14 სიგნალის დამუშავება

მანქანური ხედვა მოითხოვს ციფრული გამონასახის სწრაფ დამუშავებასა და გაიგივებას. ეს პროცესი საჭიროებს არა მხოლოდ სწრაფ ალგორითმს, არამედ დახვეწილ და ფაქიზ ტექნოლოგიას, რომელიც უზრუნველყოფს დიდი მოცულობის გამონასახის სწრაფ ჩამოტვირთვას კომპიუტერის მეხსიერებაში. გამონასახის დამუშავების პროცესი მოიცავს რამდენიმე საფეხურს, რომელიც სასურველია განხორციელდეს ერთმანეთის პარალელურად. ამ პროცესში მნიშვნელოვანი ადგილი უკავია ციფრული ფილტრაციის ოპერაციას. ამ ამოცანას ემსახურება ციფრული მიკროპროცესორი, რომელიც წარმოადგენს ციფრულ ელექტრონულ კომპონენტს ნახევარგამტარულ ელექტრონულ სქმაზე დამონტაჟებული ტრანზისტორებით. კომპიუტერის ცენტრალური პროცესორი ერთი ან რამდენიმე მიკროპროცესორისაგან შედგება.

14.1 სიხშირული ანალიზი

ამ თავში განვიხილავთ MATLAB –ის ფუნქციებს, რომლებიც დაკავშირებულია სიგნალის დამუშავებასთან. განვიხილავთ ოთხი კატეგორიის ფუნქციებს: სიხშირული ანალიზი, ფილტრის ანალიზი, ფილტრის დიზაინი და ფილტრის მოდელი. განვსაზღვრავთ აღნიშვნებს, რომლებიც გამოიყენება სიგნალის დამუშავების თეორიაში. განვიხილავთ როგორც ანალოგური, ასევე ციფრული სიგნალის დამუშავების თეორიას. უფრო დაწვრილებით – ციფრულ სიგნალს.

ანალოგური სიგნალი არის უწყვეტი ფუნქცია(ჩვეულებრივ დროისა) რომელიც წარმოადგენს გარკვეულ ინფორმაციას, მაგალითად: ხმოვანი სიგნალი, სისხლის წნევის გამომხატველი სიგნალი, ანდა სეისმური სიგნალი. იმისათვის, რომ ეს ინფორმაცია დავამუშაოთ კომპიუტერის საშუალებით, ანალოგური სიგნალი უნდა აღირიცხოს ყოველ T წამში და ასე შეიქმნას ციფრული სიგნალი, რომელიც წარმოადგენს საწყისი ანალოგური სიგნალის რიცხვითი სიდიდეების მწკრივს.

$$f_k = f(kT)$$

ციფრული სიგნალი არის შერჩეულ სიდიდეთა მწკრივი f_k.

დრო, როცა დავიწყეთ ანათვლების აღება, ჩვეულებრივ ითვლება ნულის ტოლად და პირველი მნიშვნელობაც მწკრივში იქნება f_0.

თუ ანათვლებს ვიღებთ 100 ჰც სიხშირით (წამში 100 ანათვალი), ციფრული სიგნალის პირველი 3 მნიშვნელობა იქნება:

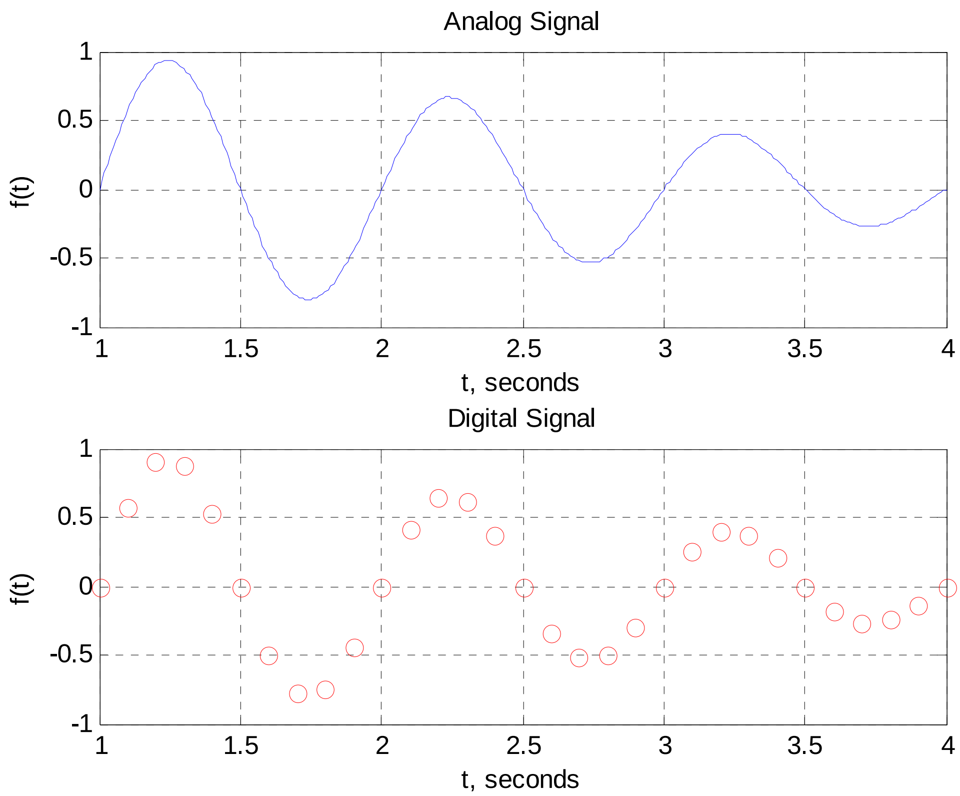
$$f_0 = f(0T) = f(0.0)$$
$$f_1 = f(1T) = f(0.01)$$
$$f_2 = f(2T) = f(0.02)$$

სიგნალის ანალიზი წარმოებს დროით და სიხშირულ გარემოში. სიხშირული სიგნალი შესაძლოა წარმოდგენილი იყოს კომპლექსური რიცხვების სახით, როგორც ჯამი სინუსოიდებისა (კოსინუსური ფორმით), რომელთაგანაც შედგება სიგნალი (ჰარმონიკების სახით). ზოგიერთი ინფორმაცია უკეთ ჩანს სიგნალის დროითი ანალიზიდან, მაგალითად თუ შევხედავთ დროით მრუდს, შეგვიძლია ვთქვათ პერიოდულია თუ არა სიგნალი. დროითი მონაცემებით გამოვითვლით სიგნალის საშუალოს, სტანდარტულ გადახრას, გადახრას და სიმძლავრეს. სიგნალის სიხშირული მახასიათებლების შესაფასებლად კი გვჭირდება მისი სიხშირული ანალიზი. სიგნალის სიხშირულ შემცველობას მის სპექტრს უწოდებენ.

დროითი სიგნალის გადასაყვანად სიხშირულ გარემოში გამოიყენება ფურიეს დისკრეტული გარდაქმნა. საწყისი მონაცემები ფურიეს გარდაქმნისათვის არის დროითი მონაცემების მწკრივი f_k. ალგორითმი გამოითვლის კომპლექსურ სიდიდეთა მწკრივს F_k რომელიც წარმოადგენს სიხშირულ ინფორმაციას. **DFT** ალგორითმი მოითხოვს საკმაო დროს, როცა მონაცემთა მწკრივი გრძელია, მაგრამ თუ მონაცემთა რაოდენობა 2-ის ხარისხის ტოლია ($N = 2^m$), გამოიყენება განსხვავებული ალგორითმი - ფურიეს სწრაფი გარდაქმნა (**fft**), რომელიც საგრძნობლად ამცირებს გამოთვლით დროს.

რადგან ციფრული სიგნალი აითვლება ყოველ T წამში, წამში გვექნება 1/T ანათვალი ანუ ათვლის სიხშირე იქნება 1/T ჰერცი. ათვლის სიხშირე სიფრთხილით უნდა შეირჩეს, რათა თავიდან ავიცილოთ ზედდების ეფექტი (aliasing) – პრობლემა, რომელიც გამოწვეულია იმით, რომ შერჩევა არ ხდება შესაბამისი სიხშირით. შეიძლება ვაჩვენოთ, რომ ამის თავიდან ასაცილებლად სიგნალის ათვლის სიხშირე უნდა აღემატებოდეს სიგნალის შემადგენელი

ნებისმიერი სინუსოიდის სიხშირის გაორმაგებულ მნიშვნელობას. მაგალითად თუ სიგნალი შეიცავს ორ სინუსოიდას, ერთის სიხშირეა 10 ჰც, მეორის კი 35 ჰც, სიგნალის ათვლის სიხშირე უნდა იყოს 70 ჰც. **ნაიქვისტის სიხშირე** ტოლია ათვლის სიხშირე/2 და წარმოადგენს ზედა საზღვარს იმ სიხშირეებისას, რომელსაც შეიძლება შეიცავდეს ციფრული სიგნალი.



ნახ. 14.1 ანალოგური და ციფრული სიგნალი

MATLAB –ის ფუნქცია ***fft*** გამოითვლის სიგნალის სიხშირულ სპექტრს. მას შეიძლება ჰქონდეს ერთი ან ორი არგუმენტი. თუ მხოლოდ ერთი არგუმენტი აქვს, ეს უნდა იყოს ვექტორი, რომლის ელემენტებია სიგნალის ამპლიტუდები დროის განსაზღვრულ ინტერვალში. შედეგად უნდა მივიღოთ იგივე სიგრძის ვექტორი, რომელიც წარმოადგენს კომპლექსურ რიცხვთა მწკრივს და გვაძლევს საწყისი სიგნალის სიხშირულ სპექტრს.

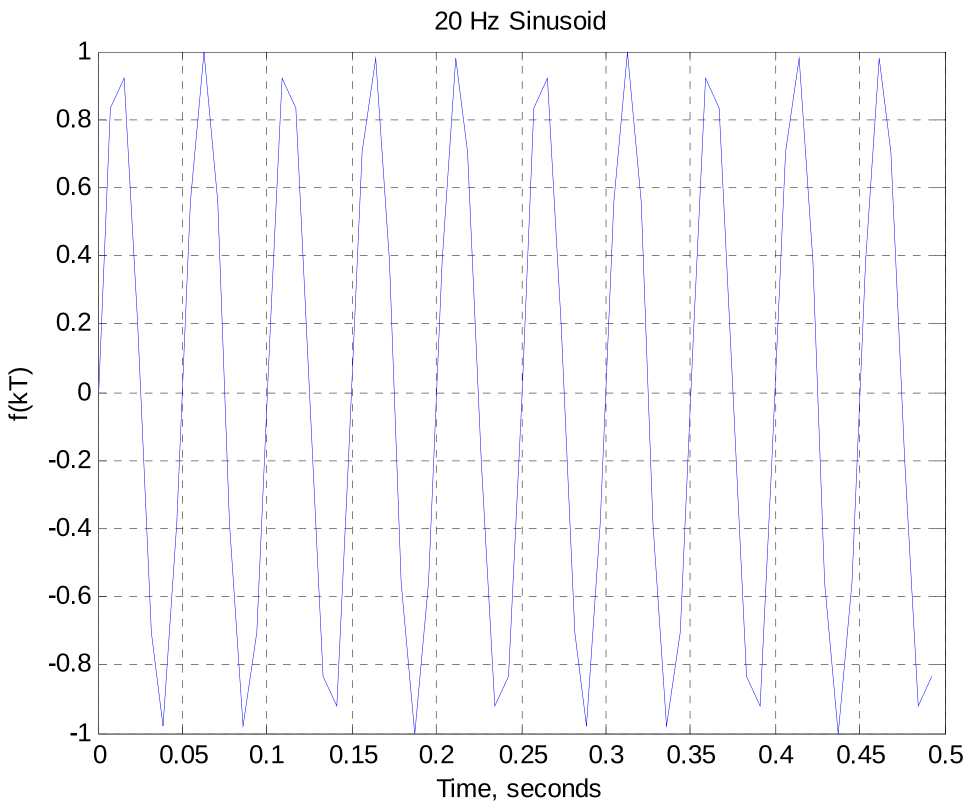
fft შედეგად მიღებული ვექტორის ურთიერთმომდევნო ელემენტები ერთმანეთისგან განსხვავდებიან 1/NT ჰერცით. ე. ი. თუ გვაქვს დროითი სიგნალის 32 ანათვალი, რომელიც ათვლილი იყო 1000 ჰც სიხშირით, ***fft*** ალგორითმით დათვლილი მნიშვნელობები შეესაბამება 0ჰც, 1/0.032ჰც, 2/0.032ჰც და ა.შ. ანუ 0ჰც, 31.25ჰც, 62.5ჰც ... ნაიქვისტის სიხშირე = 1/2T და ეს შეესაბამება F_{16} . რადგან ფურიეს დისკრეტული ფუნქცია პერიოდულია, ნაიქვისტის სიხშირის ზევით მონაცემები ახალ ინფორმაციას არ შეიცავს და ვსარგებლობთ შედეგად მიღებული ვექტორის ელემენტთა მხოლოდ პირველი ნახევრით.

