ӘЛ-ФАРАБИ АТЫНДАҒЫ ҚАЗАҚ ҰЛТТЫҚ УНИВЕРСИТЕТІ

Т. Ю. Гревцева, Е.Т. Кожагулов, С.А. Хохлов

«СИГНАЛДАРДЫ САНДЫҚ ӨҢДЕУ» КУРСЫ БОЙЫНША ЗЕРТХАНАЛЫҚ ЖҰМЫСТАР

Алматы
«Қазақ университеті»
2018

ӘОЖ 681.06 (075.8)
КБЖ 32.973

Баспаға әл-Фараби атындағы Қазақ ұлттық университеті физика-техникалық факультетінің Ғалыми кеңесі және Редакциялық-баспа кеңесі шешімімен ұсынылған

П і к і р ж а з ғ а н:

физика-математика ғылымдарының докторы, профессор
З.Ж. Жаңабаев

Гревцева Т.Ю., Кожагулов Е.Т., Хохлов С.А. «Сигналдарды сандық өңдеу» курсы бойынша зертханалық жұмыстар / Алматы: Қазақ университеті, 2018. – 156 б.

Оқу-әдістемелік құрал «5B071900 – Радиотехника, электроника және телекоммуникациялар» мамандығы бойынша физика-техникалық факультетінің 2 курс студенттеріне арналған «Сигналдарды сандық өңдеу» бағдарламалық курсына сәйкес құрастырылған. Оқу-әдістемелік құралда зертханалық жұмыстардың сипаттамасы келтірілген, орындау барысында сигналдарды өңдеу әдістерінің (корреляциялық, спектрлік, вейвлетті, ақпараттық-энтропиялық, фракталдық және т.б. әдістердің) теориялық негіздерін меңгеруге, сондай-ақ Matlab ортасында компьютерлік модельдеу көмегімен олардың іске асырылуын үйренуге мүмкіндік береді.

МАЗМҰНЫ

КІРІСПЕ

Сигналдарды сандық өңдеу (ССӨ) ғылым мен техниканың жеке бағыты ретінде өткен ғасырдың 50-ші жылдарында пайда болды. Бастапқыда сигналдарды сандық өңдеу радиоэлектрониканың бір саласы болып қарастырылды, бірақ электроникада жеткен үлкен жетістіктердің нәтижесінде сигналдарды сандық өңдеу жүйелері біздің күнделікті өмірімізге CD және DVD ойнатқыштар, модемдер, ұялы телефондар және басқада көптеген құрылғылар түрінде енді.

Сигналдарды сандық талдаудың дамуын бірнеше негізгі кезеңдерге бөлсе болады.

Бірінші кезең. Сандық сүзгілеу және спектрлік талдау. Дамудың осы кезеңінде (1965 – 1975 жж.) ССӨ теориясының негізгі пәндік саласы – сандық сүзгілеу мен спектрлік талдау болды. Дамушы бағыттардың жалпы негізі – іріктелген жиілікті сандық сүзгілерді синтездеу болды. Сол уақытта белгілі машиналық алгоритімдердің жиынын (жылдам Фурье түрлендіру алгоритмін) қолдану арқылы ССӨ теориясының негізгі ережелері іс жүзінде дискретті жүйелер мен тізбектер теориясына негізделді және зерттелді.

Екінші кезең. Көпжылдамдықты сүзгілеу және сигналдарды бейімді өңдеу. 70-ші жылдардың басында бірқристалды микропроцессорлар шыға бастады. ССӨ теориясы 1975-1985 жылдар аралығында өзінің кезекті даму кезеңіне өтті. Дәл осы кезеңде заманауи ССӨ теориясының бір-бірімен байланысқан төрт негізгі бағыты қалыптасады. Бірінші бағыт – сигналдардың сандық жиілікті іріктемесі. Осы бағытқа байланысты неғұрлым айтарлықтай ерекшеленген жұмыстар, сигналдарды жылдамдығың ауқымды түрде өңдейтің теориямен байланысты, я олардың жиілігі мен сирету уақыты бойынша негізге алынды. Екінші бағыт – сигналдарды өңдеудің жылдам алгоритмі, жоғары жылдамдықтағы алгоримтдердің құрылуына байланысты ССӨ-нің енуі, демек «артық» операцияларды алып тастау жолы. Осында өңдеу және күрделі көбейту алгоритмдерді басқа операциялармен алмастыру, оларға қосу және ығысу қолданылады. Үшінші бағыт – адаптивті және оңтайлы түрде сигналдарды өңдеу, оңтайлы іріктеу және зерттелетін динамикалық үрдістің табиғаты туралы априорлы белгісіздік жағдайында сигналдарды өңдеу мәселелерін шешудің кең ауқымын қамтиды. Төртінші бағыт – көп өлшемді сигналдар мен өрістерді өңдеу, көпөлшемді сандық жүйелердің бір өлшемді сигналдарды өңдеудің табиғи дамуы болып табылады. 1970-жылдары сандық бейнелеуді өңдеу мен дыбыс ақпараттары табысты дами бастады, олар ғылым мен техниканың тәуелсіз саласы болды. Сандық түрде бейнені өңдеу, аналогты әдістен икемділігі және тиімділігі жағынан асып түседі.

Дыбысты түрде келген ақпаратты өңдеудің сандық әдістері сөйлеуді өңдеуге және кодтауға, сондай-ақ кең жолақты аудио сигналдарды тиімді түрлендіруде кеңінен қолданылады.

Үшінші кезең. Сигналды процессорларда оңтайлы жобалау. 80-ші жылдардың бірінші жартысында NEC (Жапония) фирмасы, содан Texas Instruments (АҚШ) фирмасы алғашқы сигналдық mPD7720 және TMS32010 процессорлардың өндірістік шығарылымын жариялады, осылайша технологиядағы жаңа дәуірдің ашылуын атап өтті. Микропроцессорлық жүйелердің жаңа класы, шын мәнінде біркристалды микроЭВМ отбасында болды, демек, ССӨ-нің классикалық алгоритмдері бағдарламалық және аппараттық қамтамасыз етудің жоғары тиімділігіне арналған ішкі архитектура. Салыстырмалы түрде, қысқа уақыт мезетінде – 15 жыл – сандық сигналдарды өңдеу процессорлары дамудың бірнеше сатысынан өтті. Интенсивті дамудың арқасында, есептеу жағынан өнімділік және бір кристалды сингалды сандық түрде өңдейтің процессорлардың ішкі ресурстары сенімді түрде қалыптасып өсті, осыған орай ССӨ-ні түсінетін микропроцессорлық жүйелеріне қуатты бағдарламалық және аппараттық құрал-жабдықтар шықты.

Төртінші кезең. Алғашқы біркристалды көп процессорлы жүйелер және логикалық оңтайлы сызбаларда бағдарланатын логикалық интегралды схемалар (БЛИС) құрылған. Әдістердің дамуы және сигналдарды өңдеу әдісінің дамуының этапы, 90-шы жылдың екінші жартысында жаңа көппроцессорлы технологиялардың бірегей мүмкіндіктері ретінде сипатталады, сонымен бірге БЛИС-тің қолдануы жатады.

ССӨ-нің одан әрі дамуы микропроцессорлық құралдың дамуымен тікелей байланысты, соған орай жаңасын ойлап шығару және сигналды өңдеу әдісін одан ары тиімділеу.

Компьютерлік үлгілеу әдісін пайдаланбай, сигналды өңдеуді жүзеге асыру мүмкін емес. Бұл мәселелерді шешудің ең тиімді әдістерінің бірі – Matlab компьютерінің ортасын қолдану. Matlab аясының негізгі кітапханасы, сонымен қатар оған кіретін Signal Processing, Filter Design, Communications, Fixed-Point қосымша пакеттері және басқалары көптеген функцияларды қамтиды. Соның көмегімен сандық сигналдарды өңдеу түрлі есептерді тездетіп шығаруға көмектеседі.

MATLAB ОРТАСЫНДА ЖҰМЫС ЖАСАУ НЕГІЗДЕРІ

Matlab компьютерлік ортасы MathWorks компаниясымен құрылды және қазіргі таңда адам қызметінің түрлі салаларында туындайтын есептерге байланысты шешуші аспап болып табылады және де ол қуатты және әмбебап болып келеді. Matlab арқылы зерттеуге болатын бірқатар мәселелерге ССӨ-нің матрицалық талдауынан басқа, математикалық физиканың есептері, оңтайдырылған есептер, ақпараттарды өңдеу және бейнелеу, картографиялық белгілермен жұмыстар, нейронды желілердің параметрлерін есептеу және т.б. жатады.

Matlab 2D және 3D деректерін бейнелеу үшін тамаша мүмкіндіктерге ие. График редакторы нәтижені қажетті жолмен шығаруға мүмкіндік береді: көрсеткілерді, түсіндірме жазбаларды, түзулерді, түзулердің түсін өзгертуге, яғни, баяндамаға немесе мақалаға орналастыру үшін жарамды сурет шығаруға көмектеседі.

Matlab теңдеулерді шешу үшін классикалық сандық алгоритмдерді қамтиды, керекті интегралдардың мәндерін табуға, интерполяциялауға, жүйені және дифференциалдық теңдеулерді шешуге мүмкіндік береді. Қарапайым енгізілген бағдарламалау тілі өзіңіздің бағдарламаларыңызды оңай жасауға мүмкіндік береді. Matlab – аудармашы (интерпретатор), яғни, бағдарламаның әрбір жолы кодқа айналады, содан кейін орындалады.

Matlab аясында жұмыс істеу негіздерін қарастырайық.

Matlab-тың жұмыс аясы

Matlab терезесі келесі негізгі элементтерден тұрады:

a) мәзір;
b) құрыл-сайман тақтасы;
c) Command History және Current Directory терезесімен жабдықталған, олардың көмегімен алдын-ала енгізілген командаларды қарауға және қайта шақыруға болады, сондай-ақ ағымдағы каталогты орнату үшін арналған;
d) командалық терезе;
e) күй жолағы;

Matlab аясында қарапайым арифметикасы есептеулер

Командалық жолда 1+2 теріп, <Enter> батырмасын басыңыз. Matlab бағларламасы не істеді? Біріншіден ол 1+2-нің суммасын есептеп, содан кейін шыққан нәтижені арнайы ans айнымалысына көшіріп және нәтижені командалық терезеге шығарды.

Ал егер алдыңғы есеппен жұмысты жалғастырғыңыз келсе, мысалы, (1+2)/4.5 сомасын есептеу үшін, командалы жолда ans/4.5 осындай сөйлемді теріп <Enter> батырмасын басуға болады. Осыдан

```
>> ans =
              0.6667
>>
```

Matlab аясында арифметикалық операциялар әдеттегі тәртіппен орындалады: дәрежеге шығару (^); көбейту және бөлу (* және /); қосу және азайту (+ және -). Жақша – арифметикалық операциялардың орындалатын тәртібін өзгерту үшін пайдаланылады.

Мысалы, $\frac{3(x-4)}{2}$ өрнегінің мәнін табу керек, $x=9$ кезінде.

Берілген оңай математикалық операцияны шешу үшін, біріншіден x айнымалысына 9 орнату керек, содан өрнектің мәнін есептейміз. Ыңғайлы болу үшін, командаларға арналған түсініктемелерді ескертулер түрінде жазуға болады, олар орындалмайтын символдар қатары болып табылады. Matlab-та ескертулер % белгісінен басталады және ереже бойынша ол автоматты түрде жасыл түспен жазылады. Түсініктеме сызығы бөлек жолда немесе кез-келген командалар жазылған жолда тіркелуі мүмкін. Сонымен, берілген өрнектің қалаған мәнін табу үшін командалардың келесі реті енгізілуі мүмкін:

```
>> x=9;        % айнымалыға 9 мағынасын орналастырамыз.
>> 3*(x-4)/2  % өрнектің ізделінген мәнін есептеу.
ans =
    7.5000
>>
```

Командалық жолға қайта енгізу үшін ↑ немесе ↓ пернелерін қолдануға болады.

Сонымен қатар *Command History* ұяшығының ішіндегі жазуладрды қолдануға болады, ішіндегі мәліметтерді көшіріп, содан кейін *Command Window* жолына енгізуге болады. Егер *Command History*-ұяшығынан ақпаратты жою мақсаты туындаса, сол берілген жолды белгілеп және *Delete* батырмасын басу керек.

Егер *Command History* ұяшығының ішінде қосымша көмек керек болса, тышқанды ұяшықтың ішіне өткізіп оң жақтағы тұрған батырманы шерту керек, сол кезде қосымша терезе пайда болады. *Copy* көшіру элементі таңдалынса, бұл ақпарат Windows аралық сақтағышына көшіріледі. *Evaluate Selection* (таңдау санын анықтау үшін) арқылы таңбаланған командалар тобын орындай аласыз. Ағымдағы команданы жою үшін *Delete Selection* таңдау керек, ал егер ағымға дейінгі барлық командаларды жою мақсаты туындаса – *Delete to Selection,* сонымен қатар барлық командаларды жою – *Delete Entire History* (барлық командаларды жою).

Санды нөлге бөлу кезінде келесідей ақпарат экран бетіне шығады:

```
>> 2/0
Warning: Divide by zero.
ans =
   Inf
>>
```

Егер нөлді нөлге бөлсеңіз, есептеудің анықталмаған нәтижесі туралы хабарды көресіз:

```
>> 0/0
Warning: Divide by zero.
ans =
   NaN
>>
```

Теріс саннан түбірді есептеу кезінде қате туралы хабар шықпайды, нәтиже күрделі(комплексті) пішінде шығады.

```
>> sqrt(-1)   % «-1» санынан квадрат түбірін есептеу
ans =

        0 + 1.0000i
>>
```

Енгізілген қарапайым функциялар

Математикалық функциялардың ең танымал кластарының бірі, қарапайым функциялар болып табылады. Matlab ортасында осындай қарапайым функциялардың үлкен жиынтығы бар, оларға тригонометриялық, гиперболалық, экспоненциалық, логарифмдік функциялар, сонымен қатар күрделі сандармен жұмыс істеу функциялары және әртүрлі тәсілдермен жуықтау функциялары.

Matlab аясында тригонометриялық және оларға кері функциялар келесі түрде жазылады: `sin` – синус, `cos` – косинус, `tan` – тангенс, `cot` – котангенс, `sec` – секанс ($sec(x) = {}^{1}\!/_{cos(x)}$), `csc` – косеканс ($csc(x) = {}^{1}\!/_{sin(x)}$), `asin` – арксинус, `acos` – арккосинус, `atan` – арктангенс, `acot` – арккотангенс.

Тригонометриялық функциялардың аргументтері радиандармен көрсетілуі керек жәнеде кері тригонометриялық функциялар нәтижені радиан түрінде қайтарады.

Matlab-тағы гиперболалық және кері оған функциялар келесідей берілуі мүмкін: `sinh` – гиперболалық синус, `cosh` – гиперболалық косинус, `tanh` – гиперболалық тангенс, `coth` – гиперболалық котангенс, `sech` – гиперболалық секанс, `csch` – гиперболалық косеканс, `asinh` – гиперболалық арксинус, `acosh` – гиперболалық арккосинус, `atanh` – гиперболалық арктангенс, `acoth` – гиперболалық арккотангенс, `asech` – гиперболалық арксеканс**,** `acsch` – гиперболалық арккосеканс.

Экспоненциалды функцияларға: `exp` – экспоненциалды функция, `log` – натурал логарифм, `log10` – 10 негізі бойынша логарифм, `log2` – 2 негізі бойынша логарифм, `sqrt` – квадрат түбірі.

Комплекс сандарымен жұмыс жасау үшін берілген командалармен қолдану керек, олар `angle` – комплекс санының аргументі, `complex` – нақты және қиял бөліктерден комплекс нөмірлерді құру, `conj` – комплексті жұптастыру, `image` – комплекс санының қиял бөлігі, `real` – комплекс санының шыңайы бөлігі.

Қалдық табу мен жуықтау операциялары келесі операторлардың көмегімен іске асады: `fix` – нөлге жуықтау, `floor` – минус шексіздік жағына жуықтау, `ceil` – плюс шексіздік жағына жуықтау, `round` – ең жақын бүтін санға дейін жуықтау, `mod` – таңбаны ескере отырып қалдықты табу, `rem` – қалдықтың қалған бөлігін табу, `sign` – белгі функциясы.

Массивпен жұмыс жасау

Массив (вектор) – реттелген, нөмірі бойынша орналасқан ақпараттар жиыны. Массивте өзінің атауы болуы тиіс, демек таңбалар реті, соның ішінде әріптер не сандар, төменгі сызықша. Массив атауы цифрден басталмайды және оның атауында символдар арасында алшақтық болмауы тиіс.

Массив өзінің көлемі бойынша ерекшеленеді: бір өлшемді, екі өлшемді, көп өлшемді деректер жинақтары.

Массив өлшемі - әрбір өлшем бойынша элементтер саны.

Массив элементтерін нөмірлеу бір санынан басталады (яғни, индекстер бірден үлкен немесе тең болуы керек жәнеде олар натурал сан болады).

Массивті енгізу

Массивті енгізудің бірінші әдісін қарастырамыз. Бұл әдісте массивтің элементтері бос орындар арқылы бөлінеді. Экранда массивтің элементтері бір жолға орналастырылады. Мысалы, төрт элементтерден тұратын массив құру қажет: 2; 3; 6; -1. Ол үшін келесі команданы тереміз:

```
>> x=[2 3 6 -1]    % Массив элементтерін енгізу
x =
     2     3     6    -1
```

Екінші әдісте массив элементтерін нүктелі үтірмен бөлеміз. Экранда массив элементтері баған бойынша бейнеленеді:

```
>> x=[2; 3; 6; -1]   % Массив элементтерін енгізу
x =
     2
     3
     6
    -1
```

Массив элементін шақыру

Қажет болса, белгілі бір нөмірге тиесілі массив элементіне қатынаса аласыз. Мысалы, x атты массивтің бірінші және үшінші элементтеріне шақырылу керек.

```
>> x=[0 2 4 6]; % x массивінің сандық элементтерін енгізу
>> x(1)         % массивтің бірінші элементіне жүгіну
ans =
     0
>> x(3)         % массивтің үшінші элементіне жүгіну
ans =
     4
```

Бірнеше қатарлы массив элементтерінің мәндерін көрсету үшін операциялардың келесі тізбесін қолдануға болады:

```
>> x=[0 2 4 6]; % x массивінің сандық элементтерін енгізу
>> y=x(1:3)     % y массив құрастыру, алабында x
```

```
  % x массивінің бірінші үш элементін қамту
y =
     0     2     4
```

Массив элементтерін қосу және алу, айнымалы мәндерді қосу және алу сияқты әдіспен орындалады, мысалы:

```
>> x=[2 3 6 -1];   % x массивінің сандық элементтерін енгізу
>> y=[1 2 -2 1];   % y массивінің сандық элементтерін енгізу
>> z=x+y           % x және y массивтерін қосу
z =
     3     5     4     0
>> z=x-y           % x массивінен y массивін азайту.
z =
     1     1     8    -2
```

Егер векторларының өлшемі яғни, массив элементтерінің саны бір-біріне сәйкес келмесе, оған қосылу не азайту операциясы жұмыс жасамайды, қате туралы хабар көрсетіледі:

```
??? Error using ==> plus
Matrix dimensions must agree.
```

Массив өлшемін анықтау

Массивтің өлшемі `size` функциясы арқылы анықталады, ол осы массивте жолдар мен бағандардың санын қайтарады. Төмендегі мысалда жауап екі сан екені көрінеді. 1-саны массивтің бір өлшемді екенін көрсетеді, ал 4-саны массивтің ішінде элементтер саның көрсетеді:

```
>> x=[2 3 6 -1];   % x массивінің сандық элементтерін енгізу
>> size(x)         % массивтің өлшемін анықтау
ans =
        1    4
```

Массив элементтерінің санын анықтау

Массив ішіндегі элементтер санын анықтау үшін `length` функциясы қолданылады. Төмендегі мысалдағы команданың нәтижесі, зерттелетін x массивінің жеті элементі бар екенің көрсетеді.

```
% массив элементтерін енгізу
>> x=[1 2 3 5 3 3 3];
```

```
% массив ішінде элементтер санын санау
>> length(x)
ans =
     7
```

Массивтің әрбір элементінен тригонометриялық функцияны табу

Matlab-та массивтің барлық элементтерінен тригонометриялық функцияның мәнін табуға болады. Осылай, төменде келтірілген массив элементтерінің әрқайсысынан синусалды мәндерді қалай табуға болатынын көрсеткен.

```
% функция аргументінің мәнін енгізу
>> x=[0 pi/4 pi/2 3*pi/4 pi];
% Функцияның мәнін есептеу
>> y=sin(x)
y =
         0    0.7071    1.0000    0.7071    0.0000
```

Массивтің әрбір элементінен квадрат түбірін табу

Алдыңғы жағдайға ұқсас, массивтің әрбір элементі үшін кез келген қарапайым функция берлуі мүмкін, мысалы, массивтің әрбір элементі үшін квадрат түбірі табылды:

```
% Массив үшін сандық элементтер енгізу
>> x=[0 9 25 49 81];
% x массивінен квадрат түбірін анықтау
>> y=sqrt(x)
y =
     0     3     5     7     9
>>
```

Элементтер бойынша векторлармен жұмыс жасау

Элементтерді көбейту, бөлу және массивтерді дәрежеге тұрғызуды көрсету үшін, математикалық операциялардың тиісті белгілерінің алдында «.» қойылады. Мысалы, екі массивті элементтері бойынша көбейту келесідей орындалады. Ол үшін екі векторлы жол енгіземіз:

```
>> x1=[2 3 4 -1];
>> x2=[1 0 -1 -2];
```

«.*» операциясы бірдей ұзындықтағы векторлады көбейтеді.

```
>> y=x1.*x2
```

Соңында экранда келесідей мән шығады:

```
y =
     2     0    -4     2
```

Төменде Matlab аясында элементтерге қатысты дәрежеге тұрғызу, бөлу, саны бар векторлады қосу және векторды санға көбейту сияқты мысалдар келтірілген.

x1 және x2 атты екі массивтерді элементтері бойынша дәрежеге тұрғызу.

```
>> x1=[0 1 -1 2];
>> x2=[1 1 2 3];
>> y=x1.^x2
y =
     0     1     1     8
```

Осылайша, келтірілген қарапайым командалардың қатарынан бірінші `x1` массивінің элементтері, `x2` элементінің біріншісіне көбейтіледі, содан соң, `x1` массивінің екінші элементі `x2` массивінің екінші элементіне көбейтіледі және с.с.

Элементтері бойынша массивті бөлу

```
>> x1 = [0 2 10 20];    % x1 массивінің элементтерін енгізу
>> x2 = [1 2 5 4]; % x2 массивінің элементтерін енгізу
% x1 массивін x2 массивіне элементтері бойынша бөлу
>> y=x1./x2
y =
     0     1     2     5
```

«.» белгісі, тек элементтері бойынша екі массивтің көбейтілуінде, бөлінуінде және өрнектелуінде енгізіледі. Қалған жағдайда бұл символ қолданбайды.

Төменде келесі математикалық әрекеттер көрсетіген: векторды санмен қосу және векторды санға көбейту.

