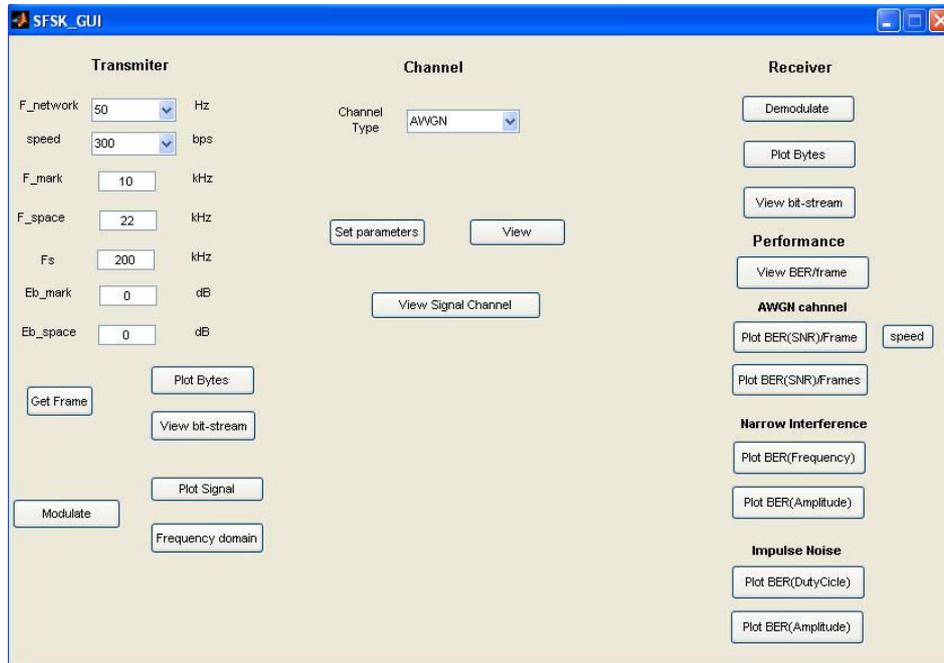
**F<sub>mark</sub>=10kHz, F<sub>space</sub>=25 kHz, F<sub>s</sub>=300 KHz**



**F<sub>mark</sub>=30kHz, F<sub>space</sub>=45 kHz, F<sub>s</sub>=500 KHz**

## Cap.2 Graphical User Interface

Pentru a deschide interfata grafica care inglobeaza functiile si script-urile de mai sus se introduce in linia de comanda *SFSK\_GUI* care deschide fereastra de mai jos.



In interfata deschisa se seteaza datele din cimpurile care sint editabile. Se pot lasa si datele care sint implicit. Dupa care se apasa butonul **Get Frame**. Apare un mesaj de confirmare ca cadrul a fost generat cu success. Mai departe se poate vizualiza cadrul generat in forma Byte Stream sau Bit Stream prin apasarea butoanelor **Plot Bytes** si respectiv **View bit-stream**. Prin apasarea butonului **Modulate** se obtine semnalul modulat. Deasemenea apare si in acest caz un mesaj ca s-a format semnalul modulat ce corespunde unui cadru de 360 biti. Prin apasarea butonului **Plot Signal** se poate vizualiza o portiune de semnal in domeniul timp, iar prin apasarea butonului **Frequency Domain** se vizualizeaza semnalul in domeniul frecventa prin FFT si respectiv Short Time FFT (spectrograma semnalului).

Mai departe se seteaza tipul de canal pe care se va dori sa se transmita semnalul modulat. Se poate alege din meniul **Channel Type** unul din canale pe care apare un anumit tip specific de zgomot. Dupa care se apasa butonul **Set Parameters** pentru a seta parametrii tipului de canal selectat:

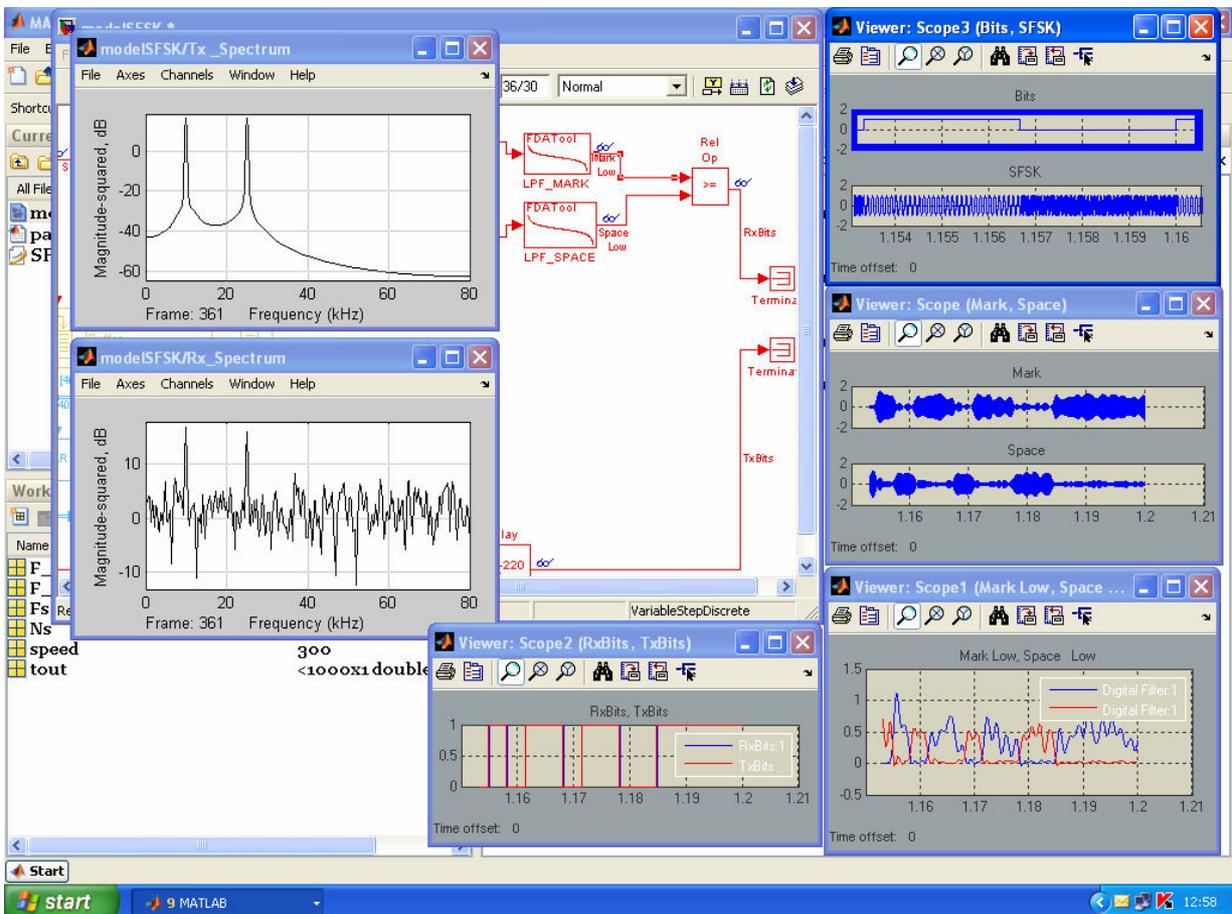
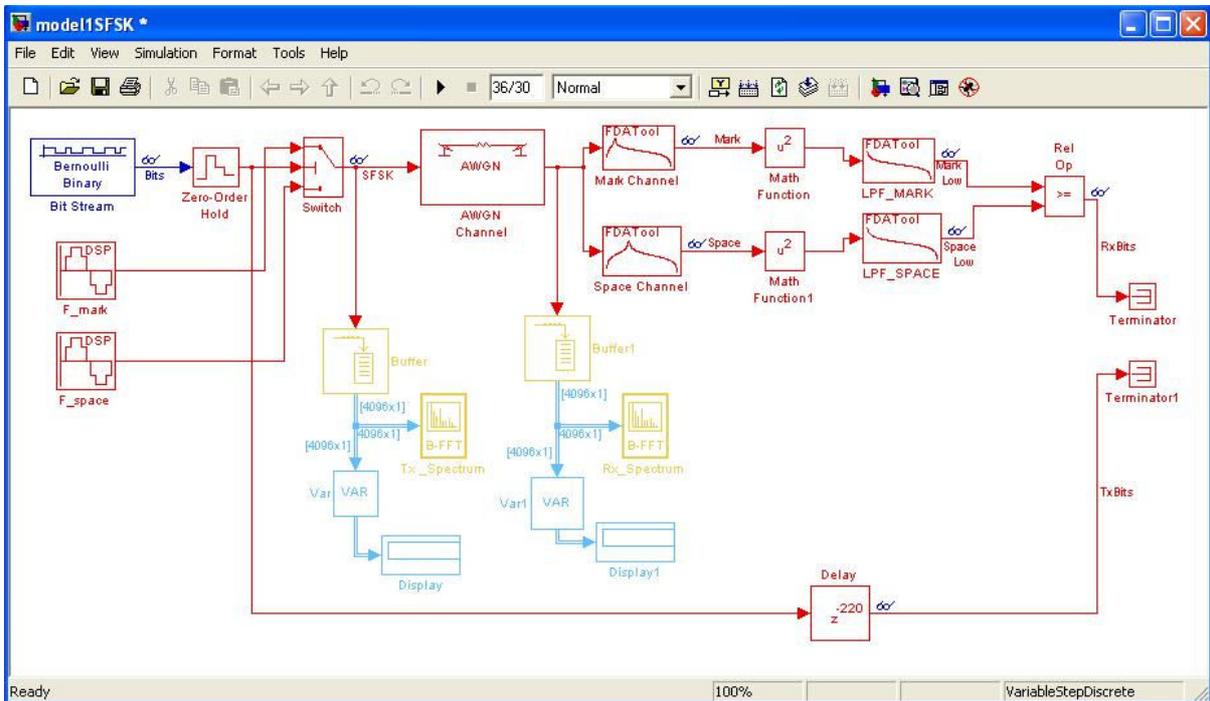
- **canal AWGN** - utilizatorul va fi promptat sa introduca valoarea SNR ce corespunde acestui canal
- **canal Narrow Band Interference** - utilizatorul va fi promptat sa introduca valoarea de amplitudine si frecventa ce corespunde semnalului armonic de interferenta
- **canal Impulse Noise** – utilizatorul este promptat sa introduca amplitudinea, factorul de umplere si frecventa trenului de impulsuri ce reprezinta zgomotul de canal

