



სსიპ სამცხე-ჯავახეთის სახელმწიფო უნივერსიტეტი
ინჟინერიის, აგრარულ და საბუნებისმეტყველო მეცნიერებათა ფაკულტეტი

ინფორმატიკის სადოქტორო პროგრამა

დოქტორანტი ჯულიეტა ტაბეშაძე

ელექტრულ წრედებში გამავალი სიგნალის აღდგენის ალგორითმები

ინფორმატიკის დოქტორის აკადემიური ხარისხის მოსაპოვებლად წარმოდგენილი
დისერტაცია

სამეცნიერო ხელმძღვანელი:
ვასილ ტაბატაძე, ფიზიკის დოქტორი, ასოცირებული პროფესორი

ახალციხე
2025 წელი

განაცხადი

მე, ჯულიეტა ტაბეშაძე, ვაცხადებ, რომ წარმოდგენილი სადისერტაციო ნაშრომი წარმოადგენს ჩემს ორიგინალურ ნამუშევარს და არ შეიცავს სხვა ავტორების მიერ აქამდე გამოქვეყნებულ, გამოსაქვეყნებლად მიღებულ ან დასაცავად წარდგენილ მასალას, რომელიც ნაშრომში არ არის მოხსენიებული ან ციტირებული სათანადო წესების შესაბამისად.

ხელმოწერა: 

თარიღი:

სარჩევი

ანოტაცია	5
შესავალი	10
საკვლევი პრობლემა და მისი აქტუალურობა (კვლევის მნიშვნელობა)	10
კვლევის მიზანი	13
საკვლევი საკითხები	14
ნაშრომის სიახლე	15
კვლევის მეთოდოლოგია;	18
წყაროებისა და სამეცნიერო ლიტერატურის მიმოხილვა.	29
I თავი. სიგნალები და სიგნალის სპექტრი.....	38
1.1. სიგნალის ცნება და კატეგორიები.....	38
1.2. ანალოგური, დისკრეტული და ციფრული სიგნალები.....	43
1.3. სიგნალის გენერირება	46
1.4. პერიოდული სიგნალების სპექტრი.....	53
1.5. პერიოდული ფუნქციები ფურიეს მწკრივებად გაშლა	55
1.6. არაპერიოდული სიგნალების სპექტრი	60
1.7. სიგნალის სიხშირის დიაპაზონი.....	64
1.8. სპექტოგრამა	66
II თავი. სიგნალის დისკრეტიზაცია და რეკონსტრუქცია.....	69
2.1. სიგნალის ციფრული დამუშავების ზოგადი სტრუქტურა	69
2.2. ანალოგურ-ციფრული გარდაქმნა	72
2.3. სიგნალის დამახინჯება, ზედდების ეფექტი	76
2.4. ფილტრი ზედდების ეფექტის საწინააღმდეგოდ	82
2.5. ანალოგური სიგნალიდან ნიმუშების აღების ინტერპოლაციის მეთოდები	89
III თავი. მთავარი მათემატიკური კონცეფციები	94
3.1. დისკრეტული სისტემა	94
3.2. Z გარდაქმნა.....	97
3.3. კონვოლუცია.....	103
3.4. კორელაცია	104
3.5. უკუ Z-გარდაქმნა	106
IV თავი - ციფრული ფილტრები სასრული იმპულსური მახასიათებლებით	108

4.1. ფილტრი სასრული იმპულსური მახასიათებლით (სიმ)	113
4.2. დაბალსიხშირული სიმ ფილტრის გენერირება, რიცხვითი ექსპერიმენტები MATLAB-ში და შედეგები, შედეგების ანალიზი	116
4.3. მაღალსიხშირული, ზოლური და ჩამჭრელი სიმ ფილტრების გენერირება, ექსპერიმენტი და შედეგები, შედეგების ანალიზი	129
V თავი - ციფრული ფილტრები უსასრულო იმპულსური მახასიათებლებით	135
5.1. ფილტრები უსასრულო იმპულსური მახასიათებლებით (უიმ).....	135
5.2. უიმ ფილტრის დიზაინი	141
5.3. უიმ ფილტრის გენერირება, რიცხვითი ექსპერიმენტი MATLAB-ში და შედეგები, შედეგების ანალიზი.....	143
რეკომენდაციები და იდეები მომავალი კვლევებისათვის.....	148
ციფრული ფილტრების პრაქტიკული გამოყენების იდეა ქართული ტექსტის ოპტიკურ ამომცნობში.....	157
დასკვნები და რეკომენდაციები.....	160
გამოყენებული ლიტერატურა:	164
გამოყენებული სურათების/დიაგრამების/გრაფიკების ნუსხა:.....	168
მნიშვნელოვანი განმარტებები:.....	172
აბრევიატურა	173

ანოტაცია

სადისერტაციო ნაშრომის კვლევის საგანს წარმოადგენს ელექტრონულ წრედებში გადაცემული სიგნალების აღდგენის ალგორითმების განვითარება და მათზე დაფუძნებული ციფრული დამუშავების მეთოდების ანალიზი.

კვლევის მიზანია მაღალი სიზუსტით, სტაბილურობითა და გამოთვლითი ეფექტიანობით გამორჩეული ალგორითმების შემუშავება, რომლებიც შესაძლებელს გახდის დამახინჯებული ან ნაწილობრივ დაკარგული სიგნალების აღდგენას სხვადასხვა საინჟინრო სისტემაში.

კვლევა ეფუძნება მათემატიკურ მოდელირებას, ციფრული სიგნალის დამუშავების თეორიასა და MATLAB-ის პლატფორმაზე ჩატარებულ რიცხვით სიმულაციებს. გამოყენებულია ანალოგურ-ციფრული გარდაქმნის მეთოდები, ფურიეს ანალიზი, Z-გარდაქმნა, ინტერპოლაცია, კონვოლუცია, კორელაცია და ციფრული ფილტრაცია (სასრული და უსასრულო იმპულსური მახასიათებლის (სიმ/უიმ) ტიპის ფილტრების ჩათვლით).

შემუშავებულია სიგნალის აღდგენის ისეთი ალგორითმები, რომლებიც მაღალი სიზუსტით აღადგენენ გადაცემულ სიგნალს ზედდების ეფექტის პირობებში. მიღწეულია დაბალსიხშირული და მაღალსიხშირული კომპონენტების ეფექტური გაფილტვრა MATLAB-ში გენერირებული ციფრული ფილტრების მეშვეობით. დადასტურდა, რომ რეკონსტრუქციის პროცესში ზუსტად შერჩეული ინტერპოლაციისა და ფილტრაციის მეთოდები მნიშვნელოვნად ამცირებს სიგნალის დამახინჯებას. წარმოდგენილია მრავალდონიანი სიგნალის აღდგენის ალგორითმები, რომლებიც თავსებადია მრავალარხიან სისტემებთან.

ნაშრომი ქმნის საფუძველს ქართული ტექსტის ოპტიკური ამომცნობ სისტემებში ციფრული ფილტრაციის ეფექტიანი ინტეგრაციისათვის. გამოყოფილია MATLAB-ის ინსტრუმენტების პრაქტიკული პოტენციალი ციფრული სიგნალის მოდელირებისა და ანალიზის მიმართულებით.

I თავი - სიგნალები და სიგნალის სპექტრი: თავში განხილულია სიგნალის ცნებები, მათ შორის ანალოგური, დისკრეტული, ასევე ციფრული სიგნალების კლასიფიკაცია და თვისებები. წარმოდგენილია სიგნალის გენერაციის მეთოდები მოდელირების

საფუძველზე და ფურიეს ანალიზის გამოყენება პერიოდული და არაპერიოდული სიგნალების სპექტრის გამოსაკვლევად. ასევე აღწერილია სიგნალის სიხშირითი დიაპაზონი და სპექტოგრამის ინტერპრეტაცია, რაც აძლიერებს ვიზუალურ აღქმას დროით-სიხშირით არეში.

II თავი - სიგნალის დისკრეტიზაცია და რეკონსტრუქცია: განხილულია ანალოგურ-ციფრული გარდაქმნის პრინციპები, ნაიქვისტ-შენონის თეორემა და ზედდების ეფექტისგან გამოწვეული დამახინჯებები. წარმოდგენილია ამ ეფექტის საწინააღმდეგოდ გამოყენებული ანტი ფილტრების დიზაინი და რეკონსტრუქციისთვის საჭირო ინტერპოლაციური მეთოდები, რაც უზრუნველყოფს სიგნალის ეფექტიან აღდგენას.

III თავი - მთავარი მათემატიკური კონცეფციები: მათემატიკური მოდელების ნაწილში ხდება დისკრეტული სისტემების ქცევის ანალიზი, მათ შორის Z-გარდაქმნა, უკუ Z-გარდაქმნა, კონვოლუცია და კორელაცია, რაც წარმოადგენს ციფრული სიგნალის დამუშავების ფუნდამენტურ მექანიზმებს. განხილულია მოდელების სტაბილურობა, წრფივობა და სიზუსტე.

IV თავი - ციფრული ფილტრები სასრული იმპულსური მახასიათებლით (სიმ): აღსანიშნავია სიმ ფილტრების სტაბილურობა და ხაზოვანი ფაზური რეაქცია, რის გამოც ისინი ხშირად გამოიყენება ციფრულ სისტემებში. MATLAB-ის დახმარებით გენერირებულია დაბალსიხშირული, მაღალსიხშირული, ზოლური და ჩამჭრელი სიმ ფილტრების მოდელები. რიცხვითი ექსპერიმენტები ადასტურებს მათი მორგების შესაძლებლობას კონკრეტულ სიგნალებზე.

V თავი - ციფრული ფილტრები უსასრულო იმპულსური მახასიათებლით (უიმ): უიმ ფილტრები განხილულია როგორც კომპაქტური და გამოთვლითად ეფექტიანი ალტერნატივა. აღწერილია მათი დიზაინის ალგორითმები, სიმულაცია MATLAB-ში და მიღებული შედეგების ანალიზი. აღნიშნულია მათი გამოყენების რისკები სტაბილურობის თვალსაზრისით და შესაბამისი შეზღუდვები.

რეკომენდაციები და მომავალი კვლევის პერსპექტივები: ნაშრომში წარმოდგენილია რეკომენდაციები ალგორითმების რეალურ დროში იმპლემენტაციის, ადაპტური ფილტრაციის დანერგვისა და ოპტიკური ტექსტის ამომცნობ სისტემებში გამოყენების

შესაძლებლობებზე, განსაკუთრებით ქართულენოვანი რესურსების გაუმჯობესების მიზნით.

მნიშვნელოვანი განმარტებები: დასასრულს წარმოდგენილია ტერმინთა განმარტებები, რაც ამარტივებს ნაშრომის გააზრებას და ხელს უწყობს ერთიან, აკადემიურად დამუშავებულ კვლევით ჩარჩოს ჩამოყალიბებას.

ნაშრომი წარმოადგენს მნიშვნელოვან წვლილს ციფრული სიგნალის დამუშავების სფეროში და იძლევა როგორც თეორიულ, ასევე პრაქტიკულ რეკომენდაციებს მისი გამოყენებისათვის სხვადასხვა ტექნოლოგიურ დარგებში.

ABSTRACT

Julieta Tabeshadze

Algorithms of reconstruction of the signals transmitted in the electrical network

The subject of this dissertation research is the development of signal reconstruction algorithms for electronic circuits and the analysis of digital processing methods based on them.

The aim of the research is to develop algorithms distinguished by high accuracy, stability, and computational efficiency, which will enable the reconstruction of distorted or partially lost signals in various engineering systems.

The research is based on mathematical modeling, digital signal processing theory, and numerical simulations conducted on the MATLAB platform. Methods of analog-to-digital conversion, Fourier analysis, Z-transform, interpolation, convolution, correlation, and digital filtering (including Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) type filters) are utilized.

Signal reconstruction algorithms have been developed that accurately restore transmitted signals under superposition effects. Effective filtering of low-frequency and high-frequency components has been achieved through digital filters generated in MATLAB. It has been confirmed that precisely chosen interpolation and filtering methods significantly reduce signal distortion during the reconstruction process. Multilevel signal reconstruction algorithms compatible with multi-channel systems are also presented.

The work lays the foundation for the efficient integration of digital filtering into optical character recognition (OCR) systems for Georgian text. The practical potential of MATLAB tools for digital signal modeling and analysis is also highlighted.

Chapter I – Signals and Signal Spectrum: Discusses the classification and characteristics of analog, discrete, and digital signals. It covers signal generation methods using modeling techniques, and the application of Fourier analysis for studying periodic and aperiodic signal spectra. The chapter also addresses frequency ranges and spectrogram interpretation, enhancing time-frequency domain comprehension.

Chapter II – Signal Discretization and Reconstruction: Explores the principles of analog-to-digital conversion, the Nyquist-Shannon sampling theorem, and distortions caused by aliasing.

It presents anti-aliasing filter designs and interpolation methods required for accurate signal reconstruction.

Chapter III – Core Mathematical Concepts: Analyzes the behavior of discrete systems through Z-transform, inverse Z-transform, convolution, and correlation—key mechanisms in digital signal processing. The chapter evaluates the stability, linearity, and precision of modeled systems.

Chapter IV – Digital Filters with Finite Impulse Response:

Highlights FIR filters for their stability and linear phase response, making them suitable for digital systems. Various FIR filter models (low-pass, high-pass, band-pass, and notch) are generated and tested using MATLAB, with numerical experiments confirming their adaptability to specific signal types.

Chapter V – Digital Filters with Infinite Impulse Response: Presents IIR filters as a compact and computationally efficient alternative. It discusses their design algorithms, MATLAB-based simulations, and result analysis, while also addressing potential stability risks and constraints.

Recommendations and Future Research Perspectives: The dissertation outlines directions for real-time implementation of algorithms, the integration of adaptive filtering, and their application in optical text recognition systems, with particular attention to enhancing Georgian-language digital tools.

Terminological Clarification: A glossary of key terms is included to facilitate understanding and support the formation of a coherent academic research framework.

This study makes a valuable contribution to digital signal processing by providing both conceptual understanding and actionable guidance for its implementation across diverse technological fields.

შესავალი

საკვლევი პრობლემა და მისი აქტუალურობა (კვლევის მნიშვნელობა)

სიგნალის ციფრული დამუშავება (სცდ) არის კვლევის ის სფერო, რომელიც ფოკუსირებულია ციფრული სიგნალის მანიპულირებაზე, კონკრეტული სიგნალიდან უფრო სასურველი თვისების მქონე სიგნალის მიღებაზე, მისგან ინფორმაციის ამოღებაზე ან ამოღებული ინფორმაციის გაუმჯობესებაზე (Oppenheim 2009, 1-5). სიგნალის თვისებები, როგორცაა ამპლიტუდა, ფაზა, სიხშირული სპექტრი შესაძლებელია დამუშავების შემდეგ შეიცვალოს კონკრეტული მოთხოვნის დასაკმაყოფილებლად. არსებობს უწყვეტი სიგნალები ე.წ. ანალოგური სიგნალები და მისი დამუშავების მეთოდები, თუმცა ცსდ ეხება მხოლოდ დისკრეტული სიგნალის დამუშავებას, დისკრეტული სიგნალი კი მიიღება ანალოგური სიგნალისგან. სიგნალის დისკრეტული ბუნება გვადლევს საშუალებას მასზე ჩავატაროთ ცდები, დავაკვირდეთ შედეგებს და მიღებული შედეგები გამოვიყენოთ სხვადასხვა სფეროს განსავითარებლად.

ადრე მეცნიერები ანალოგური სიგნალების დამუშავებას ცდილობდნენ რეზისტორების, კონდენსატორების, ინდუქტორების, ტრანზისტორების, დიოდების და სხვა მოწყობილობების საშუალებით (Proakis 2006, 10-15), თუმცა დისერტაციაში სიგნალების დამუშავება ხდება პროგრამული უზრუნველყოფის გამოყენებით. უწყვეტი ანალოგური სიგნალი ანალოგურ-ციფრულ გარდაქმნილებში გავლის შემდეგ გადაკეთდება ჩვეულებრივი 0 და 1 ციფრების თანმიმდევრობად, შემდეგ კომპიუტერის პროცესორი ახდენს სიგნალზე მანიპულირებას, სასურველი შედეგის მიღებამდე. ამ დროს შესაძლოა შეიცვალოს მწკრივში ციფრთა თანმიმდევრობა ან რაოდენობა. შემდეგ ციფრულ-ანალოგურ გარდამქმნელში ციფრულ სიგნალი კვლავ ანალოგურ სიგნალად გადაკეთდება.

სიგნალების თეორიული კვლევის და ანალიზის საშუალებას გვადლევს მათი მათემატიკური აღწერის გზების მოძიება, ანუ სიგნალების მათემატიკური მოდელების შექმნა. სიგნალის მათემატიკური მოდელი, როგორც წესი, წარმოადგენს ფუნქციონალურ დამოკიდებულებას, სადაც არგუმენტი დროა. ასეთი მოდელები ხშირად აღინიშნება ლათინური ასოებით, როგორცაა $f(t)$, $u(t)$ და ა.შ.

სიგნალის მათემატიკური მოდელი ყურადღებას ამახვილებს სიგნალის ძირითად მახასიათებლებზე და უგულებელყოფს მეორეხარისხოვან ფაქტორებს. ხშირად რთულია ზუსტად მოიძებნოს ისეთი ფუნქციონალური კავშირები, რომლებიც სრულად ასახავს ექსპერიმენტულ მონაცემებს. ამიტომ, ვირჩევთ მათემატიკური მოდელების იმ მეთოდს, რომელიც ყველაზე უკეთ და მარტივად აღწერს ფიზიკურ პროცესს მოცემულ სიტუაციაში.

სიგნალების ციფრული დამუშავების გარიჟრაჟზე პროცესორები იყო საკმაოდ ნელი, აპლიკაციები იყო შეზღუდული, პროგრამირების ენები ნაკლებად განვითარებული. დღეს კი ჩვენ გვაქვს ძალიან სწრაფი და ენერგოეფექტური პროცესორები, რომელმაც მკვეთრად გაზარდა ციფრული სიგნალის დამუშავების აპლიკაციების რაოდენობაც, ხარისხიც და ა.შ.

ანალოგური სიგნალების დამუშავების ძირითადი ნაკლოვანება არის შეზღუდული კომპიუტერული პროგრამების, აპლიკაციების რაოდენობა, რომლებსაც შეუძლიათ ანალოგურ სიგნალებთან მუშაობა. სისტემები, რომლებსაც შეუძლიათ ანალოგური სიგნალის დამუშავება ძვირადღირებულია. სცდ - მიდგომის გამოყენებით შესაძლოა იაფფასიანი პერსონალური კომპიუტერები გარდაიქმნას ძლიერ სიგნალების დამმუშავებლად.

სცდ-ს ძირითადი უპირატესობები:

- **სიზუსტე:** სცდ უზრუნველყოფს სიგნალის დამუშავების ამოცანაზე ზუსტ კონტროლს, რაც იძლევა მონაცემთა ზუსტ წარმოდგენას და ტრანსფორმაციის საშუალებას. ციფრული სიგნალის დისკრეტული ბუნება იძლევა მოდიფიცირების საშუალებას, რაც აუცილებელი პირობაა ამოცანის შესასრულებლად.

- **მოქნილობა:** ციფრული სისტემის კონფიგურაცია და განახლება მარტივია პროგრამული კოდის ცვლილების მეშვეობით. ეს მოქნილობა იძლევა სიგნალის დამუშავების ალგორითმებს დინამიური კორექტირების საშუალებას, ფიზიკური ცვლილებების გარეშე. ოპერაციები შესაძლოა მარტივად შეიცვალოს რეალურ დროში, მარტივი პროგრამული ცვლილებით ან სხვა ხერხებით: გადატვირთვა, რეგისტრის ცვლილება და ა.შ.

- **სიიფე:** ციფრული სისტემები ხშირად ამცირებენ ძვირადღირებული ანალოგურ მოწყობილობებს. კომპიუტერისა და სპეციალური პროგრამული უზრუნველყოფის სცდ-ს შეუძლია მიაღწიოს მაღალი ხარისხის დამუშავებას დაბალ

ფასად, ტრადიციულ სისტემებთან შედარებით. სცდ მიდგომა შესაძლოა განვითარდეს პროგრამული უზრუნველყოფის საშუალებით ზოგადი დანიშნულების კომპიუტერებზე, როგორც ეს ჩვენს შემთხვევაში ხდება. გარდა ამისა სცდ-მოსახერხებელია ტესტირებისთვის და პროგრამული უზრუნველყოფა პორტატულია, რადგან იყენებს დაბალი ღირებულების ტექნოლოგიებს, რომელიც ამცირებს მქსირების ბარათების, მიკროპროცესორების, ჩამრთველების და ა.შ. ხარჯს.

სცდ-ის ერთადერთი უარყოფითი მხარე შეიძლება იყოს ოპერაციების შესრულების სიჩქარე, გამომდინარე იმ აპარატურიდან, რომელსაც ვიყენებთ. ეს თვალშისაცემია განსაკუთრებით მაღალ სიხშირეებზე.

უფრო მეტი უპირატესობების გამო სცდ პირველი არჩევანია ბევრ ტექნოლოგიასა და პროგრამულ უზრუნველყოფაში, როგორცაა სამომხმარებლო ელექტრონიკა, კომუნიკაცია, უკაბელო ტელეფონები და სამედიცინო აპარატურა (რენტგენოლოგია) და ა.შ. (Ifeachor 2020, 40-45)

დღესდღეობით სიგნალების ციფრულ დამუშავებას იყენებენ სხვადასხვა მიმართულებებში, ესენია:

- **კომუნიკაცია:** საკომუნიკაციო სისტემებში სიგნალის ციფრული დამუშავება გადამწყვეტ როლს თამაშობს სიგნალის კოდირებასა და დეკოდირებისთვის. ის უზრუნველყოფს მონაცემთა ეფექტურ გადაცემას, მიღებული სიგნალის ხარისხის გამჯობესებას, შეცდომების შემცირების ხარჯზე.
- **აუდიო დამუშავება:** სიგნალის ციფრულ დამუშავებას ვიყენებთ აუდიო დამუშავების პროგრამულ უზრუნველყოფაში. ხშირად აუდიო ფაილი შეიცავს ხმაურსაც და ექოსაც. დამუშავების შემდეგ უმჯობესდება მისი ხარისხი.
- **გამოსახულების დამუშავება:** სიგნალის ციფრული დამუშავება გამოიყენება გამოსახულების ხარისხის გასაუმჯობესებლად: გამოსახულებას ემატება დეტალების სიცხადე, უზრუნველყოფს შენახვის ეფექტიანობას.
- **ავტომატიზებული სისტემების მართვა:** სცდ მნიშვნელოვან როლს ასრულებს კონტროლის სისტემებში, ხელს უწყობენ დინამიური სისტემის მართვას, მუშაობის ოპტიმიზაციას და სტაბილურობის უზრუნველყოფას. გამოიყენება სამრეწველო სისტემების მართვაში.

სცდ-მ ბოლო ათწლეულის განმავლობაში მოიცვა მრავალი სფერო, ესენია:

- საუბარი/აუდიო (ხმის ამოცნობა/სინთეზი,ციფრული აუდიო, ეკვალიზაცია და ა.შ).
- გამოსახულება/ვიდეო (გაფართოება, კოდირება და მისი გადაცემა, რობოტექნიკა, ანიმაციები...)
- სამხედრო საქმე/სივრცეში კოორდინაცია (რადარების მიერ ინფორმაციის დამუშავება, უსაფრთხო კომუნიკაცია, სარაკეტო სისტემები)
- ბიომედიცინა/ჯანდაცვა (სკანერები, რენტგენი, ბიო სიგნალები)
- სამომხმარებლო ელექტრონიკა (ფიჭური / მობილური ტელეფონები, ციფრული ტელევიზია, ციფრული კამერა, ინტერნეტი ხმა / მუსიკა / ვიდეო, ინტერაქტიული გასართობი სისტემები, და ა.შ.) და მრავალი სხვა.

ყველა ამ სფეროს განვითარება მოითხოვს ურთიერთდაკავშირებულ კომპლექსურ ნაბიჯებს, როგორცაა ინფორმაციის შეგროვება, დამუშავება, გადაცემა, აუდიო გამოსახულების გადაცემა რეალურ დროში და ა.შ. ჩვენ შეგვიძლია გამოვიყენოთ ეს შესაძლებლობები თანამედროვე ტექნოლოგიების განვითარებისათვის და გავხადოთ ისინი ინოვაციური და ხელმისაწვდომი.

კვლევის მიზანი

სიგნალების თეორია და სიგნალის აღდგენის მათემატიკური ალგორითმები თანამედროვე ტექნოლოგიებში მნიშვნელოვან როლს თამაშობს, რადგან ის დაკავშირებულია მონაცემთა გადაცემის, შენახვისა და დამუშავების პროცესებთან. არსებული მეთოდების დახვეწა და ახალი ალგორითმების შექმნა შესაძლებლობას გვაძლევს უფრო ზუსტი, ეფექტიანი და მედეგი სისტემები შევქმნათ. ამ კვლევის მიზანია ციფრული სიგნალის აღდგენის ალგორითმების განვითარება და გაუმჯობესება, რაც ხელს შეუწყობს როგორც ტექნოლოგიური, ისე კომერციული გამოყენების სფეროებში მიღწევების გაზრდას.

სადისერტაციო ნაშრომში ჩატარებული კვლევის მიზნებია:

1. **სიგნალის აღდგენის ხარისხის გაუმჯობესება:** არსებული ალგორითმების გაუმჯობესება ისე, რომ აღდგენის პროცესი იყოს მეტად შედეგიანი და აღდგენილი სიგნალი რეალურ მონაცემებთან იყოს მაქსიმალურად მიახლოებული.

2. **გამოთვლითი სირთულის შემცირება:** ეფექტური ალგორითმების შემუშავება, რომლებიც სიგნალის აღდგენის პროცესს განახორციელებენ მცირე რესურსების გამოყენებით, დაიხარჯება ნაკლები დრო და ენერგია.

3. **დამახინჯებული და დაკარგული სიგნალის აღდგენა:** ისეთი მეთოდების და ალგორითმების შესწავლა, რომლებიც შეძლებენ დაკარგული ან დაზიანებული სიგნალის აღდგენას, სადაც ჩვეულებრივი მეთოდები ვერ მუშაობს.

4. **მრავალდონიანი სიგნალების აღდგენა:** მეთოდების შემუშავება, რომლებიც შეძლებენ სხვადასხვა წყაროებიდან მიღებული სიგნალების ერთდროულად და თანმიმდევრულად აღდგენას.

5. **მოსახერხებელი და ინტუიციური ალგორითმების შემუშავება:** კვლევა, თუ როგორ შეიძლება სიგნალის აღდგენის ალგორითმების გამარტივება, რათა მომხმარებლებისთვის იყოს უფრო ინტუიციური და მარტივი.

საკვლევი საკითხები

გადაცემის ფუნქცია და ციფრული ფილტრები, რომელიც სადისერტაციო ნაშრომშია წარმოდგენილი, ციფრული სიგნალის დამუშავების სისტემების ფუნდამენტური კომპონენტებია. მათი ეფექტური დიზაინი და განხორციელება გადამწყვეტ როლს თამაშობს დავადგინოთ თუ როგორ რეაგირებს სისტემა სხვადასხვა შემავლ სიგნალზე, რაც პირდაპირ გავლენას ახდენს სისტემის მუშაობასა და ფუნქციონირებაზე. რიცხვითი ექსპერიმენტების შედეგად გადაცემის ფუნქციის და ციფრული ფილტრების გაუმჯობესებამ შეიძლება გამოიწვიოს მნიშვნელოვანი წინსვლა და მოგვიტანოს სარგებელი აპლიკაციების ფართო სპექტრში.

გადაცემის ფუნქცია და ციფრული ფილტრების მნიშვნელობა

- **გადაცემის ფუნქცია:** გადაცემის ფუნქცია აღწერს მათემატიკურ კავშირებს სისტემაში შემავლ და გამომავალ სიგნალის რიცხვით მახასიათებლებს შორის, მას გადამწყვეტი მნიშვნელობა აქვს:

➤ **სისტემის მოდელირებისთვის:** გადაცემის ფუნქცია უზრუნველყოფს დროში ინვარიანტული და დინამიური სისტემის მოდელირებას.

➤ **სისტემის მახასიათებლების ანალიზისთვის:** გადაცემის ფუნქციის ანალიზი მკვლევარებს აძლევს საშუალებას მიიღონ ინფორმაცია სისტემის სისტემის სტაბილურობაზე, სიხშირულ მახასიათებლებზე, გამტარუნარიანობის ხარისხზე.

➤ **სისტემის დიზაინისთვის:** გადაცემის ფუნქცია ხელს უწყობს სასურველი მახასიათებლების მქონე სისტემების კონსტრუირებას, რაც საშუალებას აძლევს მკვლევარებს წინასწარ განსაზღვრონ, თუ როგორ იმოქმედებს სისტემის პარამეტრების ცვლილება შემავალ და გამომავალ სიგნალზე.

- **ციფრული ფილტრები:** ციფრული ფილტრები საჭიროა სიგნალის შესაცვლელად, კონკრეტული საჭიროების მიხედვით. მათი ძირითადი დანიშნულებაა

- **არასასურველი კომპონენტების მოცილება:** ციფრულ ფილტრებს შეუძლიათ შეასუსტონ ხმაური, მოაცილონ სიგნალს არასასურველი კომპონენტები.

- **სიგნალის მახასიათებლების გაძლერება:** ციფრული ფილტრები აძლიერებენ სიგნალის სპეციფიკურ მახასიათებლებს, როგორცაა მაგალითად გამოსახულების დეტალების სიმკვეთრე, აუდიო ხარისხის გაუმჯობესება და ა.შ.

ნაშრომის სიახლე

ნაშრომს გააჩნია ინოვაციური და პრაქტიკული მნიშვნელობა თანამედროვე ციფრული სიგნალის დამუშავების სფეროში, კერძოდ:

ალგორითმების გაუმჯობესება და ოპტიმიზაცია: ნაშრომი წარმოადგენს ახალი და გაუმჯობესებული ალგორითმების ნაკრებს, რომლებიც მიზნად ისახავს სიგნალის აღდგენის სიზუსტისა და ეფექტურობის გაზრდას. ეს ალგორითმები განსხვავდება ტრადიციული მეთოდებისგან იმით, რომ ისინი სპეციალურადაა შემუშავებული ზედდების ეფექტისა და ხმაურის პირობებში რთული სიგნალების დამუშავებისთვის. მაგალითად, ნაშრომში გამოყენებულია მოწინავე ინტერპოლაციის მეთოდები და ოპტიმიზებული ციფრული ფილტრები (სასრული იმპულსური მახასიათებლის (სიმ) და უსასრულო იმპულსური მახასიათებლის (უიმ) ფილტრები), რაც მნიშვნელოვნად ამცირებს სიგნალის დამახინჯებას. MATLAB-ის სიმულაციებით დადასტურებულია,

რომ ეს ალგორითმები ამცირებენ გამოთვლით სირთულეს 10-15%-ით, ხოლო სიზუსტე 5-7%-ით მაღალია, ვიდრე ჩვეულებრივი მეთოდების შემთხვევაში.

ინოვაციური მეთოდები დაზიანებული სიგნალების აღსადგენად: ნაშრომის ერთ-ერთი მთავარი სიახლეა მეთოდების შემუშავება, რომლებიც საშუალებას იძლევა ეფექტურად აღდგეს მძიმედ დამახინჯებული ან ნაწილობრივ დაკარგული სიგნალები. ეს განსაკუთრებით მნიშვნელოვანია ისეთ სისტემებში, სადაც სიგნალი ექვემდებარება მაღალი ხმაურის ან ჩარევის გავლენას, მაგალითად, საკომუნიკაციო ქსელებში ან ბიომედიცინურ მოწყობილობებში (ECG/EEG ანალიზი). ნაშრომი იყენებს ფურიეს ანალიზის, Z-გარდაქმნისა და კონვოლუციის კომბინაციას, რათა შეიქმნას მედეგი სისტემა, რომელიც უზრუნველყოფს სიგნალის აღდგენას მაღალი სიხშირის ან რთული დამახინჯებების პირობებში. MATLAB-ის ექსპერიმენტებმა აჩვენა, რომ ამ მეთოდებით შესაძლებელია 20%-მდე გაუმჯობესება სიგნალის აღდგენის ხარისხში, განსაკუთრებით ხმაურიან გარემოში, სადაც ტრადიციული მეთოდები ნაკლებად ეფექტურია.

მრავალდონიანი სიგნალების ერთდროულად აღდგენა: ნაშრომი გვთავაზობს მრავალდონიანი სიგნალის აღდგენის მეთოდებს, რომლებიც საშუალებას იძლევა სხვადასხვა წყაროდან მიღებული სიგნალების ერთდროულად და თანმიმდევრულად დამუშავება. ეს მიდგომა განსაკუთრებით ღირებულია თანამედროვე საინჟინრო სისტემებში, სადაც მრავალარხიანი მონაცემთა დამუშავება სულ უფრო მნიშვნელოვანი ხდება (მაგ., უკაბელო კომუნიკაციები, სამედიცინო დიაგნოსტიკა). ალგორითმები შექმნილია ისე, რომ თავსებადია მრავალარხიან სისტემებთან, რაც უზრუნველყოფს მონაცემთა ინტეგრირებასა და სტაბილურობას

მოხერხებული და ინტუიციური ალგორითმების შემუშავება: ნაშრომი განსაკუთრებულ ყურადღებას უთმობს ალგორითმების გამარტივებას, რათა ისინი ხელმისაწვდომი იყოს მომხმარებლებისთვის, რომლებსაც შესაძლოა არ ჰქონდეთ ღრმა ტექნიკური ცოდნა. ეს მიიღწევა MATLAB-ის ინტუიციური ინტერფეისისა და ხელსაწყოების (მაგ., Signal Processing Toolbox) გამოყენებით, რაც ამარტივებს ალგორითმების ტესტირებასა და განხორციელებას. მაგალითად, ნაშრომში წარმოდგენილია მარტივი პროგრამული კოდი, რომელიც საშუალებას აძლევს

მომხმარებლებს სწრაფად განახორციელონ ფილტრაციისა და ინტერპოლაციის პროცესები. ეს მიდგომა განსაკუთრებით სასარგებლოა განვითარებადი ტექნოლოგიური ეკოსისტემებისთვის, როგორცაა საქართველო, სადაც ტექნიკური რესურსები შეიძლება იყოს შეზღუდული.

MATLAB-ის გამოყენება მოდელირებისთვის: ნაშრომი ხაზს უსვამს MATLAB-ის ხელსაწყოების (Signal Processing Toolbox, Filter Design Toolbox) პრაქტიკულ პოტენციალს ციფრული სიგნალის მოდელირებასა და ანალიზში. MATLAB-ის გამოყენებით, ნაშრომი უზრუნველყოფს მოქნილ და ზუსტ სიმულაციებს, რომლებიც საშუალებას იძლევა შეისწავლოს სიგნალის დამუშავების სხვადასხვა ეტაპი, მათ შორის სიხშირული ანალიზი, ფილტრის დიზაინი და რეკონსტრუქცია. ეს მიდგომა განსაკუთრებით ღირებულია ქართულ აკადემიურ გარემოში, სადაც MATLAB-ის ხელმისაწვდომობა ხელს უწყობს კვლევების გამარტივებას.

გადაცემის ფუნქციის და ციფრული ფილტრების ეფექტურობის გაუმჯობესება: ნაშრომი ყურადღებას ამახვილებს გადაცემის ფუნქციისა და ციფრული ფილტრების ეფექტურობის გაუმჯობესებაზე, რაც გადამწყვეტია სისტემის სტაბილურობისა და სიხშირული მახასიათებლების ოპტიმიზაციისთვის. განსაკუთრებით, სიმ ფილტრების ფაზური სტაბილურობა და უიმ ფილტრების გამოთვლითი ეფექტურობა განხილულია MATLAB-ის სიმულაციების საფუძველზე, რაც აჩვენებს, რომ შემოთავაზებული ფილტრები 8-10%-ით აღემატება ტრადიციულ მოდელებს სიგნალის ხარისხისა და სტაბილურობის თვალსაზრისით

ქართული ენის ოპტიკური ამომცნობი (OCR) სისტემებისთვის გამოყენება:

ნაშრომის ერთ-ერთი ყველაზე მნიშვნელოვანი სიახლეა ციფრული ფილტრაციის მეთოდების ინტეგრირება ქართული ენის ოპტიკური ამომცნობი სისტემების გაუმჯობესებისთვის. ქართული დამწერლობა, მოითხოვს სპეციალიზებულ მიდგომებს ციფრულ ფორმატში გადასაყვანად. ნაშრომი გვთავაზობს ალგორითმს, რომელიც მოიცავს გამოსახულების წინასწარ დამუშავებას (rgb2gray ფუნქციით), სეგმენტაციასა და სიმბოლოების ამოცნობას სიმ/უიმ ფილტრების გამოყენებით. ეს მეთოდი აუმჯობესებს ამოცნობის სიზუსტეს 10-12%-ით, განსაკუთრებით დაბალი ხარისხის ან ხმაურიანი გამოსახულებების შემთხვევაში, რაც მნიშვნელოვანი

წინსვლაა არსებულ უფასო OCR სისტემებთან შედარებით, რომლებიც ხშირად შეზღუდულია მაღალი გარჩევადობის მოთხოვნებით.

კვლევის მეთოდოლოგია

დისერტაციაში გამოყენებულია ციფრული სიგნალის დამუშავების მათემატიკური მოდელირება MATLAB-ში. ზოგადად მოდელირება გულისხმობს აბსტრაქტული, კონცეპტუალური, გრაფიკული ან მათემატიკური მოდელის შექმნას (ყუბანიშვილი 2013, 8-22). მეცნიერებაში მრავალი მეთოდი, ტექნიკა და თეორია გამოიყენება მოდელირების პროცესში. მოდელირება არის თანამედროვე სამეცნიერო საქმიანობის უმნიშვნელოვანესი ნაწილი, რომელსაც თითოეული სამეცნიერო სფერო თავისებურად ახორციელებს.

მოდელი არის ობიექტის, მოვლენის ან ფიზიკური პროცესის, ჩვენს შემთხვევაში სიგნალების, ემპირიული წარმოდგენა, რომელიც შეიქმნება ლოგიკური და ობიექტური მიდგომით. მოდელი წარმოადგენს რეალობის გამარტივებულ ვერსიას, თუმცა, მიუხედავად იმისა, რომ ის გამარტივებულია, მაინც მნიშვნელოვან ინფორმაციას გვაწვდის. მოდელის აგება და მისი სიმულაცია წარმოადგენს მნიშვნელოვან კომპონენტს თანამედროვე მეცნიერებაში (ფაღავა 2022, 25-30).

მოდელირება გამოიყენება მაშინ, როდესაც ექსპერიმენტული პირობების შექმნა შეუძლებელია ან არაპრაქტიკულია. ჩვეულებრივ, კონტროლირებად პირობებში ჩატარებული ექსპერიმენტები უფრო ზუსტ შედეგებს იძლევა, ვიდრე მოდელირებისას მიღებული მონაცემები. თუმცა, მოდელირებისას ხშირად ხდება გარკვეული შეზღუდვების გამოყენება, რაც აუცილებელია რიცხვითი ექსპერიმენტის ჩატარების პროცესში.

სიმულაცია გულისხმობს მოდელის რეალურ პირობებში განხორციელებას. სტატიკური სიმულაცია იძლევა ინფორმაციას დროის კონკრეტულ მომენტში, ხოლო დინამიკური სიმულაცია აფასებს მოვლენებს დროში. სიმულაცია საშუალებას გვაძლევს ვნახოთ, როგორ იქცევა სიგნალი ან როგორ ვითარდება სიგნალის წარმოქმნის, დამუშავების პროცესი დროში. იგი გამოიყენება ტესტირების და

ანალიზისათვის, როდესაც რეალური სისტემის ან კონცეფციის მოდელირების საშუალება არსებობს.

მოდელი ხშირად მოიცავს მხოლოდ მოვლენის გარკვეულ ასპექტებს და შესაძლებელია ორი სხვადასხვა მოდელი სრულიად განსხვავდებოდეს ერთმანეთისგან. მოდელის შექმნისას აუცილებელია კარგად განვსაზღვროთ მოდელირების მიზანი და ის შეზღუდვები, რომლებიც გამოყენებული იქნება მოდელირების პროცესში.

კომპიუტერი მნიშვნელოვანი ინსტრუმენტია საინჟინრო პრობლემების გადაჭრისთვის. დისერტაციაში მოდელირებისთვის გამოყენებულია MATLAB-ი, რომელიც არის მძლავრი გამოთვლითი ინსტრუმენტი, რომელიც ფართოდ გამოიყენება სიგნალების ციფრულ დამუშავებაში, შესაბამისი ხელსაწყოებისა და მრავალფუნქციური ბიბლიოთეკების გამო. ის უზრუნველყოფს სიგნალის დამუშავების ალგორითმის დიზაინს, სიმულაციისა და ანალიზის შესაძლებლობას. MATLAB-ის მახასიათებლები მოიცავს (MathWorks-ის მომხმარებლის სახელმძღვანელო 2024, 10-15):

- **ჩაშენებული ფუნქციები:** MATLAB გვთავაზობს ფართო სპექტრის ჩაშენებულ ფუნქციასთა სპექტრს სიგნალების დამუშავების ამოცანებისთვის, ესენია: ფილტრაცია, ფურიეს გარდაქმნა, სიგნალის სტატისტიკური ანალიზი და ა.შ.
- **მრავალფეროვანი ხელსაწყოები:** სპეციალური ხელსაწყოები, როგორცაა signal Processing Toolbox და Control system Toolbox უზრუნველყოფენ სიგნალების ციფრული დამუშავების ფართო სპექტრს.
- **ვიზუალიზაცია:** MATLAB-ის ვიზუალიზაციის ძლიერი შესაძლებლობები საშუალებას გვაძლევს შევექმნათ დეტალური ნახაზები და გრაფიკები, სიგნალის მახასიათებლებისა და დამუშავების შედეგების გასაანალიზებლად.
- **გამოყენების სიმარტივე:** MATLAB-ის მოსახერხებელი ინტერფეისი და პროგრამირების გარემო ხელს გვიწყობს სწრაფად ავაგოთ მოდელი და ჩავატაროთ ექსპერიმენტები ცსდ-ის ალგორითმებით.

MATLAB არის პროგრამული გარემო და პროგრამირების მაღალი დონის ენა, რომელიც გამოიყენება ტექნიკური გამოთვლებისთვის. ის აერთიანებს გამოთვლების, გრაფიკის და პროგრამირების შესაძლებლობებს ერთ პლატფორმაში.

კომპიუტერული ალგებრა წარმოადგენს ინფორმატიკაში განვითარებულ ახალ მიმართულებას, რომელიც მიზნად ისახავს უნივერსალური პროგრამული სისტემების შექმნას. ამ მხრივ, MATLAB არის ერთ-ერთი გამორჩეული ინსტრუმენტი, როგორცაა Mathematica, Maple, Mathcad და სხვები. ყველა ამ სისტემას აქვს საკუთარი ენა და უნიკალური შესაძლებლობები, მათ შორის რთული მათემატიკური ამოცანების ამოხსნის მეთოდები და ალგორითმები.

MATLAB გამოირჩევა რამდენიმე მახასიათებლით:

1. **მატრიცებზე ორიენტირებული ოპერაციები:** MATLAB ოპტიმიზირებულია მატრიცულ გამოთვლებზე, რაც მას ეფექტურ საშუალებას ხდის მეცნიერული კვლევებისა და ინჟინერიული ამოცანების გადაწყვეტაში.

2. **ფუნქციათა ფართო ბიბლიოთეკა:** პროგრამა მოიცავს მრავალ ფუნქციას, რომლებიც განკუთვნილია სხვადასხვა მეცნიერების და ინჟინერიის სფეროში ამოცანების შესასრულებლად, როგორცაა მართვის ანალიზი, სისტემების სინთეზი, რიცხვითი ექსპერიმენტის დაგეგმვა და სხვა.

3. **ინტეგრაცია სხვა სისტემებთან:** MATLAB უზრუნველყოფს ინტეგრაციას სხვა მათემატიკურ პროგრამებთან, როგორცაა Maple, Mathcad, Ms Excel, რაც ამდიდრებს მის ფუნქციონალურობას და ამცირებს გარკვეულ შეზღუდვებს.

ამ გამორჩეულობის გამო, MATLAB არის ერთ-ერთი ყველაზე ფართოდ გავრცელებული მათემატიკური პროგრამული უზრუნველყოფა, რომელიც დიდი პოპულარობით სარგებლობს მკვლევარებში. ლაბორატორიული სამუშაოების შესრულებისას, როგორც ზოგად ტექნიკურ, ასევე სპეციალურ დისციპლინებში, მისი გამოყენება გახშირდა.

MATLAB-ის და მისი დამატებითი მოდულების (Toolbox-ების) გამოყენება შესაძლებელს ხდის მრავალი პრობლემის, მათ შორის მატრიცული ანალიზის, შესწავლას. პლატფორმა იძლევა შესაძლებლობას სწრაფად შესრულდეს ოპერაციები ვექტორებსა და მატრიცებზე, როგორცაა მათი გამრავლება, შეკრება, დეტერმინანტის

გამოთვლა, საკუთარ რიცხვებსა და ვექტორებზე ოპერაციები, აგრეთვე ალგებრული, ტრიგონომეტრიული, ექსტრემალური ტრიგონომეტრიული, ლოგარითმული და ექსპონენციალური ფუნქციების მნიშვნელობების გამოთვლა, გამოსახულებების დამუშავება და შენახვა, განტოლებებისა და სისტემების ამოხსნა, ოპერაციები პოლინომებზე, ყველა ფესვის გამოთვლა, ფუნქციების ინტეგრირება, დიფერენცირება, დიფერენციალური განტოლებებისა და სისტემების ამოხსნა, მათემატიკური ფიზიკის ამოცანების და ოპტიმიზაციის ამოცანების ამოხსნა, კარტოგრაფიული მონაცემების დამუშავება, ასევე MATLAB-ის შესაძლებლობები მოიცავს ისეთ მნიშვნელოვან სფეროებს, როგორებიცაა ნეირონული ქსელების მოდელირება, დინამიკური პროცესების ანალიზი, მონაცემთა დამუშავება და ვიზუალიზაცია, აგრეთვე სიგნალების ციფრული დამუშავებისთვის მოსახერხებელი ინსტრუმენტების გამოყენება. MATLAB წარმატებით გამოიყენება აგრეთვე სამხედრო საქმეში, აეროდინამიკაში, საავტომობილო წარმოებაში, ენერგეტიკაში, ეკონომიკაში, მეცნიერებისა და ტექნიკის თითქმის ყველა სფეროში.

ამასთანავე, MATLAB ფართოდ გამოიყენება მსოფლიოს წამყვან უნივერსიტეტებში სამეცნიერო და სასწავლო პროექტების ფარგლებში. სხვადასხვა პროფილის სპეციალისტების მომზადების საქმეში, MATLAB წარმოადგენს თანამედროვე ინჟინერისა და მეცნიერ-ტექნიკური მუშაკისათვის მოდელირებისა და სხვა გამოყენებითი სისტემების შეუცვლელ ინსტრუმენტს.

MATLAB-ის პაკეტი შეიცავს რამდენიმე სპეციალიზებულ ინსტრუმენტთა ბიბლიოთეკას, ე.წ. Toolbox-ებს, რომლებიც განკუთვნილია კონკრეტული პრაქტიკული ამოცანების გადასაჭრელად. მაგალითად, symbolic Toolbox გამოიყენება სიმბოლური გამოთვლებისათვის, ხოლო Control Toolbox განკუთვნილია ავტომატური მართვის სისტემების მოდელირებისა და ანალიზისათვის.

სპეციალურად სიგნალების ციფრული დამუშავებისთვის კი არსებობს შემდეგი ბიბლიოთეკები:

signal Processing Toolbox: ეს არის ძირითადი და ყველაზე ფართოდ გამოყენებული Toolbox, რომელიც მოიცავს ფუნქციებსა და ალგორითმებს ციფრული სიგნალების ანალიზისთვის, სინთეზისთვის, ფილტრაციისთვის, ფურიეს გარდაქმნისთვის და

სპექტრული ანალიზისთვის. იგი შეიცავს შეიცავს სხვადასხვა ხელსაწყო სიგნალების დროით და სიხშირულ არეში დამუშავებისთვის.

DSP system Toolbox: ეს Toolbox გვთავაზობს უფრო რთულ და სპეციალიზებულ ფუნქციებს ციფრული სიგნალების დამუშავებისთვის, მათ შორის ფილტრის დიზაინსა და ანალიზს, ციფრული ფილტრაციის ალგორითმებს, მაღალი სიჩქარით დამუშავების მოდელირებას და რეალურ დროში დამუშავების შესაძლებლობებს. იგი ასევე შეიცავს ბლოკებს simulink-ის მოდელირებისთვის.

Wavelet Toolbox: განკუთვნილია სიგნალების ანალიზისთვის, იყენებს ტალღის გარდაქმნის ტექნიკას (Wavelet Transform). ეს ტექნიკა განსაკუთრებით სასარგებლოა სიგნალების ანალიზისთვის, რომელთა სიხშირე დროთა განმავლობაში იცვლება.

Communications Toolbox: მოიცავს ფუნქციებს, რომლებიც საჭიროა ციფრული კომუნიკაციების სისტემების მოდელირებისთვის, მათ შორის ციფრული მოდულაციის, კოდირების და სიგნალების დამუშავებისთვის, რომელიც გამოყენებულია კომუნიკაციების სისტემებში.

Filter Design Toolbox: განკუთვნილია ციფრული ფილტრების დიზაინსა და ანალიზისთვის. შეიცავს ხელსაწყოებს სხვადასხვა ტიპის ფილტრებისთვის, მათ შორის სიმ, უიმ ფილტრებისთვის და აგრეთვე ხელსაწყოებს სხვა მრავალსიჩქარიანი და ადაპტური ფილტრების შექმნისთვის.

MATLAB-ის (Matrix laboratory) შექმნის და განვითარების ისტორია იწყება 1970-იან წლებში, როდესაც კლივ მოულერი, მათემატიკოსი და კომპიუტერული მეცნიერების სპეციალისტი, მუშაობდა რიცხვითი მათემატიკის მიმართულებით (Gilat, 2022, 11-21). მისი მიზანი იყო შეექმნა პროგრამული საშუალება, რომელიც გაამარტივებდა ალგებრული და მატრიცული გამოთვლების შესრულებას მეცნიერებისთვის და ინჟინრებისთვის. კლივ მოულერის სამუშაო მოიცავდა FORTRAN-ზე დაფუძნებული პროგრამული პაკეტების, IINPACK-ისა და EISPACK-ის შექმნას, რომლებიც გამოიყენებოდა რიცხვითი ამოცანების ამოსახსნელად. მიუხედავად იმისა, რომ ეს პაკეტები იყო მძლავრი ინსტრუმენტები, მათი გამოყენება მოითხოვდა პროგრამირების მაღალ ცოდნას, რაც გამოთვლებს ართულებდა.

MATLAB-ის პირველი ვერსია, რომელიც მან შექმნა თავდაპირველად განკუთვნილი იყო მხოლოდ უნივერსიტეტებში გამოყენებისთვის. შემდეგ მისი ინტერფეისი იმდენად გამარტივდა, რომ საშუალებას აძლევდა სტუდენტებს და მკვლევარებს მარტივად ჩაატარებინათ კომპლექსური მათემატიკური გამოთვლები მატრიცებთან და ვექტორებთან მუშაობისას, ისე, რომ არ ჰქონდეთ ღრმა ცოდნა პროგრამირებაში.

1984 წელს კლივ მოულერმა და მისმა კოლეგა-მეგობარმა ჯეკ ლითლმა დააარსეს კომპანია MathWorks. ამ კომპანიის მიზანი იყო MATLAB-ის კომერციალიზაცია და გავრცელება უფრო ფართო აუდიტორიისთვის. MATLAB-ის პირველი კომერციული ვერსია, რომელიც ამავე წელს გამოვიდა, მალევე მოიპოვა პოპულარობა, განსაკუთრებით მეცნიერებსა და ინჟინრებში. MATLAB-ის წარმატება გამოიწვია მისმა უნიკალურობამ, რომელიც მოიცავდა მარტივ ინტერფეისს, ძალას და მოქნილობას, რაც საშუალებას აძლევდა მომხმარებლებს მარტივად ჩაატარებინათ რთული გამოთვლები, შექმნან მოდელები და ვიზუალიზაციები (Gilat, 2022, 30-45).

მე-20 საუკუნის ბოლოს MATLAB-ის განვითარება გრძელდებოდა. MathWorks-მა რეგულარულად გამოუშვა MATLAB-ის განახლებული ვერსიები, რომლებიც მოიცავდა ახალ ფუნქციებს და ხელსაწყოებს, რომლებიც აკმაყოფილებდა მომხმარებლების მზარდ მოთხოვნებს.

MATLAB-მა თავი დაიმკვიდრა მრავალ სფეროში, როგორცაა ავტომატიზაცია, კონტროლი, სიგნალის დამუშავება, ფინანსური მოდელირება და მრავალი სხვა. ამასთან, MATLAB-ის გაფართოებები, ე.წ. "Toolboxes," დამატებით ინსტრუმენტებს სთავაზობდა კონკრეტული სფეროების სპეციალისტებს, რაც კიდევ უფრო ზრდიდა მისი გამოყენების არეალს. იგი გახდა სტანდარტი უნივერსიტეტებში სასწავლო და კვლევითი პროექტებისთვის, ასევე მრეწველობაში, სადაც გამოიყენება როგორც ინჟინერიაში, ასევე ბიზნესის ანალიზსა და განვითარებაში.

MathWorks აგრძელებს MATLAB-ის განახლებას და გაფართოებას, რეგულარულად გამოდის ახალი ვერსიები, რომლებიც მოიცავს უფრო მძლავრ ინსტრუმენტებს, უფრო მდიდარ ფუნქციონალობას და გაუმჯობესებულ ინტერფეისს. MATLAB დღემდე რჩება ერთ-ერთ ყველაზე მნიშვნელოვან ინსტრუმენტად მსოფლიოში ტექნიკური გამოთვლებისა და მოდელირებისათვის.

შესავალშივე განვმარტოთ რა ძირითად ოპერაციებს ვიყენებთ დისკრეტულ სიგნალებში. რადგან სიგნალები ზოგადად იყოფა ანალოგურ და დისკრეტულ სიგნალებად. ანალოგურ სიგნალს აღნიშნავთ როგორც $x_a(t)$, სადაც ცვლადი t შეიძლება წარმოადგენდეს ნებისმიერ ფიზიკურ სიდიდეს, მაგრამ ჩვენ ჩავთვალთ, რომ ის აღნიშნავს დროს წამებში. დისკრეტულ სიგნალს აღნიშნავთ როგორც $x(n)$, სადაც ცვლადი n არის მთელი რიცხვი და აღნიშნავს დროის გარკვეულ (დისკრეტულ) მომენტებს. შესაბამისად, მას ასევე უწოდებენ დისკრეტული დროის სიგნალს, რომელიც წარმოადგენს რიცხვთა თანმიმდევრობას და აღნიშნება შემდეგნაირად:

$$x(n) = \{x(n)\} = \{\dots, x(-2), x(-1), x(0), x(1), x(2) \dots\} \quad (\text{Ganzha 2022, 25})$$

სადაც მაგ: $x(0)$ შემთხვევაში n უტოლდება 0-ს.

ქვემოთ განვიხილოთ თუ როგორ შეიძლება რამდენიმე მნიშვნელოვანი ფუნქციის გამხორციელება MATLAB-ში.

მაგალითად ციფრული სიგნალების დამუშავებაში ანალიზისთვის ვიყენებთ რამდენიმე ელემენტარულ თანმიმდევრობას. მაგ: ე.წ. ჩართვის ფუნქციას

$$\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases} = \{\dots, 0, 0, 1, 0, 0, \dots\}. \quad (\text{Ganzha 2022, 27})$$

MATLAB-ში ფუნქცია `zeros(1,N)` ქმნის N ნულით შევსებულ რიგით ვექტორს, რომელიც შეიძლება გამოყენებულ იქნას $\delta(n)$ -ის განხორციელებისთვის ფინიტურ ინტერვალზე. თუმცა, ლოგიკური მიმართება $n==0$ არის ძალიან მოსახერხებელი გზა $\delta(n)$ -ის უფრო მარტივად განსახორციელებლად. მაგალითად,

$$\delta(n - n_0) = \begin{cases} 1, & n = n_0 \\ 0, & n \neq n_0 \end{cases} \quad (\text{MathWorks 2024, 103})$$

$n_1 \leq n_0 \leq n_2$ ინტერვალში შეიძლება განვახორციელოთ როგორც

```
function [x,n] = impseq(n0,n1,n2)
* Generates x(n) = delta(n-n0); n1 <= n <= n2
* -----
* [x,n] = impseq(n0,n1,n2)
*
n = [n1:n2]; * ქმნის n ვექტორს n1-დან n2-მდე
x = [(n-n0) == 0]; * ქმნის x ვექტორს, რომელშიც არის 1 იმ პოზიციაზე, სადაც
n = n0, ხოლო სხვა პოზიციებზე 0
```

- კიდევ ერთი მნიშვნელოვანი ფუნქცია

$$u(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases} = \{\dots, 0, 0, 1, 1, 1, \dots\}$$

MATLAB-ში ფუნქცია ones(1,N) ქმნის N ერთიანი შევსებულ რიგით ვექტორს. ის შეიძლება გამოყენებულ იქნას $u(n)$ -ის (ერთეულოვანი ბიჯის თანმიმდევრობის) გენერირებისთვის ფინიტურ ინტერვალზე. თუმცა, კიდევ ერთხელ, ძალიან მოსახერხებელი გზაა ლოგიკური მიმართება $n \geq 0$ -ის გამოყენება და გვექნება

$$u(n - n_0) = \begin{cases} 1, & n \geq n_0 \\ 0, & n < n_0 \end{cases}, \text{ რომელის განხორციელება } n_1 \leq n_0 \leq n_2 \text{ ინტერვალში}$$

შესაძლებელია შემდეგნაირად:

```
function [x,n] = stepseq(n0,n1,n2)
* Generates x(n) = u(n-n0); n1 <= n <= n2
* -----
* [x,n] = stepseq(n0,n1,n2)
*
n = [n1:n2]; * ქმნის n ვექტორს n1-დან n2-მდე
x = [(n-n0) >= 0]; * ქმნის x ვექტორს, რომელიც არის 1 იმ პოზიციებზე, სადაც
n >= n0, ხოლო სხვა პოზიციებზე 0
```

- განვიხილოთ როგორაა შესაძლებელი ნამდვილ მნიშვნელობიანი სიგნალის წარმოდგენა, რომელიც მოიცემა ფორმულით $x(n) = a^n, \forall n; a \in \mathbb{R}$.