Векторды санмен қосу

```
>> x=[0 2 10 20]; % x массивінің элементтерін енгізу
% x массивінің элементтерін «1» санына қосу
>> y=x+1
y =
     1     3    11    21
```

Векторды санға көбейту

```
>> x=[0 2 10 20]; % (массив) вектордың элементтерін енгізу
% вектор элементтерін «2» санына көбейту
>> y=x*2
y =
        0     4    20    40
```

Векторлардың скаляр туындысы

a және *b* векторлардың скаляр туындысы *N* (яғни, әрбір вектор *N* элемент қамтиды) ұзындығы нақты сандардан тұрады, ол келесі формуламен анықталады $a \cdot b = \sum_{k=1}^{N} a_k b_k$.. Екі вектордың скаляр туындысын табу үшін `sum` функциясы қолданылады. Төменде *a* және *b* массивіне скаляр туындысының қолданылуы көрсетілген.

```
>> a=[1.2 -3.2 0.7];    % a массиві үшін элементтер енгізу
>> b=[4.1 6.5 -2.9];    % b массиві үшін элементтер енгізу
% a және b векторларының скаляр туындысын есептеу
>> s=sum(a.*b)

s =
  -17.9100
```

Векторлардың векторлық туындысы

Векторлардың векторлық туындысын табу үшін `cross` фукциясы қолданылады.

```
>> a=[1.2 -3.2 0.7];    % a массиві үшін элементтер енгізу
>> b=[4.1 6.5 -2.9];    % b массиві үшін элементтер енгізу
% a және b векторларының векторлық туындысын есептеу
>> c=cross(a,b)

c =
    4.7300    6.3500   20.9200
```

Матрицамен жұмыс жасау

Матрица – бұл тікбұрышты таблица түрінде жазылатын элементтері бар математикалық объект, оның бойында жолдар мен бағандар жиынтығы бар, оның қиылысында элементтері болады.

Matlab ортасында матрицаға қатысты кейбір операцияларды қарастырайық.

Мысалға, берілген $A=\begin{pmatrix} 3 & 1 & -1 \\ 2 & 4 & 3 \end{pmatrix}$ матрицаны енгізу керек. Ол үшін нүктелі үтір арқылы матрицаның элементтерін жолға жазамыз:

```
>> A=[3 1 -1; 2 4 3] % матрицаның элементтерін енгізу

A =
     3     1    -1
     2     4     3
```

Матрицаның белгілі бір элементіне сілтемені бағдару мына түрде орындалады:

```
>> A(2,3)% матрицаның екінші жолындағы және үшінші
         % бағанының элементіне сілтеу бұру
ans =
     3
```

Төменде бір матрицаны келесімен беттестіруге (қосуға), бір-бірінен азайтуға, оларды көбейтіп және массивтің әрбір элементін дәрежеге көтеруге қатысты мысалдар талқыланған.

Матрицаларды қосу

```
>> A=[3 1 -1; 2 4 3];  % A матрицасының элементтерін енгізу
>> C=[3 -1 7; 4 2 0];  % C матрицасының элементтерін енгізу
>> S=A+C  % A және C матрицаларының элементтерін қосу

S =
     6     0     6
     6     6     3
```

Матрицаларды азайту

```
>> A=[3 1 -1; 2 4 3];  % A матрицасының элементтерін енгізу
>> C=[3 -1 7; 4 2 0];  % C матрицасының элементтерін енгізу
         >> S=C-A    % A және C матрицаларының элементтерін
         азайту
```

```
S =
     0    -2     8
     2    -2    -3
```

Матрицаларды көбейту

```
>> C=[3 -1 7; 4 2 0];  % C матрицасының элементтерін енгізу
>> B=[4 3 -1; 2 7 0; -5 1 2]; % B элементтерін енгізу
>> P=C*B  % A матрицасының элементтерін C матрицасының
элементтеріне көбейту

P =
   -25     9    11
    20    26    -4
```

Ескерту: матрицалармен жұмыс жасау кезінде, элементтері бойынша көбейту, бөлу және дәрежеге көтеру сияқты Массивте қолданылатын операциялар қабылданбайды.

Дәрежеге көтеру

$B = \begin{pmatrix} 4 & 3 & -1 \\ 2 & 7 & 0 \\ -5 & 1 & 2 \end{pmatrix}$ матрицасының әрбір элементін дәрежеге көтеру делік.

Матрицалық элементтерді квадратқа салу үшін келесі командалардың қатарын енгіземіз:

```
% B матрицасының элементтерін енгізу
>> B=[4 3 -1; 2 7 0; -5 1 2];
% B2 матрицасының пайда болуы,
% B матрицасының элементтерін квадраттайды.
>> B2=B^2

B2 =
    27    32    -6
    22    55    -2
   -28    -6     9
```

Функцияның мәні үшін есептеу нәтижелерін визуализациялау

Есептеу нәтижелерін көрнекілендіру түрлі жолдармен жүзеге асырылуы мүмкін: есептеу нәтижелерін қамтитын кесте түрінде, екі өлшемді, үш өлшемді графиктер түрінде және т.б.

Төмендегі мысалды қарастырып көрейік. $[0,1]$ бөлігінде және 0.05 қадаммен аттайтын $y(x)=e^{-x}sin10x$ функциясы үшін мәндер кестесі көрсетілсін делік. Ол үшін келесі бағдарлама құрамыз:

```
>> x=0:0.05:1; % абциссаның мәндерін енгізу
% Қажетті функцияның мәндерін есептеу:
>> y=exp(-x).*sin(10*x);
>> x   % Экранға есептеу нәтижелерін көрсету
```

Бұл жағдайда есептеу нәтижелері төмендегі формада көрсетіледі:

```
x =
  Columns 1 through 7
0    0.0500    0.1000    0.1500    0.2000    0.2500    0.3000
  Columns 8 through 14
0.3500    0.4000    0.4500    0.5000    0.5500    0.6000
0.6500
  Columns 15 through 21
0.7000    0.7500    0.8000    0.8500    0.9000    0.9500
1.0000

y =
  Columns 1 through 7
0    0.4560    0.7614    0.8586    0.7445    0.4661    0.1045
  Columns 8 through 14
-0.2472   -0.5073   -0.6233   -0.5816   -0.4071   -0.1533
0.1123
  Columns 15 through 21
 0.3262    0.4431    0.4445    0.3413    0.1676   -0.0291   -
0.2001
```

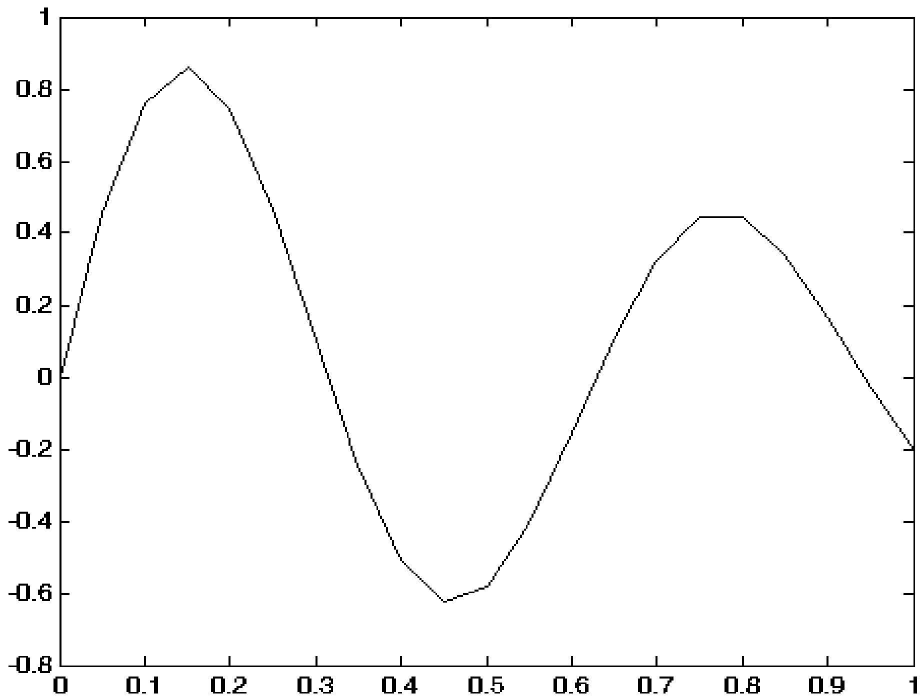
Matlab аясында екі өлшемді график

Эксперименттік есептеулердің нәтижелерін көрсету және графикалық түріндегі функциялардың мәндерін есептеу кезінде осы операция ыңғайлы.

Екі өлшемді графиктерді тұзғызу үшін `plot` командасын пайдаланыңыз. Төменде анықтау облысының 0-ден бастап 1-ге дейін және оның қадамы 0.05 болатын $y(x)=e^{-x}\sin(10x)$ тәуелділікті құру келтірілген.

```
>> x=0:0.05:1;  % Функция аргументінің мәнін енгізу
>> y=exp(-x).*sin(10*x);% функция мәнін есептеу
>> plot(x,y) % графикалық терезеге
             % есептеу нәтижелерін шығару
```

Сипатталған әрекеттердің нәтижесінде төмендегі кесте көрсетіледі:



1 сурет. $y(x) = e^{-x}\sin(10x)$ функциясының графигі

Графикке қисық сызықтарды қосу `hold on` командасы арқылы орындалады. Бір графикалық терезеде бірнеше графиктерді шығару `subplot` командасы арқыды жүзеге асады. Нүктелер санының көбеюі, графиктегі сигналдың неғұрлым нақты көрсетілуіне әкеледі.

Matlab ортасында үшөлшемді графиктар

Үшөлшемді график үшін Matlab `plot3` командасы қисықты визулдау үшін қолданылады. Бұл қисықтың анимациясын алу үшін `comet3` функциясы қолданылады. Жазықтық бетін тұрғызу үшін `mesh`, `surf`, `surfl` командаларын пайдалануға болады.

Мысал ретінде қандай да бір қаңқалы беттің графигін тұрғызайық. Екі айнымалысы бар фукнциялы бетті тұрғызу үшін бізге керек:

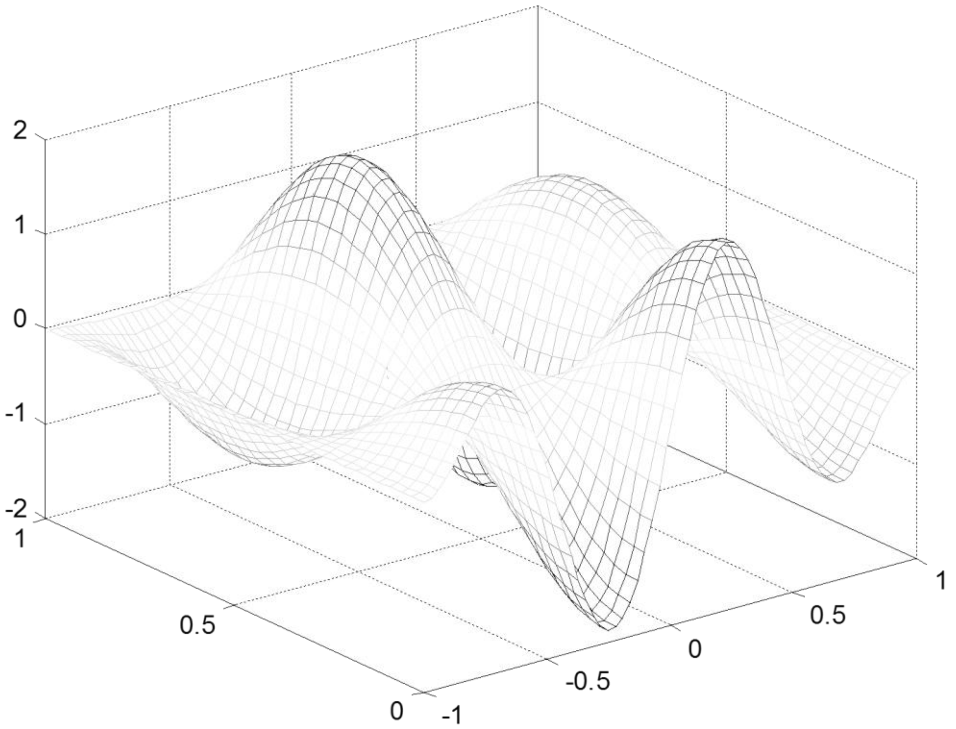
1. тікбұрышты функияның анықталу аямағында координаталары түйін болатын матрицаны түрлендіру;
2. тордың түйіндеріндегі функциясын есептеу керек және шыққан нәтижені матрицаға жазу;
3. графикалық функциялардың біреуін қолдану;
4. графикке қосымша мәліметтерді еңгізу керек, функцияның мағынасына қарай түстерін сәйкестерндіру.

Келесі мысалды қарастырайық. Анықталу аймағы тіктқөртбұрыш болатын $x \in [-1,1], y = [0,1]$, екі айнымалысы бар функцияның графигін тұрғызу керек болсын $z(x,y) = 2 \cdot sin\pi x \cdot cos1.5\pi y \cdot (1 - x^2)$.

Ол үшін келесі командаларды ретімен еңгіземіз.

```
%Координаталары түйін болатын торлы матрицаны түрлендіреміз
>> [x,y]=meshgrid(-1:0.03:1, 0:0.03:1);
% тордың әр бір түйініндегі функцияны есептейміз
>> z=2*sin(2*pi*x).*cos(1.5*pi*y).*(1-x.^2);
% каркасты бетті визуализациялаймыз
>> mesh(x,y,z)
```

Бұл прогамманы орындау нәтижесі 2 суретте көрсетілген. Мұндай графикті визуализациялау үшін *x* және *y* осьтерін бірдей еңгізу міндетті еместігіне назар аударған жөн. Сонымен қатар осьтердің қадамын да бірдей беру міндетті емес.



2 Сурет. Каркасты бет

Толық түрде, сигналдарды өңдеу үшін тікелей қолданылатын Matlab ортасында компьютерлік модельдеу әдістері төмендегі зертханалық жұмыстардың сипаттамасында сипатталған.

Зертханалық жұмыс №1

MATLAB ОРТАСЫНДА СИГНАЛДАРДЫ МОДЕЛЬДЕУ

Жұмыстың мақсаты: Matlab ортасының Communications Toolbox және Signal Processing Toolbox кеңейту пакеттері көмегімен сигналдарды модельдеу дағдыларын меңгеру.

Қысқаша теориялық кіріспе

Signal Processing Toolbox кеңейту пакеті табиғаты әр түрлі сигналдар мен бейнелерді модельдеу және өңдеу үшін қолданылады. Communications Toolbox кеңейту пакеті Matlab ортасында телекоммуникациялық жүйелерді модельдеумен байланысты есептерді орындау үшін қолданылады. Осы пакеттердің көмегімен сигналдарды генерациялауға және оларды өңдеуге , соның ішінде Фурье-түрлендіруді орындау, спектограммаларды, периодограммаларды құру және т.б. болады.

Сигнал бір өлшемнің екіншіге тәуелділігін көрсетеді. Басқаша айтқанда, математикалық тұрғыдан сигнал математикалық функция ретінде қарастырыла алады.

Компьютерде сигналдарды өңдеу үшін оларды дискретті түрге айналдыру керек. Айналдырудың бір тәсілі – біркелкі уақытта сигнал мәндерін белгілі бір уақыт аралығында біркелкі өлшеу және амплитудалардың мәндерін компьютерге енгізу. Өлшеулерді жиі жасаған болса, онда алынған дискреттік сигналдан бастапқы үздіксіз сигналдың көрінісін қалпына келтіре алуға болады. Әдетте, компьютерлік математика жүйесінде үздіксіз сигналдар абстракция болып табылады. Есептеу жасау үшін белгілі бір уақыт аралығында анықталған дискретті сигналдар беріледі - көбінесе тұрақты қадаммен. Matlab-та сондай сигналды сипаттау үшін уақыт векторы көрсетіледі, мысалы: `t=0:0.1:10`. Мұнда 11 уақыт үлгілері 0.1 қадаммен 0-ден 10-ға дейін орнатылады. Жалғыз (немесе бір арналы) сигналдар келесідей орнатылады.

```
y1=sin(t) % Синусоидалды сигнал
y2=t         % Сызықты өспелі сигнал
y3=t.^2      % Квадратты сигнал
y4=exp(-t)   % Экспоненциалды құлдырайтын сигнал
```

Бұл сигналдардың әрқайсысы уақыт векторының t өлшеміне тең вектормен ұсынылған. Matlab-да бірнеше (көп арналы) сигналдарды көрсету мүмкін, мысалы:

```
ym=[y1 y2 y3 y4]
```

немесе

```
ym=[sin(t)   t   t.^2    exp(-t)]
```

Мұндай сигнал матрицамен бейнеленеді.

Шуылдарды модельдеу үшін белгілі бір тарату заңымен кездейсоқ сандар генераторы қолданылады. Signal Processing Toolbox ішіндегі ең маңызды сигналдар арнайы функциялармен белгіленеді, бірақ кез-келген сигнал жоғарыда сипатталған мысалдарда Matlab арқылы жасалуы мүмкін.

Көптеген сигналдар *детерминирленген* болады, яғни олардың $y(t)$ уақыттық тәуелділігі аналитикалық түрде t уақыттың кез-келген моментінде анықталған. Детерминирленген сигналдар аналитикалық сипаттама және өңдеу жүйелерінің анализі мен сигналдардың түрлендіруі үшін қолдануға ыңғайлы және сынақ сигналдарының рөлінде кеңінен қолданылады.

Дегенмен, көптеген сигналдар детерминирленген емес. Бұлардың себептерінің бірі – белгілі бір бөлу заңымен кездейсоқ сипатқа ие шу мен кедергілер арқылы сигналдардың бітелуі. Signal Processing Toolbox ішіндегі шулы сигналдарды модельдеу кездейсоқ сандар генераторы арқылы қамтамасыз етіледі. Тағы бір себебі – олардың ақпарат тасымалдаушылары ретінде болуы болып табылады.

Детерминирленген және детерминирленген емес сигналдар арасында келісімге келу кілті іске қосылған сайын қайталанатын шу компоненті бар детерминирленген сигналдар болып табылады. Бұл, бір жағынан, шудың әсерін ескеруге, ал екінші жағынан, бірнеше модельдеу процесінде қайталанатын нәтижелерді қамтамасыз етуге мүмкіндік береді.

Бұдан әрі радио және телекоммуникациялық жүйелерде байқалатын детерминирленген және детерминирленген емес сигналдар кейбір (бірақ түгел емес) модельдеу мысалдарын қарастырамыз.

Шулы сигнал құру

Шудың құрамдас бөлігінің орташа мәні нөлге тең, екі синусоидалық компоненттері бар - бірінші (іргелі) және үшінші гармоникадан тұратын шулы сигналды модель құру қажет. Сигнал $s(t) = A_1 \sin(2\pi f_1 t) + A_2 \sin(2\pi f_2 t).$ байланыспен сипатталады. Сигналды табу аймағы нөлден 4-ке дейін. Амплитудалардың сандық мәндері және сигнал құрамдас бөліктерінің жиілігі сәйкесінше 1.1, 0.2, 1 және 3 болып табылады. Шу амплитудасы 0,2-ге тең.

Төменде шулы сигнал құру бойынша бағдарлама тізімі берілген.

Екі синусоидалы компоненттері бар шулы сигналды құру бағдарламасы

```
clear; clc; % Workspace және Command Window тазарту
t=(0:.01:4)'; % Уақыт вектор-бағаны
W1=1; w2=3; % Жиіліктің мәндері
A1=1.1; A2=0.2; % Сигнал компоненттерінің амплитудалары
An=0.2; % Шуыл амплитудасы
s=A1*sin(2*pi*w1*t)+A2*sin(2*pi*w2*t);% Шуыл жоқ сигнал
sn=s+An*randn(size(t)); % Шуылданған сигнал
plot(t,s,t,sn) % Шуыл қосылған сигнал визуализациясы
xlabel ('t') % Абсциссалар осінің аты
ylabel ('sn')% Ординаталар осінің аты
```

Бағдарламаның бірінші жолы компьютердің жадындағы айнымалылардың сандық мәндерін алып тастау және Matlab жұмыс терезесінің аумағын тазалау үшін қажет.

Екінші жол уақыттың сигналын квантизациялауды қамтамасыз ететін векторлық бағанды көрсетеді. Апостроф белгісі вектордың транспозициясын білдіреді, яғни оны баған векторына векторлық жолдан түрлендіреді. Дегенмен, бірінші кезекте, мұндай трансформация маңызды мәнге ие болмаса да, ол өте маңызды болып табылады - көптеген Matlab функциялары векторларды нақты векторлар түрінде немесе баған векторларының түрінде бірқалыпты спецификациялауды талап етеді.

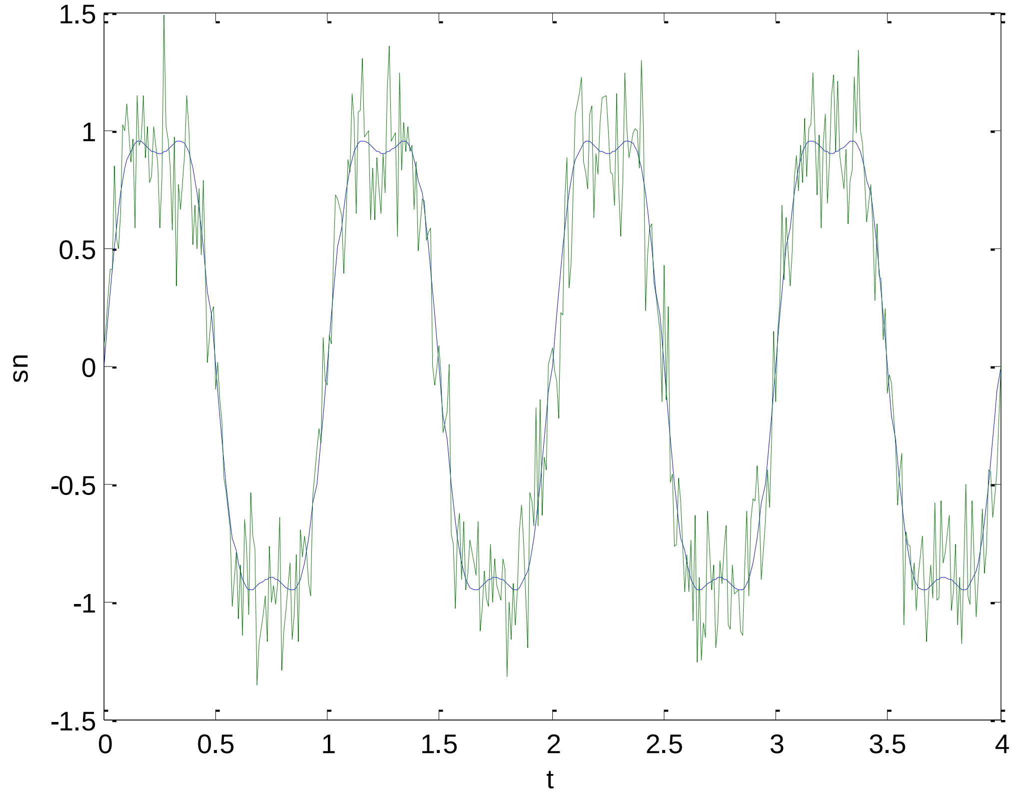
Бағдарламаның үшіншіден бесінші жолға дейін амплитудалардың сандық мәндері және синусоидалы сигналдың компоненттерінің жиіліктері, сондай-ақ сигналға енгізілген шудың амплитудасы енгізіледі.