განვიხილოთ MATLAB-ის ბრძანებათა შემდეგი წყება:

```
>>N=64;
>>T=1/128;
>>k=0:N-1;
```

```
>>f=sin(2*pi*20*k*T);
>>plot(k*T,f)
>> grid
>> xlabel('Time, seconds')
>> ylabel('f(kT)')
>> title('20 Hz Sinusoid')
```

f წარმოადგენს 20 ჰერციან სინუსოიდას რომელიც ათვლილია ყოველ 1/128 წამში, რასაც შეესაბამება ათვლის სიხშირე 128 ჰერცი.(სინუსოიდის სიხშირე 20ჰც-ია, ამიტომ ის უნდა ავითვალოთ უფრო დიდი სიხშირით, ვიდრე 40ჰც. 128 ჰერცი ამ შემთხვევაში საკმარისია.)

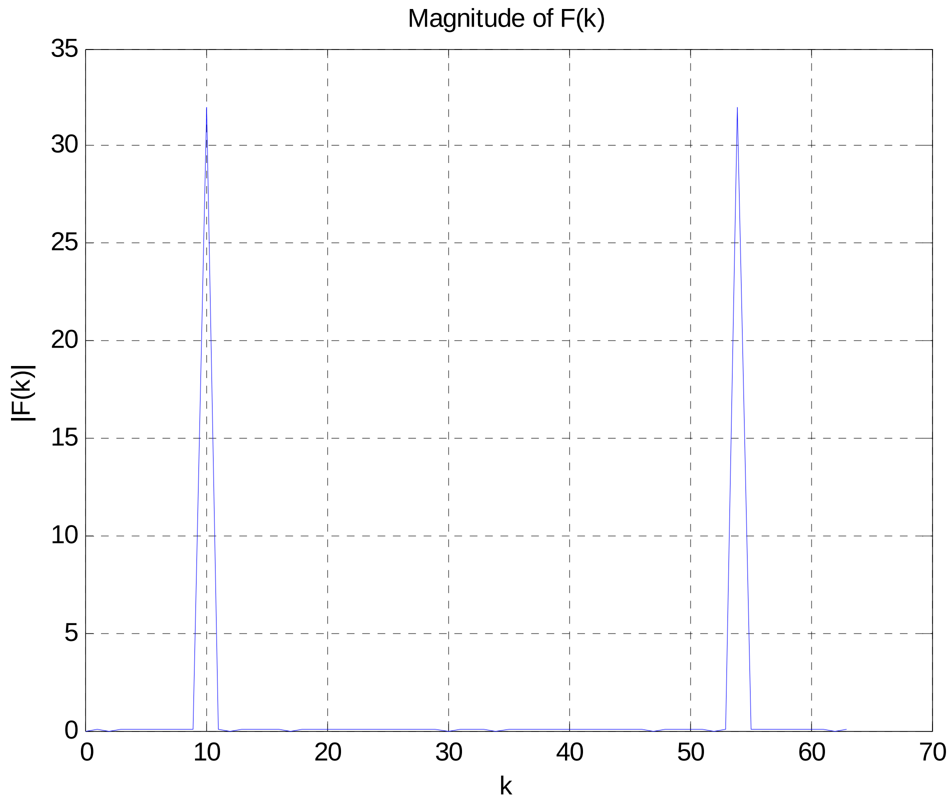


ნახ. 14.2 20 ჰერციანი სინუსოიდა

რადგან ვიცით, რომ სინუსოიდა 20 ჰერციანია, მოსალოდნელია, რომ სიხშირის მნიშვნელობა = 0 ყველგან, გარდა იმ წერტილისა, რომელიც შეესაბამება 20 ჰც. რომ განვსაზღვროთ F_k, რომელიც შეესაბამება 20 ჰც, უნდა გამოვთვალოთ სხვაობა სიხშირულ წერტილებს შორის ჰერცებში, ეს არის 1/NT = 2ჰც.

ამრიგად, 20 ჰერციანი კომპონენტი უნდა გამოჩნდეს F_{10} მნიშვნელობაზე. ნახ. 14.3.

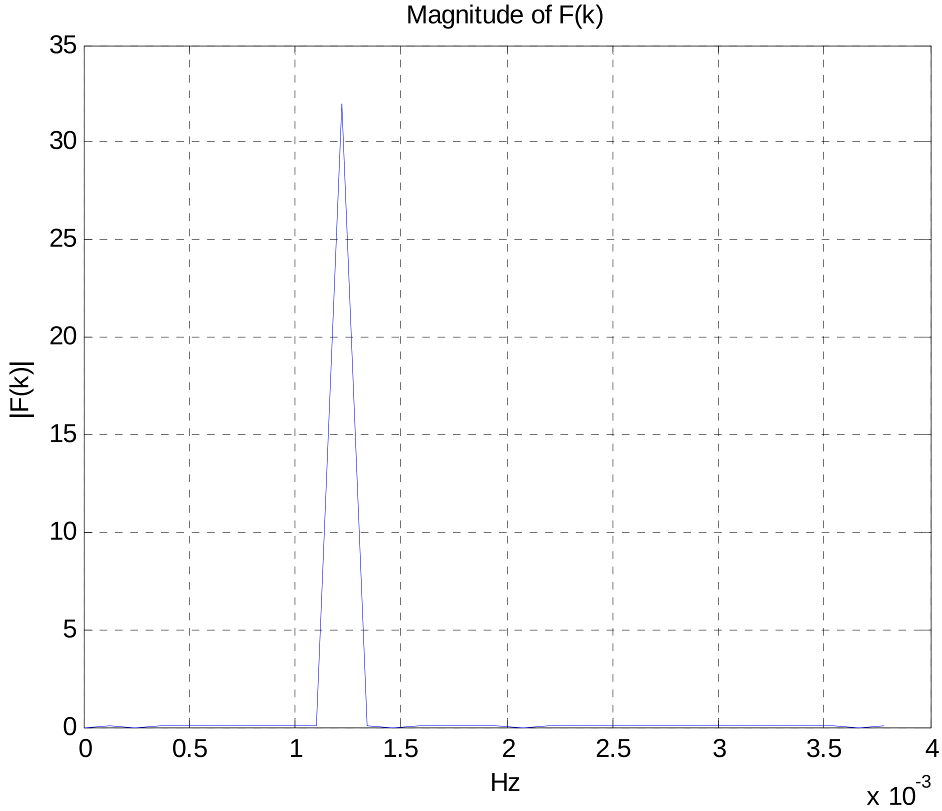
```
>> F=fft(f);
>> magF=abs(F);
>> plot(k, magF),title('Magnitude of F(k)'),...
xlabel('k'), ylabel('|F(k)|'),grid
```



ნახ. 14.3 F_k მოდული

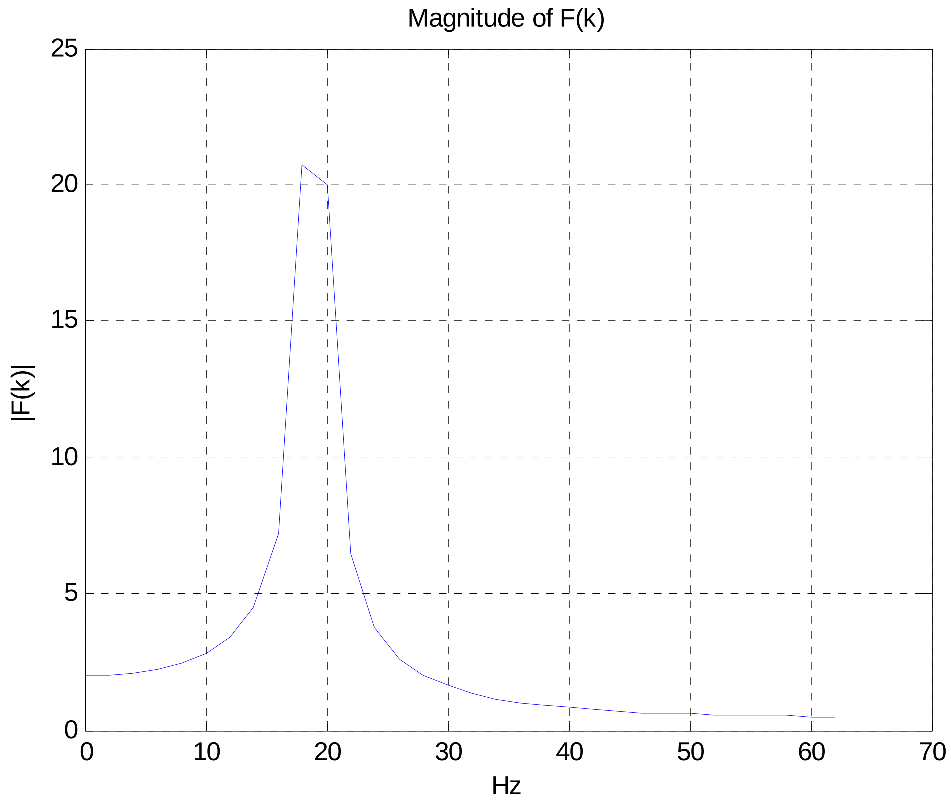
ამ ნახაზზე ჩანს სიმეტრიული ანასახი 20 ჰც კომპონენტისა, რაც გამოწვეულია DFT პერიოდულობით. 20 ჰც კომპონენტი თავს იჩენს როგორც F(54). უნდა აღვნიშნოთ რომ ვექტორი k შეიცავს ინდექსებს, რომელიც შეესაბამება f, F ვექტორთა ელემენტებს. საზოგადოდ რეკომენდებულია აიგოს მონაცემების მხოლოდ პირველი ნახევარი, რომ თავიდან ავიცილოთ სიმეტრიული კომპონენტი. უფრო მოხერხებულია ავაგოთ F_k ჰერცებში გამოსახული სიდიდეებით ნაცვლად k ინდექსებისა.

```
>> hertz = k*(1/N*T);
>> plot(hertz(1:N/2), magF(1:N/2)),...
title('Magnitude of F(k)'),...
>> xlabel('Hz'),ylabel('|F(k)|'),grid
```



ნახ. 14.4 F_k ჰერცებში

დავუშვათ ამ მაგალითში განხილული სინუსოიდის სიხშირე 19 ჰერცია. რადგან სხვაობა ვექტორის ორ მომდევნო მნიშვნელობას შორის 2 ჰერცია, ეს სიხშირე თავს იჩენს k=9.5-ზე, მაგრამ k მნიშვნელობები მთელი რიცხვებია, ამიტომ არ გვაქვს $F_{9.5}$, ამ შემთხვევაში სინუსოიდის სიხშირული კომპონენტი გამოჩნდება F_9 და F_{10} შორის.



ნახ. 14.5 19 ჰერციანი სინუსოიდის სიხშირული მრუდი

ნახ. 14.4 და ნახ. 14.5 ორივე წარმოადგენს ერთი სინუსოიდის სიხშირულ სპექტრს, მაგრამ ერთი მათგანი ხვდება ზუსტად FFT ალგორითმით მიღებულ წერტილზე, მეორე კი არა. ეს არის მაგალითი ე.წ. გაჟონვისა (leakage).

ifft წარმოადგენს ფურიეს შებრუნებული გარდაქმნის ალგორითმს. მისი საშუალებით გამოითვლება დროითი სიგნალი შესაბამისი კომპლექსური სიხშირული მნიშვნელობების მიხედვით. მოცემული ჩანაწერი გამოითვლის F კომპლექსურ მნიშვნელობებს და შემდეგ **ifft** საშუალებით აღადგენს საწყის დროით სიგნალს. საბოლოოდ გამოითვლება სხვაობა სიგნალის საწყის და **ifft**-ს საშუალებით აღდგენილ მნიშვნელობათა შორის, ამ შემთხვევაში იგი ნულის ტოლია.

```
>> N=64;
>> T=1/128;
>> k=0:N-1;
>> f=sin(2*pi*19*k*T);
>> sum(f-ifft(fft(f)))

ans =

 -3.3307e-016
```

FFT ალგორითმი მძლავრი საშუალებაა სიგნალის დამუშავებასა და ანალიზისათვის. ჩვენ განვიხილეთ F_k მოდული, მაგრამ არანაკლებ მნიშვნელოვან ინფორმაციას იძლევა F_k ფაზის გამოთვლა **phase(F_k)**.

სავარჯიშო

გამოთვალეთ და ააგეთ შემდეგი სიგნალის 128 წერტილის მნიშვნელობა. ააგეთ დროითი სიგნალი. შემდეგ FFT ალგორითმის საშუალებით იპოვეთ სიგნალის სიხშირული მნიშვნელობები და ააგეთ შესაბამისი მრუდი მხოლოდ პირველი 64 წერტილის მიხედვით. ათვლის სიხშირედ აიღეთ 1 კილოჰერცი. შეამოწმეთ მართლაც იმ სიხშირეებზე მიიღეთ თუ არა პიკი, სადაც მოსალოდნელი იყო.

