Tema 2

1. Descrierea programului

Se proiectează în Matlab un filtru digital IIR folosind metoda directă de proiectare cu specificațiile următoare:

Filtru: Filtru Trece Sus, Eliptic
Ordin: 4
Frecvența de tăiere: 2600 Hz
Frecvența de eșantionare: 8000 Hz
Ripluri maxime în banda de trecere: 0.5 dB
Ripluri maxime în banda de oprire: 20 dB

Se generează în Matlab un semnal sinusoidal de frecvență variabilă între 0 Hz și 1300 Hz cu ajutorul funcției chirp. Durata sa este de 1 secundă și se va scrie într-un fișier input.dat.

De asemenea se realizează un proiect C pentru StarCore140 care implementează filtrul digital cu coeficienții obținuți în urma scalării funcției de transfer. Se aplică la intrarea filtrului semnalul sinusoidal generat in Matlab și se scrie semnalul de ieșire într-un fișier output.dat.

2. Detalii implementare

În prima parte a programului Matlab se calculează coeficienții filtrului digital IIR. Se prezintă următoarea parte de cod cu comentarii pe fiecare linie.

```
7   % declare the sampling freq, the cutoff freq,
8   % max Ripple value [dB] in the pass band and
9   % max Ripple value [dB] in the stop band
10  Fs      = 8000;
11  Fcutoff = 2600;
12  Rp      = 0.5;
13  Rs      = 20;
14
15  % declare the stop band limits and the pass band limits
16  Ws = (Fcutoff/Fs)*2 - 0.1;
17  Wp = (Fcutoff/Fs)*2;
18
19  % determine the order of the HP elliptic filter (not required)
20  [orderComputed, Wcutoff] = ellipord(Wp, Ws, Rp, Rs);
21
22  % determine the filter coefficients
23  order = 4; % was computed before for sanity check
24  [b, a] = ellip(order, Rp, Rs, Wp, 'high');
25
```

Figura 1: Cod Matlab liniile 7-25

Linia 10-13: Se declară variabilele în care se stochează frecvența de eșantionare, frecvența de tăiere, amplitudinea maximă a riplurilor din banda de trecere și amplitudinea maxima a riplurilor din banda de oprire.

Linia 17: Se declară variabila în care se stochează frecvența de tăiere normată.

Linia 23: Se declară variabila în care se stochează ordinul filtrului.

Linia 24: Se calculează coeficienții filtrului trece sus cu parametrii de proiectare prezentați pe prima pagină.

1.b) coeficienții a

	1	2	3	4	5
1	1	1.6114	1.8788	0.9846	0.3078

Figura 2: Coeficienții a nenormați calculați în Matlab

coeficienții b

	1	2	3	4	5
1	0.1570	0.0052	0.2539	0.0052	0.1570

Figura 3: Coeficienții b nenormați calculați în Matlab

```
26    % plot the frequency response and poles position
27    figure(1)
28    freqz(b,a), title('Frequency response of HP IIR filter')
29
30    figure(2)
31    zplane(b,a), title('Poles and zeroes position of the HP IIR filter')
32
33    % compute and owerwrite the b coeficients with
34    % the scaled ones by using L1 law
35    h  = impz(b, a);
36    k0 = sum(abs(h));
37    bScaled  = b / k0;
38
39    % compute the scaling coefficient 's1' for
40    % the input signal
41    h1 = impz(1, a);
42    k1 = sum(abs(h1));
43    s1 = 1 / k1;
```