Алтыншы жол шуылмен бөлінбейтін `s` сигнал болып табылатын функцияның векторын анықтайды.

Келесі жол сызықтық шу болып табылатын ақ шуыл қосымша қосылатын сигналдың сипаттамасына арналған, спектральды компоненттері барлық жиілік диапазонында біркелкі таратылады. Ақ шуды кездейсоқ сандар генераторымен (`randn` командасы) қолдануға болады. Бұл функция теріс және оң кездейсоқ сандарды бірдей генерациялайды. Кездейсоқ сандарды генерациялау үшін `rand` функциясын қолдануға болады. Бұл жағдайда 0-ден 1-ге дейінгі ауқымда тек оң кездейсоқ сандар шығарылады.

Соңғы үш жол шулы сигналды бейнелеу және координат осіне атау қою үшін қолданылады.

Бұл бағдарламаны іске асырудың нәтижесі 1.1 суретте көрсетілген.



1.1 сурет. Шуылдан тұратын сигнал
екі синусоидалы компонент

Шу амплитудасының өзгеруі сигналдың едәуір өзгеруіне әкеледі, бұл параметр `An` мәнін өзгерту арқылы көрінеді.

Сигналға ақ шуды қосу `awgn` функциясын пайдалану арқылы орындалуы мүмкін. Бұл жағдайда жоғарыда аталған бағдарламалар тізімінде жолдың орнына

```
sn=s+An*randn(size(t));
```

жазу керек

```
sn=awgn(s,10,'measured');
```

Мұнда `s` - дыбыссыз сигналдың үлгілері бар массивтің атауы, 10 - децибелде сигналдан шуыл қатынасы шамасы, ал `'measured'` сигнал күші нөл деп есептелмейтінін, бірақ автоматты түрде есептелетін параметрді білдіреді.

Айнымалы жиілікті косинусоида модельдеу

Осы мақсатта, `chirp` функциясы қолданылады. Төменде көрсетілген:

```
s=chirp(t,f0,t1,f1,['method',phi]).
```

chirp функциясы косинус сигналының іріктемесін (дискреттік мәндерді) f_0 бастапқы t уақыт кезеңінен соңғы t_1. уақыт кезіне дейінгі f_1 жиілікте қалыптастырады. Әдетте $t=0,\ f_0=0,\ f_1=100.$ Қосымша параметр phi (мәні әдетте нөлге тең) сигналдың бастапқы фазасын орнатады. Басқа қосымша параметр 'method' жиіліктің өзгеру заңдылығын орнатады. Берілген параметр келесідей мәндерді қабылдай алады:

- linear – жиілік өзгерісінің сызықтық заңы $f_j(t)=f_0+at$, осыдан $a=(f_1-f_0)/t$;
- quadratic – жиіліктің өзгеруінің квадраттық заңы $f_j(t)=f_0+at^2$, осыдан $a=(f_1-f_0)/t_1$;
- logarithmic – жиіліктің өзгеруінің логарифмдік заңы $f_j(t)=f_0+10at$, осыдан $a=\left[\log 10(f_1-f_0)\right]/t_1$ и $f_1>f_0$.

Әдепкі мән method=linear. Әдепкі мәндер тиісті айнымалы болмаса немесе бос мән көрсетілсе қолданылады.

Chirp функцияны пайдалануды қарастырамыз. Оның жиілігі полиномиялық заңға байланысты өзгеретін косинус сигналын көрсету қажет. Тиісті бағдарламаның тізімі төменде көрсетілген.

Айнымалы жиілікті косинусоида модельдеу бағдарламасының листингі

```
clear; clc; % Workspace және Command Window тазарту
t= 0:0.25:2; % Уақыт векторының берілуі
f=[0 150 200 150 100 130 150 240 300]; % жиілік векторының
% берілуі
p=polyfit(t,f,4) % 4-ші реттік полином регрессиясы
t=0:0.001:2; % Уақыт векторының берілуі
s=chirp(t,p) % Сигнал генерациясы

% Бір графикалық терезеде екі график ішіндегі біріншісінің
% визуализациясы
subplot(211);
plot(t,polyval(p,t));
title('Модельдеуші полиномдық функция')

% жиілікті-модульденген сигнал спектрограммасын бірінші
% график құрылған графикалық терезеде тұрғызу
set(gca,'ylim',[0 500]); % осьтердің қасиеттерін сипаттау
subplot(212);
specgram(s,128,1E3,128,120);
title ('Сигнал спектрограммасы')
```

Графикалық терезеде subplot функциясы бірнеше графиканы бір уақытта көрсетуге мүмкіндік береді. Графикалық терезені m × n ішкі терезеге subplot (m, n, p) немесе subplot (mnp) функциясы арқылы бөледі, m – тік бағандардағы папкалар саны, n – жолдың саны, p – ағымдық панельдің нөмірі.

Модульдің алғашқы үш жолы косинус сигналының жиілігін модульдеу үшін пайдаланылатын уақыт функциясын сипаттайтын төртінші ретті полиномның құрылысын көрсетеді – келесі екі жол. Полином коэффициенттерін есептеу үшін бағдарлама polyval функциясын қолданады. Осы команданың нақты тереңдетілген қолданылуы келесі зертханалық жұмыста қарастырылады.

Спектрограмманы құру үшін (жалпы айтқанда, сигнал спектрі уақытқа байланысты оның лездік спектрі деп аталады) келесі синтаксисі бар specgram функцияның пайдаланылуы қолданылады:

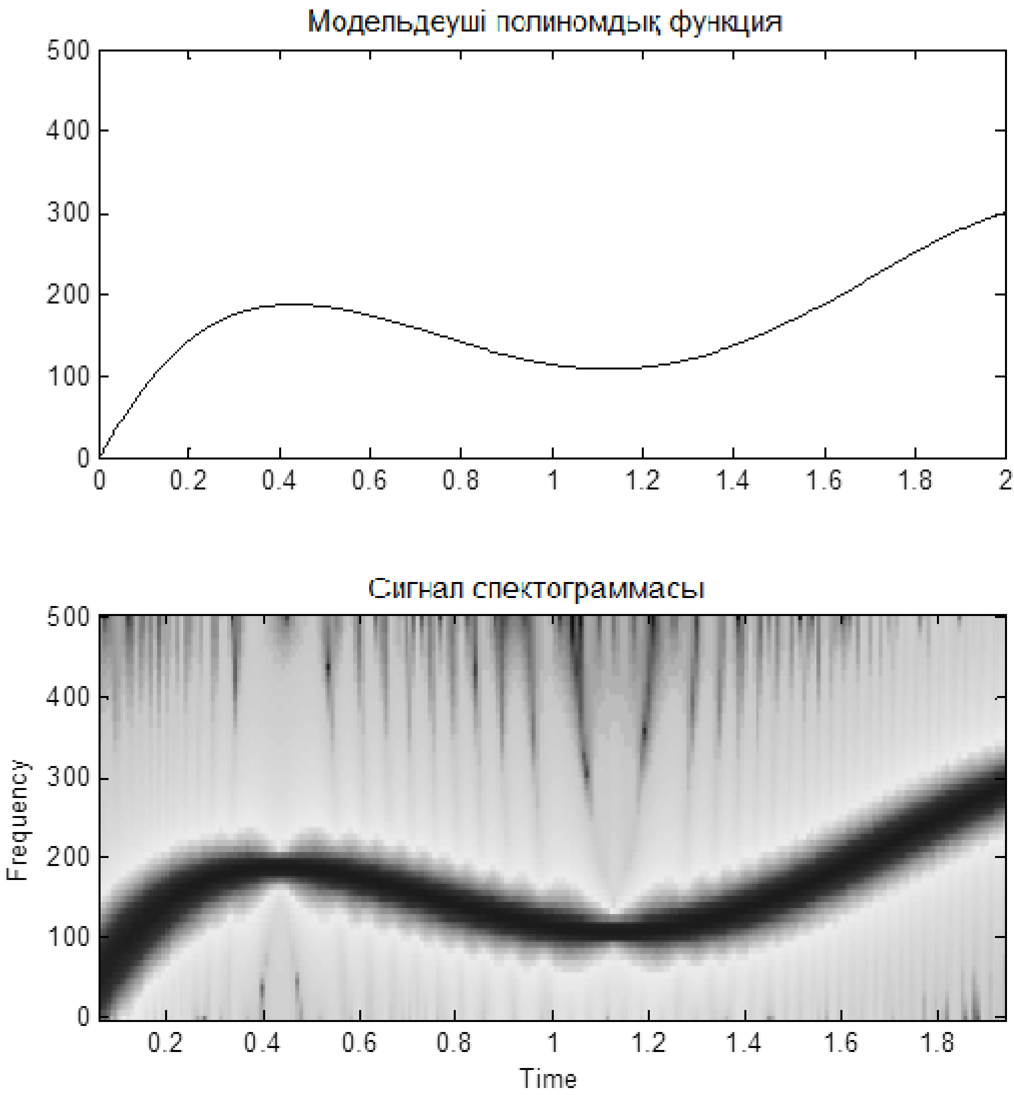
```
B = specgram(s,nfft,Fs,window,noverlap)
```

Параметр s - спектрограммасы құрастырылған сигналдың атауы. Параметр nfft дифференциалды Фурье түрлендіруі орындалатын жиіліктерді анықтайды. Fs – бұл сигналдың жиілігінің дискреттеу жиілігін сипаттайтын скалярлық мөлшер. Window параметрі спектрограмманы жоспарлау үшін пайдаланылатын үлгілердің санын анықтайды. Қабатталатын аймақтардағы үлгілер саны noverlap параметрінің мәнімен сипатталады.

Белгіленген set функция графикалық объектінің қасиеттері үшін белгілі бір мәндерді орнату үшін пайдаланылады. Графикті құру үшін ағымдық оське көрсеткішті алуға қызмет ететін gas функциясы да пайдаланылады. gas функциясын шақырғанда, егер олар бұрын құрылған болмаса, осьтер құрылады. Егер графикалық терезе бастапқыда жасалмаса, ол осы функция арқылы құрылады. 'ylim' командасы у осіндегі ең үлкен және ең аз мәндерді анықтайды.

1.2-суретте осы функцияда көрсетілген заңмен модификацияланған модельдік полиномды функцияның графигі және сигнал спектрограммасы (уақытқа қатысты вектордың амплитудалық компоненті) көрсетілген. Фурье түрлендіру спектрограммасы синусоидалы сигналдың модуляциясын анықтайды, ол модуляциялық функцияны құрастырады.

Суретте полиномиальды функцияның графиктері және берілген функция заңдылығымен сипатталған сигнал спектограммасы көрсетілген.



1.2 Сурет. Модельдік полиномдық функцияның графигі
және осы функциямен берілген заңға сәйкес модуляцияланған
спектрограмма сигналы

Импульстік генерация

Телекоммуникациялық жүйелердегі көптеген сигналдар импульстік сигналдар ретінде жіктелуі мүмкін.

Импульстік сигнал тұрақты күйдегі қысқа мерзімді өзгеріске ие тұрақты күйдегі уақыттық сипаттамалармен салыстырғанда қысқа уақыт интервалымен сипатталатын сигнал болып табылады. Импульс қысқа мерзімді сигнал деп айтуға болады.

Импульстік сигнал физикалық мөлшердегі қысқа мерзімді өзгеріс (өріс, материал ортасының параметрі және т.б.) ретінде түсініледі. Импульстік сигналдар табиғатқа байланысты акустикалық, электромагниттік, электрлік және т.б. бола алады.

Импульстік сигналдардың қасиеттерін анықтайтын негізгі параметрлер:

– ұзақтығы (кеңістіктегі ұзындық);
– амплитудасы, яғни, Белгілі бір деңгейден максималды ауытқудың мәні,
– майданның және ұзындықтың (ұзындықтың) ұзақтығы (рецессия);
– қоршаған ортадағы қозғалыс жылдамдығы.

Радиоэлектроникада бірыңғай импульстік сигналдар әдетте *бейне импульс* деп аталады, ал видео импульсінің заңына сәйкес өзгеретін жоғары жиілікті тербелістердің қысқа пакеттері - *радио импульс* болып табылады.

Радиолокацияда қолданылатын радиоимпульстік сигналдар амплитудалық-модуляциялы тербелістердің жеке жағдайы ретінде қарастырылуы мүмкін.

Түрлі пішіндегі импульстік сигналдар сандарын жасау үшін функция `s=pulstran(t,d,'func'[,p1,p2,...])` қолданылады. `func` параметрімен сигнал формасы беріледі, оның мәндері болуы мүмкін:

– `gauspuls` – синусоида, Гаусс заңымен модульденген ;
– `rectpuls` –тікбұрышты импульс;
– `tripuls` – үшбұрышты импульс.

`t` уақыттық векторы бойынша берілген уақытты санау үшін `s` векторы есептелінеді, формула бойынша:

```
s=func(t-d(1))+func(t-d(2))+…
```

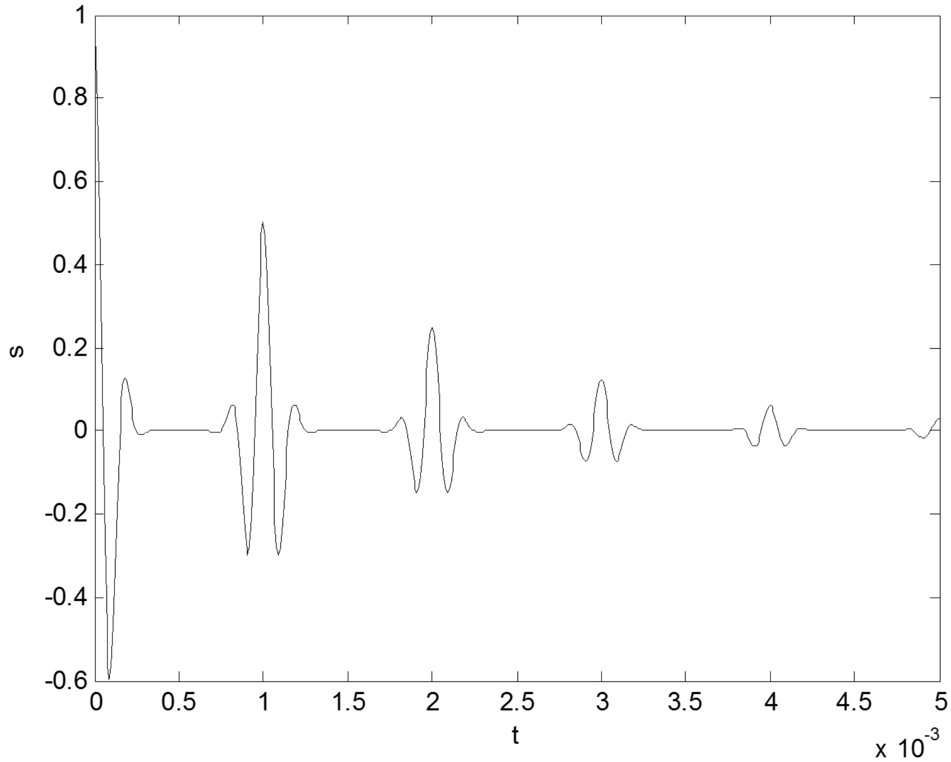
Уақыт аралығындағы импульстар саны `d` вектордың ұзындығы бойынша беріледі, яғни `length(d)`. Шартты емес параметрлер `p1, p2,…` керек кезде `'func'` функциясын шақыру арқылы қосымша параметрлерді енгізу мүмкіндігі. Мысалы, `func(t-d(1), p1,p2,…)`. `s=pulstran(t,d,p,[fs])` функциясы ретінде жазған кезде `fs` дискретизацияның жиілігін беруге болады(әдетте 1 Гц мәні қолданылады).

Төменде `pulstran` функциясын қолданатын импульстік сигналдарды шығаратын бағдарлама көрсетіледі.

`pulstran` *функциясы қолдану кезіндегі импульстік сигналдарды модельдеу бағдарламасының листингі*

```
clear; clc;
t=0:.00001:.005; % Сигнал анықтау облысы
d=[0:.001:.01;0.5.^(0:10)]'; % Импульс санын беру
% Импульсты сигнал генерациясы
s=pulstran(t,d,@gauspuls,5000,.8);
plot(t,s) % Сигнал визуализациясы
xlabel ('t') % Абсциссалар осінің аты
ylabel ('s') % Ординаталар осінің аты
```

Осы бағдарлама нәтижесінде алынған импульстік сигнал 1.3-суретте көрсетілген. Таңдау жиілігін өзгерту импульстердің пішінін де, олардың салыстырмалы орналасуын да айтарлықтай өзгерте алады, себебі `fs` параметрін өзгерту оңай көрінеді.



1.3 Сурет. `pulstran` функциясымен салынған сигнал графигі

Дирихле функциясын пайдалану арқылы сигналды модельдеу

Дирихле функциясы (немесе мерзімді sinc-функция) - бұл иррационалды (дәлелсіз сан m/n функциясы ретінде көрсетіле алмайтын болса, дәлел 0 рационалды сан және 0 болса, 1 мәнін алатын функция, мұнда m бүтін және сан болып табылады, n табиғи). Сандық сигналдарды өңдеу және байланыс теориясында нормаланған sinc функцияның (латын sinus cardinalis - «кардиналды синус») келесідей анықталады

$$diric(x,n)=\begin{cases} -1^{\frac{x}{2\pi}(n-1)}, & x=0;\pm 2\pi;\pm 4\pi,... \\ \dfrac{\sin(nx/2)}{n\sin(x/2)}, & x\neq 0;\pm 2\pi;\pm 4\pi,... \end{cases}.$$

Бұл жерде *n* кез-келген оң сан. Дирихле функциясының минимал және максимал мәндері -1 және 1 ге тең.

Байланыс теориясында sinc-функциясы аналогты сигналды оның санағы (отсчеты) бойынша бірыңғай және жоғалтусыз қайта қалпына келтіруге мүмкіндік береді, ал sinc-функциясының квадраты, амплитудасы sinc-функциясымен сипатталатын сигналдың интенсивтілігін немесе қуатын анықтайды. Sinc-фильтр сигнал спектрінде қандай да бір кесу жиілігінен жоғары болатын барлық жиіліктерді басады, ал сол жиіліктен төмен барлық жиіліктерді өзгеріссіз қалтырады.

Matlab та бұл функция келесідей сипатталады:

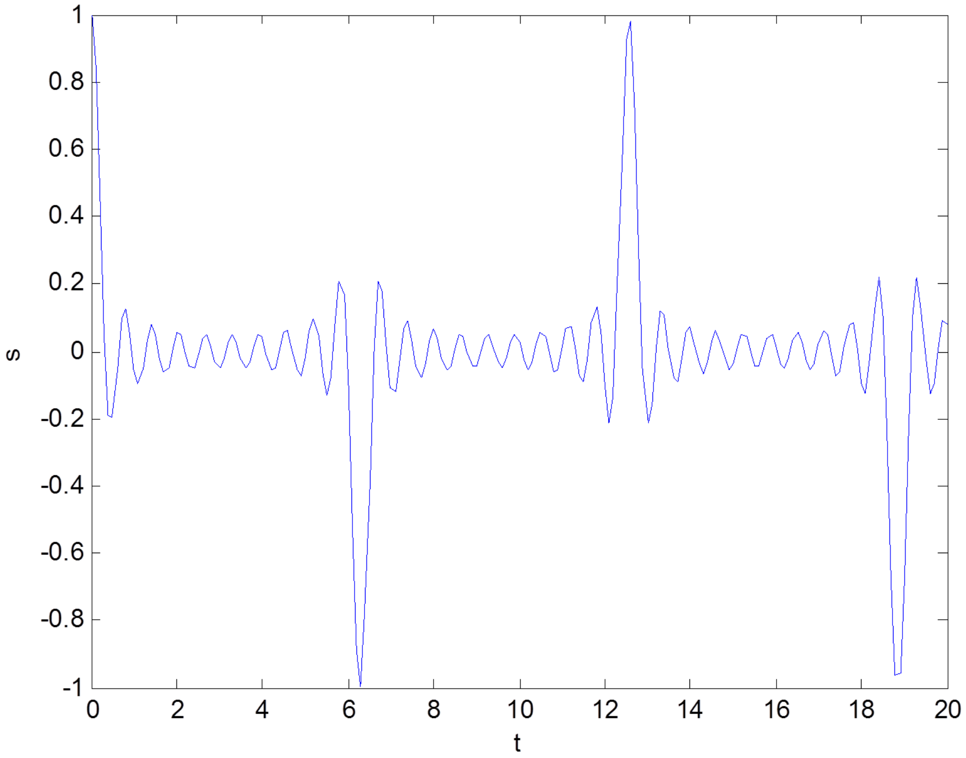
```
s=diric(x,n).
```

N-саны Дирихле функциясының түріне елеулі түрде әсер етеді. Төменде n=20 болғандағы Дирихле функциясын модельдеуге мүмкіндік беретін программа берілген.

Дирихле функцисын модельдеуге арналған программа листингі

```
clear; clc;
x=0:0.1:20;% сигналдың анықталу облысы
n=20; % Дирихле функциясының түрін анықтайтын параметр
s=diric(x,n); % Дирихле функциясын модельдеу
plot(x,s) % сигналды визуализациялау
xlabel ('t'); ylabel ('s')
```

Осы программаны санаққа жүктеу нәтижесінде 1.4 суретте көрсетілген график алынады.



1.4 сурет. Дирихле функциясы арқылы құрылған сигнал

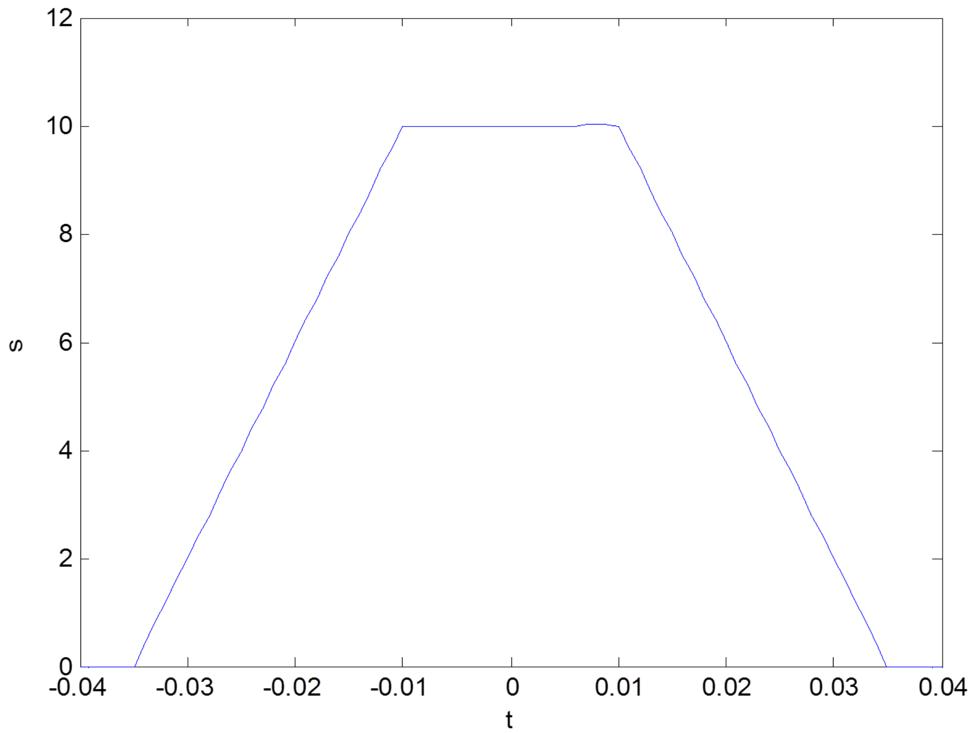
Осы әдіс арқылы сигналдарды модельдеу кезінде мақсатты түрде n параметрінің сигнал түріне әсерін қарастыру қажет.

