

Universitatea  
Transilvania  
din Brașov  
FACULTATEA DE INGINERIE ELECTRICĂ  
ȘI ȘTIINȚA CALCULATOARELOR

# Proiect PDS

*Ciocoiu Andrei-Ovidiu*

*Pandrea Vlad-Andrei*

## **filtre IIR**

Filtrele IIR (răspuns la impuls infinit) sunt în general alese pentru aplicațiile în care faza liniară nu este prea importantă și memoria este limitată. Au fost implementate pe scară largă în egalizarea audio, procesarea semnalului senzorilor biomedicali, senzori inteligenți IoT/IloT și aplicații de telecomunicații/RF de mare viteză.[1]

### **Avantaje**

Cost redus de implementare: necesită mai puțini coeficienți și memorie decât filtrele FIR pentru a satisface un set similar de specificații, adică frecvența de tăiere și atenuarea benzii de oprire.

Latență scăzută: potrivit pentru control în timp real și aplicații RF de foarte mare viteză datorită numărului redus de coeficienți.

Echivalent analogic: poate fi utilizat pentru a imita caracteristicile filtrelor analogice utilizând transformări de mapare în plan s-z.

### **Dezavantaje**

Caracteristicile de fază neliniară: Caracteristicile de fază ale unui filtru IIR sunt în general neliniare, în special în apropierea frecvențelor de tăiere. Filtrele de egalizare all-pass pot fi utilizate pentru a îmbunătăți caracteristicile fazei benzii de trecere.

Analiză mai detaliată: necesită mai multă scalare și analiză numerică de depășire atunci când este implementată în punct fix. Structura filtrului Direct Form II este deosebit de sensibilă la efectele cuantizării și necesită o atenție specială în timpul fazei de proiectare.[2]

Stabilitate numerică: mai puțin stabilă numeric decât omologii lor FIR (răspuns la impuls finit), datorită căilor de feedback.

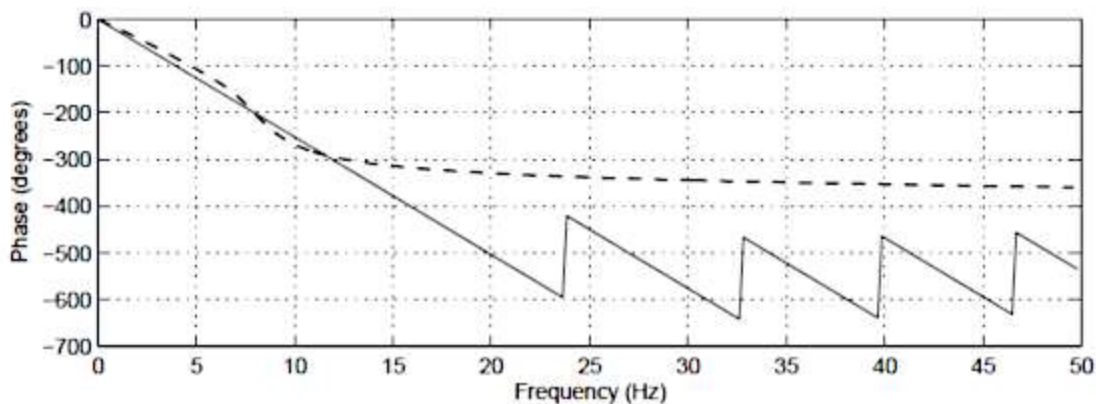


Fig.1 IIR: frequency response of a 14th order FIR (solid line), and a 4th order Chebyshev Type I IIR (dashed line)

Filtrele IIR sunt filtre recursive la care pentru calcularea valorii curente a semnalului de ieșire sunt folosite atât valorile semnalului de intrare cât și valorile vechi ale semnalului de ieșire. Astfel se poate realiza un răspuns aproape “infini” fără a utiliza foarte mulți coeficienți.[3]

### Formule matematice:

$$H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{1 + \sum_{k=1}^N a_k z^{-k}} = \frac{B(z)}{A(z)} \quad [4]$$

După cum se vede, funcția de transfer este o reprezentare în domeniul frecvenței a filtrului. Observați, de asemenea, că polii acționează asupra datelor de ieșire, iar zerourile asupra datelor de intrare. Deoarece polii acționează asupra datelor de ieșire și afectează stabilitatea, este esențial ca razele lor să rămână în interiorul cercului unitar (adică  $<1$ ) pentru stabilitatea BIBO (intrare limitată, ieșire mărginită). Razele zerourilor sunt mai puțin critice, deoarece nu afectează stabilitatea filtrului. Acesta este motivul principal pentru care filtrele FIR (răspuns la impuls finit) cu zero sunt întotdeauna stabile.

### Implementare proiect:

Am realizat proiectul aferent acestei materii într-un thin client, la care m-am conectat remote cu numele de utilizator și parola personală.



Am creat un folder cu numere iir care contine cele 2 versiuni de program si un fisier care ne ajuta la crearea unui executabil pentru a rula cele 2 programe.