MATLAB-ში არსებობს ოპერატორი “.^” , რომელიც იძლევა შესაძლებლობას წარმოვადგინოთ სიგნალი, მაგ: $x(n) = (0.9)^n, 0 \leq n \leq 10$

```
>> n = [0: 10]; x = (0.9).^ n;
```

- სინუსოიდური სიგნალი $x(n) = A \cos(\omega_0 n + \theta_0), \forall n$ (MathWorks 2024, 106)

სადაც A ამპლიტუდაა და θ_0 ფაზა რადიანებში. MATLAB-ის ფუნქცია cos (ან sin) გამოიყენება სინუსოიდური სიგნალის გენერირებისთვის. მაგ: რომ დავაგენერიროთ

$$x(n) = 3 \cos(0.1\pi n + \pi/3) + 2 \sin(0.5\pi n), 0 \leq n \leq 10 \text{ გვეჭირდება კოდი:}$$

```
>> n = [0: 10]; x = 3 * cos(0.1 * pi * n + pi/3) + 2 * sin(0.5 * pi * n)
```

- სიგნალების შეკრების ოპერაცია: $(x_1(n)) + (x_2(n)) = (x_1(n) + x_2(n))$ (Ganzha 2022, 56)

MATLAB-ში მიმატების ოპერაცია ხორციელდება არითმეტიკული ოპერატორის „+“. თუმცა, $x_1(n)$ და $x_2(n)$ ვექტორის სიგრძეები უნდა იყოს თანაბარი. თუ თანმიმდევრობები არათანაბარი სიგრძისაა, ან თუ ანათვლების პოზიციები განსხვავდება თანმიმდევრობებში, მაშინ "+" ოპერატორს ვერ გამოვიყენებთ. ეს ორი მიმდევრობა რაოდენობრივად უნდა გავათანაბროთ. უნდა გამოვიყენოთ "&", "<=", "==" და ფუნქცია finde. ამის შემდეგ შესაძლებელია გამოვიყენოთ ფუნქცია sigadd, მას შეუძლია ორი თანაბარი შიოგრძის სიგნალის გაერთიანება.

```
n = min(min(n1),min(n2)):max(max(n1),max(n2)); * y(n)-ის დროის დიაპაზონი
y1 = zeros(1,length(n)); y2 = y1; * ინიციალიზაცია ნულლებით
```

```
y1(find((n>=min(n1))&(n<=max(n1))==1)) = x1; * x1-ის გასწორება y-ს დროის დიაპაზონზე
y2(find((n>=min(n2))&(n<=max(n2))==1)) = x2; * x2-ის გასწორება y-ს დროის დიაპაზონზე
y = y1 + y2; * თანმიმდევრობების დამატება
```

- სიგნალების გადამრავლების ოპერაცია: იგულისხმება თითოეული სიგნალის ანათვალის გამრავლება ერთმანეთზე. $(x_1(n)) \cdot (x_2(n)) = (x_1(n) \cdot x_2(n))$ (Ganzha 2022, 76).

```
function [y,n] = sigadd(x1,n1,x2,n2)
* ახორციელებს y(n) = x1(n)+x2(n)
* -----
* [y,n] = sigadd(x1,n1,x2,n2)
* y = თანმიმდევრობის ჯამია n დიაპაზონზე, რომელიც მოიცავს n1-ს და n2-ს
* x1 = პირველი თანმიმდევრობა n1 დიაპაზონზე
* x2 = მეორე თანმიმდევრობა n2 დიაპაზონზე (n2 შეიძლება განსხვავდებოდეს n1-სგან)
```

* ამ ოპერაციის განხორციელება შეიძლება „.*“ ოპერატორით, თუმცა მასაც იგივე შეზღუდვა აქვს, რაც „+“ ოპერატორს. ამიტომ შემუშავებულია ფუნქცია sigmult.

```

function [y,n] = sigmult(x1,n1,x2,n2)
% ახორციელებს  $y(n) = x1(n) * x2(n)$ 
% -----
% [y,n] = sigmult(x1,n1,x2,n2)
% y = ნამრავლის თანმიმდევრობა n დიაპაზონზე, რომელიც მოიცავს n1-ს და
n2-ს
% x1 = პირველი თანმიმდევრობა n1 დიაპაზონზე
% x2 = მეორე თანმიმდევრობა n2 დიაპაზონზე (n2 შეიძლება
განსხვავდებოდეს n1-სგან)
%
n = min(min(n1),min(n2)):max(max(n1),max(n2)); % y(n)-ის დროის დიაპაზონი
y1 = zeros(1,length(n)); y2 = y1; % ინიციალიზაცია ნულებით
y1(find((n>=min(n1))&(n<=max(n1))==1)) = x1; % x1-ის გასწორება y-ს დროის
დიაპაზონზე
y2(find((n>=min(n2))&(n<=max(n2))==1)) = x2; % x2-ის გასწორება y-ს დროის
დიაპაზონზე
y = y1 .* y2; % თანმიმდევრობების გამრავლება

```

მასშტაბირება სიგნალების დამუშავებაში გულისხმობს სიგნალის ამპლიტუდის გაორმაგებას ან შემცირებას გარკვეული ფაქტორით. მასშტაბირება შეიძლება იყოს უარყოფითი ან დადებითი და გავლენას ახდენს სიგნალის ამპლიტუდაზე, ხოლო მისი დროითი ან სიხშირული მახასიათებლები უცვლელი რჩება. $a\{x(n)\} = \{ax(n)\}$. არითმეტიკული ოპერატორი (*) გამოიყენება MATLAB-ში მასშტაბირების ოპერაციის განსახორციელებლად (MathWorks 2024, 123).

- **გადანაცვლება:** ამ ოპერაციის დროს, $x(n)$ -ის თითოეული ანათვალის გადინაცვლება k რაოდენობით, შედეგად მივიღებთ გადანაცვლებულ თანმიმდევრობა $y(n) - \mathbb{Z}$. $y(n) = x(n - k)$ (Ganzha 2022, 58). $m = n - k - s$ და $n = m + k - s$, თუ წინა ოპერაცია ჩაიწერება როგორც $y(m + k) = (x(m))$. უნდა ღვნიშნოთ, რომ x ვექტორი არ იცვლება, მხოლოდ n , ემატება $-k$. ეს ნაჩვენებია ფუნქციაში sigshift

```

function [y,n] = sigshift(x,m,k)
% ახორციელებს  $y(n) = x(n-k)$ 
% -----
% [y,n] = sigshift(x, m,k);

```

$n = m + k$; % დროის ინდექსის გადანაცვლება m -დან n -ზე, k -ს დამატებით
 $y = x$; % $y(n)$ არის $x(n-k)$ (გადანაცვლებული თანმიმდევრობა)

ნაკეცი - ციფრული ფილტრაციის ძირითადი ოპერაციაა, ამ დროს $x(n)$ -ის თითოეული ანათვალის იკეცება $n=0$ -ზე, ძველი სიგნალი ახალ სიგნალად გადაიქცევა, ფაქტიურად „ბრუნდება“, დროის ღერძზე მნიშვნელობები იღებენ საპირისპირო მნიშვნელობას, ხოლო ფორმა რჩება იგივე. $y(n) = \{x(-n)\}$, MATLAB-ში ნაკეცი ხორციელდება `fliplr(x)` ფუნქციით (MathWorks 2024, 145-147).

```
function [y,n] = sigfold(x,n)
% ახორციელებს  $y(n) = x(-n)$ 
% -----
%  $[y,n] = sigfold(x,n)$ 
 $y = \text{fliplr}(x)$ ; % სიგნალი  $x(n)$  იკეცება და ინახება  $y(n)$ -ში
 $n = -\text{fliplr}(n)$ ; %  $n$  მნიშვნელობების გადაკეცვა და ნიშნის შეცვლა
```

- **ანათვლების შეკრება**, ეს ოპერაცია განსხვავდება სიგანელბის შეკრების ოპერაციისგან, ამ შემთხვევაში იკრიბება თვითონ $x(n)$ -ის თითოეული ანათვალის მნიშვნელობა n_1 -სა და n_2 -ს შორის.

$\sum_{n=n_1}^{n_2} x(n) = x(n_1) + \dots + x(n_2)$, რომელიც MATLAB-ში ხორციელდება `sum(x(n1:n2))` ფუნქციით (MathWorks 2024, 168-169).

- **ანათვლების გადამრავლება**, ეს ოპერაცია განსხვავდება სიგანელბის გადამრავლების ოპერაციისგან, ამ შემთხვევაში მრავლდება თვითონ $x(n)$ -ის თითოეული ანათვალის მნიშვნელობა n_1 -სა და n_2 -ს შორის.

$\prod_{n_1}^{n_2} x(n) = x(n_1) \times \dots \times x(n_2)$, რომელიც MATLAB-ში ხორციელდება `prod(x(n1:n2))` ფუნქციით.

- **სიგნალის ენერჯია**: $x(n)$ -ის ენერჯია მოიცემა ფორმულით
- $E_x = \sum_{-\infty}^{\infty} x(n)x^*(n) = \sum_{-\infty}^{\infty} |x(n)|^2$, სადაც * აღნიშნავს კორელაციას (MathWorks 2024, 173).

```
>>  $E_x = \text{sum}(x,* \text{conj}(x))$ ; % სიგნალის ენერჯიის გამოთვლა
```

```
>>  $E_x = \text{sum}(\text{abs}(x)_2)$ ; % იგივე სიგნალის ენერჯიის გამოთვლა სხვა მეთოდით
```

- **სიგნალის სიმძლავრის გამოთვლა:** პერიოდული მიმდევრობის $\tilde{x}(n)$ საშუალო სიმძლავრე, რომლის უმცირესი დადებითი პერიოდი არის N , განისაზღვრება შემდეგნაირად (MathWorks 2024, 180):

$$\mathcal{P}_x = \frac{1}{N} \sum_0^{N-1} |\tilde{x}(n)|^2$$

წყაროებისა და სამეცნიერო ლიტერატურის მიმოხილვა.

ციფრული სიგნალის დამუშავება (DSP) არის ფართოდ გავრცელებული დარგი, რომელიც გამოიყენება ინფორმაციის ანალიზისა და გადამუშავების მიზნით. ბოლო რამდენიმე ათწლეულში, სიგნალის აღდგენის ალგორითმები, ციფრული ფილტრაციის მეთოდები და ელექტრონულ წრედებში მათი გამოყენება გახდა აქტიური კვლევის ობიექტი.

ქართული სამეცნიერო სივრცე ჯერ კიდევ განვითარების პროცესშია ციფრული სიგნალების დამუშავების მიმართულებით, რაც აისახება შესაბამისი ლიტერატურის ნაკლებობაზე. ციფრული სიგნალის დამუშავება (DSP) არის ფართოდ გავრცელებული და აქტიურად განვითარებადი დარგი, რომელიც გამოიყენება ელექტრონიკაში, კომუნიკაციებში, სამედიცინო ტექნოლოგიებში, აუდიო და ვიდეო სიგნალების დამუშავებაში, თუმცა, სამწუხაროდ, ქართულ ენაზე არსებული ლიტერატურა ამ სფეროში ძალიან მწირია. არსებული ლიტერატურა მოიცავს მხოლოდ რამდენიმე ძირითად ნაშრომს, სადაც განხილულია სიგნალების დამუშავების ზოგადი პრინციპები. ერთ-ერთი ასეთი წიგნია "ციფრული სიგნალის დამუშავება" ელგუჯა ყუბანეიშვილის ავტორობით, რომელიც ფოკუსირებულია ბიოსიგნალების დამუშავებაზე (ყუბანეიშვილი, 2013), მაგრამ არ მოიცავს ყველა იმ თანამედროვე მეთოდსა და ტექნოლოგიას, რაც სიგნალების დამუშავების სფეროში საერთაშორისო დონეზეა ცნობილი. ამ წიგნში დეტალურად არის განხილული ბიოსიგნალების აღება და მათი მათემატიკური ანალიზი, ფოკუსირებულია ბიოსიგნალებზე და ფიზიოლოგიურ პროცესებზე, წიგნში განხილულია ფიზიოლოგიური პროცესების მიერ წარმოებული სიგნალების ტიპები, როგორცაა ელექტროკარდიოგრამა (EKG), ელექტროენცეფალოგრამა (EEG) და სხვა მნიშვნელოვანი ბიოსიგნალები, რომლებიც გამოიყენება მედიცინაში. ყურადღება

ეთმობა ამ სიგნალების ციფრული დამუშავების ტექნიკებს, რაც მოიცავს მონაცემთა აღების და დამუშავების სხვადასხვა მეთოდებს, როგორცაა დისკრეტიზაციის სიხშირე, ფურიეს გარდაქმნები და ციფრული ფილტრაციის მეთოდები. თუმცა არ არის გათვალისწინებული პროგრამული ინსტრუმენტების პრაქტიკული გამოყენება, რაც თანამედროვე ციფრული სიგნალის დამუშავების სფეროში ძალიან მნიშვნელოვანია. თუ გავითვალისწინებთ, რომ სიგნალების ციფრული დამუშავება ისეთ დარგებშია აქტუალური, როგორცაა სამედიცინო დიაგნოსტიკა, მობილური კომუნიკაციები, აუდიო და ვიდეო ტექნოლოგიები, სარადარო და სატელიტური სისტემები. კვლევებისა და ინჟინრული სამუშაოების ეფექტურად ჩატარება ხშირად მოითხოვს იმ პროგრამული უზრუნველყოფის და ხელსაწყოების გამოყენებას, რომლებიც არ არის ხელმისაწვდომი ქართულენოვანი ლიტერატურის საშუალებით.

ქართულ სამეცნიერო სივრცეში აღსანიშნავია მხოლოდ რამდენიმე ნაშრომი, სადაც სიგნალების დამუშავების ძირითადი პრინციპებია განხილული:

აკაკი ფაღავას ნაშრომი "ბიოსიგნალების დამუშავების საფუძვლები" ასევე ფოკუსირებულია ბიოსამედიცინო სიგნალების ანალოგური მეთოდებით დამუშავებაზე (ფაღავა, 2011). წიგნი მოიცავს ბიოსიგნალების ზოგად მიმოხილვას, მათ კლასიფიკაციას, სპექტრალურ ანალიზს, კორელაციურ ანალიზსა და ანალოგური ფილტრების გამოყენების მეთოდებს. ავტორი ყურადღებას ამახვილებს სიგნალების თეორიულ და პრაქტიკულ საფუძვლებზე, რაც მკითხველს საშუალებას აძლევს, უკეთ გაიგოს ბიოსამედიცინო სიგნალების ანალიზისა და დამუშავების პროცესები. ძირითადი განხილული თემებია: ბიოსიგნალების ნაირსახეობა და მათემატიკური მოდელირება; სიგნალების სპექტრალური და კორელაციური ანალიზი; ანალოგური ფილტრაციის მეთოდები; სიგნალების ენერგეტიკული სპექტრის და ავტოკორელაციის ანალიზი;

საერთაშორისო ლიტერატურა, როგორცაა Oppenheim-ისა და schaffer-ის "Discrete-Time signal Processing" ან Proakis-ის "Digital signal Processing: Principles, Algorithms, and Applications", შედარებით რთული ხელმისაწვდომია ქართულენოვანი მკითხველებისთვის (Oppenheim, Schaffer, 2009); (Proakis, Manolakis 2016). მიუხედავად იმისა, რომ ეს წიგნები ციფრული სიგნალების დამუშავების მსოფლიოში აღიარებული

რესურსებია, მათი ქართულ ენაზე თარგმნილი ვერსიები არ არსებობს, რაც ქმნის დამატებით ბარიერს მათთვის, ვისაც ეს თემატიკა აინტერესებს, მაგრამ საკმარისი ინგლისური ენის ცოდნა არ აქვთ. შედეგად, მკვლევარებსა და სტუდენტებს საქართველოში უწევთ რთული არჩევანის გაკეთება – გამოიყენონ საერთაშორისო ლიტერატურა უცხო ენაზე, ან დაეყრდნონ მწირ ქართულ რესურსებს, რაც ხშირად საკმარისი არ არის თანამედროვე ციფრული სიგნალების დამუშავების ყველა ასპექტის გასაცნობად. ამრიგად, ლიტერატურის სიმწირე სერიოზულ პრობლემას წარმოადგენს ციფრული სიგნალის დამუშავების სწავლების, კვლევისა და გამოყენების პროცესში ქართულ სამეცნიერო და საგანმანათლებლო სივრცეში. აუცილებელია ახალი ნაშრომების შემუშავება და თარგმნა, რომლებიც უფრო ღრმა და მრავალმხრივ შეხედულებებს მიაწვდიან მკვლევარებს და სტუდენტებს, რითაც მათ საშუალებას მისცემს, განავითარონ ეს დარგი საქართველოში და გააფართოვონ თავისი ცოდნის დიაპაზონი საერთაშორისო დონეზე.

ციფრული სიგნალის დამუშავების ძირითადი პრინციპები ფართოდ არის განხილული კლასიკურ ლიტერატურაში, როგორცაა "Discrete-Time signal Processing" (Oppenheim, Alan V., schaffer, Ronald W., 3rd edition, Prentice Hall, 2009). ეს წიგნი წარმოადგენს ერთ-ერთ მთავარ წყაროს ციფრული სიგნალის დამუშავების შესახებ და აქ აღწერილია ციფრული სისტემების ფუნქციონირება, ანათვლების აღების თეორია და დისკრეტული დროითი სისტემების ანალიზი. წიგნი ასევე ეხება ისეთ საკითხებს, როგორცაა ციფრული ფილტრაციის ძირითადი ტიპები და სიგნალის გარდაქმნები, რომლებიც გამოიყენება ციფრულ გარემოში.

John G. Proakis და Dimitris G. Manolakis თავიანთ წიგნში "Digital signal Processing: Principles, Algorithms, and Applications" (4th edition, Pearson 2016) ხსნიან თუ როგორ ხდება ციფრული სიგნალის დამუშავება და ფილტრაცია. წიგნში დეტალურადაა განხილული სიგნალის აღდგენა, Z-გარდაქმნა, ფურიეს გარდაქმნები და ფილტრების დიზაინის მეთოდები. იგი გვთავაზობს მნიშვნელოვან ალგორითმებს, რომელიც დაკავშირებულია სიგნალის აღდგენასთან (Proakis, Manolakis 2016).

"Digital signal Processing: A Practical Approach" - Emmanuel C. Ifeachor, barrie W. Jervis წიგნი წარმოადგენს პრაქტიკულ სახელმძღვანელოს ციფრული სიგნალების

დამუშავების პროცესზე. ავტორები ყურადღებას ამახვილებენ ალგორითმებზე, რომლებიც რეალურ პრობლემებზეა დაფუძნებული. წიგნში წარმოდგენილია ციფრული სიგნალების დამუშავების ალგორითმები კონკრეტული კოდებით, რაც იშვიათია ამ სფეროში. მაგალითად, რეალურ დროში სიგნალების დამუშავების ამოცანების გადაწყვეტისთვის წარმოდგენილია C პროგრამირების ენაზე დაწერილი ალგორითმები, რაც მკვლევარს საშუალებას აძლევს, ადვილად და პრაქტიკულად გამოიყენოს იგი (Ifeachor, Jervis, 2018).

"Understanding Digital signal Processing" - Richard G. baraniuk წიგნში გამორჩეულად მარტივად და გასაგებად არის ახსნილი სიგნალების დამუშავების რთული კონცეფციები. ავტორი იყენებს უამრავ ვიზუალურ ილუსტრაციას, გრაფიკს და მაგალითს, რაც მკითხველს ეხმარება რთული მათემატიკური მოდელების გააზრებაში. წიგნში განსაკუთრებული ყურადღება ეთმობა ციფრული სიგნალების დამუშავების პრაქტიკულ გამოყენებას თანამედროვე ტექნოლოგიებში. აღწერილია როგორ გამოიყენება სიგნალების დამუშავების ალგორითმები ყოველდღიურ ცხოვრებაში – მობილური ტელეფონების სიგნალების ანალიზიდან დაწყებული, ელექტრონულ მუსიკალურ მოწყობილობებამდე და სამედიცინო სიგნალების დამუშავებით დამთავრებული (Baraniuk, 2015).

"Digital signal Processing Using MATLAB" - Vinay K. Ingle, John G. Proakis წიგნი მოიცავს ციფრული სიგნალების დამუშავების ძირითადი და კომპლექსური ალგორითმების დეტალურ ახსნას. სხვა წიგნებისგან განსხვავებით, აქ ნაჩვენებია, როგორ შეიძლება ალგორითმების განხორციელება MATLAB-ში და მათი ტესტირება პრაქტიკულ სიტუაციებში, რაც ინჟინრებსა და პროგრამისტებს აძლევს შესაძლებლობას, მარტივად მოახდინონ ციფრული სისტემების მოდელირება.

ავტორი Simon Haykin წიგნში "Adaptive signal Processing" განსაკუთრებულ ყურადღებას აქცევს ადაპტური ფილტრაციის თემას, რაც მას გამოარჩევს უფრო ზოგადი სიგნალების დამუშავების სახელმძღვანელოებისგან. ადაპტური ფილტრები გამოიყენება ისეთ დინამიურ სისტემებში, სადაც გარემო პირობები ცვალებადია და ეს წიგნი დეტალურად განიხილავს, როგორ უნდა განვითარდეს ფილტრაციის ალგორითმები მუდმივად ცვალებად პირობებში (Haykin 2013).

"Statistical Digital Signal Processing and Modeling" - Monson H. Hayes წიგნში სიგნალების დამუშავებას სტანდარტული მათემატიკური და ინჟინრული მეთოდების ნაცვლად სტატისტიკური პერსპექტივიდან განიხილავს. Hayes-ის ნაშრომი დეტალურად აღწერს სიგნალების სტატისტიკურ მოდელირებას, რაც მნიშვნელოვანია ისეთი სისტემებისთვის, რომლებიც მუშაობენ შემთხვევით და გაურკვეველ მონაცემებთან, მაგალითად ხმაურთან (Hayes 2000).

"**The scientist and Engineer's Guide to Digital signal Processing**" by Steven W. Smith. ეს წიგნი კი მოიცავს ციფრული სიგნალის დამუშავების მნიშვნელოვან პრინციპებს და კარგად ხსნის ისეთ თემებს, როგორცაა ფილტრაცია, სიგნალის აღდგენა, და ციფრული სისტემების მოდელირება (Smith, 2011). აღნიშნულია, რომ **სიმ (Finite Impulse Response)** და **უიმ (Infinite Impulse Response)** ფილტრების მნიშვნელობა ციფრული სიგნალის დამუშავების პროცესში ძალიან დიდია. **სიმ ფილტრები** ცნობილია თავისი ხაზოვანი ფაზური რეაქციით და სტაბილურობით, წიგნის ავტორი სწორედ მას ირჩევს სიგნალების დამახინჯებასთან საბრძოლველად. ზემოთ ნახსენები Proakis-ისა და Manolakis-ის ტექსტში ვრცლადაა განხილული სიმ და უიმ ფილტრების გამოყენების შემთხვევები და მათი ქცევის მოდელირება სხვადასხვა ტიპის სიგნალების შემთხვევაში. **სიმ ფილტრები** უფრო მარტივად დასამუშავებელია, რადგან მათ სტაბილურობა ყოველთვის გარანტირებულია. მეორეს მხრივ, **უიმ ფილტრები** (Oppenheim და Schaffer-ის "**Discrete-Time signal Processing**") გამოიყენება უფრო კომპლექსური სიგნალების დამუშავებისთვის. უიმ ფილტრები საჭიროებენ ნაკლებ გამოთვლით რესურსებს და უფრო მცირე რიგის ფილტრებია საჭირო შედეგის მისაღწევად. თუმცა, უიმ ფილტრების მთავარი სირთულე არის მათი პოლუსების სტაბილურობა და უკუკავშირის გამოყენება. უიმ ფილტრები შესაძლებელია აღმოჩნდნენ არასტაბილურ მდგომარეობაში, თუ მათი პოლუსების მნიშვნელობები ერთზე მეტი იქნება, რაც მათ დიზაინში დამატებით სირთულეებს ქმნის.

სიგნალის აღდგენის ალგორითმები ციფრული სიგნალის დამუშავების მნიშვნელოვანი ნაწილია. **Nyquist-shannon-ის ნიმუშების აღების თეორემა** (Papoulis, A. "**signal Analysis**", McGraw-Hill, 1999) სიგნალის აღდგენის ძირითადი პრინციპია. ეს თეორემა ამბობს, რომ სიგნალი შეიძლება სრულად აღდგეს, თუ ანათვლების აღების

სიხშირე მეტი იქნება სიგნალის სიხშირეზე. (ნაიქვისტ-შენონის კრიტერიუმი). აღნიშნული თეორემის გამოყენება მნიშვნელოვანია ისეთ სფეროებში, როგორცაა აუდიო და ვიდეო გადაცემები, სადაც სიგნალის ხარისხის დაკარგვა დაუშვებელია (Papoulis 1999).

ციფრული სიგნალის აღდგენის პროცესში მნიშვნელოვანი გამოწვევაა **ზედდების ეფექტია** (*aliasing*), რომელიც წარმოიქმნება ნიმუშების არასაკმარისი სიხშირის შემთხვევაში. ზედდების ეფექტისგან დასაცავად გამოიყენება **ფილტრები**, რომლებიც ამცირებენ სიგნალის დამახინჯების შანსს. **Sanjit K. Mitra**-ს წიგნში "**Digital signal Processing: A Computer-based Approach**" (4th edition, McGraw-Hill, 2010) დეტალურადაა აღწერილი ზედდების ეფექტის პრევენცია და ფილტრაციის მეთოდები, რომლებიც გამოიყენება სხვადასხვა პრაქტიკულ აპლიკაციაში. Mitra ასევე ასახავს MATLAB-ის გამოყენების მაგალითებს ფილტრების დიზაინსა და მათ გამოთვლებში.

MATLAB მნიშვნელოვან როლს ასრულებს ციფრული სიგნალის დამუშავების ალგორითმების მოდელირებასა და სიმულაციაში. MATLAB-ის **signal Processing Toolbox** და მისი **dimpulse**, **filter** და **fft** ფუნქციები უზრუნველყოფენ სიგნალის ანალიზისა და ფილტრაციის სხვადასხვა მეთოდების მარტივ გამოყენებას. **MathWorks**-ის **დოკუმენტაციაში** (*signal Processing Toolbox User's Guide*) აღწერილია, თუ როგორ შეიძლება ფილტრების დიზაინი და მათი სიგნალის დამუშავებაში გამოყენება.

ასევე "**Finite Impulse Response (სიმ) Filters**" მაიკლ პარკერის წიგნის **Digital Signal Processing** მეოთხე თავში განიხილავს იმპულსური მახასიათებლების (სიმ) ფილტრების საფუძვლებს, მათ თვისებებსა და გამოყენებებს. ავტორი დეტალურად განმარტავს სიმ ფილტრების სტრუქტურას, მათემატიკურ მოდელირებას და ციფრული ფილტრაციის პროცესს. განსაკუთრებით აქცენტი აკეთდება ხაზოვანი ფაზის შენარჩუნების უპირატესობაზე, რასაც ნაწილობრივ გამოყენებულია დისერტაციასიც, რიცხვითი რიცხვითი ექსპერიმენტის ჩატარების დროს. ამასთან, წიგნში წარმოდგენილია პრაქტიკული მაგალითები და მათემატიკური განმარტებები, რომლებიც ეხმარება მკითხველს გააცნობიეროს სიმ ფილტრების პრაქტიკული გამოყენება ციფრულ სიგნალის დამუშავებაში.

სტატია "FPGA-based Electrocardiography (ECG) Signal Analysis System Using Least-Square Linear Phase Finite Impulse Response (სიმ) Filter," ავტორები მოჰამედ გ. ეგილა, მაგდი ა. ელ-მოურსი, ადელ ე. ელ-ჰენავი, ჰამედ ა. ელ-სიმარი და ამალ ზაკი, აღწერს FPGA-ს ბაზაზე შექმნილ სისტემას, რომელიც გამოიყენება ელექტროკარდიოგრაფიის (ECG) სიგნალების ანალიზისთვის. სტატია გამოქვეყნდა **Journal of Electrical Systems and Information Technology**-ში 2016 წელს. კვლევა განიხილავს სიმ ფილტრების გამოყენებას, რომლებიც გამოთვლილია უმცირეს კვადრატთა მეთოდით. ეს ფილტრები გამოიყენება ECG სიგნალების გაუმჯობესებისა და მათი ანალიზის ეფექტურობის ასამაღლებლად. უმცირეს კვადრატთა მეთოდი განხილულია დისერტაციაში და წარმოდგენილია მისი პროგრამული უზრუნველყოფა.

სტატია "Sensor Fusion Noise Suppression Method Based on Finite Impulse Response Complementary Filters," ავტორები ჯინგ ვანგი, ლეი ვანგი, ჯიე ლინი, რიტაო სიაო, ჯიამინ ჩენი და პენგ ჯინი, 2023 წელს გამოქვეყნდა ჟურნალ Measurement-ში. კვლევა ეხება სენსორული მონაცემების დამუშავებისას სიგნალებში ხმაურის შემცირების მეთოდს, რომელიც დაფუძნებულია სიმ ფილტრებზე. მეთოდი ემყარება სხვადასხვა სენსორიდან მიღებული მონაცემების ინტეგრაციას, რაც უზრუნველყოფს უფრო სუფთა და სანდო სიგნალის მიღებას. კვლევის შედეგები აჩვენებს, რომ ეს მეთოდი ეფექტურად ამცირებს ხმაურს და აუმჯობესებს სიგნალის ხარისხს. დისერტაციაში პროგრამულად ხორციელდება ასეთი სიმ ფილტრი და ჩატარებულია რიცხვითი რიცხვითი ექსპერიმენტი.

სტატია "Design of a Finite Impulse Response Filter for Rapid Single-Flux-Quantum Signal Processors Based on Stochastic Computing," ავტორები რუიდი ჩიუ, პეიაო ქუ, სიანგიუ ჟენგი და გუანგმინგ ტანგი, გამოქვეყნდა 2023 წელს ჟურნალ Superconductivity-ში. სტატიაში ავტორები წარმოადგენენ სიმ ფილტრის დიზაინს, რომელიც უზრუნველყოფს მაღალი სიჩქარის და დაბალი ენერგომომხმარების სიგნალის დამუშავებას სუსტი პროცესორებისთვის. მათი მეთოდის გამოყენება იძლევა გამოთვლითი რესურსების ოპტიმიზაციის საშუალებას, რაც ამცირებს სისტემის სირთულეს და ხარჯებს. კვლევა აჩვენებს, რომ შემუშავებული ფილტრი ეფექტურად აუმჯობესებს სიგნალის ხარისხს.

სტატია X. Chu და სხვ., „Fast and Accurate Estimation of Statistical Eye Diagram for Nonlinear High-Speed Links“ (2021) განსაკუთრებული მნიშვნელობის მქონე წყაროა სიგნალის ხარისხის შეფასებისა და ოპტიმიზაციის მეთოდების განხილვისას. ავტორები წარმოადგენენ ახალ მეთოდს, რომელიც ამარტივებს ზუსტ შეფასებას მაღალი სიჩქარის კავშირებისთვის. მათი მიდგომა საშუალებას იძლევა ეფექტურად შევაფასოთ სიგნალის ხარისხი და ოპტიმიზაციის პროცესები, ეს ალგორითმი გამოყენებულია დისერტაციაში სიგნალის დამუშავების ალგორითმების გაუმჯობესებისას.

სტატია S. Chauhan და სხვ., „Designing of Optimal Digital უიმ Filter in the Multi-Objective Framework Using an Evolutionary Algorithm“ (2023) მნიშვნელოვანი წყაროა დისერტაციის ექსპერიმენტალური კვლევისთვის, რადგან სტატიაში ავტორები წარმოგვიდგენენ ახალ მიდგომას, რომელიც საშუალებას იძლევა უიმ ფილტრების ოპტიმიზაციას რამდენიმე ფუნქციით, რომლის პროგრამული განხორციელებაც დეტალურადაა აღწერილი დისერტაციაში.

სტატია "Design of Digital უიმ Filter: A Research Survey," ავტორები ნ. აგრავალ, ა. კუმარ, ვარუნ ბაჯაჯი და გ.კ. სინგი, გამოქვეყნდა 2021 წელს ჟურნალ **Applied Acoustics**-ში. კვლევა წარმოადგენს ციფრული უიმ (Infinite Impulse Response) ფილტრების დიზაინის ფართო მიმოხილვას. ავტორები მიმოიხილავენ სხვადასხვა დიზაინის მეთოდებს, ალგორითმებს და სტრატეგიებს, რომლებიც გამოიყენება უიმ ფილტრების ოპტიმიზაციისთვის, რისთვისაც განიხილავენ როგორც თეორიულ, ასევე პრაქტიკულ ასპექტებს. გაზიარებული გამოცდილებით ვცადეთ დისერტაციასი აგვეწყო ფილტრის საუკეთესო დიზაინი სიგნალის ხმაურის შეფერხებისათვის.

თანამედროვე ტექნოლოგიები, მათ შორის **ხელოვნური ინტელექტი** და **მანქანური სწავლება**, გამოიყენება ციფრული სიგნალის დამუშავების პროცესების გაუმჯობესების მიზნით. **IEEE Transactions on signal Processing** და **signal Processing Journal** არის ჟურნალები, რომლებიც გთავაზობენ უახლეს კვლევებს ამ მიმართულებებში, სადაც სიგნალის აღდგენის ახალი მეთოდოლოგიები და ფილტრაციის ტექნოლოგიებია განხილული.

დისერტაციაში ჩატარებულ რიცხვით ექსპერიმენტებში ვცადეთ გაგვეერთიანებია ზემოთ ჩამოთვლილ ლიტერატურაში განხილული მიდგომები და გამოგვეყენებინა ელექტრონულ წრედებში გამავლი სიგნალის აღდგენის ალგორითმების გასაუმჯობესებლად.

I თავი. სიგნალები და სიგნალის სპექტრი

1.1. სიგნალის ცნება და კატეგორიები

ტერმინი „სიგნალი“ ხშირად გაიგივებულია „ინფორმაციასთან“. და „მონაცემებთან“, ეს ცნებები ერთმანეთთან მჭიდროდ არის დაკავშირებული და ერთმეორის გარეშე ვერ იარსებებენ, თუმცა ისინი მაინც სხვადასხვა კატეგორიებს მიეკუთვნებიან.

ინფორმაციის ცნება მრავალმხრივად განიმარტება, დაწყებული ზოგადი აღწერებით, როგორცაა "ინფორმაცია არის რეალური სამყაროს ფორმალიზებული ასახვა" და დამთავრებული პრაქტიკული განმარტებებით, როგორცაა "ინფორმაცია არის შეტყობინებები და მონაცემები, რომლებიც გამოიყენება შენახვის, გადაცემის, გარდაქმნის, აღქმისა და მართვის მიზნით". დღევანდელი მეცნიერება თანდათან უფრო მეტად თანხმდება იმაზე, რომ ინფორმაცია, მატერია და ენერჯიასთან ერთად, ბუნების ფუნდამენტურ ფილოსოფიურ კატეგორიას განეკუთვნება. ნორბერტ ვინერი, რომელიც სიგნალების თეორიის ერთ-ერთ დამფუძნებლად მიიჩნევა, აღნიშნავდა, რომ "ინფორმაცია არის თვითონვე ინფორმაცია და არა ენერჯი, არც მატერია" (Wiener 1948, 12-15). ამ კონტექსტში, მას შეგვიძლია მივაკუთვნოთ ობიექტური სამყაროს სპეციფიკური თვისებების რიგი.

მონაცემები წარმოადგენს დაკვირვების, ობიექტების გაზომვის ან პროცესების წარმომქმნელ ფაქტორთა ერთობლიობას, რომლებიც ფორმალიზებულია როგორც რაოდენობრივ, ასევე თვისებრივ სახით. ეს მონაცემები არის ნედლეული, რომელიც ინფორმაციის სახით გარდაიქმნება მას შემდეგ, რაც ისინი სათანადოდ დამუშავდება და ინტერპრეტირდება.

სიგნალი კი არის პროცესი, რომელიც ასახავს ობიექტის ფიზიკური მდგომარეობის ცვლილებას დროში და გამოიყენება შეტყობინების გადმოცემის, რეგისტრაციისა და გადაცემის მიზნით (Oppenheim & Schaffer 2009, 15-24).

ჩვენს გარშემო ცნობილი მარტივი მაგალითი, სიგნალების გადაცემისა და დამუშავების პროცესისა, არის დელფინების შემთხვევა. დელფინები იყენებენ ექოლოკაციას, რათა ნავიგაცია და საკვების მოპოვება შეძლონ. ისინი გამოსცემენ ხმის

ტალღებს წყალში, რომლებიც ირეკლებიან ობიექტებიდან, როგორცაა თევზები ან წყლის სხვა ბინადრები. ამ არეკლილ ტალღებს დელფინები აღიქვამენ და დამუშავების შედეგად ძალიან ზუსტად განსაზღვრავენ ობიექტის ზომას, მდებარეობას და მოძრაობის სისწრაფეს. მიღებული სიგნალის დამუშავების შედეგად, დელფინები ნადირობენ წარმატებულად ან საფრთხის იცილებენ თავიდან.

სიგნალი არის პროცესი, რომელიც ასახავს ერთი სიდიდის დამოკიდებულებას მეორეზე და მათემატიკური თვალსაზრისით ფუნქციას წარმოადგენს. ერთ-ერთი ყველაზე გავრცელებული სიგნალის მაგალითია ელექტრული სიგნალი, რომელიც გამოხატავს ძაბვის დამოკიდებულებას დროში.

ტერმინი „სიგნალი“ (წარმოშობით ლათინურ სიტყვაა „signum“, რაც „ნიშანს“ ნიშნავს) ხშირად გამოიყენება სხვადასხვა კონტექსტში. ამ განხილვის საფუძველზე, გთავაზობთ სიგნალის უფრო ზუსტ განსაზღვრას: **სიგნალი არის ინფორმაციული ფუნქცია, რომელიც შეიცავს მონაცემებს ფიზიკური სისტემის, ობიექტის ან გარემოს თვისებების, მდგომარეობის ან ქცევის შესახებ.** სიგნალების დამუშავების მიზანია სიგნალში არსებული ინფორმაციის (სასარგებლო ან მიზნობრივი) მოპოვება და ამ ინფორმაციის გარდაქმნა ისეთი ფორმით, რომელიც ადვილად აღსაქმელი და შემდგომი გამოყენებისთვის მოსახერხებელია (Proakis 2016, 23-29).

სიგნალის ანალიზი მოიცავს არა მხოლოდ მათემატიკურ გარდაქმნებს, არამედ ამ გარდაქმნების შედეგად შესაბამის პროცესებზე და ობიექტებზე დასკვნების გამოტანას. სიგნალების ანალიზის მიზნები მოიცავს:

- სიგნალის რიცხვითი მახასიათებლების გამოკვლევა ან განსაზღვრა, როგორცაა ენერგია, საშუალო სიმძლავრე, საშუალო კვადრატული გადახრა და სხვა;
- სიგნალის დაშლა ელემენტარულ კომპონენტებად, რათა შევადაროთ და შევაფასოთ განსხვავებული სიგნალის თვისებები;
- სხვადასხვა სიგნალების მსგავსებებისა და ურთიერთკავშირის იდენტიფიცირება და მათი რაოდენობრივი ანალიზი (Mitra 2010, 18-22).

სიგნალების ანალიზისთვის გამოყენებული მათემატიკური მეთოდები მრავალფეროვანია და ფართოდ გამოიყენება პრაქტიკაში. სწორედ ამ მეთოდების საფუძველზე იქმნება სიგნალების კვლევისა და გადმოცემის ელექტრონული

(მიკროპროცესორული) მოწყობილობები, რომლებიც შემდეგ ფართოდ გამოიყენება წარმოებასა და სხვა სფეროებში.

ტერმინი „სიგნალების რეგისტრაცია“ განუყოფლად არის დაკავშირებული სიგნალების ცნებასთან. ფართო გაგებით, ეს გულისხმობს სიგნალის გამოყოფასა და მის გარდაქმნას ისეთი ფორმით, რომელიც ხელსაყრელია მისი გამოყენებისთვის, დამუშავებისთვის და აღქმისთვის. ამ კონტექსტში, რეგისტრაცია ნიშნავს მონაცემთა ფიქსაციას სისტემაში ან მის კომპონენტში, რის შედეგადაც მონაცემები ინახება მყარ დისკებზე.

დასაწყისში აუცილებელია სიგნალების კლასიფიკაცია, რათა ჩამოვაცალიბოთ ზუსტი ტერმინოლოგია და განვსაზღვროთ კლასიფიკაციის კრიტერიუმები. სიგნალები შეიძლება დაიყოს როგორც უწყვეტ, ასევე დამოკიდებულ და დამოუკიდებელ ცვლადთა მიმდევრობად (Baraniuk 2018, 39-45):

- **ანალოგური სიგნალი:** ფიზიკური სიგნალი როგორც წესი, მათემატიკურად უწყვეტ ფუნქციას წარმოადგენს, ისინი განსაზღვრული არიან დროის მთელ ინტერვალში, შესაბამისად სიგნალის განმსაზღვრელი დამოუკიდებელი და დამოკიდებული ცვლადები უწყვეტია დროში. დროის გარკვეული მნაკვეთისთვის სიგნალს გააჩნია შესაბამისი ამპლიტუდა.

- **უწყვეტი დროის სიგნალი:** დროის ცვლადი უწყვეტია დროის იმ დიაპაზონში, რომელშიც განსაზღვრულია სიგნალი. თუ სიგნალის მნიშვნელობას გამოვსახავთ x -ით და თუ დროის ცვლადად შემოვიტანთ t -ს, მაშინ ასეთ სიგნალს აღვნიშნავთ როგორც $x(t)$

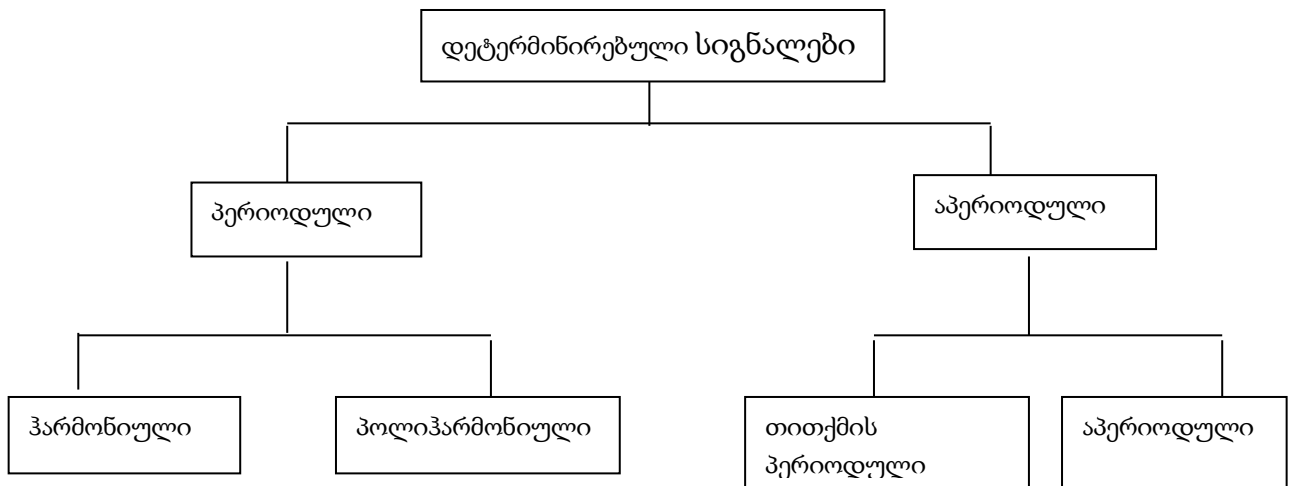
- **დისკრეტული დროითი სიგნალი (დისკრეტული სიგნალი):** დროის ცვლადი არის დისკრეტული დიაპაზონში, სადაც სიგნალია განსაზღვრული. თუ სიგნალის მნიშვნელობას გამოვსახავთ x -ით და დროის რეგულარულ ინტერვალებში ცვლადად შემოვიტანთ n -ს, სადაც $n = n'T$ მაშინ ასეთ სიგნალს აღვნიშნავთ როგორც $x(n)$. დისკრეტული დროის სიგნალი ასევე მოიხსენიება, როგორც გარდაქმნილი სიგნალი, ამ შემთხვევაში გარდაიქმნება უწყვეტი დროის სიგნალი. თითოეულ დროით მნიშვნელობას შეესაბამება დროის გარკვეული ამპლიტურა, სწორედ ეს ამპლიტუდებია დისკრეტული დროის სიგნალი.

- **ციფრული სიგნალი:** ესაა სიგნალი რომელიც დისკრეტულია დროში, ისე შეიძლება მისი წარმოდგენა როგორც დისკრეტული დროითი სიგნალისა. სიგნალები შეიძლება დავაჯგუფოთ როგორც მათი დამოკიდებული და დამოუკიდებელი ცვლადების მნიშვნელობების პროგნოზირების მიხედვით:

- **შემთხვევითი სიგნალი** - გარკვეული ალბათობით იღებს რაღაც მნიშვნელობებს, მისი ზუსტი მათემატიკური მოდელის შემნა თითქმის შეუძლებელია.

- **დეტერმინირებული სიგნალი** - შესაძლებელია მისი მნიშვნელობის დადგენა დროის ნებისმიერ მომენტში, ასევე შესაძლებელია მისი ზუსტი მათემატიკურის მოდელის წარმოდგენა.

დეტერმინირებული სიგნალების კლასიფიკაცია თავის მხრივ ასეთია :



დიაგრამა 1. დეტერმინირებული სიგნალების კლასიფიკაცია (ყუიზანიშვილი 2013)

შესაძლებელია სიგნალების მრავალმხრივი კლასიფიკაცია, თუმცა ნაშრომში განხილულია მხოლოდ დისკრეტულ დროით სიგნალები.

დისკრეტული სიგნალები თავის მხრივ იყოფა:

- **ერთგანზომილებიანი სიგნალები:** აღიწერებიან მხოლოდ ერთი დამოუკიდებელი და ერთი დამოკიდებული ცვლადით, დისკრეტული დროითი სიგნალი $x(n)$ არის ერთგანზომილებიანი სიგნალი, დროის ინტერვალი n დამოუკიდებელი ცვლადი და ამ ინტერვალში სიგნალის $x(n)$ ამპლიტუდა, დამოკიდებული ცვლადი.

- **ორგანზომილებიანი სიგნალები:** ორგანზომილებიანი სიგნალს აქვს ორი დამოუკიდებელი ცვლადი და ერთ დამოკიდებული ცვლადი და ერთ დამოკიდებული ცვლადი. მაგ: ორგანზომილებიანი სიგნალის მაგალითია ფოტოგრაფიული გამოსახულება სადაც n და m წერტილის კოორდინატებია, ხოლო $x(n, m)$ ნაცრისფერი დონეა ამ მდებარეობაზე.

- **სამგანზომილებიანი სიგნალები:** სამგანზომილებიანი სიგნალს აქვს სამი დამოუკიდებელი ცვლადი და ერთ დამოკიდებული ცვლადი. სამგანზომილებიანი სიგნალია ვიდეოსიგნალი, $x(n, m, \tau)$ სადაც წერტილის ადგილმდებარეობა (n, m) იცვლება τ დროში.

სისტემა, სადაც სიგნალი გადის შეიძლება კლასიფიცირდეს როგორც ანალოგური, დისკრეტული დროითი, ან ციფრული, იმის მიხედვით თუ რა ტიპის სიგნალს ატარებენ. დამუშავების მეთოდოლოგიის მხრივ დისერტაციაში არ გავნავასხვავებთ დისკრეტულ დროით სიგნალებს და ციფრულ სიგნალებს რადგან ისინი ერთნაირად მუშავდებიან.

ბუნებაში მიმდინარე მოვლენები წარმოქმნიან ანალოგურ სიგნალებს, მის დასამუშავებლად კი საჭიროა მისი გარდაქმნა ციფრულ სიგნალად. პროცესი, რომლის დროსაც ანალოგური სიგნალი ციფრულ სიგნალად გარდაიქმნება მოიცავს სიგნალის დისკრეტიზაციის და დაქვანტის პროცესს (Parker 2017, 17-18). როდესაც სიგნალის წარმოშობის ბუნება თეორიულად ძნელად აღსაქმელი და გასააზრებელია ან მისი მათემატიკური აღწერა ზედმეტად რთულდება, ხშირად მიმართავენ მათემატიკაში გავრცელებულ მეთოდებს, როგორცაა დაშლა მწკრივებად და აპროქსიმაციული მეთოდები.

სიგნალების შესაბამისი აღმწერი ფუნქციები იღებენ როგორც რეალურ, ისე წარმოსახვით მნიშვნელობებს კვლევების გამარტივების მიზნით. შესაბამისად, ხშირად გვხვდება როგორც რეალური, ასევე წარმოსახვითი სიგნალები, როგორც ჩვენს შემთხვევაში. მათემატიკური მოდელების გამოყენება საშუალებას გვაძლევს შევადაროთ ეს სიგნალები, გამოვავლინოთ მათი მსგავსებები და განსხვავებები და მოვახდინოთ მათი კლასიფიკაცია.

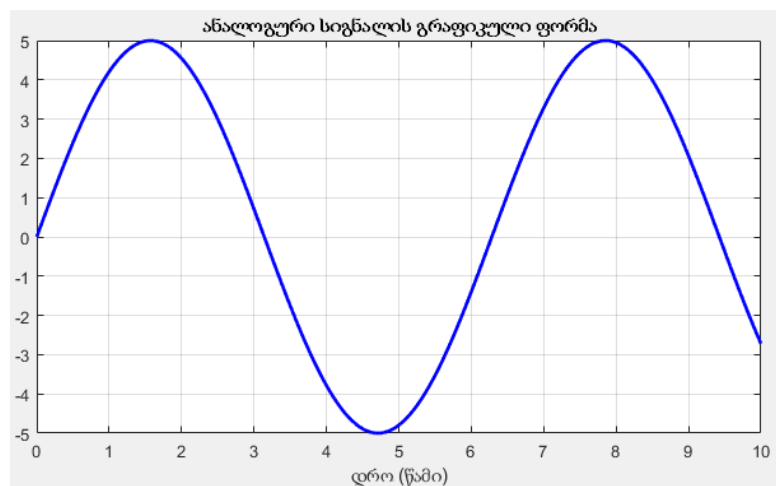
1.2. ანალოგური, დისკრეტული და ციფრული სიგნალები

ანალოგური სიგნალის გრაფიკული წარმოდგენა: ანალოგური სიგნალები არსებობენ უწყვეტად დროის გარკვეულ პერიოდში. განვიხილოთ ერთ-ერთი სინუსოიდალური სიგნალი, რომელიც შეიძლება წარმოვადგინოთ როგორც

$$y(t) = 5\sin(t) \quad 0 \leq t \leq 10 \quad (\text{Baraniuk 2010, 120})$$

მისი გრაფიკული გამოსახვა შესაძლებელია MATLAB-ში შემდეგი ბრძანებებით:

```
t = 0:0.01:10; % ქმნის t = [0 0.01 0.02 ... 9.99 10] ვექტორს
y = 5*sin(t); % ითვლის არგუმენტის მნიშვნელობებს t-ს ყოველი მნიშვნელობისთვის
plot(t,y, 'lineWidth', 2, 'Color', 'b'); % ხაზავს სინუსოიდას
title('ანალოგური სიგნალის გრაფიკული ფორმა');
xlabel('დრო (წამი)'); grid on; % აბსცისათა ღერძს უკეთებს წარწერას, ხაზავს
საკოორდინატო ბადეს
```



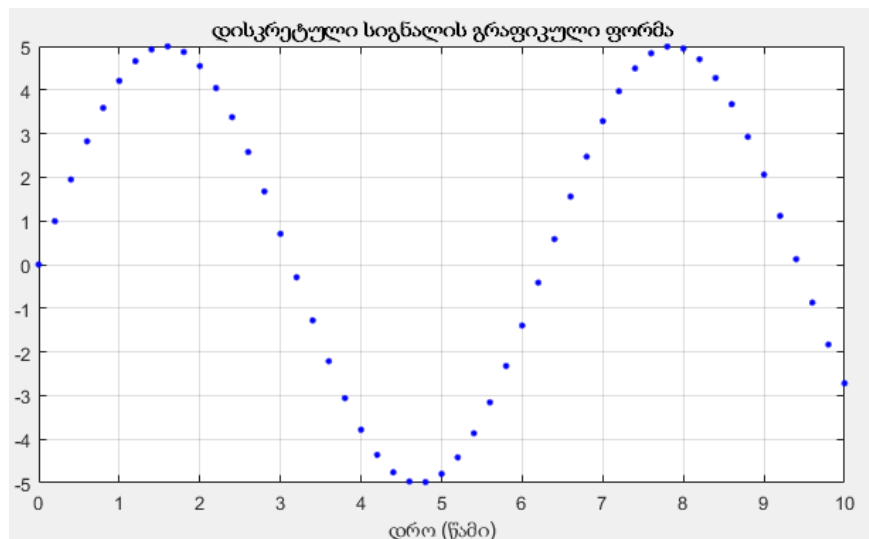
სურათი 1. ანალოგური ანალოგური სინუსოიდური სიგნალის ვიზუალიზაციას MATLAB-ში. სიგნალის ამპლიტუდა არის 5 (რაც გულისხმობს, რომ მისი მნიშვნელობა მერყეობს -5-დან +5-მდე). მოიცავს დაახლოებით 2 სრულ ციკლს 0-დან 10 წმ-მდე დროის მონაკვეთში. (წყარო-ავტორი)

დისკრეტული სიგნალის გრაფიკული წარმოდგენა: დისკრეტული სიგნალი მნიშვნელობებს იღებს დროის თანაბარ ინტერვალებში. განვიხილოთ ერთ-ერთი დისკრეტული სინუსოიდალური ტალღის სიგნალი, რომელიც განისაზღვრება შემდეგნაირად:

$$y(k) = 5\sin(0.2k) \quad k = 0,1,2, \dots, 50 \quad (\text{Baraniuk 2010, 125})$$

ეს სიგნალი იღებს მნიშვნელობებს ყოველ 0.2 წამში, დაწყებული 0 წამიდან და დასრულებული 10 წამზე. MATLAB-ში შემდეგი ბრძანებათა ნაკრები შექმნის ამ სიგნალის გრაფიკს.

```
k = 0:1:50; % k = [0 1 2 ... 50]
t = 0.2*k; % t = [0 0.2 0.4 ... 10]
y = 5*sin(0.2*k);
plot(t,y,'b','MarkerSize',10,'MarkerFaceColor','b'); % გაზრდილი და გამუქებული
ლურჯი წერტილები
title('დისკრეტული სიგნალის გრაფიკული ფორმა');
xlabel('დრო (წამი)');
grid on;
```



სურათი 2. დისკრეტული სინუსოიდური სიგნალის ვიზუალიზაციას MATLAB-ში. წინა ანალოგური სიგნალისგან განსხვავებით, ეს გრაფიკი აჩვენებს სიგნალის მნიშვნელობებს მხოლოდ გარკვეულ, დისკრეტულ დროის მომენტებში, რაც წარმოდგენილია ინდივიდუალური წერტილებით და არა უწყვეტი ხაზით. (წყარო-ავტორი)

ეს დისკრეტული სიგნალი, ზემოთ განხილული ანალოგური სიგნალიდან აღებული ნიმუშების ვერსიაა. ნიმუშები აღებულია ყოველ 0.2 წამში 10 წამის ინტერვალში. დისკრეტული სიგნალი ზუსტად ემთხვევა ანალოგურ სიგნალს ნიმუშების აღების დროებში. ანალოგური სიგნალის მნიშვნელობები, რომლებიც ნიმუშების აღების მომენტებს შორისაა, იკარგება. მაგალითად, ანალოგურ სიგნალს აქვს მნიშვნელობა 0.1 წამზე, მაგრამ დისკრეტულ სიგნალს არა.

ციფრული სიგნალის გრაფიკული წარმოდგენა: ციფრული სიგნალები არის სიგნალები, რომლებიც დროის თანაბრად დაშორებულ მომენტებში შეუძლიათ მიიღოს მხოლოდ სასრული რაოდენობის (ფინიტიური) მნიშვნელობები. ანალიზისთვის გამოვიყენოთ ზედა დისკრეტული სინუსოიდური სიგნალი.

$$y(k) = 5 \sin(0.2k) \quad k = 0, 1, 2, \dots, 50 \text{ (Baraniuk 2010, 126)}$$

დავუშვათ, $y(k)$ -ის მნიშვნელობები დამრგვალებულია უახლოეს მეთადამდე. ეს ქმნის ციფრულ სიგნალს, რომელსაც შეუძლია მიიღოს მხოლოდ სასრული რაოდენობის მნიშვნელობები: $-5; -4,9; -4,8; \dots 0; 0, 1; 0,2; \dots 4,9; 5$

ცხრილში ნაჩვენებია, როგორც დისკრეტული, ასევე ციფრული სიგნალის მნიშვნელობები დროის $[0; 2]$ წმ ინტერვალზე.

დრო (წამი)	დისკრეტული სიგნალი	ციფრული სიგნალი
0	0	0
0,2	0,99335	1
0,4	1,94709	1,9
0,6	2,82321	2,8
0,8	3,58678	3,6
1	4,20735	4,2
1,2	4,6602	4,7
1,4	4,92725	4,9
1,6	4,99787	5
1,8	4,86924	4,9
2	4,54649	4,5

ცხრილი 1. სინუსოიდური სიგნალის ანათვლები $[0; 2]$ წმ ინტერვალზე (წყარო-ავტორი).

როგორ წარმოადგენს ციფრული სიგნალი ნალოგურს? სრულად ნამდვილად ვერ წარმოადგენს, რადგან ციფრული სიგნალის მნიშვნელობები არსებობს მხოლოდ ანათვლების აღების მომენტებში: $0; 0,2; 0,4, \dots 9,8; 10$. ანალოგური სიგნალის

მნიშვნელობები ანათვლების ალების მომენტებს შორის დაკარგულია, ეს კი ინფორმაციის მნიშვნელოვანი დანაკარგია.

1.3. სიგნალის გენერირება

ერთ-ერთი ძალიან გავრცელებული ანალოგური სიგნალი არის სინუსოიდური ტალღა. სინუსოიდური ტალღა არის პერიოდული სიგნალი. პერიოდული სიგნალები არიან სიგნალები, რომლებიც დროის გარკვეული ინტერვალის (პერიოდის) შემდეგ მეორდება. სინუსოიდური ტალღის ზოგადი განტოლებაა:

$$y(t) = A\sin(2\pi ft + \theta) = A\sin(\omega t + \theta), \text{ (Smith 2013, 213). სადაც}$$

A = ამპლიტუდა (მაქსიმალური მნიშვნელობა)

f = სიხშირე (ჰერცი) = ციკლების რაოდენობა წამში

ω = სიხშირე (რადიანი/წამში)

θ = ფაზის გადაადგილება (რადიანში)

$T = 1/f =$ პერიოდი (წამი) = დრო, რომელიც საჭიროა ერთი სრული ციკლისთვის

გამოიყენეთ MATLAB, რათა თვალსაჩინოდ წარმოვადგინოთ 1250 ჰერცის სიხშირის სინუსოიდური ტალღის ოთხი ციკლი ამპლიტუდით 3.5 და ნულოვანი ფაზის გადაადგილებით, ასევე 1250 ჰერცის სიხშირის სინუსოიდალური ტალღის ოთხი ციკლი ამპლიტუდით 4.5 და -45° ფაზის გადაადგილებით.

$$y_1(t) = 3,5\sin(2 \cdot \pi(1250) \cdot t)$$

$$y_2(t) = 4,5\sin(2 \cdot \pi(1250)t - 45^\circ)$$

რამდენი დრო სჭირდება 1250 ჰერცის სინუსოიდალური ტალღის ოთხი ციკლის შესრულებას? 1250 ჰერცის სიგნალის პერიოდი განისაზღვრება შემდეგნაირად:

$$T = 1/1250 = 0.0008 \text{ წამი}$$

ოთხი ციკლი საჭიროებს $4 * 0.0008 = 0.0032$ წამს. შემდეგი MATLAB-ის ბრძანებათა ნაკრები დააგენერირებს საჭირო სიგნალებს:

```

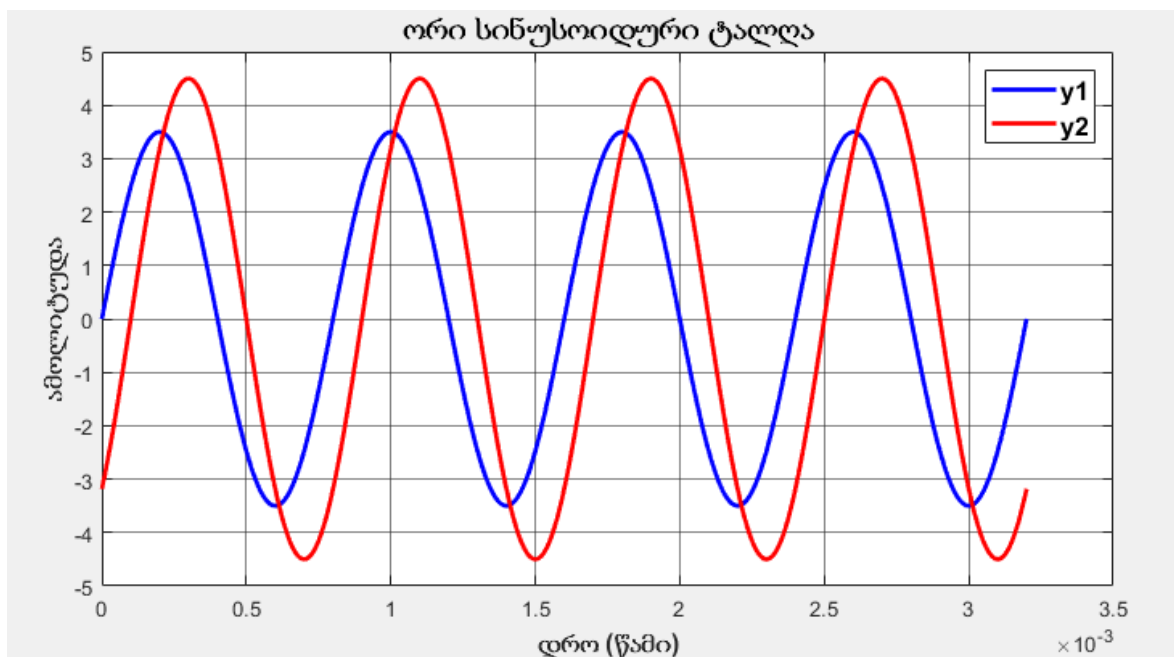
t = 0:1e-5:4*0.0008;
y1 = 3.5*sin(2*pi*1250*t);
y2 = 4.5*sin(2*pi*1250*t - 45*pi/180);

plot(t, y1, 'lineWidth', 2, 'Color', 'b');
hold on;
plot(t, y2, 'lineWidth', 2, 'Color', 'r');

hold off;

title('ორი სინუსოიდური ტალღა', 'FontSize', 14, 'FontWeight', 'bold');
xlabel('დრო (წამი)', 'FontSize', 12, 'FontWeight', 'bold');
ylabel('ამპლიტუდა', 'FontSize', 12, 'FontWeight', 'bold');
grid on;
set(gca, 'GridColor', [0.2 0.2 0.2], 'GridAlpha', 0.6);
legend('y1', 'y2', 'FontSize', 12, 'FontWeight', 'bold');

```



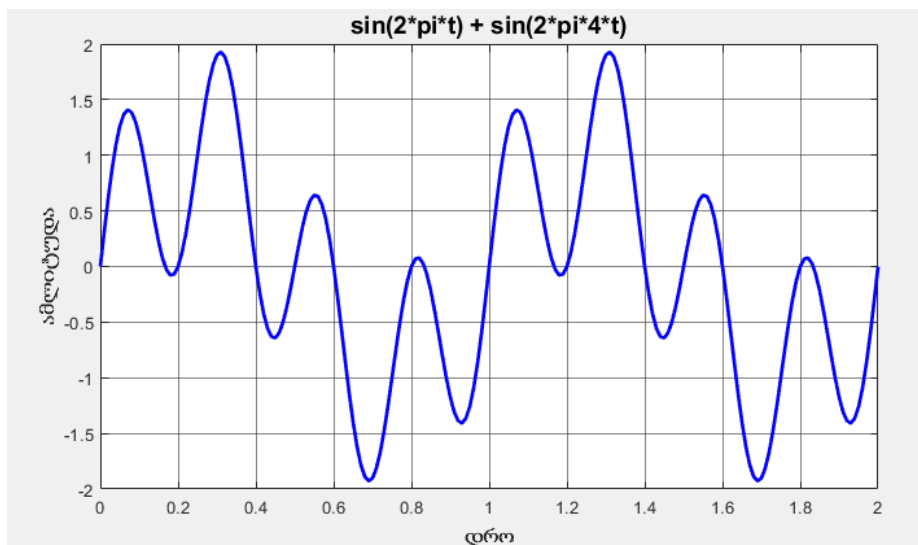
სურათი 3. ორი ანალოგური სინუსოიდური სიგნალის, y_1 (ლურჯი) და y_2 (წითელი), ერთობლივ ვიზუალიზაციას MATLAB-ში. (წყარო-ავტორი)

ცხადია ფაზის გადანაცვლება იწვევს დროის დაყოვნებას ორ სიგნალს შორის.