Трапеция түріндегі импульсті модельдеу

Амплитудасы 10В, жоғарғы және төменгі негіздері сәйкесінше 20 және 70 мс болатын симметриялы трапеция түріндегі импульсті құрастырамыз. Дискретизация жиілігі 1кГц ке тең. Сәйкес программа листингі төменде берілген.

```
clear; clc;
Fs=1000;  % Дискретизация жиілігі
A=10; % Амплитуда
T1=0.02; % Трапецияның жоғарғы негізінің өлшемі (секунд)
T2=0.06; % Трапецияның төменгі негізінің өлшемі (секунд)
% Импульс құратын уақыт интервалы
t=-0.04:1/Fs:0.04;
y=A*(T2*tripuls(t,T2)-T1*tripuls(t,T1))/(T2-T1);
% Трапеция түріндегі импульсті модельдеу
plot(t,y); % Сигнал визуализациясы
xlabel ('t'); ylabel ('s') % Осьтер белгілеулері
```

Нәтижесінде экранға трапеция түріндегі жалғыз импульс шығарылады (1.5 сурет).



1.5 сурет. Трапеция түріндегі импульс

Тікбұрышты импульстар тізбегін модельдеу

Тікбұрышты импульстар тізбегі `square` функциясы арқылы құралады. `Square` функциясы жалпы түрде 2 кіріс параметрін қабылдайды, уақыт мәндері векторы `t` және алынатын тізбектің қуыстылығын реттеуге мүмкіндік беретін `duty` параметрі:

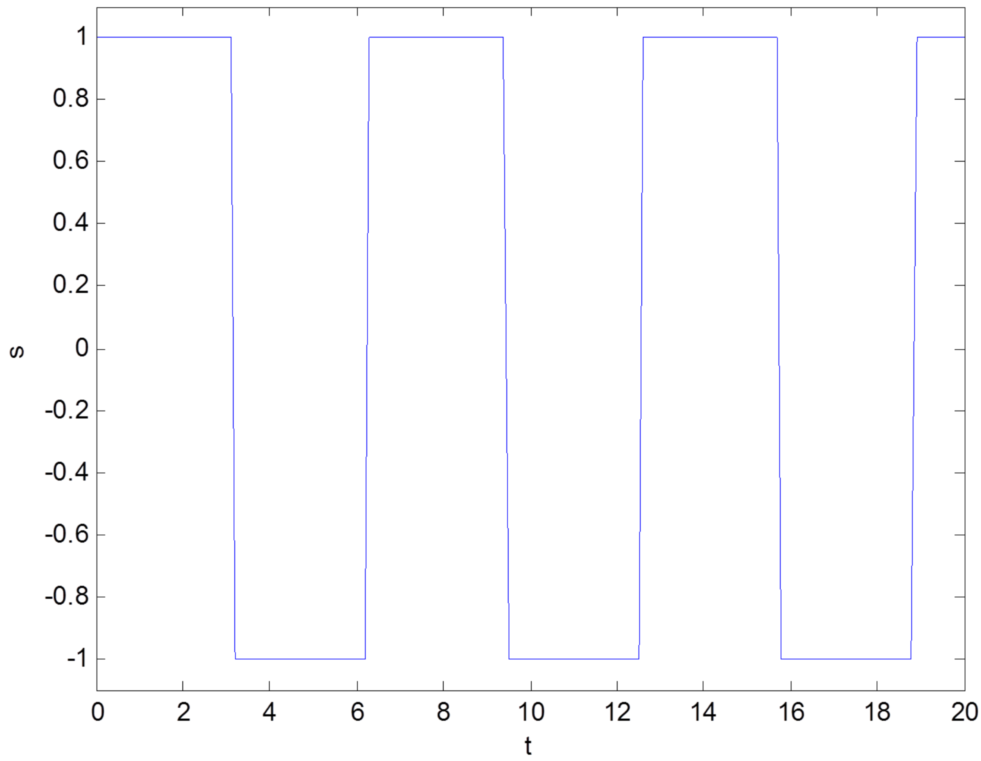
```
s=square(t,duty).
```

“`duty`” параметрі *қуыстылықтың* өзін емес (қуыстылық – периодтың импульс ұзақтығына қатынасы), оған кері өлшемділікті – толтыру коэфиценті (пайызбен), яғни импульс ұзақтығының периодқа қатынасын енгізеді.

Мысал ретінде 0 ден 20 с дейінгі уақыт интервалында 50% қуыстылыққа ие тікбұрышты сигналды модельдеу әдісін қарастырайық.

Тікбұрышты импульстер тізбегін модельдеуге арналған программа листингі.

```
clear; clc;
t=0:0.1:20; % уақыт интервалының сипаты
s=square(t,50); % тікбұрышты импульстерді модельдеу
plot(t,s) % сигнал визуализациясы
ylim([-1.1 1.1]); % ордината осьінің шекараларын анықтау
```



1.6 сурет. Тікбұрышты импульстер тізбегі.

Ара тәріздес сигналды модельдеу

Ара тәріздес сигналды модельдеу үшін `sawtooth` функциясы қолданылады, ол функцияның екі кіріс параметрі бар- уақыт мәндерінің векторы `t` және `width` параметрі, ол арқылы «кері жүріс» - сигнал деңгейі сызықты түрде 1 ден -1 ге дейін құлайтын аймақ ұзақтығын басқаруға болады. `Width` параметрінің мәні 0 ден 1 ге дейінгі мәнді қабылдау керек.

```
y=sawtooth(t,width)
```

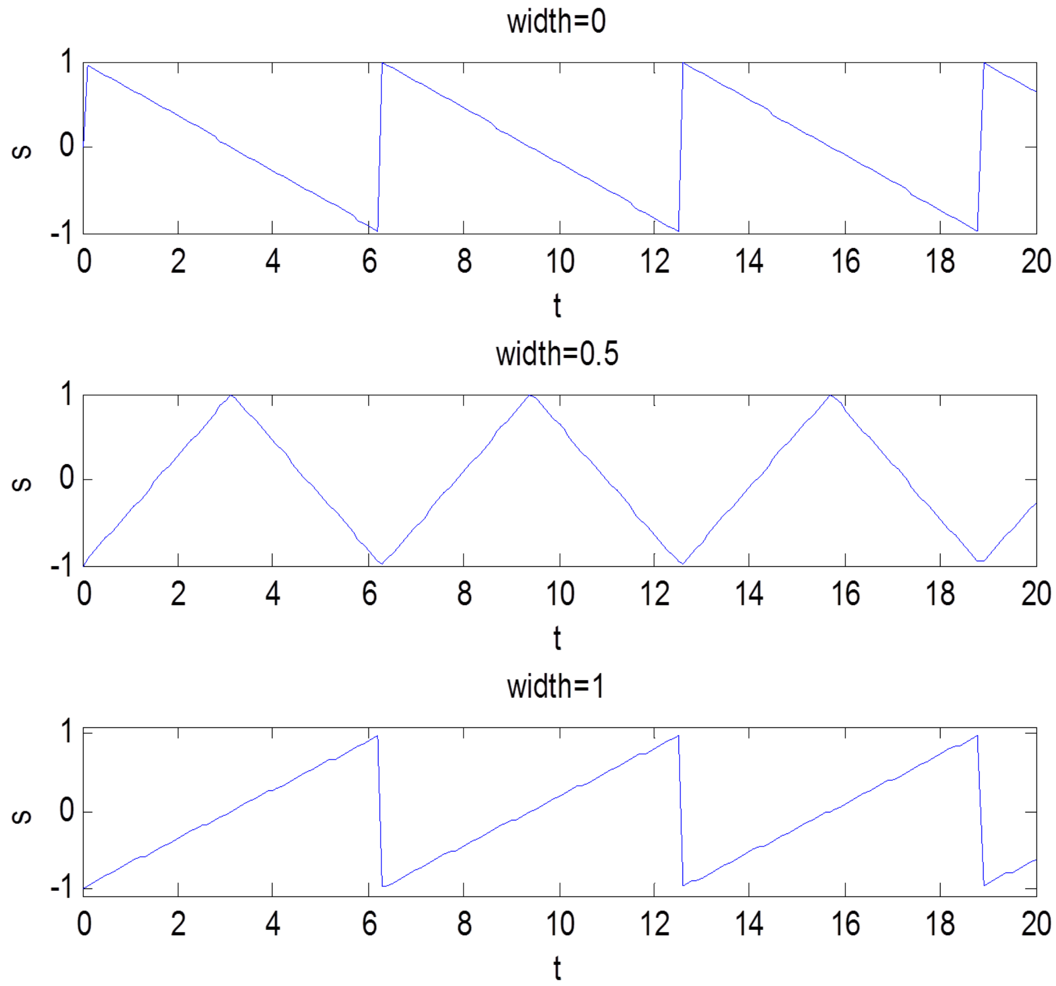
0 ден 20 с уақыт интервалында үшбұрышты сигналды құрастырамыз. `Width` параметрінің сандық мәнінің сигнал формасына әсерін зерттейміз.

Ара тәріздес сигналды модельдеу үшін программа листингі

```
clear; clc;
t=0:0.1:20;% Уақыттық интервал сипаттамасы
s_0=sawtooth(t,0); % Ара тәріздес сигналды модельдеу
% width=0 тең болғанда
s_0_5=sawtooth(t,0.5); % Ара тәріздес сигналды модельдеу
% width=0.5 болғанда
s_1=sawtooth(t,1); % Ара тәріздес сигналды модельдеу
% width=1 болғанда
subplot(3,1,1); % Графикалық терезені құру
```

```
plot(t,s_0);% width=0 тең болғанда сигнал визуализациясы
xlabel('t');ylabel('s') % Осьтер белгілеулері
title('width=0') % Бірінші графикке белгілеу
subplot(3,1,2); plot(t,s_0_5);
title('width=0.5');xlabel('t');ylabel('s')
subplot(3,1,3); plot(t,s_1);xlabel('t');ylabel('s')
title('width=1')
ylim([-1.1 1.1]); % Ордината осінің шекарасын анықтау
```

Программа орындалу нәтижесі 1.7 суретте көрсетілген.



1.7 сурет. `Width` параметрінің ара тәріздес сигнал түріне әсері

Гаусс қисығы бар радиоимпульсті модельдеу

Гаусс қисығы бар радиоимпульсті генерациялау үшін келесі функция қолданылады.

```
yi = gauspuls(t,fc,bw,bwr)
```

Бұл функция бірлік амплитудалы гаусс радиоимпульсін `t` уақыт интервалында қайтарады. `Bw` параметрі осы импульстың салыстырмалы спектр енін сипаттайды. Салыстырмалы спектр ені спектрлік шыңға қатысты `bwr` деңгейі бойынша децибелл арқылы өлшенеді. `Bwr` параметрінің мәні тек теріс бола алады, өйткені бұл параметр шыңдық деңгейге байланысты спектрлік функцияның қажетті төмендеуін енгізеді. Әдетте `bwr` параметр мәні -6дБ болып алынады. `Gauspuls` функциясы отсечка уақытын, яғни импульс амплитудасы қандай да бір берілген деңгейге дейін төмендеу уақытын анықтау үшін қолданылады. Бұл жағдайда функция келесі түрде енгізілуі қажет

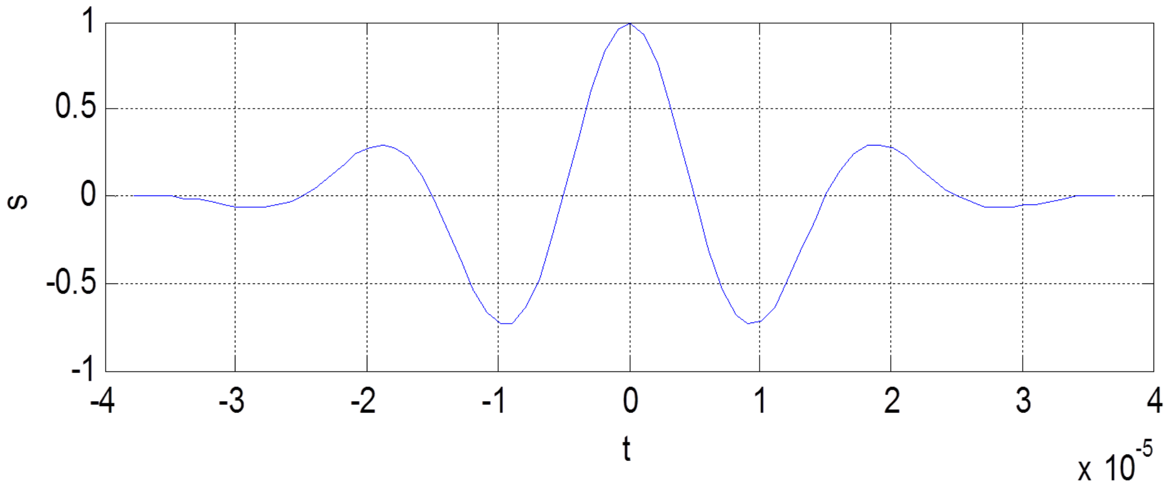
```
yi = gauspuls('cutoff',fc,bw,bwr, tpe)
```

бұл жерде `tpe`- амплитуда импульсы өзінің максимал мәніне қатысты төмендейтін деңгей (децибеллмен өлшенеді).

Тасушы жиілігі 50кГц, қатыстылық жолағы 60%, оның санақ жиілігі 1 Мгц болатын гаусс радиоимпульсының графигін саламыз. Амплитуда импульсы шыңдық мәнімен салыстырғанда -40дБ деңгейіне дейін төмендейтін уақыт интервалын қолданамыз.

Гаусс радиоимпульсін модельдеуге арналған программа листингі

```
clear; clc;
Fs=10^6; % Дискретизация жиілігі
tc = gauspuls('cutoff', 50000, 0.6, [], -40); % отсечка
уақыты
t = -tc : 1/Fs : tc; % уақыт мәндерінің векторы
s = gauspuls(t,50e3,0.6); % радио мәндерінің есептеулері
plot(t,s); xlabel('t'); ylabel('s'); grid on % 1.8 сурет
```



1.8 Сурет. Гаусс радиоимпульсі

Дискреттелген сигналды тұрғызу

Үздіксіз (аналогты) сигналдың дискретизациясы оның көптеген дискретті бөлшектерге түрленуінен тұрады. Matlab та дискреттелген сигналды визуализациялау әдістерін қарастырамыз.

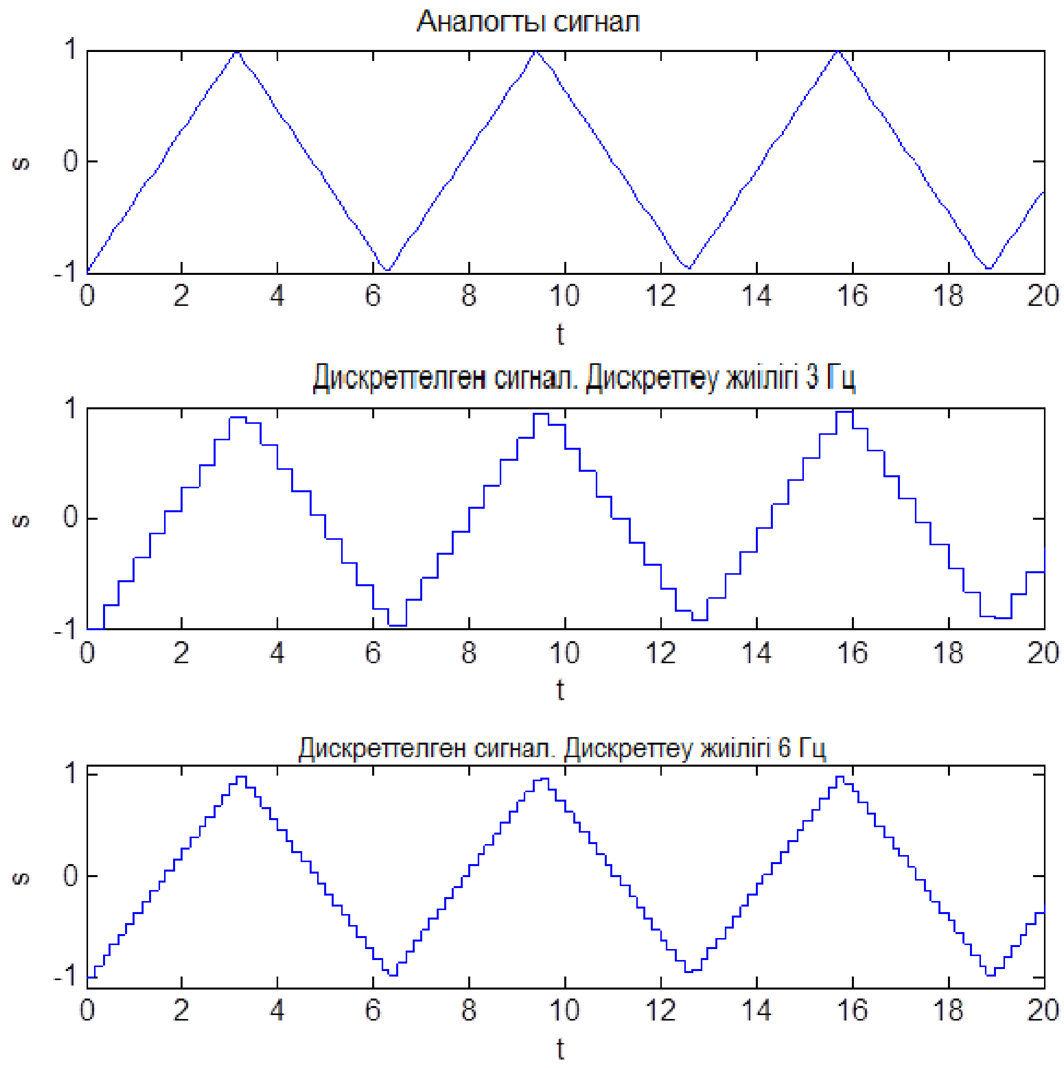
0 ден 20с дейін уақыт интервалында симметрияы үшбұрышты импульстердің дискреттелген тізбегін құрамыз, егер алғашқы аналогты сигналдың дискретелу жиілігі 3Гц ке тең болса. Дискреттелу жиілігін екі есе арттырған жағдайда графиктің түрленуін қарастырамыз.

Дискреттелген сигналды визуализациялау үшін `stairs` командасы қолданылады. Осы команданы қолдану арқылы салынған графиктер баспалдақты деп аталады. Сәйкес программа листингі және нәтижесі (1.9 сурет) төменде берілген.

Симметриялы үшбұрышты импульстердің дискреттелген тізбегін визуализациялауға арналған программа листингі

```
clear; clc;
t=0:0.1:20;% Уақыт интервалының сипаты
s=sawtooth(t,0.5); % Ара тәріздес сигналды модельдеу
subplot(3,1,1); % Графикалық терезені құру
plot(t,s); % Сигнал визуализациясы
xlabel('t');ylabel('s') % Осьтерге белгілеулер
title('Аналогты сигнал') % Бірінші графикке белгілеу

Fs=3; % Сигналдың дискреттелу жиілігі
t=0 : 1/Fs : 20; % Уақыт интервалы
% 1/Fs — дискреттеу периоды
s=sawtooth(t,0.5); subplot(3,1,2);
stairs(t,s); % Дискреттелген сигналды визуализациялау
xlabel('t');ylabel('s') % Осьтерге белгілеулер
title('Дискреттелген сигнал. Дискретизация жиілігі 3 Гц');
 t=0 : 1/(2*Fs) : 20; % Уақыт интервалы,
% 1/(2*Fs) — дискреттеу периоды
s=sawtooth(t,0.5); subplot(3,1,3);
stairs(t,s);xlabel('t');ylabel('s')
title('Дискреттелген сигнал.Дискретизация жиілігі 6 Гц');

ylim([-1.1 1.1]); % Ордината осінің шектеулерін анықтау
```

Программаны құру кезінде үшбұрышты импульстер симметриялы болу үшін `width` параметрі 0,5 ке тең болып алынды.



1.9 Сурет. Дискреттеу жиілігінің әртүрлі мәндеріндегі аналогты сигнал мен дискреттелген сигнал

1.9. суреттен көріп отырғанымыздай дискреттеу жиілігі, сонымен қоса оған кері өлшем болып табылатын дискреттеу периоды графиктің түріне ықпал етеді.

Тапсырма

1. Синусоидалы, сызықты өспелі, квадратты және экспоненциалды құлдырайтын сигналдарды тұрғызыңыз.
2. Signal Processing Toolbox кеңейту пакетінің жоғарыда көрсетілген программалар листингтері көмегімен 1.1-1.9 суреттерде берілген графиктерді тұрғызыңыз.
3. Синусоидалды, сызықты өспелі, квадратты, экспоненциалды құлдырайтын, тікбұрышты, трапеция тәріздес және аратәріздес сигналдарға амплитудасы әртүрлі ақ гаусстық шуылды қосыңыз.
4. Берілген сигналдардың дискреттелген графиктерін құрыңыз: синусоидалды, тікбұрышты және трапеция тәріздес.

Бақылау сұрақтары

1. Шуылданған синусоидалды сигнал тұрғызу методикасын сипаттаңыз. Сигналға ақ гауссты шуылды қосу үшін қандай функциялар қолданылады?
2. Ауыспалы жиілікті косинусоидалды сигналды тұрғызу үшін қандай функциялар қолданылады?
3. Сигнал спектограммасы дегеніміз не? Қандай тәсілдермен оны визуализациялауға болады?
4. Дирихле функциясы қандай математикалық теңдеумен өрнектеледі? Оның мәні неде? Matlab – та оны визуализациялау үшін қандай функция қолданылады?
5. Тікбұрышты импульстердің аналогты және дискретті реттілігінің модельдеу әлісін сипаттаңыз.
6. Аналогты және дискретті жалғыз трапеция тәріздес импульстың модельдеу әдісін сипаттаңыз.
7. Әртүрлі көлбеулі импульстермен тұрғызылған аналогты және дискретті аратәріздес сигналды модельдеу әдісін сипаттаңыз.
8. Келесі функицялар мақсаттарын түсіндіріңіз: clear, clc, plot, ylim, subplot, title, stairs.

Қолданылған әдібиеттер тізімі

1. Жанабаева З.Ж., Иманбаева А.К., Алмасбеков Н.Е. Компьютерное моделирование в радиофизике и электронике. – Алматы: Қазақ университеті, 2005. – 144 с.
2. Дьяконов В.П. Matlab 6.5 SP/7+Simulink 5/6. Основы применения. – М.:СОЛОН-Пресс, 2005. – 800 с.
3. Дьяконов В.П., Круглов В.С. Математические пакеты расширения Matlab: специальный справочник. – СПб.: Питер, 2010. – 480 с.
4. Ярмоленко В.И., Приоров А.Л. Сигналы в радиотехнике и телекоммуникационных системах: учеб.пособие. – Ярославль: ЯрГу, 2009. – 100 с.
5. Дьяконов В.П. Matlab и Simulink для радиоинженеров. – М.: ДМК Пресс, 2011. – 976 с.