1. $f_k = 2\sin(2\pi 50kT)$
2. $f_k = \cos(250\pi kT) - \sin(200\pi kT)$
3. $f_k = 5 - \cos(1000kT)$
4. $f_k = 4\sin(250\pi kT - \pi/4)$

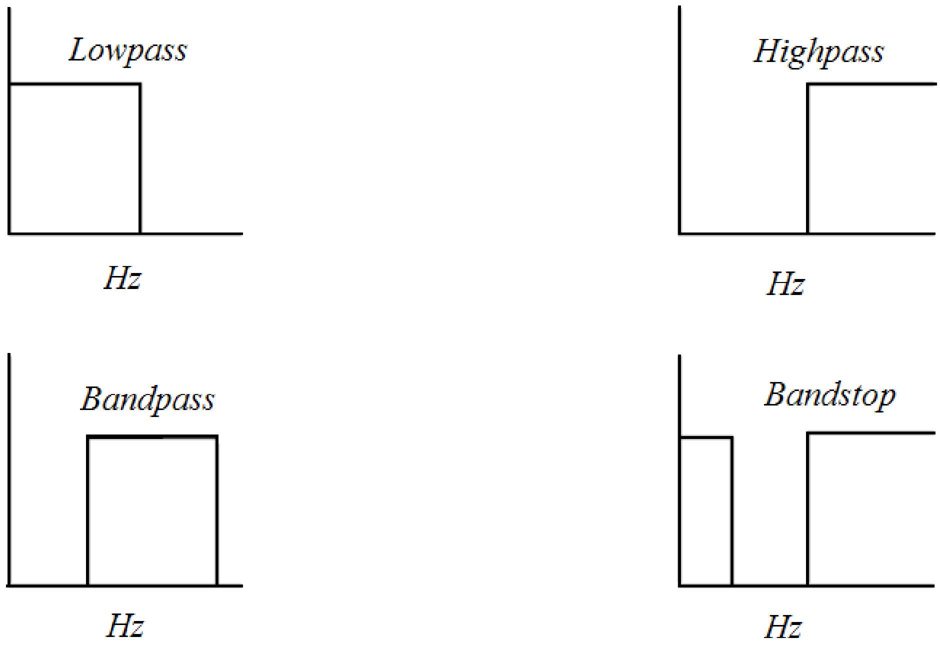
14.2 ფილტრის ანალიზი

ანალოგური სისტემის გადაცემის(transfer function) ფუნქცია წარმოადგენს კომპლექსურ ფუნქციას *H(s)*, ხოლო ციფრული სისტემის გადაცემის ფუნქციაა – *H(z)*, ეს ფუნქცია ასახავს სისტემის გავლენას შემოსულ სიგნალზე, ანუ სისტემის ფილტრულ ეფექტს. ორივე ფუნქცია სიხშირის უწყვეტი ფუნქციაა, სადაც $s = j\omega$, $z=e^{j\omega T}$ (გავიხსენოთ, რომ ω არის სიხშირე, გამოსახული რადიანი/წამში.) ამრიგად მოცემულია ω_0, ვუშვებთ, რომ გადაცემის ფუნქციის ამპლიტუდა არის K და ფაზა φ. თუ შემავალი სიგნალი შეიცავს სინუსოიდას ω_0 სიხშირით, გამომავალი სიგნალისათვის ამ სინუსოიდის ამპლიტუდა გამრავლდება K-ზე ფაზა კი გაიზრდება φ-ით.

$$A\sin(\omega_0 t+\theta) \rightarrow \boxed{} \rightarrow A * k\sin(\omega_0 t+\theta+\varphi)$$

$$A\sin(\omega_0 kT+\theta) \rightarrow \boxed{} \rightarrow A * k\sin(\omega_0 kT+\theta+\varphi)$$

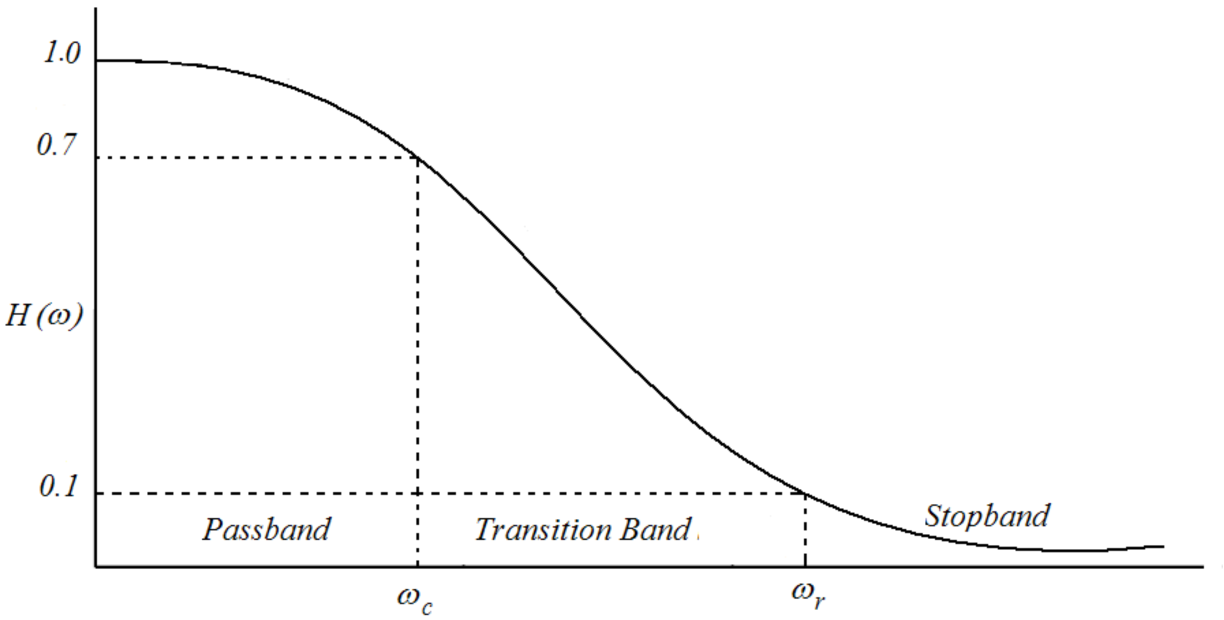
ნახ. 14.6 ფილტრის გავლენა სინუსოიდაზე



ნახ. 14.7 გადაცემის ფუნქცია (იდეალური შემთხვევა)

რადგანაც ფილტრის გადაცემის ფუნქცია განსაზღვრავს ფილტრის ეფექტს სიხშირეზე, იგი უნდა ხასიათდებოდეს სიხშირეთა ზოლით, რომელსაც ის ატარებს. მაგალითად დაბალსიხშირული ფილტრი გაატარებს სიხშირეებს ზღვრულ სიხშირის ქვევით და მოკვეთს დანარჩენს. ზოლოვანი ფილტრი გაატარებს სპექტრის მხოლოდ მოცემულ მონაკვეთს დანარჩენს კი მოკვეთს. bandstop ფილტრი არ გაატარებს სიხშირული სპექტრის მხოლოდ განსაზღვრულ მონაკვეთს. ნახ. ნახ. 14.7-ზე ნაჩვენებია ოთხი ძირითადი ფილტრის გარდაქმნის ფუნქცია იდეალურ შემთხვევაში. პრაქტიკაში შეუძლებელია ასეთი მახასიათებლების მქონე ფილტრის განხორციელება.

მაგალითია ტიპიური დაბალსიხშირული ფილტრისა. გვაქვს 3 უბანი – გატარების, გარდამავალი და მოკვეთის. ეს უბნები განისაზღვრება მოკვეთის ω_c და ω_r გარატების სიხშირეებით. მიღებულია,რომ სიხშირე, რომელიც შეესაბამება ამპლიტუდას 0.7, ჩაითვალოს ზღვრულ(cutoff) სიხშირედ, ხოლო 0.1 ჩამკეტ(rejection) სიხშირედ. ასეთი აღნიშვნის შემოღების შემდეგ, შეგვიძლია ვთქვათ, რომ გატარების ზოლი შეიცავს სიხშირეებს, რომელთა მოდული მეტია ზღვრულ სიხშირეზე, გარდამავალი ზოლი შეიცავს სიხშირეებს, რომელთა მოდული ნაკლებია ზღვრულ (მოკვეთის) და მეტია ჩამკეტ სიხშირეზე, stopband შეიცავს ჩამკეტის მოდულზე ნაკლები მოდულის შესაბამის სიხშირეებს.



ნახ. 14.8 ტიპიური დაბალსიხშირული ფილტრი

რადგან გადაცემის ფუნქცია კომპლექსურია, შესაბამისი ფილტრის ანალიზი ხშირად საჭიროებს მოდულის და ფაზის გრაფიკულ წარმოდგენას. MATLAB-ის ფუნქცია **abs, angle** და **unwrap** შეგვიძლია გამოვიყენოთ კომპლექსური სიდიდეების *H(s), H(z)* მოდულისა და ფაზის გამოსათვლელად. ამასთანავე, ფუნქციები **freqs** და **freqz** გამოიყენება თავად *H(s), H(z)* მნიშვნელობათა გამოსათვლელად.

ანლოგური სისტემის გადაცემის ფუნქცია

ანალოგური ფილტრი განისაზღვრება ფუნქციით $H(s)$ სადაც $s = j\omega$. გადაცემის ფუნქციის ზოგადი სახეა:

$$H(s) = \frac{B(s)}{A(s)} = \frac{b_0 s^n + b_1 s^{n-1} + b_2 s^{n-2} + \cdots + b_n}{a_0 s^n + a_1 s^{n-1} + a_2 s^{n-2} + \cdots + a_n}$$

ეს ფუნქცია შეესაბამება n რიგის ანალოგურ ფილტრს. მისი ზოგიერთი მაგალითია:

$$H(s) = \frac{0.5279}{s^2 + 1.0275s + 0.5279}$$

$$H(s) = \frac{bs^2}{s^2 + 0.1117s + 0.0062}$$

$$H(s) = \frac{1.05s}{s^2 + 1.05s + 0.447}$$

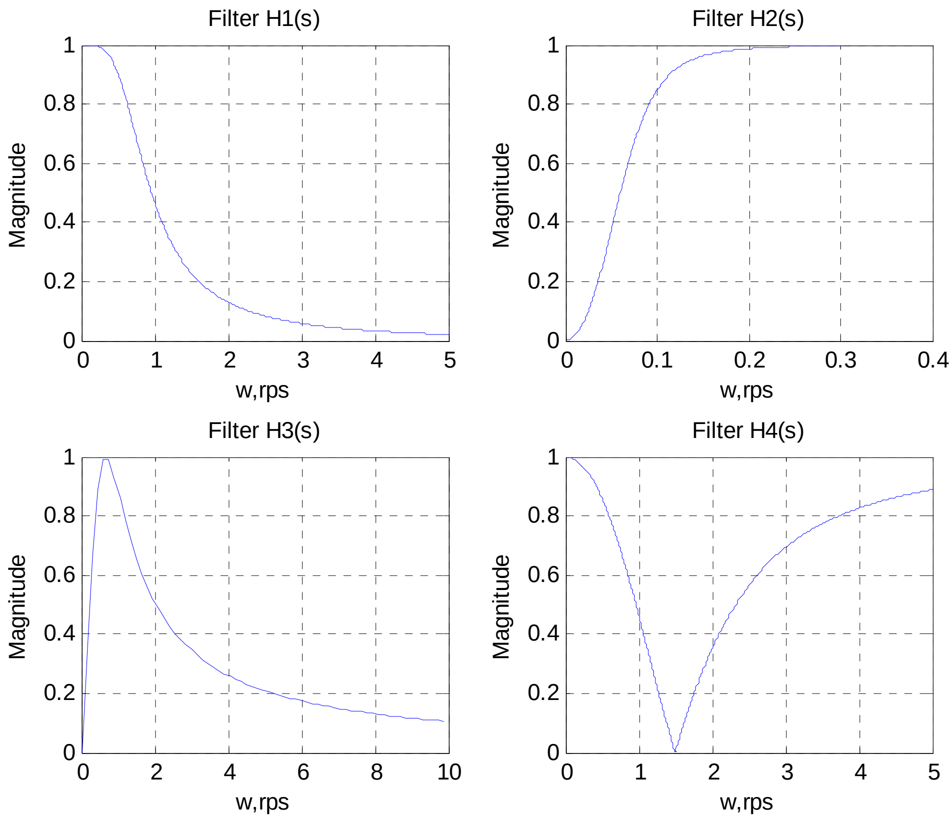
$$H(s) = \frac{s^2 + 2.2359}{s^2 + 2.3511s + 2.2359}$$

იმისათვის, რომ განვსაზღვროთ გადაცემის აქ წარმოდგენილი ფუნქციის მქონე სისტემის მახასიათებლები, ავაგოთ მათი მოდულისა და ფაზის მრუდები. **MATLAB** –ის ფუნქცია

freqs გამოითვლის კომპლექსური ფუნქციის მნიშვნელობებს სამი არგუმენტის საშუალებით. პირველი არგუმენტია ვექტორი, რომელიც შეიცავს $B(s)$ პოლინომის კოეფიციენტებს, მეორე - ვექტორი, რომელიც შეიცავს $A(s)$ პოლინომის კოეფიციენტებს, ხოლო მესამე - ვექტორი სიხშირეების მნიშვნელობებით რადიანი/წამში. კოეფიციენტების ვექტორები პირდაპირ გადაცემის ფუნქციიდან უნდა ავიღოთ, გარკვეული გამოცდილებაა საჭირო სიხშირეთა ინტერვალის სათანადოდ შესარჩევად. საზოგადოდ, გვჭირდება სიხშირეთა არე, რომელიც იწყება 0 დან და მოიცავს ყველა კრიტიკულ ინფორმაციას ფილტრში. უნდა შევძლოთ განვსაზღვროთ ფილტრის ტიპი (დაბალსიხშირული, მაღალსიხშირული, ზოლოვანი, ჩამკეტი stopband) და კრიტიკული სიხშირეები ზღვრული და ჩამკეტი).