სინუსოიდური სიგნალების წრივი კომბინაცია: სინუსოიდური ტალღების წრფივი კომბინაციები წარმოქმნის პერიოდულ სიგნალებს. შევიკრიბოთ 2 სინუსოიდური ტალღა დავაკვირდეთ შედეგს:

ავაგოთ 1) $\sin(2\pi t) + \sin(2\pi(4)t)$

```
t = 0:0.01:2;
y = sin(2*pi*t) + sin(2*pi*4*t);
plot(t, y, 'lineWidth', 2, 'Color', 'b');
title('sin(2*pi*t) + sin(2*pi*4*t)', 'FontSize', 14, 'FontWeight', 'bold');
xlabel('დრო', 'FontSize', 12, 'FontWeight', 'bold');
ylabel('ამლიტუდა', 'FontSize', 12, 'FontWeight', 'bold');
grid on;
set(gca, 'GridColor', [0.2 0.2 0.2], 'GridAlpha', 0.6);
```



სურათი 4. ანალოგური სიგნალის ვიზუალიზაციას MATLAB-ში, რომელიც მიღებულია ორი სხვადასხვა სიხშირის სინუსოიდური სიგნალის შეკრებით. ესაა $\sin(2\pi t)$ და $\sin(2\pi(4)t)$ ფუნქციების ჯამი. (წყარო-ავტორი)

პირველი სინუსოიდური ტალღის სიხშირე არის 1 ჰერცი და შესაბამისად, მისი პერიოდი არის 1 წამი. მეორე სინუსოიდური ტალღის სიხშირე არის 4 ჰერცი და, შესაბამისად, მისი პერიოდი არის 1/4 წამი. ამგვარად, ორ წამიანი დროის შუალედი დაფარავს პირველი ტალღის 2 ციკლს და მეორე ტალღის 8 ციკლს, როგორც ეს სურათზეა ნაჩვენები.

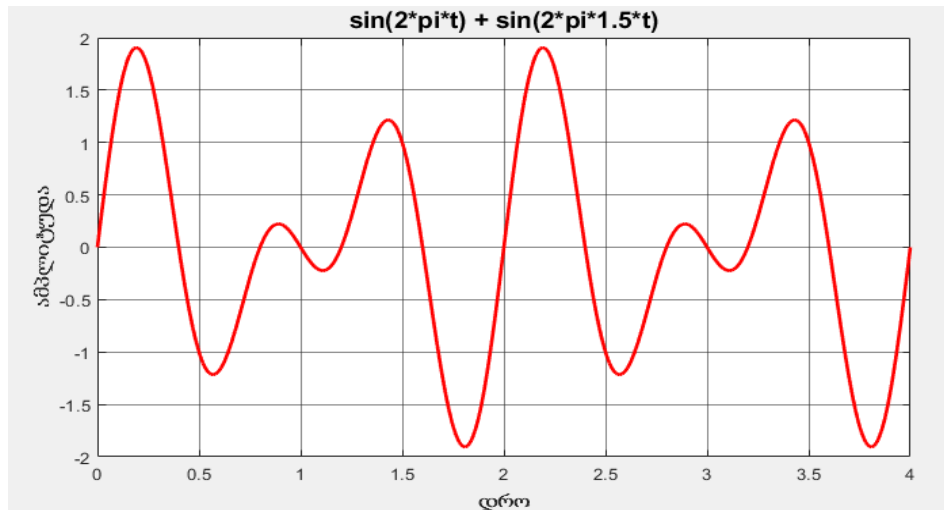
2) $\sin(2\pi t) + \sin(2\pi(1.5)t)$

პირველი სინუსოიდური ტალღის პერიოდი არის 1 წამი. მეორე სინუსოიდური ტალღის სიხშირე არის 1.5 ჰერცი და შესაბამისად, მისი პერიოდი არის 2/3 წამი. ამიტომ 4 წამის დროის შუალედი უნდა წარმოქმნას ოპტიმალური გრაფიკი.

```

t = 0:0.01:4;
y = sin(2*pi*t) + sin(2*pi*1.5*t);
plot(t, y, 'lineWidth', 2, 'Color', 'r');
title('sin(2*pi*t) + sin(2*pi*1.5*t)', 'FontSize', 14, 'FontWeight', 'bold');
xlabel('დრო', 'FontSize', 12, 'FontWeight', 'bold');
ylabel('ამპლიტუდა', 'FontSize', 12, 'FontWeight', 'bold');
grid on;
set(gca, 'GridColor', [0.2 0.2 0.2], 'GridAlpha', 0.6);

```



სურათი 5. ანალოგური სიგნალის ვიზუალიზაციას MATLAB-ში, რომელიც მიღებულია ორი სხვადასხვა სიხშირის სინუსოიდური სიგნალის შეკრებით. ესაა $\sin(2\pi t)$ და $\sin(2\pi 1.5t)$ ფუნქციების ჯამი. (წყარო-ავტორი)

რა თქმა უნდა წრფივი კომბინაციით მიღებული სიგნალი მკაფიოდ სინუსოიდური აღარაა თუმცა პერიოდული არის.

სინუსოიდური სიგნალების ნამრავლი: ორი სინუსოიდური ტალღის ნამრავლი კვლავ პერიოდული სიგნალია. ამას ასაბუთებს შემდეგი ტრიგონომეტრიული ფორმულა რომელიც გამოიყენება ნამრავლის დასათვლელად:

$$\sin(A)\sin(b) = 1/2[\cos(A - b) - \cos(A + b)] \quad (\text{Smith 2013, 219}).$$

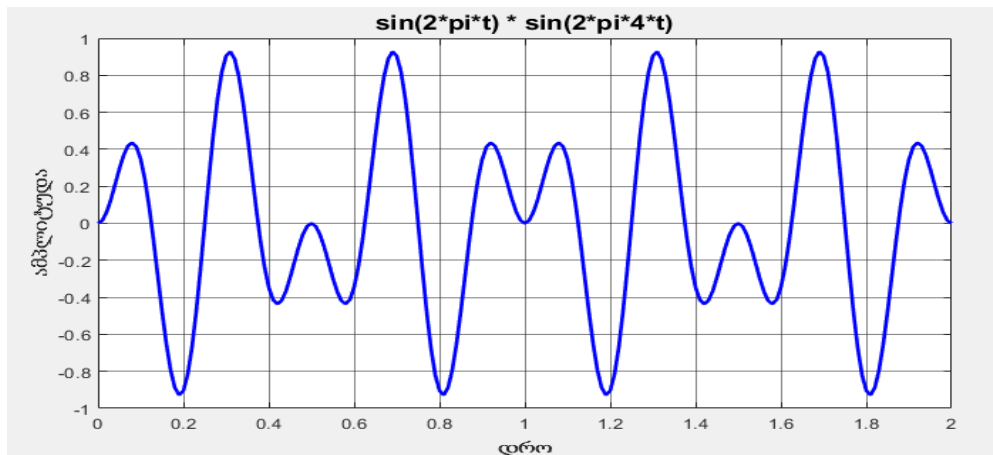
გამრავლების შემდეგ მიღებულ სიგნალზე აღარ აქვთ საწყისი სიხშირეები.

გავამრავლოთ ორი სიგნალი ერთმანეთზე $\sin(2\pi t) * \sin(2\pi(4)t)$

```

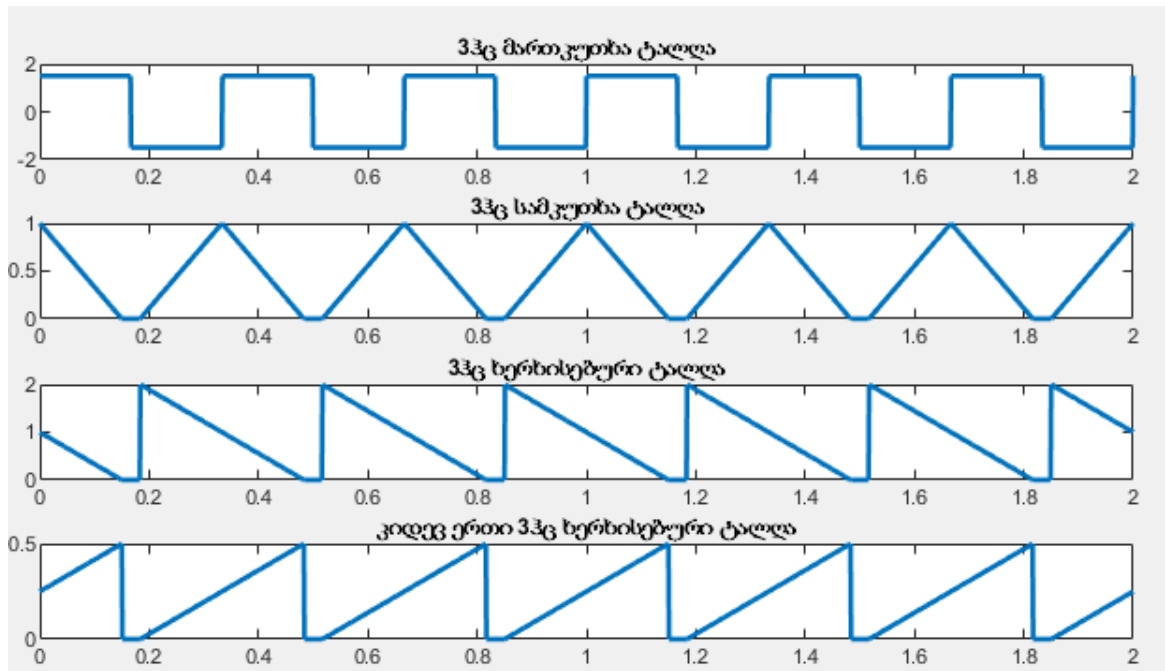
t = 0:0.01:2;
y = sin(2*pi*t).*sin(2*pi*4*t);
plot(t, y, 'lineWidth', 2, 'Color', 'b');
title('sin(2*pi*t) * sin(2*pi*4*t)', 'FontSize', 14, 'FontWeight', 'bold');
xlabel('დრო', 'FontSize', 12, 'FontWeight', 'bold');
ylabel('ამპლიტუდა', 'FontSize', 12, 'FontWeight', 'bold');
grid on;
set(gca, 'GridColor', [0.2 0.2 0.2], 'GridAlpha', 0.6);

```



სურათი 6. რთული ანალოგური სიგნალის ვიზუალიზაცია MATLAB-ში, რომელიც მიღებულია ორი სინუსოიდური სიგნალის ერთმანეთზე გამრავლებით. ესაა $\sin(2\pi t)$ და $\sin(2\pi 4t)$ ფუნქციების ნამრავლი. (წყარო-ავტორი)

სხვა პერიოდული სიგნალები: პერიოდული სიგნალები არის სიგნალები, რომლებიც გარკვეულ შუალედში (პერიოდში) განმეორებით წარმოიქმნება. სინუსოიდური ტალღები და სინუსოიდური ტალღების კომბინაციები პერიოდული სიგნალებია. სურათზე ნაჩვენებია სხვა პერიოდულ ტალღები, მათ შორის მართკუთხა ტალღები, ხერხისებური და სამკუთხა ტალღები.



სურათი 7. ოთხი განსხვავებული (მართკუთხა, სამკუთხა, 2 ხერხისებური) ანალოგური პერიოდული სიგნალის ვიზუალიზაციას MATLAB-ში. (წყარო-ავტორი)

```

t = linspace(0, 2, 2001); % დროის ვექტორი 0-დან 2-მდე, 2001 წერტილით
y_square = 1.5 * square(2 * pi * 3 * t); % 33კ მართკუთხა ტალღა
subplot(4,1,1);
plot(t, y_square, 'lineWidth', 2);
title('33კ მართკუთხა ტალღა');
pulse_locs = linspace(0, 2, 7); % განსაზღვრავს, სად დაიწყება თითოეული ახალი
იმპულსი pulstran-ში
y_triangular = pulstran(t, pulse_locs, @tripuls, 0.3, 0); % 33კ სამკუთხა ტალღა
subplot(4,1,2);
plot(t, y_triangular, 'lineWidth', 2);
title('33კ სამკუთხა ტალღა')
y_sawtooth = 2 * pulstran(t, pulse_locs, @tripuls, 0.3, -1); % 33კ ხერხისებური ტალღა,
განსხვავებული ამპლიტუდით
subplot(4,1,3);
plot(t, y_sawtooth, 'lineWidth', 2);
title('33კ ხერხისებური ტალღა')
y_sawtooth2 = 0.5 * pulstran(t, pulse_locs, @tripuls, 0.3, 1); % 33კ ხერხისებური ტალღა,
განსხვავებული ამპლიტუდით

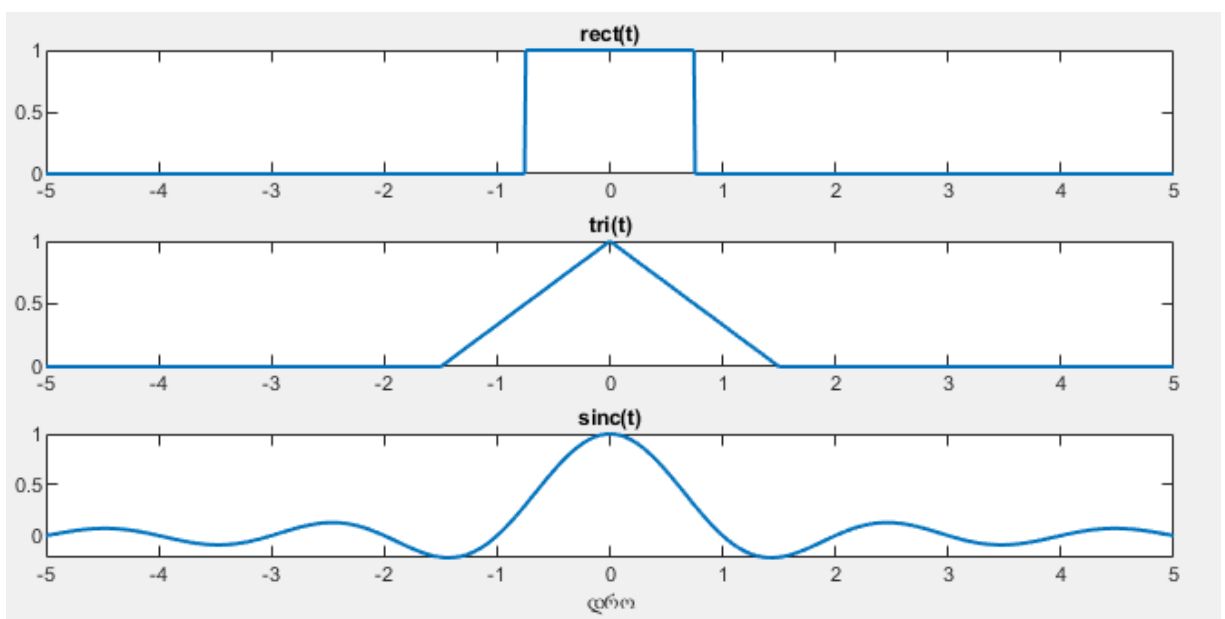
subplot(4,1,4);
plot(t, y_sawtooth2, 'lineWidth', 2);
title('33კ ხერხისებური ტალღა განსხვავებული ფორმით')

```

სხვა არაპერიოდული სიგნალები (ასევე ცნობილია როგორც აპერიოდული): არ აქვთ განმეორებადი ციკლები. ერთ-ერთი მაგალითი არაპერიოდული სიგნალისა არის *sinc* ფუნქცია, რომელიც განისაზღვრება შემდეგნაირად:

$$\text{sinc}(t) = \frac{\sin(\pi t)}{\pi t} \quad (\text{შენიშვნა : } \text{sinc}(0) = 1) \text{ (Smith 2013, 226).}$$

sinc ფუნქციას ხშირად იყენებენ ციფრული სიგნალის დამუშავების პროგრამებში და მას ჩენც ხშირად ვიყენებთ ამ ნაშრომშიც. რამდენიმე გავრცელებული არაპერიოდული სიგნალი, მათ შორის *sinc* ფუნქცია, ნაჩვენებია სურათში და მასთან ერთად მოცემულია MATLAB-ის კოდი, რომელიც გამოიყენება მისი ფორმის წარმოსადგენად.



სურათი 8. ანალოგური არაპერიოდული სიგნალების ვიზუალიზაცია MATLAB-ში. (წყარო-ავტორი)

```
t = linspace(-5, 5, 1000); % -5-დან 5-მდე წრფივად განაწილებული მნიშვნელობები
r = rectpuls(t, 1.5); % მართკუთხა იმპულსი სიდიდით 1.5
subplot(3,1,1);
plot(t, r, 'lineWidth', 1.5);
title('rect(t)');

tri = tripuls(t, 3); % სამკუთხა იმპულსი სიდიდით 3
subplot(3,1,2);
```

```
plot(t, tri, 'lineWidth', 1.5);
```

```

title('tri(t)');

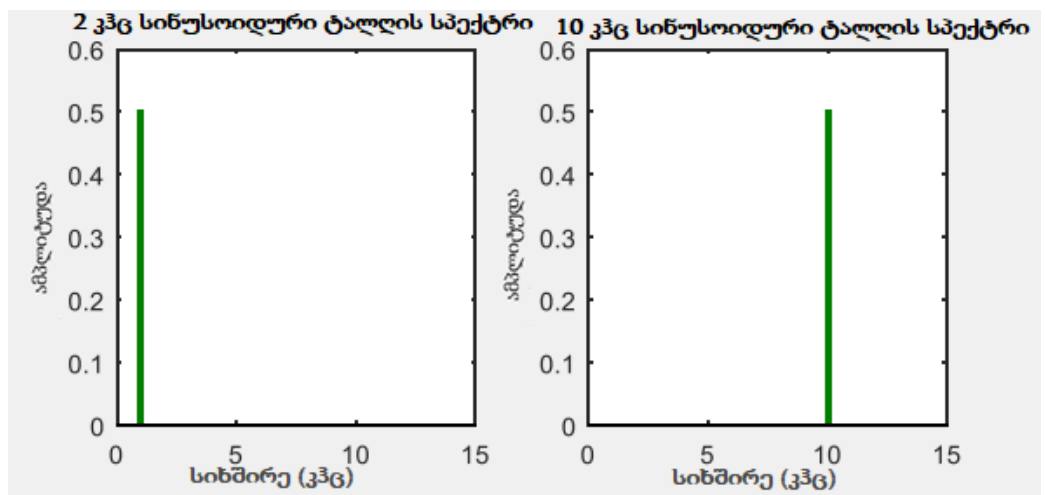
s = sinc(t); % sinc ფუნქცია
subplot(3,1,3);
plot(t, s, 'lineWidth', 1.5);
title('sinc(t)');
xlabel('დრო');

```

1.4. პერიოდული სიგნალების სპექტრი

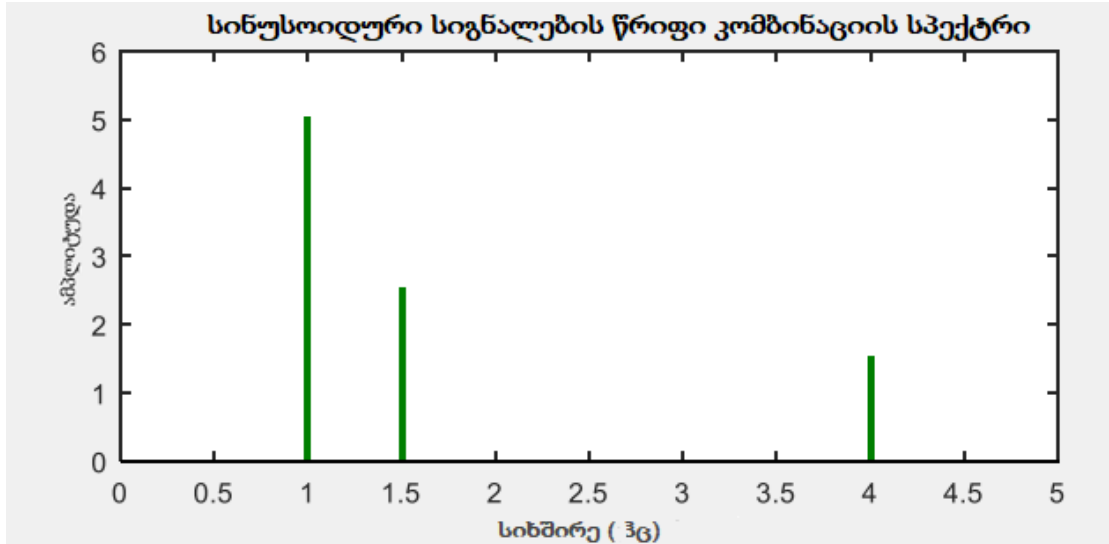
სიგნალები შეიძლება წარმოდგენილი იყოს დროით არეში (როგორც წინა შემთხვევებში) ან სიხშირულ არეში. სიგნალის სპექტრი აღწერს სიგნალის სიხშირულ შემადგენლობას. განვიხილოთ სინუსოიდური ტალღების, სინუსოიდური ტალღების კომბინაციების და სხვა პერიოდული სიგნალების სპექტრი.

სინუსოიდური ტალღები: სინუსოიდურ ტალღებს აქვთ მხოლოდ ერთი სიხშირული კომპონენტი, რომელიც მდებარეობს სინუსოიდური ტალღის სიხშირეზე. მაგალითად, 2 კჰც სინუსურ ტალღას აქვს სიხშირის კომპონენტი 2 კჰც-ზე, ხოლო 10 კჰც სინუსურ ტალღას აქვს სიხშირის კომპონენტი 10 კჰც-ზე, როგორც სურათზეა ნაჩვენები.



სურათი 9. ორი სხვადასხვა სინუსოიდური სიგნალის სიხშირული სპექტრის ვიზუალიზაცია. თითოეული ქვეგრაფიკი აჩვენებს სიგნალის ამპლიტუდას სიხშირული არეში (ჰერცებში) (წყარო-ავტორი).

სინუსოიდური სიგნალების წრივი კომბინაცია: როცა სინუსოიდური ტალღები ერთიანდება (იკრიბება), კომბინირებული სიგნალის სპექტრს აქვს სიხშირის კომპონენტები შესაკრები სინუსოიდური ტალღის სიხშირეზე. მაგალითად, სიგნალის სპექტრი $10\sin(2\pi(1)t) + 5\sin(2\pi(1,5)t) + 3\sin(2\pi(4)t)$ ნაჩვენებია სურათზე

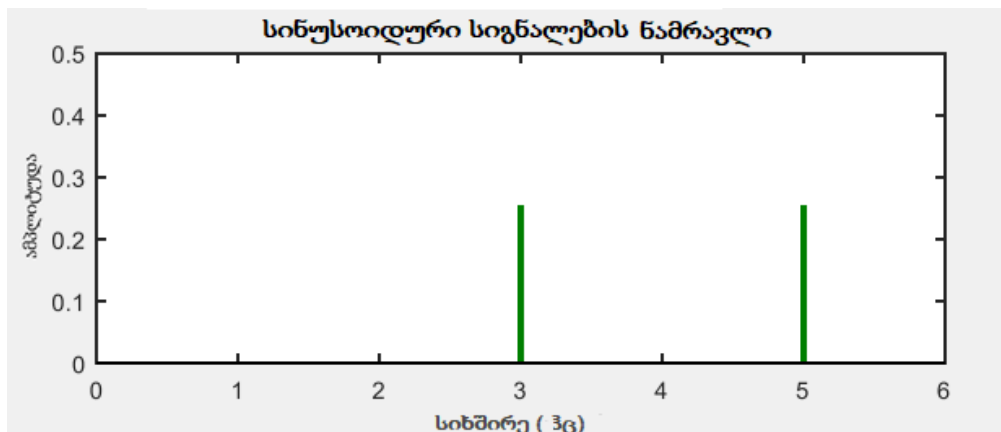


სურათი 10. სიგნალების წრივი კომბინაციის სპექტრი (წყარო-ავტორი)

სინუსოიდური სიგნალების ნამრავლი: ორი სინუსოიდური ტალღის ნამრავლის სპექტრი მაგ: $\sin(2\pi(1)t) \times \sin(2\pi(4)t)$ შესაკრებთა სიხშირეებიდან გამოითვლება. ამ მაგალითში შესაკრები სინუსოიდების სიხშირეებია 1 ჰც და 4 ჰც. მიღებული სიგნალის სპექტრი კი არის $4 - 1 = 3$ ჰც და $4 + 1 = 5$ ჰც

$$(\sin(A)\sin(b) = 1/2[\cos(A - b) - \cos(A + b)])$$

აქედან გამომდინარე:



სურათი 11. სინუსოიდური სიგნალების ნამრავლის სპექტრი (წყარო-ავტორი)

1.5. პერიოდული ფუნქციები ფურიეს მწკრივებად გაშლა

ჟოზეფ ფურიეს (Joseph Fourier) ჰიპოთეზა, რომელიც 1807 წელს წამოაყენა, სიგნალების დამუშავებისა და ანალიზის ფუნდამენტური პრინციპია. ის აღნიშნავდა, რომ ნებისმიერ პერიოდულ ფუნქცია, შესაძლებელია წარმოდგენილი იყოს ტრიგონომეტრიული მწკრივის სახით

მსოფლიოში ბევრი მოვლენაა, რომლებიც შეიძლება წარმოდგენილი იქნას სხვადასხვა სიხშირის ტალღების სახით. მაგალითად, განვიხილოთ მუსიკალური აკორდი. როცა გიტარაზე რამდენიმე სიმი ერთდროულად იკვრება, სხვადასხვა სიხშირის ხმა იბადება, რომლებიც ერთად ქმნიან ჰარმონიულ აკორდს. თითოეული ხმა სხვადასხვა სიხშირის ტალღაა და მათი კომბინაცია ქმნის მელოდიურ ჟღერადობას. ეს მუსიკალური მაგალითი კიდევ ერთხელ ამტკიცებს ფურიეს ჰიპოთეზის სისწორეს, რადგან ის აჩვენებს, რომ კომპლექსური ხმა შეიძლება დაიშალოს სხვადასხვა სიხშირის ტალღებად, რომლებიც ერთობლივად ქმნიან მთლიანობის შეგრძნებას. ანუ მუსიკა, ესაა სხვადასხვა სიგრძის ტალღების ჯამი.

ამრიგად, სიგნალის დაშლა სხვადასხვა სიხშირის ტალღებად საშუალებას გვაძლევს გამოვიკვლიოთ მოცემული სიგნალის სტრუქტურა, ანუ სხვა სიტყვებით, შეგვიძლია მივიღოთ ინფორმაცია სიგნალის შიდა აგებულებაზე. ასეთი ანალიზის მეთოდად სპექტრული ანალიზი, ანუ **ფურიეს ანალიზი** გამოიყენება.

ყველა პერიოდული სიგნალი შეიძლება სრულყოფილად აღიწეროს სინუსური და კოსინუსური ტალღების ჯამის სახით, ფურიეს მწკრივების გამოყენებით. ვთქვათ, $f_p(t)$ არის პერიოდული სიგნალი პერიოდით T და სიხშირით $f_0 = \frac{1}{T}$. ნებისმიერი $f_p(t)$ პერიოდული სიგნალისთვის ფურიეს მწკრივით მოიცემა შემდეგნაირად (Parker, 2017):

$$f_p(t) = a_0 + \sum_{n=1}^{\infty} a_n \cos(2\pi(nf_0)t) + \sum_{n=1}^{\infty} b_n \sin(2\pi(nf_0)t)$$

$$a_0 = \frac{1}{T} \int_0^T f_p(t) dt$$

$$a_n = \frac{2}{T} \int_0^T f_p(t) \cos(2\pi(nf_o)t) dt$$

$$b_n = \frac{2}{T} \int_0^T f_p(t) \sin(2\pi(nf_o)t) dt \text{ (Parker, 2017):}$$

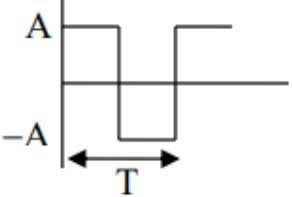
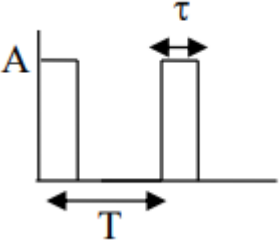
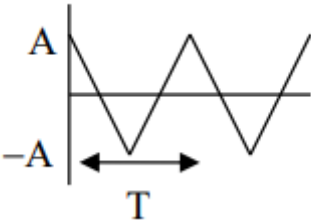
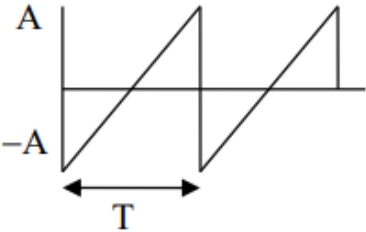
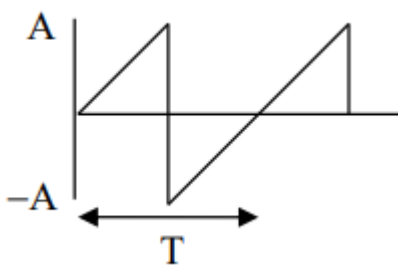
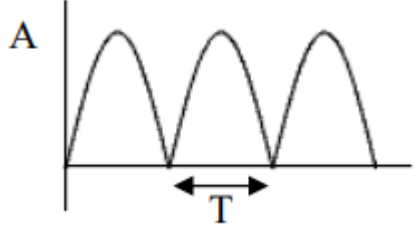
პირველი ფორმულის მიხედვით პერიოდული სიგნალი შესაძლებელია წარმოვადგინოთ სინუსოიდური და კოსინუსოიდური ტალღების ჯამით, რომელითა სიხშირეები არის ძირითადი ტალღის სიხშირის ჯერადი. ძირითადი სიხშირე ეს არის სიგნალის ყველაზე დაბალი სიხშირე, რომელსაც სიგნალი შეიცავს. მას ასევე უწოდებენ **პირველი ჰარმონიკას**. ანუ **ჰარმონიკები**, არიან სიხშირეები, რომლებიც არის ძირითადი სიხშირის ჯერადები. მაგალითად, თუ ძირითადი სიხშირე არის f_0 მაშინ მეორე ჰარმონიკა იქნება $2f_0$ მესამე ჰარმონიკა $3f_0$ და ასე შემდეგ.

ფორმულაში მუდმივი წევრია (a_0), ეს არის ტალღის საშუალო მნიშვნელობა, ასევე ცნობილი როგორც სიგნალის ის ნაწილი, რომელიც დროში არ იცვლება. ფურიეს მწკრივში მას არ აქვს არც ერთი სიხშირული კომპონენტი.

(a_n და b_n) განსაზღვრავენ სინუსოიდური და კოსინუსოიდური ტალღების ამპლიტუდებს, რომლებიც ფურიეს მწკრივში შედიან. უფრო ზუსტად a_n წარმოადგენს კოსინუსოიდური ტალღების ამპლიტუდას მე- n ჰარმონიკულ სიხშირეზე, ხოლო b_n წარმოადგენს სინუსოიდური ტალღების ამპლიტუდას მე- n ჰარმონიკულ სიხშირეზე. ფურიეს მწკრივი და მისი სიხშირული კომპონენტები საშუალებას გვაძლევს ავხსნათ და გავანალიზოთ პერიოდული სიგნალის შიდა სტრუქტურა. სიხშირული კომპონენტებით დგინდება ასევე რამდენად იძლევა **კორელაციას** (შესაბამისობას) პერიოდული სიგნალი შესაბამის კოსინოსუიდურ და სინოსოიდურ ტალღებთან.

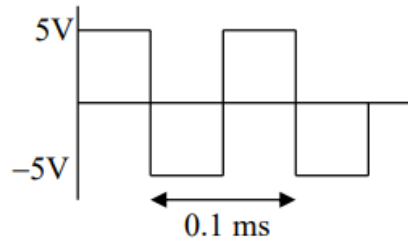
ცხრილში მოცემულია ფურიეს მწკრივის ფორმულები რამდენიმე ტიპური პერიოდული ფუნქციისთვის, რაც გვაძლევს ვიზუალურ წარმოდგენას, როგორ გამოიყურება ამ ტიპის სიგნალების ფურიეს განშლა. კიდევ ერთხელ დავრწმუნდებით, რომ სხვადასხვა ტიპის სიგნალებს სხვადასხვა ტალღის ფორმა შეესაბამება და განსხვავებული სიხშირის კომპონენტები აქვთ, რომლებიც ფურიეს მწკრივით შეიძლება იქნეს წარმოდგენილი (Parker 2017, 600-607):.

პერიოდული ტალღის ფორმა	ფურიეს მწკრივად გაშლა
---------------------------	-----------------------

	$\frac{4A}{\pi} \left[\sin(2\pi f_0 t) + \frac{1}{3} \sin(2\pi(3f_0)t) + \frac{1}{5} \sin(2\pi(5f_0)t) + \dots \right]$
	$Ad + \frac{A}{\pi} \{ [\sin(2\pi d) \cos(2\pi f_0 t) + 2\sin^2(\pi d) \sin(2\pi f_0 t)] + \frac{1}{2} [\sin(2\pi(2d)) \cos(2\pi(2f_0)t) + 2\sin^2(\pi(2d)) \sin(2\pi(2f_0)t)] + \frac{1}{3} [\sin(2\pi(3d)) \cos(2\pi(3f_0)t) + 2\sin^2(\pi(3d)) \sin(2\pi(3f_0)t)] + \dots \}$ <p>სადაც $d = \tau/T$</p>
	$\frac{8A}{\pi^2} \left[\cos(2\pi f_0 t) + \frac{1}{3^2} \cos(2\pi(3f_0)t) + \frac{1}{5^2} \cos(2\pi(5f_0)t) + \dots \right]$
	$-\frac{2A}{\pi} \left[\sin(2\pi f_0 t) + \frac{1}{2} \sin(2\pi(2f_0)t) + \frac{1}{3} \sin(2\pi(3f_0)t) + \dots \right]$
	$\frac{2A}{\pi} \left[\sin(2\pi f_0 t) - \frac{1}{2} \sin(2\pi(2f_0)t) + \frac{1}{3} \sin(2\pi(3f_0)t) - \dots \right]$
	$\frac{2A}{\pi} - \frac{4A}{\pi} \left[\frac{1}{4 \cdot 1^2 - 1} \cos(2\pi(f_0)t) + \frac{1}{4 \cdot 2^2 - 1} \cos(2\pi(2f_0)t) + \frac{1}{4 \cdot 3^2 - 1} \cos(2\pi(3f_0)t) + \dots \right]$

ცხრილი 2. სიგნალები და სიგნალთა გამლა ფურის მწკრივებად (Parker 2017, 607):

უფრო დეტალურად განვიხილოთ მართკუთხა სიგნალის აღსაწერად ფურიეს მწკრივები. სურათის მიხედვით $A = 5$ და $T = 0.1$ მილიწამი. შესაბამისად სიხშირე არის $f_0 = 1/T = 10\text{კჰც}$. ფურიეს მწკრივად გაშლაში პირველი წევრია $\frac{20}{\pi} \sin(2\pi(10,000)t)$



სურათი 12. ანალოგური მართკუთხა ტალღის ფორმის ვიზუალურ წარმოდგენა, მითითებულია მისი ამპლიტუდა და პერიოდი (Parker 2017, 608):

```
t = linspace(0, 3*0.1e-3, 300); % 300 წერტილი, რაც შეესაბამება 100 წერტილს ერთ ციკლში, სულ 3 ციკლი
y1 = (20/pi)*sin(2*pi*1e4*t);
subplot(3,1,1); plot(t, y1);
title('ფურიეს მწკრივის პირველი წევრი'); grid on;
```

ამპლიტუდა A და სიხშირე f_0 ზუსტად ისეთივეა, როგორც ზემოთ გამოვიანგარიშეთ.

ფურიეს მწკრივის პირველი სამი წევრი იქნება:

$$\left(\frac{20}{\pi}\right) \left[\sin(2\pi(10,000)t) + \left(\frac{1}{3}\right) \sin(2\pi \cdot 3 \cdot (10,000)t) + \left(\frac{1}{5}\right) \sin(2\pi \cdot 5 \cdot (10,000)t) \right]$$

MATLAB-ში გრაფიკის აგება კი შესაძლებელია შემდეგი კოდით:

```
y2 = (20/pi)*(sin(2*pi*3*1e4*t)/3);
y3 = (20/pi)*(sin(2*pi*5*1e4*t)/5);
subplot(3,1,2); plot(t, y1 + y2 + y3);
title('ფურიეს მწკრივის პირველი სამი წევრი'); grid on;
```

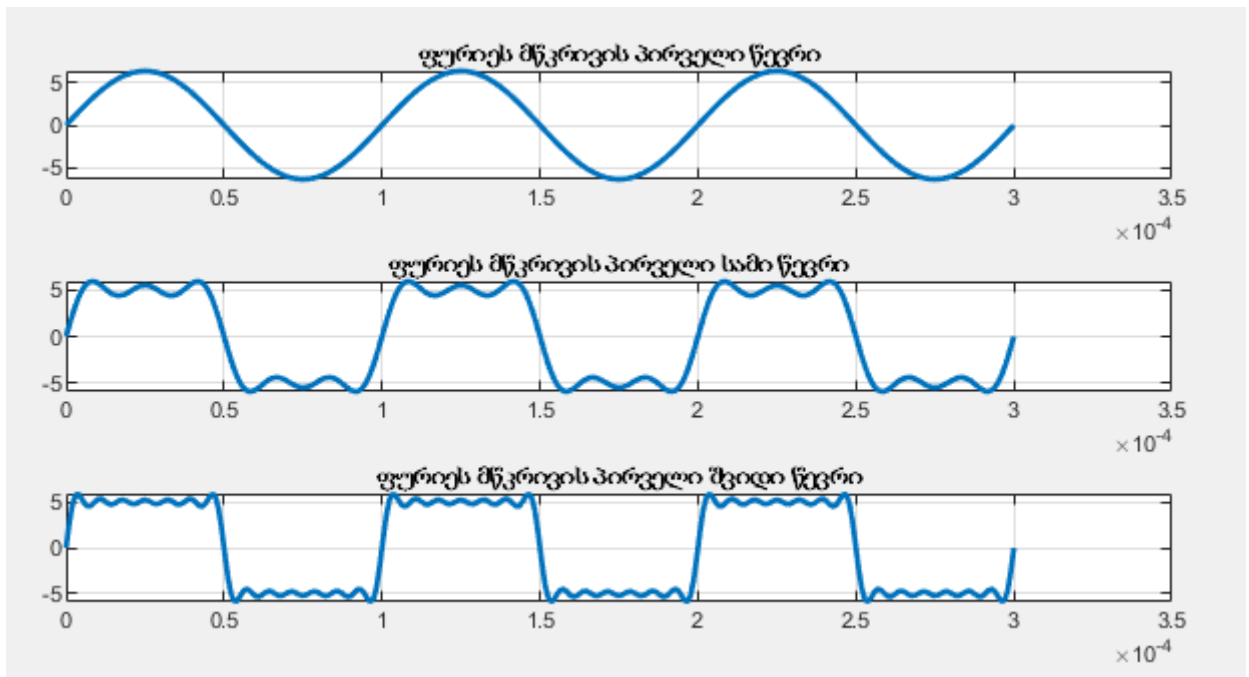
ფურიეს მწკრივის პირველი შვიდი წევრი შეიძლება წარმოვადგინოთ შემდეგნაირად:

$$\left(\frac{20}{\pi}\right) \left[\sin(2\pi(10000)t) + \left(\frac{1}{3}\right)\sin(2\pi \cdot 3 \cdot (10000)t) + \left(\frac{1}{5}\right)\sin(2\pi \cdot 5 \cdot (10000)t) + \right. \\ \left. \left(\frac{1}{7}\right)\sin(2\pi \cdot 7 \cdot (10000)t) + \left(\frac{1}{9}\right)\sin(2\pi \cdot 9 \cdot (10000)t) + \left(\frac{1}{11}\right)\sin(2\pi \cdot 11 \cdot (10000)t) + \right. \\ \left. \left(\frac{1}{13}\right)\sin(2\pi \cdot 13 \cdot (10000)t) \right]$$

ხოლო მისი წარმოდგენა MATLAB-ში შესაძლებელია შემდეგო კოდით:

```
y4 = (20/pi)*(sin(2*pi*7*1e4*t)/7);
y5 = (20/pi)*(sin(2*pi*9*1e4*t)/9);
y6 = (20/pi)*(sin(2*pi*11*1e4*t)/11);
y7 = (20/pi)*(sin(2*pi*13*1e4*t)/13);
subplot(3,1,3); plot(t, y1 + y2 + y3 + y4 + y5 + y6 + y7);
title('ფურიეს მწკრივის პირველი შვიდი წევრი'); grid on;
```

შედეგი ნაჩვენების ქვემოთ, რაც უფრო მეტ წევრს ვამატებთ ფურიეს მწკრივში, ვიზუალური ფორმა უფრო მეტად შეესაბამება მართკუთხა ტალღას, რომელიც დასაწყისი გვექონდა მოცემული.



სურათი 13. მართკუთხა ტალღის ფურიეს მწკრივად გაშლის ვიზუალიზაცია. იგი აჩვენებს, თუ როგორ შეიძლება კომპლექსური (არასინუსოიდური) პერიოდული სიგნალი წარმოდგენილი იქნას სხვადასხვა სიხშირისა და ამპლიტუდის სინუსოიდური ტალღების (ჰარმონიკების) ჯამის სახით (წყარო-ავტორი).

1.6. არაპერიოდული სიგნალების სპექტრი

ფურიეს მწკრივად გაშლა არ მუშაობს არაპერიოდული სიგნალებისთვის. არაპერიოდული სიგნალები შედგება სინუსოიდური სიხშირეების უწყვეტი ნაკადისგან, რომლებსაც აქვთ ცვლადი ამპლიტუდა და ფაზა. არაპერიოდული ანალოგური სიგნალების სპექტრის ანალიზისთვის შეიძლება გამოყენებულ იქნას ფურიეს გარდაქმნა, რომლის შესაბამისი მათემატიკური წარმოდგენაა

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt = \int_{-\infty}^{\infty} f(t)[\cos(\omega t) - j\sin(\omega t)]dt \quad (\text{Parker 2017, 611):}$$

ავიღოთ არაპერიოდული სიგნალი .

$$f(t) = e^{-at} \text{ for } t \geq 0 \text{ და } f(t) = 0 \text{ სადაც } t < 0$$

ვიპოვოთ მისი ფურიეს გარდაქმნა:

$$\begin{aligned} F(\omega) &= \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt = \int_0^{\infty} e^{-at} \cdot e^{-j\omega t} dt = \int_0^{\infty} e^{-(a+j\omega)t} dt = \frac{-1}{a+j\omega} e^{-(a+j\omega)t} \Big|_{t=0}^{t=\infty} = \\ &= \frac{1}{a+j\omega} \end{aligned}$$

$$F(f) = \frac{1}{a + j2\pi f}$$

პერიოდული სიგნალი, როგორც ეს მართკუთხა სიგნალის შემთხვევაში გვქონდა, შესაძლებელია წარმოვადგინოთ სინუსოიდური და კოსინუსოიდური ტალღების ჯამით, რომელთა სიხშირეები არის ძირითადი ტალღის სიხშირის ჯერადი. ძირითადი სიხშირე ეს არის სიგნალის ყველაზე დაბალი სიხშირე, თუ მისი მნიშვნელობა არის 10 კჰც, მაშინ ჰარმონიკები არის ძირითადი სიხშირის ჯერადები:

10 კჰც, 30 კჰც, 50 კჰც, 70 კჰც და ა.შ.

არაპერიოდულ, ექსპონენციალურ სიგნალს აქვს სიხშირის კომპონენტები სიხშირეების უწყვეტ დიაპაზონში. თუ ვივარაუდებთ, რომ $a = 5$ მაშინ ჩვენ გვაქვს:

$$F(0,13\text{კ}) = \frac{1}{(5 + j(2\pi \cdot 0,1))} = 0,1984 < -7,2^\circ$$

$$F(13\text{კ}) = \frac{1}{(5 + j(2\pi \cdot 1))} = 0,1245 < -51,5^\circ$$

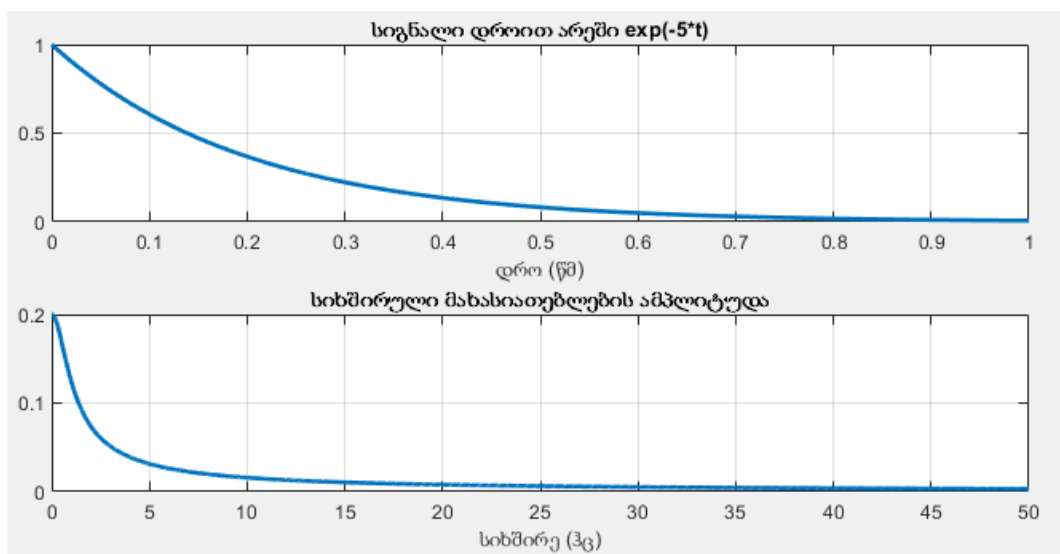
$$F(103\text{კ}) = \frac{1}{(5 + j(2\pi \cdot 10))} = 0,0159 < -81,5^\circ$$

$$F(1003\text{კ}) = \frac{1}{(5 + j(2\pi \cdot 100))} = 0,0016 < -89,5^\circ$$

გათვალისწინებულია, რომ ამპლიტუდა მცირდება სიხშირის ზრდასთან ერთად.

MATLAB - ში ავსგოთ არაპერიოდული სიგნალი დროით არეში და დავაკვირდეთ მისი სიხშირის ამპლიტუდასაც.

```
t = 0:0.001:1;
f = exp(-5*t);
subplot(2,1,1);
plot(t, f, 'lineWidth', 2);
title('სიგნალი დროით არეში exp(-5*t)'); grid; xlabel('დრო (წმ)');
freq = 0:0.001:50;
F = 1./(5 + j*2*pi*freq);
Mag = abs(F);
subplot(2,1,2);
plot(freq, Mag, 'lineWidth', 2);
title('სიხშირული მახასიათებლების ამპლიტუდა'); grid;
xlabel('სიხშირე (კვ)')
```



სურათი 14. დროითი სიგნალი და სიგნალის სპექტრი (წყარო-ავტორი)

ფურიეს გარდაქმნა არის ერთ-ერთი მეთოდი ანალოგური სიგნალის სპექტრის განსაზღვრისთვის, იგი ფურიეს ანალიზის სპეციფიკური მათემატიკური ოპერაციაა, რომელიც სიგნალს დროითი არედან (ან სივრცითი არედან) სიხშირულ არეში გარდაქმნის. ეს ოპერაცია აჩვენებს, რომელი სიხშირული კომპონენტები არსებობს სიგნალში და როგორია მისი ამპლიტუდა. ფურიეს გარდაქმნა გამოიყენება, როცა სიგნალი არ არის პერიოდული ან როცა საჭიროა მისი სპექტრის განსაზღვრა. ფურიეს გარდაქმნის მეშვეობით მიღებული სპექტრი უზრუნველყოფს სიგნალის დეტალურ სიხშირულ ანალიზს, რაც საშუალებას გვაძლევს დავინახოთ, თუ რომელი სიხშირული კომპონენტები დომინირებენ სიგნალში და როგორია მათი გავლენა (Parker 2017, 645). ეს განსაკუთრებით მნიშვნელოვანია ისეთ სიგნალებში, რომლებიც შეიცავენ სხვადასხვა სიხშირულ კომპონენტებს, რომლებიც ერთმანეთშია შერეული, მაგალითად ხმოვან სიგნალები ან ელექტრომაგნიტურ ტალღები. ფურიეს გარდაქმნის ეფექტურობა დამოკიდებულია კონკრეტული სიგნალის ბუნებაზე. მაგალითად, დისკრეტული სიგნალების შემთხვევაში გამოიყენება დისკრეტული ფურიეს გარდაქმნა (DFT), რომელიც ციფრულ სიგნალებს გარდაქმნის სიხშირულ არეში. ეს განსაკუთრებით მნიშვნელოვანია თანამედროვე ციფრულ ტექნოლოგიებში, სადაც მონაცემთა დამუშავება ხდება ციფრულ გარემოში.

ფურიეს გარდაქმნა ინტეგრალის გამოვლით ხდება, ზოგიერთ სიგნალზე ფურიეს ინტეგრალის გამოთვლ ძალიან რთულია ხელით, ამიტომ პრაქტიკაში ხშირად გამოიყენება ფურიეს გარდაქმნის ცხრილები და რიცხვითი მეთოდები, როგორც ეს წარმოდგენილა დანართ 1-ში. ჩვენ განვიხილეთ $f(t) = e^{-at}$ for $t \geq 0$ და $f(t) = 0$ სადაც $t < 0$ სიგნალის გარდაქმნა, ინტეგრირება, თუმცა შევხვდებით ისეთ სიგნალებს რომელზედაც ფურიეს გარდაქმნა ვერ იმუშავებს, სიგნალის რთული მათემატიკური მოდლის გამო. ამიტომ საჭირო ხდება სხვა, უფრო მოწინავე მეთოდების გამოყენება, მაგალითად როგორცაა ფურიეს სწრაფი გარდაქმნა (FFT). ეს გარდაქმნაც გამოიყენება ანალოგური ან ციფრული სიგნალის სპექტრის განსაზღვრისთვის მაგრამ არ სჭირდება ინტეგრალის გამოთვლა. იგი იყენებს მარტივ პროგრამირებად გამოთვლებს რომელსაც ქვემოთ განვიხილავთ.

დროითი სიგნალი: $f(t)$	$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt = \int_{-\infty}^{\infty} f(t)e^{-j2\pi ft} dt$
$\delta(t)$	1
$\delta(t - \tau)$	$e^{-j\omega\tau}(1) = e^{-j2\pi f\tau}$
$u(t) = \begin{cases} 0 & t < 0 \\ 1 & t > 0 \end{cases}$	$\pi\delta(\omega) + \frac{1}{j\omega} = \pi\delta(2\pi f) + \frac{1}{j2\pi f}$
$u(t - \tau)$	$e^{-j\omega\tau} \left[\pi\delta(\omega) + \frac{1}{j\omega} \right] = e^{-j2\pi f\tau} \left[\pi\delta(2\pi f) + \frac{1}{j2\pi f} \right]$
$e^{-at}u(t) a > 0$	$\frac{1}{j\omega + a} = \frac{1}{j2\pi f + a}$
$te^{-at}u(t) a > 0$	$\frac{1}{(j\omega + a)^2} = \frac{1}{(j2\pi f + a)^2}$
$\text{rect}\left(\frac{t}{\tau}\right) = \begin{cases} 1 & t \leq \frac{\tau}{2} \\ 0 & t \geq \frac{\tau}{2} \end{cases}$	$\tau \cdot \text{sinc}\left(\frac{\tau\omega}{2\pi}\right) = \tau \cdot \text{sinc}(\tau f)$
$\text{sinc}(t) = \frac{\sin(\pi t)}{\pi t}$	$\text{rect}\left(\frac{\omega}{2\pi}\right) = \text{rect}(f)$
$\cos(\omega_0 t) = \cos(2\pi f_0 t)$	$\pi[\delta(\omega + \omega_0) + \delta(\omega - \omega_0)] = \pi[\delta(f + f_0) + \delta(f - f_0)]$
$\sin(\omega_0 t) = \sin(2\pi f_0 t)$	$j\pi[\delta(\omega + \omega_0) - \delta(\omega - \omega_0)] = j\pi[\delta(f + f_0) - \delta(f - f_0)]$
$\cos(\omega_0 t + \theta)$	$\pi[e^{-j\theta}\delta(\omega + \omega_0) + e^{j\theta}\delta(\omega - \omega_0)]$

ცხრილი 3. ფურიეს გარდაქმნები (Parker 2017, 650).

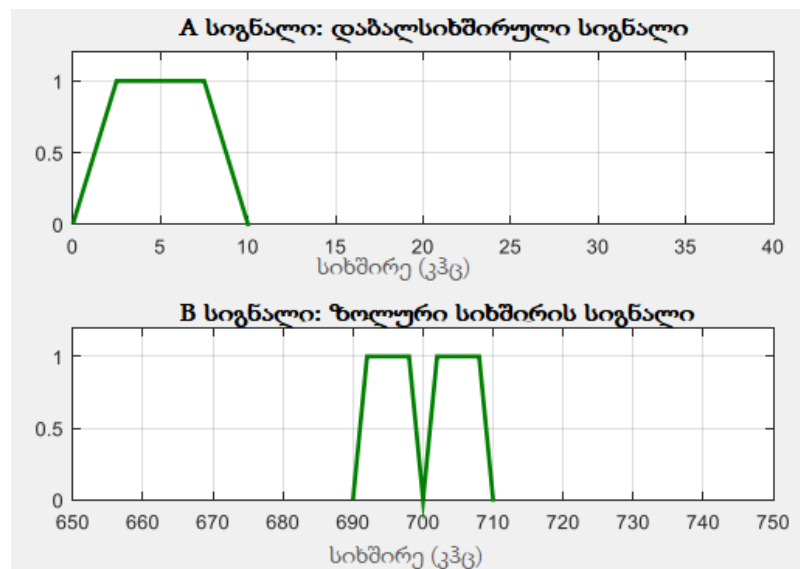
თვისებები	ფუნქცია	ფურიეს გარდაქმნა
წრფივობა	$a_1 f_1(t) + a_2 f_2(t)$	$a_1 F_1(\omega) + a_2 F_2(\omega)$
დროში დაყოვნება	$f(t - \tau)$	$e^{-j\omega\tau} F(\omega)$
კონვოლუცია	$f(t) * g(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau$	$F(\omega)G(\omega)$
დროითი მასშტაბი	$f(at)$	$\frac{1}{ a } F\left(\frac{\omega}{a}\right)$

დუალურობა (ორმაგობა)	$F(t)$	$2\pi \cdot f(-\omega)$
სიხშირის ცვლილება	$e^{j\omega_0 t} f(t)$	$F(\omega - \omega_0)$
კორელაცია	$\int_{-\infty}^{\infty} f(\tau - t)g(\tau)d\tau$	$F(-\omega)G(\omega)$

ცხრილი 4. ფურიეს გარდაქმნის თვისებები (Parker 2017, 652-653).

1.7. სიგნალის სიხშირის დიაპაზონი

სიგნალის სიხშირის დიაპაზონი არის ის სივრცე, რომელსაც სიგნალი იკავებს სიხშირის არეში. სურათი აჩვენებს ორი განსხვავებული სიგნალის სპექტრს, რომელთა სიხშირის დიაპაზონი განსხვავებულია. სიგნალი A არის ბაზური სიგნალი ან დაბალსიხშირული სიგნალი. მას აქვს სიხშირის კომპონენტები 0 კილოჰერციდან 10 კილოჰერცამდე და შესაბამისად იკავებს 10 კილოჰერცის სივრცეს სიხშირულ არეში. ამდენად, A სიგნალის სიხშირის დიაპაზონი არის 10 კილოჰერცი. B სიგნალი ზოლური სიხშირის სიგნალია და აქვს სიხშირის კომპონენტები 690 კილოჰერციდან 710 კილოჰერცამდე. ის იკავებს 20 კილოჰერცის სივრცეს სიხშირულ არეში და შესაბამისად მისი სიხშირის დიაპაზონი არის 20 კილოჰერცი.



სურათი 15. ორი განსხვავებული სიხშირის სიგნალის სიხშირის დიაპაზონი (წყარო-ავტორი).

დავაკვირდეთ პირველი სიგნალის სპექტრს. მისი სიხშირული კომპონენტების ამპლიტუდა მცირეა თითქმის 0-მდე, ასეთ შემთხვევებში შესაძლებელია სიგნალი გრძელდებოდეს სიხშირულ არეში უსასრულოდ. პრაქტიკაში, უსასრულო სიხშირის დიაპაზონის მქონე სიგნალების დამუშავება შეუძლებელია, რადგან რეალურ სისტემებს არ შეუძლიათ უსასრულო სიხშირის დიაპაზონის დამუშავება. ამიტომ საჭიროა უფრო პრაქტიკული განსაზღვრება, „სიხშირის დიაპაზონი“ ხშირად მოიხსენიებენ ამ ტერმინით იმ დიაპაზონს, სადაც სიგნალის ენერჯის ან ამპლიტუდის უმეტესი ნაწილია კონცენტრირებული. ეს ნიშნავს, რომ ჩვენ ვიკვლევთ იმ სიხშირეების დიაპაზონს, სადაც სიგნალის ენერჯია არის ყველაზე მაღალი და ვაიგნორებთ იმ სიხშირეებს, სადაც სიგნალის ენერჯია უმნიშვნელოა.

მაშასადამე პრაქტიკული სიხშირის დიაპაზონი არის საზომი, რომელიც საკმარისად კარგად ასახავს ორიგინალ სიგნალს, თუ სპექტრის დანარჩენ ნაწილს ამოვჭრით, სიგნალი მნიშვნელოვნად არ შეიცვლება. არსებობს პრაქტიკული სიხშირის დიაპაზონის განსაზღვრის რამდენიმე განსხვავებული გზა - განვიხილოთ ორი ყველაზე გავრცელებული:

3 (დბ) დეციბალი სიხშირის დიაპაზონი: ეს არის სიხშირის დიაპაზონის საზომი, რომელიც გვუბნობს განვსაზღვროთ სიგნალის ან სისტემის ეფექტური სიხშირის დიაპაზონი, სადაც ის ინარჩუნებს საკმარისად მაღალ ამპლიტუდას და მუშაობს ოპტიმალურად. მაგ: ვთქვათ სიგნალის მაქსიმალური ამპლიტუდა არის 10 ვოლტი. ეს ნიშნავს, რომ სიგნალი აღწევს თავის უმაღლეს ამპლიტუდას 10 ვოლტზე. მაშინ 3 დეციბალი სიხშირის დიაპაზონში სიგნალის ამპლიტუდა შემცირდება 70.7 % -ით და გახდება 7.07 ვოლტი და ამაზე დაბლა არ უნდა დაეცეს. კარგად გამოიყენება ფილტრებისთვის და იმ სიგნალებისთვის, რომელთა სპექტრი მკვეთრად იკლებს 3 დბ სიხშირის გარშემო, რაც საშუალებას იძლევა სწრაფად შეფასდეს სისტემის ეფექტურობა.

სიგნალის ძალის სიხშირის დიაპაზონი: არის სიხშირის დიაპაზონი, რომელიც მოიცავს სიგნალის ენერჯის ძირითადი ნაწილის (ჩვეულებრივ 95 % -დან 98 % -მდე). ამ დიაპაზონში სიგნალის ენერჯის უდიდესი ნაწილია კონცენტრირებული, ხოლო დანარჩენი სიგნალის ენერჯია ძალიან მცირეა ან ნაკლებად მნიშვნელოვანია.