Figura 4: Cod Matlab liniile 26-43

Linia 28: Se generează răspunsul în frecvență al filtrului digital

Linia 31: Se generează poziția polilor și a zerourilor pentru filtrul digital

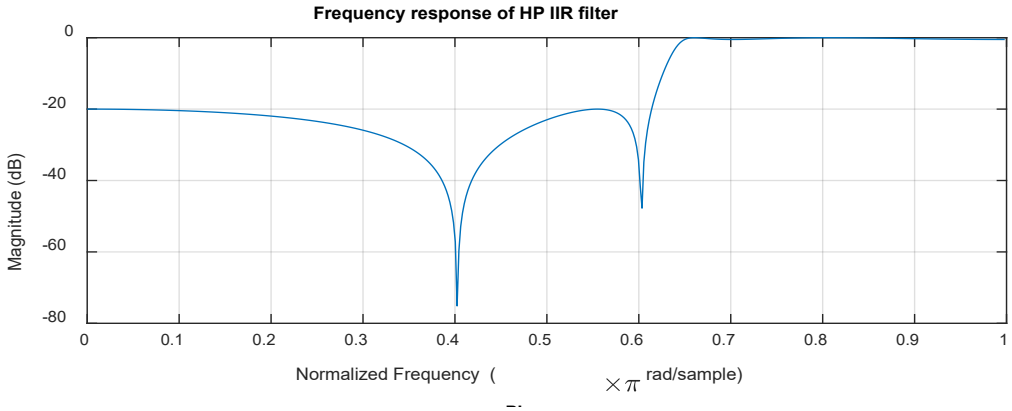


Figura 5: Răspuns în frecvență al filtrului digital trece sus IIR

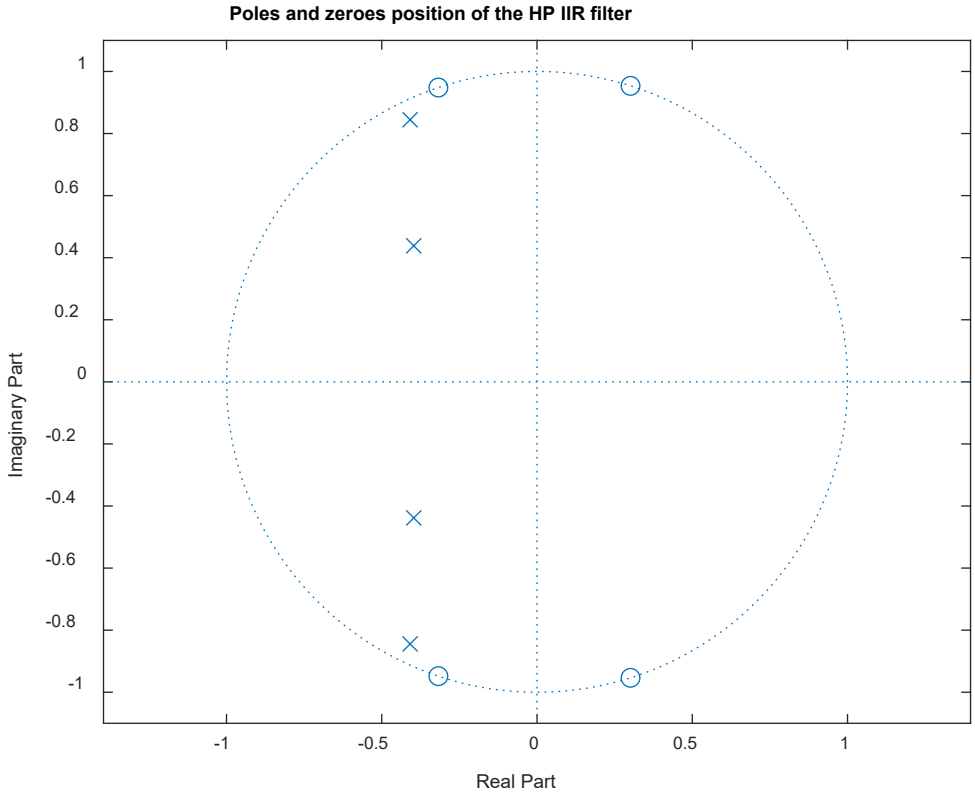


Figura 6: Poziția polilor și a zerourilor pentru filtrul trece sus IIR

1.c)