ქვემოთ მოყვანილი პროგრამა განსაზღვრავს და აგებს გადაცემის 4 ფუნქციის მახასიათებელ მრუდებს

```
%     This programm datermines and plots the
%     magnitudes of four analog filters
%
w1 = 0:0.05:5;
B1 = [0.5279];
A1 = [1,1.0275,0.5279];
H1s = freqs(B1,A1,w1);
%
w2 = 0:0.001:0.3;
B2 = [1,0,0];
A2 = [1,0.1117,0.0062];
H2s = freqs(B2,A2,w2);
%
w3 = 0:0.15:10;
B3 = [1.05,0];
A3 = [1,1.05,0.447];
H3s = freqs(B3,A3,w3);
%
w4 = 0:0.005:5;
B4 = [1,0,2.2359];
A4 = [1,2.3511,2.2359];
H4s = freqs(B4,A4,w4);
clf
subplot(221),plot(w1,abs(H1s)),title('Filter H1(s)'),...
    xlabel('w,rps'),ylabel('Magnitude'),grid
subplot(222),plot(w2,abs(H2s)),title('Filter H2(s)'),...
    xlabel('w,rps'),ylabel('Magnitude'),grid
subplot(223),plot(w3,abs(H3s)),title('Filter H3(s)'),...
    xlabel('w,rps'),ylabel('Magnitude'),grid
subplot(224),plot(w4,abs(H4s)),title('Filter H4(s)'),...
    xlabel('w,rps'),ylabel('Magnitude'),grid
```



ნახ. 14.9 სხვადასხვა ტიპის ანალოგური ფილტრების მახასიათებელი მრუდები

ფილტრის ფაზები შეგვიძლია ავაგოთ **angle** ან **unwrap** ფუნქციის გამოყენებით. **angle** გვაძლევს კუთხეს რადიანებში მხოლოდ [-π,π] ინტერვალში. ამ ხარვეზის თავიდან ასაცილებლად ვსარგებლობთ ფუნქციით **unwrap** ასეთი სახით unwrap(angle(w)).

14.3 დისკრეტული გადამცემი ფუნქცია

დისკრეტული ფილტრი განისაზღვრება გადაცემის კომპლექსური ფუნქციით *H(z),* სადაც $z=e^{j\omega T}$. z შეიძლება ჩაიწეროს როგორც სიხშირის (ω) ან ნორმირებული სიხშირის (ωT) ფუნქცია, თუ z სიხშირის ფუნქციაა, *H(z)* აგრეთვე სიხშირის ფუნქცია იქნება. თუკი *H(z)* მოვარგეთ (applied) სიგნალს ათვლის ინტერვალით T, სიხშირეების შესაბამისი არე იქნება 0 დან ნაიქვისტის სიხშირემდე, რომელიც ტოლია π/T რადიანი/წმ, ან 1/2T ჰერცი. თუ დავუშვებთ, რომ z არის ნორმირებული სიხშირის ფუნქცია, *H(z)* ექნება სიხშირეთა შესაბამისი ინტერვალი 0 დან π-მდე.

H(z) გადაცემის ფუნქციის ზოგადი სახეა:

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_n z^{-n}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n}}$$

ეს ფუნქცია შეესაბამება n რიგის ციფრულ ფილტრს. განვიხილოთ ზოგიერთი მაგალითი:

$$H(z) = \frac{0.2066 + 0.4131z^{-1} + 0.2066z^{-2}}{1 - 0.3695z^{-1} + 0.1958z^{-2}}$$
$$H(z) = \frac{0.894 - 1.789z^{-1} + 0.894z^{-2}}{1 - 0.778z^{-1} + 0.799z^{-2}}$$
$$H(z) = \frac{0.42 - 0.42z^{-2}}{1 - 0.44z^{-1} + 0.159z^{-2}}$$
$$H(z) = \frac{0.5792 + 0.4425z^{-1} + 0.5792z^{-2}}{1 + 0.4425z^{-1} + 0.1584z^{-2}}$$

თუ ფუნქცია ჩაწერილია როგორც z-ის დადებითი ხარისხების ფუნქცია, მრიცხველი და მნიშვნელი უნდა გავყოთ z სათანადო ხარისხზე, რომ მივიღოთ ზემოთ მოყვანილის ანალოგიური გამოსახულება.

ციფრული ფილტრი შესაძლოა წარმოდგენილი იყოს სტანდარტული სხვაობითი განტოლების სახით (standard difference equation). მას აქვს ზოგადი სახე:

$$y_n = \sum_{k=-N_1}^{N_2} b_k x_{n-k} - \sum_{k=1}^{N_3} a_k y_{n-k}$$

არსებობს პირდაპირი კავშირი გადამცემ განტოლებასა და სტანდარტულ სხვაობით განტოლებას სორის თუ დავუშვებთ, რომ N_1=0,

$$y_n = \sum_{k=0}^{N_2} b_k x_{n-k} - \sum_{k=1}^{N_3} a_k y_{n-k}$$

ასეთ ფორმაში a_k და b_k კოეფიციენტები წარმოადგენს გადაცემის ფუნქციის კოეფიციენტებს a_0 = 1-ით. ამრიგად განტოლება, რომელიც შეესაბამება პირველ მაგალითს, ასე ჩაიწერება:

$$y_n = 0.2066x_n + 0.4131x_{n-1} + 0.2066x_{n-2} + 0.3695y_{n-1} - 0.1958y_{n-2}$$

თუ a_k ყველა კოეფიციენტი 0-ის ტოლია, გარდა a_0, რომელიც 1-ის ტოლია, მაშინ შესაბამისი გადაცემის ფუნქციის მნიშვნელი ტოლია 1-ის:

$$y_n = 0.5x_n - 1.2x_{n-1} + 0.25x_{n-3}$$
$$H(z) = 0.5 - 1.2z^{-1} + 0.25z^{-3}$$

თუ გადაცემის ფუნქციის მნიშვნელი 1-ის ტოლია, მოცემული გვაქვს ფილტრი სასრული იმპულსური მახასიათებლით (FIR-finite impulse response) , თუ არა - უსასრულო იმპულსური მახასიათებლით (IIR- infinite impulse response) . ფილტრის ორივე ტიპი ფართოდ გამოიყენება სიგნალის დამუშავებაში.

იმისათვის რომ გავიგოთ მოცემული გადაცემის ფუნქციის მქონე სისტემის მახასიათებლები, უნდა ავაგოთ მისი მახასიათებელი მრუდი. **MATLAB** –ს ფუნქცია **freqz** გამოითვლის კომპლექსური ფუნქციის მნიშვნელობებს სამი არგუმენტის საშუალებით. პირველი არგუმენტია ვექტორი, რომელიც შეიცავს *B(z)* პოლინომის კოეფიციენტებს, მეორე არგუმენტია ვექტორი, რომელიც შეიცავს *A(z)* პოლინომის კოეფიციენტებს, ხოლო მესამე

არგუმენტია მთელი რიცხვი, რომელიც განსაზღვრავს ნორმირებულ სიხშირეთა რაოდენობას, რომელიც გამოყენებულია $[0.\ \pi]$ ინტერვალზე. კოეფიციენტები განისაზღვრება ფუნქციიდან, ხოლო წერტილების რაოდენობა, რომელიც გარდაქმნის ფუნქციის მნიშვნელობათა გამოსათვლელად გამოიყენება განსაზღვრავს გარჩევას, რომელიც ისე უნდა შეირჩეს რომ შევძლოთ ფილტრის ტიპის დადგენა.

პროგრამა გამოითვლის და აგებს მახასიათებელ მრუდებს ზემოთგანხილული 4 ფილტრისათვის:

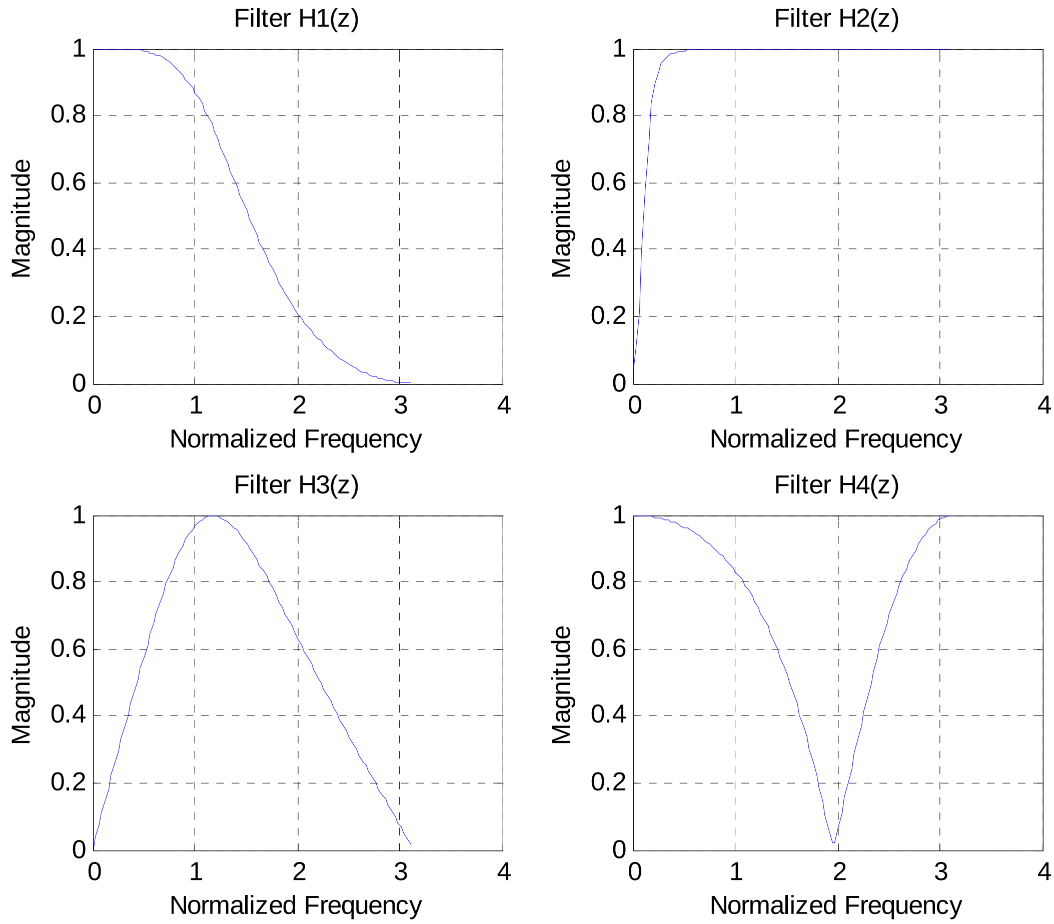
```
%     This programm datermines and plots the
%     magnitudes of four digital filters
%

B1 = [0.2066,0.4131,0.2066];
A1 = [1,-0.3695,0.1958];
[H1z,w1T] = freqz(B1,A1,100);
%

B2 = [0.894,-1.789,0.894];
A2 = [1,-1.778,0.799];
[H2z,w2T] = freqz(B2,A2,100);
%

B3 = [0.42,0,-0.42];
A3 = [1,-0.443,0.159];
[H3z,w3T] = freqz(B3,A3,100);
%

B4 = [0.5792,0.4425,0.5792];
A4 = [1,0.4425,0.1584];
[H4z,w4T] = freqz(B4,A4,100);
clf
subplot(221),plot(w1T,abs(H1z)),title('Filter H1(z)'),...
    xlabel('Normalized Frequency'),ylabel('Magnitude'),grid
subplot(222),plot(w2T,abs(H2z)),title('Filter H2(z)'),...
    xlabel('Normalized Frequency'),ylabel('Magnitude'),grid
subplot(223),plot(w3T,abs(H3z)),title('Filter H3(z)'),...
    xlabel('Normalized Frequency'),ylabel('Magnitude'),grid
subplot(224),plot(w4T,abs(H4z)),title('Filter H4(z)'),...
    xlabel('Normalized Frequency'),ylabel('Magnitude'),grid
```



ნახ. 14.10 სხვადასხვა ტიპის ციფრული ფილტრების მახასიათებელი მრუდები

MATLAB-ს აქვს აგრეთვე ფუნქცია **grpdelay**, რომელიც გამოიყენება ციფრული ფილტრის ჯგუფური შეყოვნების განსასაზღვრავად. ჯგუფური შეყოვნება ეს არის ფილტრის, როგორც სიხშირის ფუნქციის საშუალო შეყოვნების ზომა. ის განისაზღვრება როგორც ფილტრის ფაზური მახასიათებლის უარყოფითი პირველი წარმოებული. თუ გადაცემის ფუნქციის ფაზური მახასიათებელია $\theta(\omega)$, მაშინ ჯგუფური შეყოვნება იქნება:

$$\tau(\omega) = -\frac{d\theta(\omega)}{d\omega}$$

grpdelay ფუნქციას აქვს 3 არგუმენტი: B(z) და A(z) კოეფიციენტები და მთელი რიცხვი, რომელიც განსაზღვრავს შეყოვნების ზომას ნორმირებული სიხშირის 0 - π ინტერვალზე. რამდენადაც ეს ფუნქცია იყენებს **fft** ფუნქციას, სასურველია მესამე არგუმენტის მნიშვნელობა შეირჩეს ისე, რომ იყოს 2-ის ხარისხის ტოლი.