Zgomotul generat prin setarea canalului si a paramtrilor acestuia se poate vizualiza prin apasarea butonului **View**. Deasemena se poate vizualiza in frecventa semnalul modulat afectat de zgomot prin apasarea butonului **View Signal Channel**.

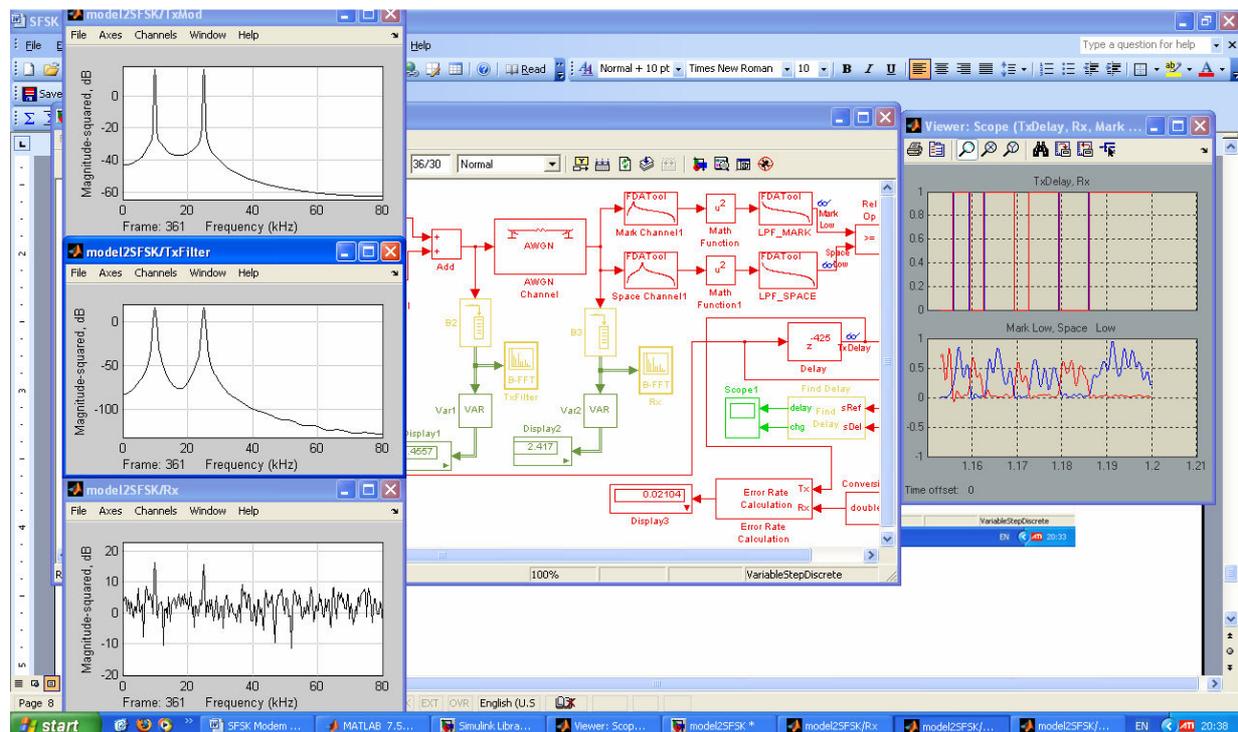
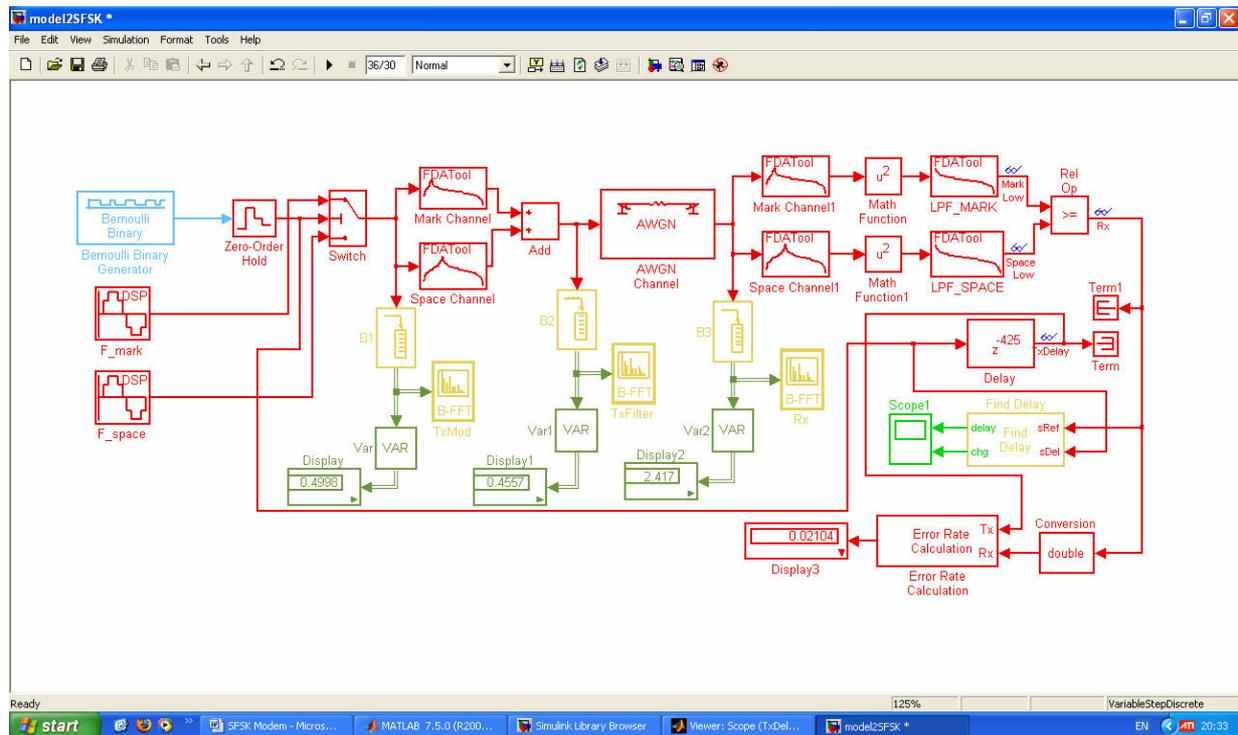
In sectiunea Receiver din GUI se poate realize operatia de demodulare a semnalului primit de pe canal. Operatia de demodulare se termina cu un mesaj de confirmare. Prin apasarea butonului **Plot Bytes** se vizualizeza secventa de octeti transmisa sic ea receptionata in aceasi fereastra. Se poate astfel vizualiza efectul canalului asupra receptiei. Efectul cantitativ al canalului asupra transmisiei se poate obtine prin apasarea butonului **View BER/frame**, care face display valorii BER pentru cadrul current ce a fost receptionat.