Linia 37: Se normează coeficienții b conform regulii de scalare L1

	1	2	3	4	5
1	0.0714	0.0024	0.1154	0.0024	0.0714

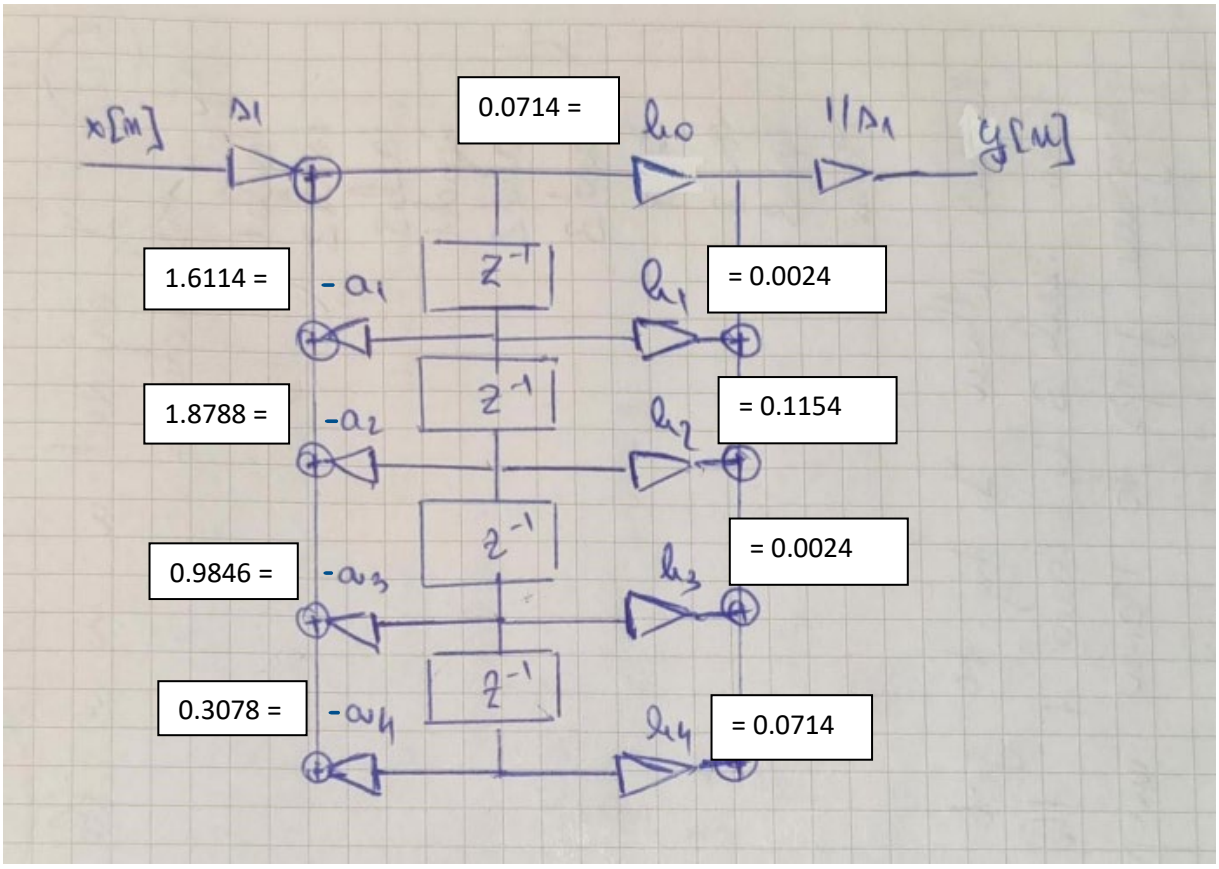
Figura 7: Coeficienții b normați conform regulii L1

Linia 41-43: Se determină coeficientul de scalare s1 pentru semnalul de intrare

	1
1	0.0513

Figura 8: Valoare coeficient de scalare s1 calculat in Matlab

1.d)



Unde b0, b1, b2, b3, b4 sunt scalați cu k0 si au valorile:

	1	2	3	4	5
1	0.0714	0.0024	0.1154	0.0024	0.0714

iar a1, a2, a3, a4 au valorile:

	1	2	3	4	5
1	1	1.6114	1.8788	0.9846	0.3078

s1 este egal cu:

	1
1	0.0513

2.a)

În următoarea parte, se va prezenta codul Matlab care generează semnalul de intrare pentru programul C.

```
45  %% Generate the input signal sinusoidal signal
46
47  t = 0 : 1/Fs : 1-1/Fs;              % 1 secs @ 8kHz sample rate
48
49  % generate a sinusoidal signal with linear increasing frequency
50  inputSignal = chirp(t,0,1,Fs/2);              % Start @ DC, cross 4kHz at t=1sec
51
52  % scale the input signal
53  inputSignalScaled = inputSignal / k1;
54
55  % write the values in the input file for the C program
56  fid = fopen('..\input.dat','w','b');
57  fwrite(fid,inputSignalScaled.*2^15,'int16');
58  fclose(fid);
59
60  % compute BlockLength for 160 samples / Block
61  BlockLength = fix(length(inputSignalScaled)/160);
62  L           = BlockLength * 160;
```

Figura 9: Cod Matlab liniile 45-62

Linia 50: se generează semnalul de intrare nescalat, folosind funcția Matlab chirp.