რაციონალურ წილადთა ჯამად დაშლა

ანალოგური და ციფრული ფილტრების ანალიზისას ხშირად გვჭირდება გადაცემის ფუნქციის დაშლა რაციონალურ წილადთა ჯამად. ეს პროცესი შეიძლება გამოვიყენოთ იმისათვის, რომ ფილტრი წარმოვადგინოთ ქვეფუნქციათა კასკადური ან პარალელური სტრუქტურის სახით. გარდა ამისა რაციონალურ წილადად დაშლა შეიძლება გამოვიყენოთ იმისათვის, რომ

შევასრულოთ უკუგარდაქმნა სიხშირულიდან დროით გარემოში. **MATLAB**-ს აქვს ფუნქცია **residue** რომელიც ახდენს ორი პოლინომის განაყოფის რაციონალურ წილადთა ჯამად დაშლას. ეს ფუნქცია შეგვიძლია გამოვიყენოთ გადაცემის ფუნქციისათვის, რადგან იგი სწორედ ორი პოლინომის განაყოფის სახითაა წარმოდგენილი.

დავუშვათ G არის υ ცვლადიანი ორი პოლინომის განაყოფი. ის შეიძლება წარმოვადგინოთ როგორც შერეული წილადი;

$$G(\upsilon)=\frac{B(\upsilon)}{A(\upsilon)}=\sum_{n=0}^{N}k_n\upsilon^n+\frac{N(\upsilon)}{D(\upsilon)}$$

წილადური ნაწილი არის ორი პოლინომის განაყოფი. მისი მრიცხველი შესაძლოა წარმოვადგინოთ როგორც რამდენიმე თანამამრავლის ნამრავლი, რომლებიც უდრის ამ პოლინომის ცვლადისა და მისი ფესვების სხვაობას. (მრიცხველი პოლინომის ფესვებს ეწოდება ფუნქციის ნულები, ხოლო მნიშვნელისას – პოლუსები).

$$\frac{N(\upsilon)}{D(\upsilon)}=\frac{b_1\upsilon^{n-1}+b_2\upsilon^{n-2}+\cdots+b_{n-1}\upsilon+b_1}{(\upsilon-p_1)^{m_1}(\upsilon-p_1)^{m_2}\cdots(\upsilon-p_1)^{m_r}}$$

რომელიც შეგვიძლია ჯამის სახით წარმოვადგინოთ;

$$\frac{N(\upsilon)}{D(\upsilon)}=\frac{C_{1,1}}{\upsilon-p_1}+\frac{C_{1,2}}{(\upsilon-p_1)^2}+\cdots+\frac{C_{1,m_1}}{(\upsilon-p_2)^{m_1}}+\frac{C_{2,1}}{\upsilon-p_2}+\frac{C_{2,2}}{(\upsilon-p_2)^2}+\cdots+\frac{C_{2,m_1}}{(\upsilon-p_2)^{m_1}}$$
$$+\frac{C_{r,1}}{\upsilon-p_r}+\frac{C_{r,2}}{(\upsilon-p_r)^2}+\cdots+\frac{C_{r,m_1}}{(\upsilon-p_r)^{m_r}}$$

ფუნქციას **residue** აქვს ორი **input** მნიშვნელობა: A და B პოლინომის კოეფიციენტები., გამოითვლის და გვაძლევს სამ ვექტორს – *r, p, k. r* ვექტორი შეიცავს $C_{i,j}$ კოეფიციენტებს, *p* – პოლუსების მნიშვნელობებს p_n და *k* – k_n მნიშვნელობებს. იმისათვის რომ სწორი ინტერპრეტაცია მივცეთ **residue** ფუნქციის შედეგად მიღებულ მნიშვნელობებს, კარგად უნდა დავუკვირდეთ ფორმულაში შემოღებული აღნიშვნების მნიშვნელობას. მნიშვნელოვანია გავითვალისწინოთ, რომ მოცემული წილადი რაციონალური წილადების ჯამის სახით რამდენიმეგვარად შეიძლება წარმოვადგინოთ. თუმცა აქვე აღვნიშნავთ, რომ ეს ფუნქცია ყოველთვის ერთიდაიგივე შედეგს მოგვცემს მრიცხველისა და მნიშვნელის მოცემული წყვილისათვის.

ფუნქციის საილუსტრაციოდ განვიხილოთ მაგალითი:

$$F(z)=\frac{z^2}{z^2-1.5z+0.5}$$

MATLAB საშუალებით იგი შეგვიძლია დავშალოთ რაციონალური წილადების ჯამად:

```
B=[1, 0, 0];
A=[1, -1.5, 0.5];
[r,p,k]=residue(B,A)
```

შედეგად მივიღებთ:

$$r=\begin{bmatrix}2.0\\-0.5\end{bmatrix}\quad p=\begin{bmatrix}1.0\\0.5\end{bmatrix}\quad k=[1]$$

ამიტომ გვექნება:

$$F(z)=\frac{z^2}{z^2-1.5z+0.5}=1+\frac{2}{z-1.0}-\frac{0.5}{z-0.5}$$

განვიხილოთ შემდეგი წილადი:

$$H(z)=\frac{1}{z^2-3.5z+1.5}$$

```
B=[1];
A=[1, -3.5, 1.5];
[r,p,k]=residue(B,A)
```

შედეგად მივიღებთ:

$$r=\begin{bmatrix}0.4\\-0.4\end{bmatrix}\quad p=\begin{bmatrix}3.0\\0.5\end{bmatrix}\quad k=[\ ]$$

ესე იგი გვექნება:

$$F(z)=\frac{1}{z^2-3.5z+1.5}=\frac{0.4}{z-3}-\frac{0.4}{z-0.5}$$

თუ გვსურს ჩავწეროთ ფორმულა z უარყოფითი ხარისხებით, წილადის მრიცხველი და მნიშვნელი გავამრავლოთ z^{-1}:

$$F(z)=\frac{0.4z^{-1}}{1-3z^{-1}}-\frac{0.4z^{-1}}{1-0.5z^{-1}}$$

სავარჯიშო

თითოეული ფუნქციისათვის ააგეთ მახასიათებელი მრუდი (მაგნიტუდე). განსაზღვრეთ ფილტრის ტიპი და მახასიათებლები. ციფრული ფილტრებისათვის ისარგებლეთ ნორმირებული სიხშირით.

1. $H(s)=\dfrac{s^2}{s^2+\sqrt{2}s+1}$
2. $H(z)=\dfrac{0.707z-0.707}{z-0.414}$
3. $H(z)=-0.163-0.058z^{-1}+0.116z^{-2}+0.2z^{-3}+0.116z^{-4}-0.058z^{-5}-0.163z^{-6}$
4. $H(s)=\dfrac{5s+1}{s^2+0.4s+1}$

14.4 ციფრული ფილტრის განხორციელება

ანალოგური ფილტრები ხორციელდება მოწყობილობებში ისეთი კომპონენტების გამოყენებით, როგორიცაა რეზისტორი და კონდენსატორი. ციფრული ფილტრი ხორციელდება პროგრამულად. ციფრული ფილტრი შეგვიძლია განვახორციელოთ როგორც გადაცემის ფუნქცია H_z ან სტანდარტული difference განტოლება. ფუნქციის **filter** საწყისი არგუმენტია სიგნალი და საბოლოო შედეგიც სიგნალია, რომელმაც გაიარა ფილტრი. განტოლება განსაზღვრავს შემავალის გამომავალ სიგნალად გარდაქმნის საფეხურებს.

დამოკიდებულება შემავალსა x_n და გამომავალ y_n სიგნალს შორის ასეთია:

$$y_n = \sum_{k=-N_1}^{N_2} b_k x_{n-k} - \sum_{k=1}^{N_3} a_k y_{n-k}$$

მაგალითად:

$$y_n = 0.04x_{n-1} + 0.17x_{n-2} + 0.25x_{n-3} + 0.17x_{n-4} + 0.04x_{n-5}$$
$$y_n = 0.42x_n - 0.42x_{n-2} + 0.44x_{n-1} - 0.16x_{n-2}$$
$$y_n = 0.33x_{n+1} + 0.33x_n + 0.33x_{n-1}$$

პირველი ფილტრის გამომავალი სიგნალი დამოკიდებულია შემავალი სიგნალის მხოლოდ წარსულ მნიშვნელობებზე. მაგალითად რომ გამოვთვალოთ y_{10} უნდა ვიცოდეთ x_9, x_8, x_7, x_6, x_5 მნიშვნელობები. ამ ტიპის ფილტრი წარმოადგენს *FIR* ფილტრს და მისი გადამცემი ფუნქციის მნიშვნელი 1-ის ტოლია. მეორე ფილტრი მოითხოვს არა მარტო შემავალი სიგნალის მნიშვნელობებს, არამედ გამომავალი სიგნალის წრსულ მნისვნელობებსაც, რომ გამოითვალოს ახალი გამომავალი სიგნალი. ამ ტიპის ფილტრს უწოდებენ *IIR* ფილტრს. მესამე ფილტრი არის აგრეთვე *FIR* ფილტრი, მაგრამ გამომავალი სიგნალი დამოკიდებულია მხოლოდ შემავალ სიგნალზე. თუმცა უნდა შევნიშნოთ, რომ ინდექსები მესამე განტოლებაში მოითხოვს, რომ შეგვეძლოს წინასწარ განვჭვრიტოთ შემავალი სიგნალის მნიშვნელობები. ასე, რომ თუ გვინდა გამოვითვალოთ y_5 უნდა ვიცოდეთ x_6, x_5, x_4.
ეს მოთხოვნა პრობლემას არ წარმოადგენს, თუ შემავალ სიგნალს ვითვლით ფუნქციის საშუალებით, ან მისი მნიშვნელობები მოცემულია ფაილის სახით. მაგრამ პრბლემა შეიქმნება, თუ ეს სიგნალი იქმნება რეალურად, ექსპერიმენტის მიმდინარეობისას.

MATLAB-ში სიგნალის 'გაფილტრვა' ხდება ფუნქციით **filter**. იგულისხმება, რომ სტანდარტულ difference განტოლებას აქვს სახე:

$$y_n = \sum_{k=0}^{N_2} b_k x_{n-k} - \sum_{k=1}^{N_3} a_k y_{n-k}$$

რაც შეესაბამება გადაცემის ფუნქციას:

$$H(z) \;= \frac{B(s)}{A(s)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_n z^{-n}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n}}$$

filter ფუნქციის პირველი 2 არგუმენტია $[a_k]$ და $[b_k]$ კოეფიციენტების ვექტორი.

მესამე არგუმენტია შემავალი სიგნალი.

```
>>B=[0.0, 0.04, 0.17, 0.25, 0.17, 0.04];
>>A=[1];
>>y=filter(B,A,x);
```

ეს ფუნქცია არ შეგვიძლია გამოვიყენოთ მესამე ფილტრისათვის, რადგან დიფერენციალურ განტონებას შეუძლია გამოითვალოს სიგნალის მნიშვნელობები მხოლოდ k=0 დან. მესამე განტოლებაში მოითხოვება, რომ პირველი ჯამი დაიწყოს k=-1 მნიშვნელობით. ამ შემთხვევაში ფილტრს 'მოვარგებთ' ე.წ. ვექტორული არითმეტიკის საშუალებით. დავუშვათ,რომ შემავალი სიგნალი წარმოდგენილია x ვექტორის სახით. შესაბამისი გამომავალი y სიგნალი შეგვიძლია გამოვითვალოთ შემდეგი ბრძანებების საშუალებით:

```
N=length(x);
y(1)=0.33*x(1)+0.33*x(2);
for n=2:N-1
       y(n) = 0.33*x(n-1) + 0.33*x(n) + 0.33*x(n-1);
end
y(N) = 0.33*x(N-1) + 0.33*x(N);
```

ჩვენ დავუშვით, რომ x-ის ის მნიშვნელობები, რომელთათვისაც არ გვაქვს სიდიდეები მოცემული, 0-ის ტოლია [x(-1) და x(N+1)] . გამოთვლის სხვა გზაც არსებობს:

```
N=length(x);
y(1)=0.33*x(1)+0.33*x(2);
y(2:N-1) = 0.33*x(3:N) + 0.33*x(2:N-1) + 0.33*x(1:N-2);
y(N)=0.33*x(N-1) + 0.33*x(N);
```

ნებისმიერი ფილტრი შეგვიძლია განვახორციელოთ ვექტორული ოპერაციებით, მაგრამ ფუნქცია ყოველთვის იძლევა უფრო მარტივ ამოხსნას.