## Cap.3 Simulink Model

**Cazul fara filtru postmodulator:** Modelul care implementeaza modemul fara filtru postmodulator este model1SFSK.mdl din directorul /Matlab\_Work/simulink.



**Cazul cu filtru postmodulator:** Modelul care implementeaza modemul cu filtru postmodulator este model2SFSK.mdl din directorul /Matlab\_Work/simulink



**Observatii:**

- utilizarea filtrelor de postmodulare duce la reducerea componentelor spectrale din afara benzii care a fost considerata ca fiind compusa din 2 benzi disjuncte  $:[F\_mark-2*speed, F\_mark+2*speed]$  si  $[F\_space-2*speed, F\_space+2*speed]$ . Deasemenea utilizarea a 2 filtre reduce componentele spectrale dintre liniile spectrale de semnalizare, adica in banda  $[F\_mark+4*speed, F\_space-4*speed]$ .
- in toate simularile de mai sus s-a realizat demodularea utilizand o singura oportunitate oferita de un system SFSK, si anume utilizarea celor doua canale pentru decizie. Pentru a creste robustetea la zgomot

si intereferente se pot demodula separat cele 2 canale prin decizie comparativa cu o valoare de prag iar in final decizia sa se ia pe principiul "majoritatii" dintre cei trei biti.