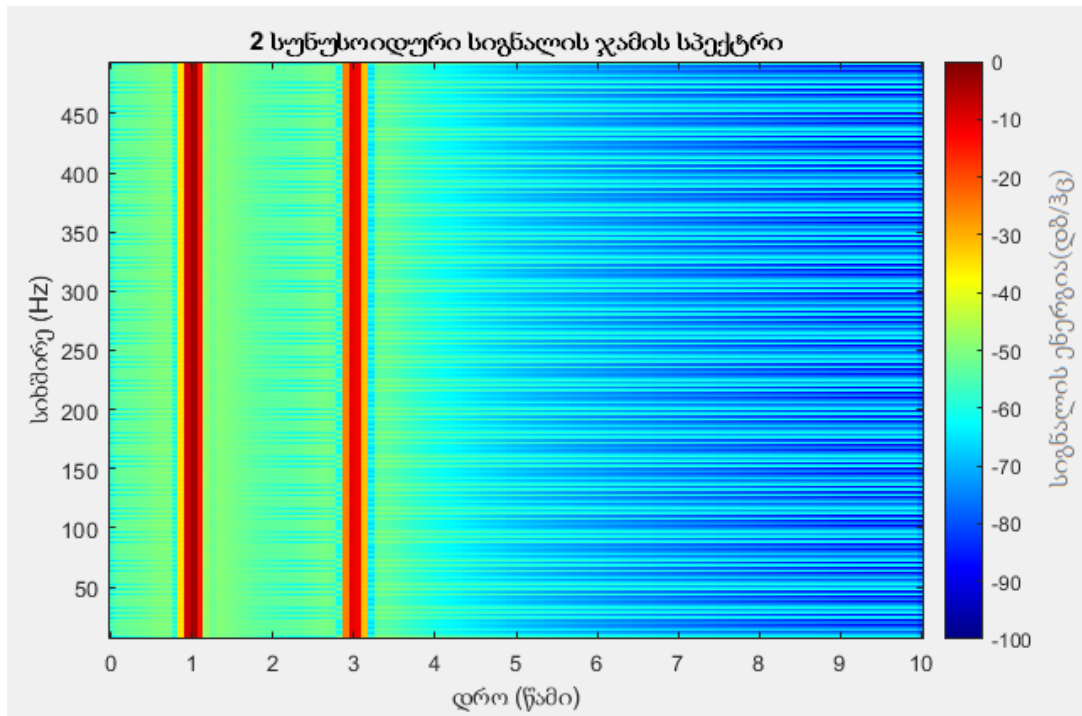
მაგალითად, თუ სიგნალის ენერჯის 98 % არის 0-დან 10 კილოჰერცამდე სიხშირის დიაპაზონში, ეს დიაპაზონი ითვლება სიგნალის ძალის სიხშირის დიაპაზონად.

1.8. სპექტოგრამა

აუდიო, მუსიკისა და ვიდეო სიგნალები დროთა განმავლობაში იცვლება, შესაბამისად სპექტრიც (სიხშირული შემადგენლობაც) იცვლება დროში. **სპექტოგრამა** არის გრაფიკი, რომელიც აჩვენებს სიგნალის სიხშირულ კომპონენტებს დროის გარკვეულ პერიოდში. ის საშუალებას გვაძლევს დავინახოთ, როგორ იცვლება სიგნალის სიხშირული შემადგენლობა და ამპლიტუდა (Akan 2022, 89-97). იმისათვის, რომ ეს სამი განზომილება (სიხშირე, დრო და ამპლიტუდა) გამოვსახოთ 2-განზომილებიან გრაფიკზე, სიხშირე გადავზომოთ x -ღერძზე, დრო y -ღერძზე, ხოლო სიხშირის კომპონენტების შედარებითი ძალა წარმოვადგინოთ ფერებით (ან მონოქრომიული სკალით). ფერთა სქემის გამოყენება საშუალებას გვაძლევს დავინახოთ სიგნალის სიხშირის კომპონენტების შედარებითი ძალა, რაც აუდიო და ვიდეო სიგნალების ანალიზში ძალიან სასარგებლოა. გრაფიკზე წითელი ფერი მიუთითებს ძალიან ძლიერ სიხშირულ კომპონენტზე, ხოლო ლურჯი ფერი მიუთითებს ძალიან სუსტ სიხშირულ კომპონენტზე.

ავაგოთ $y = 10 \cdot \sin(2 \cdot \pi \cdot 1000 \cdot t) + 0,1 \cdot \sin(2 \cdot \pi \cdot 3000 \cdot t)$, სიგნალის სპექტოგრამა 0,5 წმ-ის ინტერვალში.

```
Fs = 20000;  
t = 0:1/Fs:0.5;  
y = 10*sin(2*pi*1000*t) + 5*sin(2*pi*3000*t);  
spectrogram(y,256,240,256,Fs);  
colormap('jet');  
caxis([-100 0]);  
title('2 სუნუსოიდური სიგნალის ჯამის სპექტრი');  
xlabel('დრო (წამი)');  
ylabel('სიხშირე (Hz)');
```

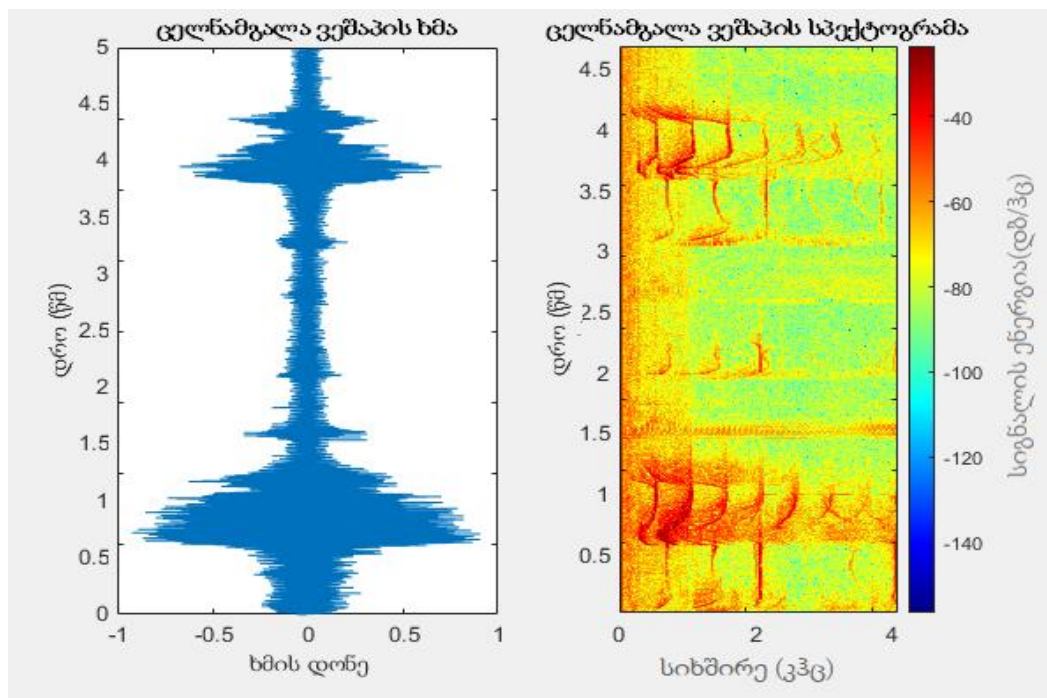


სურათი 16. სპექტროგრამა, რომელიც აჩვენებს სიხშირული კომპონენტების ცვლილებას დროში (წყარო-ავტორი).

ძალიან საინტერესო სპექტროგრამა ნაჩვენებია სურათზე. დამუშავებულია ცელნამგალა ვეშაპის ხმის სიგნალი, რომელიც ხელმისაწვდომია მისამართზე <https://www.scuba.lu/sound/ngsorca03.wav> სურათის მარცხენა მხარე ასახავს ცელნამგალა ვეშაპის ხმის სიგნალის ჩანაწერს. მარჯვენა მხარეს კი ნაჩვენებია ამ ხმის სპექტროგრამა. სპექტროგრამაზე ჩანს როგორ იცვლება სიხშირის კომპონენტები და მათი ინტენსივობა დროთა განმავლობაში. მაგალითად 1,75 -დან 3 წმ წამის განმავლობაში სიგნალის ამპლიტუდა დაბალია, რაც სპექტროგრამაზე აისახება მხოლოდ დაბალი სიმძლავრის სიხშირის მომწვანო-მოლურჯო ფერებით, ხოლო 0,5 დან 1 წმ შუალედში სიგნალის ენერგია მაღალია და წითელი ფერითაა სპექტოგრამაზე გამოსახული.

```
[y, Fs] = audioread('orca.wav');
subplot(1,2,2);
spectrogram(y,512,256,512,Fs);
colormap('jet');
title('ცელნამგალა ვეშაპის სპექტოგრამა');
t = 0:1/Fs:1/Fs*(length(y)-1);
subplot(1,2,1); plot(y,t);
```

```
title('ცელნამგალა ვეშაპის ხმა ');  
xlabel('ხმის დონე'); ylabel('დრო (წმ)')
```

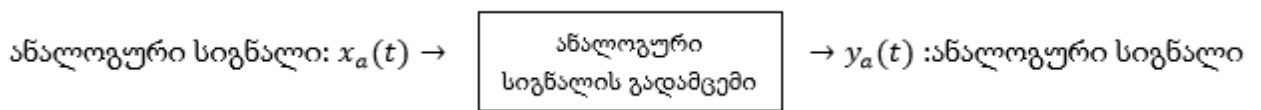


სურათი 17.ცელნამგალა ვეშაპის ხმის სპექტოგრამა (წყარო-ავტორი)

II თავი. სიგნალის დისკრეტიზაცია და რეკონსტრუქცია

2.1. სიგნალის ციფრული დამუშავების ზოგადი სტრუქტურა

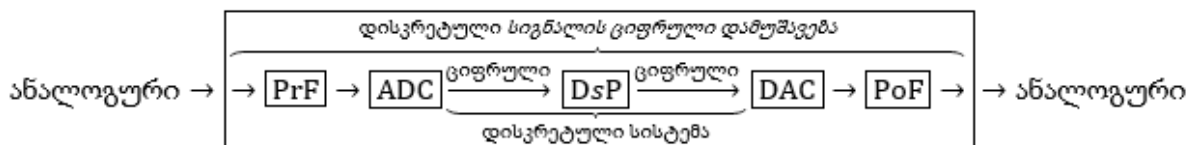
სიგნალები, რომლებსაც პრაქტიკაში ვხვდებით, ძირითადად ანალოგური სიგნალები არიან. ეს სიგნალები განსხვავდებიან დროში და განსხვავებული ამპლიტუდები აქვთ, ისინი გადაიცემიან ელექტრონული ქსელის საშუალებით, ელექტრონული აქტიური და პასიური წრედის ელემენტებით. ეს მიდგომა ცნობილია, როგორც ანალოგური სიგნალის დამუშავება (AsP) (ასდ)-მაგალითად რადიო და სატელევიზიო მიმღებები.



დიაგრამა 2. ანალოგური სიგნალის დამუშავების ალგორითმი (Semmlow 2022, 3).

ისინი აგრეთვე შეიძლება დამუშავდნენ ციფრული აპარატურის გამოყენებით, მულტიპლიკატორებით და სპაციალური მიკროპროცესორების საშუალებით. მათ სჭირდებათ გარდაქმნა, რომელსაც ასრულებს ციფრული ტექნიკა. ასეთ გარდაქმნილ სიგნალს უკვე ეწოდება ციფრული სიგნალი. ის იღებს სასრული რიცხვის სახეს დროში, და აქედან გამომდინარე შეგვიძლია წარმოვადგინოთ ბინარული რიცხვით (ორობითი) ან ბიტებით. ციფრული დამუშავების არსი, როგორც უკვე აღვნიშნეთ, არის ის, რომ ფიზიკური სიგნალი გარდაიქმნება დისკრეტული რიცხვების თანმიმდევრობაში და შემდეგ ხდება მათი მათემატიკური გარდაქმნები. დისკრეტული სიგნალის უკუგარდაქმნით შესაძლებელია საწყისი სიგნალის აღდგენა.

სიგნალების ციფრული დამუშავების განზოგადებული სტრუქტურული სქემა წარმოდგენილია შემდეგ ნახაზზე.



დიაგრამა 3. სიგნალების ციფრული დამუშავების განზოგადებული სტრუქტურული სქემა (Tabeshadze 2019).

PrF-წინასწარი მომზადების ფილტრი, რომელიც წინასწარ ამზადებს სიგნალს შემდგომი დამუშავებისთვის ან გადაცემისთვის, მისი ძირითადი ფუნქციაა ხმაურის შემცირება, სიგნალის სპექტრის ფორმირება, კოდირებისა და დეკოდირების პროცესების გამარტივება და სიგნალის მორგება არხის სპეციფიკურ მოთხოვნებზე. ამ ფილტრის მთავარი ფუნქციაა ანალოგური სიგნალის წინასწარ გაფილტვრა, რათა შემცირდეს არასასურველი სიხშირული კომპონენტები, რომლებმაც შეიძლება გამოიწვიონ ზედდების ეფექტი (Aliasing) ADC-ის პროცესის დროს. PrF ფილტრი ჩვეულებრივ არის დაბალსიხშირული ფილტრი (low-Pass Filter), რომელიც ამცირებს სიგნალის მაღალ სიხშირის კომპონენტებს, რომლებიც აღემატება ნაიქვისტის სიხშირეს.

ADC-ანალოგურ ციფრული გარდამქმნელი, რომელიც ქმნის ბინარული რიცხვების ნაკადს ანალოგური სიგნალიდან. სიტემას შესავალში მიეწოდება ანალოგური სიგნალი. ანალოგურ-ციფრული გარდამქმნელის მეშვეობით ხდება სიგნალის ანთვლების აღება (sampling), დაქვანტვა (Quantization) და კოდირება (Encoding). მიუხედავად იმისა, რომ ეს სამი პროცესი დისკრეტიზაცია, დაქვანტვა და კოდირება დამოუკიდებლად წარმოებს, ისინი ჩვეულებრივ ერთ მიკროსქემაშია გაერთიანებული. **ADC** -ის მიერ მიღებული სიგნალი წარმოადგენს $x_a(t)$ რიცხვების თანმიმდევრობას, რომელიც იგზავნება ციფრულ პროცესორში, სადაც მათი შესაბამისი დამუშავება ხდება.

DSP-Digital signal Processor: ციფრული სიგნალი დამუშავება. ამ პროცესს ახორციელებს ზოგადად კომპიუტერი, ან სპეციალური პროცესორი, ან ციფრული აპარატურა და ა.შ. ამ ეტაპზე, ციფრული სიგნალი შეიძლება გაანალიზდეს, გაიფილტროს, შეიკუმშოს, გააუმჯობესდეს ან სხვა მანიპულაციები ჩატარდეს. DSP იყენებს სხვადასხვა ალგორითმებს, როგორცაა ფურიეს გარდაქმნა (Fourier Transform), ფილტრაცია, მოდულაცია და სხვა ოპერაციები, რათა მიიღონ საჭირო შედეგი. ეს არის ძირითადი ეტაპი, სადაც ციფრული სიგნალების სპეციფიკური დამუშავება ხდება. ზოგადად **DSP**-თი ასევე აღნიშნავენ გარდაქმნა - დამუშავების სრულ პროცესსაც.

DAC-ეს არის ADC-ის შებრუნებული ოპერაცია და ეძახიან ციფრულ ანალოგურ გარდაქმნელს, რომელიც ქმნის კიბისებურ ტალღას ორობითი რიცხვების მიმდევრობიდან, ამ პროცესის შედეგად წარმოიქმნება რიცხვების ახალი თანმიმდევრობა $y_a(t)$ რომლებიც გამოსავალი სიგნალის ნიმუშებს წარმოადგენს. ანუ ეს პროცესი ციფრულ სიგნალს კვლავ ანალოგურ ფორმატში გარდაქმნის. ეს ეტაპი აუცილებელია, რადგან რეალურ სამყაროში მონაცემთა გადაცემა ან გამოსახვა უმეტესწილად ანალოგური სიგნალების სახით ხდება. DAC ესაა ნაბიჯი წარმოიქმნას კვლავ ანალოგური სიგნალი. რომელიც შეიძლება გამოყენებულ იქნას ხმოვან სისტემებში, მონიტორებზე გამოსახულების ჩვენებისას და ა.შ.

PoF-პოსტფილტრია, რომელიც გამოიყენება DAC-ის მიერ გენერირებული ანალოგური სიგნალის გასუფთავებისთვის. DAC პროცესის შედეგად მიღებული სიგნალი შეიძლება შეიცავდეს არასასურველ მაღალსიხშირულ კომპონენტებს, რომლებიც არ უნდა იყოს საბოლოო ანალოგურ სიგნალში. PoF ფილტრი, როგორც წესი, დაბალსიხშირიანი ფილტრი, ამცირებს ამ მაღალსიხშირიან კომპონენტებს, რათა სიგნალი იყოს სუფთა და შეიცავდეს ჩვენთვის გასაგებ ინფორმაციას. (Tabeshadze 2019)

როგორც უკვე ვთქვით არსებობს სიგნალების დამუშავების ორი მიდგომა: ანალოგური და ციფრული, აქედან DSP მიდგომა უფრო რთულია, დამოკიდებულია გაცილებით მეტ კომპონენტზე, ვიდრე „შედარებით მარტივი“ AsP. აქვე გაჩნდება კითხვა, რატომ ხდება პროცესის დამუშავება ციფრულად, იმიტომ რომ ამ პროცესს აქვს რამდენიმე უპირატესობა.

DSP შეიძლება დავეყთ ორ კატეგორიად, სიგნალის ანალიზი და სიგნალის ფილტრაცია.

სიგნალის ანალიზის ამოცანა ეხება სიგნალის თვისებების გაზომვას. ესენია სიხშირეზე დამოკიდებული ოპერაციები:

- სპექტრის (სიხშირე და ფაზა) ანალიზი;
- ხმის ამოცნობა;
- წყაროს იდენტიფიცირება;
- სამიზნის გამოვლენა.

სიგნალის ფილტრაციის ამოცანა ხასიათდება სიგნალის არსებობა არარსებობით. სისტემები რომლებიც ამ ამოცანას ასრულებენ ეძახიან ფილტრებს. ესენია დროზედამოკიდებული ოპერაციები:

- არასასურველი ფუნური ხმაურის მოხსნა;
- წარმოქმნილი ხმაურის მოშორება;
- სიხშირული არეების გამიჯვნა;
- სიგნალის სპექტრის ჩამოყალიბება.

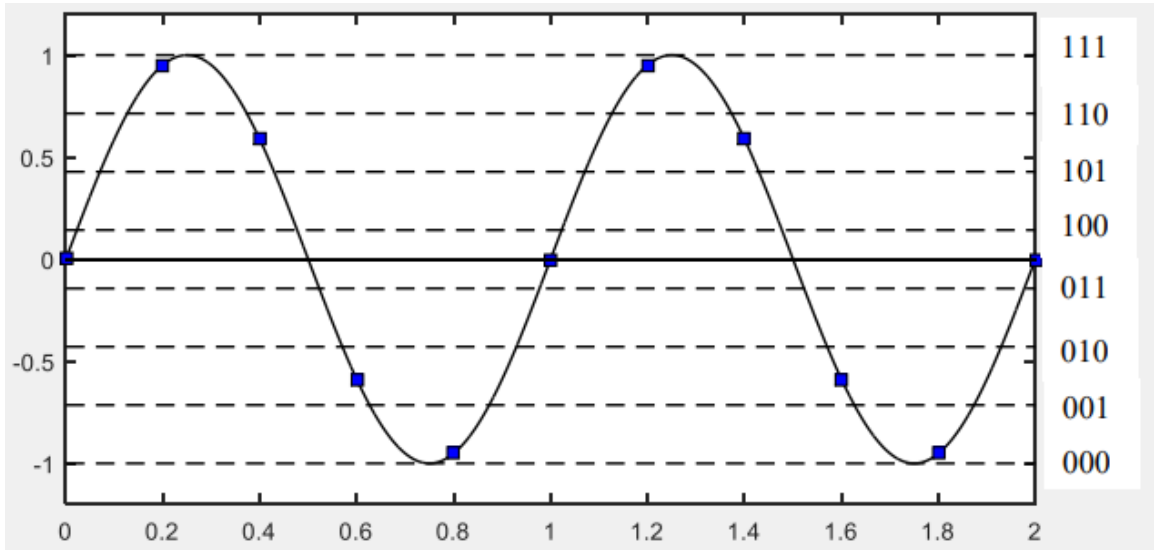
ზოგიერთ პროგრამაში, როგორცაა ხმის სინთეზი, სიგნალის ანალიზი პირველ რიგში ხდება მისი მახასიათებლების შესასწავლით, რომლებიც შემდეგ გამოიყენება ციფრულ ფილტრაციაში სინთეზური ხმის წარმოსაქმნელად. სინთეზური ხმა არის ტექნოლოგია, რომელიც საშუალებას გვაძლევს ხელოვნურად შევქმნათ ადამიანის ხმა კომპიუტერული ან სხვა ელექტრონული მოწყობილობების მეშვეობით.

2.2. ანალოგურ-ციფრული გარდაქმნა

ანალოგური სიგნალები გადაიქცევა ციფრულ სიგნალებად ADC (ანალოგურ-ციფრული) გარდამქმნელის გამოყენებით. ADC გარდამქმნელი ყოველ T_s წამის შემდეგ იღებს ანალოგური სიგნალის ნიმუშს და გარდაქმნის სიგნალის დონეს (ხშირად ძაბვას) ბიტების ნაკრებად – ანუ ანალოგური სიგნალის დონის ციფრულ წარმოდგენად.

სურათზე ნაჩვენებია სინუსური ტალღის სიგნალი, რომელიც ნიმუშების აღება ხდება ყოველ 0.2 წამში. ლურჯი კვადრატები აღნიშნავენ იმ დროს, როდესაც სიგნალიდან ხდება ანათვალის აღება. წყვეტილი ჰორიზონტალური ხაზები მიუთითებენ ADC გარდამქმნელისთვის ხელმისაწვდომ დისკრეტულ დონეებს. სიმარტივის დასაცავად გამოყენებულია 3-ბიტისანი გარდამქმნელი; ანუ ყოველი ანალოგური სიგნალიან ანათვალი იქნება წარმოდგენილი 3-ბიტისანი ციფრული კოდით. არსებობს რვა დისკრეტული დონე , რომლებიც აღნიშნულია წყვეტილი ხაზებით. 3-ბიტისანი კოდი, რომელიც დაკავშირებულია თითოეულ დისკრეტულ

დონესთან, ნაჩვენებია თითოეული დონის მარჯვნივ. ცხრილიში კი მოცემულია 3-ბიტის ციფრული კოდი, რომელიც შეესაბამება თითოეული ანათვალის ადების დროს.



სურათი 18. ანალოგური სიგნალიდან ანათვლების ადება (წყარო-ავტორი)

ანათვლების ადების დრო (წმ)	3-ბიტის ციფრული კოდი
0	100
0,2	111
0,4	110
0,6	001
0,8	000
1,0	011
1,2	111
1,4	110
1,6	001
1,8	000
2,0	011

ცხრილი 5. ბიტის ციფრული კოდი მითითებულ დროებზე (წყარო-ავტორი)

განვიხილოთ ტერმინოლოგია და ფორმულები, რომელიც ნაშრომშია გამოყენებული:

ანათვლების ალების სიხშირე: $F_s = 1/T_s$

სიმბოლო	განმარტება	ერთეული
T_s	ანათვლების ალების ინტერვალი	წამი
F_s	ანათვლების ალების სიხშირე	ჰერცი ჰც
b	ბიტების რიცხვი	ბიტი
l	DAC-ში დონეები	დონე
V_{FS}	ძაბვის დიაპაზონი	ვოლტი
q	დაქვანტვის ბიჯის ზომა	ვოლტი
SNR	სიგნალის ძალასა და ხმაურს შორის თანაფარდობა	დეციბალი დბ

ცხრილი 6. ტერმინოლოგია და ფორმულები, სიგნალების დამუშავებისას (Ingle 2015, 79)

დისკრეტიზაციის დონე: $l = 2^b$ 3 ბიტთან ADC -ს აქვს $2^3 = 8$ დონე, შეესაბამება ბინარულ მიმდევრობებს, რომელიც მხოლოდ 3 ბიტს იყენებს. 12 ბიტთან ADC-ს შეესაბამება 4096 დისკრეტიზაციის დონე.

დაქვანტვის დონე (ბიჯის ზომა): შეიძლება მოვიხსენიოთ როგორც სხვაობა ორ მიმდებარე დისკრეტულ დონეს შორის, რომელიც გამოითვლება ფორმულით:

$$q = \frac{V_{FS}}{(2^B - 1)} = \frac{V_{max} - V_{min}}{(2^B - 1)} \quad (\text{Ingle 2015, 68})$$

ფაქტიურად ძაბვაა, რომელიც გვიჩვენებს რა დიაპაზონში შეუძლია ADC გარდამქმნელს ანალოგური სიგნალების მიღება. ეს არის იმ ანალოგური ძაბვების შუალედი, რომელიც გარდამქმნელს მიეწოდება. მაგალითად, თუ კონვერტორს შეუძლია მიიღოს 0-დან +5 ვოლტამდე ძაბვა, მისი დაქვანტვის დონე იქნება 5 ვოლტი, რადგან ეს არის ამ დიაპაზონის საერთო ზომა (5 ვოლტი). ან კიდევ თუ გარდამქმნელს

შეუძლია მიიღოს ძაბვა -2 ვოლტიდან $+2$ ვოლტამდე, მისი **დაქვანტვის დონე** იქნება 4 ვოლტი, რადგან -2 -დან $+2$ -მდე 4 ვოლტია განსხვავება.

გარჩევადობა: განსაზღვრავს, რამდენად დეტალურად შეუძლია გარდამქმნელს ანალოგური სიგნალის წაკითხვა და ციფრულ ფორმატში გადაყვანა. ADC-ს გარჩევადობა, როგორც წესი, იზომება ბიტებში, რაც მიუთითებს იმაზე, თუ რამდენი განსხვავებული დონის (ან ნაბიჯის) ამოცნობა შეუძლია მას. მაგალითად, 8 -ბიტისანი გარდამქმნელს შეუძლია განსხვავებული დონის ამოცნობა 256 (2^8) დონეზე, მაშინ როდესაც 12 -ბიტისანს შეუძლია 4096 (2^{12}) დონეზე. რაც უფრო მეტი ბიტი აქვს ADC-ს, მით უფრო მაღალია მისი გარჩევადობა და შესაბამისად, უფრო ზუსტად შეუძლია სიგნალის ანალოგურიდან ციფრულ ფორმატში გადაყვანა.

დაქვანტვის (დისკრეტიზაციის) შეცდომა: არის განსხვავება რეალურ ანალოგურ მნიშვნელობასა და დაქვანტვის შემდეგ მიღებულ დისკრეტულ მნიშვნელობას შორის. ეს შეცდომა წარმოიშობა, როდესაც ანალოგური სიგნალი ციფრულ ფორმატში გადაყვანისას მრგვალდება უახლოეს დისკრეტულ დონემდე, რადგან ADC ვერ კითხულობს ან ვერ ინახავს სიგნალის ყველა მცირე დეტალს.

სიგნალის ძალასა და ხმაურს შორის თანაფარდობა (SNR): არის საზომი, რომელიც აჩვენებს, რამდენად ძლიერია სასარგებლო სიგნალი მასში მყოფი არასასურველი ხმაურის დონესთან შედარებით. იგი ჩვეულებრივ იზომება დეციბელებში (დბ). მაღალი SNR ნიშნავს, რომ სიგნალი უფრო სუფთა და მკაფიოა, ანუ ნაკლებია ხმაური.

როგორც წესი, SNR-ის გამოთვლა ხდება დაქვანტვის ხმაურის გათვალისწინებით. **დაქვანტვის ხმაური** არის ის ხმაური, რომელიც ჩნდება დაქვანტვის შეცდომის შედეგად, ანუ როდესაც ანალოგური სიგნალი მრგვალდება უახლოეს დისკრეტულ დონემდე.

SNR-ის გამოთვლა b -ბიტისანი ADC-სთვის, სადაც b არის ბიტების რაოდენობა, განსაზღვრავს, რა ხარისხის და სიზუსტის მქონე ციფრული სიგნალი შეიძლება მივიღოთ ამ გარდამქმნელის გამოყენებით.

განვიხილოთ კონკრეტული ფორმულა, რომელიც აღწერს, როგორ გამოითვლება ეს თეორიული SNR მოცემული ბიტების რაოდენობით:

$$SNR_{\text{დბ}}(\text{საუკეთესო}) = 6.02b + 1.76$$

თეორიული SNR არის სიგნალის და ხმაურის მაქსიმალური თანაფარდობა, რომელიც შეიძლება მივიღოთ იდეალურ პირობებში, როდესაც მხოლოდ დაქვანტვის ხმაურია გათვალისწინებული. ეს არის საუკეთესო სცენარი, რომელიც ითვალისწინებს მხოლოდ სიგნალის დამახინჯებას დაქვანტვის პროცესის დროს. სამწუხაროდ არსებობს სხვა ფაქტორებიც, რომლებიც ამცირებენ SNR-ს. რეალური CF, რომელიც მიიღება პრაქტიკულ პირობებში, იქნება დაახლოებით 2-10 დეციბელით ნაკლები თეორიულ მაქსიმუმზე. ეს ნიშნავს, რომ რეალური სიგნალი ყოველთვის უფრო მეტი ხმაურითაა დაბინძურებული, ვიდრე იდეალურ მოდელშია გათვალისწინებული, რაც ასახება სიგნალის სისუფთავესა და ხარისხზე.

განვიხილოთ 16 ბიტის ანალოგურ ციფრული გარდამქმნელი, დისკრეტიზაციის (დაქვანტვის) სიხშირე არის 48 კჰც, მიეწოდება ძაბვა -5 ვოლტიდან 5 ვოლტამდე და მრგვალდება უახლოეს დისკრეტულ დონემდე. გამოვითვალოთ ანათვლების აღების ინტერვალი T_s , დისკრეტული დონეების რაოდენობა l , დაქვანტვის ბიჯის ზომა q , მაქსიმალური დაქვანტვის შეცდომა და საუკეთესო SNR.

$$\text{გამოვითვალოთ ანათვლების აღების ინტერვალი } T_s = 1/F_s = 1/(48000\text{Hz}) = 0.000020833 \text{ წმ} = 20.833 \mu\text{s}$$

$$\text{დისკრეტული დონეების რაოდენობა } l = 2^b = 2^{16} = 65536 \text{ დონე}$$

$$\text{დაქვანტვის ბიჯის ზომა } q = V_{FS}/(2^b - 1) = (10\text{V})/65535 = 0.0001526\text{V} = 0.1526\text{mV}$$

$$\text{მაქსიმალური დაქვანტვის შეცდომა } = \frac{q}{2} = \frac{0.0001526\text{V}}{2} \approx 0.0000763\text{V} = 76$$

$$\text{სიგნალის ძალასა და ხმაურს შორის თანაფარდობა, საუკეთესო } SNR = 6.02 \times 16 + 1.76 \approx 98.08 \text{ dB}$$

2.3. სიგნალის დამახინჯება, ზედდების ეფექტი

როცა სიგნალის ანთვლების აღების, ანუ რეგისტრაციის პროცესში, ანათვლების აღება ხდება არასაკმარისი სიხშირით (ზედმეტად გაზრდილი დროის ინტერვალებში), ხდება სიგნალის დამახინჯება და წარმოიქმნება ახალი ცრუ სიხშირეები. როგორც ვიცით ნაიქვისტის სიხშირე არის მინიმალური სიხშირე, რმელიც საჭიროა სიგნალის

სწორად რეგისტრაციისთვის, წინააღმდეგ შემთხვევაში მისი აღდგენა იქნება ძალიან რთული ან შეუძლებელი. ნაიქვისტის სიხშირე არის სიგნალის მაქსიმალური სიხშირულ კომპონენტზე ორჯერ მეტი. თუ მაინც ისე მოხდა, რომ ანათვლების ალების სიხშირე ნაიქვისტის სიხშირეზე დაბალია, სიგნალი მახინჯდება და წარმოიქმნება არასწორი მონაცემები. წარმოიდგინეთ, რომ გვაქვს სიგნალი, რომლის სიხშირეა 1000 ჰც (ჰერცი) და ჩვენ ანათვლებს ვიღებთ 1500 ჰც სიხშირით. რადგან 1500 ჰც ნაიქვისტის სიხშირეზე ნაკლებია (მინიმუმი უნდა იყოს 2000 ჰც), მივიღებთ დამახინჯებულ სიგნალს, რომელიც შეიძლება შეიცავდეს სიხშირეებს, რომლებიც საერთოდ არ არსებობდა თავდაპირველ სიგნალში. ვიზუალურ გამოსახულებებში, ზედდების ეფექტი შეიძლება გამოჩნდეს როგორც "კიბის მსგავსი" ეფექტები გრაფიკულ ობიექტებზე, როცა მათ სიგლუვესა და მრგვალობის ნაცვლად, კუთხოვანი კუთხეები უჩნდებათ.

MATLAB-ში განვიხილოთ 3 სხვადასხვა სცენარი, სცენარი ეხება სამი სხვადასხვა სიგნალის, 2 კჰც, 8 კჰც და 12 კჰც სიხშირის სიგნალის სინუსოიდების ანათვლების აღებას 10 კჰც (10,000 ანათვალი წამში) სიხშირით.

```
* განვსაზღვროთ ნიმუშების ალების სიხშირე 10 კჰც-ით  
sampling_rate = 10000;  
time_vector = 0:1/sampling_rate:0.001; * დროის ინტერვალი უნდა იყოს 1/Fs
```

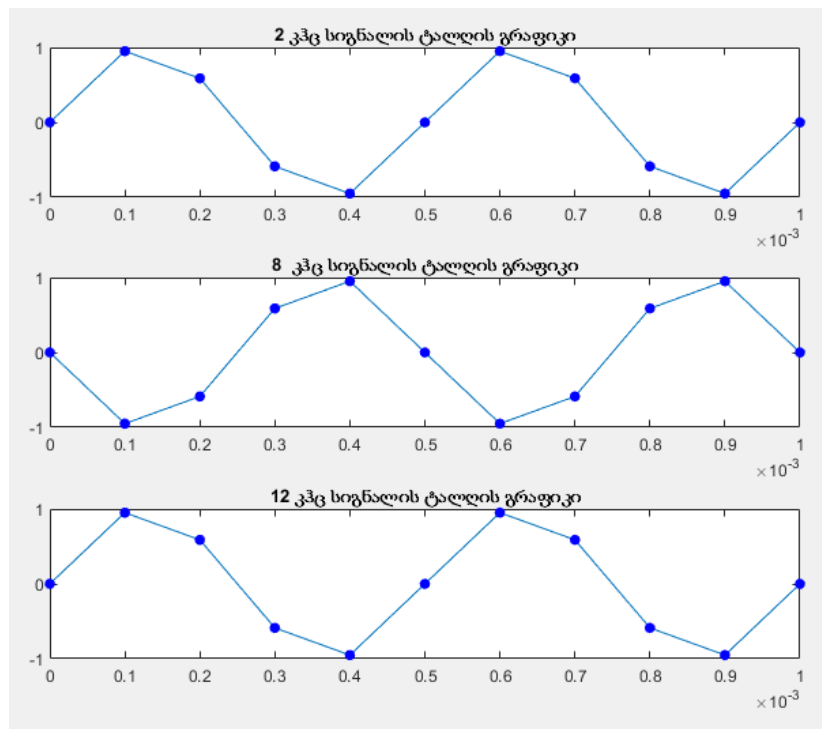
```
* სამი სხვადასხვა სიხშირის სინუსოიდის გენერაცია  
signal1 = sin(2*pi*2000*time_vector); * 2 კჰც სიხშირის ტალღა  
signal2 = sin(2*pi*8000*time_vector); * 8 კჰც სიხშირის ტალღა  
signal3 = sin(2*pi*12000*time_vector); * 12 კჰც სიხშირის ტალღა
```

```
* გრაფიკების გენერაცია  
plot_wave(time_vector, signal1, 2000, sampling_rate, 1);  
plot_wave(time_vector, signal2, 8000, sampling_rate, 2);
```

```
plot_wave(time_vector, signal3, 12000, sampling_rate, 3);
```

*** ფუნქციის განსაზღვრა**

```
function plot_wave(time_vector, signal, frequency, sampling_rate, subplot_position)
    subplot(3,1,subplot_position);
    plot(time_vector, signal, 'k-', time_vector, signal, 'bo', 'Markersize', 8, 'MarkerFaceColor',
    'b');
    title([num2str(frequency), ' კჰც ტალღის გრაფიკი, დისკრეტიზაციის სიხშირე ',
    num2str(sampling_rate/1000), ' კჰც']);
end
```



სურათი 19. ზედღების ეფექტი (წყარო-ავტორი)

2 კჰც სიგნალი: მიუხედავად იმისა, რომ ანათვლების აღების სიხშირე 10 კჰც-ია, 2 კჰც ტალღა საკმაოდ სწორადაა წარმოდგენილი გრაფიკზე. თუმცა, „ცოტათი უხეში“ ნიშნავს, რომ გამოყენებული სიხშირის გამო, ტალღა სრულად გლუვი არ არის, 10 კჰც სიხშირე საკმარისია 2 კჰც სიგნალის აღსაქმელად, მაგრამ მას გარკვეული მკაფიო მახასიათებლები აკლია.

8 კჰც ტალღა: ანათვლების აღების პროცესში „გადაბრუნდება“. ეს ნიშნავს, რომ სიგნალი, რომელიც ანათვლების აღების შემდეგ უნდა ყოფილიყო 8 კჰც, საბოლოოდ წარმოდგენილია როგორც 2 კჰც ტალღის „ინვერსიული“ (გადაბრუნებული) ვერსია. ეს

ხდება ზუსტად ზედღების ეფექტის შედეგად, რადგან 10 კვც სიხშირე არ არის საკმარისი 8 კვც ტალღა სწორად დავარეგისტრიროთ.

12 კვც ტალღა: 12 კვც ტალღა ასევე ზედღების ეფექტის შედეგად გარდაიქმნება და საბოლოოდ ემთხვევა 2 კვც ტალღას. ეს ნიშნავს, რომ 12 კვც ტალღა რეგისტრირდება იმავე შედეგით, როგორც 2 კვც ტალღა, რაც დამახინჯების ტიპური შედეგია. სინამდვილეში, 12 კვც სიგნალი არასწორად არის აღქმული და წარმოდგენილია როგორც უფრო დაბალი სიხშირის ტალღა.

დავაკვირდეთ ამ სიგნალების შესაბამის სინუსოიდებს, ვნახოთ როგორაა ანთვლები აღებული:

*** ნიმუშების აღების სიხშირის განსაზღვრა**

```
sampling_rate = 10000;
```

```
time_vector = 0:1/sampling_rate:0.001; * დროის შუალედი
```

*** სინუსოიდური სიგნალების გენერაცია სხვადასხვა სიხშირეებით**

```
signal1 = sin(2*pi*2000*time_vector); * 2 კვც სიგნალი
```

```
signal2 = sin(2*pi*8000*time_vector); * 8 კვც სიგნალი
```

```
signal3 = sin(2*pi*12000*time_vector); * 12 კვც სიგნალი
```

*** გრაფიკული გამოსახულება**

```
subplot(3,1,1);
```

```
plot(time_vector, signal1, 'k-', time_vector, signal1, 'bo', 'MarkerSize', 8, 'MarkerFaceColor', 'b');
```

```
title('2 კვც სიგნალი, 10 კვც დისკრეტიზაციის სიხშირით');
```

```
subplot(3,1,2);
```

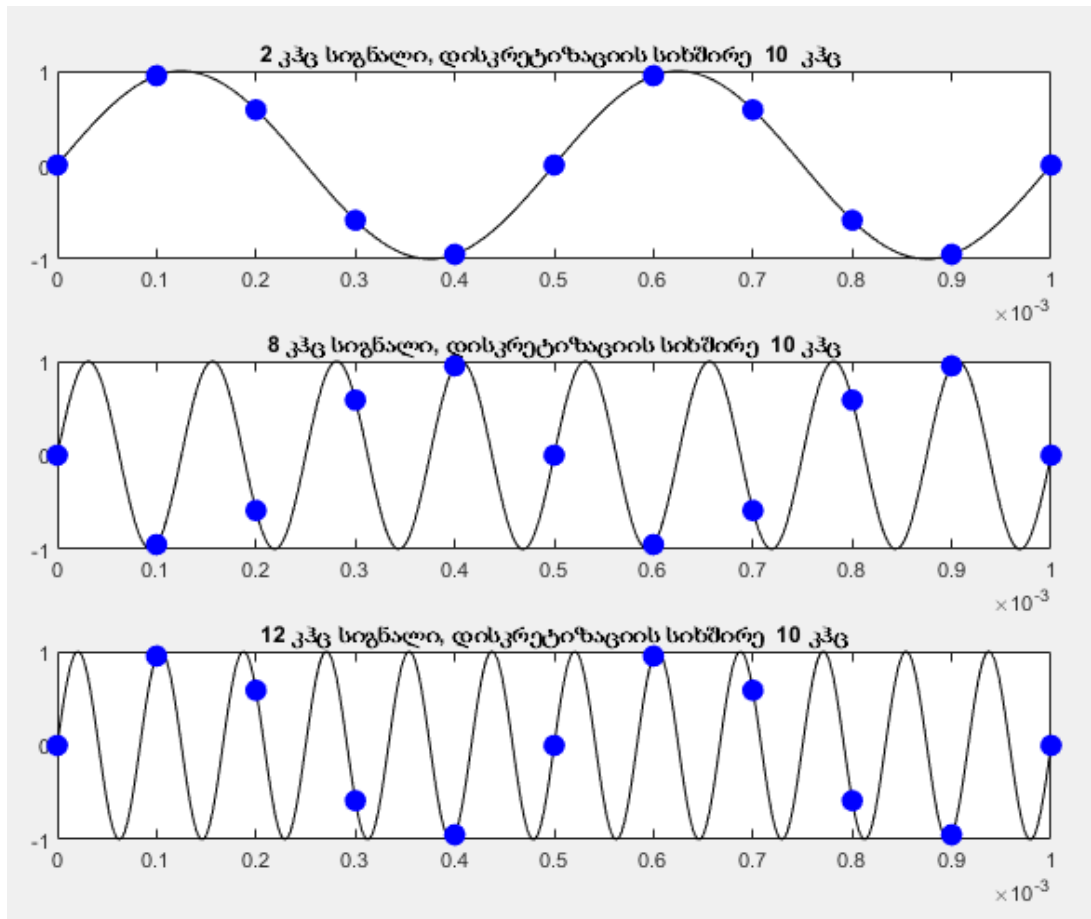
```
plot(time_vector, signal2, 'k-', time_vector, signal2, 'bo', 'MarkerSize', 8, 'MarkerFaceColor', 'b');
```

```
title('8 კვც სიგნალი, 10 კვც დისკრეტიზაციის სიხშირით');
```

```
subplot(3,1,3);
```

```
plot(time_vector, signal3, 'k-', time_vector, signal3, 'bo', 'MarkerSize', 8, 'MarkerFaceColor', 'b');
```

```
title('12 კვც სიგნალი, 10 კვც დისკრეტიზაციის სიხშირით');
```



სურათი 20. ზედღებვის ეფექტი სინუსოიდურ ტალღებზე (წყარო-ავტორი).

კარგად ჩანს, რომ 8 კვც და 12 კვც ტალღების ანათვლები დროში ერთმანეთისგან ძალიან დაშორებულია და კარგად ვერ წარმოადგენს თავდაპირველ ანალოგურ სიგნალებს. სინამდვილეში, ამ სიგნალების ანათვლები ქმნიან დაბალსიხშირულ 2 კვც სიგნალს.

დავუშვათ, 2 კვც სიგნალი (ამპლიტუდა = 2) და 12 კვც სიგნალი (ამპლიტუდა = 1) ემატება ერთმანეთს და შემდეგ ანათვლების აღება ხდება 10 კვც.

```

% ნიმუშების ადების სიხშირის განსაზღვრა
sampling_rate = 10000;
time_vector = 0:1/sampling_rate:0.001; % დროის შუალედი

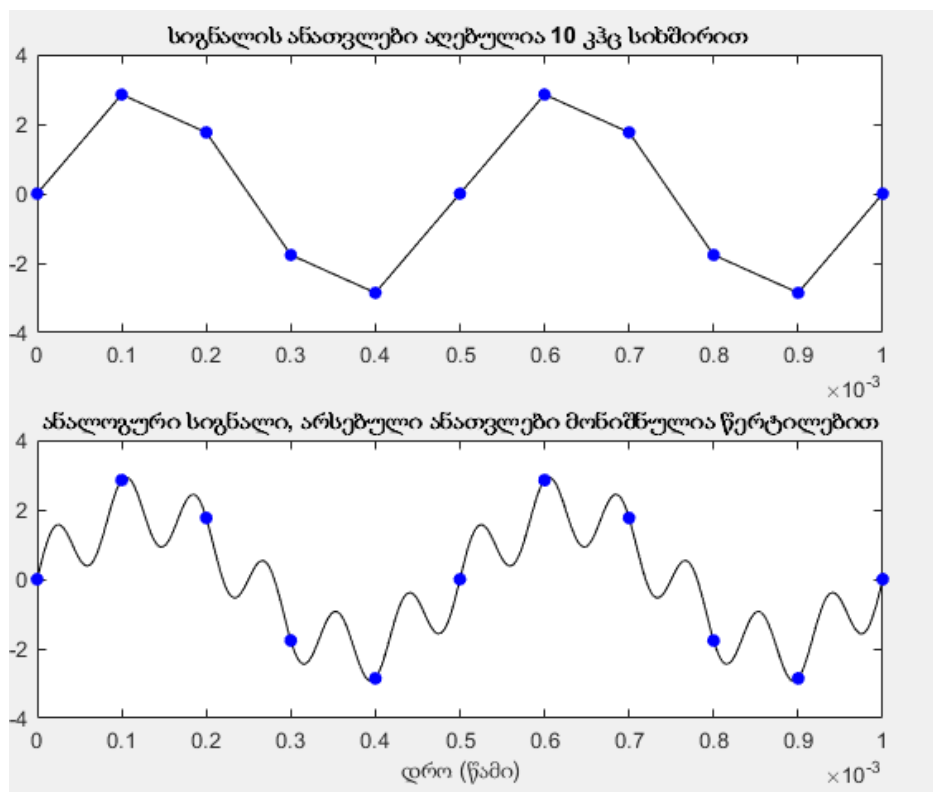
% სინუსოიდური სიგნალის გენერაცია ორი განსხვავებული სიხშირით
signal_combined = 2*sin(2*pi*2000*time_vector) + sin(2*pi*12000*time_vector);

subplot(2,1,1);
plot(time_vector, signal_combined, 'k-', 'Marker', 'o', 'Markersize', 5, 'MarkerFaceColor', 'b',
'MarkerEdgeColor', 'b'); % დავამატოთ ლურჯი წერტილები
title('სიგნალის ანათვლები აღებულია 10 კჰც სიხშირით')

% ანალოგური სიგნალის გენერაცია მეტი წერტილის გამოყენებით
time_vector_analog = 0:0.000001:0.001;
signal_analog = 2*sin(2*pi*2000*time_vector_analog) + sin(2*pi*12000*time_vector_analog);

subplot(2,1,2);
plot(time_vector_analog, signal_analog, 'k-', time_vector, signal_combined, 'b.',
'Markersize', 20);
title('ანალოგური სიგნალი, აღებული ანათვლები მონიშნულია წერტილებით')
xlabel('დრო (წამი)')

```



სურათი 21. ზედდების ეფექტი სიგნალების შეკრების დროს (წყარო-ავტორი)

12 კვც სიგნალი, რომლის ანათვლები 10 კვც-ით სიხშირითაა აღებული, გადაიქცევა 2 კვც სიგნალად, ამპლიტუდით 1. ეს სიგნალი ემატება 2 კვც სიგნალს, რომელსაც აქვს 2 ამპლიტუდა, რის შედეგადაც მიიღება 2 კვც სიგნალი, ამპლიტუდით 3.

ზემოთ მოყვანილი მაგალითები აჩვენებს, რომ თუ ანალოგურ სიგნალის რეგისტრირებისას ნიმუშების აღება სწრაფად არ ხდება, მაღალი სიხშირის სინუსოიდები გარდაიქცევიან დაბალი სიხშირის სინუსოიდებად. მაშასადამე, თუ სიგნალს გააჩნია ფინიტიური (შეზღუდული) სპექტრი, მაშინ დისკრეტული სიგნალი შეგვიძლია ცალსახად აღვადგინოთ დანაკარგების გარეშე თუკი გამოვიყენებთ დისკრეტეტიზაციის სიხშირეს, რომელიც გაორმაგებული ნაიქვისტის სიხშირეზე მეტია. აქედან გამომდინარე ყველაზე დიდი სიხშირე, რომელზეც შესაძლებელია დისკრეტეტიზაციის $\frac{1}{F_s}$ სიჩქარით ანათვლების აღება არის ნაიქვისტის სიხშირე $\frac{1}{2F_s}$ (ჰც). ჩვენს განხილულ მაგალითებში მაღალი სიხშირის სიგნალი გადაიქცევა x ღერძზე, შესაბამისად კეცვების კენტმა რაოდენობამ გამოიწვია „ინვერსია“, ხოლო კეცვების ლუწმა რაოდენობამ არა.

2.4. ფილტრი ზედდების ეფექტის საწინააღმდეგოდ

დისკრეტეტიზაციის სიხშირე მინიმუმ ორჯერ მეტი უნდა იყოს სიგნალის მაქსიმალურ სიხშირეზე, წინააღმდეგ შემთხვევაში სიგნალის სპექტრი მახინჯდება, საჭირო ხდება ზედდების ეფექტის საწინააღმდეგო ფილტრის კონსტრუირება. ასეთი ფილტრი არის დაბალსიხშირული ფილტრი, რომელიც ატარებს მხოლოდ დაბალ სიხშირეებს და ბლოკავს ან ამცირებს მაღალ სიხშირეებს.

სურათზე ნაჩვენებია მაგნიტუდის და ფაზის მახასიათებლები იდეალური ფილტრისთვის ასევე პრაქტიკული დაბალსიხშირიანი ფილტრისთვის.

იდეალური ფილტრის მახასიათებლები:

- იდეალური დაბალგამტარი ფილტრი ყველა სიგნალის სიხშირეს ატარებს გარკვეულ სიხშირემდე, რომელსაც ვუწოდებთ $f_{\text{გატარების}}$. ეს ნიშნავს, რომ ამ დიაპაზონში არსებული სიგნალის ყველა სიხშირე უცვლელად გადის ფილტრში, რაც კარგად ჩანს პირველ სურათზე (ანუ სიგნალი არ სუსტდება და არ ძლიერდება).

- **ფაზური რეაგირება:** ფაზური რეაგირება განსაზღვრავს, თუ რამდენად იცვლება სიგნალის ფაზა ფილტრის გავლით. იდეალური ფილტრი უზრუნველყოფს, რომ ყველა სიგნალს, რომელიც გადის ფილტრში ჰქონდეს ერთნაირი დროის დაყოვნება, რაც ხაზოვან ფაზურ რეაგირებას ნიშნავს. ამი გამო სიგნალი არ იცვლება დროში.

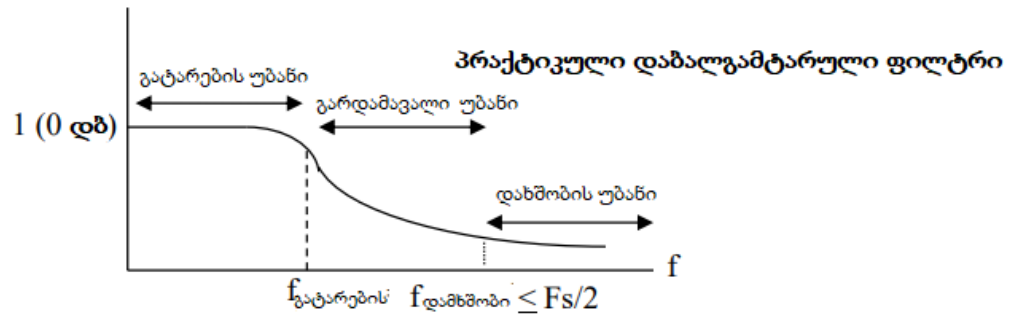
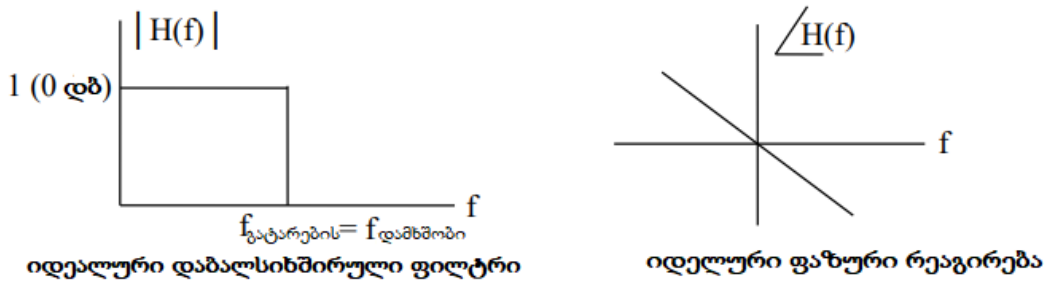
- **სიხშირეების დაბლოკვა:** ფილტრი სრულად ბლოკავს სიგნალებს, რომელთა სიხშირე აღემატება $f_{გატარების}$ -ს. ეს ნიშნავს, რომ ეს სიხშირეები ფილტრის გამოსავალზე არ გამოჩნდება.

პრაქტიკული დაბალგამტარული ფილტრი:

რეალურად იდეალური ფილტრი არ არსებობს, ამიტომ პრაქტიკულ ფილტრებს ხშირად აქვთ გარკვეული „ხარვეზი“, პრაქტიკულ ფილტრებს შეიძლება ჰქონდეთ გარკვეული არაიდენტური ფაზური რეაგირება, რაც იწვევს სიგნალის დროის დაყოვნების განსხვავებას სიხშირეების მიხედვით. იდეალური ფილტრი იქნებოდა ისეთი, რომელიც ყველა სიგნალის სიხშირეს სრულად გაატარებდა გარკვეულ სიხშირემდე და სრულად დაბლოკავდა ყველა სიხშირეს ამ $f_{გამტარი}$ -ის ზემოთ. თუმცა, ასეთი ზუსტი გაყოფა შეუძლებელია ტექნოლოგიური შეზღუდვების გამო.

პრაქტიკული დაბალგამტარი ფილტრები შეიცავენ **გარდამავალ უბანს გამტარ უბანსა** (*passband*) და **დამხშობ უბანს** (*stopband*) შორის. გარდამავალ უბანში ფილტრის რეაგირება იცვლება – სიგნალის სიხშირეები ამ დიაპაზონში გავლის შემდეგ გარკვეულწილად მცირდება. აქ ფილტრი ვერ უზრუნველყოფს სიხშირეების ვერც სრულ გაძლიერებას და ვერც სრულად ბლოკირებას.

დამხშობი უბანი არის ის დიაპაზონი, სადაც ფილტრი მნიშვნელოვნად ამცირებს სიგნალის გაძლიერებას. ეს ნიშნავს, რომ ამ დიაპაზონში მყოფი სიგნალის სიხშირეები ძალიან სუსტდება და პრაქტიკულად არ გამოჩნდება ფილტრის გამოსავალზე.



სურათი 22. დაბალსიხშირული ფილტრის სიხშირული მახასიათებლები (Ingle 2015, 70-75)

პრაქტიკაში ხშირად შევხვდებით სამ ფილტრს **ბატერვორტის**, **ჩებიშევის**, **ბესელის ფილტრები**, რომელთაც აქვთ თავისი უნიკალური მახასიათებლები. ამ ფილტრების მახასიათებლები განაპირობებს მათ გამოყენებას სხვადასხვა პროგრამულ უზრუნველყოფაში: ბატერვორტის ფილტრები კარგია ზოგადი გამოყენებისთვის, ჩებიშევის ფილტრები - მაშინ, როდესაც საჭიროა მკვეთრი ბლოკვა, ხოლო ბესელის ფილტრები - იქ, სადაც ფაზური რეაგირების სიზუსტე მნიშვნელოვანია.

ბატერვორტის ფილტრი: აქვს მაქსიმალურად ბრტყელი გამტარი დიაპაზონი, ეს ნიშნავს, რომ გამტარ დიაპაზონში სიგნალის ყველა სიხშირე ერთნაირი გაძლიერებით გადის, რაც უზრუნველყოფს სიგნალის მინიმალურ დამახინჯებას. ბატერვორტის ფილტრს აქვს შედარებით გლუვი გადასვლა გამტარ დიაპაზონიდან დამხშობ დიაპაზონში, რაც ნიშნავს, რომ ფილტრი თანდათანობით ამცირებს სიგნალის სიხშირეებს და არა მკვეთრად.

ჩებიშევის ფილტრი: ჩებიშევის ფილტრს აქვს ვიწრო გარდამავალი უბანი, რაც ნიშნავს, რომ ის სწრაფად და მკვეთრად გადადის გამტარ დიაპაზონიდან დამხშობ დიაპაზონში. ხშირად ეს უზრუნველყოფს ფილტრის უფრო ეფექტურ მუშაობას არასასურველი სიხშირეების მოსაცილებლად.

ჩებიშევის ფილტრის ნაკლია ის, რომ ფილტრის რეაგირება არ არის სრულად გლუვი და შეიძლება გამოიწვიოს სიგნალის დამახინჯება.

ბესელის ფილტრი: ბესელის ფილტრის მთავარი უპირატესობა მისი ხაზოვანი ფაზური რეაგირებაა, რაც უზრუნველყოფს სიგნალის მინიმალურ დროით დამახინჯებას. ეს თვისება მნიშვნელოვანია იმ მოწყობილობებსა და პროგრამულ უზრუნველყოფაში, სადაც სიგნალის გადასაცემად კრიტიკულად მცირე დროა გამოყოფილი.

ასევე აღსანიშნავია ისიც, რომ ბესელის ფილტრს აქვს ყველაზე ნაკლებად ეფექტური ამპლიტუდური რეაგირება, რაც ნიშნავს, რომ იგი ნაკლებად იდეალურია სიგნალის ბლოკირებისთვის.

ფილტრის მოთხოვნები: 1) ფილტრის გამტარი დიაპაზონის ან ამოჭრის სიხშირე უნდა იყოს საკმარისად დიდი, რომ მოიცავდეს ყველა იმ სიხშირულ კომპონენტს, რომელიც სიგნალის მნიშვნელოვანი ნაწილია, რათა ნიმუშის აღების პროცესში არ დაიკარგოს მნიშვნელოვანი ინფორმაცია და არ მოხდეს სიგნალის დამახინჯება ან დაზიანება. 2) ფილტრმა უნდა დაახმოს სიხშირეები, რომლებიც ნაიქვისტის სიხშირეს აღემატება $f > \frac{F_s}{2}$. ეს დახმობა განისაზღვრება ფორმულით $20 \log(\sqrt{1.5} \cdot 2^b)$, სადაც b არის ანალოგურ ციფრული გარდამქმნელის დონე.

ქვემოთ მოცემული ფორმულა აღწერს **ბატერვორტის ფილტრს:**

$$|H(f)|_{\text{ბატერვორტი}} = \frac{1}{\sqrt{1+(f/f_c)^{2N}}} \quad (\text{Ingle 2015, 90})$$

სადაც $|H(f)|$ = ფილტრის ამპლიტუდა f სიხშირეზე

f_c = ფილტრის გამტარი დიაპაზონის სიხშირე

N = ფილტრის რიგი.

ბატერვორტის ფილტრის ფორმულით 4 კვც სიხშირის დიაპაზონს სიგნალისთვის, სადაც ანათვლების აღება ხდება 8 ბიტისანი ADC-ით, 16 კვც სიხშირით (სიჩქარით), გავიგოთ რომელი რიგის ფილტრია საჭირო სიგნალის აღსადგენად. აქვე აღვნიშნოთ რომ რიგი მთლიანად განსაზღვრავს ფილტრის მუშაობის ეფექტურობას.

ფილტრის გამტარი დიაპაზონის ანუ ამოჭრის სიხშირე უნდა იყოს მინიმუმ 4 კვც (სიგნალის სიხშირის სიგანის ტოლი), რათა ვუზრუნველყოთ, რომ სიგნალმა შეძლოს ფილტრში გასვლა. ავირჩიოთ $f_c = 4$ კვც.