ფილტერ ფუნქცია იძლევა ორ გამოსავალ არგუმენტს:

```
[y, state] = filter(b,a,x)
```

y ვექტორი შეიცავს გამომავალ არგუმენტს, როცა *x* ვექტორი – საწყისი სიგნალია, ხოლო ვექტორი state სიდიდეთა საბოლოო მწკრივს, რომელსაც ფილტრი იყენებს. მაგალითად, თუ გვინდა გავფილტროთ x2 ვექტორი, რომელიც x სიგნალის სხვა სეგმენტს წარმოადგენს, შეგვიძლია მივუთითოთ, რომ საწყისი პირობები განსაზღვრულია ვექტორში state, ამ შემთხვევაში x და x2 ვექტორები განიხილება როგორც ერთი გრძელი ვექტორი ნაცვლად 2 სხვადასხვა ვექტორისა.

```
y  = filter(b,a,x2,state);
```

ბოლოს, ფუნქცია **conv** შეგვიძლია გამოვიყენოთ, იმისათვის, რომ გამოვითვალოთ FIR ფილტრის გამოსავალი მნიშვნელობები. ეს ფუნქცია გამოიყენება მხოლოდ FIR ფილტრისათვის. უმჯობესია **conv** გამოვიყენოთ პოლინომების გადამრავლებისათვის, ხოლო **filter** სიგნალის გასაფილტრად.

სავარჯიშო

შემდეგი გადაცემის ფუნქცია შეიქმნა იმისათვის, რომ ფილტრმა გაატაროს სიგნალი სიხშირეებზე 500 – 1500 ჰერცს შორის. სიგნალი ათვლილია 5 კილოჰერცი სიხშირით:

$$H(z) = \frac{0.42z^2 - 0.42}{z^2 - 0.443z + 1.59}$$

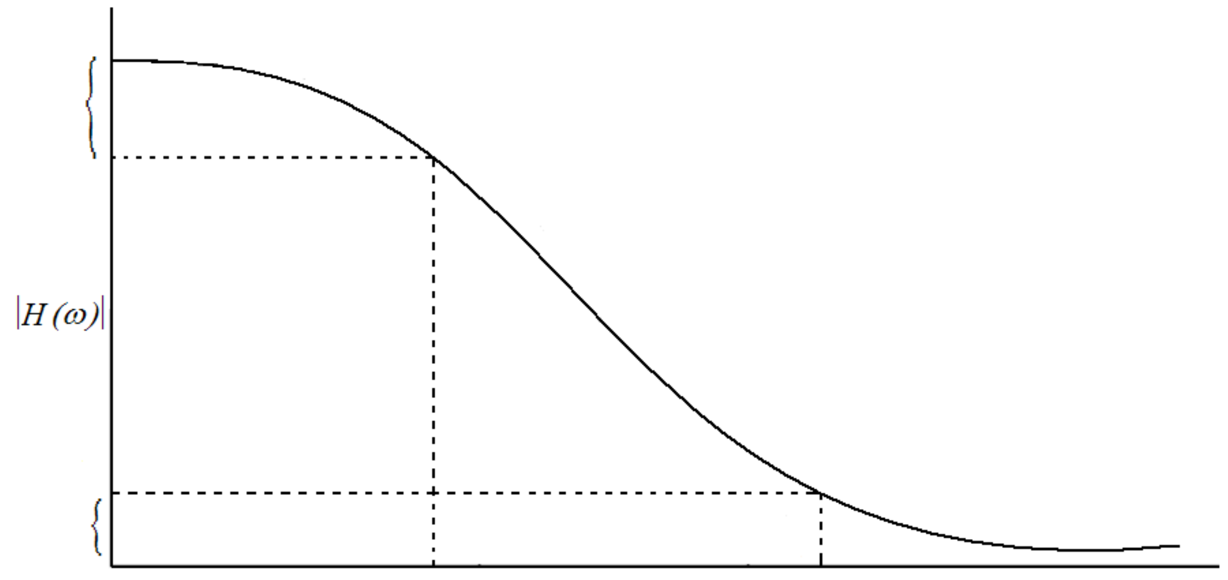
აიღეთ შემდეგი სიგნალი, გამოთვალეთ როგორი იქნება იგი ფილტრში გავლის შემდეგ. ააგეთ ორივე სიგნალის სიხშირული მრუდი ერთიოდაიგივე ნახაზზე, რომ თვალსაჩინოდ გამოჩნდეს ფილტრის ეფექტი.

1. $x_k = \sin(2\pi 1000kT)$
2. $x_k = 2\cos(2\pi 100kT)$
3. $x_k = -\sin(2\pi 200kT)$
4. $x_k = \cos(2\pi 1600kT)$

14.5 ციფრული ფილტრის დიზაინი

IIR (infinit impulse response filter) (ფილტრი უსასრულო იმპულსური მახალიათებლით)

MATLAB –ს აქვს ფუნქციები ოთხი სხვადასხვა ტიპის ციფრული ფილტრისათვის, რომელიც ეყრდნობა ანლოგური ფილტრის დიზაინს. Butterworth ფილტრს აქვს ბრტყელი მახასიათებელი მრუდი ბრტყელი გატარების(ზღვრული) და მოკვეთის(ცუტოფფ) არეებით. Chebishev-ის და ელიფსური ფილტრების მახასიათებელი მრუდი კი ტალღოვანია თითქმის მთელ უბანზე. მაგრამ Chebishev-ის ფილტრს აქვს ყველაზე ვიწრო გარდამავალი უბანი, რაც მის უპირატესობას წარმოადგენს სხვა ფილტრებთან შედარებით.



ნახ. 14.11 გადაცემის ფუნქციის ტალღოვანი უბნები

MMATLAB-ის ფუნქციებს დაბალსიხშირული IIR ფილტრებისათვის ანალოგურ პროტოტიპზე დაყრდნობით შემდეგი სახე აქვს:

```
[B,A] = butter(N,Wn);
[B,A] = cheby1(N,Rp,Wn);
[B,A] = cheby2(N,Rs,Wn);
[B,A] = ellip(N,Rp,Rs,Wn);
```

ფუნქციის შესავალ არგუმენტებს წარმოადგენს ფილტრის რიგი (N), (Rp , Rs) ripple და ნორმირებული მომკვეთი (cutoff) სიხშირე(Wn). ნორმირებული სიხშირე ემყარება ნაიქვისტის სიხშირეს, რომელიც 1-ის ტოლად მიიჩნევა. (აღვნიშნავთ, რომ ეს განსხვავდება **freqz** ფუნქციის ნორმირებული სიხშირისაგან). შედეგად მიღებული ვექტორები B და A არის გადაცემის ფუნქციის კოეფიციენტები.

რომ შევქმნათ ზოლოვანი ფილტრი, არგუმენტები ისეთივე უნდა გამოვიყენოთ, როგორც დაბალსიხშირული ფილტრისას, მხოლოდ Wn იქნება ორელემენტიანი ვექტორი, რომელიც ახასიათებს ზოლის საწყის და საბოლოო ნორმირებულ სიხშირეებს (Wn(1)-Wn(2)).

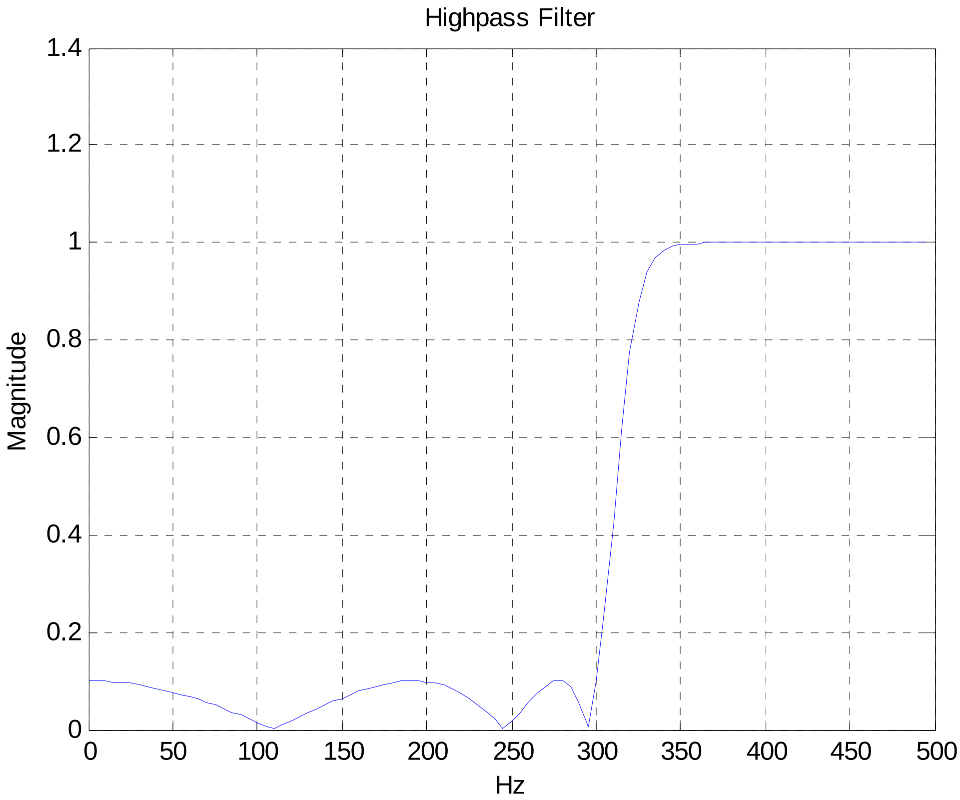
მაღალსიხშირული ფილტრის შესაქმნელად გვჭირდება დამატებითი პარამეტრი ‘high’,

```
[B,A] = butter(N,Wn,’high’);
[B,A] = cheby1(N,Rp,Wn, ’high’);
[B,A] = cheby2(N,Rs,Wn, ’high’);
[B,A] = ellip(N,Rp,Rs,Wn, ’high’);
```

bandstop ფილტრის შესაქმნელად არგუმენტები იგივე რჩება, მხოლოდ ნაცვლად ‘high’ იწერება ‘stop’, თანაც Wn არგუმენტი უნდა იყოს ორელემენტიანი ვექტორი, რომელიც ზოლის საზღვრებს ახასიათებს.

საილუსტრაციოდ, დავუშვათ გვინდა შევქმნათ მე-6 რიგის ჩებიშევის II ტიპის ფილტრი. ვთქვათ ასევე გვინდა შევზღუდოთ გატარების ტალღა(ripple) 0.1-ით (20 დეციბელი). ფილტრი გვინდა გამოვიყენოთ სიგნალისათვის, რომელიც აითვლება 1კჰერცზე, რაც ნიშნავს, რომ ნაიქვისტის სიხშირე 500 ჰერცია. მოკვეთის cutoff სიხშირე უნდა იყოს 300 ჰერცი, ე.ი. ნორმირებული სიხშირეა 300/500, ანუ 0.6.

```
[B,A] = cheby2(6,20,0.6, 'high');
[H,wT] = freqz(B,A,100);
T = 0.001;
hertz = wT/(2*pi*T);
plot(hertz, abs(H)), title('Highpass Filter'),...
xlabel('Hz'),ylabel('Magnitude'), grid
```



ნახ. 14.12 ჩებიშევის II ტიპის ფილტრი

ეს ფილტრი რომ სიგნალს მოვარგოთ უნდა გამოვიყენოთ ბრძანება:

y = filter(B,A,x);

B და A მნიშვნელობები შეგვიძლია გამოვიყენოთ აგრეთვე ფილტრის სტანდარტული difference განტოლების განსასაზღვრავად.

პირდაპირი IIR ფილტრი

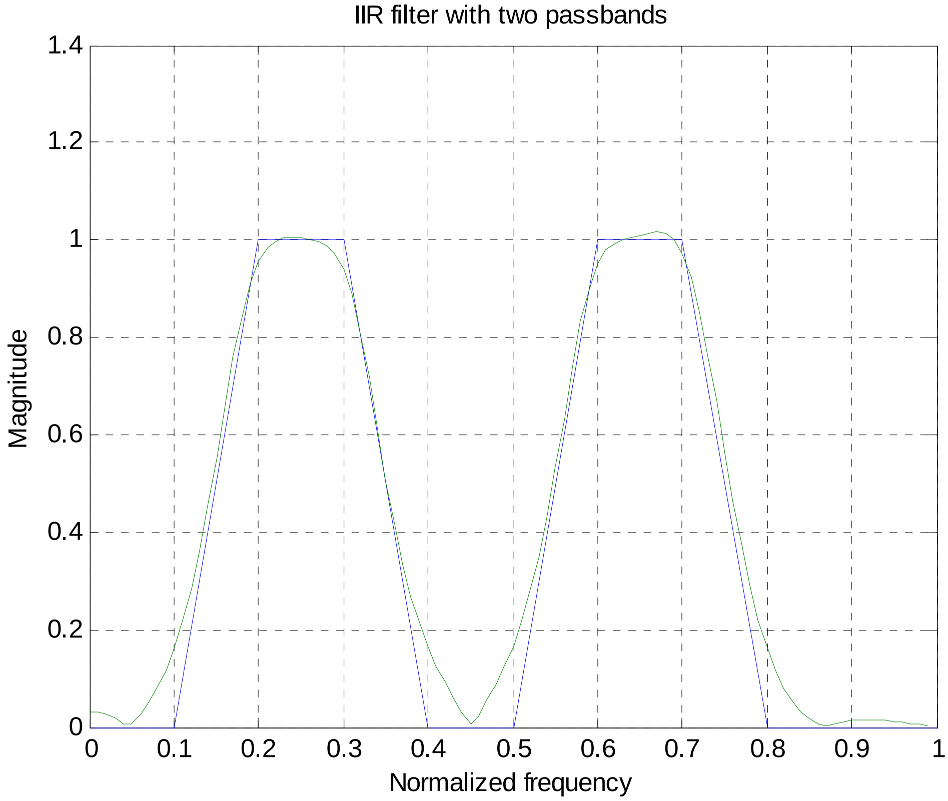
MATLAB-ს აქვს ფუნქცია Yule-Walker ფილტრისათვის. ვექტორები f და m ახასიათებს სიხშირულ-ამპლიტუდურ თვისებებს სიხშირის არეში 0-1.რაც შეესაბამება 0 დან ნაიქვისტის სიხშირემდე ინტერვალს. სიხშირეები f – ში უნდა დაიწყოს 0 – დან და დასრულდეს 1-ით.