Зертханалық жұмыс №2

MATLAB ОРТАСЫНДАҒЫ МӘЛІМЕТТЕРДІҢ АППРОКСИМАЦИЯСЫ МЕН ИНТЕРПОЛЯЦИЯСЫ

Жұмыстың мақсаты: Matlab ортасындағы мәліметтердің аппроксимацисы және интерполяциясын жүзеге асыру әдістерін зерттеу және осы әдістерді қолдану бойынша тәжірибе арттыру.

Қысқаша теориялық кіріспе

Мәліметтерді математикалық тұрғыдан өңдеу және өңдеу нәтижелерін файл түрінде сақтап отыру, сонымен қатар оларды графикалық түрде көрсету физикалық тәжірибелерге қатысты анализдер үшін үлкен маңызға ие болады. Matlab ортасына енгізуге болатын мәліметтерді математикалық өңдеу әдістеріне массивтер мәліметін өңдеу (массивтің минимады және максималды элементтерін табу, орташаны есептеу, орта ауытқуды есептеу, коррелляция коэффициентін есептеу және т.б.), мәліметтердің геометриялық анализі, Фурье түрендірілуі, орам және дискретті фильтрация әне т.б. жатқызылады. Сонымен қатар мәліметтерді көрсету әдісі мен олардың интерполяциясы және аппроксимациясына баса назар аударылады.

Мәліметтердің аппроксимациясы

Аппроксимацией деп көбінесе нақты берілмеген, оның мәліметтері жинақ немесе тәуелділік ретінде басқаның көмегі арқылы көрсетілген, қарапайым немесе біртекті тәуелділікерді қарастырамыз.

Аппроксимация күрделі тәуелділіктерді зерттеуге мүмкіндік береді, бұл үшін оларды қарапайым тәуелділіктер мен модельдерге айналдырамыз.

Аппроксимациялаушы функцияның графигі ағымдағы мәліметтердің түйіндік нүктелерінен өтуі шарт емес, бірақ оларды белгілі сөнуге алып келетіні рас.

Мәліметтер аппроксимациясының анағұрлым танымал түрі *полиномды аппроксимация* болып табылады. Бұл әдісті іске асырған кезде полиномды анықтау керек (көп мүше), және аппроксимацияланатын мәліметтер функциясын анықтау керек. Аппроксимацияның мұндай түрінде аппроксимацияланушы функцияның полином деңгейін беру керек, яғни аппроксимациялаушы көп мүшеге жататын бір мүше деңгейінің үлкенін таңдап аламыз. Аппроксимациялаушы ең үлкен полином деңгейін қалыпты жағдайда мәліметтер тізбегінің аппроксимация дәлдігін есептеу арқылы таңдап алады. Көп жағдайларда сыртық гармоникалық әрекеттің сипаттамалық гармоникасынының номеріне тең полином деңгейі таңдалынады. Егер полиномды аппроксимация кезінде бірлікке тең болатын полиномды функция деңгейін таңдап алатын болса, тапсырма регрессияға ұшырайды, яғни ең төмен шаршы әдісі іске асырылады. Регрессивті анализдің тапсырмасы математикалық формулаларды тәжірибелік мәліметтерді анағұрлым жақсы сипаттайтындай етіп таңдап алу болып табылады. Бұл әдіс статистикалық мәліметтерді өңдеген кезде, шуыл мен кедергілердің көлемі үлкен болған кездерде жиі қолданылады. Әдістің негізгі мәні функция мәнінің ауытқу шаршысының қосындысын төмендету немесе статистикалық мәліметтерді ізделініп отырылған айнымалының сандық мәнінен төмендету болып табылады. Басқаша айтатын болсақ, теориялық мәліметтердегі ауытқу квадратының қосындысы тәжірибедегіге қарағанда төмен болуы керек.

Регрессия теңдеуі мына түрде болады $y = f(x)$. Анағұрлым ең кіші квадрат жағдайында бұл тәуелділік $y = ax + b$ сызықты функцияны көрсететін болады. Осылайша, а және b өлшемдері сызықты тәуелділік коэффициентін сипаттайтын болады, ал y – ұсынылып отырылған аппроксимациялық теориялық тәуелділікті көрсететін болады. Тәжірибелік мәліметтердің теориялық мәліметтермен салыстырғандағы ауытқу шаршыларынының қосындысына тең болатын функцияны құрастырайық:

$$S = \sum_{i=1}^{n} \left(y - f(x) \right)^2 . \qquad (2.1)$$

Бұл функция өзінің минималды мәніне ұмтылатын болады.

Ең аз шаршылар әдісіне сәйкес а және b коэффициенттерін табу үшін S функциядан жергілікті туындыны табудың және оларды нөлге теңестірудің керегі жоқ:

$$\begin{cases} \sum \left(a + ax_i - f(x_i) \right) = 0, \\ \sum \left(b + ax_i - f(x_i) \right) \cdot x_i = 0 \end{cases} . \qquad (2.2)$$

Осы теңдеулер жүйесін шеше отырып, аппроксимациялаушы функцияның a және b коэффициенттерінің мәндерін есептеуге болады.

Matlab ортасындағы полиномды аппроксимация `polyfit` және `polyval` командаларының көмегімен іске асырылады.

`p=polyfit(x,y,n)` функциясы `p(x)` полиномының коэффициент векторын `n` деңгейіне қайтарады. Ол орташа квадраттық қателік арқылы `(x)` функциясын аппроксимациялайды.

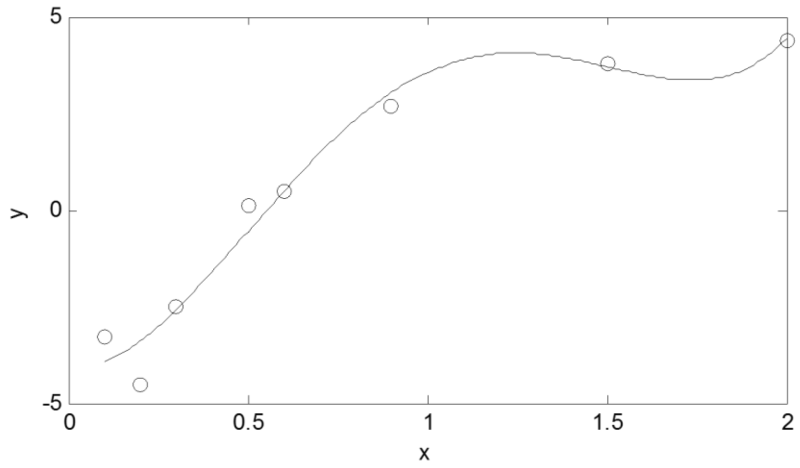
`y = polyval(p, x)` функциясы полином коэффициентінің мәнін қайтарады және осы полиномның мәнін берілген нүктелерде есептеп шығарады.

Мынадай координаталары бар тәжірибелік мәліметтер болсын деп қарастырайық $x = \{0.1,\ 0.2,\ 0.3,\ 0.5,\ 0.6,\ 0.9,\ 1.5,\ 2.0\}$ және $y = \{-3.3,\ -4.5,\ -2.5,\ 0.1,\ 0.5,\ 2.7,\ 3.8,\ 4.4\}$. Осы мәліметтердің аппроксимациясын 4-ші деңгейлі полиномын қарастырайық, сонымен қатар ең кіші шаршы әдісін қолданайық, яғни 1-ші деңгейлі полиномды да қолданамыз.

Мәліметтердің аппроксимациясын 4-ші деңгейлі полином арқылы және ең кіші шаршы әдісін қолдану арқылы іске асыруға арналған программа листингі

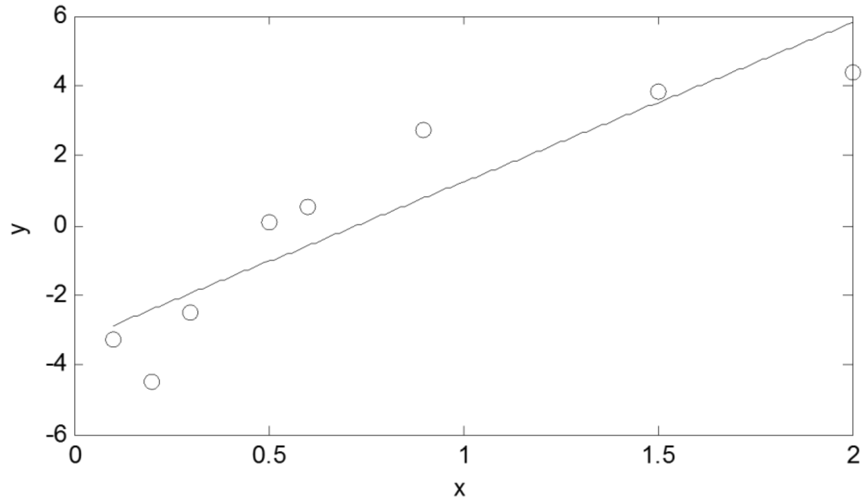
```
clear; clc;
% Нүктелер координаталарын енгізу
x=[0.1 0.2 0.3 0.5 0.6 0.9 1.5 2.0];
y=[-3.3 -4.5 -2.5 0.1 0.5 2.7 3.8 4.4];
% Бағандық график функциясын маркерлермен көрсету
plot(x,y,'ko')
% 'ko' сипаттамасында "o" белгіленудың орнына кез келген
% белгілену қолдануға болады, мысалы, "*" немесе "+".
% Сол кезде мәліметтер домалақ орнына жұлдызша тәріздес
% немесе "плюс" белгіленуімен көрсетіледі.
% Енді ең кіші квадраттар әдісіне жақын кестелі
% функциялы әртүрлі дәрежелі полиномдар
% коэффициенттерін есептеу жүреді
hold on
p4=polyfit(x,y,4); % 4 дәрежелі полином есептеу
t=[0.1:0.01:max(x)]; % аргумент анықтау үшін облыс мәнін беру
P4=polyval(p4,t); % t аргументінің полином мәнін анықтау.
plot(t,P4); % экранға аппроксимация мәндерін шығару.
xlabel('x'), ylabel('y') % осьтер аттары.
```

Жоғарыда көрсетілген программаны іске асыру нәтижесінде 2.1-суретте көрсетілген графикті аламыз.



2.1 Сурет. Тәжірибелік мәліметтерді 4-ші деңгейлі полином арқылы жақындату

Мәліметттерді бірінші деңгейлі полином арқылы жақындату ең кіші шаршы әдісін қолданумен тең болады. Бұл жағдайда аппроксимациялаушы түзу 2.2 суретте көрсетілгендей жүргізілетін болады.



Сурет 2.2. Ең кіші шаршы әдісі арқылы тәірибелік мәліметтерді жақындату

Мәліметтердің интерполяциясы

Мәліметтер интерпретациясы дегенде түйіндік нүктелер арасындағы функция мәнінің есептелуін түсінеміз. Осылайша, интерпретацияның негізгі міндеті - аралықтардағы оның түйіндік нүктелеріндегі тәуелділік мәліметтері арқылы қарастырылатын мәндерді бағалау болып табылады. Бұл үшін осыған сәйкес келетін функция қолданылады, оның мәні түйіндік нүктелерде осы нүктелердің координаталарымен сәйкес келеді. Мәліметтер интерполяциясының бірнеше түрлері бар.

Мысалы, *сызықты интерполяция* кезінде $y(x)$ тәуелділігінің түйіндік нүктелері түзу бөлектер арқылы бір-бірімен жай ғана байланысып тұрады. Сонымен қатар ізделініп отырылған нүктелер осы бөлшектерде орналасқан деп қарастырылады.

Сплайн-интерполяция кезінде бірнеше туындылар арқылы анықталудың барлық облысында үздіксіз болатын функцияларды қолдану қарастырылады. Сонымен қатар осы анықталу облысының әр бөлшегі бірнеше алгебралық көпмүше болып табылады.

Сплайн-интерполяцияны Matlab ортасында жүзеге асыру үшін `spline` функциясы қолданылады, ол мына түрде болады

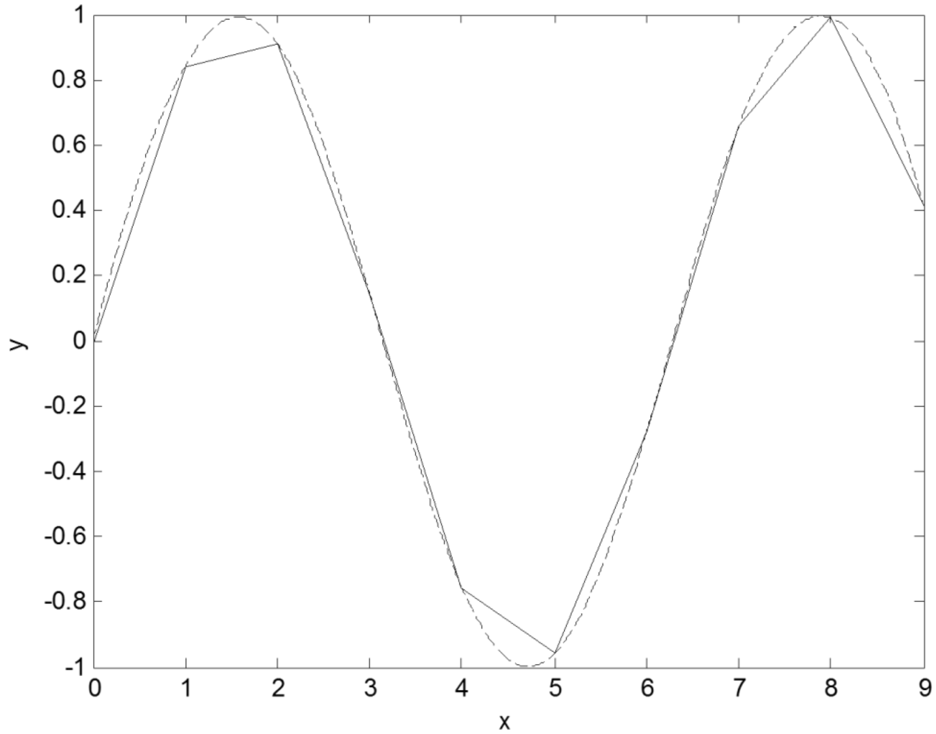
```
yi = spline(x, y, xi)
```

Мұндағы `x` ізделініп отырылған интерполяцияланбаған сигналдың абцисса өсінің координатасын қарастырады, `y` – `x` функциясына сәйкес келетін функция болып табылады. `spline` функциясы у функциясының мәнін `xi` нүктесінде анықталу облысының ішінде интерполяциялайды, бұл үшін кубтық сплайн қолданылады.

Осы функцияны гармоникалық сигналдың интерполяциясы үшін қолданатын мысалды қарастырайық, аталған сигналдың өзінің адекватты түсіндірілуі үшін жеткілікті есептеулері жоқ.

Синусоидалы сигналдың сплайн-интерполяциясы үшін қарастырылатын программаның листингі

```
clear;clc
% Синусоидалды сигналдарды тұрғызу
x=0:1:3*pi;
y=sin(x);
plot(x,y)
hold on

% Сигналдың сплайн-интерполяциясы
xmin=min(x); xmax=max(x); % максимал және минимал
% анықтау облысын анықтау
xx=xmin:0.01:xmax; % сплайн-интерполяция үшін массив
% нүктелері
yy=spline(x,y,xx); % сигналдың сплайн-интерполяциясы
plot(xx,yy,'r'); % интерполяция нәтижесін визуализациялау
xlabel('x');
ylabel('y')
```



2.3 Сурет. Бастапқы (пунктир) және интерполяцияланған (жалпақ сызық) гармоникалық сигналдар

Интерполяцияның нәтижесі 2.3 суретте көрсетілген. Осы суреттен көріп тұрғанымыздай, бастапқы сигнал абсциясса өсі бойынша үлкен қадамдар арқылы құрастырылған, сондықтан да қатыстық түрде қарастырғанда анағұрлым кіші есептеулер санына тең болып келеді. Нәтижесінде бастапқы гармоникалық сигнал сынық түрде болады (графиктегі жалпақ біртұтас сызық). Осы сигналдың жүзеге асуының сплайн-интерполяцияның нәтижесі (пунктирлі қисық) түйіндер арасындағы қосымша нүктелерді қосуға алып келеді, және нәтижесінде синусоида формасын қайтарады.

Сигналдардың интерполяциясы сигналдар есептеулерінің санын арттыру үшін қолданылуы мүмкін. Бұл тапсырма есептеулерін өшіру арқылы сигналдың дискреттелу жиілігін төмендететін сигнал децимациясына қарама-қарсы әрекет болып табылады. Басқаша сөздермен сипаттайтын болсақ, интерполяция процедурасы сигналдың дискреттелу жиілігін арттырады. Сонымен қатар интерполяция квантты сигналдардың шуылын кеміту үшін де қолданылады, яғни аналогты сигналды сандық сигналға айналдыру кезіндегі пайда болған қателіктерді азайту үшін қолданылады. Интерполяция процесі сигналдар есептеулерінің арасына нөл мәнін қоюға және сигналды төмен жиілікті сүзгі арқылы тізбектей өткізуге алып келеді.

Signal Processing Toolbox пакетінде қарапайым түрде бірөлшемді интерполяция төмендегі функциямен сипатталады

```
y = interp(x,r,l,alpha)
```

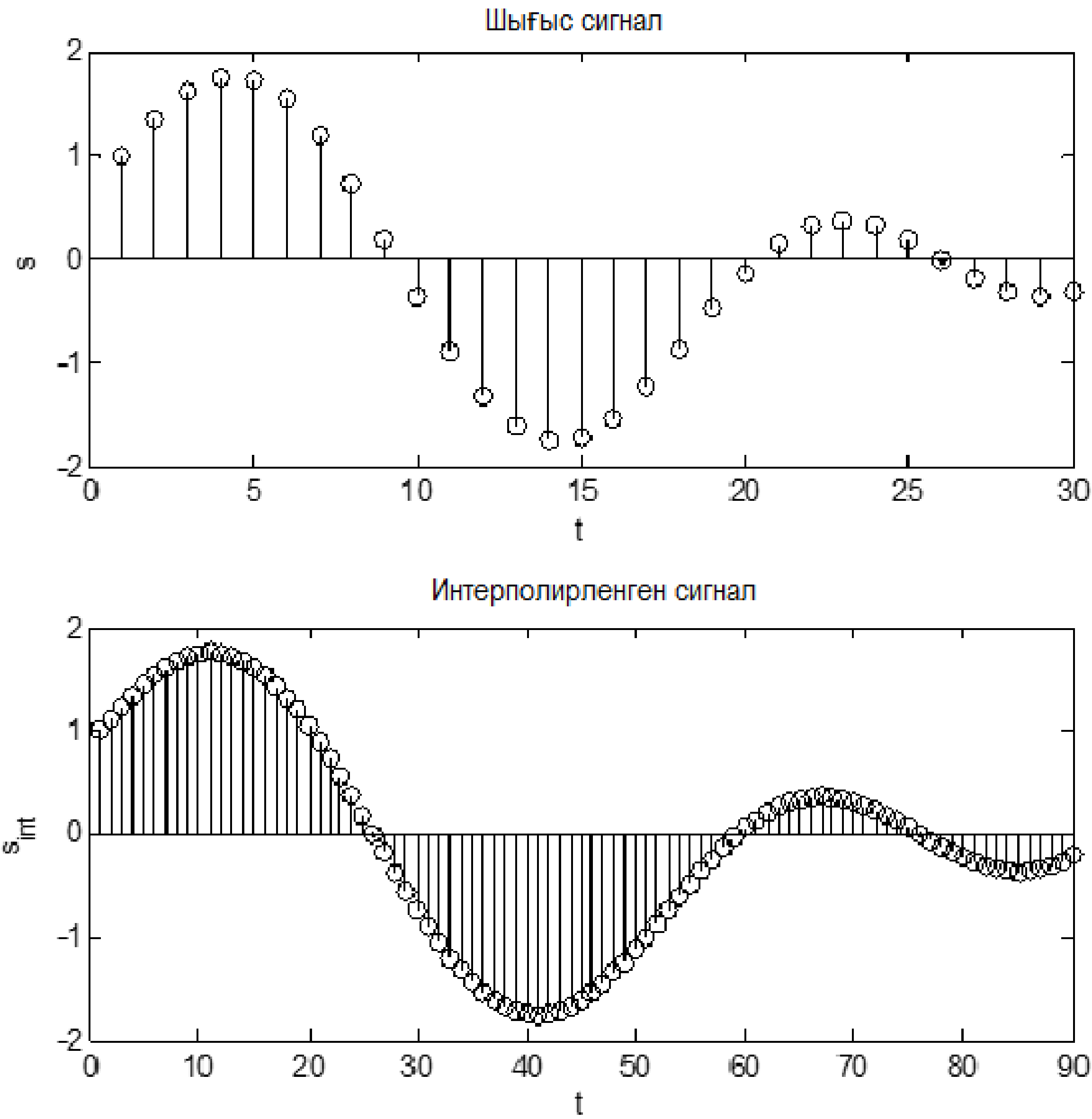
Бұл функция `x` сигналдың дискреттелу жиілігін `r` рет жоғарылатады, яғни мәліметтердің интерполяцияланған массиві `y` бастапқы `x` векторға қарағанда элементтерден `r` ретке үлкен болады. `l` параметрі қолданылып отырылған сүзгінің ретін анықтайды, ал alpha мәні - оның нормаланған кесу жиілігін анықтайды. `l` өлшемі бүтін мәнді болуы керек. Бұл параметрлер маңызды емес. Мына мән қолданылады: `l=4` және `alpha=0.5`.

Интерполяцияны ұйымдастыру үшін бірнеше қосжиілікті сигналдың векторы үшін және оның есептеулерінің графигін тұрғызу үшін программа жазамыз, одан кейін осы сигналдың интерполяциясын жүзеге асырамыз, бұл үшін `interp` функциясы қолданылады.

Қосжиілікті сигналды және оның интерполяциясын беру үшін жазылған программаның листингі

```
clear; clc; % Workspace және Command Window терезелерін
% тазарту
Fs=1000; % дискреттеу жиілігі
t=0:1/Fs:0.1; % уақыт векторын беру
s=cos(2*pi*30*t)+sin(2*pi*60*t); % екі жиілікті
% сигнал сипаттау
subplot(211) % графикалық терезені тұрғызу
N=30; % графикте көрсетілген нүктелер
stem(s(1:N)); % s массивінің алғашқы N
%элементтерінің сызықтар ретіндегі визуализациясы.
xlabel('t'); ylabel('s'); % осьтер аттары
title('Шығыс сигнал') % график тақырыбы
% Сигнал интерполяциясы
r=3; % интерполяция дискреттеу жиілігін 3 есе көтереді
s_int=interp(s,r); % s сигналының интерполяциясы
subplot(212) % графикалық терезені тұрғызу
stem(s_int(1:N*r)); % интерполирленген сигналдың кері есептеу
% визуализациясы
xlabel('t'); ylabel('s_i_n_t'); % осьтер аттары
title('Интерполирленген сигнал')% график аты
```

Осы программаның іске асу көрінісі 2.4-суретте көрсетілген.