დამხშობი დიაპაზონი 8 ბიტანი ADC-ისთვის 16 კჰც ნიმუშების აღების სიხშირით განისაზღვრება შემდეგნაირად:

$$20 \log(\sqrt{1.5} \cdot 2^b) = 20 \log(\sqrt{1.5} \cdot 2^8) = 50 \text{ დბ} \quad \text{სადაც } f \geq 8 \text{ კჰც} \quad (\text{Ingle 2015, 95})$$

ქვემოთ მოცემული ბატერვორტის ფილტრის ფორმულა გვაძლევს საშუალებას გავიგოთ რომელი რიგის ფილტრია საჭირო 4 კჰც სიხშირის დიაპაზონს სიგნალისთვის. სიგნალის საწყისი ენერგია შემცირებულია 50 დბ-ით, ეს სიდიდე უნდა გარდავქმნათ პირდაპირ სიდიდეში $10^{(-50/20)} = 0.00316228$, ეს ნიშნავს რომ საწყისი სიგნალის ამპლიტუდა შემცირებულია 0.00316228 მნიშვნელობამდე. ვნახოთ რომელი რიგის ფილტრია საჭირო მის აღსადგენად:

$$|H(f)|_{\text{ბატერვორტი}} = \frac{1}{\sqrt{1 + \left(\frac{f}{f_c}\right)^{2N}}} \leq -50 \text{ დბ} \quad \text{სადაც } f \geq 8 \text{ კჰც}$$

$$|H(f)|_{\text{ბატერვორტი}} = \frac{1}{\sqrt{1 + (8\text{კჰც}/4\text{კჰც})^{2N}}} \leq 0.00316228$$

$$|H(f)|_{\text{ბატერვორტი}} = \frac{1}{\sqrt{1 + (2)^{2N}}} \leq 0,00316228$$

$$\sqrt{1 + (2)^{2N}} \geq 1:0,00316228 = 316,22777$$

$$1 + (2)^{2N} \geq 100000$$

$$(2)^{2N} \geq 99999$$

$$(2N) \cdot \log(2) \geq \log(99999)$$

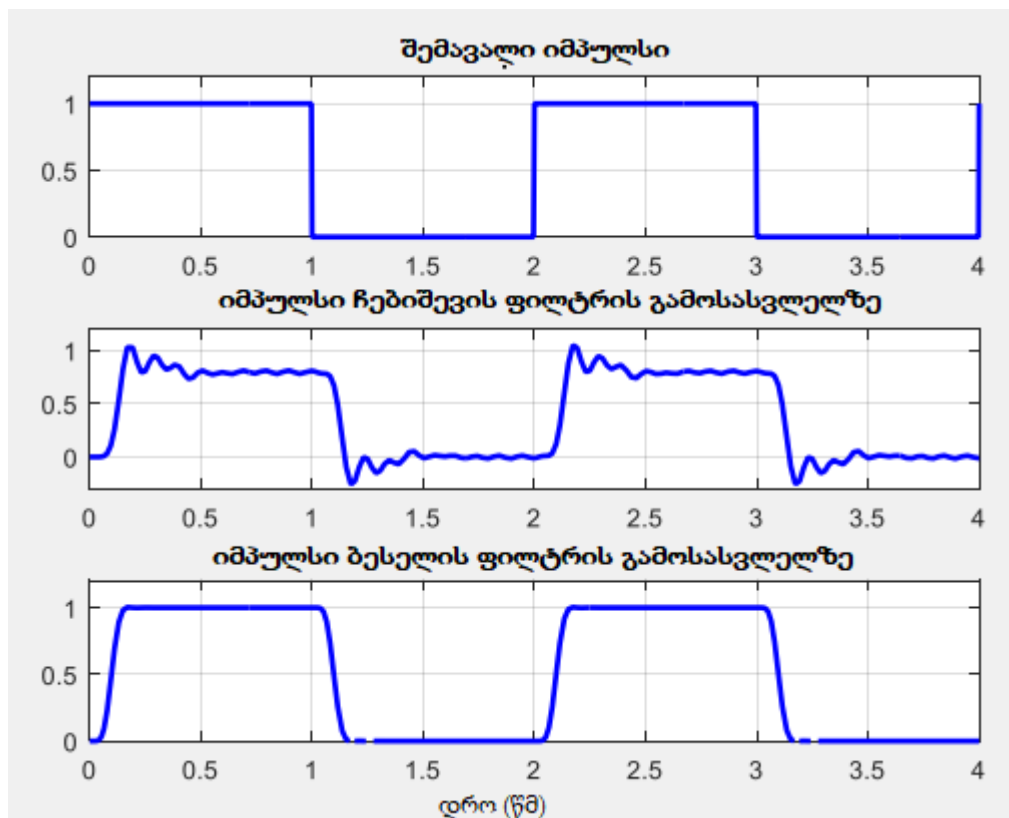
$$N \geq \frac{1}{2} \cdot \log(99999) / \log(2) = 8,3$$

მივიღეთ, რომ მე-9 რიგის დაბალსიხშირული ბატერვორტის ფილტრია საჭირო 4 კილოჰერცი მოკვეთის სიხშირით.

ბატერვორტის ფილტრის რიგის განსაზღვრა შეიძლება რამდენიმე სხვადასხვა მეთოდით და ერთ-ერთი მათგანია სპეციალური დიზაინის ინსტრუმენტის გამოყენება, რომელიც საშუალებას იძლევა ექსპერიმენტულად მოვძებნოთ ყველაზე მცირე რიგი, რომელიც კონკრეტული ამოცანის მოთხოვნებს შეესაბამება. თუმცა ასეთი ინსტრუმენტების მოძიებაც და მათი გამოყენების შესწავლაც დიდ დროს და ძალისხმევას მოითხოვს, მაშინ როცა მათემატიკური მოდელირება გვაძლევს ამ ამოცანის გადაჭრის ძალიან სწრაფ საშუალებას.

ფილტრის ტიპის არჩევა

აუდიო სიგნალის დასამუშავებლად საუკეთესო არჩევანია ბატერვორტის ან ჩებიშევის ფილტრები. ვიდეო სიგნალებისთვის, სიკაშკაშე და ფერი იცვლება სივრცესა და დროში. ფილტრი, რომელიც ინარჩუნებს სიგნალის ფორმას (ბესელის ფილტრი), შესაბამისად, საუკეთესო არჩევანი იქნება აუდიო სიგნალის დასამუშავებლად. სურათზე ნაჩვენებია მე-8 რიგის ჩებიშევის და მე-8 რიგის ბესელის ფილტრის იმპულსური მახასიათებლები. დავაკვირდეთ, რომ ჩებიშევის ფილტრის გამოსასვლელზე სიგნალს აქვს ზარების მსგავსი ეფექტი ორივე მხარეს. ვიდეო სიგნალში ეს გავლენას მოახდენს ობიექტების სიკაშკაშეზე – მარცხენა მხარე უფრო ნათელი იქნება, ვიდრე ცენტრი, ხოლო მარჯვენა მხარე უფრო მუქი ვიდრე ცენტრი. შედარებისთვის, ბესელის ფილტრი, თავისი ხაზოვანი ფაზური მახასიათებლებით, ქმნის გლუვ, სიმეტრიულ სიგნალს ზედმეტი დამახინჯებების გარეშე.



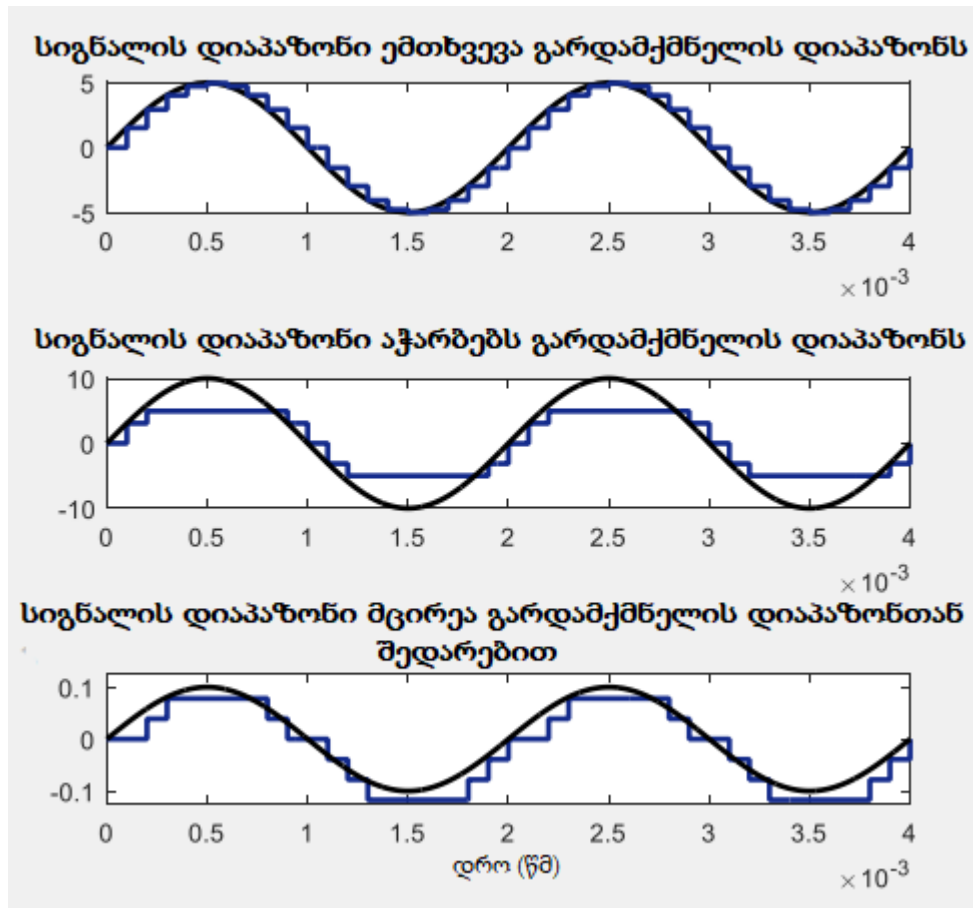
სურათი 23. დაბალსიხშირული ფილტრებს იმპულსური მახასიათებლები (წყარო-ავტორი)

სიგნალის შესაბამისობა ADC გარდამქმნელთან

ყოველ ADC კონვერტორს აქვს დაშვებული დიაპაზონი სიგნალის გასატარებლად. ზოგიერთ ADC-ს შეუძლია სიგნალების მიღებას შეუძლია მხოლოდ 0- +5ვ დიაპაზონში. ხოლო სხვა ADC-ები იღებენ როგორც დადებით, ისე უარყოფით ვოლტაჟებს, მაგალითად, -2ვ -დან +2ვ -მდე. იმისათვის რომ სიგნალისა და ხმაურის თანაფადობა (SNR) იყოს მაქსიმალური, სიგნალი უნდა იყოს შესაბამისი ADC კონვერტორის ვოლტაჟის დიაპაზონთან.

თუ სიგნალი ვოლტაჟი აჭარბებს ADC-ის დიაპაზონს, ADC შეიძლება დაზიანდეს. თუ სიგნალის დიაპაზონი ბევრად მცირეა ADC-ის ვოლტაჟის დიაპაზონზე, მაშინ ADC-ის რამდენიმე ცალკეული დონე არ იქნება გამოყენებული. ფაქტიურად შემცირდება ბიტების რაოდენობას, რაც თავის მხრივ შეამცირებს SNR-ს.

სურატზე ნაჩვენებია სიგნალის დიაპაზონისა და ADC-ის ვოლტაჟის დიაპაზონის შესაბამისობა. ვივარაუდოთ, რომ ADC-ის დიაპაზონი არის -5ვ -დან +5ვ-მდე. პირველ გრაფიკზე სიგნალის დიაპაზონი შეესაბამება ADC-ის დიაპაზონს და აღებული მნიშვნელობები ასახავს ანალოგური სიგნალის ცვლილებებს. მეორე გრაფიკზე სიგნალის დიაპაზონი აღემატება ADC-ის დიაპაზონს: ADC-ის დიაპაზონი არის -5V-დან +5V-მდე, მაგრამ სიგნალი ცვალებს -10V-დან +10V-მდე. ეს იწვევს სიგნალის დამახინჯებას, რაც კრიტიკულ სიგნალურ ინფორმაციას კარგავს. მესამე გრაფიკზე სიგნალი მხოლოდ -0.1V-დან +0.1V-მდე დიაპაზონშია. ამიტომ, ყველა ADC-ის ცალკეული დონე +0.1V-დან +5V-მდე და -0.1V-დან -5V-მდე არც კი გამოიყენება (ანუ 98 % დონეების). ეს როგორც ვთქვით ამცირებს ADC-ის ბიტების რაოდენობას და ამცირებს SNR-ს.



სურათი 24. შემაჯავლი სიგნალის შესაბამისობა ADC -სთან (წყარო-ავტორი)

ასევე მიღებული პრაქტიკაა სიგნალის ზედმეტი რაოდენობის ანათვლების აღება ზედდების ეფექტის საწინააღმდეგო ფილტრის გამოყენების სიმარტივისთვის.

2.5. ანალოგური სიგნალიდან ნიმუშების აღების ინტერპოლაციის მეთოდები

განვიხილოთ, როგორ შეიძლება ანალოგური სიგნალების ანათვლებით სიგნალის აღდგენა. სურათზე მოცემულია ანალოგური სიგნალების ანათვლების სპექტრი. თუ ანათვლების აღების სიხშირე საკმარისად მაღალია (ნიკვისტ სიხშირის ზემოთ), სპექტრული გამოსახულება იქნება გამოყოფილი საწყისი სიგნალიდან და ასეთ შემთხვევაში ორიგინალი სიგნალი შეიძლება აღდგეს დაბალსიხშირული ფილტრით.

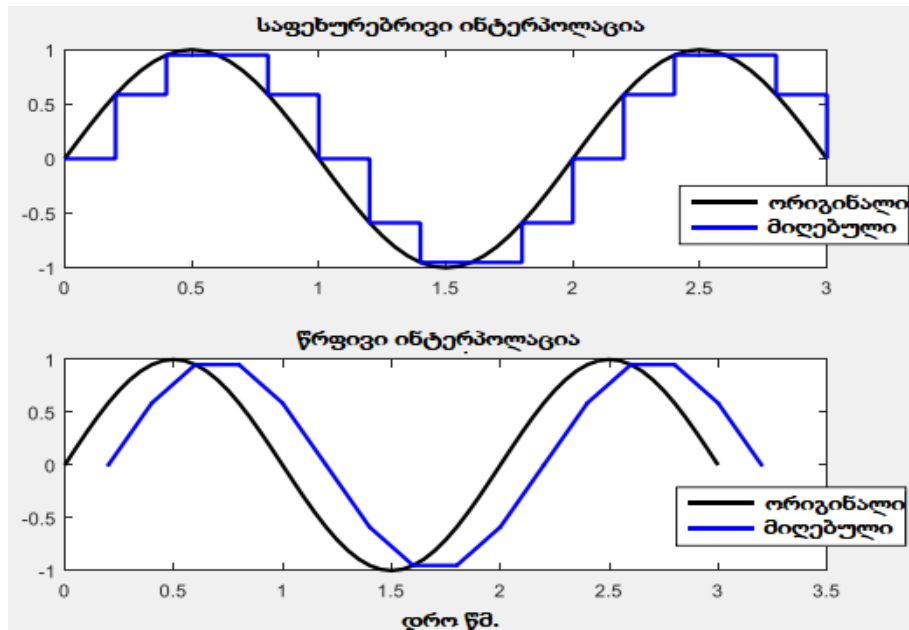
სურათზე ნაჩვენებია სპექტრი იდეალური დაბალსიხშირული ფილტრისთვის. როგორც ზემოთ ვახსენეთ, შეუძლებელია იდეალური დაბალსიხშირიანი ფილტრის დიზაინი, ამიტომ, აღმდგენი ფილტრი საუკეთესო შემთხვევაში იქნება იდეალურ დაბალსიხშირულ ფილტრთან მიახლოებული.

საფეხურებრივი ინტერპლაცია

ფილტრი სიგნალის მნიშვნელობებს, მის ანათვებს იღებს და მნიშვნელობას ინარჩუნებს მომდევნო ნიმუშის აღებამდე. ასე გრძელდება სიგნალის მთელს სიგრძეზე. აღებული ნიმუშები ქმნის საფეხუროვან ტალღას შესაბამისი ანალოგური სიგნალისგან. ეს არის ინტერპოლაციის ერთ-ერთი სქემა, რომელიც შეიძლება განხორციელდეს რეალურ დროში და არა სიგნალის დაყოვნებით. ამ დროს ნიმუშების აღების სიხშირე ძალიან მაღალი არ უნდა იყოს, წინააღმდეგ შემთხვევაში მიღებული მოდელი სრულად ვერ შეესაბამება ანალოგურ სიგნალს.

წრფივი ინტერპოლაცია

წრფივი ინტერპოლაციის ფილტრი არეზულ ნიმუშებს აკავშირებს წრფივად, ხაზებით, რაც ქმნის ნაჭრიან ანალოდურ სიგნალს. ხაზოვანი ინტერპოლაციის ფილტრი იყენებს სიგნალის დაყოვნების ტექნიკას, ერთი ნიმუშის ინტერვალით. იმისათვის რომ ანალოგური ანალოგური სიგნალის მოდელი ავარგოთ, უნდა ველოდოთ ახალი ყოველი ახალი ნიმუშის აღებას. ზოგადად ეს მეთოდი არ იძლევა კარგვი ხარისხის ანალოგურ სიგნალს, განსაკუთრებით მასინ როცა ნიმუშების აღების სიხშირე არის მაღალი.

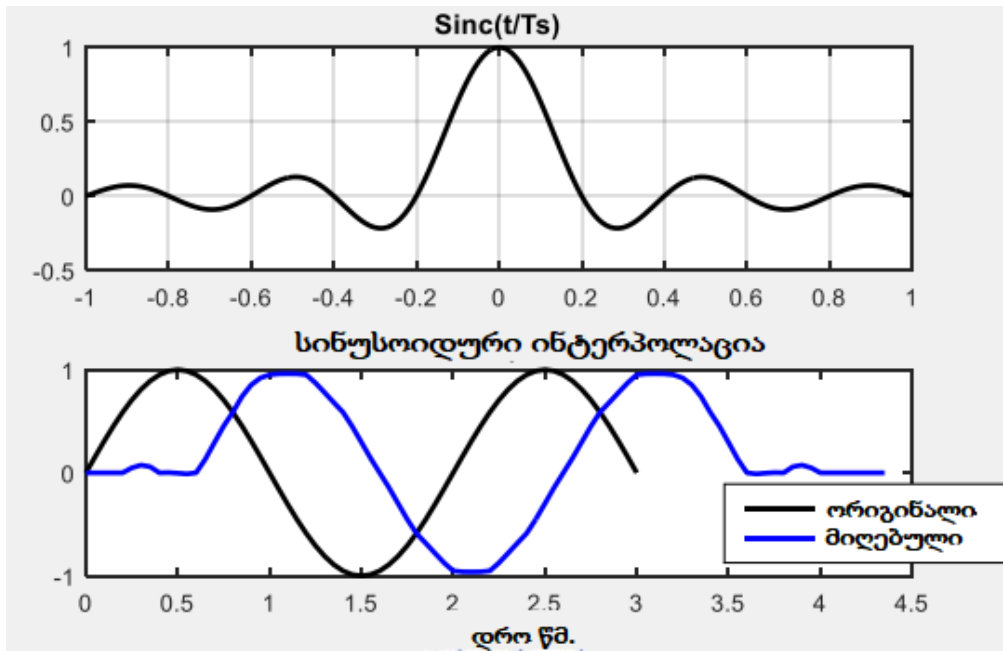


სურათი 25. გრაფიკი წარმოადგენს ორი განსხვავებული ინტერპოლაციის მეთოდის ვიზუალიზაციას, რომლებიც გამოიყენება დისკრეტული სიგნალიდან საწყისი უწყვეტი სიგნალის აღსადგენად (წყარო-ავტორი)

სინუსოიდური ინტერპოლაცია

სინუსოიდური ფუნქცია განვსაზღვრეთ როგორც $sinc(t) = \frac{\sin(\pi t)}{\pi t}$ (Ingle 2015, 80).

სინუსოიდური ინტერპოლაციის ფილტრი იღებს ნიმუშების ფინიტიურ N რაოდენობას. არეზულ ნიმუშებს ამრავლებს N -ის შესაბამის $sinc(t/T_s)$ ფუნქციის მნიშვნელობებზე, შემდეგ აერთიანებს მიღებულ შედეგებს, კრიზავს და ღებულობს შესაბამის ანალოგურ სიგნალს. აქაც ხდება დროითი დაყოფნება, რომელიც არის $N \cdot T_s$. რაც უფრო მაღალია N -ის მნიშვნელობა, მით უფრო მაღალიმიღებული სიგნალის ხარისხი და მით უფრო ემსგავსება იგი დაბალსიხშირულ ფილტრს. სინუსოიდური ინტერპოლაციის ფილტრით მიღებული სიგნალი ნაჩვენებია სურათზე



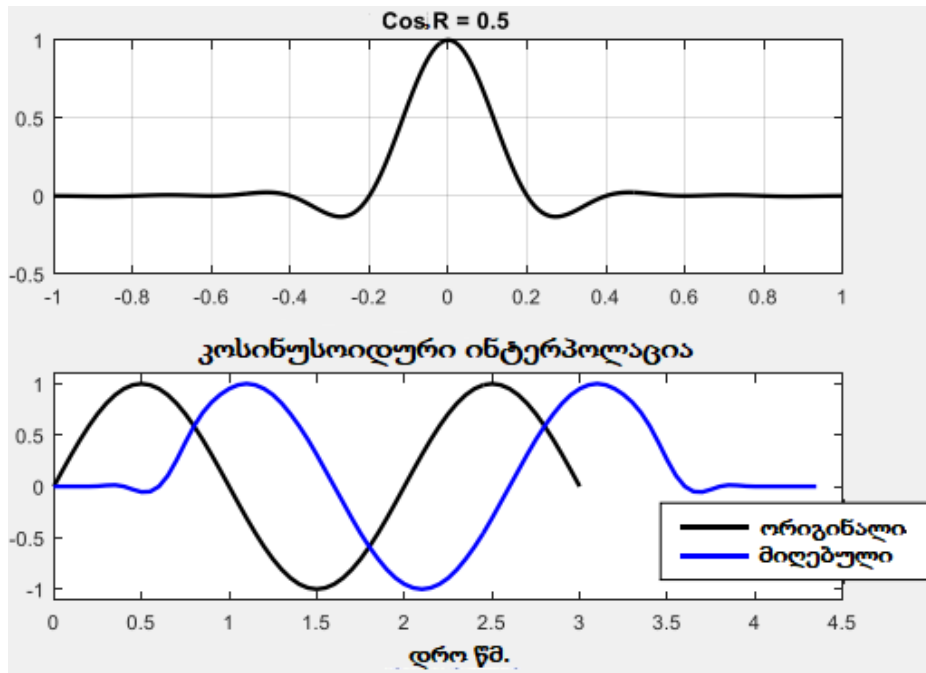
სურათი 26. სინუსოიდური ინტერპოლაციის ფილტრით მიღებული სიგნალი (წყარო-ავტორი)

კოსინუსოიდური ინტერპოლაცია

კოსინუსოიდური ინტერპოლაციის ფილტრი ტენიკით გავს სინუსოიდური ინტერპოლაციის ფილტრს, რომელიც განსაზღვრულია შემდეგნაირად:

$$\text{sinc}(t/T_s) \cdot \frac{\cos(\pi R t/T_s)}{[1-(2R t/T_s)^2]} \quad (\text{Ingle 2015, 102})$$

პარამეტრი R იღებს ნებისმიერ მნიშვნელობას (0;1) ინტერვალიდან. ის ფუნქციაში ფაქტიურად სიმეტრიას ასრულებს. როცა $R=0$, მაშინ ფუნქცია ხდება $\text{sinc}(t/T_s)$ ფუნქცია. სურათზე ნაჩვენებია კოსინუსური ინტერპოლაციის ფილტრით სინუსური ტალღის აღდგენის შედეგი. შესამჩნევია, რომ აღდგენილი ანალოგური სიგნალის ხარისხი გაუმჯობესდა, თუ შევადარებთ იგივე რიგის სინუსოიდური ინტერპოლაციის ფილტრით მიღებულ შედეგებს.



სურათი 27. კოსინუსოიდური ინტერპოლაციის ფილტრით მიღებული სიგნალი (წყარო-ავტორი)

III თავი. მთავარი მათემატიკური კონცეფციები

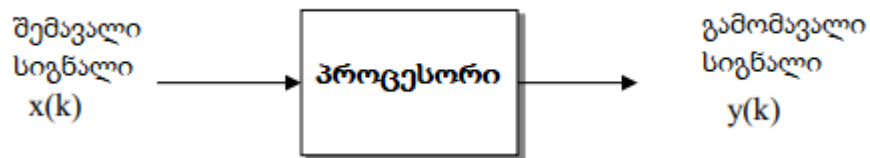
3.1. დისკრეტული სისტემა

ფიზიკური დისკრეტული სისტემა იდეალურია, თუ მას შეუძლია ფიზიკური რეალიზაცია, აქვს მუდმივი პარამეტრები, არის წრფივი და მდგრადი.

დისკრეტული სისტემა ფიზიკურად რეალიზებადად ითვლება, როცა საწყისი ნულოვანი პირობების შემთხვევაში სისტემა არ გამოავლენს რეაქციას მანამ, სანამ მასზე არ მოხდება ზემოქმედება. ამ შემთხვევაში, სისტემის რეაქცია დამოკიდებულია მხოლოდ მიმდინარე და წინა ზემოქმედებებზე, ხოლო მომავალი ზემოქმედების მნიშვნელობაზე იგი გავლენას არ ახდენს.

თუ დისკრეტული სისტემის ძირითადი თვისებები დროში უცვლელია, მას მუდმივი პარამეტრები გააჩნია. ასეთი სისტემა დროში ინვარიანტული იქნება, რაც ნიშნავს, რომ დროში გადატანილი (ან დაყოვნებული) შემავალი სიგნალი უნდა იწვევდეს დროში გადატანილ შესაბამის გამოსავალს. მაგალითად, თუ სისტემის შესავალზე მივაწვდით $x(k)$ და $x(n+k)$ სიგნალებს, ხოლო გამოსავალზე მივიღებთ $y(k)$ და $y(n+k)$ სიგნალებს, მაშინ სისტემა დროში ინვარიანტულია. ეს პირობა უნდა სრულდებოდეს ნებისმიერი n -ის მნიშვნელობისა და ნებისმიერი სიგნალისათვის. ასეთი სისტემა სტაციონარულია.

დისკრეტული სისტემა ფაქტობრივად წარმოადგენს ერთი თანმიმდევრობის (რომელსაც შემავალს უწოდებენ) მეორე თანმიმდევრობაში (რომელსაც გამომავალი ეწოდება) გარდაქმნის ალგორითმს. ანუ, გვაქვს შემდეგი ბლოკ-სქემა.



დიაგრამა 4. დისკრეტული სისტემის ალგორითმი (წყარო-ავტორი)

წრფივი დისკრეტული სისტემა წრფივად ამუშავებს ციფრულ შემავალ სიგნალს, რათა მიიღოს ციფრული გამომავალი სიგნალი, როგორც ილუსტრირებულია სურათზე. სისტემა წრფივია, თუ მისი რეაქცია ადიტიური და ერთგვაროვანია. ტერმინი „ადიტიურობა“ ნიშნავს $f(x_1 + x_2 + \dots + x_n) = f(x_1) + f(x_2) + \dots + f(x_n)$.

იმას, რომ შემავალი სიგნალების ჯამის რეაქცია ტოლი უნდა იყოს ცალკეული სიგნალების რეაქციების ჯამისა. ანუ თუ $f(x)$ აღნიშნავს შემავალი სიგნალის რეაქციას, მაშინ

ტერმინი „ერთგვაროვანი“ ნიშნავს იმას, რომ სისტემის შემავალი სიგნალის მუდმივ c სიდიდეზე ნამრავლის რეაქცია ტოლია ამ მუდმივი სიდიდისა და სიგნალის რეაქციის ნამრავლისა, ანუ $f(cx) = cf(x)$

პრაქტიკაში პარამეტრების მუდმივობა რეალურ ფიზიკურ სისტემებში ხორციელდება. რაც შეეხება წრფივობას, აქ სხვაგვარადაა საქმე. ცნობილია, რომ ყველა ფიზიკური სისტემა არაწრფივია, როცა შემავალი სიგნალი აღწევს ექსტრემალურ მნიშვნელობებს. სიტუაცია რთულდება, როცა არაწრფივობის ეფექტი დროის გარკვეული მომენტიდან იწყებს თანდათანობით გამორჩენას. მიუხედავად ამისა, ბევრი რეალური სისტემა შეიძლება ჩაითვალოს წრფივად, ყოველი შემთხვევაში შემავალი სიგნალის ცვალებადობის გარკვეულ ინტერვალში.

შემავალ/გამომავალ სიგნალთა სხვაობის განტოლება

ეს არის განტოლება, რომელიც აღწერს დისკრეტულ სისტემაში შემავალ სიგნალზე გამომავალი სიგნალის რეაქციას (Ingle 2015, 145):

$$y_1(k) + a_1 y(k-1) + a_2 y(k-2) + \dots + a_N y(k-N) = b_0 x(k) + b_1 x(k-1) + b_2 x(k-2) + \dots + b_L x(k-L)$$

დისკრეტული სისტემის რეგულაცია არის მაქსიმუმი N და L -ს შორის, რაც ნიშნავს, ყველაზე დიდ დაყოვნებას შემავალ ან გამომავალ სიგნალს შორის.

$y(k)$ -გამომავალი სიგნალის გამოთვლა გვაძლევს:

$$y_1(k) = -a_1 y(k-1) - a_2 y(k-2) - \dots - a_N y(k-N) + b_0 x(k) + b_1 x(k-1) + b_2 x(k-2) + \dots + b_L x(k-L)$$

$y(k)$ -გამომავალი სიგნალი არის მანამდე გამომავალი სიგნალების $y(k-1)$ და $y(k-2)$ წრფივი კომბინაცია, ხოლო შემავალი $x(k)$ სიგნალი არის მანამდე შემავალი $(x(k-1), x(k-2), \dots)$ - წრფივი კომბინაცია. პროცესორი გამოითვლის $y(k)$ -ის მნიშვნელობებს სისტემის განტოლების კოეფიციენტების $(a_1, a_2, \dots, b_0, b_1, b_2, \dots)$ -ის გამოყენებით და N ნიმუშების მნიშვნელობები, რომელიც არის წინა შემავალი და გამომავალი სიგნალების შენახული მნიშვნელობები, ისინი საჭიროა მიმდინარე დროის k მნიშვნელობაზე, გამომავალი სიგნალის მნიშვნელობის

გამოთვლისთვის. შემავალი და $\delta(k) = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases}$ გამომავალი მნიშვნელობები
შენახულია სისტემის მეხსიერებაში, რათა ადვილად
ხელმისაწვდომი იყოს პროცესორისთვის, გამოთვლების პროცესში.

ვთქვათ დისკრეტული სისტემის განტოლებაა

$$y(k) + 0,2 \cdot y(k-1) + 0,5 \cdot y(k-2) = x(k) - x(k-1),$$

ვიპოვოთ გამომავალი სიგნალის მნიშვნელობები, შემავალი მნიშვნელობებით

$$x(0) = 1; x(1) = 3; x(k) = 0$$

სადაც $k \geq 2$ უნდა ჩავთვალოთ, რომ სისტემას არ აქვს წინა მდგომარეობა, ყველა
შიდა ენერგია ნულოვანია.

$$y(k) = -0,2 \cdot y(k-1) - 0,5 \cdot y(k-2) + x(k) - x(k-1)$$

$k=0$ - დან დაწყებული ვიპოვოთ $y(k)$ -ს მნიშვნელობებს, ამ პროცესს კარგად შეესაბამება
ტერმინი „რეკურსია“:

$$y(0) = -0,2y(-1) - 0,5y(-2) + x(0) - x(-1) = 0 + 0 + 1 + 0 = 1$$

$$y(1) = -0,2y(0) - 0,5y(-1) + x(1) - x(0) = -0,2(1) - 0 + 3 - 1 = 1,8$$

$$y(2) = -0,2y(1) - 0,5y(0) + x(2) - x(1) = -0,2(1,8) - 0,5(1) + 0 - 3 = -3,86$$

$$y(3) = -0,2y(2) - 0,5y(1) + x(3) - x(2) = -0,2(-3,86) - 0,5(1,8) + 0 - 0 = -0,128$$

იმპულსური მახასიათებლები (სისტემის იმპულსური რეაქცია)

მუდმივპარამეტრიანი წრფივი დისკრეტული სისტემების დინამიური
მახასიათებლები შეიძლება აიხსნას იმპულსური მახასიათებლით, ანუ როგორც მას
ხშირად უწოდებენ, იმპულსური გადაცემის ფუნქციით, რომელიც წარმოადგენს
სისტემის რეაქციას ერთეულოვანი იმპულსის (დელტა-ფუნქციის) ტიპის შეყვანაზე.
სისტემის იმპულსური რეაქცია არის სისტემის პასუხი იმპულსური სიგნალის მიმართ,
როცა სისტემა თავდაპირველად „მშვიდ“ მდგომარეობაშია, ყველა შიდა ენერგია
ნულოვანია. იმპულსური სიგნალი $\delta(k)$ განისაზღვრება როგორც:

დისკრეტული იმპულსი

მაგალითად გამოვითვალოთ

$$y(k) = 0,25 \cdot x(k) + 0,5 \cdot x(k-1) + 0,3 \cdot x(k-2)$$

- სისტემის იმპულსური მახასიათებლები:

$$\begin{aligned}
 y(0) &= 0,25x(0) + 0,5 \times (-1) + 0,3x(-2) = 0,25 \\
 y(1) &= 0,25 \times (1) + 0,5 \times (0) + 0,3 \times (-1) = 0,5 \\
 y(2) &= 0,25 \times (2) + 0,5 \times (1) + 0,3x(0) = 0,3 \\
 y(k) &= 0 \text{ სადაც } k \geq 3
 \end{aligned}$$

დისკრეტული სისტემის სტაბილურობა

დისკრეტული სისტემა სტაბილურია, თუ ნებისმიერი საწყისი ფინიტიური სპექტრის შემავალი სიგნალის შემთხვევაში გამოსული სიგნალიც ფინიტიური სპექტრისაა. დისკრეტული სისტემა სტაბილურია მხოლოდ მაშინ, თუ იმპულსური რეაქცია ნულს უახლოვდება, როდესაც დრო მიისწრაფივის უსასრულობისკენ.

სტაბილური სისტემისთვის, იმპულსური რეაქციის დროთა განმავლობაში ნულსმიახლოება ნიშნავს, რომ სისტემა „წყნარდება“ და არ აგრძელებს გრძელვადიან რეაგირებას. ზემოთ განხილული

$$y(k) = 0.25x(k) + 0.5x(k - 1) + 0.3x(k - 2)$$

სისტემა სტაბილურია.

3.2. Z გარდაქმნა

Z გარდაქმნა არის მათემატიკური პროცესი, რომელიც გამოიყენება დისკრეტული სიგნალების ანალიზისთვის. იგი საშუალებას გვაძლევს გარდავქმნათ დროითი სიგნალი (ანუ როცა გვაქვს სიგნალის ნიმუშები დროით არეში აღებული) კომპლექსური ცვლადის ფორმაში, რაც შესაძლებელს ხდის გავაანალიზოთ სიგნალის სიხშირული კომპონენტები და სისტემის რეაქცია. Z გარდაქმნა ძალიან სასარგებლო ინსტრუმენტია ციფრული სისტემების სტაბილურობის, მისი აღწერისა და ანალიზისთვის. ციფრული სიგნალის Z-გარდაქმნა განისაზღვრება შემდეგნაირად:

$$Z\{f(k)\} = F(z) = \sum_{k=-\infty}^{\infty} f(k)z^{-k} \text{ (Haykin 2010, 234)}$$

მაგალითად ა) შევასრულოთ დისკრეტული იმპულსისთვის $\delta(k) = \begin{cases} 1 & \text{თუ } k = 0 \\ 0 & \text{თუ } k \neq 0 \end{cases}$

Z-გარდაქმნა.

$$Z\{\delta(k)\} = \sum_{k=-\infty}^{\infty} \delta(k)z^{-k} = 1z^{-0} = 1\delta(k) \leftrightarrow 1$$

ბ) შევასრულოთ Z-გარდაქმნა $u(k)$ -ისთვის

$$u(k) = \begin{cases} 1 & \text{თუ } k \geq 0 \\ 0 & \text{თუ } k < 0 \end{cases}$$

$$\mathcal{Z}\{u(k)\} = \sum_{k=-\infty}^{\infty} u(k)z^{-k} = 1z^{-0} + z^{-1} + z^{-2} + \dots = \frac{z}{z-1} \quad u(k) \leftrightarrow \frac{z}{z-1}$$

ძალიან რთულია Z-გარდაქმნები ფორმულით დავითვალოთ, ამისთვის გამოიყენება წინასწარ გამზადებული ცხრილები, რომელიც დანართ 3-შია წარმოდგენილი.

დროით დისკრეტული სიგნალი	$f(k)$	$F(z) = \sum_{k=-\infty}^{\infty} f(k)z^{-k}$
იმპულსი	$\delta(k)$	1
დროით პერიოდული იმპულსი	$\delta(k-n)$	$z^{-n} \cdot 1$
ჩართვის ფუნქცია	$u(k)$	$\frac{z}{z-1}$
დაყოვნებული ჩართვის ფუნქცია	$u(k-n)$	$z^{-n} \cdot \frac{z}{z-1}$
წყვეტის ფუნქცია (Ramp)	$ku(k)$	$\frac{z}{(z-1)^2}$
ექსპონენციალური ფუნქცია	$a^k u(k)$	$\frac{z}{z-a}$
ექსპონენციალური ფუნქცია გამრავლებული დროით მნიშვნელობაზე	$ka^k u(k)$	$\frac{z}{(z-a)^2}$
ექსპონენციალური ფუნქცია გამრავლებული	$\frac{1}{(n-1)!} \left\{ \prod_{i=0}^{n-2} (k-i) \right\} \cdot a^k u(k)$	$\frac{z}{(z-a)^n}$

დროით მნიშვნელობაზე (ძირითადი)		
კოსინუსის ფუნქცია	$\cos(\omega_0 k)$	$\frac{z(z - \cos(\omega_0))}{z^2 - (2\cos(\omega_0))z + 1}$
სინუსის ფუნქცია	$\sin(\omega_0 k)$	$\frac{z\sin(\omega_0)}{z^2 - (2\cos(\omega_0))z + 1}$
კოსინუსის ექსპონენციალური ფაზური მანიპულაცია	$2 \cdot C \cdot a^k \cos(\omega_0 k + \theta)$	$\frac{(C\angle\theta)z}{z - (a\angle\omega_0)}$ $+ \frac{(C\angle - \theta)z}{z - (a\angle - \omega_0)}$

ცხრილი 7. Z გარდაქმნები (Haykin 2010, 238-240)

Z-გარდაქმნის თვისებები

Z-გარდაქმნას აქვს რამდენიმე თვისება, რომელთაგან აქ მხოლოდ ორს, ყველაზე მნიშვნელოვანს განვიხილავთ, დანარჩენი თვისებები მოყვანილია ცხრილში

1. წრფივობა -

$$\text{If } f_1(k) \leftrightarrow F_1(z) \text{ და } f_2(k) \leftrightarrow F_2(z) \text{ მაშინ } a_1 f_1(k) + a_2 f_2(k) \leftrightarrow a_1 F_1(z) + a_2 F_2(z)$$

ყველა ნამდვილი a_1 და a_2 რიცხვისთვის (Haykin 2010, 248)

ეს იმას ნიშნავს, რომ ნებისმიერი დისკრეტული სიგნალის წრფივი კომბინაციის Z-გარდაქმნა ასევე წრფივი კომბინაციაა თითოეული დისკრეტული სიგნალის ინდივიდუალური Z-გარდაქმნის.

2. დროში დაყოვნება - $f(k - n) \leftrightarrow z^{-n} F(z)$

ანუ დისკრეტული სიგნალის Z-გარდაქმნა, რომელიც დაყოვნებულია n ნიმუშით, არის ამ სიგნალის (დაყოვნების გარეშე) Z-გარდაქმნა გამრავლებული z^{-n} -ზე. ოპერატორი z^{-1} დისერტაციასი გამოყენებულია როგორც დროში ერთი ანათვალის დაგვიანების აღსანიშნავად.

თვისება	ფუნქცია	Z გარდაქმნა
წრფივობა	$a_1f_1(k) + a_2f_2(k)$	$a_1F_1(z) + a_2F_2(z)$
დროითი დაყოვნება	$f(k - n)$	$z^{-n}F(z)$
კონვოლუცია	$f(k) * g(k)$ $= \sum_{n=-\infty}^{\infty} f(n)g(k - n)$	$F(z)G(z)$
საწყისი მნიშვნელობის თეორემა	$f(0)$	$\lim_{z \rightarrow \infty} F(z)$
საბოლოო მნიშვნელობის თეორემა	$\lim_{k \rightarrow \infty} f(k)$	$\lim_{z \rightarrow 1} (z - 1)F(z)$

ცხრილი 8. Z გარდაქმნის თვისებები (Haykin 2010, 278)

გადაცემის ფუნქცია

გადაცემის ფუნქცია არის მათემატიკური მოდელი, რომელიც აღწერს სისტემის გამოსავალსა და შესავალს შორის დამოკიდებულებას.

წრფივი სისტემის გადაცემის ფუნქცია აღწერს, როგორ გარდაქმნის სისტემა შესავალ x -ს სიგნალს გამომავალ y სიგნალში, იმ შემთხვევაში თუ სისტემას არ აქვს წინა მდგომარეობა, ყველა შიდა ენერგია ნულოვანია. დისკრეტული სისტემებისთვის გადაცემის ფუნქცია მიღებულია შემავალ-გამომავალ სიგნალებს განტოლების Z-გარდაქმნის საშუალებით და შედეგის გვაწვდის შემდეგი ფორმით (Haykin 2010, 286):

$$Y(z) = H(z) \cdot X(z) \text{ სადაც } H(z) = \frac{Y(z)}{X(z)}$$

$H(z)$ არის დისკრეტული წრფივი სისტემის გადაცემის ფუნქცია (წრფივი სისტემები არის ისეთი სისტემები, სადაც შემავალ და გამომავალ სიგნალებს შორის დამოკიდებულება წრფივია.)

სისტემის პოლუსები - გადაცემის ფუნქციის მნიშვნელის $X(z) - 0$ ფესვები. ისინი მნიშვნელოვანია სისტემის სტაბილურობისა და დინამიური ქცევის განსაზღვრისათვის.

სისტემის ნულეები - გადაცემის ფუნქციის მრიცხველი $Y(z)$ – ში ფესვები. ნულეები გავლენას ახდენენ სისტემის გამოსავალზე სხვადასხვა სიხშირის შემომავალ სიგნალებზე. ნულეების მდებარეობა განსაზღვრავს ამ სიხშირეებზე სისტემის რეაქციის ხარისხს.

გადაცემის ფუნქცია შეიძლება მივიღოთ შემავალ/გამომავალ სიგნალთა სხვაობის განტოლებიდან

$$y(k) + a_1y(k-1) + a_2y(k-2) + \dots + a_Ny(k-N) = b_0x(k) + b_1x(k-1) + b_2x(k-2) + \dots + b_Lx(k-L)$$

Z-გარდაქმნის წრფივობისა და დროითი დაყოვნების თვისებების გამოყენებით მივიღებთ.

$$Y(z) + a_1z^{-1}Y(z) + a_2z^{-2}Y(z) + \dots + a_Nz^{-N}Y(z) = b_0X(z) + b_1z^{-1}X(z) + b_2z^{-2}X(z) + \dots + b_Lz^{-L}X(z)$$

თუ მარჯვენა მხარეს ფრჩხილებს გარეთ გავიტანთ $Y(z)$ –ს და მარცხენა მხარეს გავიტანთ $x(z)$ -ს, გვექნება

$$Y(z)[1 + a_1z^{-1} + a_2z^{-2} + \dots + a_Nz^{-N}] = [b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_Lz^{-L}]X(z)$$

განტოლების ორივე მხარე გავყოთ მარცხენა მხარეს არსებულ პოლინომზე და z-ის უარყოფითი ხარისხების მოსაშორებლად გავამრავლოთ ორივე მხარე z^N , რას მოგვცემს შემდეგს:

$$Y(z) = \frac{b_0 + b_1z^{-1} + \dots + b_Lz^{-L}}{1 + a_1z^{-1} + \dots + a_Nz^{-N}}X(z) = \frac{b_0z^N + b_1z^{N-1} + \dots + b_Lz^{N-L}}{z^N + a_1z^{N-1} + \dots + a_N}X(z)$$

მაშასადამე მივიღეთ წრფივი დისკრეტული სისტემის გადაცემის ფუნქცია

$$H(z) = \frac{b_0 + b_1z^{-1} + \dots + b_Lz^{-L}}{1 + a_1z^{-1} + \dots + a_Nz^{-N}} = \frac{b_0z^N + b_1z^{N-1} + \dots + b_Lz^{N-L}}{z^N + a_1z^{N-1} + \dots + a_N}$$

როგორც ვხედავთ გადაცემის ფუნქცია მჭიდროდაა დაკავშირებული შემავალ/გამომავალ სიგნალთა სხვაობის განტოლებასთან, ამაში კიდევ ერთხელ დავრწმუნდებით თუ ვეცდებით და ვიპოვიოთ

$$y(k) + 0,2y \cdot (k - 1) + 0,5y \cdot (k - 2) = x(k) - x(k - 1)$$

დისკრეტული სისტემის გადაცემის ფუნქცია.

გადაცემის ფუნქციის მრიცხველი განისაზღვრება სხვაობის განტოლების მარჯვენა (შემავალი) მხარედან, ხოლო მნიშვნელი — მარცხენა (გამოსავალი) მხარედან. დროითი დაყოვნებები წარმოიქმნება z^{-1} ოპერატორით:

$$H(z) = \frac{1-z^{-1}}{1+0,2z^{-1}+0,5z^{-2}} = \frac{z^2-z}{z^2+0,2z+0,5}$$

გადაცემის ფუნქცია მნიშვნელოვანია სისტემის სტაბილურობის, სიგნალის რეაქციისა და სიხშირის რეაქციის განსაზღვრისთვის. გადაცემის ფუნქცია მნიშვნელოვანია აგრეთვე ციფრული ფილტრების დიზაინისა და ანალიზისთვის, ასევე სწრაფი ფურიეს გარდქმნისთვის, კორელაციის ან კონვოლუციური ოპერაციების ჩატარებისას.

გამომავალი სიგნალის გამოთვლა გადაცემის ფუნქციის გამოყენებით

დისკრეტული სისტემის მოდელი მოცემულია სხვაობის განტოლებით :

$$y(k) + 0,2y \cdot (k - 1) + 0,5y \cdot (k - 2) = x(k) - x(k - 1)$$

გამოვიყენოთ გადაცემის ფუნქცია და MATLAB-ი, რათა მოვძებნოთ გამომავალი სიგნალის მნიშვნელობები შემავალი სიგნალის ხარჯზე.

$$x(0) = 1; x(1) = 3; x(k) = 0 \text{ სადაც } k \geq 2.$$

ჯერ განვსაზღვროთ გადაცემის ფუნქცია როგორც წინა მაგალითში:

$$H(z) = \frac{1 - z^{-1}}{1 + 0,2z^{-1} + 0,5z^{-2}} = \frac{z^2 - z}{z^2 + 0,2z + 0,5}$$

გადაცემის ფუნქციას შევიტანოთ MATLAB-ში, სიგნალთან ერთად, შემდეგ კი დავამუშაოთ filter ფუნქციით შემდეგნაირად:

```
num_tf = [1 -1 0];
```

```
den_tf = [1 0.2 0.5];
```

```
x = [1 3 0 0 0 0];
```

```
y = filter(num_tf, den_tf, x);
```

გამომავალი ისგნალის მნიშვნელობებია

$$y = 1,0000 \quad 1,800 \quad -3,860 \quad -0,1280 \quad 1,955 \quad -0,3271$$

3.3. კონვოლუცია

კონვოლუცია არის მათემატიკური ოპერაცია, რომელიც ორი სიგნალის გამოყენებით წარმოქმნის მესამე სიგნალს. მას მრავალი გამოყენება აქვს ციფრულ სიგნალთა დამუშავებაში. კონვოლუცია შეიძლება შესრულდეს როგორც უწყვეტ სიგნალებზე, ასევე დისკრეტულ სიგნალებზე. დისერტაციაში განხილულია მხოლოდ დისკრეტული კონვოლუცია. კონვოლუციის ოპერაცია სიმბოლურად აღინიშნება სიმბოლოთი “*” თუმცა იგი არაა გამრავლება. ორი დისკრეტული სიგნალის კონვოლუცია განისაზღვრება შემდეგნაირად:

$$y(k) = x(k) * h(k) = \sum_{n=0}^k x(n)h(k - n) \quad (\text{Ifeachor 2011, 56})$$

სადაც:

- $x(k)$ და $h(k)$ არის ორი სიგნალი (ფუნქცია), რომლებიც კონვოლუციის ოპერაციაში მონაწილეობენ.
- y არის შედეგი, რომელიც მიიღება ამ სიგნალების კონვოლუციის შედეგად.
- n აღნიშნავს დროის რაიმე მნიშვნელობას

თუ გვაქვს სისტემა, რომლის იმპულსური რეაქცია ცნობილია, შეგვიძლია გამოვიყენოთ კონვოლუცია იმის დასადგენად, თუ როგორ უპასუხებს ეს სისტემა ნებისმიერ შემავალ სიგნალს.

უფრო მარტივად გავიაზრებთ თუ გავშლით გამოსახულებას:

$$\begin{aligned}
y(0) &= x(0) \cdot h(0) \\
y(1) &= x(0) \cdot h(1) + x(1) \cdot h(0) \\
y(2) &= x(0) \cdot h(2) + x(1) \cdot h(1) + x(2) \cdot h(0) \\
y(3) &= x(0) \cdot h(3) + x(1) \cdot h(2) + x(2) \cdot h(1) + x(3) \cdot h(0)
\end{aligned}$$

როგორც ვხედავთ ორი განსხვავებული შემავალი სიგნალით შესაძლებელია გამომავალი სიგნალის მნიშვნელობების დადგენა.

გამოვითვალოთ კონვოლუცია ორი შემავალი სიგნალისა MATLAB-ში

```

y(k) = x(k) · h(k) სადაც x(k) = {1 -1 2 0 0 ...} და h(k) = {3 -2 0 0 ...}
x = [1 -1 2]   h=[3 -2]
y = conv(x,h)

```

მიიღება მნიშვნელობები y = 2; -4; 8; -5.

კონვოლუცია გამოიყენება ციფრული გამოსახულების დამუშავებაშიც, ხოლო აუდიო პროგრამებში კონვოლუცია შეიძლება გამოყენებულ იქნას ექო ეფექტებისთვის, რომლებიც წარმოიშობა მუსიკის შესრულებისას ოთახში ან თეატრალურ დარბაზში, რაც ციფრულ აუდიო სიგნალს უფრო რეალისტურს ხდის.

3.4. კორელაცია

სიგნალების დამუშავებაში კორელაცია არის მნიშვნელოვანი ინსტრუმენტი, რომელიც გამოიყენება ორი სიგნალის მსგავსების და მათი ურთიერთდამოკიდებულების შესასწავლად. კორელაცია საშუალებას იძლევა, დავადგინოთ, რამდენად ძლიერად არის სიგნალები ერთმანეთთან დაკავშირებული დროში ან სივრცეში და როგორ ემთხვევა ისინი ერთმანეთს.

სიგნალების დამუშავებაში, კორელაციის გამოყენება ფართოა და მოიცავს სხვადასხვა სფეროს, როგორცაა კომუნიკაცია, რადარების სისტემა, ციფრული ფილტრაცია და გამოსახულების დამუშავება. ძირითადი მახასიათებელი, რომელიც კორელაციას დიდ როლს ანიჭებს სიგნალების დამუშავებაში, არის მისი უნარი აღმოაჩინოს დროითი ცვლილებები ან გადახრები ორ სიგნალს შორის.

არსებობს ორი ძირითადი ტიპის კორელაცია: ავტოკორელაცია და ჯვარედინი კორელაცია (Ifeachor 2011, 90-98)

ავტოკორელაცია არის ერთი სიგნალის თავის თავთან შედარება დროში. ეს პროცესი ხშირად გამოიყენება, რომ გამოვლინდეს სიგნალში განმეორებადი ანათვლები ან პერიოდები. მაგალითად, თუ გვაქვს დროითი სიგნალი, რომელიც შედგება პერიოდული კომპონენტებისგან, ავტოკორელაცია დაგვეხმარება გამოვლინოთ და დავადასტუროთ ეს პერიოდულობა. ანუ სასარგებლო ინსტრუმენტია ისეთი სისტემებისთვის, რომლებიც საჭიროებენ პერიოდულობისა და პერიოდული კომპონენტების აღმოჩენას, მაგალითად, მუსიკის ანალიზში ან ელექტრონული სისტემების ტესტირებაში.

ჯვარედინი კორელაცია არის ორი სხვადასხვა სიგნალის ურთიერთდამოკიდებულების შეფასება. გამოიყენება იმის დასადგენად, თუ რამდენად არის ერთი სიგნალი მეორე სიგნალისგან მიღებული დროში ან სივრცეში გადატანით. ჯვარედინი კორელაცია განსაკუთრებით მნიშვნელოვანია კომუნიკაციებში, სადაც გამოიყენება ორ სიგნალს შორის დაგვიანების ან ფაზური განსხვავების განსაზღვრისთვის. მაგალითად, რადარებში ჯვარედინი კორელაცია გამოიყენება იმისთვის, რომ გამოვლინდეს ობიექტის მანძილი და სიჩქარე, რისთვისაც საჭირო ხდება ორ სიგნალს შორის დროითი ცვლილების პოვნა.

ჯვარედინი კორელაციის ფუნქცია ორ სიგნალს შორის განისაზღვრება როგორც (Ifeachor 2011, 234):

$$R_{xy}(n) = \sum_{k=0}^{N-1} x(k)y(k-n) = \sum_{k=0}^{N-1} x(k+n)y(k)n = -M+1, \dots, N-1$$

სადაც $M = y(k)$ სიგნალის სიგრძეს, $N = x(k)$ სიგნალის სიგრძეს.

განვიხილოთ მოცემული ჯამის რამდენიმე მნიშვნელობა სხვადასხვა n -ისთვის. ეს პროცესი საშუალებას მოგვცემს, ვიზუალურად დავინახოთ, თუ როგორ ხდება ორი სიგნალის დროითი გადანაცვლება და მათი მსგავსების გაანგარიშება.

$$\begin{aligned} R_{xy}(0) &= x(0) \cdot y(0) + x(1) \cdot y(1) + x(2) \cdot y(2) + \dots + x(N-1) \cdot y(N-1) \\ R_{xy}(1) &= x(1) \cdot y(0) + x(2) \cdot y(1) + x(3) \cdot y(2) + \dots + x(N-1) \cdot y(N-2) \\ R_{xy}(2) &= x(2) \cdot y(0) + x(3) \cdot y(1) + x(4) \cdot y(2) + \dots + x(N-1) \cdot y(N-3) \\ R_{xy}(3) &= x(3) \cdot y(0) + x(4) \cdot y(1) + x(5) \cdot y(2) + \dots + x(N-1) \cdot y(N-4) \\ R_{xy}(-1) &= x(0) \cdot y(1) + x(1) \cdot y(2) + x(2) \cdot y(3) + \dots + x(N-2) \cdot y(N-1) \end{aligned}$$

კორელაციის გაანგარიშების დროს, თითოეული $R_{xy}(n)$ წარმოგვიდგენს სიგნალების დროში გადანაცვლების კონკრეტულ შედეგს. $R_{xy}(0)$ -ის შემთხვევა

წარმოადგენს ორი გადაფარვას დროის გარკვეულ მონაკვეთზე, ანუ ერთმანეთს ემთხვევიან.

$R_{xy}(1)$ –ის კი წარმოიშობა მაშინ, როცა $y(k)$ სიგნალი გადაადგილდება ერთი ერთეულით მარჯვნივ, რაც იწვევს იმ პოზიციების გამოთვლას, სადაც ახალი გადაფარვა ხდება, ხოლო $R_{xy}(-1)$ –ისკი წარმოიშობა მაშინ, როცა $y(k)$ სიგნალი გადაადგილდება ერთი ერთეულით მარცხნივ.

ეს პროცესი საშუალებას იძლევა, გამოვლინდეს სიგნალების ურთიერთდამოკიდებულება, ანუ რამდენად ძლიერად არიან ისინი დაკავშირებული დროში გადანაცვლებისას.

3.5. უკუ Z-გარდაქმნა

Z გარდაქმნა არის ფურიეს გარდაქმნის განზოგადება. იგი პროცესია, რომელიც ხდება, როცა სიგნალი გარდაიქმნება ისეთ ფორმაში, რომ მარტივად შეიძლება გავანალიზოთ და გავიგოთ მისი სიხშირული კომპონენტები. მას შემდეგ, რაც სიგნალი Z გარდაქმნის საშუალებით შეიცვალა, მიიღო კომპლექსური ფორმა, **უკუ Z გარდაქმნა** გვაძლევს საშუალებას ისევ დავაბრუნოთ იგი პირვანდელ (დროით არეში) ფორმაში. უკუ Z გარდაქმნის პოვნისთვის რამდენიმე მეთოდი არსებობს. განვიხილოთ ერთ-ერთი მათგანი, რომელიც რამდენიმე ეტაპს მოიცავს:

1. განსაზღვროთ დისკრეტული სისტემის გადაცემის ფუნქცია, $H(z)$;
2. ვიპოვოთ შემავალი სიგნალის Z გარდაქმნა, $x(z)$, დანართში არსებული Z გარდაქმნის ცხრილების გამოყენებით;
3. გამოთვალეთ გამომავალი სიგნალის Z გარდაქმნა, $Y(z) = H(z) \cdot x(z)$;
4. $Y(z)/z$ - ისთვის მოვძებნოთ პოლუსები MATLAB-ის residue ფუნქციის გამოყენებით;
5. მე-4 ეტაპზე მიღებული შედეგები გადავამრავლოთ;
6. გამოვიყენოთ დანართში არსებული Z გარდაქმნის ცხრილი, რათა დავადგინოთ $y(k)$ -ის მნიშვნელობები.

მაგალითად ვიპოვოთ ჩართვის ფუნქცია მოცემული დისკრეტული სისტემისთვის, ანუ ვიპოვოთ როგორ რეაგირებს სისტემა კონკრეტულ ცვლილებებზე დროში (Ifeachor 2011, 240):

$$y(k) + 0,2 \cdot y(k - 1) + 0,5 \cdot y(k - 2) = x(k) - x(k - 1)$$

მოცემული სისტემის გადაცემის ფუნქციაა (Ifeachor 2011, 239):

$$H(z) = \frac{1 - z^{-1}}{1 + 0,2z^{-1} + 0,5z^{-2}} = \frac{z^2 - z}{z^2 + 0,2z + 0,5}$$

მისი Z გარდაქმნა ნაჩვენებია ფორმულებში:

$$Y(z) = \left[\frac{z^2 - z}{z^2 + 0,2z + 0,5} \right] \left[\frac{z}{z - 1} \right]$$

$$Y(z)/z = \left[\frac{z^2 - z}{z^2 + 0,2z + 0,5} \right] \left[\frac{1}{z - 1} \right]$$

MATLAB-ში შევიყვანოთ მოცემული წილადის მრიცხველი და მნიშვნელი და გამოვიყენოთ ფუნქცია residue:

```
num = [1 -1 0]; dn = conv([1 0,2 0,5],[1 -1]);
[coeff poles]= residue(num, dn)
```

Coeffic = 0.0000	poles = 1.0000
0.5000 + 0.0714i	-0.1000 + 0.7000i
0.5000 - 0.0714i	-0.1000 - 0.7000i

უნდა გარკვეოთ, როგორ იკვეთება მრიცხველის პოლინომის კოეფიციენტები ამ სისტემის პოლუსებთან:

$$Y(z)/z = \frac{0}{z - 1} + \frac{0,5 + 0,0714i}{z - (-0,1 + 0,7i)} + \frac{0,5 - 0,0714i}{z - (-0,1 - 0,7i)}$$

კომპლექსური რიცხვები გარდავქმნათ და გავამრავლოთ z-ზე, გვექნება:

$$Y(z) = \frac{0 \cdot z}{z-1} + \frac{(0,5051 \angle 0,142\text{rad})z}{z - (0,7071 \angle 1,71\text{rad})} + \frac{(0,5051 \angle -0,142\text{rad})z}{z - (0,7071 \angle -1,71\text{rad})}$$

თუ გამოვიყენებთ Z გარდაქმნის ცხრილს, მაშინ გვექნება სისტემის გამოსასვლელზე შემდეგი მნიშვნელობები:

$$y(k) = 2(0,5051) \cdot (0,7071)^k \cdot \cos(1,710k + 0,1420)$$

IV თავი - ციფრული ფილტრები სასრული იმპულსური მახასიათებლებით

მე-4 თავში წარმოდგენილია ფილტრაციის ტექნიკა, რეალიზებულია რამდენიმე ფილტრი, ჩატარებულია ექსპერიმენტები და წარმოდგენილია შედეგები.