დალაგებული უნდა იყოს ზრდის მიხედვით. ამპლიტუდები m–ში უნდა შეესაბამებოდეს სიხშირეებს f-ში. შემდეგი მაგალითში განხილულია ფილტრი გატარების 2 ზოლით;

```
m=[0 0 1 1 0 0 1 1 0 0];
f=[0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 1];

[B,A] = yulewalk(12,f,m);
[H,wT] = freqz(B,A,100);
T = 0.001;

plot(f,m,wT/pi,abs(H)),...
title('IIR filter with two passbands'),...
xlabel('Normalized frequency'),ylabel('Magnitude'),grid
```



ნახ. 14.13 Yule-Walker ფილტრი

პირდაპირი FIR ფილტრი

ასეთი ფილტრი MATLAB-ში ხორციელდება Parks-MacClelanis ალგორითმით, რომელიც თავის მხრივ იყენებს **remez**-ის ალგორითმს. გავიხსენოთ, რომ FIR ფილტრს ესაჭიროება მხოლოდ B კოეფიციენტები, რადგან H(z) მნიშვნელის პოლინომი 1-ის ტოლია. MATLAB – ის ფუნქცია **remez** გვაძლევს ერთ ვექტორს :

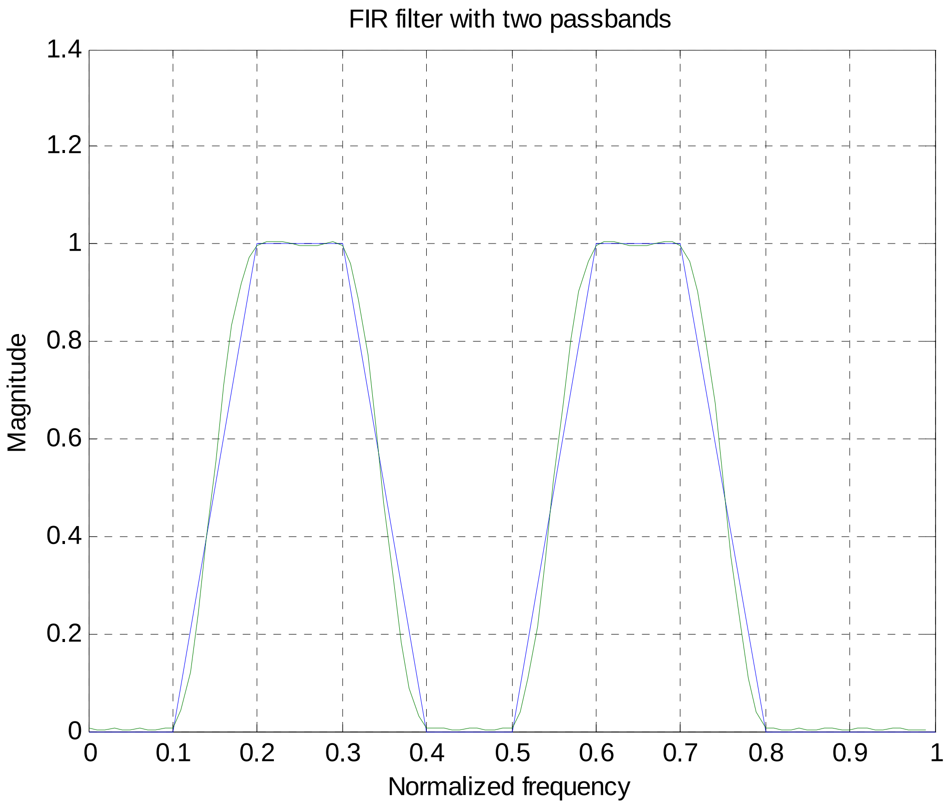
```
B = remez(n,f,m)
```

პირველი არგუმენტია ფილტრის რიგი, f და m ძალიან გავს წინა თავში განხილულ სიდიდეებს, მაგრამ დამატებით წერტილების რაოდენობა f და m ვექრორებში უნდა იყოს ლუწი.

```
m=[0 0 1 1 0 0 1 1 0 0];
f=[0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 1];

[B,A] = remez(50,f,m);
[H,wT] = freqz(B,[1],100);


plot(f,m,wT/pi,abs(H)),…
title('FIR filter with two passbands'),…
xlabel('Normalized frequency'),ylabel('Magnitude'),grid
```



ნახ. 14.14 ფილტრი სასრული იმპულსური მახასიათებლით

სავარჯიშო

გამოყენეთ MATLAB –ის ფუნქციები შემდეგი სახის ფილტრების შესაქმნელად. ააგეთ ფილტრის მახასიათებელი მრუდი

1. დაბალსიხშირული IIR ფილტრი ზღვრული ცუტოფფ სიხშირით 75 ჰერცი, ათვლის სიხშირე 500 ჰერცი. (ფილტრის რიგი 5)

2. მაღალსიხშირული IIR ფილტრი ზღვრული cutoff სიხშირით 100 ჰერცი, ათვლის სიხშირე 1 კილოჰერცი. (ფილტრის რიგი 6)
3. დაბალსიხშირული FIR ფილტრი ზღვრული cutoff სიხშირით 75 ჰერცი, ათვლის სიხშირე 500 ჰერცი. (ფილტრის რიგი 5)
4. ზოლოვანი FIR ფილტრი გატარების ზოლით 100-200 ჰერცი, ათვლის სიხშირე 1 კილოჰერცი. (ფილტრის რიგი 80)

პრობლემა: არხების გამყოფი ფილტრი

კოსმოსური ხომალდის მიერ ღია სივრცეში გადაღებული გამოსახულება გადმოიცემა დედამიწაზე. ეს მონაცემები გარდაიქმნება ციფრულ სიგნალად, რომლის მიღების და დამუშავების შემდეგ მიიღება საწყისი გამოსახულება. ინფორმაციის მიღება წარმოებს რამდენიმე სენსორზე. სენსორზე მიღებული სიგნალის სიხშირული შედგენილობა დამოკიდებულია გაზომილი მონაცემების ტიპზე.

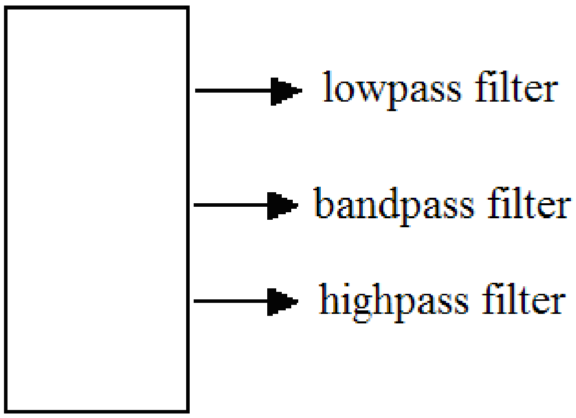
სხვადასხვა სიხშირის სიგნალები შეიძლება გაერთიანდეს ერთ სიგნალად მოდულაციის საშუალებით. მაგალითად დავუშვათ, გვინდა გადავცეთ 3 სიგნალი პარალელურად. პირველი სიგნალი შეიცავს სიხშირულ კომპონენტებს 0-დან 100 ჰერცამდე, მეორე-500 ჰერციდან 1 კილოჰერცამდე, მესამე კი 2 კილოჰერციდან 5 კილოჰერცამდე. დავუშვათ სიგნალი რომელიც შეიცავს ამ სამ სიგნალს აითვლება სიხშირით 10 კილოჰერცი. იმისათვის, რომ მიღებული სიგნალიდან გამოვყოთ თითოეული მდგენელი ცალკ-ცალკე, გვჭირდება დაბალსიხშირული ფილტრი ზედა ზღვრული სიხშირით 100 ჰერცი, ზოლოვანი ფილტრი სიხშირეთა ინტერვალით 500-1000 ჰერცი და მაღალსიხშირული ფილტრი ქვედა ზღვრული სიხშირით 2 კილოჰერცი. ფილტრის რიგი საკმაოდ მაღალი უნდა იყოს, რომ მივიღოთ ვიწრო გარდამავალი ზოლი რათა სიხშირეები რაც შეიძლება მკვეთრად გაიმიჯნოს.

1. ამოცანის დასმა:

შევქმნათ 3 ფილტრი, რომელიც უნდა გამოვიყენოთ 10 კილოჰერც სიხშირით ათვლილი სიგნალისათვის. ერთი მათგანი უნდა იყოს დაბალსიხშირული ფილტრი, რომელიც ჩამოჭრის 100 ჰერცზე უფრო მაღალ სიხშირულ კომპონენტებს, მეორე უნდა იყოს ზოლური ფილტრი, რომელიც გაატარებს სიხშირეებს 500-1000 ჰერც ინტერვალში, მესამე ფილტრი უნდა იყოს მაღალსიხშირული, რომელიც გაატარებს ყველა სიხშირეს 2 კილოჰერციდან.

2. INPUT/OUTPUT აღწერა

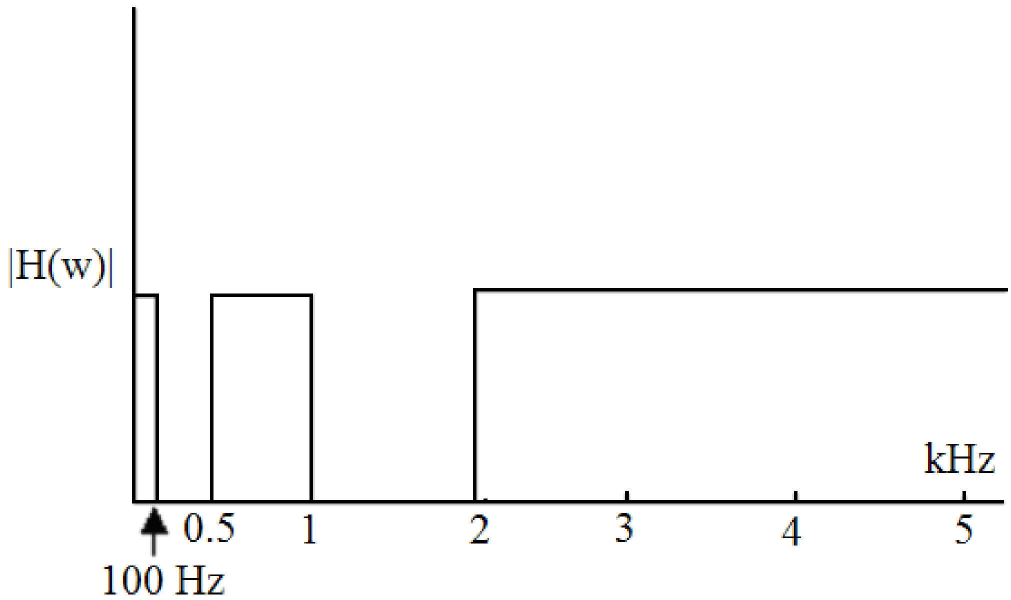
ამ პრობლემაში არ გვაქვს საწყისი მონაცემები, ხოლო შედეგად ვრებულობთ ვექტორებს კოეფიციენტებით, რომელიც განსაზღვრავს ამ სამი ფილტრის გადაცემის ფუნქციებს. იხ. დიაგრამა ნახ. 14.15.



ნახ. 14.15 I/O დიაგრამა

3. სახელდახელო ამოხსნა

ნახ. 14.17 ნაჩვენებია სიხშირული ინტერვალი 0-დან ნაიქვისტის სიხშირემდე (500 კჰც) სამი ფილტრითურთ. უნდა გამოვიყენოთ უტტერწორტჰ ფილტრი იმისათვის, რომ მივიღოთ გატარების და მოკვეთის ბრტყელი ზოლი. შესაძლოა დაგვჭირდეს ექსპერიმენტი ფილტრის რიგის შესარჩევად, იმისათვის რომ დავრწმუნდეთ ფილტრების გატარების ზოლები არ ფარავენ ერთმანეთს და ამდენად არხები მკვეთრად გაიმიჯნება.



ნახ. 14.16 ფილტრის ესკიზი

4. MATLAB ამოხსნა

ეს პროგრამა ითვლის ნორმირებულ სიხშირეებს(0-დან 1-მდე 1-შეესაბამება ნაიქვისტის სიხშირეს) ზღვრული სიხშირეებისათვის **butter** ფუნქციაში, მას შემდეგ რაც გამოვითვლით კოეფიციენტებს, მივმართვთ **freqz** ფუნქციას, რომ ავაგოთ ფილტრის მახასიათებელი მრუდი. გავიხსენოთ, რომ freqz ფუნქცია ახდენს სიხშირეთა ნორმირებას 0 - π ინტერვალში, სადაც π წარმოადგენს ნაიქვისტის სიხშირეს. სიხშირის ერთეულად ავიღებთ ჰერცს.