2.4 Сурет. Қосжиілікті бастапқы
және интерполяцияланған сигнал

Matlab ортасында үшөлшемді көріністің *екі сызықты интерполяциясын* іске асыруға болады. Екі сызықты интерполяция дегенде сызықты интерполяцияның қос айнымалы функция үшін кеңеюін түсінеміз. Бірінші бір бағыт бойынша сызықты интерполяцияны іске асырамыз, мысалы, X өсі бойымен, одан кейін - оған перпендекуляр бағытта, яғни Y өсі бойынша іске асырылады. Екі сызықты интерполяцияның формулалары оның төрт биіктігі бойынша координаталар арқылы еркін тіктөртбұрыштағы қосымша нүкте координатасын табуға мүмкіндік береді. Одан кейін осы функцияны қалған барлық фигураға қатысты аппроксимациялайды. Matlab ортасында осы мақсатта `shading interp` функциясы қызмет етеді, ол көлемді фигураны визуалдау үшін жазылған командадан кейін жазылған болуы мүмкін. `Shading interp` функциясы әр ұяшықтың немесе шектерді түрлі түспен орнатады, олар тордың түйіндері бойынша екі сызықты интерполяцияны анқтайды.

Matlab ортасындағы екі сызықты интерполяцияның жүзеге асуын нақты мысал арқылы қарастырайық.

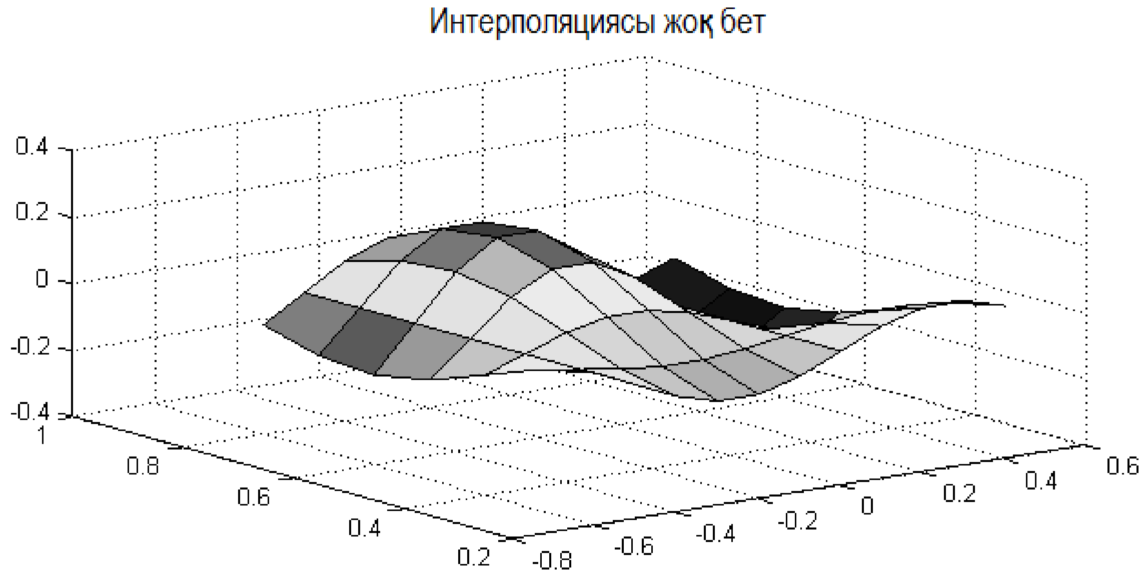
Қисық беткі аймақты $z(x,y) = \sin(2\pi x)\cos(1.5\pi y)\,y(1-y)$ қатынасы арқылы сипатталсын. Мұндағы x және y координатасы -0.6-дан 0.4-ке дейінгі және 0.2-ден 0.8-ге дейінгі аралықта сәйкесінше жатады. X өсі бойынша қадам 0.04-ке тең, ал y өсі бойынша қадам 0.03-ке тең. $z(x,y)$ тәуелділігін анықтайтын боялған беткі қабатты қарастырайық, одан кейін оған екі сызықты интерполяцияны қолданамыз.

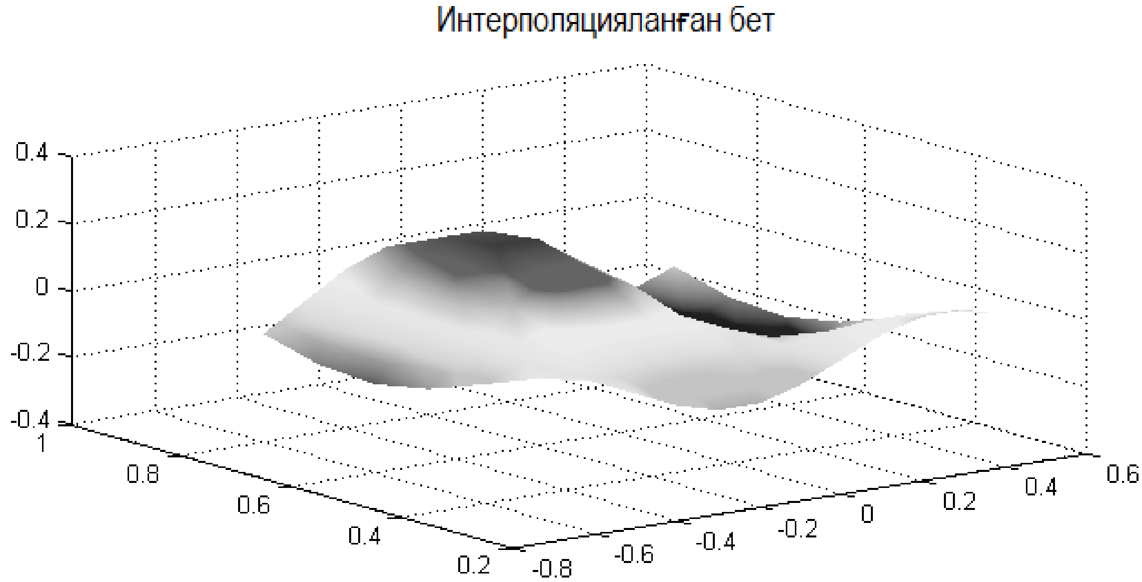
Сәйкес программаның листингі төменде көрсетілген.

Үшөлшемді объектті екі сызықты интерполяциялауға арналған программаның листингі.

```
clear;clc; % Workspace және Command Window терезелерін
% тазарту

[x,y]=meshgrid(-0.6:0.1:0.4, 0.2:0.1:0.8); % Бет тұрғызылған
% тор түйіндерінің координаттарының сипаттамасы

z=sin(2*pi*x).*cos(1.5*pi*y).*y.*(1-y);
% z координаталарының өлшемі

subplot(211); % Графикалық терезенің құрылуы
surf(x,y,z) % Беттің визуализациясы (интерполяция жоқ)
title ('Интерполяциясы жоқ бет')% График аты
subplot(212); % Графикалық терезенің құруы
surf(x,y,z);
shading interp % Бисызықты интерполяция беті
title ('Интерполяцияланған бет') % График атауы
```

Осы программаны жіберген кезде 2.5-суретте көрсетілген графикті аламыз.





2.5 Сурет. Үшөлшемді көріністің бисызықты интерполяциясы

2.5-сурет түрлі беткі қабаттардың арасындағы айырмашылықты қарастырады, ол берілген функция бойынша осы функция арқылы жасалған интерполяцияның және интерполяцияланған беткі қабаттың қолданылуынсыз құрастырылған.

Беткі қабаттардың қосөлшемді интерполяциясы үшін `interp2` функциясы қолданылуы мүмкін. Бұл функция мәліметтерді интерполяциялайды, аталған мәліметтер қосөлшемді торда `{x,y}` бірнеше беткі қабатты анықтайды; бұл кездегі шығыс массиві `zi` одан да кіші тор `{xi,yi}` арқылы анықталған болуы мүмкін.

`interp2` функциясы мына түрде жазылады:

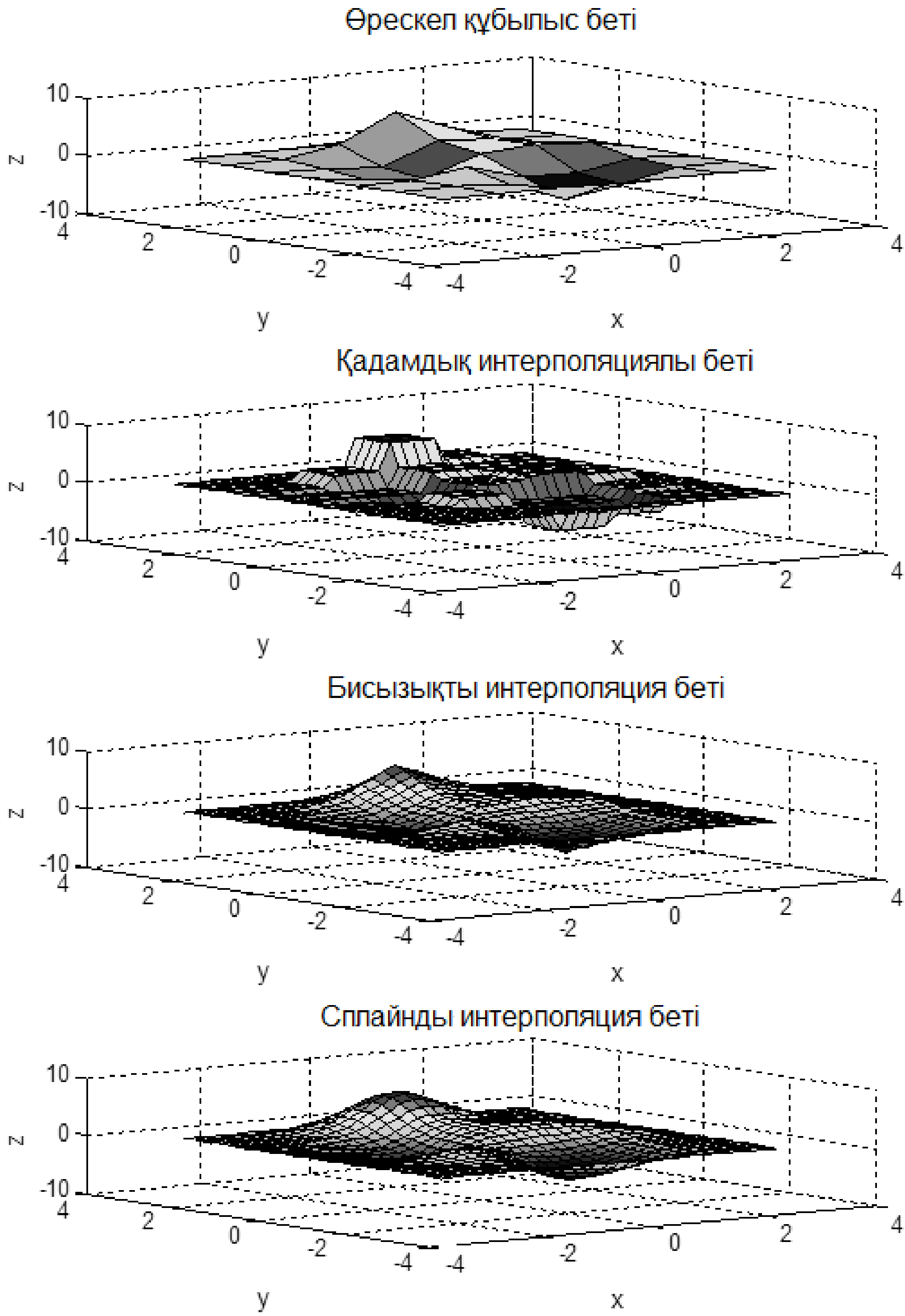
```
zi = interp2(x,y,z,xi,yi,'method')
```

`'method'` параметрі беткі қабаттың интерполяциясының түрін анықтайды. Осы параметрді `'nearest'` түрінде берген кезде баспалдақты интерполяция жүзеге асады, немесе оны басқаша атауға болады, жақын көрші әдісі арқылы жүзеге асатын интерполяция. Мұндай интерполяция кезінде аралық мән ретінде жақын функция мәні таңдалып алынады. `'bilinear'` таңдау бисызықты интерполяцияға алып келеді, ал `'bicubic'` – беткі қабаттың сплайн-интерполяциясына алып келеді.

Төменде беті қабаттың түрлі интерполяциясын жүзеге асыруға мүмкіндік беретін программа көрсетілген, онда бірнеше пик болады (2.6 сурет) және `interp2` функциясының көмегімен іске асады.

`interp2` *функциясының көмегімен беткі қабатты интерполяциялау үшін жазылған программаның листингі*

```
% Интерполирленбеген беттің құрылуы
[x,y]=meshgrid(-3.1:1:3.1);% x және y мәндер облысының
% анықтауы
z=peaks(x,y); % пиктердің құрылуы
subplot(411); % графикалық терезенің құрылуы
surf(x,y,z); % беттің визуализациясы
xlabel('x'); ylabel('y'); zlabel('z');
title('Өрескел құбылыс беті')

% Беттің қадамдық интерполяциясы
[xi,yi]=meshgrid(-3.1:0.2:3.1);% қадамды
% кішірейткен сайын, беттегі сызықтар санын көбейеді
zi1=interp2(x,y,z,xi,yi,'nearest'); % қадамдық интерполяция
subplot(412);surf(xi,yi,zi1) % беттің визуализациясы
xlabel('x'); ylabel('y'); zlabel('z');
title('Қадамдық интерполяциялы бет')

% Беттің бисызықты интерполяциясы
zi2=interp2(x,y,z,xi,yi,'bilinear'); % бисызықты интерполяция
subplot(413); surf(xi,yi,zi2) % беттің визуализациясы
xlabel('x'); ylabel('y'); zlabel('z');
title('Бисызықты интерполяция беті')

% Беттің сплайнды интерполяциясы
zi3=interp2(x,y,z,xi,yi,'bicubic'); % сплайнды интерполяция
subplot(414); surf(xi,yi,zi3) % беттің визуализациясы
xlabel('x'); ylabel('y'); zlabel('z');
title('Сплайнды интерполяция беті')
```



2.6 Сурет. `interp2` функциясы арқылы іске асатын беткі қабат интерполяциясы

Сплайнды интерполяцяны беткі қабаттарды тегістеу үшін қолданудағы оның артықшылықтарын оңай байқауға болады.

Matlab ортасы арқылы мәліметтердің тізбектілігінің гистограммасын құрастыру

Физикалық тәжірибелерді немесе компьютерлік модельдеулі іске асырған кезде көбінесе мәліметтердің үлкен көлемімен жұмыс жасауға тура келеді. Сондықтан да оларды көрнекі түрде көрсету маңызды болып саналады. Сандық мәліметтерді график немесе түрлі диаграмма ретінде көрсетуге болады. Келесідей диаграмма мен графиктердің мысалдарын келтіруге болады: сызықты диаграмма, бағанды диаграмма, жолақты диаграмма, кумулятивті қисық (уақыт өтуіне орай мәліметтер жинақталып отырады), пиктограмма (мәліметтер белгілі ретпен көрсетілген көрініс ретінде қарастырылады), логарифмдік диаграмма, дөңгелек диаграмма және т.б.

Гистограмма (*бағанды диаграмма*) – дегеніміз әрқайсысы жеке аралыққа қатысты болатын бағандар тізбегі, ал бағанның биіктігі - жағдайлар жиілігі немесе оның саны болып табылады. Бір горизонтальді шкаланы бір бөлек аралыққа оңынан және солынан алынған диапазон бойынша бөлу қарастырылады. Бағананың ортасы аралық ортасымен бірігеді, тәжірибеде оны контур формасында көрсетеді, ол үшін вертикаль сызықтарды төмен түсіреді. *Гистограмма* түсінігімен қатар таралу полигоны деген түсінік қоса жүреді. Таралу полигоны - нақ сол гисторграмма, бірақ сызықтар әр разрядты интервалдың бағанының ортасын байланыстарады. Оң және сол жақтағы разрядтарда жиілік таралу разрядынан бөлек жиілік нөл мәніне ие болады, сондықтан да таралу полигонын төменгі бағаланудан аз аралық орталығында және жоғары бағаланудан асыра горизонталь өске дейін жалғастырады. Гистограмманы тек бір таралу қолданылған кезде пайдалану оңай болады. Егер екі немесе одан көп таралуларды салыстыру керек болса, шиеленіскен көріністің алдын алу үшін полигонды пайдаланамыз.

Гистограмманы құрастырған кезде биндердің белгілі мәнін беру керек, мұнда әр бинге қанша сан сәйкес келетінін анықтап, мұны бағаналы немесе баспалдақты диаграмма ретінде суреттеу керек. *Бин* – дегеніміз айнымалы арқылы интервалға ұсақтау саны. Осыған орай таралу саналады және құрылады.

Matlab ортасында `hist` функциясы бар, оған `hist(y)` түрінде әрекет ететін болсақ, ол $y_{\max}$ және $y_{\min}$ аралығында тең бөлінген 10 бинді гистограмманы есептейді және бейнелейді. Сонымен қатар `hist(y)` функциясы екінші аргументке ие болуы мүмкін. Егер осы аргумент - бүтін сан болса, онда осы сан биндер санын анықтайды. Егер екінші аргумент - вектор болса, онда бұл вектор қолданылып отырылған биндердің орталықтарын анықтайды. Мұндай жағдайда биндер ортасы тең орнатылған болуы керек, ал осы орталықтардың координаталары өсіп отыратын ретпен орналастырылған болуы керек. Осы шарттардың кез-келген бірі бұзылса, нәтиже дұрыс болмайды.

Классикалық гистограмма Y векторының элементтер мәнінің M интервалға түсу санын сипаттайды, ол аталған санды диаграмманың бағаны ретінде көрсетеді. Гистограмманың мәліметтерін алу үшін `hist (y)` функциясы мына түрде жазылған болуы мүмкін:

- `N=hist(Y)` автоматты түрде таңдап алынатын 10 интервал үшін тура келу санынының векторын қайтарады. Егер `Y` – матрица болса, онда оның бағандары үшін тура түсу туралы мәліметтер массиві беріледі.
- `N=hist(Y,M)` жоғарыда қарастырылған командаға ұқсас, бірақ ол `M` интервалды қолданады (мұнда `M` – скаляр).
- `N=hist(Y,X)`. Бұл команда `Y` векторының элементттерінің интервалға түсу санын қайтарады, олардың орталықтары `X` векторының элементтері арқылы берілген.

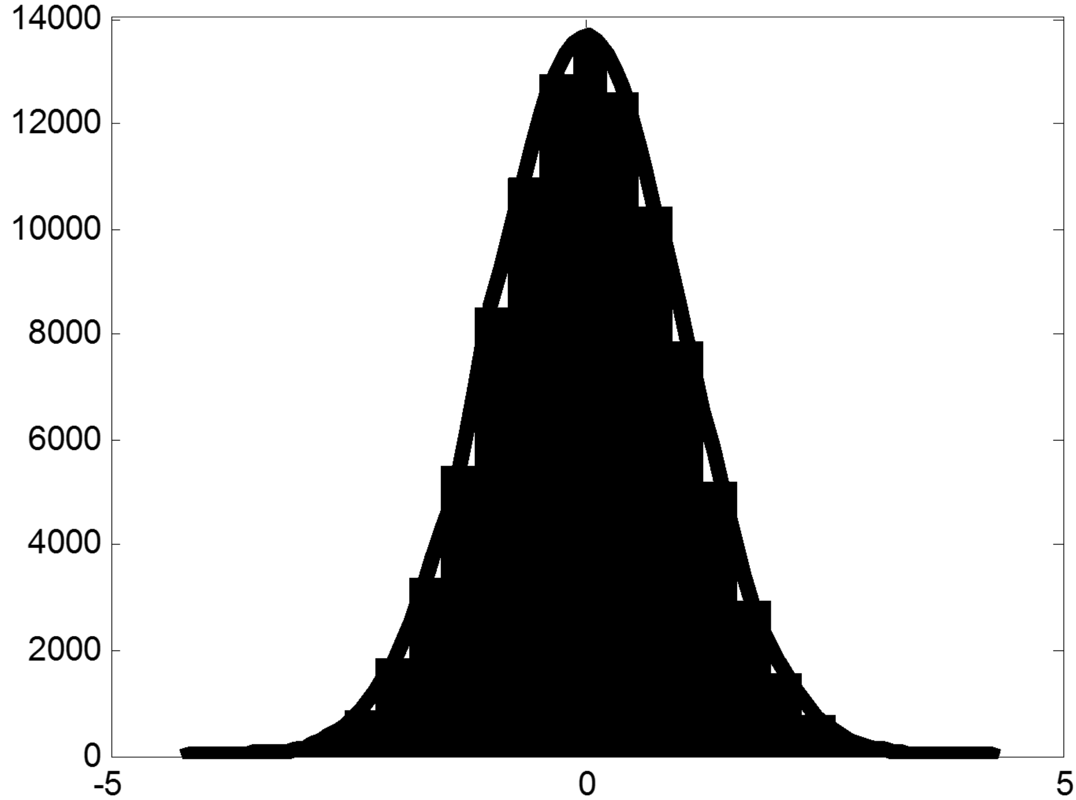
Matlab ортасында 100 000 кездейсоқ сандар үшін гистограмма құрастырайық, олар оң және теріс сандар үшін тең ықтималды болсын. Сплайн-интерполяция көмегімен осы гистограмманың иілгішін саламыз (гистограмманың қисығы таралу тығыздығының формасын сипаттайды).

100 000 кездейсоқ санның және оның қисығының гистограммасын тұрғызу үшін қарастырылатын программаның листингі

```
clear;clc;
y = randn(1,100000); % Кездейсоқ сандар массивінің
% генерациясы
N=25; % Гистограммадағы бағандар саны
hist(y,N); % Гистограмма тұрғызу
[heights,centers] = hist(y,N); % Гистограмма
% бағандарының центрларының координаттарын анықтау
Hold on; % Графикты ұстап тұру командасы
w = centers(2)-centers(1); % Бағандар орталарының
% арақашықтығы
t = linspace(centers(1)-w/2,centers(end)+w/2,N+1);
% Вектор мәндерінің қалыптастыруы
p = fix(N/2); % Шамаларды жуықтау
hold off   %  holdon ға кері команда
dt = diff(t); % t векторының дифференциалдануы
Fvals = cumsum([0,heights.*dt]); % Массив
% элементтерінің сомасын есептеу
F = spline(t, [0, Fvals, 0]); % Мәліметтердің сплайн-
% интерполяциясы
```

```
DF = fnder(F);  % Шыққан сплайндарды тексеруге арналған
кідіріс мәндерін есептеу
hold on;
fnplt(DF, 'r',5) % сплайн графигін тұрғызу.
% Жанама қалыңдығы 5 саны көрсетеді.
```

Осы программаның орындалу нәтижесі 2.7-суретте көрсетілген.



2.7 Сурет. Кездейсоқ сандар тізбегінің гистограммасы

Гистограмманың көрінісі оның көрінісі үшін таңдалып алынған нүктелер санына байланысты болады. Мәліметтерді көрсету дәлдігі гистограмма құралатын нүктелер санының көбеюімен арта түседі.