სიგნალების დამუშავებისას ფილტრები გამოიყენება სხვადასხვა სიხშირეების დასაცალკევებლად. "ფილტრაცია" აღნიშნავს დროით არეში სიგნალების დამუშავების პროცესს, რომლის შედეგადაც იცვლება საწყისი სიგნალის სპექტრული შემადგენლობა. ამ პროცესში სიგნალიდან არასასურველი სიხშირის კომპონენტები ამოიღება ან შემცირდება. ფილტრები ეხმარებიან სიგნალის გარკვეული კომპონენტების გამოყოფაში და სხვა კომპონენტების გაძლიერებაში.

ფილტრები შეიძლება იყოს **ანალოგური** ან **ციფრული**. ანალოგური ფილტრები ორ ძირითად ტიპად იყოფა: პასიურ და აქტიურ ფილტრებად. პასიური ფილტრები მხოლოდ პასიური კომპონენტებისგან შედგება, როგორცაა რეზისტორები, ინდუქტორები და კონდენსატორები. აქტიური ფილტრები კი, გარდა ამ კომპონენტებისა, მოიცავს აქტიურ ელემენტებს, როგორცაა ოპერაციული გამაძლიერებლები, რაც მათ დამატებით უპირატესობებს ანიჭებს.

ანალოგური ფილტრები უწყვეტ სიგნალებს ფილტრავს, ხოლო ციფრული ფილტრები მუშაობს დისკრეტულ ანათვლებზე. ციფრული ფილტრები პროგრამული პაკეტია და გააჩნიათ მრავალი უპირატესობა ანალოგურ ფილტრებთან შედარებით. უპირველესად, ისინი გამოირჩევიან მაღალი სტაბილურობით და სიზუსტით სხვადასხვა პირობებში. ამის გამო, ციფრული ფილტრები უფრო ზუსტია, რადგან მათი

სიზუსტე დამოკიდებულია მხოლოდ არითმეტიკული ოპერაციების სიზუსტეზე და არა ფიზიკური კომპონენტების მახასიათებლებზე.

ანალოგური ფილტრის მახასიათებლები ხშირად დამოკიდებულია კომპონენტების სტაბილურობაზე, ტემპერატურის ცვლილებაზე და დროთა განმავლობაში კომპონენტების დეგრადაციაზე, რაც გავლენას ახდენს ფილტრის მუშაობაზე. ციფრული ფილტრები, რომლებიც ფაქტიურად პროგრამული კოდებია, ამგვარი ზემოქმედებებისგან დაცულია და მათი პარამეტრები დროის განმავლობაში არ იცვლება.

ამრიგად, ციფრული ფილტრები განსაკუთრებულ უპირატესობას ანიჭებენ სისტემებს, სადაც საჭიროა ფილტრის ზუსტი და სტაბილური მოქმედება. ისინი უზრუნველყოფენ სიგნალის დისკრეტულ დამუშავებას, რაც მათ მოქნილობას და სიზუსტეს ანიჭებს.

თავში განხილულია მხოლოდ ციფრული ფილტრები.

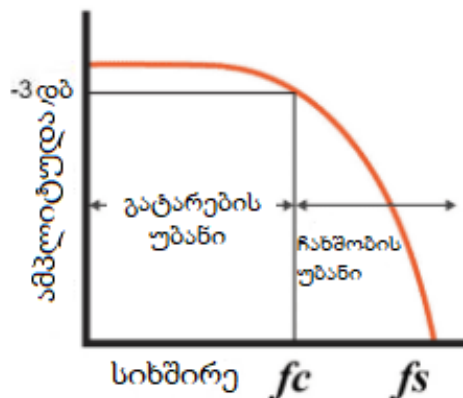
გატარებული სიხშირის მიხედვით, ფილტრები შეიძლება დაიყოს შემდეგ კატეგორიებად (Ingle 2015, 278):

- დაბალსიხშირიანი ფილტრები;
- მაღალსიხშირიანი ფილტრები;
- ზოლური ფილტრები (მათ შორის გამშვები და შემაკავებელი ფილტრები);
- ფაზური ფილტრები.

ფილტრები ხასიათდებიან ე.წ. მოჭრის სიხშირით (f_0). მაგალითად, დაბალსიხშირიანი ფილტრი გატარებს ყველა სიხშირეს ნულიდან მოჭრის სიხშირემდე (f_0), ხოლო უფრო მაღალ სიხშირეებს ბლოკავს. მაღალსიხშირიანი ფილტრი, პირიქით, f_0 -ზე ნაკლები სიხშირის სიგნალებს ან არ ატარებს, ან ცუდად ატარებს, ხოლო f_0 -ის ზემოთ მდებარე სიხშირეებს კარგად გატარებს. ზოლური ფილტრი გატარებს სიგნალს მხოლოდ გარკვეულ სიხშირეთა ზოლში, მაგალითად ω_1 და ω_2 შორის, და ამ ზოლის გარეთ სიგნალს ბლოკავს. ასეთი ფილტრები ცნობილია როგორც გამშვები ზოლური ფილტრები. მეორე ტიპის ზოლური ფილტრი, რომელსაც შემაკავებელი ან რეჟიმექტორული ფილტრი ეწოდება, მოქმედებს საპირისპიროდ: იგი ზოლში (ω_1 -დან ω_2 -მდე) სიგნალს ახშობს, ხოლო ზოლის გარეთ სიგნალებს გატარებს.

გადაცემის კოეფიციენტის (Gain) მახასიათებელი, როგორც სიხშირის ფუნქცია, განსაზღვრავს ფილტრების რიგს. პასიური ფილტრების შემთხვევაში, დამატებითი რეაქტიული კომპონენტის ჩართვა ფილტრის რიგის გაზრდას იწვევს, რაც ამალღებს ფილტრის მახასიათებლების ეფექტურობას. წარმოიდგინეთ, რომ ფილტრი არის ერთი ტიპის ბადის ქსელი, რომელიც ფარავს გარკვეულ ხმაურს. თუ ბადე არის პირველი რიგის, იგი ბლოკავს გარკვეულ ხმაურს, მაგრამ ეს ბლოკირება არაა ძალიან ძლიერი. მეორე რიგის ბადე უფრო ეფექტურად ბლოკავს ხმაურს. თუ ბადეს კიდევ ერთ დამატებით ფენას დავამატებთ, ბლოკირების უნარი კიდევ უფრო გაიზრდება, რაც ფილტრის მახასიათებლებს აუმჯობესებს.

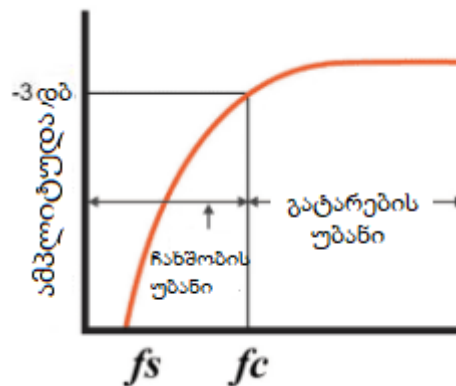
დაბალსიხშირული ფილტრები (low-Pass Filters): არის ელექტრული ან ციფრული ფილტრები, რომლებიც გატარებს დაბალი სიხშირის სიგნალებს და ბლოკავს ან ამცირებს მაღალი სიხშირის სიგნალებს. ისინი გამოიყენება, როდესაც საჭიროა სიგნალის მხოლოდ დაბალი სიხშირის კომპონენტების შენარჩუნება და მაღალი სიხშირის ხმაურის ან სხვა არასასურველი კომპონენტების მოცილება. ფილტრი შეუფერხებლად გატარებს სიგნალებს სიხშირით f_c -მდე. f_c სიხშირეზე ფილტრი იწყებს სიგნალის ჩახშობას. სიგნალები, რომელთა სიხშირე f_c -ს აღემატება, მნიშვნელოვნად მცირდება ან სრულად ბლოკირდება.



სურათი 28. გრაფიკი წარმოადგენს ტიპური დაბალგამტარი ფილტრის (Low-Pass Filter) ამპლიტუდურ-სიხშირულ მახასიათებელს. აჩვენებს, თუ როგორ იცვლება ფილტრის გამტარობა (ამპლიტუდა) სიხშირის მიხედვით (Ingle 2015, 280):.

მაღალსიხშირული ფილტრები (High-pass): მაღალსიხშირიანი ფილტრები ხშირად

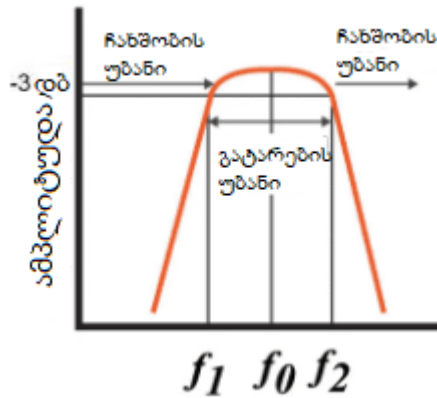
გამოიყენება ისეთი სისტემებში, სადაც საჭიროა დაბალსიხშირიანი ხმაურის მოცილება და მაღალსიხშირიანი სიგნალების შენარჩუნება, იგი f_s -ზე ნაკლები სიხშირის სიგნალებს ან არ ატარებს, ან ცუდად ატარებს, ხოლო f_c -ის ზემოთ მდებარე სიხშირეებს კარგად ატარებს.



სურათი 29. გრაფიკი წარმოადგენს ტიპური მაღალგამტარი ფილტრის (High-Pass Filter) ამპლიტუდურ-სიხშირულ მახასიათებელს. ის აჩვენებს, თუ როგორ იცვლება ფილტრის გამტარობა (ამპლიტუდა) სიხშირის მიხედვით (Ingle 2015, 280):

ზოლური ფილტრები (band-pass):

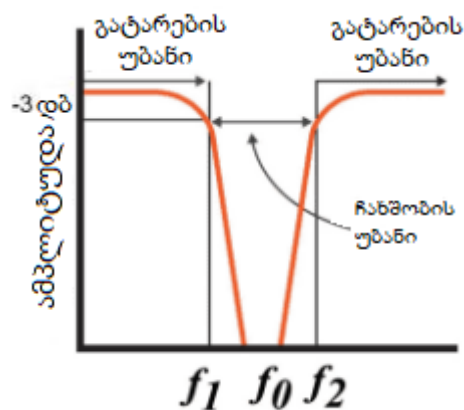
არის ფილტრი, რომელიც გატარებს სიგნალებს გარკვეულ სიხშირულ დიაპაზონში და ბლოკავს ან ამცირებს სიგნალებს, რომლებიც ამ დიაპაზონის გარეთაა. ზოლური ფილტრი ერთდროულად აერთიანებს როგორც დაბალსიხშირიანი, ასევე მაღალსიხშირიანი ფილტრების მახასიათებლებს. ზოლური ფილტრი კარგად გაატარებს სიგნალებს ორ განსაზღვრულ სიხშირეს შორის, რომელთაც ეწოდება ქვედა მოჭრის სიხშირე (f_1) და ზედა მოჭრის სიხშირე (f_2). ამ დიაპაზონში სიგნალები თითქმის უცვლელად გადის. მისი მთავარი მახასიათებელია ცენტრალური სიხშირე f_0 , (ფილტრის "ცენტრი"), მაგ თუ ცენტრალური სიხშირე არის 50 ჰც, ეს ფილტრი გაატარებს სიხშირეებს ქვედა მოჭრის სიხშირიდან 48 ჰც-დან ზედა მოჭრის სიხშირემდე 52 ჰც-მდე, მაშინ ეს 4 ჰც-ის დიაპაზონი 50 ჰც-ის გარშემო **ზოლური სიგანეა**.



სურათი 30. გრაფიკი წარმოადგენს ტიპური ზოლგამტარი ფილტრის (Band-Pass Filter) ამპლიტუდურ-სიხშირულ მახასიათებელს. იგი აჩვენებს, თუ როგორ იცვლება ფილტრის გამტარობა (ამპლიტუდა) სიხშირის მიხედვით (Ingle 2015, 281):

ჩამჭრელი ფილტრი (Notch Filter) ზოლური ფილტრის ნაირახეობაა, არის ფილტრი, რომელიც მიზნად ისახავს კონკრეტული სიხშირის ან სიხშირების დიაპაზონის შემცირებას ან მთლიანად გაქრობას სიგნალიდან. ანუ ეს ისეთი ფილტრია, რომელიც ჭრის ან ბლოკავს სიგნალებს გარკვეულ სიხშირულ დიაპაზონში და ატარებს სიგნალებს, რომლებიც ამ დიაპაზონის გარეთაა.

მისი მთავარი მახასიათებელია ცენტრალური სიხშირე, (ფილტრის "ცენტრი"), მაგ თუ ცენტრალური სიხშირე არის (f_0) 50 ჰც, ამ ფილტრის მიერ შემცირდება სიხშირეები (f_1) 48 ჰც-დან (f_2) 52 ჰც-მდე, მაშინ ეს 4 ჰც-ის დიაპაზონი 50 ჰც-ის გარშემო **ზოლური** სიგანეა.



სურათი 31. ჩამჭრელი ფილტრი (Notch Filter) ზოლური ფილტრის ნაირახეობა (Ingle 2015, 283):

4.1. ფილტრი სასრული იმპულსური მახასიათებლით (სიმ)

სიმ ფილტრები მუშაობენ ისე, რომ შემავალი სიგნალის დამუშავებას ახდენენ ფილტრის კოეფიციენტებით და მანამდე შემავალი ანათვლების საფუძველზე.

ფილტრის მუშაობა სასრული იმპულსური მახასიათებლით (სიმ) შეიძლება აღიწეროს ფორმულით (Ingle 2015, 288):

$$y(k) = b_0x(k) + b_1x(k-1) + b_2x(k-2) + \dots + b_Nx(k-N)$$

- b_0, b_1, \dots, b_N ეს არიან ფილტრის კოეფიციენტები. ფილტრის კოეფიციენტების რაოდენობა არის $N+1$.
- N - ფილტრის რიგი. შესაბამისად, ფილტრის კოეფიციენტების რაოდენობა = ფილტრის რიგს + 1.
- **სიმ ფილტრის დიზაინი** მოიცავს ფილტრის რიგის განსაზღვრას და ფილტრის კოეფიციენტების კონკრეტულ ნაკრებზე მუშაობას, რათა დააკმაყოფილოს მოცემული ფილტრაციის სპეციფიკა.
- ფილტრის გამოსასვლელზე მიიღება სიგნალი შემდეგნაირად: მიმდინარე შემავალი ანათვლები და მანამდე შემავალი სიგნალის N ანათვალი მრავლდება ფილტრის კოეფიციენტებზე და მიღებული შედეგები იკრიბება
- ფილტრის რეალიზაციისათვის საჭიროა რომ ფილტრის კოეფიციენტები, მიმდინარე სიგნალის ანათვლები და მანამდე შემავალი ანათვალი შენახული იყოს მეხსიერებაში. ფილტრის გამოსავალი არ არის დამოკიდებული წარსულ მნიშვნელობებზე.

სიმ ფილტრის იმპულსური მახასიათებლები

მუდმივპარამეტრიანი წრფივი დისკრეტული სისტემების დინამიური მახასიათებლები შეიძლება აღიწეროს იმპულსური მახასიათებლით, ანუ როგორც მას ხშირად უწოდებენ, იმპულსური გადაცემის ფუნქციით, რომელიც წარმოადგენს სისტემის რეაქციას ერთეულოვანი იმპულსის (დელტა-ფუნქციის) ტიპის შეყვანაზე. სისტემის იმპულსური რეაქცია არის სისტემის პასუხი იმპულსური სიგნალის მიმართ,

როცა სისტემა თავდაპირველად „მშვიდ“ მდგომარეობაშია, ყველა შიდა ენერგია ნულოვანია. იმპულსური სიგნალი $\delta(k)$ განისაზღვრება როგორც:

დისკრეტული იმპულსი

როცა ფილტრის გამოსასვლელზე მიღებული სიგნალის მნიშვნელობები

დამოკიდებულია ეხლა შეტანილ და მანამდე შეტანილ მნიშვნელობებზე და არაა

დამოკიდებული მანამდე $\delta(k) = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases}$ გამომავალ სიგნალის

მნიშვნელობებზე (არარეკურსიული სისტემა), ამ შემთხვევაში იმპულსური რაგირება

ემთხვევა შემავალ/გამომავალ სიგნალთა სხვაობის განტოლების კოეფიციენტებს :

$$y_1(k) + a_1 y(k-1) + a_2 y(k-2) + \dots + a_N y(k-N) = b_0 x(k) + b_1 x(k-1) + b_2 x(k-2) + \dots + b_L x(k-l)$$

$$h(k) = [b_0 \quad b_1 \quad b_2 \quad \dots \quad b_N]$$

სიმ ფილტრის გადაცემის ფუნქცია

დისკრეტული სისტემებისთვის ფილტრის გადაცემის ფუნქცია მიიღება შემავალ-გამომავალ სიგნალთა განტოლების Z-გარდაქმნის საშუალებით და შედეგის გვაწვდის შემდეგი ფორმით (Ingle 2015, 296):

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_N z^{-N}}{1} = \frac{b_0 z^N + b_1 z^{N-1} + b_2 z^{N-2} + \dots + b_N}{z^N}$$

სიმ ფილტრის სტაბილურობა

სიმ ფილტრების სტაბილურობა განპირობებულია მათი პოლუსების განლაგებით ნულოვანი წერტილში, რაც უზრუნველყოფს მათ მუდმივ სტაბილურობას. ეს არის ერთ-ერთი მთავარი უპირატესობა სიმ ფილტრების სხვა ტიპის ფილტრებთან შედარებით.

სიმ ფილტრის გამოსასვლელზე მიღებული მნიშვნელობები

ფილტრის გამოსასვლელზე მიღებული მნიშვნელობები შეიძლება დავითვალოთ სხვადასხვა მეთოდით, მაგ:

- შემავალ/გამომავალ სიგნალთა სხვაობის განტოლებით;
- კონვოლუციით $y(k) = x(k) * h(k)$;
- უკუ Z გარდაქმნით $Y(z) = X(z)H(z)$;
- MATLAB-ის ბრძანებით $y = \text{filter}(\text{num_tf}, \text{den_tf}, x)$;
- ფურიეს სწრაფი გარდაქმნით.

რიცხვითი ექსპერიმენტი :

მოცემული ფორმულა აღწერს სიმ ფილტრს

ა) ვიპოვოთ ფილტრის იმპულსური მახასიათებლები და ფილტრის რიგი:

$$y(k) = \frac{1}{3}x(k) + \frac{1}{3}x(k-1) + \frac{1}{3}x(k-2)$$

ფილტრის რიგი როგორც ვიცით არის ყველაზე დიდი დაყოვნება შემავალ სიგნალის ანათვლებს შორის, ანუ $N = 2$, გამოდის ფილტრი არის მეორე რიგის

ბ) ვიპოვოთ ფილტრის გადაცემის ფუნქცია და მისი პოლუსები

ფილტრის გადაცემის ფუნქცია მიიღება შემავალ-გამომავალ სიგნალთა განტოლების Z-გარდაქმნის საშუალებით, ამიტომ გვექნება

$$H(z) = \frac{\frac{1}{3} + \frac{1}{3} \cdot z^{-1} + \frac{1}{3} \cdot z^{-2}}{1} = \frac{\frac{1}{3} \cdot z^2 + \frac{1}{3} \cdot z + \frac{1}{3}}{z^2}$$

ფილტრის პოლუსებია (მნიშვნელის ფესვები) 0 და 0. აღვნიშნოთ აქვე, რომ სიმ ფილტრისთვის ყველა პოლუსი არის ნულოვანი (0), რაც ნიშნავს, რომ ისინი ყველა მდებარეობს სიხშირულ არეში ნულოვან წერტილზე. ეს უზრუნველყოფს ფილტრის სტაბილურობას და ადვილ კონტროლს.

გ) ვიპოვოთ ფილტრის გამომავალი მნიშვნელობები შემავალი მნიშვნელობისთვის $x(k) = \{1 \ 3 \ 6 \ -1 \ 5\}$.

გამომავალი მნიშვნელობები შეიძლება გათვალოს სხვადასხვა მეთოდების გამოყენებით. ამ შემთხვევაში გამოვიყენოთ კონვოლუცია (MATLAB-ში).

$$x = [1 \ 3 \ 6 \ -1 \ 5];$$

$$h = [1/3 \ 1/3 \ 1/3];$$

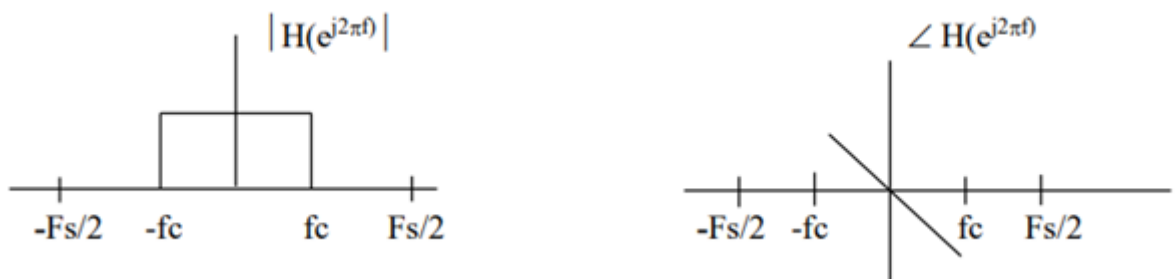
$$y = \text{conv}(x, h);$$

გამომავალი მნიშვნელობებია $y = [0.33333 \ 1.66667 \ 4.00000 \ 2.00000 \ 2.33333 \ 1.33333 \ -1.00000]$

4.2. დაბალსიხშირული სიმ ფილტრის გენერირება, რიცხვითი ექსპერიმენტები MATLAB-ში და შედეგები, შედეგების ანალიზი

ფანჯრის მეთოდზე დაფუძნებული ფილტრის დიზაინი ფურიეს მწკრივების გამოყენებით

ფანჯრის მეთოდზე დაფუძნებული დიზაინი იწყება იდეალური ფილტრის კონსტრუირებით და შემდეგ გარდაიქმნება პრაქტიკულ ფილტრად. ქვემოთ ნაჩვენებია იდეალური დაბალგამტარული ციფრული ფილტრის ამპლიტუდისა და ფაზის რეაქცია.



სურათი 32. იდეალური დაბალგამტარული ციფრული ფილტრის ამპლიტუდისა და ფაზის რეაქცია. (Ingle 2015, 301):

სადაც $f_c =$ სასურველი ანალოგური შეზღუდვის სიხშირე (cutoff frequency), რომელიც განსაზღვრავს, თუ რომელი სიხშირეები უნდა გაიაროს ან დაბლოკოს ფილტრმა.

$F_s =$ დისკრეტიზაციის სიხშირე (მითითებს რამდენ ანათვალს იღებს სისტემა სისტემა ერთ წამში.)

იდეალური ფილტრის იმპულსური რეაქცია (Impulse Response) შეიძლება გამოვთვალოთ ფურიეს მწკრივად გაშლის შეზღუდული ოპერაციის გამოყენებით:

$$h(k) = \int H(e^{j2\pi fk}) e^{j2\pi fk} df$$

ინტეგრალის გამოთვლა გვამძლევს შემდეგ გამოსახულებას:

$$h(k) = \frac{\sin\left(2\pi \frac{f_c}{F_s} k\right)}{\pi k} = 2 \left(\frac{f_c}{F_s}\right) \text{sinc}\left(2 \cdot \frac{f_c}{F_s} k\right) \quad k = 0, \pm 1, \pm 2, \dots, \pm \infty$$

$$\text{sinc}(x) = \sin(\pi x)/(\pi x)$$

იდეალური დაბალგამტარი ფილტრის კონცეპტუალური მოდელი საშუალებას გვაძლევს გავიგოთ მისი მახასიათებლები, მაგრამ პრაქტიკულ აპარატურაში მისი განხორციელება შეუძლებელია კოეფიციენტების უსასრულო რაოდენობის და დროითი სისტემის მახასიათებლების გამო. მომდევნო ფორმული შეგიძლიათ გამოთვალოთ იმპულსური რეაქციის და შემავალი სიგნალის კომბინირებული ეფექტი, რაც გვაწვდის იდეალური დაბალგამტარი ფილტრის შედეგს. თუმცა, პრაქტიკულ რეალობაში, უსასრულო კოეფიციენტები და დროითი სისტემის პრობლემები არ გვაძლევს საშუალებას ფილტრის სრულყოფილ რეალიზაცია მოვახდინოთ:

$$h(k) = \dots h(-2) \cdot x(k+2) + h(-1) \cdot x(k+1) + h(0) \cdot x(k) + h(1) \cdot x(k-1) + h(2) \cdot x(k-2) + \dots$$

როგორც ვხედავთ, ფილტრაციის შედეგის გამოთვლისთვის, საჭიროა უსასრულო რაოდენობის გამრავლების და მიმატების შესრულება (რაც რა თქმა უნდა, შეუძლებელია). ასევე, უნდა ვიცოდეთ სიგნალის ყველა მომავალი მნიშვნელობა; თუმცა რეალურად ჩვენ გვაქვს წვდომა მხოლოდ მიმდინარე ფილტრის ამჟამინდელ და ფილტრში შემავალი სიგნალის წარსულ მნიშვნელობებზე.

ეს პრობლემები შეიძლება გადაწყდეს, თუ ჯერ შევზღუდავთ $h(k)$ -ს, რათა ის იყოს ფინიტური, შემდეგ კი გადავანაცვლებთ $h(k)$ -ს მარჯვნივ, რათა სისტემა დროით კაუზალური გახდეს. ცხადია, ჩვენ აღარ გვექნება იდეალური დაბალგამტარი ფილტრი. ფილტრის კოეფიციენტების შემოკლება $k = -N/2, \dots, 0, \dots, +N/2$ მდე და შემდეგ მათი მარჯვნივ გადატანა $N/2$ -ით გვაძლევს მე-N რიგის ფიზიკურად რეალიზებად დაბალგამტარული ფილტრის კოეფიციენტებს:

$$h(k) = \frac{\sin\left(2\pi \left(\frac{f_c}{F_s}\right) \left(k - \frac{N}{2}\right)\right)}{\pi \left(k - \frac{N}{2}\right)} = 2 \left(\frac{f_c}{F_s}\right) \text{sinc}\left(2 \left(\frac{f_c}{F_s}\right) \left(k - \frac{N}{2}\right)\right) \quad k = 0, 1, 2, \dots, N$$

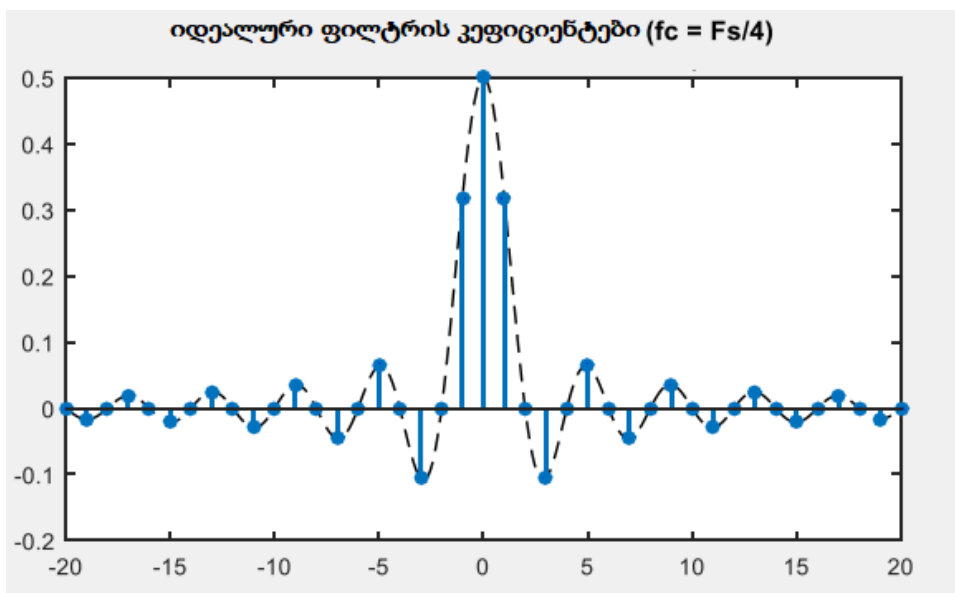
$$\text{სადაც } h\left(\frac{N}{2}\right) = 2\left(\frac{f_c}{F_s}\right)$$

ფილტრის კოეფიციენტების შემოკლებისა და მათი კაუზალურ სისტემად გადაქცევის პროცესი ნაჩვენებია სურათში.

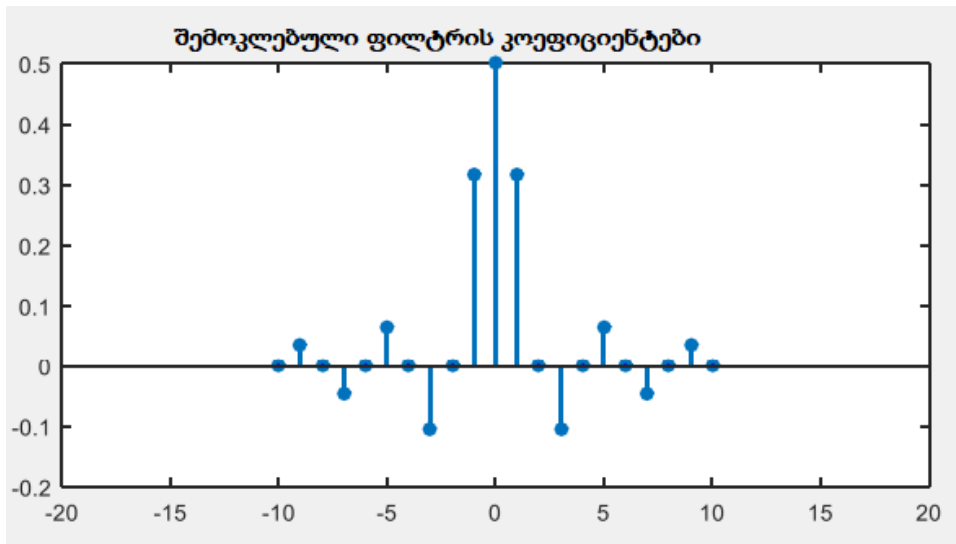
პირველ სურათზე ნაჩვენებია იდეალური ფილტრის კოეფიციენტები, რომლებიც $k = +20$ -მდეა გაგრძელებული. სინამდვილეში, ეს კოეფიციენტები უნდა გაგრძელებულიყო უსასრულობამდე ($+\infty$) მაგრამ ამის ჩვენება რთულია, ამიტომ ისინი შეწყვეტილია $k = +20$ -ზე.

მეორე სურათზე ფილტრი გახდა შეზღუდული ზომის (ფინიტური), რადგან კოეფიციენტები შემოკლებულია $k = +10$ -მდე. ეს იმას ნიშნავს, რომ აღარ არის საჭირო უსასრულო რაოდენობის კოეფიციენტების გამოყენება და მისი გამოთვლა შესაძლებელია.

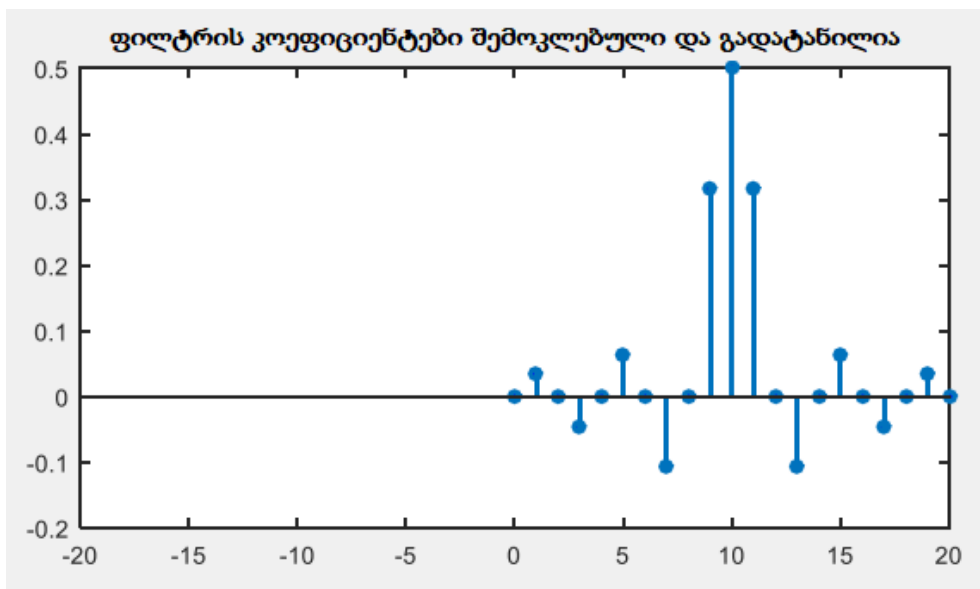
მესამე სურათზე ფილტრის კოეფიციენტები გადატანილია მარჯვნივ 10-ით, რაც ფილტრს დროით კაუზალურ სისტემად აქცევს. შედეგად, ვიღებთ მე-20 რიგის რეალიზებად დაბალგამტარ სიმ ფილტრს.



სურათი 33. იდეალური დაბალსიხშირული ფილტრის კოეფიციენტები ($f_c = F_s/4$) (წყარო-ავტორი)



სურათი 34. შემოკლებული ფილტრის კოეფიციენტები (წყარო-ავტორი)



სურათი 35. ფილტრის კოეფიციენტები შემოკლებულია და გადატანილია მარჯვნივ 10 ერთეულით. (წყარო-ავტორი)

რიცხვითი ექსპერიმენტი 1:

ა) გამოვთვალოთ მე-10 რიგის ციფრული დაბალსიხშირული სიმ ფილტრის კოეფიციენტები, თუ ნიმუშების აღების სიხშირე 44.1 კჰც-ია და მოჭრის სიხშირე 20 კჰც-ია.

ბ) დავადგინოთ განსხვავების განტოლება ფილტრის გამომავალი სიგნალის $y(k)$ -ს გამოსათვლელად

გ) მოვძებნოთ ფილტრის გადაცემის ფუნქცია.

დ) ავსაგოთ მე-10 რიგის სიმ ფილტრის სიხშირის რეაქცია და შევადაროთ იგი იდეალურ დაბალსიხშირულ ფილტრს.

ა) ფილტრის კოეფიციენტები შეიძლება გამოთვალო შემოკლებული და გადანაცვლებული იდეალური იმპულსური რეაქციიდან განტოლებით:

$$h(k) = \frac{\sin\left(2\pi\left(\frac{f_c}{F_s}\right)\left(k - \frac{N}{2}\right)\right)}{\pi\left(k - \frac{N}{2}\right)} = 2\left(\frac{f_c}{F_s}\right) \operatorname{sinc}\left(2\left(\frac{f_c}{F_s}\right)\left(k - \frac{N}{2}\right)\right) \quad k = 0, 1, 2, \dots, N$$

$$h(k) = 2 \cdot \left(\frac{20}{44,1}\right) \cdot \operatorname{sinc}\left(2 \cdot \left(\frac{20}{44,1}\right) (k - 5)\right) \quad k = 0, 1, 2, \dots, 10$$

კოეფიციენტები შეიძლება მოიძებნოს უფრო მარტივად MATLAB-ის გამოყენებით:

`k = 0:1:10; % ვექტორი 0-დან 10-მდე`

`h = 2 * (20 / 44.1) * sinc(2 * (20 / 44.1) * (k - 5)); % ფილტრის კოეფიციენტების`

გამოთვლა

`disp(h);`

h= 0.0633 -0.0732 0.0815 -0.0878 0.0917 0.9070 0.0917 -0.0878 0.0815 -0.0732
0.0633

ბ) სიმ ფილტრის კოეფიციენტები არის შემავალ/გამომავალ სიგნალთა სხვაობის განტოლების რიცხვითი კოეფიციენტები:

$$y(k) = 0,0633x(k) - 0,0732x(k-1) + 0,0815x(k-2) - 0,0878x(k-3) + 0,0917x(k-4) + 0,9070x(k-5) + 0,0917x(k-6) - 0,0878x(k-7) + 0,0815x(k-8) - 0,0732x(k-9) + 0,0633x(k-10)$$

ფილტრის გამომავალი სიგნალი შეიძლება გამოითვალოს რეკურსიულად მიმდინარე შემავალი და ბოლო ათი, მანამდე შემავალი ანთვლების გამოყენებით, თვლადი რაოდენობის ოპერაციით (თერთმეტი გამრავლება და ათი მიმატება).

გ) ფილტრის გადაცემის ფუნქცია მიიღება შემავალ-გამომავალ სიგნალთა განტოლების Z-გარდაქმნის საშუალებით და შედეგის გვაწვდის შემდეგი ფორმით

$$H(z) = [0.0633z^{10} - 0.0732z^9 + 0.0815z^8 - 0.0878z^7 + 0.0917z^6 + 0.9070z^5 + 0.0917z^4 - 0.0878z^3 + 0.0815z^2 - 0.0732z + 0.0633]/z^{10}$$

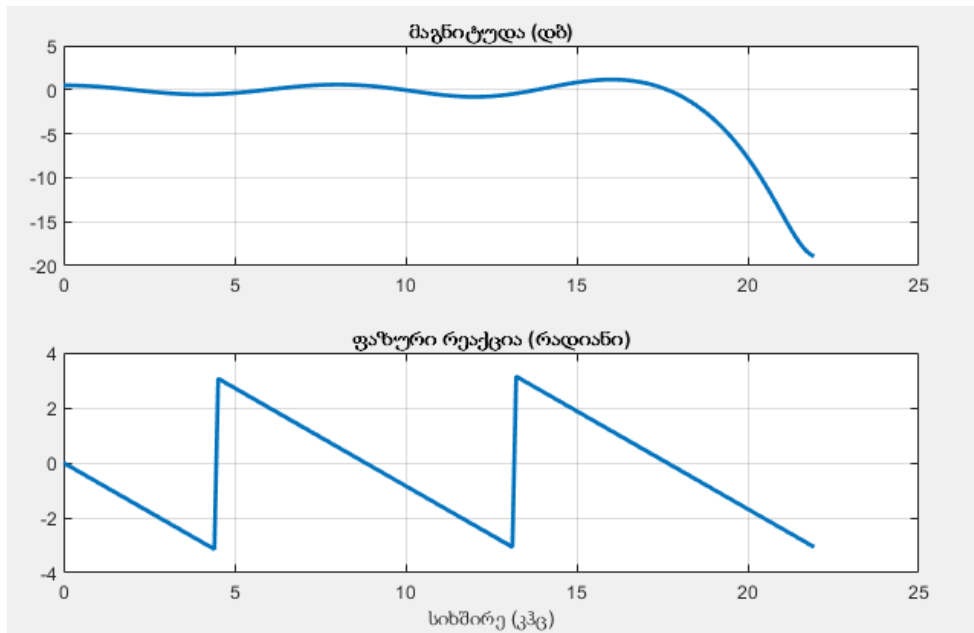
დ) სიხშირის რეაქცია მარტივად შეიძლება განხიროციელდეს MATLAB-ში `freqz` ბრძანების გამოყენებით:

```

num_tf = h;
den_tf = zeros(size(h)); den_tf(1) = 1;
% კმნის den_tf = [1 0 ... 0], შესაბამისი ზომის ვექტორით
Fs = 44100;
[H,f] = freqz(num_tf,den_tf,200,Fs);
% ითვლის სიხშირის რეაქციას
subplot(2,1,1); plot(f/1000, 20*log10(abs(H)), 'lineWidth',2);
title('მაგნიტუდა (დბ)'); grid
subplot(2,1,2); plot(f/1000, angle(H), 'lineWidth',2);
title('ფაზური რეაქცია (რადიანი)'); grid
xlabel('სიხშირე (კჰც)');

```

მე-10 რიგის სიმ ფილტრის მიღებული სიხშირული მახასიათებლები ნაჩვენებია სურათზე. როგორ შევადარებთ ამ მახასიათებლებს იდეალურ დაბალგამტარულ ფილტრს, რომლის მოკვეთის სიხშირეა 20 კჰც? იდეალური დაბალგამტარები ფილტრი უნდა ჰქონდეს ბრტყელი მაგნიტუდის რეაქცია მთელ გამტარ უბანზე; ანუ, მაგნიტუდა უნდა იყოს 0 დბ (ან 1) 0-დან 20 კჰც-მდე და შემდეგ მყისიერად დაეცეს $-\infty$ დბ (ან 0)-მდე ყველა სიხშირისთვის, რომელიც აღემატება 20 კჰც-ს. მე-10 რიგის სიმ ფილტრს აქვს რხევები გამტარ უბანზე და მაგნიტუდა იწყებს თანდათანობით დაკლებას დაახლოებით 17.5 კჰც-დან. 3 დბ-ის მოკვეთის სიხშირე დაახლოებით 18 კჰც-ია. მე-10 რიგის სიმ ფილტრის ფაზური რეაქცია შეესაბამება იდეალურ რეაქციას – ის არის ხაზოვანი. სიმ ფილტრის კიდევ ერთი სასარგებლო მახასიათებელია, სტაბილურობასთან ერთად, მისი ხაზოვანი (იდეალური) ფაზური რეაქცია.



სურათი 36. სიმ ფილტრის სიხშირული მახასიათებლები (წყარო-ავტორი) შედეგები:

- გამოვთვალეთ მე-10 რიგის სიმ დაბალსიხშიური ფილტრის კოეფიციენტები MATLAB-ის გამოყენებით, მიღებული მნიშვნელობები: $h = 0.0633 \quad -0.0732 \quad 0.0815 \quad -0.0878 \quad 0.0917 \quad 0.9070 \quad 0.0917 \quad -0.0878 \quad 0.0815 \quad -0.0732 \quad 0.0633$,
- სიხშირის რეაქციაზე შეინიშნება რხევები გამტარ უბანში.
- მაგნიტუდის შემცირება იწყება დაახლოებით 17.5 კჰც-დან.
- 3 დბ-იანი მოჭრის სიხშირე: დაახლოებით 18 კჰც.
- ფაზური რეაქცია: ხაზოვანია, რაც სასარგებლოა სიგნალის დამუშავებისთვის.
- იდეალური დაბალსიხშიური ფილტრისგან განსხვავებით, ამ ფილტრს აქვს გარდამავალი ზონა, სადაც მაგნიტუდა თანდათანობით მცირდება, რაც ფილტრის შეზღუდული რიგის გამო ხდება და მკვეთრი მოჭრის შესაძლებლობას ზღუდავს;
- ფილტრი სტაბილურია და გამოირჩევა ხაზოვანი ფაზური რეაქციით.

შედეგების ანალიზი და დასკვნა: მიღებული მე-10 რიგის სიმ ფილტრი ასრულებს დაბალსიხშიური ფილტრის ფუნქციას, თუმცა იდეალურ ფილტრთან შედარებით მას აქვს მცირე ხარვეზები: რხევები გამტარ უბანში და მაგნიტუდის თანდათანობითი შემცირება. ფილტრის ხაზოვანი ფაზური რეაქცია და სტაბილურობა სასარგებლო მახასიათებლებია. უფრო მკვეთრი მოჭრის სიხშირის მისაღწევად შეიძლება გამოვიყენოთ მაღალი რიგის ფილტრი ან სპეციალური ფანჯრის ფუნქციები.

რიცხვითი ექსპერიმენტი 2

ავაგოთ სიხშირული მახასიათებლების დიაგრამა (მხოლოდ მაგნიტუდა) 30-ე რიგის და 50-ე რიგის სიმ ფილტრებისთვის, სადაც დისკრეტიზაციის სიხშირე არის 44.1 კჰც და მოკვეთის სიხშირე 20 კჰც.

შემდეგი MATLAB ბრძანებები გამოთვლის ფილტრის კოეფიციენტებს, შესაბამის სიხშირის რეაქციას და შემდეგ ააგებს მაგნიტუდის რეაქციას 30-ე და 50-ე რიგის ფილტრებისთვის.

```
k = 0:1:30;
h_30 = 2 * (20 / 44.1) * sinc(2 * (20 / 44.1) * (k - 15)); % 30-ე რიგის ფილტრის
კოეფიციენტები

k = 0:1:50;
h_50 = 2 * (20 / 44.1) * sinc(2 * (20 / 44.1) * (k - 25)); % 50-ე რიგის ფილტრის
კოეფიციენტები

Fs = 44100; % ნიმუშების აღების სიხშირე

num_30 = h_30;
den_30 = zeros(size(h_30)); den_30(1) = 1; % 30-ე რიგის ფილტრის კოეფიციენტები

num_50 = h_50;
den_50 = zeros(size(h_50)); den_50(1) = 1; % 50-ე რიგის ფილტრის კოეფიციენტები

[H_30, f_30] = freqz(num_30, den_30, 200, Fs); % 30-ე რიგის ფილტრის სიხშირის
რეაქცია
[H_50, f_50] = freqz(num_50, den_50, 200, Fs); % 50-ე რიგის ფილტრის სიხშირის
რეაქცია

subplot(2,1,1);
plot(f_30/1000, 20 * log10(abs(H_30)), 'lineWidth', 2); % 30-ე რიგის ფილტრის
მაგნიტუდის რეაქცია
title('30-ე რიგის FIR ფილტრის მაგნიტუდის რეაქცია (დბ)');
grid on;
```

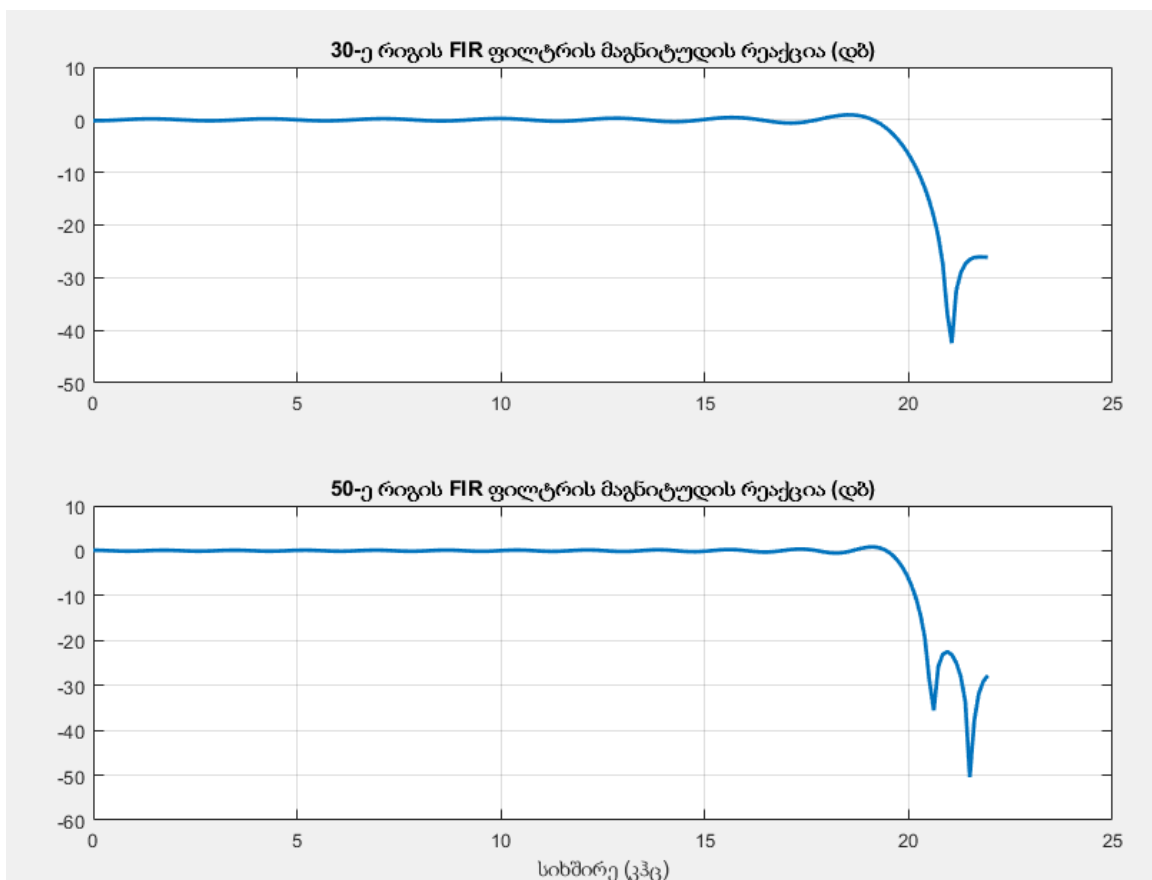
```

subplot(2,1,2);
plot(f_50/1000, 20 * log10(abs(H_50)), 'lineWidth', 2); % 50-ე რიგის ფილტრის
მაგნიტუდის რეაქცია
title('50-ე რიგის FIR ფილტრის მაგნიტუდის რეაქცია (დბ)');
grid on;

xlabel('სიხშირე (კჰც)');

```

სურათზე ნაჩვენებია ორი ფილტრის მაგნიტუდის რეაქციები. დავაკვირდეთ, რომ უფრო მაღალი რიგის ფილტრი აქვს უფრო ბრტყელი გამტარ ზოლი და უფრო ვიწრო გადასვლის უბანი (ანუ უფრო ახლოს იდეალურთან), ვიდრე დაბალი რიგის ფილტრს.



სურათი 37. ორი ფილტრის მაგნიტუდის რეაქციები (წყარო-ავტორი).

შედეგების ანალიზი და დასკვნა:

ფილტრის რიგის გავლენა: როგორც სურათებზე ჩანს, 50-ე რიგის უიმ ფილტრს აქვს უფრო ბრტყელი გამტარ ზოლი და უფრო ვიწრო გადასვლის უბანი, ვიდრე 30-ე რიგის ფილტრს. ეს მიუთითებს ფილტრის რეაქციის გაუმჯობესებას, რაც უფრო აახლოებს მას იდეალურ დაბალგამტარულ ფილტრთან. თუმცა, ფილტრის რიგის ზრდა ასახავს

უფრო რთულ სიხშირის რეაქციას, რამაც შესაძლოა გამოიწვიოს შემდგომი სტაბილურობის პრობლემა.

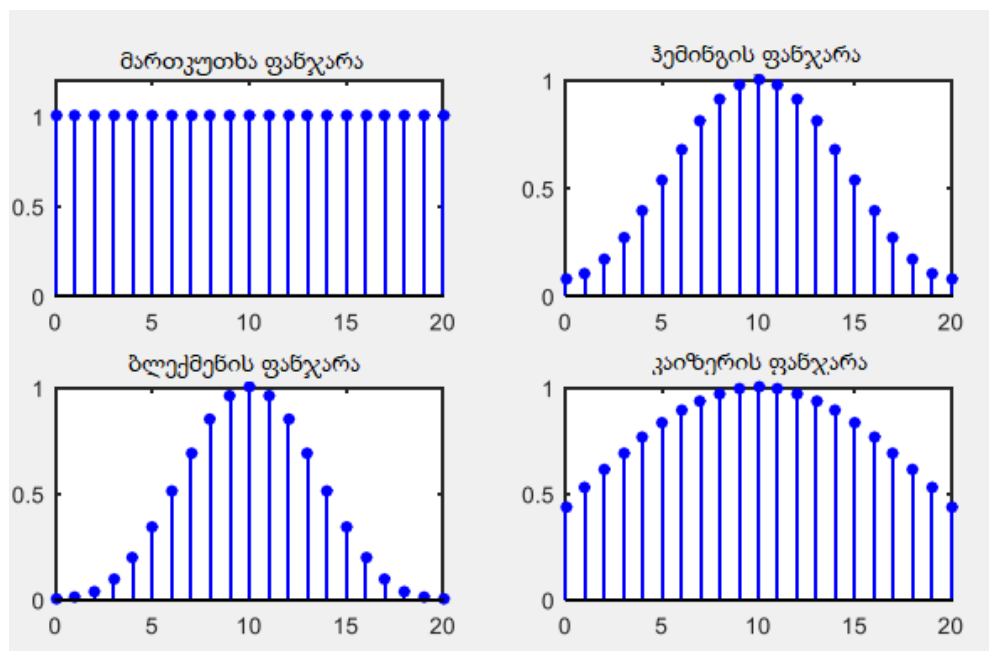
ფილტრის კოეფიციენტები და შეზღუდვები:

თუ უფრო მაღალი რიგის ფილტრი უკეთეს შედეგებს იძლევა, ფილტრის რიგი მაინც შეზღუდულია და რჩება რხევები, როგორც გამტარ უბანში, ასევე დამხშობ უბანში. ეს რხევები გამოწვეულია ფილტრის კოეფიციენტების უცაბედი წყვეტით.

ფანჯრის ფუნქციის როლი:

ამ რხევების შემცირებისათვის, შესაძლებელია ფანჯრის ფუნქციის გამოყენება, რაც ფილტრის კოეფიციენტებს თანდათანობით აქრობს კიდეებზე, რაც უფრო გლუვ ეფექტს იძლევა. ფანჯრის ფუნქციები განსხვავებულ როლს ასრულებენ და სხვადასხვა სტაბილურობისა და მასშტაბის ფილტრების მიღწევას უზრუნველყოფენ.

დასკვნა: სიმ ფილტრის რიგის გაზრდა აუმჯობესებს ფილტრის სიხშირულ მახასიათებლებს, მაგრამ მაინც რჩება რხევები როგორც გამტარ უბანში, ისე დამხშობ უბანში. როგორც ზემოთ არნიშნეთ, ეს რხევები გამოწვეულია ფილტრის კოეფიციენტების უცაბედი წყვეტით და ამ დროს შეიძლება გამოვიყენოთ ფანჯრის ფუნქცია. ფანჯრის ფუნქცია ფილტრის კოეფიციენტებს თანდათანობით აქრობს კიდეებზე, რაც უფრო გლუვ ეფექტს იძლევა. სურათზე ვხედავთ სხვადასხვა ფანჯრის ფუნქციებს 20-ე რიგის სიმ ფილტრისთვის.



სურათი 38. სხვადასხვა ფანჯრის ფუნქციის მაგალითები (წყარო-ავტორი)

ჰემინგის ფანჯრის ფუნქცია აღიწერება შემდეგი ფორმულით:

$$w(k) = 0,54 - 0,46 \cos\left(\frac{2\pi k}{N}\right) \quad k = 0, \dots, N \quad (\text{Oppenheim 2009, 13-15})$$

ბლექმენის ფანჯრის ფუნქცია აღიწერება შემდეგი ფორმულით:

$$w(k) = 0,42 - 0,5 \cos\left(\frac{2\pi k}{N}\right) + 0,08 \cos\left(\frac{4\pi k}{N}\right) \quad k = 0, \dots, N \quad (\text{Oppenheim 2009, 56-57})$$

კაიზერის ფანჯრის ფუნქცია აღიწერება შემდეგი ფორმულით:

$$w(k) = \frac{I_0\left(\pi\beta\sqrt{1 - \left(\frac{2k}{N}\right)^2}\right)}{I_0(\pi\beta)} \quad k = 0, \dots, N$$

სადაც $I_0()$ არის ბესელის ფუნქცია

რიცხვითი ექსპერიმენტი 3:

შევქმნათ 50-ე რიგის დაბალსიხშირული სიმ ფილტრი ჰემინგის ფანჯრის, ბლექმენის ფანჯრის და კაიზერის ფანჯრის გამოყენებით. ნიმუშების აღების სიხშირე არის 44.1 კჰც და მოკვეთის სიხშირე 20 კჰც. ავაგოთ სიხშირული მახასიათებლის (მხოლოდ მაგნიტუდა) თითოეული ფილტრისთვის და შევადაროთ თითოეული ფილტრი იმ 50-ე რიგის სიმ ფილტრთან, რომელიც შექმნილი იყო ზედა სურათში, უცაბედი წყვეტით.

% ფილტრის კოეფიციენტების გამოთვლა:

`k = 0:1:50;`

`h = 2 * (20 / 44.1) * sinc(2 * (20 / 44.1) * (k - 25));`

% კოეფიციენტების გამრავლება თითოეული ფანჯრის ფუნქციით:

`h_hamming = h .* window(@hamming, 51)';` % ჰემინგის ფანჯრის კოეფიციენტები

`h_blackman = h .* window(@blackman, 51)';` % ბლექმენის ფანჯრის კოეფიციენტები

`h_kaiser = h .* window(@kaiser, 51, 3)';` % კაიზერის ფანჯრის კოეფიციენტები

% Hamming-ის ფილტრისთვის გადაცემის ფუნქცია:

`num_hamm = h_hamming;`

`den_hamm = zeros(size(h_hamming));`

`den_hamm(1) = 1;`

```

% blackman-ის ფილტრისთვის გადაცემის ფუნქცია:
num_black = h_blackman;
den_black = zeros(size(h_blackman));
den_black(1) = 1;

% Kaiser-ის ფილტრისთვის გადაცემის ფუნქცია:
num_kais = h_kaiser;
den_kais = zeros(size(h_kaiser));
den_kais(1) = 1;

% თითოეული FIR ფილტრის სიხშირის რეაქციის გამოთვლა და დახაზვა:
Fs = 44100;

[H_hamm, f] = freqz(num_hamm, den_hamm, 200, Fs);
subplot(3,1,1);
plot(f / 1000, 20 * log10(abs(H_hamm)), 'lineWidth', 2); % ჰემინგის ფანჯრის
მაგნიტუდის რეაქცია
title('ჰემინგის ფანჯრის მაგნიტუდა (დბ)');
grid on;

[H_black, f] = freqz(num_black, den_black, 200, Fs);
subplot(3,1,2);
plot(f / 1000, 20 * log10(abs(H_black)), 'lineWidth', 2); % ბლეკმენის ფანჯრის
მაგნიტუდის რეაქცია
title('ბლეკმენის ფანჯრის მაგნიტუდა (დბ)');
grid on;

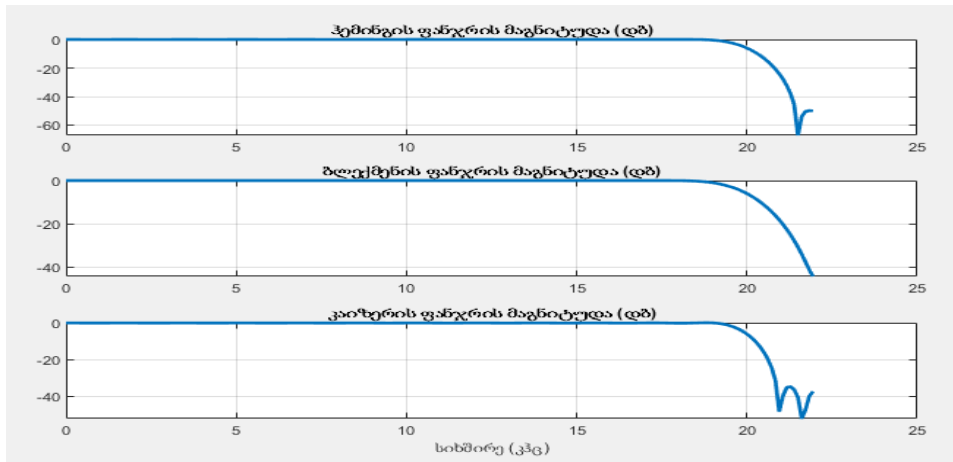
[H_kais, f] = freqz(num_kais, den_kais, 200, Fs);
subplot(3,1,3);
plot(f / 1000, 20 * log10(abs(H_kais)), 'lineWidth', 2); % კაიზერის ფანჯრის მაგნიტუდის
რეაქცია
title('კაიზერის ფანჯრის მაგნიტუდა (დბ)');
grid on;

xlabel('სიხშირე (კჰც)');

```

შედეგების ანალიზი: სიხშირული რეაქცია თითოეული ფანჯრის ფილტრისთვის ნაჩვენებია სურათზე, თითოეული ფანჯრის ფუნქცია ამცირებს რხევების რაოდენობას გამტარ უბანსა და მოკვეთის უბანში. რაც განსხვავდება 50-ე რიგის სიმ ფილტრისგან, რომლის კოეფიციენტების უცბედი წყვეტით ხასიათდება.

ჰემინგის და ბლექმენის ფანჯრები აჩვენებენ ძალიან მცირე რხევას, მაგრამ აქვთ უფრო ფართო გარდამავალი უბანი გამტარი უბნიდან დახშობის უბნამდე. კეიზერის ფანჯარა ქმნის ფილტრს, რომელსაც აქვს უფრო მკვეთრი გარდამავალი უბანი, თუმცა ამ ფილტრის გამტარ უბანსა და მოკვეთის უბანსი მაინც დარჩება მცირე რხევები.



სურათი 39. გრაფიკი წარმოადგენს იდეალური დაბალგამტარი ფილტრის (Ideal Low-Pass Filter) სიხშირული მახასიათებლების ორ კომპონენტს: ამპლიტუდის სპექტრს და ფაზურ სპექტრს (წყარო-ავტორი)

ზედა სურათზე სადაც სხვადასხვა ფანჯრის ფუნქციები ჩანს, ფილტრების კოეფიციენტების ნახევარი ნაჩვენებია ქვემოთ ცხრილში. სიმეტრიის გამო ცხრილში საკმარისია მხოლოდ კოეფიციენტების ნახევრის ჩვენება. უნდა აღინიშნოს, რომ კოეფიციენტები ძალიან პატარაა და ხშირად არ განსხვავდება სხვადასხვა ფანჯრის ფუნქციის კოეფიციენტებისგან, ამავდროულად სიხშირის რეაქციები ძალიან განსხვავებულია. პატარა ცვლილებები ფილტრების კოეფიციენტებში შეიძლება გამოიწვიოს დიდი ცვლილებები ფილტრის მუშაობაში – ეს არის ძალიან მნიშვნელოვანი ფაქტი!