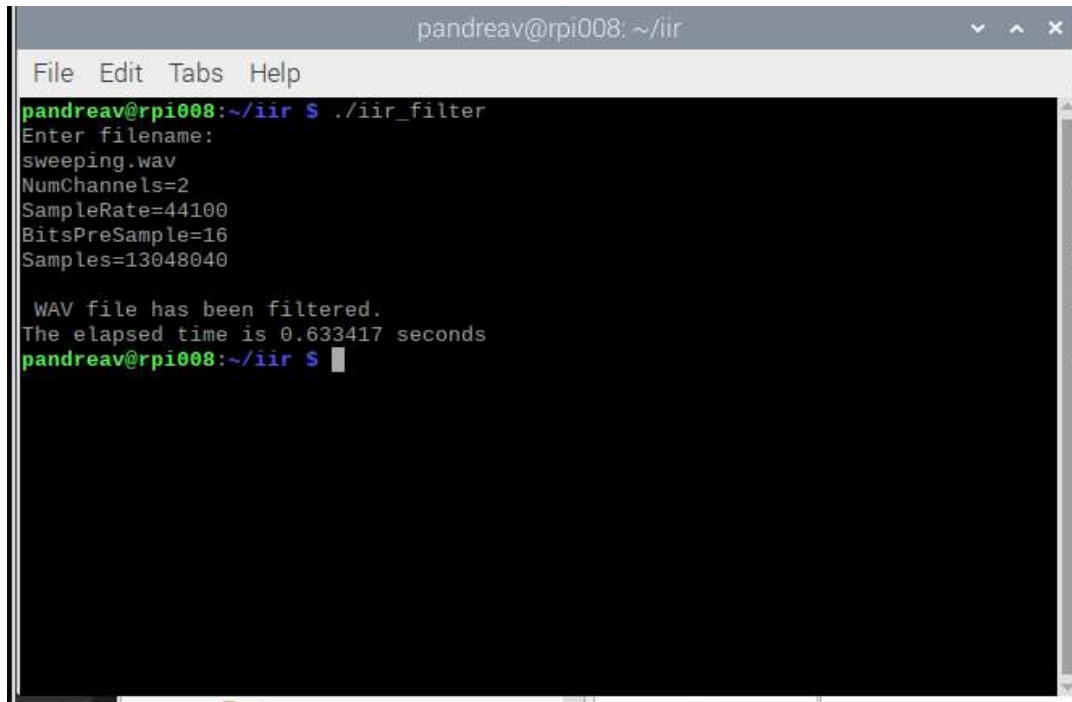
Pentru a rula programul, se introduce numele fisierului .wav care va fi filtrat de catre cod. La final se va forma un nou fisier .wav filtrat

```
pandreav@rpi008: ~/iir
File Edit Tabs Help
pandreav@rpi008:~/iir $ ./iir_filter
Enter filename:
sweeping.wav
NumChannels=2
SampleRate=44100
BitsPreSample=16
Samples=3262010

WAV file has been filtered.
The elapsed time is 0.340174 seconds
pandreav@rpi008:~/iir $
```

Fig.2 Executarea programulu

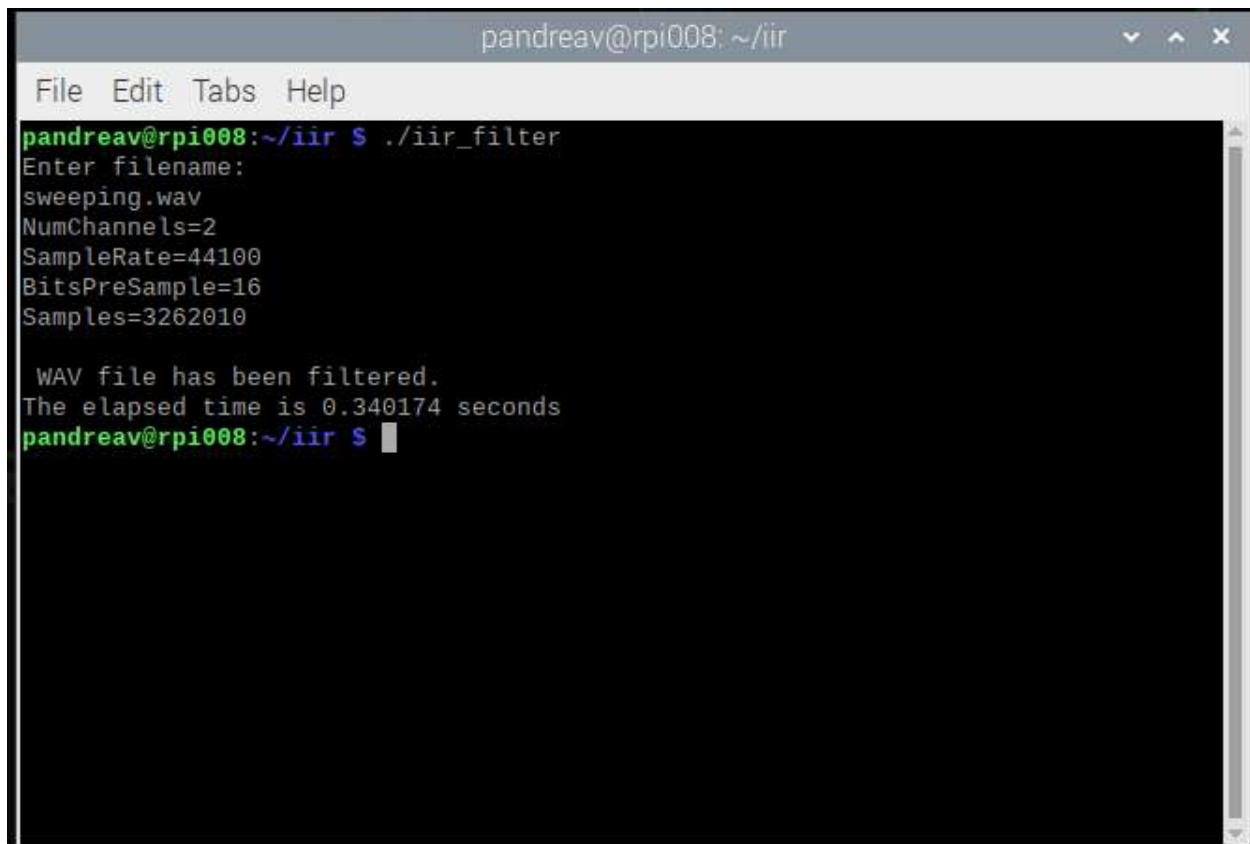
Compararea celor 2 coduri se poate vedea in figurile de mai jos:



```
pandreav@rpi008: ~/iir
File Edit Tabs Help
pandreav@rpi008:~/iir $ ./iir_filter
Enter filename:
sweeping.wav
NumChannels=2
SampleRate=44100
BitsPreSample=16
Samples=13048040

WAV file has been filtered.
The elapsed time is 0.633417 seconds
pandreav@rpi008:~/iir $
```