Linia 53: se generează semnalul de intrare scalat, folosind coeficientul de scalare al semnalului de intrare calculat mai sus.

Linia 56-58: Se scrie în fișierul input.dat semnalul de intrare scalat. Acest fișier va fi folosit de către programul C pentru a citi semnalul de intrare scalat pentru filtrul digital.

În continuare, se va prezenta programul C din CodeWarrior.

```
1   #include <prototype.h>
2   #include <stdio.h>
3   #define DataBlockSize 160   /* dimensiunea unui bloc de date */
4   #define BlockLength 50      /* numarul de blocuri de date */
5
6   Word16 x[DataBlockSize],y[DataBlockSize];
7   Word16 w  = 0;
8   Word16 w1 = 0;
9   Word16 w2 = 0;
10  Word16 w3 = 0;
11  Word16 w4 = 0;
12
13  Word16 b[]={WORD16(0.0714/2), WORD16(0.0024/2), WORD16(0.1154/2), WORD16(0.0024/2), WORD16(0.0714/2)};
14  Word16 a[]={WORD16(1.6114/2), WORD16(1.8788/2), WORD16(0.9846/2), WORD16(0.3078/2)};
15
16  Word32 sum;
17
18  int main()
19  {
20      /* Declare the local variables */
21      short n,i;
22      FILE *fpx,*fpy;
23
24      /* Open the file with the input signal from Matlab */
25      fpx=fopen("../input.dat","r+b");
26      if (!fpx)
27          printf("\nNu s-a deschis");
28
29      /* Open the file with the filtered signal computed
30         in this program */
31      fpy=fopen("../output.dat","w+b");
32      if (!fpy)
33          printf("\nNu s-a deschis");
34
```

Figura 10: Cod program C liniile 1 – 34

Liniile 3-4: Se definesc macrourile necesare procesării pe blocuri. Astfel se vor stoca în memoria programului C doar 160 de eșantioane pentru procesare, în loc de numărul total de 8000 eșantioane

Linia 6: Se declară vectorul în care se vor stoca eșantioanele semnalului de intrare

Liniile 7-11: Se declară variabilele de întârziere, în care se stochează eșantioanele intermediare, din structura filtrului

Linia 13: Se declară vectorul cu coeficienții b scalați ai filtrului digital

Linia 14: Se declară vectorul cu coeficienții a ai filtrului digital

Liniile 20-34: Se declară variabilele locale programului C și se deschide atât fișierul input.dat de la programul Matlab și fișierul output.dat, care se va folosi în programul Matlab pentru a citi semnalul de ieșire al programului C.

Liniile 35-61: Se implementează filtrul digital IIR în programul C CodeWarrior conform structurii formei directe 2

Linia 63: Se scriu în fișierul output.dat eșantioanele semnalului de ieșire

Liniile 67-68: Se închid fișierele input.dat și output.dat, deschise în cadrul programului C

```
35      /* Do the filtering */
36      for (i=0; i<BlockLength; i++)
37      {
38          fread(x,sizeof(Word16),DataBlockSize,fpx);
39
40          for (n=0; n<DataBlockSize; n++)
41          {
42              sum = L_deposit_h(shr(x[n], 1));
43              sum = L_msu(sum,a[0], w1);
44              sum = L_msu(sum,a[1], w2);
45              sum = L_msu(sum,a[2], w3);
46              sum = L_msu(sum,a[3], w4);
47
48              w = shl(round(sum), 1);
49
50              sum = L_mult(b[0],    w);
51              sum = L_mac(sum,b[1], w1);
52              sum = L_mac(sum,b[2], w2);
53              sum = L_mac(sum,b[3], w3);
54
55              y[n]= shl(mac_r(sum,b[4], w4), 1);
56
57              w4 = w3;
58              w3 = w2;
59              w2 = w1;
60              w1 = w;
61          }
62
63          fwrite(y,sizeof(Word16),DataBlockSize,fpy);
64      }
65
66      /* Close the files */
67      fclose(fpx);
68      fclose(fpy);
69
70      return(0);
71  }
72
```