Тапсырма

1. Ең кіші шаршы әдісін қолдана отырып мәліметтер жақындауын 4-ші деңгейлі полиноммен құрастырыңыз және егер ағымдағы мәліметтер вольт-амперлік сипаттама алу кезінде алынған болса және төмендегідей мәндерден тұратын болса, ең кіші шаршы әдісі арқылы құрастырыңыз. Кернеу 0-ден 0,2 В аралығында өзгеріп отырсын. Ток күшінің мәні : 0; 0; 0.2 А; 0 ;0.4 А; 0,8 А; 1.1 А; 0.9 А; 1 А; 1.1 А; 0.9 А; 1.5 А; 1.3А; 1.4 А; 1.8 А; 1.9 А; 2.4 А; 2.1 А; 2.5 А; 2.6 А; 2.4 А; 2.7 А; 2.8А; 3.0 А; 2.9А; 3.0 А. Графикке тәжірибеден алынған мәліметтерді жұлдызша ретінде енгізіңіз.
2. 1000 оң кездейсоқ сан тізбегінің гистограммасын және оның қисығын құрыңыз. Аналогты график алыңыз, ол үшін 100 ретке үлкен кездейсоқ санды алыңыз.

3. $n_{i+1} = \left\{ n_i + sign\left(\xi_i\right)\cos^2\left(a + p \cdot i\right)\right\} \left| n_i \right|^{-1/\gamma}$ қатынасына сәйкес сипатталатын көріністің i қадамына тәуелді n_i электрондар концентрациясының таралу гистограммасын салыңыз. Қатыстық бірліктегі электрондардың бастапқы концентрациясын $n_1 = 1$ тең деп аламыз, қоспа концентрациясын – $a = 1/2$, кемтіктер концентрациясын – $p = 1$, скейлинг көрсеткішін – $\gamma = 2.806$, $i \in \{1, \dots N\}$, мұндағы $N =$ 500. ξ_i - әр i мәні үшін түрлі мәнге ие болатын кездейсоқ сан,

Бақылау сұрақтары

1. Мәліметтер аппроксимациясының маңызы неде?
2. Мәліметтер интерполяциясы дегеніміз не?
3. Функцияларды полином арқылы жақындату әдісі қалай түсіндіріледі?
4. Ең кіші шары әдісін түсіндіріңіз.
5. Сплайн нені қарастырады?
6. Полиномды жақындау үшін Matlab ортасында андай функциялар қолданылады?
7. «Гистограмма», «таралу полигоны», «бин» мағынасын түсіндіріп беріңіз.
8. Matlab ортасында гистограмма құру үшін қандай функция қолданылады?
9. Гистограмманың қисығын қалай түсіндіруге болады және ол Matlab ортасында қалай тұрғызылады?

Қолданылған әдібиеттер тізімі

1. ДьяконовВ.П. Matlab 6.5 SP1/7+Simulink 5/6. Основы применения. – М.: СОЛОН-Пресс, 2005. – 800 с.
2. МэтьюзДж.Г., Финк К.Д. Численные методы. Использование Matlab. – М.: Изд. дом «Вильямс», 2010. – 720 с.
3. Сергиенко А.Б. Цифровая обработкасигналов. СПб.: БХВ_Петербург, 2013. – 768 с.
4. Сато Ю. Без паники! Цифровая обработка сигналов.– М. Додека-XXI, 2010. – 178 с.
5. Поршнев С.В. Компьютерное моделирование физических процессов в пакете Matlab. – М.: Горячая линия – Телеком, 2008. – 593 с.

Зертханалық жұмыс №3

ФРАКТАЛДЫ ОБЪЕКТІЛЕРДІ ЖОБАЛАУ

Жұмыстың мақсаты: фракталды объектілерді зерттеу және оларды Matlab ортасында құру, сонымен бірге олардың фракталдық өлшемділігінің есептеу.

Қысқаша теориялық кіріспе

Фракталдар деп қатты тілімденген пішінге және өзұқсас қасиеттерге ие сызықтар, беткейлер, кеңістік денелері сияқты геометриялық объектілерді атаймыз. Фрактал сөзі латын тілінің fractus деген *сөзінен шыққан және бөлшектік, сынық деп аударылады. Өзұқсастық фракталдың негізгі сипаттамасы ретінде оның масштабтың кең диапазонында бірқалыпты орналасқанын білдіреді. Фракталдың кіші фрагменттері ұлғаю кезінде үлкендерге ұқсас болады. Идеалды жағдайда* мұндай ұқсастықта фрактал обьектісі созылуға сай инвариантты болып келеді, яғни оған дилатационды симметрия тән. Ол фракталдың негізгі геометриялық ерекшеліктерінің масштабын ұлғайтқанда өзгермеуін шамалайды.

Евклид кеңістігінде қандай да бір шектелген аймақты қамтитын фракталды объектіні қарастырайық. Ол оны құрудың қандай бір кезеңінде осы аймақта үлестірілген көптеген $N \to \infty$ нүктелерді білдіреді. Барлық аймақты δ қабырғасы және δ^d көлемі бар текшелік ұяшықтарға бөлеміз. Азайтқан кезде аймақты алып жатқан $N(\delta)$ ұяшықтардың δ саны дәрежелік заң бойынша өзгереді

$$N(\delta) \cong \frac{1}{\delta^D}, \tag{3.1}$$

D Хаусдорф немесе *фракталдық өлшемділік* болып аталады. Ара қатынасты (3.1.) логарифмдеу және δ нөлге тарту арқылы жазуға болады

$$D = -\lim_{\delta \to 0} \frac{\log N(\delta)}{\log \delta}. \tag{3.2}$$

Логарифмді бірліктен өзгеше кез-келген оң негіз бойынша, мысалы, негізі 10 немесе $e \approx 2{,}7183$ бойынша қарастыруға болады. (3.2.) Формула *D* фракталдық өлшемділіктің математикалық шамасы болады. Соған сәйкес *D* шамасы осы объектінің жергілікті сипаттамасы болады. Фракталдық өлшемділік бөлшектік шама болады.

Бір өлшемді фракталды объектілер *өзұқсастық* немесе *масштабтық инвариантты қасиеттерге* ие: объектінің бір бөлігі оның тұтас бөлігі тәрізді. Егер анықтаушы айнымалылар саны бірден көп болса және осы айнымалылардың ұқсастық коэффициенті әртүрлі болса, онда мұндай фракталды объектілер өзаффинді деп аталады. Күрделі генератордан шыққан сигналдардың қисық пішіні, жұқа қабыршық шалаөткізгізгіштің кеңістіктік және уақыттық энергетикалық спектрлері т.б. өзаффинді фракталдар болуы мүмкін.

Мультифракталдар – бейсызық физиканың аса күрделі объектілері. Табиғатта өлшемдердің аддитивті қосылатын, өлшеуге болатын шамалардың (ұзындық, аудан, көлем, масса, заряд, энергия және т.б.) кеңістіктегі таралуы біркелкі емес, алмасу қасиеті бар болады. Осындай құбылыстардың жалпы заңдылықтары мультифракталдар теориясымен тағайындалады. Мультифракталдардың жалпы қабылданған анықтамасы жоқ. Мультифракталдардың дәл, жалпы анықтамасының логикалық компоненттері болуға лайықты бірнеше тұжырымдарды келтірейік:

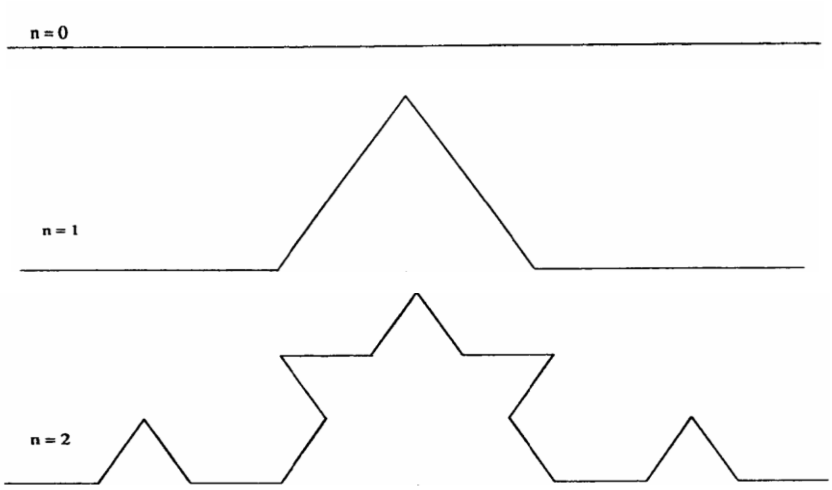
– Геометриялық тұрғыдағы өлшемнің алмаспалы таралуы мультифракталдық өлшеммен байланысты.
– Мультифракталдық объект фракталдық өлшемділіктер жиынымен сипатталады.
– Құрылымдық-иерархиялық өзара байланысты фракталдық объектілер мультифрактал құрайды.
– Мультифракталдық өлшемділік немесе жалпыланған өлшемділік Реньи формуласымен анықталады.

$$D_q = \frac{1}{q-1}\lim_{\delta\to 0}\frac{\ln N(q,\delta)}{\ln\delta}, \qquad (3.3)$$

мұнда δ – жиын ұяшығының сипатты мөлшері, $N=(q,\delta)$ – бастапқы жиын бөлігін сипаттау үшін қажетті ұяшықтардың ең аз саны, q – мультифрактал моментінің реті, ол $-\infty \le q \le \infty$ мәндерін қабылдайды. q шамасының физикалық мәнінің түсіндірмелерінің бірі - оның кері температураға (оңына да, терісіне де) эквивалеттігінде.

Триадты Кох қисық сызығы – бұл фракталды объектілер мысалдарының бірі (3.1-сурет). Кох қисық сызығын құру $L(1)=1$ бірлік ұзындықтың кесіндісінен басталады. Осы бастапқы сызық бөлім болады және кез-келген көп бұрышпен, мысалы, қабырғалары тең үшбұрышпен, төртбұрышпен ауыстырылуы мүмкін. Бөлім – бұл Кох қисық сызығының 0-дік буыны. Кох қисық сызығын құру жалғасады: әр бөлім буынын 3.1-

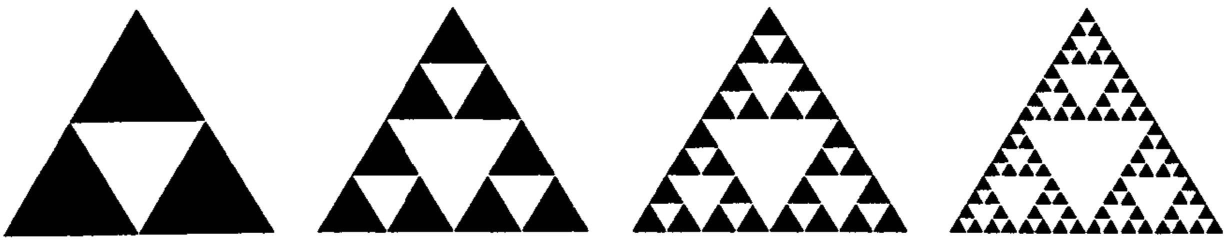
суретте n = 1 арқылы белгіленген түрлендіруші элементпен алмастырамыз.



3.1 Сурет. Триадты Кох сызығының құрылуы

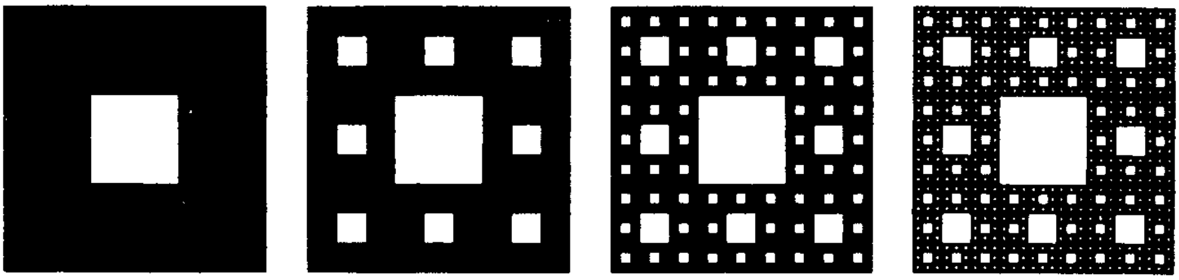
Біз осындай алмастыру нәтижесінде 1-ші буын – төрт түзу қисық сызықты аламыз, әрқайсысының ұзындығы 1/3. 1-ші буын қисық сызығының ұзындығы L(l/3) = 4/3 шаманы құрайды. Келесі буын әрбір түзу сызықты буынды ауыстыру кезінде кішірейтілген түрлендіруші элемент кезінде пайда болады. Нәтижесінде $N = 4^2 = 16$ буыннан тұратын 2-ші буын қисық сызығын аламыз, олардың әрқайсысының ұзындығы $\delta = 3^{-2}$ = 1/9. 2-ші буын қисық сызығының ұзындығы $L(l/9) = (4/3)^2 = 16/9$ тең. Алдыңғы буынның барлық буындарын кішірейтілген түрлендіруші элементпен ауыстыру арқылы жаңа қисық сызық буынын аламыз. n-ші буынның қисық сызығы кез-келген соңғы n кезінде алдыңғы фрактал деп аталады. Триадты Кох қисық сызығының фракталды өлшемділігі Кох $D = \ln 4/\ln 3 \approx 1.2628$.

Фракталдың басқа мысалы ретінде Серпинский үшбұрышы бола алады алғашқы төрт буыны 3.2-суретте келтірілген.



3.2 Сурет. Серпинский үшбұрышы. Бөлім – барлық ішкі нүктелері бар үшбұрыш. Түрлендіруші элемент бөлімде орталық үшбұрышты болдырмайды.

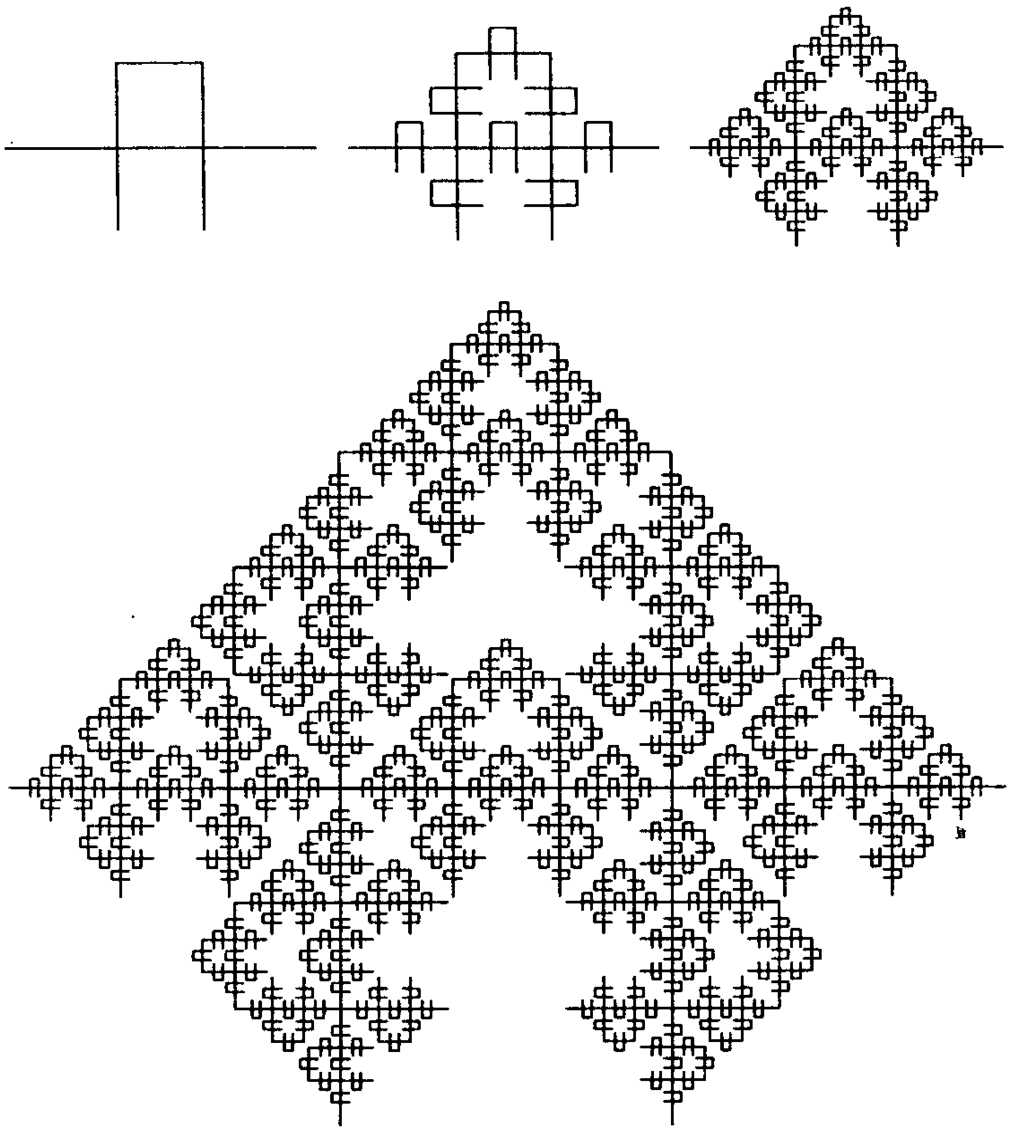
Ішкі нүктелерімен бірге қарастырылатын үшбұрыштың түрлендіруші элементін қолдану кезінде N=3 r=½ коэффициентімен кішірейтілген үшбұрышпен алмастырылады. Ұқсастық өлшемділігі осы жағдайда D = ln 3/ln 2 = 1,58... тең. Серпинский салфеткасымен *Серпинский кілемшесі* деп аталатын басқа қисық тығыз байланысты. Ол 3.3-суретте көрсетілген. Алдыңғы фракталдардың шексіз көп буындары фракталдық қисық сызықты тудырады. Алдыңғы фракталдардың «қалың» (қара) учаскелері шекті фракталдық қисық сызыққа көшу кезінде жоғалады, ал Серпинский кілемшесінде тесіктердің толық периметрі шексіз болады.



3.3 Сурет. Серпинский кілемшесінің құрылуы.
Бөлім – төртбұрыш, ал түрлендіруші элемент (сол жақтағы) бөлімнен ұқсастықты құр
(қысу) арқылы алынған $N = 8$ төртбұрыштан тұрады, $r = 1/3$ ұқсастық коэффициентіме
Ұқсастық өлшемділігі $D = \ln 8/\ln 3 \approx 1{,}89$.

Серпинский қисық сызықтары көптеген физикалық құбылыстардың моделі ретінде пайдаланылды. Зерттеулер алюминий қабықшалардың үлгілерінде Серпинский салфеткасының 10-шы буынындағы алдыңғы фракталының құрылымы барын көрсетті.

3.4-суретте бейнеленген құрылым Мандельброт пен Гивенге жатады. Осы қисық сызықтың түрлендіруші элементі түзу сызықты кесіндіні ұзындығы $r=$ 1/3 болатын бөлікке бөліп, оларды үш бөліктен тұратын, екі тармақ қосылатын петляға қосады. Мандельброт пен Гивен осы қисық пен перколяциялық кластерлер моделі ретінде ұқсас қисық сызықтарды пайдаланды. Мандельброт-Гивен қисық сызығы петлялар мен тармақтардың (дөңес) барлық ықтимал мөлшерлерінің болуымен қызықты. Әрбір итерация кезінде (алдыңғы фракталдардың бірінші буынынан келесі буынына көшу) түрлендіруші элемент әрбір түзу сызықты буынды алдыңғы фракталда $r=$ 1/3-мен кішірейтілген N = 8 буынға алмастырады. Мандельброт-Гивен қисық сызығының D = ln8/ln 3= 1,89... фракталдық өлшемділігі болады.



3.4 Сурет. Мандельброт-Гивен қисық сызығын құрудың тізбекті кезеңдері. Түрлендіруші элементтің биіктігі қисық сызық құрылымын қадағалауға болатындай біршама кішірейтілген

Төменде Matlab компьютерлік ортаның көмегімен кейбір классикалық фракталдық объектілерді құру үшін бағдарламалар листингілері төменде келтірілген.

Триадты Кох қисығының түрлі буындарын құруға арналған бағдарламаны қарастырамыз.

```
function z=Koch(N)
% N-шы буындағы Кох қисығын тұрғызу мақсатында
% координаталарды есептеу
x1=0;y1=0;x2=1;y2=0; % бастапқы координаттар
figure(1);% графикалық терезені жасау
axis([0 1 0 1]); % сұлбадағы осьтер шекарасын анықтау
```

```
hold on; %  берілген сұлбаға мәліметтерді қосу
set(gca,'xtick',[],'ytick',[]); % графикалық обьектілердің
% құраушыларына мағыналарды беру
% gca – осьтің көрсеткіші
% xtick – координаталардың өспелі мағыналарының векторы
% [] – ось бойынша түзетулерді жасыруға арналған команда
set(gca,'XColor','w','YColor','w') % түстерді беру

Coord(x1,y1,x2,y2,N); % қисық нүктелерінің координаттары
function z=Coord(x1,y1,x2,y2,n)
     if n>0
% Кох қисығын құруға арналған координат мағыналарының
% есептемесі
         dx=(x2-x1)/3; dy=(y2-y1)/3;
         x1n=x1+dx;y1n=y1+dy;
         x2n=x1+2*dx;y2n=y1+2*dy;
         xmid=dx/2-dy*sin(pi/3)+x1n;
         ymid=dy/2+dx*sin(pi/3)+y1n;
         Coord(x1,y1,x1n,y1n,n-1);
         Coord(x1n,y1n,xmid,ymid,n-1);
         Coord(xmid,ymid,x2n,y2n,n-1);
         Coord(x2n,y2n,x2,y2,n-1);
     else
         r1=[x1 y1]; r2=[x2 y2]; R=cat(1,r1,r2);
         plot(R(:,1),R(:,2),'Color','k');
     end;
```

N-шы буынның (N = 1, 2, 3…) Кох қисық сызығын көрнекілеу үшін, Command Window-дан немесе басқа m-файлдан келесі пәрменді енгізу қажет:

```
N=1; Koch(N);
```

N шамасын түрлендіре отырып, 3.1-суретте көрсетілген ұқсас суреттерді алуға болады.

Басқа классикалық фракталдың түрлі буындарының – Серпинский үшбұрышты салфеткасының суретін құрайық.