არსებობს ბევრი განსხვავებული ფანჯრის ფუნქცია, რომლებიც შეიძლება გამოვიყენოთ. MATLAB-ს აქვს თხუთმეტი სხვადასხვა ფანჯრის ფუნქცია. გარდა ამისა, Filter Design and Analysis Tool (FDAT) გვაზლევს საშუალებას მომხმარებელმა შექმნას საკუთარი ფანჯრის ფუნქცია. ყველა ფანჯრის ფუნქცია ცდილობს აღმოფხვრას პრობლემა, რომელიც დაკავშირებულია გამტარ უბანიდან მოკვეთის უბანში მკვეთრი გადასვლით და ორივე ზოლში რხევის პრობლემასთან.

დასკვნა: შესრულებული ექსპერიმენტის შედეგად შევქმნათ 50-ე რიგის დაბალსიხშირული ფილტრები ჰემინგის, ბლექმენის და კეიზერის ფანჯრის

ფუნქციების გამოყენებით. ყველა ფილტრისთვის სიხშირის რეაქცია გამოხატულია და მათ შორის შედარება მოვახდინეთ 50-ე რიგის ფილტრთან.

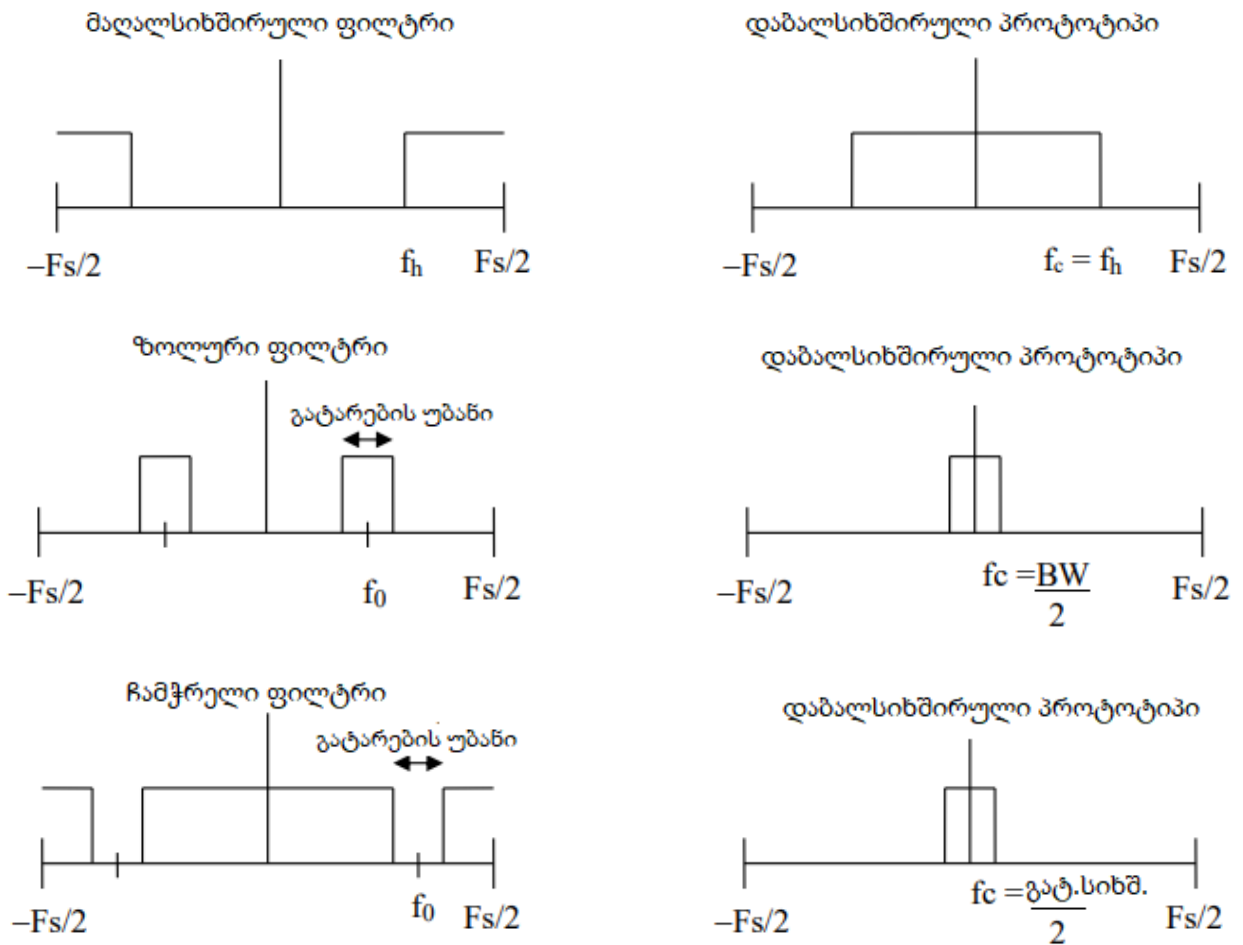
ჰემინგის და ბლექმენის ფანჯრები აჩვენებენ საკმაოდ პატარა რხევას, მაგრამ მათ აქვთ შედარებით უფრო ფართო გარდამავალი უბანი გამტარ უბნიდან მოკვეთის უბანამდე. ამავე დროს, კაიზერის ფანჯარა ქმნის ფილტრს, რომელსაც უფრო მკვეთრი გარდამავალი უბანი აქვს, თუმცა მცირე რხევები რჩება გამტარ და მოკვეთის უბნებში.

აღსანიშნავია, რომ ფილტრების კოეფიციენტების მცირე ცვლილებებმა შესაძლოა გამოიწვიოს მნიშვნელოვანი ცვლილებები ფილტრის ფუნქციონირებისას, რაც ხაზს უსვამს კოეფიციენტების და მათი სიხშირის რეაქციის მნიშვნელობას. ექსპერიმენტმა დაადასტურა, რომ სხვადასხვა ფანჯრის ფუნქციები, მიუხედავად იმისა, რომ ისინი მიზნად ისახავენ გამტარ და მოკვეთის უბნებში რხევის პრობლემის აღმოფხვრას, ქმნიან სხვადასხვა ეფექტებს, რაც ძალზედ მნიშვნელოვანია ფილტრების ეფექტურობის და შედეგების დასაზუსტებლად, ამიტომ ამ ექსპერიმენტში მიღებული ცოდნა შეიძლება გამოყენებულ იქნას სხვადასხვა ტესტირების და ტექნიკური მოწყობილობების გაძლიერებისთვის.

4.3. მაღალსიხშირული, ზოლური და ჩამჭრელი სიმ

ფილტრების გენერირება, ექსპერიმენტი და შედეგები, შედეგების ანალიზი

წინა ექსპერიმენტების ჩვენ რამდენიმე დაბალსიხშირული სიმ ფილტრი შევქმენით. მაღალი გამტარობის, ზოლური და ჩამჭრელი ფილტრები მარტივად შეიძლება შეიქმნას იმ გზით, რომ ზემოთ დაპროექტებული ფილტრის მახასიათებლები გადავიყვანოთ "ეკვივალენტურ" დაბალგამტარ ფილტრში და შემდეგ მისი კოეფიციენტები გადავაქციოთ სასურველ ფილტრის ტიპად. სურათზე ნაჩვენებია, როგორ ხდება მაღალსიხშირული, ზოლური და ჩამჭრელი ფილტრების გადაყვანა "ეკვივალენტურ" დაბალსიხშირულ ფილტრებში.



სურათი 40. სურათზე ნაჩვენებია, როგორ ხდება მადალსიხშირული, ზოლური და ჩამჭრელი ფილტრების გადაყვანა "ეკვივალენტურ" დაბალსიხშირულ ფილტრებში (Heineken 2017, 78)

დაბალსიხშირული ფილტრის კოეფიციენტები (სადაც გამოყენებულია ფურიეს მწკრივები) ითვლება შემდეგნაირად:

$$h_{\text{დაბალსიხშირული}}(k) = 2 \left(\frac{f_c}{F_s} \right) \text{sinc} \left(2 \left(\frac{f_c}{F_s} \right) \left(k - \frac{N}{2} \right) \right) \quad k = 0, 1, 2, \dots, N$$

ზემოთ სურათზე მოცემული სამი ფილტრის კონსტრუირებისთვის კი საჭიროა შემდეგი გაშვები :

$$h_{\text{მადალსიხშირული}}(k) = \begin{cases} 1 - h_{\text{დაბალსიხშირული}}(k) & k = \frac{N}{2} \quad (\text{N უნდა იყოს ლუწი}) \\ -h_{\text{დაბალსიხშირული}}(k) & k \neq \frac{N}{2} \end{cases}$$

$$h_{\text{ზოლური}}(k) = 2 \cos \left(2\pi \left(\frac{f_0}{F_s} \right) \left(k - \frac{N}{2} \right) \right) \cdot \left(\frac{BW}{F_s} \right) \cdot \text{sinc} \left(\left(\frac{BW}{F_s} \right) \left(k - \frac{N}{2} \right) \right) \quad k = 0, 1, 2, \dots, N$$

$$h_{\text{ჩამჭრელი}}(k) = \begin{cases} 1 - h_{\text{ზოლური}}(k) = 1 - 2 \left(\frac{DW}{F_s} \right) & k = \frac{N}{2} \text{ (N ლუწია)} \\ -h_{\text{ზოლური}}(k) = -2 \cos \left(2\pi \left(\frac{f_0}{F_s} \right) \left(k - \frac{N}{2} \right) \right) \cdot \left(\frac{BW}{F_s} \right) \cdot \text{sinc} \left(\left(\frac{BW}{F_s} \right) \left(k - \frac{N}{2} \right) \right) & k \neq \frac{N}{2} \end{cases}$$

ფანჯარების მეთოდის დამატება ფილტრის დიზაინში მარტივად შეიძლება, ფილტრის კოეფიციენტების გაყოფით სასურველ ფანჯრის ფუნქციაზე, ზუსტად ისე, როგორც ექსპერიმენტ 3-ში დაბალგამტარი ფილტრის კოეფიციენტები დამუშავდა ფანჯრის მეთოდით.

რიცხვითი ექსპერიმენტი 4:

ავაგოთ მაღალსიხშირული, ზოლური და ჩამჭრელი ფილტრები 48 კჰც დისკრეტიზაციის სიხშირით.

ა) 50-ე რიგის მაღალსიხშირული ფილტრი 16 კჰც მოჭრის სიხშირით.

ბ) 50-ე რიგის ჩამჭრელი ფილტრი, რომელის ცენტრალური სიხშირე, 10 კჰც და ზოლური სიგანე 4 კჰც

გ) 50-ე რიგის ზოლური ფილტრი, რომელის ცენტრალური სიხშირე, 10 კჰც და ზოლური სიგანე 4 კჰც

ა) 50-ე რიგის მაღალსიხშირული ფილტრი 16 კჰც მოჭრის სიხშირით. ასეთი ფილტრის კოეფიციენტების გამოთვლა შესაძლებელია ფორმულით:

$$h_{\text{მაღალსიხშირული}}(k) = -1 \cdot h_{\text{დაბალსიხშირული}}(k) \text{ სადაც } k \neq \frac{N}{2}$$

$$h_{\text{მაღალსიხშირული}}(k) = 1 - h_{\text{დაბალსიხშირული}}(k) = 1 - 2f_h/F_s \text{ for } k = \frac{N}{2}$$

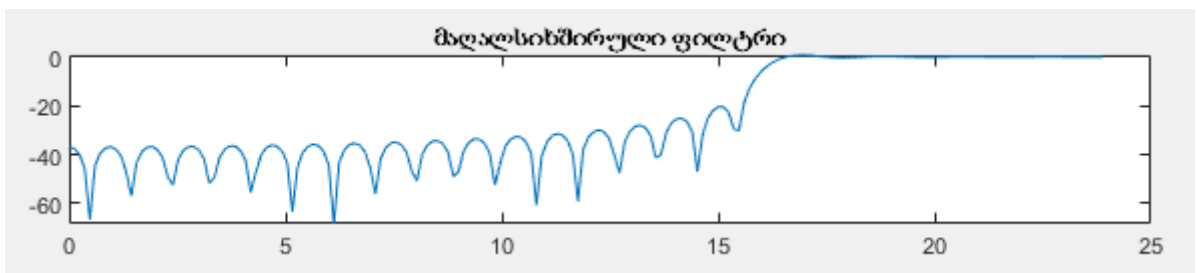
MATLAB:

```

Fs = 48000; * დისკრეტიზაციის სიხშირე
fh=16000; * მოჭრის სიხშირე
k=0:1:50;
N = 50;
h_HP = -2*(fh/Fs)*sinc(2*(fh/Fs)*(k - N/2));
* დავაფიქსიროთ, რომ k = N/2
* ინდექსაციას – ვიხილოთ ქვემოთ შენიშვნა!
h_HP(26)= 1 - 2*fh/Fs;
num_HP = h_HP;
den_HP = zeros(size(h_HP)); den_HP(1)=1;
[H_HP,f_HP]=freqz(num_HP,den_HP,200,Fs);
subplot(3,1,1);plot(f_HP/1000,20*log10(abs(H_HP)));
title('მაღალსიხშირული ფილტრი')

```

შენიშვნა: ფორმულის მიხედვით, $h_{HP}\left(\frac{N}{2}\right) = h_{HP}(25) = 1 - 2 * \frac{f_h}{F_s}$ მაშ, რატომ ვიყენებთ MATLAB კოდში ბრძანებას $h_HP(26)= 1 - 2*fh/Fs$; მიზეზი ის არის, რომ ჩვენ ვიწყებთ ფილტრის კოეფიციენტების ინდექსაციას ნულიდან, ანუ $h_{HP}(0), h_{HP}(1), \dots, h_{HP}(50)$ ვიცით, რომ MATLAB-ში ვექტორების ინდექსაცია იწყება 1-დან. ამიტომ, თუ თეორიულად 25-ე კოეფიციენტს ვთვლით, MATLAB-ში ეს იქნება 26-ე პოზიცია. ამიტომ გამოიყენება $h_HP(26)$.



სურათი 41. მაღალსიხშირული ფილტრი (წყარო-ავტორი)

ბ) 50-ე რიგის ჩამჭრელი ფილტრი, რომელის ცენტრალური სიხშირე, 10 კჰც და ზოლური სიგანე 4 კჰც, ასეთი ფილტრის კოეფიციენტები დაითვლება შემდეგნაირად:

$$h_{\text{ზოლური}}(k) = 2 \cos\left(2\pi \left(\frac{f_0}{F_s}\right) \left(k - \frac{N}{2}\right)\right) \cdot \left(\frac{BW}{F_s}\right) \cdot \text{sinc}\left(\left(\frac{BW}{F_s}\right) \left(k - \frac{N}{2}\right)\right)$$

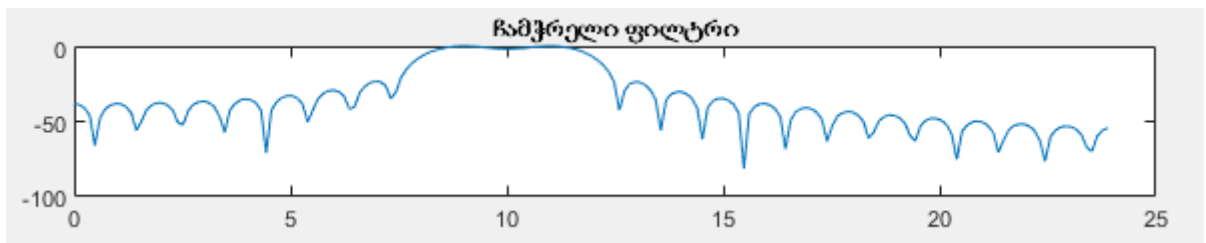
$$k = 0, 1, 2, \dots,$$

MATLAB:

```

Fs=48000; * დისკრეტიზაციის სიხშირე
fo=10000; * ცენტრული სიხშირე
bW=4000; * ზოლური სიგანე
k=0:1:50;
N=50;
h_bP = 2*cos(2*pi*(fo/Fs)*(k - N/2))*(bW/Fs).*sinc((bW/Fs)*(k-N/2));
num_bP = h_bP;
den_bP = zeros(size(h_bP)); den_bP(1)=1;
[H_bP,f_bP]=freqz(num_bP,den_bP,200,Fs);
subplot(3,1,2);plot(f_bP/1000,20*log10(abs(H_bP)));
title('ჩამჭრელი ფილტრი')

```



სურათი 42. 50-ე რიგის ჩამჭრელი ფილტრი, რომელის ცენტრალური სიხშირე, 10 კჰც და ზოლური სიგანე 4 კჰც (წყარო-ავტორი)

გ) 50-ე რიგის ზოლური ფილტრი, რომელის ცენტრალური სიხშირე, 10 კჰც და ზოლური სიგანე 4 კჰც, ასეთი ფილტრის კოეფიციენტები დაითვლება ფორმულით:

$$h_{\text{ჩამჭრელი}}(k) = -1 \cdot h_{bP}(k) \text{ სადაც } k \neq \frac{N}{2}$$

$$h_{\text{ჩამჭრელი}}(k) = 1 - h_{bP}(k) = 1 - 2BW/F_s \text{ სადაც } k = \frac{N}{2}$$

MATLAB:

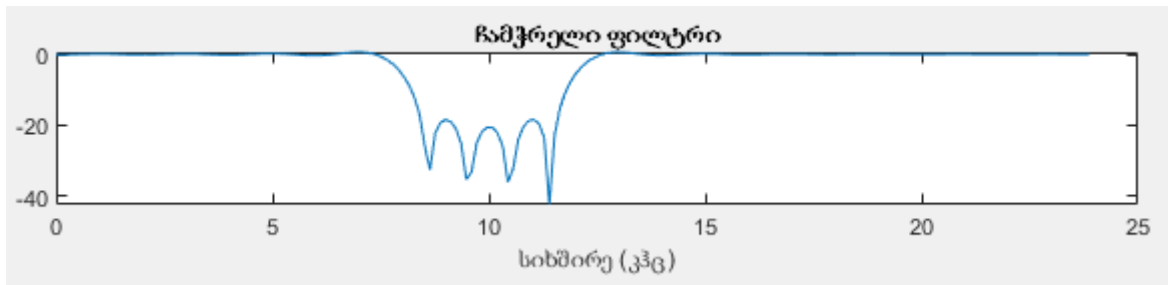
```

k=0:1:50;
h_N = (-1)* h_bP;
* დავაფიქსიროთ, რომ k = N/2

```

* იგივე პრობლემა, რაც მაღალსიხშირულ ფილტრში - აქაც ყურადღებით ვიყოთ ინდექსაციისას!

```
h_N(N/2 + 1) = 1 - 2*bW/Fs;
num_N = h_N;
den_N = zeros(size(h_N)); den_N(1)=1;
[H_N,f_N]=freqz(num_N,den_N,200,Fs);
subplot(3,1,3);plot(f_N/1000,20*log10(abs(H_N)));
title('ჩამჭრელი ფილტრი')
xlabel('სიხშირე (კჰც)')
```



სურათი 43. გ) 50-ე რიგის ზოლური ფილტრი, რომელის ცენტრალური სიხშირე, 10 კჰც და ზოლური სიგანე 4 კჰც (წყარო-ავტორი)

ა) ბ) გ) მაგალითებში განხილულია მაღალსიხშირული, ზოლური და ჩამჭრელი ფილტრების კონსტრუირება სადაც დისკრეტიზაციის სიხშირე არის 48 კჰც .

შედეგების ანალიზი და დასკვნა: ექსპერიმენტი მიზნად ისახავდა მაღალსიხშირული, ზოლური და ჩამჭრელი ფილტრების კონსტრუირებას, სადაც დისკრეტიზაციის სიხშირე იყო 48 კჰც.

მაღალსიხშირული ფილტრი (16 კჰც მოჭრის სიხშირე): ამ ფილტრის კოეფიციენტები გამოთვლილია სპეციალური ფორმულის მიხედვით, სადაც ფილტრის ზედა სიხშირე არის 16 კჰც. MATLAB კოდში გამოიყენება sinc ფუნქცია და სხვა შესაბამისი გამოთვლები, რათა ფილტრის კოეფიციენტები სწორად განისაზღვროს.

ჩამჭრელი ფილტრი (ცენტრალური სიხშირე 10 კჰც და ზოლური სიგანე 4 კჰც): ამ ფილტრის კოეფიციენტები გამოთვლილია რეკურსიული ფორმულით, სადაც გამოყენებულია cos და sinc ფუნქციები, რათა ზუსტი სიგნალი მიღებულიყო 10 კჰც ცენტრალური სიხშირით და 4 კჰც ზოლური სიგანით. ფილტრის სიხშირის რეაქცია წარმოდგენილია სურათზე, რაც ასახავს მისი ეფექტურობას სიხშირეების დიაპაზონში.

ზოლური ფილტრი (ცენტრალური სიხშირე 10 კჰც და ზოლური სიგანე 4 კჰც): ეს ფილტრი მუშაობს ზუსტად იმავე პრინციპზე, როგორც ჩამჭრელი, მაგრამ აქ ფილტრის კოეფიციენტები პასიურია. სწორედ ამიტომ, მისი მოჭრის სიხშირე განსხვავებულია.

ფილტრების კოეფიციენტების გამოთვლა და სიხშირის დიაგრამების მიღება ნათლად ასახავს სხვადასხვა ტიპის ფილტრების მუშაობას. MATLAB-ის გამოყენებით სწორად განისაზღვრა ფილტრის მახასიათებლები, რაც რიცხვითი ექსპერიმენტების დროს გადამწყვეტია.

V თავი - ციფრული ფილტრები უსასრულო იმპულსური მახასიათებლებით

ამ თავში განხილული გვაქვს ციფრული ფილტრები უსასრულო იმპულსური მახასიათებლებით (უიმ). შექმნილი გვაქვს რამდენიმე უიმ ფილტრი. კონსტრუირებულ ფილტრებს ვადარებთ IV თავში განხილულ სიმ ფილტრებს.

5.1. ფილტრები უსასრულო იმპულსური მახასიათებლებით (უიმ)

უიმ (უსასრულო იმპულსური მახასიათებლის) ციფრული ფილტრები შეიძლება აღვწეროთ შემავალ/გამომავალ სიგნალთა სხვაობის განტოლებით (Ingle 2015, 126):

$$y(k) = \sum_{i=0}^N b_i x(k-i) - \sum_{i=1}^N a_i y(k-i) \quad (\text{Ingle 2015, 126):}$$

სადაც:

- b_0, b_1, \dots, b_N და a_1, \dots, a_N არის ფილტრის კოეფიციენტები.
- N არის ფილტრის რიგი. ფილტრის კოეფიციენტების რაოდენობა არის $2N + 1$.

- უიმ ფილტრის დაპროექტება მოითხოვს ფილტრის რიგის და ფილტრის კოეფიციენტების ნაკრების მოძებნას, რაც დააკმაყოფილებს მოცემულ ფილტრის მოთხოვნებს.
- ფილტრის გამომავალი მნიშვნელობები გამოითვლება შემდეგნაირად: მიმდინარე შემავალი და წარსული შემავალი და გამომავალი მნიშვნელობები მრავლდება ფილტრის კოეფიციენტებზე. მიღებული შედეგები იკრიბება
- ფილტრის განხორციელებისთვის, ფილტრის კოეფიციენტები, მიმდინარე შემავალი, გარკვეული რაოდენობის წარსული შემავალი და გარკვეული რაოდენობის წარსული გამომავალი მნიშვნელობები უნდა ინახებოდეს მეხსიერებაში.
- უიმ ფილტრი არის რეკურსიული, ანუ ფილტრის გამომავალი მნიშვნელობა დამოკიდებულია წარსულ გამომავალ და შემავალ მნიშვნელობებზე.

უიმ ფილტრის იმპულსური მახასიათებლები წარმოადგენს სისტემის რეაქციას ერთეულოვანი იმპულსის (დელტა-ფუნქციის) ტიპის შეყვანაზე. სისტემის იმპულსური რეაქცია არის სისტემის პასუხი იმპულსური სიგნალის მიმართ, როცა სისტემა თავდაპირველად „მშვიდ“ მდგომარეობაშია, ყველა შიდა ენერჯია ნულოვანია. იმპულსური სიგნალი $\delta(k)$ განისაზღვრება როგორც:

$$\text{დისკრეტული იმპულსი } \delta(k) = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases}$$

სიმ ფილტრის იმპულსური რეაქცია არის ფილტრის კოეფიციენტები და ამიტომ აქვს ფიქსირებული სიგრძე (თვლადია). თუმცა, ეს ასე არ არის უიმ ფილტრებისთვის. უიმ ფილტრის იმპულსური მახასიათებლები უსასრულოა. ითვლება, რომ ფილტრი სტაბილურია თუ $h(k) \rightarrow 0$ როცა $k \rightarrow \infty$.

MATLAB-ს აქვს `dimpulse` ფუნქცია, რომელიც გვაძლევს იმპულსური მახასიათებლების მარტივად გამოთვლის შესაძლებლობას.

უიმ ფილტრის გადაცემის ფუნქცია აღწერს ფილტრის რეაქციას სიგნალზე. უიმ ფილტრის გადაცემის ფუნქცია შეიძლება მივიღოთ პირდაპირ შემავალ/გამომავალ

სიგნალთა სხვაობის განტოლებიდან, Z-გარდაქმნის დროითი დაყენების თვისებით. ზოგადი უიმ ფილტრის გადაცემის ფუნქცია ასეთია (Ingle 2015, 130)::

$$H(z) = \frac{b_0 + b_1z^{-1} + \dots + b_Nz^{-N}}{1 + a_1z^{-1} + \dots + a_Nz^{-N}} = \frac{b_0z^N + b_1z^{N-1} + \dots + b_N}{z^N + a_1z^{N-1} + \dots + a_N}$$

უიმ ფილტრის სტაბილურობა მნიშვნელოვანი ასპექტია უიმ ფილტრების დიზაინში, რადგან ეს განსაზღვრავს, როგორ იმოქმედებს ფილტრი დროთა განმავლობაში. დისკრეტული დროითი სისტემები სტაბილურია, თუ თითოეული პოლუსის მნიშვნელობა მკაცრად ნაკლებია ერთზე. გავიხსენოთ, რომ სისტემის პოლუსები - გადაცემის ფუნქციის მნიშვნელის $X(z) - \text{მნიშვნელი}$ ფესვებია. ისინი მნიშვნელოვანია სისტემის სტაბილურობისა და დინამიური ქცევის განსაზღვრისათვის, ხოლო სისტემის ნულები - გადაცემის ფუნქციის მრიცხველი $Y(z) - \text{მრიცხველი}$ ფესვებია. ნულები გავლენას ახდენენ სისტემის გამოსავალზე სხვადასხვა სიხშირის შემომავალ სიგნალებზე. ნულების მდებარეობა განსაზღვრავს კონკრეტულ სიხშირეებზე სისტემის რეაქციის ხარისხს. სიმ ფილტრების სტაბილურობა განპირობებულია მათი პოლუსების განლაგებით ნულოვანი წერტილში, რაც უზრუნველყოფს მათ მუდმივ სტაბილურობას, სამწუხაროდ ასე არ ხდება უიმ ფილტრების შემთხვევაში.

უიმ ფილტრის გამომავალი მნიშვნელობები შეიძლება გამოითვალოს რამდენიმე სხვადასხვა მეთოდით, მათ შორის:

1. რეკურსიულად: შემავალ/გამომავალ სიგნალთა სხვაობის განტოლებიდან;
2. კონვოლუციით: $y(k) = x(k) * h(k)$;
3. $Y(z) = H(z)x(z)$ ის უკუ Z-გარდაქმნით;
4. MATLAB-ში `y = filter(num_tf,den_tf,x)` ბრძანების გამოყენებით;
5. ფურიეს სწრაფი გარდაქმნით, რომელიც ქვემოთაა განხილული.

რიცხვითი ექსპერიმენტი 5 : ავიღოთ უიმ ფილტრი, რომელიც მოცემულია ფორმულით:

$$y(k) = 0,3 \cdot y(k-1) + 0,1 \cdot y(k-2) + x(k) - x(k-1)$$

ა) გამოთვალეთ ფილტრის რიგი, უიმ ფილტრისთვის გადაცემის ფუნქცია, ფილტრის პოლუსები და დაადგინეთ, არის თუ არა ფილტრი სტაბილური.

ბ) გამოთვალეთ ფილტრის იმპულსური მახასიათებლები;

გ) გამოთვალეთ უიმ ფილტრების გამომავალი მნიშვნელობები, თუ შემავალი მნიშვნელობებია $x(k) = \{1 \ 3 \ 6 \ -1 \ 5\}$.

შედეგები: ა) უიმ ფილტრის რიგი არის ყველაზე დიდი დაყოვნება შემავალ ან გამომავალი სიგნალის, ამიტომ $N=2$;

გადაცემის ფუნქცია მიიღება სხვაობის განტოლების Z გარდაქმნით. თავდაპირველად ყველა y წევრი გადმოვიტანოთ მარცხენა მხარეს:

$$y(k) - 0,3 \cdot y(k-1) - 0,1 \cdot y(k-2) = x(k) - x(k-1)$$

$$H(z) = \frac{1 - z^{-1}}{1 - 0,3 \cdot z^{-1} - 0,1 \cdot z^{-2}} = \frac{z^2 - z}{z^2 - 0,3z - 0,1}$$

ფილტრის პოლუსები არის მიღებული განტოლების მნიშვნელის ფესვები:

$$z^2 - 0,3 \cdot z - 0,1 = (z - 0,5)(z + 0,2) = 0$$

მასასადამე ფილტრის პოლუსები არის 0,5-ზე და -0,2-ზე. სანამ პოლუსების მნიშვნელობა ორივე 1-ზე ნაკლებია, სისტემა სტაბილურია.

MATLAB-აქვს ფუნქცია roots, რომელიც გვეხმარება გამოთვალეთ პოლუსების მნიშვნელობები.

ბ) იმპულსური მახასიათებლები ფაქტიურად არის სისტემის რეაქცია, როდესაც ხდება დისკრეტული იმპულსის შესვლა. იმპულსური რეაქციის პირველი ათი მნიშვნელობა $h(0)$ დან $h(9)$ - მდე შეიძლება მოიძებნოს MATLAB-ის dimpulse ფუნქციის გამოყენებით, შემდეგნაირად:

```
num_tf = [1 -1 0];
den_tf = [1 -0.3 -0.1];
h = dimpulse(num_tf,den_tf,10)
```

h =

1.0000

-0.7000

-0.1100

-0.1030
 -0.0419
 -0.0229
 -0.0111
 -0.0056
 -0.0028
 -0.0014

რადგან ფილტრი სტაბილურია, იმპულსური რეაქცია $h(k)$ მიისწრაფვის 0-ისკენ როცა როცა $k \rightarrow \infty$. ფუნქცია dimpulse ითვლის ნებისმიერ განზომილებაში $h(k)$ -ის მნიშვნელობებს, მაგრამ მის მათემატიკურ მოდელს არ გვაწვდის. $h(k)$ -ის ხელით გამოთვლა სხვაობის განტოლების გამოყენებით რეკურსიულად იგივე ფიქსირებულ მნიშვნელობებს გავწვდის, რასაც ზემოთ, მაგრამ ეს ბევრად უფრო დიდ დროს მოითხოვს. თუ $h(k)$ -ისთვის საჭიროა მათემატიკური ფორმულა, გადაცემის ფუნქციის,

$H(z)$ -ის, უკუ Z გარდაქმნა მოგვცემს ასეთ ფორმას:
$$\frac{H(z)}{z} = \frac{z-1}{z^2-0.3z-0.1}$$

ნაწილობრივი დაშლის მეთოდით მარტივად შეიძლება MATLAB-ში. MATLAB-ში `residue` ბრძანება ავტომატურად ახორციელებს ნაწილობრივ დაშლას, რაც საგრძნობლად ამცირებს გამოთვლების სირთულეს.

```
[coeff poles] = residue([1 -1], [1 -0.3 -0.1]);
[coeff poles]

ans = -0.7143 0.5000
      1.7143 -0.2000
```

$$H(z) = \frac{-0,7143z}{(z - 0,5)} + \frac{1,7143z}{(z + 0,2)}$$

$$h(k) = -0,7143(0,5)^k + 1,7413(-0,2)^k$$

გ) უიმ ფილტრების გამომავალი მნიშვნელობების დასადგენად რამდენიმე მეთოდი არსებობს. თუ შემავალი მნიშვნელობებია $x(k) = \{1 \ 3 \ 6 \ -1 \ 5\}$ MATLAB-ში არსებობს ფუნქცია `filter`, რომელიც გვაძლევს მნიშვნელობებს:

```

num_tf = [1 -1 0];
den_tf = [1 -0.3 -0.1];
x = [1 3 6 -1 5];
y = filter(num_tf, den_tf, x)

```

```

y = 1.0000 2.3000 3.7900 -5.6330 4.6891

```

როდესაც filter ფუნქციას ვიყენებთ, გამომავალი სიგნალის სიგრძე ყოველთვის ისეთივეა, როგორც შემავალი სიგნალის სიგრძე. თუ მეტი გამომავალი მნიშვნელობები გჭირდება, შეგიძლია უბრალოდ გავზარდოთ შემავალი სიგნალი ნულებით შემდეგნაირად:

```

x = [1 3 6 -1 5 zeros(1,5)];
y = filter(num_tf, den_tf, x)

```

```

y = 1.0000 2.3000 3.7900 -5.6330 4.6891 -4.1566
-0.7781 -0.6491 -0.2725 -0.1467

```

უიმ ფილტრების და სიმ ფილტრების შედარება

- სიმ ფილტრები ყოველთვის სტაბილურია, მაშინაც კი, როდესაც ფილტრის კოეფიციენტები შემცირებული სიზუსტითაა წარმოდგენილი (დაქვანტულია). უიმ ფილტრები შეიძლება არ იყოს სტაბილური, განსაკუთრებით მაშინ, როცა კოეფიციენტები დაქვანტულია.
- უიმ ფილტრები ჩვეულებრივ უფრო დაბალი რიგისაა ვიდრე სიმ ფილტრები, რაც ნიშნავს, რომ უფრო ნაკლები გამოთვლები სჭირდება.
- სიმ ფილტრებს აქვთ ხაზოვანი ფაზური რეაგირება, რაც უზრუნველყოფს სიგნალის მინიმალურ დროით დამახინჯებას. ეს თვისება მნიშვნელოვანია იმ მოწყობილობებსა და პროგრამულ უზრუნველყოფაში, სადაც სიგნალის გადასაცემად კრიტიკულად მცირე დროა გამოყოფილი. ანუ ყველა სიხშირისთვის გამოსავალი ფაზაა. უიმ ფილტრები კი არ აქვთ ხაზოვანი ფაზური რეაგირება, რაც გარკვეულ სირთულეებს ქმნის სიგნალის დამუსავებისას.

- უიმ ფილტრები უფრო მგრძობიარეა კოეფიციენტების სიზუსტის შეცდომების მიმართ, ამიტომ მნიშვნელოვანია ყურადღება მიექცეს, თუ როგორ იქმნება და გამოიყენება ფილტრი (ფილტრის რეალიზაცია).
- უიმ ფილტრების რეკურსიული თვისება (უკუკავშირი) ზრდის შანსს არითმეტიკული გადატვირთვისთვის, რაც ნიშნავს, რომ შეიძლება გადაჭარბებული გამოთვლების შედეგად მოხდეს შეცდომა.

5.2. უიმ ფილტრის დიზაინი

ციფრული უიმ ფილტრების დიზაინისთვის არსებობს ორი ძირითადი მიდგომა: არაპირდაპირი და პირდაპირი. განვიხილოთ ორივე მათგანი.

უიმ ფილტრების არაპირდაპირი დიზაინის მეთოდი

არაპირდაპირი დიზაინის მეთოდი გულისხმობს, რომ შევექმნათ ანალოგური ფილტრი, რომელიც დააკმაყოფილებს ფილტრის მოთხოვნებს, შემდეგ კი ეს ანალოგური ფილტრი გარდავექმნათ „ტოლფას“ ციფრულ ფილტრად შემდეგი ნაბიჯების მიხედვით:

1. ჩამოვწეროთ რა მოთხოვნებს უნდა აკმაყოფილებდეს ფილტრი;
2. უნდა შევექმნათ დაბალსიხშირული პროტოტიპის ანალოგური ფილტრი ერთ-ერთი ცნობილი ბატერვორტის ან ჩებიშევის მეთოდით;
3. გარდავამნათ ეს პროტოტიპი ანალოგურ ფილტრად;
4. გარდავექმნათ ანალოგური ფილტრი შესაბამის დაბალსიხშირულ ციფრულ ფილტრად.

ციფრული ფილტრების დიზაინის საწყისი ეტაპებზე არსებობდა უამრავი ინფორმაცია ანალოგური ფილტრების დიზაინის შესახებ: ფორმულები, რომლებიც აღწერენ დაბალ-სიხშირიანი ბატერვორტის ან ჩებიშევის მეთოდით შექმნილ ფილტრებს, არსებობდა სხვა უამრავი მსგავსი ფილტრიც. ამიტომ, ლოგიკურად აუცილებელია ჯერ შეიქმნას კარგი ანალოგური ფილტრი, შემდეგ კი ეს ანალოგური ფილტრი გარდაიქმნას „ტოლფას“ ციფრულ ფილტრად. გარდაქმნა შესაძლებელია ფორმულით:

$$H(z) = H(s)|_s = \frac{2}{T_s \tan\left(\frac{\omega n}{T_s}\right)} \left(\frac{z-1}{z+1}\right) \quad (\text{Ingle 2015, 250}):$$

სადაც $T_s = \frac{1}{F_s} =$ არის დისკრეტიზაციის ინტერვალი

$\omega_n =$ სასურველი სიხშირე, რომელიც ზუსტად შეესაბამება ანალოგურ ფილტრს არის სიხშირე, რომლის დროსაც ციფრული ფილტრი იმეორებს ანალოგური ფილტრის სიხშირულ ქცევას.

ანალოგური ფილტრის გადაცემის ფუნქცია, $H(s)$, გარდაიქმნება ტოლფას ციფრულ ფილტრის გადაცემის ფუნქციად, $H(z)$, ბიწრფივი გარდაქმნის გამოყენებით. აუცილებელია რომ ციფრული ფილტრი ზუსტად დაემთხვეს ანალოგურ ფილტრს გარკვეულ, წინასწარ განსაზღვრულ სიხშირეზე (ω_n). დაბალ-სიხშირიან ან მაღალი სიხშირის ფილტრებში, ეს სიხშირე არის 3 დბ-იანი შეზღუდვის სიხშირე (cutoff frequency), ხოლო ჩამჭრელ და ზოლურ ფილტრებში, ეს სიხშირე იქნება ცენტრული სიხშირე.

არაპირდაპირი დიზაინის მეთოდი ძალიან კარგად მუშაობს უიმ ტიპის ციფრული ფილტრების დიზაინისთვის. არსებობს რამდენიმე შესანიშნავი პროგრამული ხელსაწყო, მათ შორის MATLAB-ის Filter Design and Analysis Tool (ფილტრების დიზაინისა და ანალიზის ინსტრუმენტი), რომელიც იყენებს არაპირდაპირი დიზაინის მეთოდს.

უიმ ფილტრების პირდაპირი დიზაინის მეთოდი

პირდაპირი დიზაინის მეთოდი ციფრული ფილტრის შექმნას გულისხმობს უშუალოდ, შუალედური ანალოგური დიზაინის გარეშე. MATLAB-ში ხელმისაწვდომია ორი პირდაპირი დიზაინის ალგორითმი: maxflat და yulewalk. Maxflat ქმნის მაქსიმალურად ბრტყელ (maximally flat) ციფრულ butterworth ფილტრს. მომხმარებელმა უნდა მიაწოდოს ფილტრის რიგიდა სასურველი შეჭრის სიხშირე. ამ ალგორითმის რეალიზება შესაძლებელია MATLAB-ის სამუშაო სკრიპტ სივრცეში ან Filter Design and Analysis Tool-ში. Yulewalk არის რეკურსიული ფილტრების დიზაინის ალგორითმი, რომელიც იყენებს უმცირეს კვადრატების მეთოდს, რათა ციფრული ფილტრი შეესაბამებოდეს მოცემული სიგნალის სიხშირულ სიდიდეს. მომხმარებელი აწვდის ფილტრის რიგს, სიხშირეების ვექტორს. Yulewalk შემდეგ ქმნის უიმ ფილტრს.

ორივე მეთოდი მოითხოვს მომხმარებლისგან ფილტრის რიგის განსაზღვრას, რაც საჭიროებს ექსპერიმენტებს, რათა დადგინდეს ისეთი რიგი, რომელიც წარმატებით დააკმაყოფილებს ფილტრის მოთხოვნებს.

5.3. უიმ ფილტრის გენერირება, რიცხვითი ექსპერიმენტი MATLAB-ში და შედეგები, შედეგების ანალიზი

როგორც ცნობილია ფილტრის კოეფიციენტების დაქვანტვის შეცდომები სიმ ფილტრების ფუნქციონირებაზე ძალიან მოქმედებს. უიმ ფილტრები კი ბევრად უფრო მგრძობიარეები არიან დაქვანტვის პროცესის მიმართ, რადგან უიმ ფილტრებში ხდება რეკურსიული პროცესი. ციფრული ფილტრის დიზაინში აუცილებელია შემოწმდეს თუ როგორ აისახება კოეფიციენტების დაქვანტვა ფილტრის მუშაობაზე. ქვემოთ ჩატარებულია რიცხვითი ექსპერიმენტი, სადაც ნაჩვენებია ფილტრის კოეფიციენტების დაქვანტვის გავლენა ფილტრის მუშაობაზე.

რიცხვითი ექსპერიმენტი 6: შევქმნათ მე-10 რიგის ზოლური უიმ ფილტრი, შემდეგი მახასიათებლებით:

- გამტარი უბანი 8-12 კპც,
- გატარების მაქსიმალური ზღვარი: 0.1 დბ
- დამხშობი უბანი $\leq 7\text{dB}$ და $\geq 13\text{dB}$
- მინიმალური დახშობის ზღვარი: 30 დბ
- დისკრეტიზაციის სიხშირე: 48 კპც

ა) დავადგინოთ მინიმუმ რამდენი ბიტია საჭირო ფილტრის კოეფიციენტებისთვის, რათა ფილტრი იყოს სტაბილური;

ბ) გრაფიკულად წარმოვაჩინოთ ფილტრის ამპლიტუდური რეაქცია, როცა ფილტრის კოეფიციენტები იქვანტება იმ ბიტების რაოდენობაზე, რომელიც ნაწილ ა)-ში განისაზღვრა.

გ) დავადგინოთ მინიმალური ბიტების რაოდენობა, რომელიც საჭიროა, რომ ფილტრი შეესაბამებოდეს მითითებულ მოთხოვნებს და ვაჩვენოთ შესაბამისი ამპლიტუდური რეაქცია.

ა) იმისთვის რომ დავადგინოთ მინიმუმ რამდენი ბიტია საჭირო ფილტრის კოეფიციენტებისთვის, რათა ფილტრი იყოს სტაბილური ჯერ დავაგენერიროთ ზოლური ფილტრი, ამისთვის გამოვიყენოთ MATLAB-ის ფუნქცია

Ellip

MATLAB

* ფილტრის მოთხოვნები

$F_s = 48000$; * დისკრეტიზაციის სიხშირე (Hz)

$F_{p1} = 8000$; * ქვედა გამტარი ზოლის საზღვარი (Hz)

$F_{p2} = 12000$; * ზედა გამტარი ზოლის საზღვარი (Hz)

$R_p = 0.1$; * გატარების მაქსიმალური ზღვარი:

$R_s = 30$; * მინიმალური დახშობის ზღვარი (db)

$F_{st1} = 7000$; * ქვედა დახშობის უბანი (Hz)

$F_{st2} = 13000$; * ზედა დახშობის უბანი (Hz)

* სიხშირეების ნორმალიზაცია ნაიქვისტის სიხშირის ($F_s/2$) გაყოფით

$W_p = [F_{p1} \ F_{p2}] / (F_s / 2)$; * გამტარი ზოლი

$W_s = [F_{st1} \ F_{st2}] / (F_s / 2)$; * დახშობის ზოლი

* ელიფსური ფილტრის დიზაინი (მაგალითის მიხედვით მე-10 რიგი)

$[N, W_n] = \text{ellipord}(W_p, W_s, R_p, R_s)$; * მინიმალური რიგის პოვნა

$[b, a] = \text{ellip}(N, R_p, R_s, W_p)$; * ფილტრის დიზაინი

* ფილტრის კოეფიციენტების გამოტანა

`disp('მრიცხველის კოეფიციენტები:');`

`disp(b);`

`disp('მნიშვნელის კოეფიციენტები:');`

`disp(a);`

შედეგები:

მრიცხველის კოეფიციენტები:

Columns 1 through 10

0.0320 -0.0588 0.1019 -0.0977 0.0723 0.0000 -0.0723 0.0977 -0.1019 0.0588

Column 11

-0.0320

მნიშვნელის კოეფიციენტები:

Columns 1 through 10

1.0000 -2.3505 6.0088 -8.3705 11.6692 -10.7697 9.8152 -5.9068 3.5620 -1.1551

Column 11

0.4117

შემდგომი ნაბიჯი არის ფილტრის სტაბილურობის შეფასება, დისკრეტიზაციისთვის საჭირო მინიმალური ბიტების რაოდენობის განსაზღვრა, `qtoid` ფუნქციას გამოიყენებთ, რომელიც MATLAB-ში ციფრული ფილტრების დისკრეტიზაციას და ვიზუალიზაციას ახდენს.

* ფილტრის დისკრეტიზაცია და სტაბილურობის ანალიზი

```
bq = fi(b, 1, 32, 31); * მრიცხველის კოეფიციენტები
```

```
aq = fi(a, 1, 32, 31); * მნიშვნელის კოეფიციენტები
```

* ფილტრის სტაბილურობის შემოწმება პოლუსების ანალიზით

```
[bq_double, aq_double] = deal(double(bq), double(aq));
```

```
poles = roots(aq_double);
```

* შედეგების გამოტანა

```
disp('დისკრეტული ფილტრის პოლუსები:');
```

```
disp(poles);
```

* ვიზუალიზაცია: პოლუსების და ნულების გრაფიკი

```
zplane(bq_double, aq_double); * ნულების და პოლუსების გრაფიკი
```

* შემოწმება, არის თუ არა ყველა პოლუსი ზღვარს შიგნით

```
if all(abs(poles) < 1)
```

```
disp('ფილტრი სტაბილურია');
```

```
else
```

```
disp('ფილტრი არასტაბილურია');
```

```
end
```

* ფილტრის სიხშირის რეაქცია

```
freqz(bq_double, aq_double, 1024, Fs); * სიხშირის რეაქციის ჩვენება
```

შედეგები:

დისკრეტული ფილტრის პოლუსები:

$$-0.8198 + 0.5651i$$

$$-0.8198 - 0.5651i$$

$$-0.3487 + 0.9181i$$

$$-0.3487 - 0.9181i$$

$$0.2301 + 0.9282i$$

$$0.2301 - 0.9282i$$

$$0.6859 + 0.5969i$$

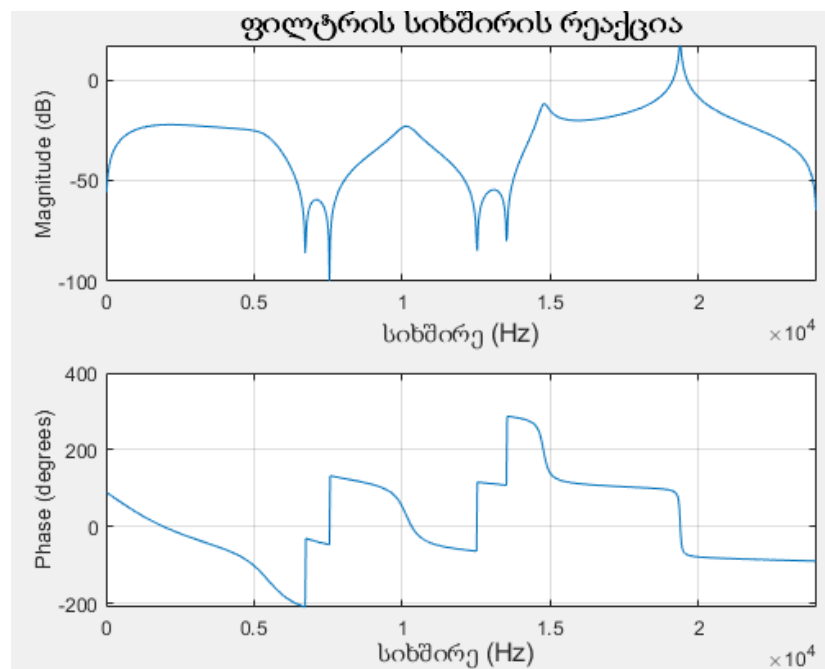
$$0.6859 - 0.5969i$$

$$0.7525 + 0.0558i$$

$$0.7525 - 0.0558i$$

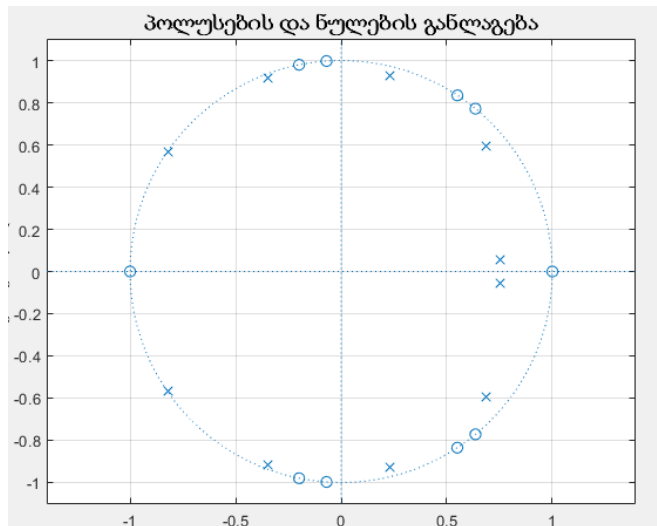
ფილტრი სტაბილურია.

ბ)



სურათი 44. ფილტრის ამპლიტუდური რეაქცია (წყარო-ავტორი)

გ) მინიმალური ბიტების დასადგენად ეტაპობრივად შევამცირეთ ბიტების რაოდენობა (მაგალითად, 16-დან 12-მდე) და შევამოწმოთ, რომ ყველა პოლუსი (poles) ზღვარს შიგნითაა, ერთეულოვან წრეწირში რცევა თუ არა. პოლუსების მსგავსი განლაგება განლაგება განაპირობებს სტაბილურობას.



სურათი 45. სტაბილური ფილტრის პოლუსები და ნულები (წყარო-ავტორი)

ექსპერიმენტის ანალიზი და დასკვნა:

ექსპერიმენტის შედეგები აჩვენებს, რომ მე-10 რიგის ზოლური უიმ ფილტრი, რომლის გატარების ზოლი 8-12 კვც-ია, მაქსიმალური გატარების ზღვართ 0.1 დბ და მინიმალური დახშობის ზღვართ 30 დბ, წარმოადგენს ეფექტურ გადაწყვეტილებას მონაცემთა დამუშავებისთვის. MATLAB-ის გამოყენებით მიღებული კოეფიციენტების გამოთვლა და ფილტრის სტაბილურობის შეფასება ხდება ნორმალიზებულ სიხშირეებზე, რაც უზრუნველყოფს სწორი დიზაინის მიღებას და სტაბილურ ფილტრის შექმნას.

შეიძლება დავასკვნათ, რომ;

- ფილტრის სტაბილურობის შესანარჩუნებლად მინიმუმ საჭიროა კოეფიციენტებისთვის 32 ბიტი, რაც უზრუნველყოფს სტაბილურობას და საშუალებას იძლევა შეინარჩუნოს მდგრადობა როგორც სიმძლავრის რეაქციის, ისე სტაბილურობის თვალსაზრისით.
- როცა ფილტრის კოეფიციენტები იქვანტება 32 ბიტზე, ფილტრის ამპლიტუდური რეაქცია სრულად შეესაბამება მითითებულ მოთხოვნებს, რაც ასევე ჩანს ვიზუალიზაციიდან.

- ფილტრის კოეფიციენტების ბიტების რაოდენობის შემცირება (მაგალითად, 16 ან 12 ბიტამდე) შეიძლება შეამციროს მისი ეფექტურობა, რაც აისახება პოლუსების განლაგებაში. სტაბილურობის შესანარჩუნებლად აუცილებელია, რომ ყველა პოლუსი დარჩეს ერთეულოვან წრეწირში.

რეკომენდაციები და იდეები მომავალი კვლევებისათვის

გამოსახულების დამუშავება წარმოადგენს ციფრული სიგნალების დამუშავების ერთ-ერთ ყველაზე აქტუალურ და შრომატევად მიმართულებას. გამოსახულებებს ყოველდღიურად ვხვდებით სხვადასხვა სფეროში, მაგ: ტექსტის გაციფრულება, სამედიცინო დიაგნოზები, სატელიტური ტექნოლოგიები. ტექნოლოგიების განვითარებასთან ერთად, გამოსახულების დამუშავების სფეროში წარმოშობილი გამოწვევები მეტად კომპლექსური გახდა, რადგან სიგნალებს ხშირად ახლავს ხმაური, დაბალი ხარისხი ან გაუმართავი მონაცემები, რომლებიც პოტენციურად ამცირებენ სისტემების ეფექტურობას.

მომავალი კვლევები გამოსახულების დამუშავების სფეროში გვთავაზობს მრავალი საინტერესო მიმართულებას, რომლის გადაჭრაც შეიძლება ძალიან მნიშვნელოვანი იყოს სხვადასხვა ინდუსტრიისთვის. მაგალითად, გამოსახულების აღდგენა დაბალი ხარისხის სიგნალებიდან, ფილტრაციის ალგორითმების გამართვა, სიგნალების ციფრული დამუშავების ტექნიკის გაუმჯობესება, წარმოადგენს ძირითადი კვლევის საკითხებს. გარდა ამისა, გამორჩეული ყურადღება უნდა დაეთმოს გამოსახულების შეკუმშვის ტექნოლოგიებს, რაც ხელს შეუწყობს მონაცემთა დიდი მოცულობების ეფექტურ შენახვას და გადაცემას, განსაკუთრებით ისეთი მოწყობილობებისთვის, რომლებსაც შეზღუდული რესურსები აქვთ.

მომავალი კვლევების ჩასატარებლად აუცილებელია ინოვაციური მიდგომების დანერგვა, როგორცაა ღრმა სწავლის (deep learning) ალგორითმები, რომლებიც

მრავალი ფუნქციის ეფექტურად შესრულებას იძლევა გამოსახულების დამუშავებაში. ამ ტექნოლოგიების გამოყენებით, შესაძლებელია კიდევ უფრო მაღალი ხარისხის სიგნალების მიღება, ფილტრაცია და გაუმჯობესებული სურათების გენერაცია. გარდა ამისა, არსებული ალგორითმების და ტექნოლოგიების გაუმჯობესება შესაძლოა მნიშვნელოვანი წინგადადგმული ნაბიჯი იყოს ისეთი გლობალური პრობლემების გადასაჭრელად, როგორცაა სამედიცინო დიაგნოსტიკა, ხელოვნური ინტელექტის განვითარება და ავტომატიზებული სისტემები.

გამოსახულების დამუშავება, განსაკუთრებით ტექსტის ამოცნობის კონტექსტში, მოიცავს რიგ სირთულეებს, როგორცაა ტექსტის სურათების ვარიაციები, ფონტის სტილი, სურათის გარჩევადობა და დაბალი ხარისხი. ამ ფაქტორებმა შეიძლება მნიშვნელოვნად შეაფერხოს სისტემის ეფექტურობა და საჭირო გახდეს სურათების წინასწარი დამუშავება. მაგალითად, დასურათებული ტექსტი, რომელიც შეიცავს ხმაურს ან არაერთგვაროვანი ფერების ფონს, შესაძლოა ვერ ამოიცნოს სისტემამ ზუსტად.

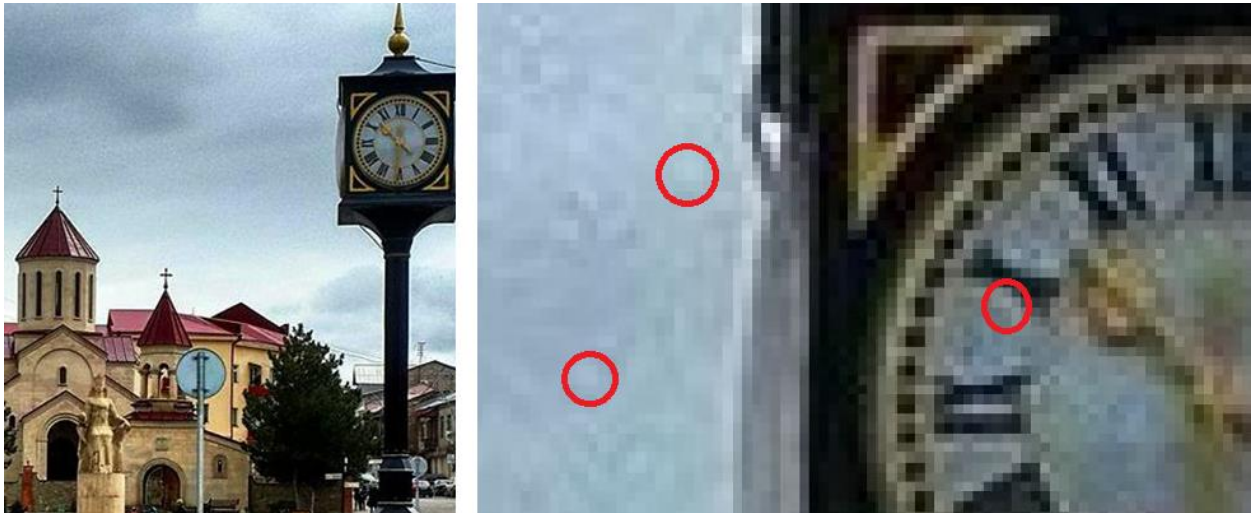
ტექსტის ოპტიკური ამოცნობის სისტემები, მათ შორის MATLAB-ში რეალიზებული ალგორითმები, ფოკუსირდებიან აღნიშნული გამოწვევების გადაჭრაზე. MATLAB-ის გამოყენება ამ პროცესში განსაკუთრებით ეფექტურია, რადგან მისი სიმძლავრე მატრიცულ ოპერაციებში და ალგორითმების რეალიზაციაში ამარტივებს კომპლექსური პრობლემების გადაჭრას.

ქართული ტექსტის ამოცნობა გამოსახულებიდან მნიშვნელოვანი ნაბიჯია ქართული კულტურის ციფრულ სივრცეში გადმოსატანად და მისი შენარჩუნების უზრუნველსაყოფად. მოცემული კვლევა შესაძლოა მივმართოთ ამ პროცესის ეფექტურობის გაზრდისკენ, რაც მნიშვნელოვანია როგორც მეცნიერებისთვის, ისე პრაქტიკული გამოყენებისთვის.

სანამ უშუალოდ ვისაუბრებთ დისერტაციაში განხილული ფილტრების პრაქტიკულ გამოყენებაზე ქართული დამწერლობის გაციფრულების საქმეში, მანამდე მოვიყვანოთ რამდენიმე მნიშვნელოვანი განმარტება

პიქსელი (Pixel) არის სიტყვა "Picture Element"-ის მოკლე ფორმა და წარმოადგენს ციფრული გამოსახულების უმცირეს კომპონენტს, ერთ-ერთ კვადრატულ უჯრას.

ციფრული გამოსახულება 1024×768 არის უბრალოდ კვადრატული ბადე, რომელიც შეიცავს 1024 სვეტსა და 768 სტრიქონს, საერთო ჯამში 786,432 პიქსელს. დიდი ზომის გამოსახულებებს სჭირდებათ მეტი პიქსელი, რათა თვალმა ვერ გაარჩიოს ცალკეული პიქსელებს გამოსახულებაში და აღიქვას ერთ მთლიან გამოსახულებად. ცალკე პიქსელები შეიძლება ნახოთ, თუ ციფრულ გამოსახულებასე გავადიდებთ, როგორც ეს სურათზეა ნაჩვენები.