Figura 11: Cod program C liniile 35 – 71

3. Rezultatele testării

Se testează în Matlab faptul că semnalul filtrat de ieșire al programului C coincide cu semnalul filtrat calculat în cadrul programului Matlab.

```
63  %% Generate the output signal in Matlab and
64  %  compare with the output signal from the C progam
65
66  fid            = fopen('..\output.dat','r','b');
67  outputSignalCW = fread(fid, L, 'int16');
68  fclose(fid);
69
70  % transform the output from CW in integer format
71  % and scale it
72  outputSignalCW= k1 * k0 * outputSignalCW'/(2^15);
73
74  % compute the output signal from Matlab
75  outputSignalM = k1 * filter(b,a,inputSignalScaled);
76
77  figure(3), plot(t,inputSignalScaled),               title('Unfiltered signal')
78  figure(4), plot(t,outputSignalM),              grid,title('Matlab output signal')
79  figure(5), plot(t,outputSignalCW),             grid,title('CW output signal')
80  figure(6), plot(t,outputSignalCW-outputSignalM),grid,title('Error')
81
```

Figura 12: Cod program Matlab liniile 63 – 81

Linia 67: Se citește semnalul de ieșire al programului C și se stochează în variabila outputSignalCW

Linia 72: Se rescalează semnalul de ieșire al programului CodeWarrior pentru a putea fi comparat cu cel calculat în Matlab

Linia 75: Se calculează semnalul de ieșire al filtrului digital IIR cu ajutorul Matlab.

Linia 77: Se afișează semnalul de intrare nefiltrat

Linia 78: Se afișează semnalul de ieșire calculat în Matlab

Linia 79: Se afișează semnalul de ieșire calculat în programul C

Linia 80: Se afișează diferența între semnalul de ieșire calculat în programul C și cel calculat în programul C