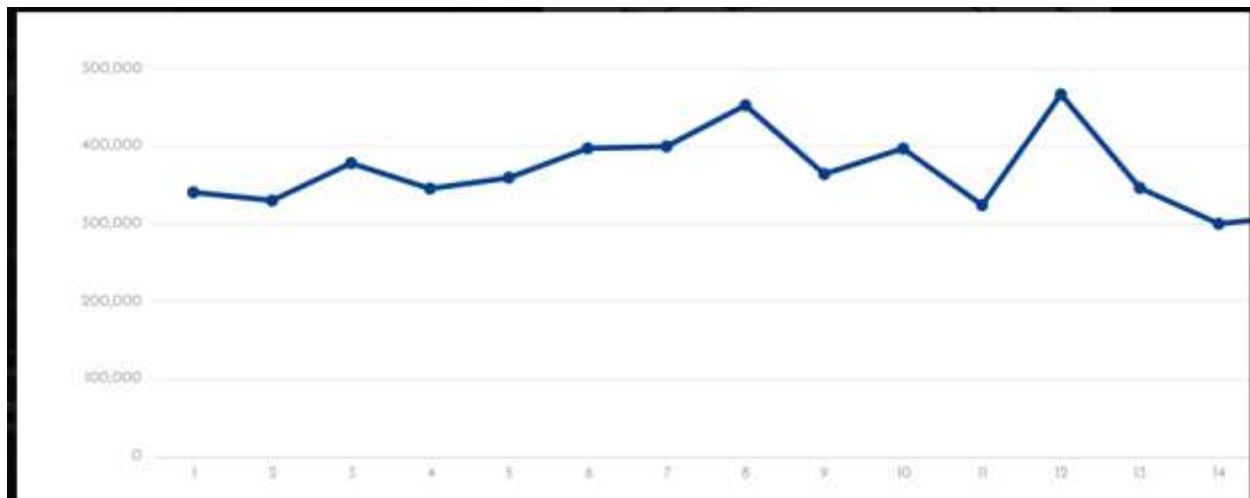
Fig.3 Primul cod



```
pandreav@rpi008: ~/iir
File Edit Tabs Help
pandreav@rpi008:~/iir $ ./iir_filter
Enter filename:
sweeping.wav
NumChannels=2
SampleRate=44100
BitsPreSample=16
Samples=3262010

WAV file has been filtered.
The elapsed time is 0.340174 seconds
pandreav@rpi008:~/iir $
```

Fig,4 Al doilea cod



*Fig.5 Diagrama timp executie*

Link github: <https://github.com/PANDREAVLAD/PDS/blob/main/pds.c>

#### Bibliografie:

- [1] <https://dspguru.com/dsp/faqs/iir/basics/>, disponibil la 18.05.2022
- [2] <https://www.micromodeler.com/articles/IntroductionToDSP/IIR.jsp> disponibil la 18.05.2022
- [3] <https://www.sciencedirect.com/topics/earth-and-planetary-sciences/iir-filter> disponibil la 18.05.2022
- [4] [http://telecom.etc.tuiasi.ro/pns/curs/cap3\\_iir\\_proiectarea%20filtrelor%20digitale.pdf](http://telecom.etc.tuiasi.ro/pns/curs/cap3_iir_proiectarea%20filtrelor%20digitale.pdf) disponibil la 18.05.2022