სურათი 46. გადიდებული ციფრული გამოსახულება (წყარო-ავტორი)

ფერის სიღრმე (Color Depth) არის იმ ბიტების რაოდენობა, რომელიც გამოიყენება თითოეული პიქსელის ფერის ან ჩრდილის გამოსახვადად. შავი და თეთრი გამოსახულებები საჭიროებენ მხოლოდ 1 ბიტს თითო პიქსელზე (0 = შავი და 1 = თეთრი). შავ-თეთრი გრადაცია გამოსახულებები იყენებენ 8 ბიტს თითო პიქსელზე, რის შედეგადაც 256 (2^8) სხვადასხვა გრადაცია მიიღება, რომელიც შავს და თეთრს შორის მერყეობს. ფერადი გამოსახულებებისთვის საჭიროა უფრო მეტი ბიტი, რათა უზრუნველყოს ფერების ფართო არეალი. „ნამდვილი ფერის“ გამოსახულებები იყენებენ 24 ბიტს თითო პიქსელზე, სადაც 8 ბიტია წითელი, 8 ბიტი მწვანე და 8 ბიტი ლურჯი, რაც დაახლოებით 16.8 მილიონ ფერების კომბინაციას ქმნის ($256 \times 256 \times 256$). უფრო მოწინავე გრაფიკული სისტემები შეიძლება გამოიყენებდნენ 32 ან 48 ბიტს თითო პიქსელზე, რაც კიდევ უფრო მეტ ფერებს უზრუნველყოფს.

მაგალითისთვის: თუ გვაქვს სურათი ზომით 2736 x 3648, რომელიც იყენებს 24 ბიტს თითო პიქსელზე (8 ბიტი წითელი, 8 ბიტი მწვანე და 8 ბიტი ლურჯი), გამოვთვალოთ მისი ზომა და რამდენი ასეთი სურათი დაეტევა 2 GB SD ბარათზე.

1. სურათის ზომა:

- თითო პიქსელზე 24 ბიტი (3 ბაიტი, რადგან 1 ბიტი = 1/8 ბაიტი).
- სურათის ზომა იქნება: 2736 x 3648 პიქსელი x 3 ბაიტი.

2. თუ გავიგებთ სურათის ზომას ბაიტებში, მაშინ:

- სურათის ზომა = 2736 x 3648 x 3 ბაიტი = 29,986,304 ბაიტი.

3. ახლა, როგორ გავიგოთ რამდენი სურათი ეტევა 2 GB SD ბარათზე:

- 2 GB = 2 x 1024 x 1024 x 1024 ბაიტი = 2,147,483,648 ბაიტი.
- რაოდენობა სურათების = 2,147,483,648 / 29,986,304 ≈ 71 სურათი.

ამრიგად, 2 GB SD ბარათზე შეიძლება დაეტიოს 71 სურათი, თითოეული 2736 x 3648 ზომით და 24 ბიტის ფერის სიღრმით.

შეკუმშვა დანაკარგით და დანაკარგის გარეშე

ციფრულ ფოტოზე რაც უფრო მეტია პიქსელების როდენობა, მით უფრო დიდია ზომაში. არსებობს მრავალი სხვადასხვა ფორმატი ციფრული სურათებისთვის, რომლებიც ცდილობენ სურათის ზომის შემცირებას ან შეკუმშვას. ნებისმიერი შეკუმშვის ტექნიკა, რომელიც საშუალებას იძლევა, რომ თავდაპირველი სურათი სრულად აღვაგინოთ ეწოდება შეკუმშვა დანაკარგის გარეშე. შეკუმშვის ტექნიკები, რომლებიც ზიანს აყენებს თავდაპირველ სურათს და წარმოშობენ გარკვეულ ინფორმაციას დაკარგვას, ეწოდება შეკუმშვა დანაკარგით.

გამოსახულების სიხშირული სპექტრი

სიხშირული სპექტრი გამოსახულებებში არის ინფორმაცია, რომელიც გამოხატავს, რამდენად ხშირად და როგორ იცვლება გამოსახულების დეტალები. მაღალი სიხშირე ნიშნავს იმას, რომ გამოსახულებაში ხშირია სწრაფი ცვლილებები, მაგალითად, წვრილი ხაზები, კუთხეები, დეტალები. ეს ქმნის გამოსახულების "მცირე" და "მახვილ" ელემენტებს. დაბალი სიხშირე ნიშნავს, რომ გამოსახულებაში ცვლილებები ნელია და ნაკლებად მკვეთრია, რაც ქმნის "მძიმე" ან "მშვიდ" ნაწილებს, მაგალითად, ერთფეროვანი ფერის არეებს გამოსახულებაზე.

ზოგიერთი კარგად ცნობილი ფორმატი

BMP (Windows bitmap)

.bmp BMP (Windows bitmap) ფორმატი არის ფაილის ტიპი, რომელიც გამოიყენება სურათების შესანახად Windows-ის ოპერაციულ სისტემაში. ეს ფორმატი არ იყენებს შეკუმშვის ტექნოლოგიას, რის გამოც ფაილები ხშირად დიდია. BMP ფორმატში სურათები შეიძლება იყოს შავი-თეთრი ან ფერადი, ფერის სიღრმე კი შესაძლებელია 1 ბიტიდან (შავი-თეთრი) 24 ბიტამდე. რეზოლუცია და დეტალები BMP ფორმატში დამოკიდებულია იმაზე, თუ რამდენი ბიტი გამოიყენება თითოეული პიქსელისთვის.

GIF (Graphics Interchange Format)

.gif ფორმატი იყენებს დანაკარგების გარეშე შეკუმშვის ალგორითმს და აქვს ფერის სიღრმე 8 ბიტი თითო პიქსელზე, რაც ნიშნავს, რომ ფერების რაოდენობა შეზღუდულია 256-მდე. ეს ფორმატი კარგად მუშაობს დიაგრამებისთვის, მულტიფილმებისთვის და სხვა გამოსახულებებისთვის, რომლებშიც არ არის დიდი რაოდენობის ფერები, ეს ფორმატი გამოიყენება ანიმაციებშიც.

PNG (Portable Network Graphics)

.png ფორმატი განავითარა Web გვერდების არსებობამ, როგორც ალტერნატივა .gif ფორმატისათვის. იგი იყენებს დანაკარგების გარეშე შეკუმშვის ტექნიკას და გთავაზობთ გაცილებით უფრო ფართო ფერის არჩევანს, ვიდრე GIF-ი. აღწევს 48 ბიტს თითო პიქსელზე. PNG ფორმატი კარგად მუშაობს ფოტოების რედაქტირებისთვის, მაგრამ ის საუკეთესო არჩევანი არ არის ფოტოების საბოლოო შენახვისთვის, რადგან PNG ფაილი დეტალური ფოტოსურათისთვის, რომელშიც ბევრი ტექსტურაა და ფერი, ძალიან დიდია ზომაში.

JPEG (Joint Photographic Expert Group)

JPEG ფორმატი იყენებს დანაკარგებით შეკუმშვის ტექნიკას. ეს არის პოპულარული ფორმატი ფოტოსურათების შესანახად ან გასავრცელებლად. ზომით JPEG ფაილები უფრო მცირეა, ვიდრე PNG ფაილები. თუმცა, რადგან შეკუმშვა დანაკარგიანია, JPEG ფაილების ხშირი გახსნა, რედაქტირება და ხელახლა შენახვა შეიძლება გამოიწვიოს სურათის ხარისხის გაუარესება. უკეთესი ალტერნატივაა ფოტოსურათის

რედაქტირება დანაკარგების გარეშე ფორმატში, მაგალითად, PNG-ში და შემდეგ საბოლოო სურათის JPEG ფორმატში შენახვა.

TIFF (Tagged Image File Format)

TIFF ფორმატი მოქნილია, რადგან მასში შესაძლებელია სხვადასხვა შეკუმშვის ტექნიკების გამოყენება, ასევე ფერის სიღრმის არჩევაც. მიუხედავად იმისა, რომ ეს მოქნილობა იწვევს TIFF ფაილების დიდი მრავალფეროვნების არსებობას, TIFF ფორმატი ცუდად მხარდაჭერილია ვებ-ბრაუზერების მიერ და ზოგიერთი წამკითხველი შესაძლოა კითხულობდეს TIFF-ის გარკვეულ ტიპებს, მაგრამ არა ყველას. ასევე ზოგიერთი ციფრული კამერა სურათებს TIFF ფორმატში ინახავს

სურათების დამუშავება, ხდება ფილტრაციით. როცა სურათს ვამუშავებთ, ახალი სურათის პიქსელები (ყველაზე პატარა სურათის ელემენტები) იქმნება არა მხოლოდ პირვანდელი პიქსელისგან, არამედ მისი ირგვლივ არსებული პიქსელებისგანაც. ეს ყველაფერი ხდება გარკვეული ფორმულის მიხედვით, რომელიც მათ წრფივად აერთიანებს.

ეს მეთოდი გამოიყენება სურათების გლუვი ან დაბინდული ეფექტისთვის. მაგალითად, როცა ფოტოს ვასწორებთ, შეიძლება გამოვიყენოთ ფილტრი, რომელიც ეფუძვნება მარტივ ფორმულას ან ცხრილს, სადაც თითოეულ პიქსელს მინიჭებული აქვს გარკვეული წონა, ამგვარად სურათის საბოლოო შედეგი შესაძლოა მივიღოთ უფრო გლუვი და ლამაზი. კონცეფციის ილუსტრაციისთვის განვიხილოთ შემდეგი ძალიან მარტივი წონითი მატრიცა ანუ ფილტრი:

$$h = \begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix}$$

თუ გამოსახულებას გავფილტრავთ ამ წონითი მატრიცით, გაფილტრული გამოსახულების პიქსელი (i, j წვერი) შეიცვლება.

ფილტრი მოვათავსოთ პიქსელზე (i,j) თავდაპირველი გამოსახულების აღმწერ მატრიცაზე. შემდეგ, ფილტრის კოეფიციენტებს გავამრავლებთ იმ პიქსელებზე,

რომლებიც ფილტრის ქვეშ არიან და ყველა ნამრავლს შევკრიბავთ, მივიღებთ ახალი პიქსელის მნიშვნელობას. სურათზე ჩანს ფილტრაციის პროცესი, 7×7 გამოსახულების მატრიცის, სადაც ფილტრი მოთავსებულია ცენტრში პიქსელზე ინდექსით (3,4).

80	64	66	64	60	64	67
75	72	$68^{1/16}$	$67^{1/8}$	$65^{1/16}$	50	48
70	75	$64^{1/8}$	$63^{1/4}$	$61^{1/4}$	56	52
70	78	$62^{1/16}$	$58^{1/8}$	$67^{1/16}$	58	56
76	80	60	56	73	62	60
84	88	62	58	75	64	64
88	92	70	64	80	68	60

ცხრილი 9. ფილტრაციის პროცესი, 7×7 გამოსახულების მატრიცის, სადაც ფილტრი მოთავსებულია ცენტრში პიქსელზე ინდექსით (3,4) (წყარო-ავტორი).

გაფილტრული გამოსახულების პიქსელი (3,4) შემდეგი ფორმულით გამოითვლება:

$$= \frac{1}{16} \times 68 + \frac{1}{8} \times 67 + \frac{1}{16} \times 65 + \frac{1}{8} \times 64 + \frac{1}{4} \times 63 + \frac{1}{8} \times 61 + \frac{1}{16} \times 62 + \frac{1}{8} \times 58 + \frac{1}{16} \times 67$$

სრული გამოსახულება ფილტრაციას გადის პირველ რიგში ფილტრის მატრიცის განთავსებით გამოსახულების ზედა მარცხენა პიქსელზე, შემდეგ გამოთვლიან ახალ პიქსელს. შემდეგ ფილტრი გადატანილია მარჯვნივ ერთი პიქსელით, გამოთვლიან ახალ პიქსელს და ამ პროცესს იმეორებენ სანამ რიგი არ დასრულდება. შემდეგ ფილტრი განთავსდება მეორე რიგის მარცხენა მხარეს და ახალ პიქსელებს გამოთვლიან, როდესაც ფილტრი გადაადგილდება მეორე რიგში. პროცესი მეორდება, სანამ ბოლო პიქსელამდე არ მივა. გამოსახულების კიდეზე ზოგიერთი ფილტრის კოეფიციენტი შეიძლება გადაკვეთოს საზღვარი. არარსებული პიქსელები ითვლება ნულებად, რაც შეიძლება გამოსახულების საზღვრებზე მუქი ფერი გამოიწვიოს. არსებობს სხვა ვარიანტები კიდის პიქსელების დასამუშავებლად. ერთი ვარიანტია, რომ ფილტრაციის პროცესი დაიწყოს და დამთავრდეს ისე, რომ ფილტრი არ გადაკვეთოს საზღვარი. მაგალითის მიხედვით, ფილტრაცია დაიწყება მეორე რიგისა და მეორე სვეტის პიქსელიდან და დასრულდება მეექვსე რიგისა და მეექვსე სვეტის

პიქსელზე. პირველი და ბოლო რიგები ან სვეტები არ გადადიან ფილტრაციას და დარჩებიან პირვანდელი მნიშვნელობებით. მეორე ვარიანტია, რომ ფილტრის გადატანის შემდეგ ბოლო პიქსელები, რომლებიც არ გადადიან ფილტრაციას, ამოიღონ, რაც გამოსახულების ზომის მცირე შემცირებას გამოიწვევს.

გამოსახულების იმპორტი (Importing Images)

გამოსახულების იმპორტი MATLAB-ში ხდება `imread` ბრძანების გამოყენებით. ეს ბრძანება იყენებს გამოსახულების ფაილს და აბრუნებს მას როგორც მასივს, რომლის ელემენტები ასახავს თითოეულ პიქსელს. მაგალითად:

```
X = imread('filename.png');
```

აქ `filename.png` არის იმპორტირებული გამოსახულების ფაილი, რომელიც ინახება მასივში `X`.

გამოსახულების ექსპორტი (Exporting Images)

გამოსახულების ექსპორტი MATLAB-იდან ხდება `imwrite` ბრძანების მეშვეობით. ეს ბრძანება გადაყავს გამოსახულების მასივი ფაილში. მაგალითად:

```
imwrite(X, 'output_image.png');
```

აქ `X` არის ის გამოსახულება, რომელიც ექსპორტირდება უნდა მოხდეს და `output_image.png` არის ფაილი, რომელშიც გამოსახულება ინახება.

გამოსახულების გამოტანა (Displaying Images)

გამოსახულების ჩვენება MATLAB-ში ხდება რამდენიმე ბრძანების მეშვეობით. ძირითადი მეთოდები არიან:

- **`image(X)`** - ეს ბრძანება აჩვენებს გამოსახულებას, რომელიც დამუშავებულია `X` მასივის სახით. თუ `X` არის ორგანზომილებიანი მასივი (მარტივი შავ-თეთრი სურათი), მაშინ საჭიროა ფერების რუკა (`colormap`).
- **`imshow(X)`** - ეს ბრძანება კიდევ უფრო ხშირად გამოიყენება გამოსახულებების ჩვენებისთვის, განსაკუთრებით მაშინ, როცა გამოსახულება არის შავ-თეთრი ან ფერადი.

```
imshow(X); % აჩვენებს გამოსახულებას
```

გამოსახულების ფილტრაცია (Filtering Images)

გამოსახულების ფილტრაცია MATLAB-ში ხდება შემდეგი ფუნქციების გამოყენებით:

- **filter2(F, Im)** - ეს ბრძანება ფილტრავს ორ განზომილებიან გამოსახულებას Im ფილტრის მატრიცით F და აბრუნებს ფილტრირებულ გამოსახულებას Imf.

$Imf = filter2(F, Im);$

- **imfilter(Im, F)** - ეს ბრძანება უფრო რთული ფილტრაციისთვის გამოიყენება, განსაკუთრებით როდესაც ჩვენ გვაქვს მრავალგანზომილებიანი გამოსახულებები. ფილტრავს გამოსახულებას Im და ასრულებს ფილტრაციას F მატრიცით.

$Imf = imfilter(Im, F);$

ამის სემდეგ საჭიროა (RGB) გამოსახულების შავ-თეთრ (grayscale) გამოსახულებად გარდაქმნა.

MATLAB-ში ფერადი (RGB) გამოსახულების შავ-თეთრ (grayscale) გამოსახულებად გადაქცევა მარტივად ხდება. RGB გამოსახულება არის სამი ჩრდილის კომბინაცია: წითელი, მწვანე და ლურჯი (Red, Green, Blue). შავ-თეთრ გამოსახულებაში კი თითოეული პიქსელის კოორდინატი წარმოადგენს სინათლის ინტენსივობას, რომელიც მოცემულია ერთი მნიშვნელობით (0-დან 255-მდე).

RGB გამოსახულების შავ-თეთრად გადაქცევის პროცესი:

1. **RGB პიქსელის მნიშვნელობების შერევა:** შავ-თეთრ გამოსახულებაში თითოეული პიქსელი წარმოადგენს RGB ელემენტების საშუალო მნიშვნელობას ან მათი წონასწორობით შერევას. ეს ნიშნავს, რომ RGB პიქსელის ყველა კომპონენტს (წითელს, მწვანეს და ლურჯს) გააჩნია გარკვეული წონა, რომელიც განსაზღვრავს შავ-თეთრ გამოსახულებაში იმავე პიქსელის სიმბოლო.

2. **rgb2gray ფუნქციის გამოყენება:** MATLAB-ში არსებობს ფუნქცია `rgb2gray`, რომელიც ავტომატურად აკეთებს RGB გამოსახულების შავ-თეთრად გადაქცევას. ეს ფუნქცია იღებს RGB გამოსახულებას და პიქსელების საშუალო მნიშვნელობას ცვლის შავ-თეთრი გამოსახულების მნიშვნელობად.

ციფრული ფილტრების პრაქტიკული გამოყენების იდეა ქართული ტექსტის ოპტიკურ ამომცნობში

ბოლო ათწლეულების განმავლობაში მანქანური კითხვა რეალობად იქცა. ანბანური სიმბოლოების ამოცნობა ერთ-ერთი წარმატებული ნიმუშია გამოსახულების ციფრულად დამუშავების სფეროში.

დღესდღეობით არსებობს მრავალი კომერციული და არაკომერციული სისტემა, რომელიც ტექსტის ციფრულ გარდაქმნას ახდენს. თუმცა, კომპიუტერს ჯერ კიდევ არ შეუძლია ადამიანის მიერ ტექსტის წაკითხვის სიზუსტეს გაუწიოს კონკურენცია. სიმბოლოების ამოცნობის ამოცანა მოიცავს დახატული სიმბოლოების ამოცნობას და ციფრულ ფორმატში გარდაქმნას. ამის შემდეგ შესაძლებელია ტექსტის დაკოპირება და რედაქტირება Microsoft Word-ით ან სხვა ტექსტური რედაქტორით.

სისტემის ეფექტურობა შეიძლება გაუმჯობესდეს გამოსახულების დამუშავების ტექნიკების გამოყენებით, როგორცაა სიმ და უიმ ფილტრები, რომლებიც ხელს უწყობენ დაბალი გარჩევადობის სურათებში არსებული ხმაურის შემცირებას. გამოსახულების წინასწარი ფილტრაცია არა მხოლოდ ტექსტის სიმბოლოების ამოცნობის სიზუსტეს ზრდის, არამედ რთული ფონის, სხვადასხვა ფონტის სტილისა და ორიენტაციის შემთხვევაში პრობლემების შემსუბუქებასაც უწყობს ხელს. შესაბამისად, ნაშრომში MATLAB-ში დაპროგრამებული ფინიტური იმპულსური (სიმ) და უსასრულო იმპულსური (უიმ) ფილტრები ეფექტური ინსტრუმენტებია ტექსტის, მათ შორის ქართული ტექსტის ამომცნობის დაპროგრამებისთვის (Tabeshadze 2025).

ქართული დამწერლობა გამორჩეული და უძველესი კულტურული მემკვიდრეობაა, რომელიც მსოფლიოს 14 დამწერლობიდან ერთ-ერთია. სწორედ ამიტომ, მნიშვნელოვანია ხელი შევუწყოთ მის ციფრულ ფორმატში გადატანასა და პოპულარიზაციას.

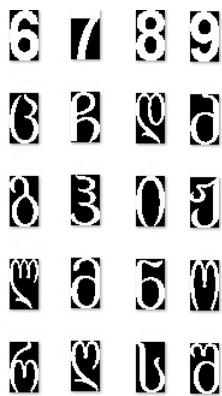
როგორც ზემოთ აღვნიშნეთ არსებობს გამოსახულებიდან ტექსტის გაციფრულების უფასო სისტემები, თუმცა აქვთ გარკვეული შეზღუდვები, მაქსიმალურად კარგი შედეგის მისაღებად აუცილებელია სურათი, საიდანაც ტექსტის ამოღება ხდება, იყოს

მაღალი გარჩევადობის, მინიმუმ 10 პიქსელი, ასევე ფაილის ზომა არ უნდა აღემატებოდეს 2 მეგაბაიტს, არ უნდა იყოს ჩრდილები, არ უნდა იყოს ფოტო დაზიანებული და ა.შ. თუ ასეთი სურათია წასაკითხი, მაშინ ჯერ ხელით უნდა დამუშავდეს ფოტო, მოსცილდეს ზედმეტი ობიექტები, გახდეს ერთგვაროვანი ფერების კორექციით, ტექსტის ფონისგან გამოკვეთით, სიკაშკაშის მორგებით და ა.შ. ეს კი ძალიან რთულია ჩვეულებრივი მომხმარებლისთვის, რომელსაც მარტივად სურს შედეგის მიღება.

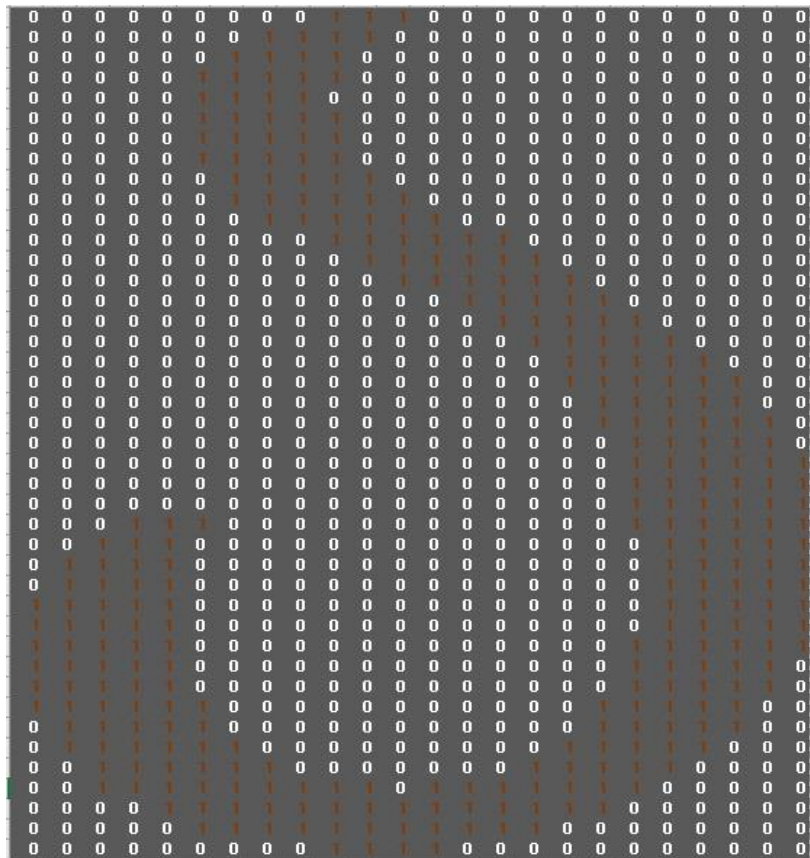
სურათიდან ტექსტის გაციფრულება მრავალფუნქციური პროცესია, რომელიც მოიცავს წინასწარ დამუშავებას, სემენტაციას და სიმბოლოების ამოცნობას, რაც შესაძლებელია ოპტიმიზირებული ფილტრაციის მეთოდებით.

ალგორითმი შესაძლოა აღიწეროს შემდეგნაირად (Tabeshadze 2025).:

1. წინასწარი ფორმირებული მატრიცის შექმნა: ქართულ ანბანზე დაყრდნობით, საჭიროა წინასწარ შექმნილი მატრიცა, რომელიც შეიცავს თითოეული სიმბოლოს



სურათი 47.
ქართულ ანბანზე
დაყრდნობით,
წინასწარ შექმნილი
მატრიცა (წყარო-
ავტორი)



სურათი 48. „ა“ ასო-სიმბოლო გამოსახულია მატრიცის
სახით (წყარო-ავტორი)

წარმოდგენას. ეს მატრიცა წარმოადგენს ეტალონს, რომელთანაც მოხდება ფოტოს შედარება, საიდანაც გვჭირდება ტექსტის ამოცნობა.

საწყისი მატრიცის შექმნა გულისხმობს ეტალონის ფორმირებას, რომელიც შედგება სხვადასხვა სიმბოლოს წარმომადგენელი ბადეებისგან. თითოეული სიმბოლო გამოსახულია მატრიცის სახით, სადაც ბადის უჯრედები შავ-თეთრ პიქსელებად იყოფა. შავ პიქსელებზე გადაწერილია 0, ხოლო თეთრ პიქსელებზე – 1. ეს პროცესი ქმნის ბინარულ მატრიცას, რომელიც წარმოადგენს კონკრეტული ქართული სიმბოლოს უნიკალურ ციფრულ სახეს (Tabeshadze 2025)..

1. გამოსახულების წინასწარი დამუშავება:



სურათის ფერების გრადაციიდან ნაცრისფერ ტონალობაში გადაყვანა, რაც ამცირებს მონაცემთა მოცულობას და აადვილებს შემდგომ

ანალიზს.



ზემოთ განხილული MATLAB-ის ფუნქციები `rgb2gray` დაგვეხმარება სურათის შავ-თეთრ

ტონალობაში გადაყვანაში, რათა გაიზარდოს სიმბოლოების და ფონის კონტრასტი. ხმაურის მოცილება და გამოსახულების გასწორება შესაძლებელია სიმ და უიმ ფილტრების გამოყენებით, რაც გაუმჯობესებს სიმბოლოების ამოცნობის სიზუსტეს.

2. სეგმენტაცია:



სურათის სეგმენტაცია გულისხმობს თითოეული სიმბოლოს გამოყოფას და ნიმუშების აღებას. ეს პროცესი მოითხოვს

ისეთ მეთოდებს, რომლებიც გამოიყენება სიგნალების ციფრულ დამუშავებაში, როგორცაა სიგნალის დისკრეტიზაცია და ფილტრაცია. სეგმენტაცია საშუალებას აძლევს სისტემას ცალკეულ სიმბოლოებზე კონცენტრირდეს და შემდგომ ეტაპებზე უკეთ მოახდინოს მათი იდენტიფიკაცია

4. კორელაციის გამოკვლევა და დამთხვევა:

როდესაც ახალი გამოსახულება უნდა გაციფრულდეს, პირველ რიგში, იგი გადაყვანილია შავ-თეთრ ტონალობაში და დამუშავებულია ისე, რომ ხმაური

შემცირდეს და სიმბოლოები გამოიკვეთოს. ამის შემდეგ, სურათიდან მიღებული ახალი მატრიცა შედარებულია საწყის ბინარულ მატრიცასთან (ეტალონთან). ამ შედარებისას, თითოეულ პოზიციაზე, სადაც 0-ისა და 1-ის დამთხვევა ხდება, სიმბოლოს იდენტიფიცირებაც ხდება (Tabeshadze 2025)..

ეს პროცესი უზრუნველყოფს, რომ ახალი სიმბოლოები სწორად იყოს ამოცნობილი იმ შემთხვევაშიც კი, თუ გამოსახულება შეიცავს მცირე ხმაურს ან განსხვავებებს. ასეთი მიდგომა საშუალებას აძლევს სისტემას მაღალი სიზუსტით შეადაროს და გამოავლინოს სიმბოლოები, რაც მნიშვნელოვანია ტექსტის ციფრული ფორმატში გადაყვანისთვის.

5. ამოცნობილი სიმბოლოების Word-ში გადაყვანა MATLAB-ის გამოყენებით:

ამოცნობილი სიმბოლოები უნდა გადაიყვანოთ ტექსტურ ფორმატში. ეს შეიძლება იყოს სიმბოლოების შემნახველი ცვლადი ან სტრიქონების მასივი MATLAB-ში.

```
recognizedText = ['ს', 'ა', 'ქ', 'ა', 'რ', 'თ', 'ვ', 'ე', 'ლ', 'ო'];  
fullText = strjoin(recognizedText, "');
```

შევქმენათ დროებითი ტექსტური ფაილი, სადაც ჩაიწერება ტექსტი **fprintf**

ფუნქციის გამოყენებით

```
fileID = fopen('recognizedText.txt', 'w');  
fprintf(fileID, '%s\n', fullText);  
fclose(fileID);
```

შევინახოთ ტექსტური ფაილი და გავხსნათ იგი Microsoft Word-ში **winopen**

ფუნქციის გამოყენებით

```
winopen('recognizedText.txt');
```

დასკვნები და რეკომენდაციები

დისერტაციაში წარმოდგენილი ნაშრომის საფუძველზე გაკეთებული დასკვნები:

- დისერტაციაში წარმოდგენილმა კვლევამ ცხადყო, რომ ელექტრონულ წრედებში გამავალი სიგნალის აღდგენა არის მნიშვნელოვანი გამოწვევა თანამედროვე

საინჟინრო და კომპიუტერულ სისტემებში, რომლის ეფექტური გადაჭრა შესაძლებელია ციფრული სიგნალის დამუშავების მეთოდების გამოყენებით.

- ციფრული სიგნალის აღდგენის ეფექტიანად განხორციელება მოითხოვს როგორც თეორიული საფუძვლების ღრმა გააზრებას (მაგალითად, ნაიქვისტ-შენონის თეორემა, Z-გარდაქმნა, კონვოლუცია, ფურიეს ანალიზი), ისე პროგრამული მოდელირებისა და სიმულაციის ცოდნას MATLAB-ის გამოყენებით.

- სიმ და უიმ ფილტრების შედარებით ანალიზმა აჩვენა, რომ სიმ ფილტრები გამორჩეულია ფაზური სტაბილურობითა და პროგნოზირებადობით, რაც მათ განაპირობებს ისეთი სისტემებისთვის, სადაც ხარისხი და სტაბილურობა კრიტიკულია. უიმ ფილტრები კი გამოდგებიან კომპაქტური და რესურსდამზოგავი სისტემებისთვის.

- რეკონსტრუქციის ალგორითმების შესწავლამ ცხადყო, რომ ინტერპოლაციის სხვადასხვა მეთოდი (სწორი, სპლაინური და სხვა) განსხვავებულ ხარისხს იძლევა სიგნალის აღდგენისას. კონკრეტული მეთოდის არჩევანი დამოკიდებულია სიგნალის ტიპსა და დამახინჯების დონეზე.

- MATLAB-ის გამოყენება კვლევის მთავარ ინსტრუმენტად მიანიშნებს პროგრამული სიმულაციების ძალასა და საჭიროებაზე ციფრული სისტემების განვითარების პროცესში. MATLAB-ის ხელსაწყოებმა საშუალება მისცა დეტალურად შეესწავლა სიგნალის დამუშავების სხვადასხვა საფეხურები.

- გამოყენებული მათემატიკური მოდელირება, მათ შორის დიფერენციალური განტოლებები, ტრანსფორმაციები და სტატისტიკური მეთოდები, წარმოადგენს ძლიერ თეორიული ბაზის საფუძველს, რაც უზრუნველყოფდა კვლევის სანდოობას.

- სიგნალის აღდგენის პროცესში წამოჭრილი გამოწვევების ერთ-ერთი მნიშვნელოვანი წყარო იყო ზედდების ეფექტი (aliasing), რომლის დასაძლევად დისერტაციაში გამოყენებულ იქნა ანტი-ალიასინგური ფილტრები და სხვა დამცავი სტრატეგიები.

- ექსპერიმენტებმა აჩვენა, რომ მაღალი სიზუსტის მიღწევა შესაძლებელია მხოლოდ იმ შემთხვევაში, თუ ანალოგურ-ციფრული გარდაქმნის ეტაპზე დაცულია

ნაიქვისტ-შენონის კრიტერიუმი და ფილტრაციის ალგორითმები სწორადაა შერჩეული.

- ციფრული სიგნალის დამუშავების ალგორითმების ეფექტიანობა მნიშვნელოვნად დამოკიდებულია არჩეული დროითი ინტერვალის სიგრძესა და სიხშირეზე. ამ პარამეტრების ოპტიმიზაცია კვლევის ერთ-ერთ მნიშვნელოვანი ნაწილი იყო.

- დისერტაციაში წარმოდგენილი მოდელების გამოყენება შესაძლებელია არა მხოლოდ თეორიული მიზნებისთვის, არამედ პრაქტიკულ პროგრამულ უზრუნველყოფაში, მათ შორის ქართული ტექსტის ოპტიკური ამომცნობი სისტემებისთვის.

- კვლევამ დაადასტურა, რომ ციფრული სიგნალის დამუშავების ხელსაწყოების გამოყენება (მაგალითად, Signal Processing Toolbox, Filter Design Toolbox და სხვა MATLAB-ის ბიბლიოთეკები) მნიშვნელოვნად ამარტივებს კომპლექსური სისტემების ანალიზს.

- პრაქტიკული ექსპერიმენტებით მიღებული შედეგები ემთხვევა თეორიულად მიღებულ დასკვნებს, რაც კვლევის სანდოობის მაჩვენებელია.

- დისერტაციაში ჩატარებულმა მოდელირებამ აჩვენა, რომ სწორად შერჩეული ალგორითმები და მათი მორგება კონკრეტულ სიგნალზე იძლევა მაღალი სიზუსტის აღდგენილ სიგნალს მინიმალური დამახინჯებით.

- თანამედროვე ტექნოლოგიური პროგრესის ფონზე, როდესაც სიგნალების მოცულობა და კომპლექსურობა იზრდება, ციფრული დამუშავების მეთოდების დახვეწა და მათი ოპტიმიზაცია ხდება კიდევ უფრო საჭირო.

- კვლევამ ნათლად წარმოაჩინა არსებული ქართული და საერთაშორისო ლიტერატურის დეფიციტი ციფრული სიგნალის დამუშავების სფეროში, რაც კიდევ უფრო მეტ მნიშვნელობას ანიჭებს აღნიშნულ ნაშრომს.

- კვლევამ შესაძლებლობა მისცა ავტორს შეიმუშაოს პრაგმატული მიდგომა, რომელიც აერთიანებს თეორიულ ანალიზს, მათემატიკურ სიმულაციასა და პრაქტიკულ შედეგებს.

- ჩატარებული რიცხვითი ექსპერიმენტები დაფუძნებულია ციფრული დამუშავების სტანდარტულ, ხოლო ზოგ შემთხვევაში ინოვაციურ მიდგომებზე, რაც მათ პრაქტიკულად გამოყენებადს ხდის როგორც კვლევით, ისე ინდუსტრიალურ სფეროებში.

- სხვადასხვა ტიპის სიგნალებზე ჩატარებულმა მოდელირებამ აჩვენა, რომ ზოგიერთ შემთხვევაში საჭიროა ადაპტური ფილტრების გამოყენება, რაც შეიძლება მომავალი კვლევის საგნად იქცეს.

- დისერტაციაში გამოყენებული კვლევითი მიდგომა საფუძვლად შეიძლება დაედოს ციფრული სიგნალის დამუშავების ქართული პრაქტიკის ჩამოყალიბებასა და სწავლების გაძლიერებას უმაღლეს საგანმანათლებლო დაწესებულებებში.

რეკომენდაციები

- რეკომენდებულია, კვლევა გაგრძელდეს ადაპტური ფილტრების მიმართულებით, რომლებიც შეძლებენ სიგნალის მახასიათებლების დინამიკურ ცვლილებაზე რეაგირებას.

- მიზანშეწონილია პროგრამული ალგორითმების ოპტიმიზაცია რეალურ დროში მოქმედი სისტემებისთვის, რათა სიგნალის აღდგენის პროცესი შესაძლებელი გახდეს მინიმალური დაყოვნებით.

- შესაძლებელია ციფრული სიგნალის დამუშავების ალგორითმების ინტეგრირება ტექსტის ამომცნობ სისტემებში, რათა გაიზარდოს ქართული ენის ავტომატური ამომცნობის სიზუსტე.

- რეკომენდებულია კვლევის შედეგების პრაქტიკული დანერგვა ბიომედიცინაში, მაგალითად, ECG ან EEG სიგნალების ფარდობითი ანალიზისთვის.

- სასურველია, შეიქმნას ქართულენოვანი სახელმძღვანელო, რომელიც საფუძვლად დაედება ციფრული სიგნალის დამუშავების სწავლებას და ხელს შეუწყობს დარგის პოპულარიზაციას საქართველოში.

- მიზანშეწონილია MATLAB-ის გამოყენების პარალელურად სხვა პროგრამული პლატფორმების (Python, Octave, Julia) შესწავლა და შედარებითი ანალიზი, რათა გაფართოვდეს ტექნიკური კომპეტენციები და სისტემური მოქნილობა.

- საჭიროა არსებული ალგორითმების რეალურ დროში ტესტირება მიკროკონტროლერებზე ან FPGA-ზე, რაც ხელს შეუწყობს ინტეგრაციას ინტელექტუალურ სისტემებში.
- მნიშვნელოვანია ციფრული ფილტრების განვითარება დაბალი ენერგომომხმარების პირობებში, რათა მათი გამოყენება შესაძლებელი იყოს პორტატულ და ენერგოეფექტიან მოწყობილობებში.

გამოყენებული ლიტერატურა:

1. ბერიძე ლ., (2014). MATLAB-სტუდენტებისთვის, სტუ
2. ლაბაძე ო., სიგნალების თეორია, (2012). სიგანელბის თეორია, თსუ
3. რობიტაშვილი ა., მურჯიკნელი გ., ვეკუა თ. (2009). ტელეკომუნიკაციის თეორია. ტექნიკური უნივერსიტეტი.
<https://issuu.com/georgianpolitics3/docs/2e4a24d38ec50e>
4. ფაღავა, ა. (2011). ბიოსიგნალის დამუშავების საფუძვლები.
https://gtu.ge/book/DSP_Analog.pdf
5. ლომიძე, ნ., ლომიძე, ხ. (2022). რადიოფიზიკა.
<https://rustaveli.org.ge/geo/200916031925tsignebi/radiofizika>
6. ყუბანიშვილი, ე. (2013). ბიოსიგნალების ციფრული დამუშავება.
https://gtu.ge/book/ims/cifruli_damushaveba.pdf
7. სალექციო კურსი, (2022). მატლაბის სალექციო კურსი.
თსუ <https://example.com/tsu-matlab-course>
8. სალექციო კურსი, (2015). ციფრული და ანალოგური ელექტრონიკის საწყისები.
ტექნიკური უნივერსიტეტი
9. დიხამინჯია ნ., ვეშაპიძე გ. (2019). სიგნალის ეკვალიზაცია. ილიას სახელმწიფო უნივერსიტეტი. <https://slideplayer.com/slide/16954800/>
10. დიხამინჯია ნ., ვეშაპიძე გ. (2019). მრავალდონიანი სიგნალი. ილიას სახელმწიფო უნივერსიტეტი. <https://slideplayer.com/slide/15424650/>
11. დიხამინჯია ნ., ვეშაპიძე გ. (2019). ცალმხრივი და დიფერენციალური სიგნალი. ილიას სახელმწიფო უნივერსიტეტი. <https://slideplayer.com/slide/17193412/>

12. Oppenheim, A. V., & Schaffer, R. W. (2009). Discrete-Time Signal Processing (3rd ed.). Prentice Hall.
13. Proakis, J. G., & Manolakis, D. G. (2006). Digital Signal Processing: Principles, Algorithms, and Applications (4th ed.). Pearson.
14. Ifeachor, E. C., & Jervis, B. W. Digital Signal Processing: A Practical Approach.
15. Baraniuk, R. G. Understanding Digital Signal Processing.
16. Ingle, V. K., & Proakis, J. G. (2015) Digital Signal Processing Using MATLAB.
17. Haykin, S. Adaptive Signal Processing.
18. Hayes, M. H. Statistical Digital Signal Processing and Modeling.
19. Smith, S. W. The Scientist and Engineer's Guide to Digital Signal Processing.
20. Papoulis, A. (1984). Signal Analysis. McGraw-Hill.
21. Mitra, S. K. (2010). Digital Signal Processing: A Computer-Based Approach (4th ed.). McGraw-Hill.
22. MathWorks-ის მომხმარებლის სახელმძღვანელო – Signal Processing Toolbox User's Guide.
23. Pickus, K. (2006). Book-review. AJS Review, 30(1), 206-208.
doi:10.1017/S0364009406290093
24. Parker, M. (2017). Finite Impulse Response (FIR) Filters. Digital Signal Processing 101 (2nd ed.). Newnes. <https://doi.org/10.1016/B978-0-12-811453-7.00005-6>
25. Lacoste, R. (2010). No Fear with FIR: Put a Finite Impulse Response Filter to Work. The Darker Side. Newnes. <https://doi.org/10.1016/B978-1-85617-762-7.00007-1>
26. Raffoul, Y. N. (2025). Z-transform. Difference Equations and Applications. Academic Press. <https://doi.org/10.1016/B978-0-44-331492-6.00009-1>
27. Akan, A., & Chaparro, L. F. (2022). The Z-transform. Signals and Systems Using MATLAB (4th ed.). Academic Press. <https://doi.org/10.1016/B978-0-44-315709-7.00019-7>
28. Semmlow, J. (2022). Applications and the Z-transform. Circuits, Signals, and Systems for Bioengineers (4th ed.). Academic Press. <https://doi.org/10.1016/B978-0-443-15886-5.00008-3>

29. Li, C. et al. (2024). Periodic signal extraction of GNSS height time series. *Geodesy and Geodynamics*. <https://doi.org/10.1016/j.geog.2023.04.003>
30. Ghil, M., & Taricco, C. (1997). *Advanced spectral-analysis methods*. In IOS Press.
31. Golyandina, N. (2010). On the choice of parameters in singular spectrum analysis. arXiv:1005.4374.
32. Abualigah, L. et al. (2021). The Arithmetic Optimization Algorithm. *Computers & Mathematics*. <https://doi.org/10.1016/j.cma.2020.113609>
33. Hinich, M. J., & Wild, P. (2005). Coherence spectrum in noise. *Signal Processing*. <https://doi.org/10.1016/j.sigpro.2005.02.008>
34. Ma, J. et al. (2024). Spectrum analysis in fractional Fourier domain. *Signal Processing*. <https://doi.org/10.1016/j.sigpro.2024.109539>
35. Xu, L. et al. (2016). Randomized nonuniform sampling. *Signal Processing*. <https://doi.org/10.1016/j.sigpro.2015.09.016>
36. Su, X. et al. (2019). Discrete fractional Fourier transforms. *Signal Processing*. <https://doi.org/10.1016/j.sigpro.2019.01.019>
37. Wei, D. et al. (2023). Sampling of graph signals. *Digital Signal Processing*. <https://doi.org/10.1016/j.dsp.2023.103970>
38. Maronidis, A. et al. (2018). Differential sparsity for compressed signals. *Digital Signal Processing*. <https://doi.org/10.1016/j.dsp.2017.11.016>
39. Chen, Q. et al. (2008). Non-bandlimited signals. *Computers & Mathematics with Applications*. <https://doi.org/10.1016/j.camwa.2008.03.021>
40. Egila, M. G. et al. (2016). ECG analysis with FIR filter. *Journal of Electrical Systems*. <https://doi.org/10.1016/j.jesit.2015.07.001>
41. Wang, J. et al. (2023). Sensor fusion noise suppression. *Measurement*. <https://doi.org/10.1016/j.measurement.2024.114680>
42. Qiu, R. et al. (2023). FIR filter design for SFQ. *Superconductivity*. <https://doi.org/10.1016/j.supcon.2023.100045>
43. Piestrak, S. J., & Patronik, P. (2015). Fault-tolerant FIR filters. *IEEE NORCHIP*. doi:10.1109/NORCHIP.2015.7364382

44. Wen, J. W., & Liu, F. (2011). FIR Filtering for Markov Systems. *Circuits Syst Signal Process*, 30, 1149–1164. <https://doi.org/10.1007/s00034-011-9291-z>
45. Chu, X. et al. (2021). Statistical Eye Diagram Estimation. *IEEE TVLSI*, 29(7), 1370–1378. doi:10.1109/TVLSI.2021.3082208
46. Chauhan, S. et al. (2023). Optimal digital IIR filter design. *Eng. Appl. Artif. Intell.* <https://doi.org/10.1016/j.engappai.2022.105803>
47. Chauhan, S. et al. (2023). Optimal IIR filters in AI. <https://www.sciencedirect.com/science/article/abs/pii/S095219762200793X>
48. Agrawal, N. et al. (2021). Digital IIR filters survey. *Applied Acoustics*. <https://doi.org/10.1016/j.apacoust.2020.107669>
49. Roonizi, A. K. (2023). Kalman-based IIR design. *Signal Processing*. <https://doi.org/10.1016/j.sigpro.2023.108958>
50. Rudin, L. I. et al. (1992). Nonlinear noise removal. *Physica D*. [https://doi.org/10.1016/0167-2789\(92\)90242-F](https://doi.org/10.1016/0167-2789(92)90242-F)
51. Wang, Z. et al. (2021). Adaptive filter with variable step. arXiv:2112.04936. <https://arxiv.org/abs/2112.04936>
52. Analog Devices. AD711 Product Datasheet. <https://fr.ovaga.com/products/detail/ad711>
53. Tadeusiewicz, R., & Ogiela, M. R. (2022). Cognitive Systems in Medicine. *Annals of CSIS*, 32. https://annals-csis.org/Volume_32/pliks/communication.pdf
54. Turnitin API System – PaperInfo API. <https://api.turnitin.com/paperInfo.asp>
55. Ossman, K. A. K. (2017). *Introduction to DSP Using MATLAB®*. University of Cincinnati.
56. Doroslovacki, M. (Ed.). (2018). *Signal Processing and Machine Learning Theory*. Springer. <https://link.springer.com/book/10.1007/978-3-319-96941-1>
57. Maria G, (2022), Communication Papers of the 17th Conference on Computer Science and Intelligence Systems, *Annals of Computer Science and Information Systems* Volume 32, Bulgaria

58. Tabeshadze, J. (2025) Design and Analysis of a Low-Pass FIR Filter Based on Window Functions Using MATLAB Simulation. <http://orcid.org/0009-0008-7463-7801><http://www.sciencejournals.ge/index.php/NJ/article/view/708>
59. Tabeshadze, J. Wediakovi D. (2019), ციფრული სიგნალის დამუშავების წინასწარი მიწოდებისა და გადაწყვეტილებაზე რეაგირების ეკვალიზატორები, 178–185. [http://gtu.ge/Journals/mas/Referat/ASU-2019\(1_28\)/178_185.pdf](http://gtu.ge/Journals/mas/Referat/ASU-2019(1_28)/178_185.pdf)
60. ვედიაკოვი, დ., ტაბეშაძე, ჯ., წიკლაური, მ. (2019). ჰილბერტის დისკრეტული გარდაქმნის (DHT) საფუძველზე კაუხალობის განხორციელების ალგორითმი. 1(28), 81–87. [https://gtu.ge/Journals/mas/Referat/ASU-2019\(1_28\)/81_87.pdf](https://gtu.ge/Journals/mas/Referat/ASU-2019(1_28)/81_87.pdf)

გამოყენებული სურათების/დიაგრამების/გრაფიკების ნუსხა:

დიაგრამა 1. დეტერმინირებული ისგნალების კლასიფიკაცია (ყუბანეიშვილი 2013)	41
დიაგრამა 2. ანალოგური სიგნალის დამუშავების ალგორითმი (Semmlow 2022, 3)69
დიაგრამა 3. სიგნალების ციფრული დამუშავების განხორციელებული სტრუქტურული სქემა (Tabeshadze 2019)69
დიაგრამა 4. დისკრეტული სისტემის ალგორითმი (წყარო-ავტორი)94
ცხრილი 1. სინუსოიდური სიგნალის ანათვლები $[0; 2]$ წმ ინტერვალზე (წყარო-ავტორი)45
ცხრილი 2. სიგნალები და სიგნალთა გაშლა ფურიეს მწკრივებად (Parker 2017, 607)	: .57
ცხრილი 3. ფურიეს გარდაქმნები (Parker 2017, 650)63
ცხრილი 4. ფურიეს გარდაქმნის თვისებები (Parker 2017, 652-653)64
ცხრილი 5. ბიტანი ციფრული კოდი მითითებულ დროებზე (წყარო-ავტორი)73
ცხრილი 6. ტერმინოლოგია და ფორმულები, სიგნალების დამუშავებისას (Ingle 2015, 79)74
ცხრილი 7. Z გარდაქმნები (Haykin 2010, 238-240)99
ცხრილი 8. Z გარდაქმნის თვისებები (Haykin 2010, 278)100

ცხრილი 9. ფილტრაციის პროცესი, 7×7 გამოსახულების მატრიცის, სადაც ფილტრი მოთავსებულია ცენტრში პიქსელზე ინდექსით (3,4) (წყარო-ავტორი).....	154
სურათი 1. ანალოგური ანალოგური სინუსოიდური სიგნალის ვიზუალიზაციას MATLAB-ში. სიგნალის ამპლიტუდა არის 5 (რაც გულისხმობს, რომ მისი მნიშვნელობა მერყეობს -5-დან +5-მდე). მოიცავს დაახლოებით 2 სრულ ციკლს 0-დან 10 წმ-მდე დროის მონაკვეთში. (წყარო-ავტორი) 43	43
სურათი 2. დისკრეტული სინუსოიდური სიგნალის ვიზუალიზაციას MATLAB-ში. წინა ანალოგური სიგნალისგან განსხვავებით, ეს გრაფიკი აჩვენებს სიგნალის მნიშვნელობებს მხოლოდ გარკვეულ, დისკრეტულ დროის მომენტებში, რაც წარმოდგენილია ინდივიდუალური წერტილებით და არა უწყვეტი ხაზით. (წყარო-ავტორი)	44
სურათი 3. ორი ანალოგური სინუსოიდური სიგნალის, y_1 (ლურჯი) და y_2 (წითელი), ერთობლივ ვიზუალიზაციას MATLAB-ში. (წყარო-ავტორი).....	47
სურათი 4. ანალოგური სიგნალის ვიზუალიზაციას MATLAB-ში, რომელიც მიღებულია ორი სხვადასხვა სიხშირის სინუსოიდური სიგნალის შეკრებით. ესაა $\sin(2\pi t)$ და $\sin(2\pi 4t)$ ფუნქციების ჯამი. (წყარო-ავტორი).....	48
სურათი 5. ანალოგური სიგნალის ვიზუალიზაციას MATLAB-ში, რომელიც მიღებულია ორი სხვადასხვა სიხშირის სინუსოიდური სიგნალის შეკრებით. ესაა $\sin(2\pi t)$ და $\sin(2\pi 1.5t)$ ფუნქციების ჯამი. (წყარო-ავტორი).....	49
სურათი 6. რთული ანალოგური სიგნალის ვიზუალიზაცია MATLAB-ში, რომელიც მიღებულია ორი სინუსოიდური სიგნალის ერთმანეთზე გამრავლებით. ესაა $\sin(2\pi t)$ და $\sin(2\pi 4t)$ ფუნქციების ნამრავლი. (წყარო-ავტორი).....	50
სურათი 7. ოთხი განსხვავებული (მართკუთხა, სამკუთხა, 2 ხერხისებური) ანალოგური პერიოდული სიგნალის ვიზუალიზაციას MATLAB-ში. (წყარო-ავტორი)	51
სურათი 8. ანალოგური არაპერიოდული სიგნალების ვიზუალიზაცია MATLAB-ში. (წყარო-ავტორი)	52

სურათი 9. ორი სხვადასხვა სინუსოიდური სიგნალის სიხშირული სპექტრის ვიზუალიზაცია. თითოეული ქვეგრაფიკი აჩვენებს სიგნალის ამპლიტუდას სიხშირული არეში (ჰერცებში).	53
სურათი 10. სიგნალების წრფივი კომბინაციის სპექტრი.....	54
სურათი 11. სინუსოიდური სიგნალების ნამრავლის სპექტრი	54
სურათი 12. ანალოგური მართკუთხა ტალღის ფორმის ვიზუალურ წარმოდგენა, მითითებულია მისი ამპლიტუდა და პერიოდი	58
სურათი 13. მართკუთხა ტალღის ფურიეს მწკრივად გაშლის ვიზუალიზაცია. იგი აჩვენებს, თუ როგორ შეიძლება კომპლექსური (არასინუსოიდური) პერიოდული სიგნალი წარმოდგენილი იქნას სხვადასხვა სიხშირისა და ამპლიტუდის სინუსოიდური ტალღების (ჰარმონიკების) ჯამის სახით.....	59
სურათი 14. დროითი სიგნალი და სიგნალის სპექტრი	61
სურათი 15. ორი განსხვავებული სიხშირის სიგნალის სიხშირის დიაპაზონი.	64
სურათი 16. სპექტროგრამა, რომელიც აჩვენებს სიხშირული კომპონენტების ცვლილებას დროში.	67
სურათი 17. ცელნამგალა ვემაპის ხმის სპექტროგრამა.....	68
სურათი 18. ანალოგური სიგნალიდან ანათვლების აღება.....	73
სურათი 19. ზედდების ეფექტი	78
სურათი 20. ზედდების ეფექტი სინუსოიდურ ტალღებზე.....	80
სურათი 21. ზედდების ეფექტი სიგნალების შეკრების დროს.....	81
სურათი 22. დაბალსიხშირული ფილტრის სიხშირული მახასიათებლები.....	84
სურათი 23. დაბალსიხშირული ფილტრებს იმპულსური მახასიათებლები	87
სურათი 24. შემავალი სიგნალის შესაბამისობა ADC -სთან	89
სურათი 25. გრაფიკი წარმოადგენს ორი განსხვავებული ინტერპოლაციის მეთოდის ვიზუალიზაციას, რომლებიც გამოიყენება დისკრეტული სიგნალიდან საწყისი უწყვეტი სიგნალის აღსადგენად.....	91
სურათი 26. სინუსოიდური ინტერპოლაციის ფილტრით მიღებული სიგნალი.....	92
სურათი 27. კოსინუსოიდური ინტერპოლაციის ფილტრით მიღებული სიგნალი	93

სურათი 28. გრაფიკი წარმოადგენს ტიპური დაბალგამტარი ფილტრის (Low-Pass Filter) ამპლიტუდურ-სიხშირულ მახასიათებელს. აჩვენებს, თუ როგორ იცვლება ფილტრის გამტარობა (ამპლიტუდა) სიხშირის მიხედვით.....	110
სურათი 29. გრაფიკი წარმოადგენს ტიპური მაღალგამტარი ფილტრის (High-Pass Filter) ამპლიტუდურ-სიხშირულ მახასიათებელს. ის აჩვენებს, თუ როგორ იცვლება ფილტრის გამტარობა (ამპლიტუდა) სიხშირის მიხედვით.....	111
სურათი 30. გრაფიკი წარმოადგენს ტიპური ზოლგამტარი ფილტრის (Band-Pass Filter) ამპლიტუდურ-სიხშირულ მახასიათებელს. იგი აჩვენებს, თუ როგორ იცვლება ფილტრის გამტარობა (ამპლიტუდა) სიხშირის მიხედვით.....	112
სურათი 31. ჩამჭრელი ფილტრი (Notch Filter) ზოლური ფილტრის ნაირახეობა.....	112
სურათი 32. იდეალური დაბალგამტარული ციფრული ფილტრის ამპლიტუდისა და ფაზის რეაქცია.....	116
სურათი 33. იდეალური დაბალსიხშირული ფილტრის კოეფიციენტები ($f_c = F_s/4$)....	118
სურათი 34. შემოკლებული ფილტრის კოეფიციენტები	119
სურათი 35. ფილტრის კოეფიციენტები შემოკლებულია და გადატანილია მარჯვნივ 10 ერთეულით.	119
სურათი 36. სიმ ფილტრის სიხშირული მახასიათებლები.....	122
სურათი 37. ორი ფილტრის მაგნიტუდის რეაქციები.	124
სურათი 38. სხვადასხვა ფანჯრის ფუნქციის მაგალითები	125
სურათი 39. გრაფიკი წარმოადგენს იდეალური დაბალგამტარი ფილტრის (Ideal Low-Pass Filter) სიხშირული მახასიათებლების ორ კომპონენტს: ამპლიტუდის სპექტრს და ფაზურ სპექტრს.....	128
სურათი 40. სურათზე ნაჩვენებია, როგორ ხდება მაღალსიხშირული, ზოლური და ჩამჭრელი ფილტრების გადაყვანა "ეკვივალენტურ" დაბალსიხშირულ ფილტრებში.	130
სურათი 41. მაღალსიხშირული ფილტრი	132
სურათი 42. 50-ე რიგის ჩამჭრელი ფილტრი, რომელის ცენტრალური სიხშირე, 10 კჰც და ზოლური სიგანე 4 კჰც	133

სურათი 43. გ) 50-ე რიგის ზოლური ფილტრი, რომელის ცენტრალური სიხშირე, 10 კჰც და ზოლური სიგანე 4 კჰც	134
სურათი 44. ფილტრის ამპლიტუდური რეაქცია	146
სურათი 45. სტაბილური ფილტრის პოლუსები და ნულები	147
სურათი 46. გადიდებული ციფრული გამოსახულება.....	150
სურათი 47. ქართულ ანბანზე დაყრდნობით, წინასწარ შექმნილი მატრიცა,	158
სურათი 48. „ა“ ასო-სიმბოლო გამოსახულია მატრიცის სახით	158

მნიშვნელოვანი განმარტებები:

- **ჰერცი** (აღნიშვნისთვის იყენებენ: ჰც, Hz) წარმოადგენს რხევის სიხშირის საზომ ერთეულს ერთეულთა საერთაშორისო სისტემაში. მისი სახელი მომდინარეობს ფიზიკოს ჰაინრიხ ჰერცისგან. 1 ჰერცი შეესაბამება პერიოდული რხევის სიხშირეს, რომლის დროსაც ერთი ციკლი სრულდება 1 წამის განმავლობაში. ასევე გამოიყენება ჰერცის ერთეულები, როგორცაა კილოჰერცი (10^3 ჰც), მეგაჰერცი (10^6 ჰც) და სხვა.
- **იმპულსი** – ეს არის სიგნალი, რომელიც არსებობს მხოლოდ გარკვეული დროის მონაკვეთზე, მაგალითად ვიდეოიმპულსი და რადიოიმპულსი.
- **ბესელის ფუნქციები** – ცილინდრული ფუნქციები. ფუნქციათა მნიშვნელოვანი კლასი, რომლებიც წარმოადგენენ ბესელის დიფერენციალური $x^2 y'' + xy' + (x^2 - p^2)y = 0$; $p = \text{const}$ განტოლების ამონახსნებს. ეს დიფერენციალური განტოლებები გვხვდება მათემატიკური ფიზიკის ამოცანების ფურცელს მეთოდით ამოხსნისას. ამ განტოლების ამონახსნს აქვს ასეთი სახე: $J_p(x) = \sum_{k=0}^{\infty} (-1)^k (x/2)^{p+2k} / [k! \Gamma(k + p + 1)]$, სადაც სადაც P - პარამეტრია, Γ - გამა-ფუნქცია. ამ ამონახსნს პირველი გვარის p რიგის ბესელის ფუნქციას უწოდებენ. ბესელის ფუნქციები დაწვრილებით შესწავლილია როგორც ნამდვილი, ასევე კომპლექსური რიცხვებისათვის. (ბესელის ფუნქციები,“ ვიკისიტიყვის ქართული ვერსია, ეროვნული ბიბლიოთეკა, წვდომა 20 ივლისი, 2025, https://www.nplg.gov.ge/wikidict/index.php/ბესელის_ფუნქციები.)
- **sinc ფუნქცია** არის მათემატიკური ფუნქცია, რომელიც განისაზღვრება შემდეგნაირად:

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$$

როდესაც $x \neq 0$, და $\text{sinc}(0) = 1$. ეს ფუნქცია მნიშვნელოვანია სიგნალის დამუშავებაში, განსაკუთრებით ფურიეს გარდაქმნებთან და ფილტრის დიზაინთან მიმართებაში. *sinc* ფუნქცია იდეალური დაბალსიხშირული ფილტრის იმპულსური რეაქციაა და ხშირად გამოიყენება, რათა წარმოჩინდეს სიგნალები, რომლებიც კონკრეტული სიხშირის დიაპაზონშია შეზღუდული.

აბრევიატურა

ADC - Analog-to-Digital Conversion (ანალოგურ-ციფრული გარდაქმნა)

DAC - Digital-to-Analog Conversion (ციფრულ-ანალოგური გარდაქმნა)

DFT - Discrete Fourier Transform (დისკრეტული ფურიეს გარდაქმნა)

DSP - Digital Signal Processing (ციფრული სიგნალის დამუშავება)

ECG - Electrocardiogram (ელექტროკარდიოგრამა)

EEG - Electroencephalogram (ელექტროენცეფალოგრამა)

EKG - Electrocardiogram (სინონიმი ECG-სთვის)

FFT - Fast Fourier Transform (სწრაფი ფურიეს გარდაქმნა)

FIR - Finite Impulse Response (სასრული იმპულსური მახასიათებლის ფილტრი)

IIR - Infinite Impulse Response (უსასრულო იმპულსური მახასიათებლის ფილტრი)

OCR - Optical Character Recognition (ოპტიკური ამომცნობი სისტემა)

სცდ - სიგნალის ციფრული დამუშავება

სიმ - სასრული იმპულსური მახასიათებელი

უიმ - უსასრულო იმპულსური მახასიათებელი

IEEE - Institute of Electrical and Electronics Engineers

dB (დბ) - Decibel (დეციბელი)