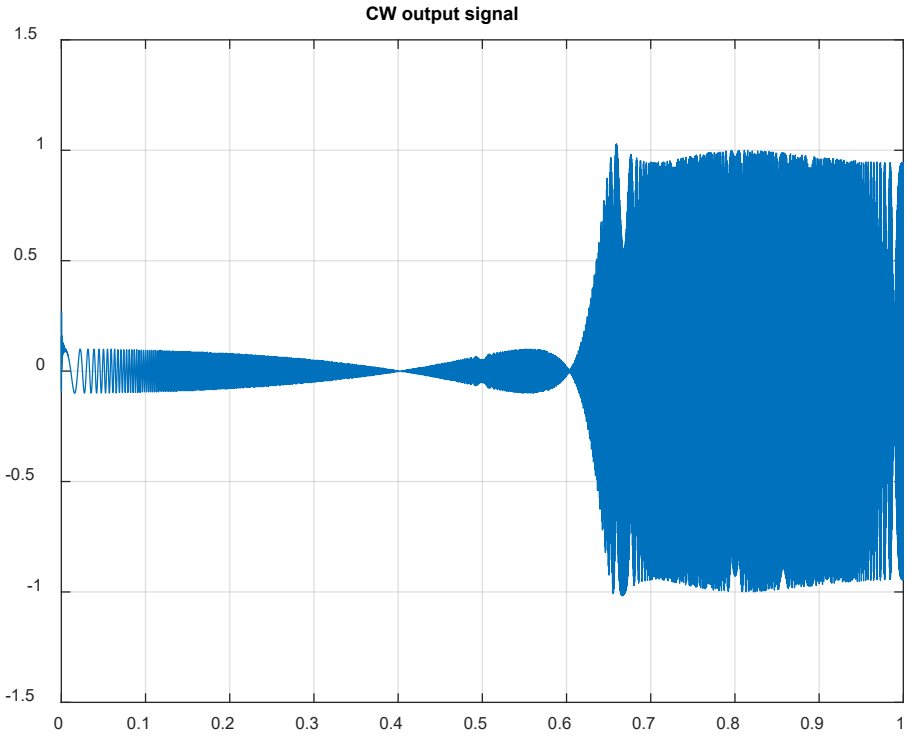


Figura 13: Semnalul de ieșire al programului C

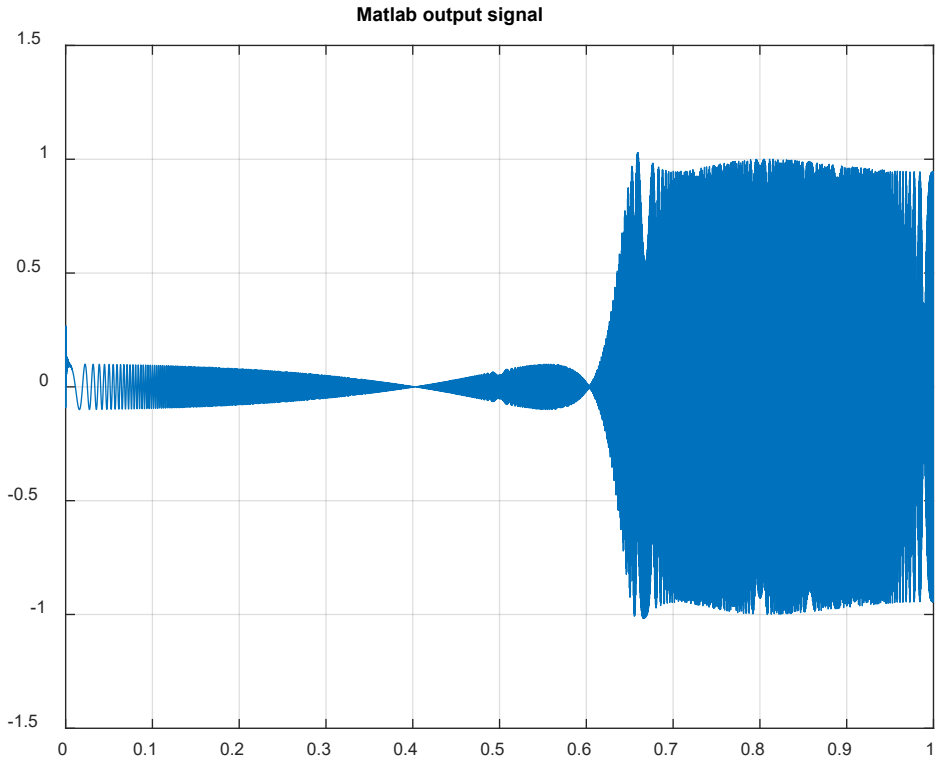


Figura 14: Semnalul de ieșire al programului Matlab

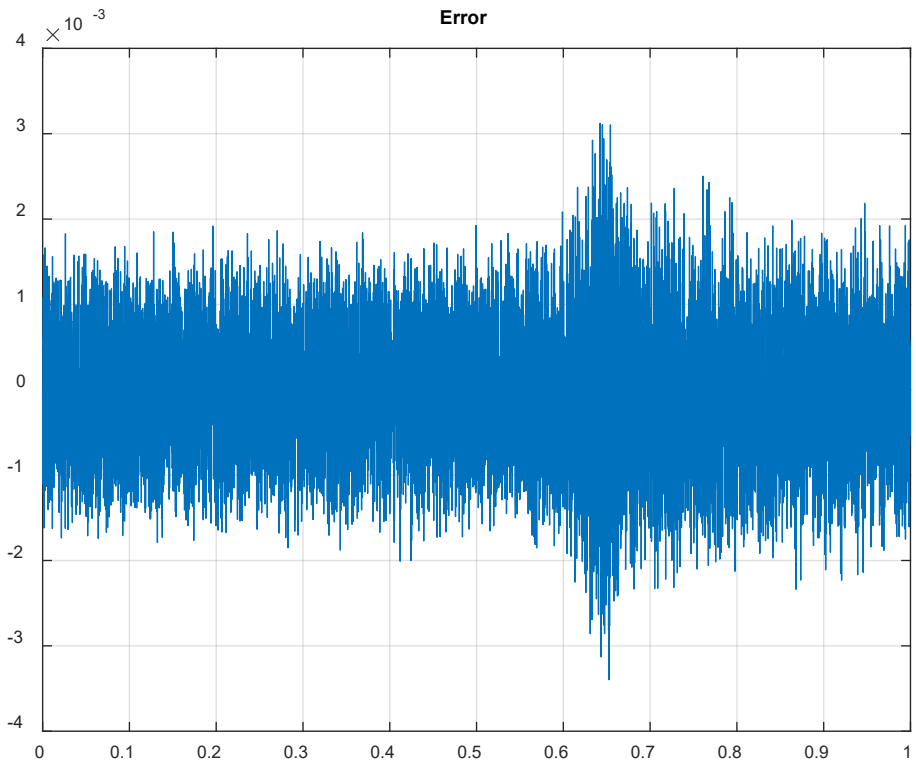


Figura 14: Semnalul de eroare

4. Explicații suplimentare

În cadrul programului Matlab se calculează și ordinul filtrului, cu toate că nu este impus de cerința temei. Acest lucru a fost realizat pentru a verifica plauzibilitatea realizării filtrului dat cu ordinul cerut în temă.