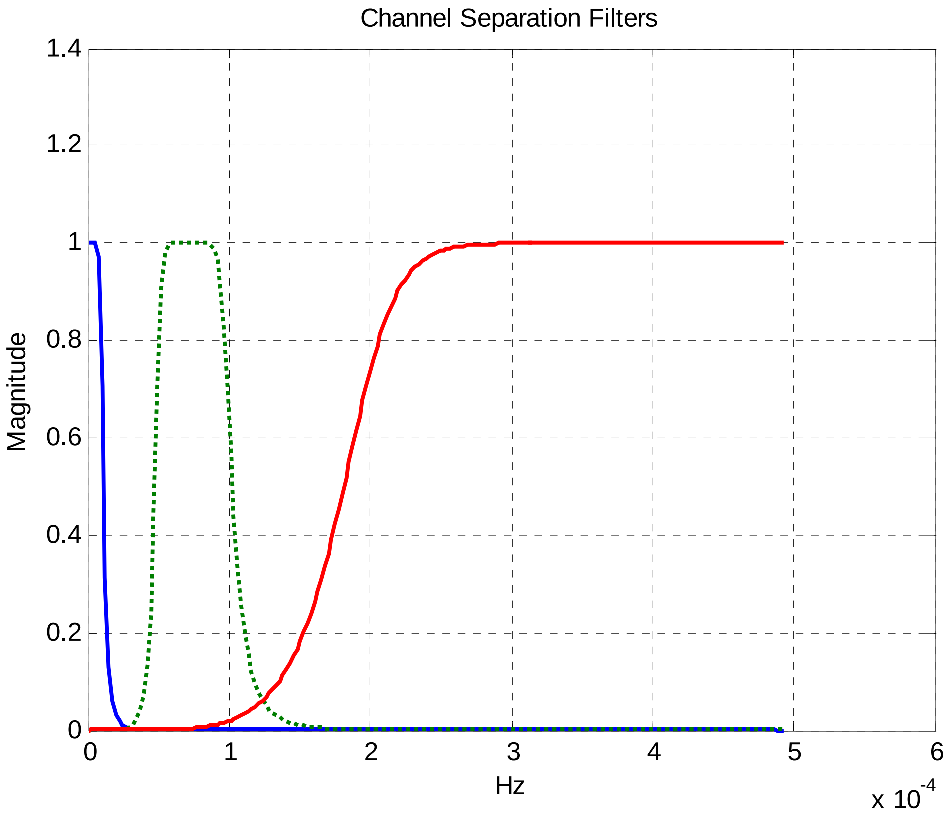
```
%         This program designs three digital filters
%         for use in a channel separation problem

fs = 10000;      %sampling frequency

T= 1/fs;         %sampling time
fn = fs/2;       %Nyquist frequency
f1n = 100/fn;    %normalized lowpass cutoff
f2n = 500/fn;    %normalized bandpass left cutoff
f3n = 1000/fn;   %normalized bandpass left cutoff
f4n = 2000/fn;   %normalized highpass cutoff
%
[B1,A1] = butter(5,f1n);
[B2,A2] = butter(5,[f2n,f3n]);
[B3,A3] = butter(5,f4n,'high');
%
[H1,wT] = freqz(B1,A1,200);
[H2,wT] = freqz(B2,A2,200);
[H3,wT] = freqz(B3,A3,200);
%
hertz = wT/2*pi*T;
plot(hertz,abs(H1),'-
',hertz,abs(H2),':',hertz,abs(H3),'r'),...
    title('Channel Separation Filters'),...
    xlabel('Hz'),ylabel('Magnitude'),grid
```

5. შემოწმება

სამივე ფილტრის მახასიათებელი მრუდი (მოდული) წარმოდგენილია



ნახ. 14.17 არხების გამყოფი სამი ფილტრის მახასიათებელი მრუდები

განხილული იქნა MATLAB-ის რამდენიმე ფუნქცია, რომელიც საჭიროა სიგნალის დამუშავებისათვის. ციფრული სიგნალის სიხშირული შედგენილობის ანალიზისათვის გამოვიყენეთ **fft** ფუნქცია. ანალოგური და ციფრული ფილტრების სიხშირული მახასიათებლების შესასწავლად გამოვიყენეთ **freqs** და **freqz** ფუნქციები. კომპლექსური სიგნალიდან შეგვიძლია გამოვითვალოთ ფილტრის ამპლიტუდა და ფაზა. **filter** ფუნქცია გამოიყენება **IIR** და **FIR** ფილტრების განსახორციელებლად. და ბოლოს განხილული იქნა რამდენიმე ფუქცია **IIR** და **FIR** ფილტრების დიზაინისთვის.

MATLAB ბრძანებები და ფუნქციები

butter	Butterworth ციფრული ფილტრის დიზაინი
cheby1	Chebyshev I ტიპის ფილტრის დიზაინი
cheby2	Chebyshev II ტიპის ფილტრის დიზაინი
ellip	ელიფსური ციფრული ფილტრის დიზაინი
fft	გამოითვლის სიგნალის სიხშირულ შემცველობას (სწრაფი ფურიე გარდაქმნა)
filter	შემომავალ სიგნალს მოარგებს ციფრულ ფილტრს
freqs	გამოითვლის ანალოგურ სიხშირულ შემცველობას
freqz	გამოითვლის ციფრულ სიხშირულ შემცველობას
grpdelay	დაადგენს ციფრული სიგნალის ჯგუფური შეყოვნების ზომას
ifft	გამოითვლის FFT შებრუნებულს (შებრუნებული ფურიე გარდაქმნა)
remez	ოპტიმალური FIR ციფრული ფილტრის დიზაინი
residue	რაციონალურ წილადად დაშლა

unwrap	გვაძლევს ფაზურ კუთხის ყველა შესაძლო მნიშვნელობას
yulewalk	ოპტიმალური IIR ციფრული ფილტრის დიზაინი

პრობლემები

1-6 ამოცანები დაკავშირებულია ამ თავში განხილულ პრობლემებთან, ხოლო 7-23 – ახალ საინჟინრო პრობლემებს უკავშირდება

არხების გამყოფი ფილტრი. ეს პრობლემა უკავშირდება ამ თავში განხილულ პრობემას არხების გამყოფი ფილტრის დიზაინის შესახებ. შევეცადოთ შევქმნათ სისტემის კომპიუტერული მოდელი:

1. პირველ რიგში უნდა შევქმნათ სიგნალი, რომელიც შეიცავს სხვადასხვა სიხშირულ მდგენელს. ამას მივაღწევთ სხვადასხვა სიხშირის და ამპლიტუდეს სინუსოიდათა შეკრებით, რომლებიც აითვლება 10 კილოჰერცი სიხშირით. პირველი სიგნალი შედგება სამი სინუსოიდისაგა, რომელთა სიხშირეებია: 25 ჰც, 40ჰც და 75ჰც, მეორე სინუსოიდა შეიცავს სამ სინუსოიდას სიხშირეებით: 500ჰც, 730 ჰც და 850ჰც, მესამე სინუსოიდა კი წარმოადგებს სამი სინუსოიდის ჯამს მაღალი სიხშირეებით: 3 500ჰც, 4 000ჰც და 4 200 ჰც. შეარჩიეთ სინუსოიდებისთვის სხვადასხვა ამპლიტუდები და ფაზები და ააგეთ 500 წერტილი პირველი, მეორე და მესამე სიგნალისათვის გრაფიკი ცალკ-ცალკე.
2. გამოთვლეთ და ააგეთ პირველ ამოცანაში აღწერილი სიგნალების სიხშირული მახასიათებლები (მაგნიტუდა და ფაზა). გრაფიკის აგებისას x ღერძზე დაიტანეთ სიხშირის მნიშვნელობები ჰერცებში.
3. შეკრიბეთ პირველ ამოცანაში აღწერილი სიგნალები. ააგეთ სიგნალი დროით და სიხშირულ სივრცეში(არეში). x ღერძზე გადაზომეთ სიხშირე ჰერცებში.
4. მოარგეთ დაბალსიხშირული ფილტრი მესამე ამოცანაში აღწერილ სიგნალს. ააგეთ შედეგად მიღებული სიგნალის დროითი და სიხშირული (მაგნიტუდე) მახასიათებლების გრაფიკი. შეადარეთ იგი 1 და 2 ამოცანაში აღწერილ სათანადო სიგნალის გრაფიკს. დროითი გრაფიკი პირველი სიგნალისათვის მისი მსგავსი უნდა იყოს შესაძლო ფაზური წანაცვლების გარეშე, ხოლო სიხშირული მრუდები თითქმის იდენტური უნდა იყოს.
5. გაიმეორეთ მე-4 ამოცანა ზოლოვანი ფილტრისათვის. შეადარეთ იგი 1 და 2 ამოცანაში აღწერილ სათანადო სიგნალის გრაფიკს. დროითი გრაფიკი მეორე სიგნალისათვის მისი მსგავსი უნდა იყოს შესაძლო ფაზური წანაცვლების გარეშე, ხოლო სიხშირული მრუდები თითქმის იდენტური უნდა იყოს.
6. გაიმეორეთ მე-4 ამოცანა მაღალსიხშირული ფილტრისათვის. შეადარეთ იგი 1 და 2 ამოცანაში აღწერილ სათანადო სიგნალის გრაფიკს. დროითი გრაფიკი მესამე სიგნალისათვის მისი მსგავსი უნდა იყოს შესაძლო ფაზური წანაცვლების გარეშე, ხოლო სიხშირული მრუდები თითქმის იდენტური უნდა იყოს.

ფილტრის მახასიათებლები. თითოეული ფილტრისათვის განსაზღვრეთ მახასიათებლები ზღვრული, გარდამავალი და მოკვეთის (stopband). ზღვრული სიხშირის შესაფასებლად გამოიყენეთ 0.7 დონე, ხოლო მოკვეთის სიხშირისათვის 0.1.

7. $H(s) = \dfrac{0.5279}{s^2 + 1.0275s + 0.5279}$

8. $H(s)=\dfrac{s^2}{s^2+0.1117s+0.0062}$

9. $H(s)=\dfrac{1.05s}{s^2+1.05s+0.447}$

10. $H(s)=\dfrac{s^2+2.2359}{s^2+2.3511s+2.2359}$

11. $H(z)=\dfrac{0.2066-0.4131z^{-1}+0.2066z^{-2}}{1-0.3695z^{-1}+0.1958z^{-2}}$

12. $H(z)=\dfrac{0.894-1.789^{-1}+0.894z^{-2}}{1-1.778z^{-1}+0.799z^{-2}}$

13. $H(z)=\dfrac{0.42-0.42z^{-2}}{1-0.443z^{-1}+0.159z^{-2}}$

14. $H(z)=\dfrac{0.5792-0.4425z^{-1}+0.1584z^{-2}}{1-0.4425z^{-1}+0.1584z^{-2}}$

15. $y_n=0.04x_{n-1}+0.17x_{n-2}+0.25x_{n-3}+0.17x_{n-4}+0.04x_{n-5}$

16. $y_n=0.42x_n-0.42x_{n-2}+0.44y_{n-1}-0.16y_{n-2}$

17. $y_n=0.33_{n+1}+0.33x_n+0.33x_{n-1}$

18. $y_n=0.33x_n+0.33x_{n-1}+0.33x_{n-2}$

19. შექმენით დაბალსიხშირული ფილტრი ზღვრული სიხშირით 1 კილოჰერცი, როცა ათლის სიხშირეა 8 კილოჰერცი. შეადარე ერთმანეთს 4 სტანდარტული IIR ფილტრის მახასიათებლები მე-8 რიგის შესაბამის ფილტრის მახასიათებლებს ერთიდაიგივე ნახაზზე.
20. შექმენით მაღალსიხშირული ფილტრი ზღვრული სიხშირით 500 ჰერცი, როცა ათლის სიხშირეა 1500 ჰერცი. შეადარე ერთმანეთს მე-8 რიგის ელიფსური და 32-ე რიგის FIR ფილტრის მახასიათებლები ერთიდაიგივე ნახაზზე.
21. შექმენით ზოლოვანი ფილტრი გატარების სიხშირით 300-4000 ჰერცი, როცა ათლის სიხშირეა 9.6 კილოჰერცი. შეადარე ერთმანეთს Butterworth და მე-8 მრიგის FIR ფილტრის მახასიათებლები ერთიდაიგივე ნახაზზე.
22. შექმენით ფილტრირომელიც არ გაატარებს სიხშირულ ზოლს 500-100 ჰც, როცა ათლის სიხშირეა 10 კილოჰერცი. შეადარე ერთმანეთს 12-ე რიგის ელიფსური და მე-12 რიგის ულე-ჭალკერ ფილტრის მახასიათებლები ერთიდაიგივე ნახაზზე.
23. შექმენით ფილტრი, რომელიც გამორიცხავს სიხშირეთა ზოლს 100-150 ჰც და 500-600ჰც, როცა ათლის სიხშირეა 2.5 კილოჰერცი. შეადარე ერთმანეთს IIR და FIR ფილტრის მახასიათებლები ერთიდაიგივე ნახაზზე.