Серпинский үшбұрышты салфеткасын тұрғызуға арналған бағдарлама листингі.

```
function  z = Serpinsky(N)
%  Серпински салфеткасының суретін қайтаруға арналған функция
% N -  фракталдар буыны
%  теңбүйірлі үшбұрышты бас координаттарын беру
```

```
x1=0;   y1=0;   x2=1;   y2=0;   x3=0.5;   y3=sin(pi/3);
h=figure(1); % графикалық терезенің инициализациясы
hold on; % кескіндерді бір графикалық терезеге салу режимін
% қосу
fill ([x1 x2 x3],[y1 y2 y3],'k'); % теңқабырғалы үшбұрышты
% бейнелеу
set(gca,'xtick',[],'ytick',[]) ; % осьтерді цифрлау режимін
% өшіру
set(gca,'XColor','w','YColor','w');   %   остерьді салуға
% арналған түстерді орнату
Simplex(x1,y1,x2,y2,x3,y3,0,N);   % ақ түсті теңқабырғалы
% үшбұрышты салуға арналған функция

hold off; % кескіндерді бір графикалық терезеге салу
%   режимін өшіру
function  z=Simplex(xl,yl,x2,y2,x3,y3,n,N) % ақ түсті
% теңқабырғалы үшбұрышты салуға арналған рекуурсивті функция

if n<N
%   теңқабырғалы үшбұрыштың бас координаттарын беру
dx=(x2-xl)/2;
dy=(y3-yl)/2;
xln=xl+dx;
yln=yl;
x2n=xl+dx+dx/2;
y2n=yl+dy;
x3n=xl+dx/2;
y3n=yl+dy;
fill([xln x2n x3n],[yln y2n y3n],'w'); % дәл осы теңқабырғалы
%  үшбұрышты салу
n=n+1;
%  рекурсия
Simplex(xl,yl,xln,yln,x3n,y3n,n,N);
Simplex(xln,yln,x2,y2,x2n,y2n,n,N);
Simplex(x3n,y3n,x2n,y2n,x3,y3,n,N);
end
```

Серпинский салфеткасын көрнекілеу, мысалы, 3-ші буындағы,

```
N=3 Serpinsky(N)
```

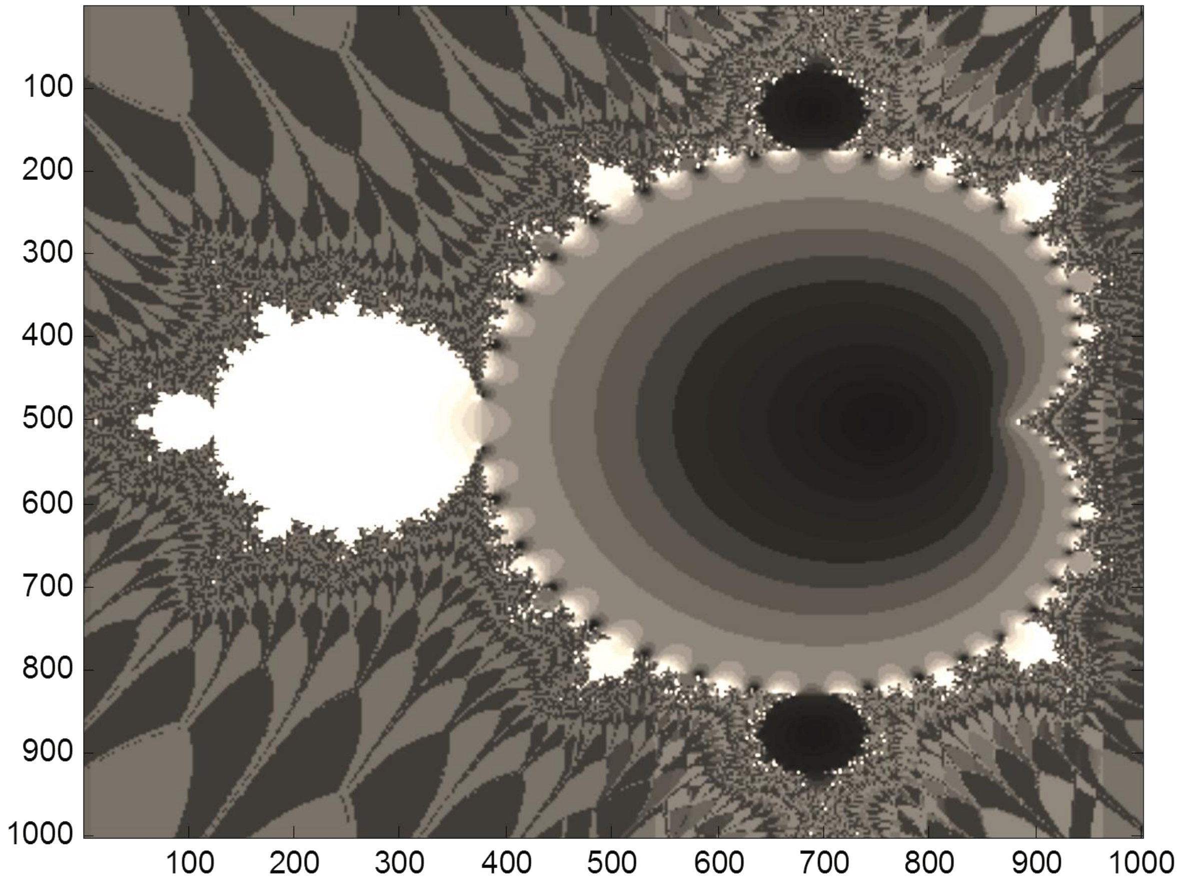
командасымен жүзеге асырылады;

Мандельброт жиынтығының түсті суретін құру үшін келесі бағдарлама пайдаланылуы мүмкін.

Мандельброт жиынтығын құруға арналған бағдарлама листингі

```
% Сурет жағындағы нүктелер саны
% және Мандельброт фракталдық жиынтығын
% есептеудегі итерация саны
npts=1000;
niter=51;
% z = 0 құру (нақты және жанама бөліктер)
zRe=zeros(npts,npts);
zIm=zeros(npts,npts);
% k константасын құру (нақты және жанама)
kRe=repmat(linspace(-1.5,0.5,npts),npts,1);
kIm=repmat(linspace(-1,1,npts)',1,npts);

% Итерациялар
for j=1:niter
    % q = z*z + k есептеу
    % q – уақытша айнымалы, комплексті аймақтағы
    qRe=zRe.*zRe-zIm.*zIm+kRe;
     qIm=2.*zRe.*zIm+kIm ;  % нәтижесін сақтайды
     % z мағынасын q мағынасымен алмастырамыз, алшақтықты
     %  болдырмау
    % -5-тен 5-ке дейінгі интервалды сақтаймыз,
     zRe=qRe;
     qgtfive= find(qRe > 5.);
     zRe(qgtfive)=5.;
     qltmfive=find(qRe<-5.);
     zRe(qltmfive)=-5.;
     zIm=qIm;
     hgtfive=find(qIm>5.);
     zIm(hgtfive)=5.;
     hltmfive=find(qIm<-5.);
     zIm(hltmfive)=-5.;
end

% Графикті тұрғызу

% Сурет саламыз
ima=log(sqrt(zRe.*zRe+zIm.*zIm)+1);
% Суретті шығарамыз
imagesc(ima);
```

3.5. суретте бағдарламаны іске асырған кезде шығатын қорытынды график келтірілген.



3.5 Сурет. Мандельброт жиынтығы

Келесі үш жағдай – сызықты еместік, тепе-теңсіз және тұйықталмағандық орындалатын жүйелерде пайда болған объектілер фракталды болады. Табиғи фракталды объектілердің мысалы ретінде өсімдіктердің жапырақтары, найзағай, жағалау сызықтары, терезенің аяз өрнектері, бұлттар (мультифракталдың типтік мысалы) және т.б. бола алады.

Фракталдық өлшемділік есебі

Фракталды объектілердің бір мысалы ретінде өлшемі 100 нм дейін нанокластер-құрылымдары бар жартылай өткізгіш қабықша болады. Осы қабықшаның көлденең қимасын білдіретін қисықтың фракталдық өлшемділігін есептеу әдісін қарастырайық, яғни осы мақсатта өткізгіш электрондар концентрациясының кеңістіктік қадамға тәуелділігін білдіретін бір өлшемді орындауда пайдалану. Бұл үшін наноқұрылымдық жартылай өткізгіште электрондардың, тесіктердің және қоспалардың бөлінуін сипаттайтын кескіндер жүйесі ретінде фракталдық эволюция теңдеуінің шамасын келесі түрде жазамыз:

$$
\begin{cases}
n_{i+1} = \left(\dfrac{1}{C_n} + \mu_i\right)\left|\dfrac{n_i}{n_0}\right|^{-\frac{1}{\gamma_n}};\ p_{i+1} = \left(\dfrac{1}{C_p} + \mu_i\right)\left|\dfrac{p_i}{p_0}\right|^{-\frac{1}{\gamma_p}};\ a_{i+1} = \left(\dfrac{1}{C_a} + \mu_i\right)\left|\dfrac{a_i}{a_0}\right|^{-\frac{1}{\gamma_a}}; \\
\mu_{n_{i+1}} = -\dfrac{1}{\gamma_n}\left(\dfrac{1}{C_n} + \mu_i\right)\left|\dfrac{n_i}{n_0}\right|^{-\frac{1}{\gamma_n}-1};\ \mu_{p_{i+1}} = -\dfrac{1}{\gamma_p}\left(\dfrac{1}{C_p} + \mu_i\right)\left|\dfrac{p_i}{p_0}\right|^{-\frac{1}{\gamma_p}-1}; \\
\mu_{a_{i+1}} = -\dfrac{1}{\gamma_a}\left(\dfrac{1}{C_a} + \mu_i\right)\left|\dfrac{a_i}{a_0}\right|^{-\frac{1}{\gamma_a}-1};\ \mu_i = \mu_{n_i} + \mu_{p_i} + \mu_{a_i}.
\end{cases} \tag{3.4}
$$

Бұл жерде n, p, a – сәйкесінше электрондардың, тесіктердің және қоспалардың тепе-теңсіз (фракталдық) концентрациялары, C_n, C_p, C_a– дәл ажыратылымдылық деңгейі, $\gamma_n, \gamma_p, \gamma_a$– электрондар, тесіктер және қоспалар жиының фракталды D және топологиялық d өлшемділіктерінің арасындағы айырма, n_0, p_0, a_0 – заряд пен қоспалар тасушыларының тепе-тең (фракталды емес) концентрациялары, μ_n, μ_p, μ_{a_i} – белгілік функция.

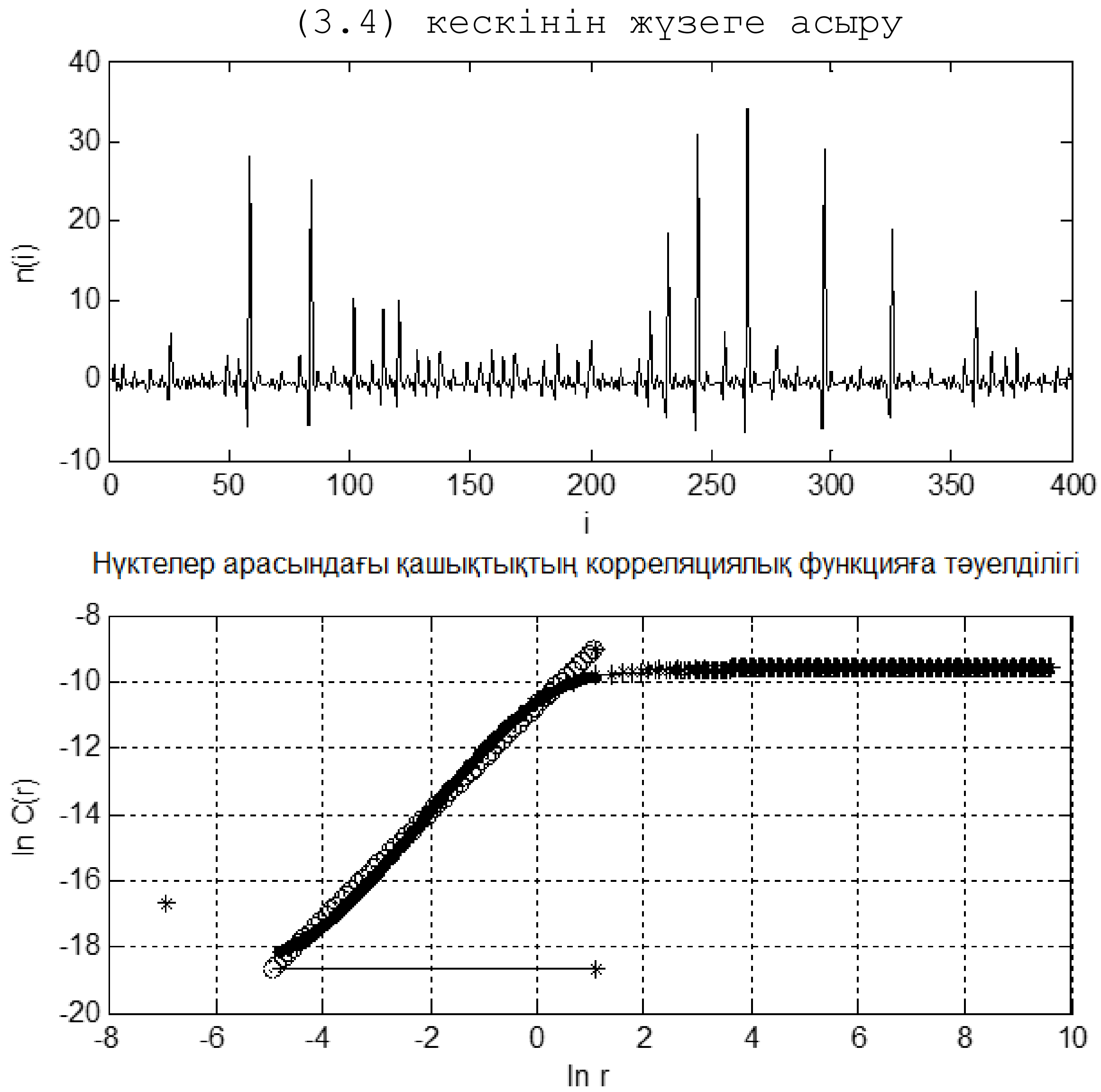
$\gamma = D - d$ параметрі теориялық шамасының n_{i+1}. бір өлшемді орындау бойынша тиісті шамасымен байланысының қандай екенін анықтау қажет. Кескіндер жүйесін (4) шешумен алынған өткізгіш электрондар концентрациясы жиынының фракталды өлшемділігін анықтау үшін келесі алгоритмді пайдаланамыз.

$$
C(r) = \frac{1}{N^2}\sum_{\substack{i,j=1 \\ i\neq j}}^{N} \Theta\left(r - \left|X_i - X_j\right|\right), \tag{3.5}
$$

формула (3.5) бойынша корреляциялық функцияны анықтаймыз, мұнда Θ – Хевисайд функциясы. Қарастырылып отырған нүктелер X_i қашықтықта тұр, r кейбір шамасынан аспайды. r біршама шағын кезінде $C(r)$ функциясы $C(r) = r^D$ ретінде өзгереді, сөйтіп, ізделіп отырған фракталдық өлшемділік ара салмақ бойынша анықталуы мүмкін (3.6).

$$
D = \lim_{r\to 0}\frac{\ln C(r)}{\ln r}. \tag{3.6}
$$

Бос электрондар концентрациясының кеңістіктік қадамға байланысын сипаттайтын орындау мысалы 3.6-суретте көрсетілген. (3.5) және (3.6) формулалар негізінде $\ln C(r)$-дың $\ln r$-ге тәуелділігін құрамыз және ең кіші квадраттар әдісімен қисық сызық көлбеуін анықтаймыз, ол формулаға (13) сәйкес зерттелетін орындаудың ізделіп отырған фракталдық өлшемділігін анықтайды (3.6-сурет).



$$\gamma_n = \gamma_p = \gamma_a = 1.5, C_n = C_p = C_a = 1.001, \; n_0 = p_0 = 0.25, a_0 = 0.53.$$

3.6 Сурет. Кескінді жүзеге асыру (4) және корреляциялық функцияның нүктелер арасындағы қашықтыққа тәуелділігі.

Орындаудың фракталдық өлшемділігінің шамасы Command Window-пен шығарылады. Берілген параметрлерде шама D=1,59.

Тапсырма

1. Кох фракталды қисық сызығының бірден беске дейінгі буынын құрыңыз.
2. Серпинский салфеткасының бірден беске дейінгі буынын құрыңыз.
3. Мандельброт жиынын көрнекілеңіз (визуализация).
4. Құрылған фракталды объектілердің фракталдық өлшемділіктерін есептеңіз.

Бақылау сұрақтары

1. Қандай объектілер фракталды болып есептеледі? Фракталды объектілердің мысалын келтіріңіз.
2. Мультифракталдар дегеніміз не? Мультифракталдарды мысал ретінде келтіріңіз.
3. Объектілердің өзіндік ұқсастығы мен өзін-өзі таныту қасиеттері қандай болады?
4. Фракталдық өлшемділік түсінігін қалай түсінесіз?
5. Фракталдық өлшемділікті есептеуге арналған формуланы жазыңыз.
6. Нысананың фракталды және топологикалық өлшемділіктері өзара қалай байланысады?
7. Фракталдық өлшемділікті Хевисайд функциясы арқылы есептеу алгоритмін сипаттаңыз.

Қолданылған әдібиеттер тізімі

1. Поршнев С.В. Matlab 7. Основы работы и программирования. – М.: ООО «Бином-Пресс», 2011. – 320 с.
2. Федер Е. Фракталы. – М.:Мир, 2002.– 608 с.
3. Nicolis G., Prigogine I. Exploring Complexity. An Introduction. – NewYork: W.H. Freeman and Company, 2007. – p. 342.
4. Zhanabaev Z.Zh, Grevtseva T.Yu, Assilbayeva R.B., Kozhagulov Y.T. Scale-Invariant Regularities of Morphology of Nanostructured Semiconductors // Full Paper Proceeding ITMAR-2016. - Vol. 3. – P. 30-39
5. Мандельброт Б. Фрактальная геометрия природы. – М.: Институт компьютерных исследований, 2002. – 656 с.
6. Жанабаев З.Ж., Иманбаева А.К. , Алмасбеков Н.Е. Компьютерное моделирование в радиофизике и электронике. – Алматы: Қазақ университеті, 2005. – 144 с.
7. Половко А.М., Бутусов П.Н. Matlab для студента. – С.-Петербург: БХВ-Петербург, 2005 – 319 с.

Зертханалық жұмыс №4

ДИНАМИКАЛЫҚ ЖҮЙЕЛЕРДЕГІ ПРОЦЕСТЕРДІҢ ФАЗАЛЫҚ КЕСКІНІН ҚҰРУ

Жұмыстың мақсаты: динамикалық жүйелердегі процестердің фазалық кескінін Matlab ортасында құру әдістерін игеру

Қысқаша теориялық кіріспе

Динамикалық жүйе деп уақыт өтуімен жүйенің бастапқы күйінің өзгерісін (эволюция) сипаттайтын заң, берілген және күй түсінігі кейбір шамалардың уақыттың қазіргі сәтіндегі жиынтығы ретінде анықталған кез-келген объект немесе процесті түсінеді. Бұл заң бастапқы күй бойынша динамикалық жүйенің болашақтағы күйін болжауға мүмкіндік береді, оны *эволюция заңы* деп атайды.

Динамикалық жүйелер – бұл механикалық, физикалық, химиялық және биологиялық объектілер, есептеуіш процестер, нақты алгоритмдерге сәйкес орындалатын ақпараттың түрлену процестері.

Динамикалық жүйенің эволюция заңының тапсырмасы үшін сипаты әртүрлі: дифференциалдық теңдеулердің, дискретті кескін, графтар теориясының, марковтық тізбектер теориясының және т.б. көмегімен.

Сипаттау әдістерінің бірін таңдау сәйкес динамикалық жүйенің математикалық моделінің нақты түрін береді. Динамикалық жүйенің математикалық моделі оның күйін анықтайтын жүйенің параметрлері (координаттары) енгізілген және эволюция заңы көрсетілген болса берілген болып есептеледі. Жақындау деңгейіне тәуелді бір жүйемен сәйкесінше әртүрлі математикалық модельдер салыстырылуы мүмкін. Әдебиетте жиі «динамикалық жүйе» түсінігінен дәл оның математикалық моделі тұспалданады.

Динамикалық жүйені анықтау үшін күйді сипаттауда $t = t_0$ кейбір уақыт мезетінде $x_1, x_2, \ldots, x_N$ шамаларды беруге рұқсат беретін объект көрсету қажет. x_i шамалары еркін мәндер қабылдай алады, әрі екі әртүрлі x_i және $x_i{}'$ шамаларға екі әртүрлі күйлер жауап береді. *Динамикалық жүйенің уақыттағы эволюция заңы* қарапайым дифференциалдық теңдеулер жүйесімен жазылады:

$$\frac{dx_i}{dt} = f_i\left(x_1, x_2, \ldots, x_N\right),\ i = 1, 2, \ldots, N. \tag{4.1}$$

Егер $x_1, x_2, \dots, x_N$ шамаларын N-өлшемді кеңістіктегі нүктенің координаттары ретінде қарастырса, онда *бейнелеу* немесе *фазалық нүктесі* деп аталатын осы нүкте түрінде динамикалық жүйенің көрнекті геометриялық көрінісі алынады, ал күй кеңістігі – динамикалық жүйенің *фазалық кеңістігі* деп аталады. Жүйе күйінің уақыттағы өзгерісіне *фазалық траектория* деп аталатын фазалық нүктенің кейбір сызықты бойлай қозғалуы жауап береді. *Фазалық кескін* – динамикалық жүйенің қозғалысы мен күйін сипаттайтын фазалық траекториялар жиынтығы. Жүйенің фазалық кеңістігінде әрбір x нүктесіне одан шығатын компоненттері (4.1) теңдеуінің оң жақ бөлігімен берілетін F(x) жылдамдық векторын салыстыратын жылдамдықтардың векторлық өрісі анықталады:

$$\left[f_1\left(x_1, x_2, \dots, x_N\right), f_2\left(x_1, x_2, \dots, x_N\right), \dots, f_N\left(x_1, x_2, \dots, x_N\right) \right]. \qquad (4.2)$$

Динамикалық жүйе (4.1) векторлық формада келесідей жазылады:

$$\frac{dx}{dt} = \boldsymbol{F}(x), \qquad (4.3)$$

мұндағы F(x) – N өлшемділігінің вектор-функциясы.

Еркіндік деңгейінің саны және динамикалық жүйенің фазалық кеңістігінің өлшемділігі түсініктерінің өзара байланысын анықтап алу қажет. Еркіндік деңгейінің саны деп жүйе күйін бірмәнді анықтау үшін қажетті тәуелсіз координаттардың ең аз санын түсінеді. Координаталар деп бастапқыда дене мен объектілердің өзара орналасуын сипаттайтын нақ кеңістіктік айнымалыларды түсіеміз. Сол уақытта сәйкес қозғалыс теңдеулерінің бірмәнді шешімі үшін координаттан басқа импульстер немесе жылдамдықтардың сәйкес бастапқы мәндерін беру қажет. Осыған байланысты n еркіндік дәрежелі жүйе өлшемділігі екі есе үлкен ($N = 2n$) фазалық кеңістікпен сипатталады.

Егер динамикалық жүйе (4.3) теңдеуімен берілген болса, онда әрбір $x(t_0)$-ға фазалық кеңістікте сәйкес $x(t)$, $t > t_0$ күйі қойылады, $t - t_0$ уақытта (4.3) теңдеуіне сәйкес қозғалатын фазалық нүкте қайда ауысатыны негіз етіп алынады. Операторлық формада (4.3)-ті келесі түрде жазуға болады:

$$x(t) = \mathbf{T}_i x(t_0), \qquad (4.4)$$

мұндағы $\mathrm{T_i}$ – эволюция заңы (оператор). Егер осы операторды $x(t_0)$ бастапқы күйге қойсақ, онда біз $x(t)$, яғни $t > t_0$ уақыт мезетіндегі күйді аламыз. $x(t_0)$ және $x(t)$ динамиалық жүйенің бір фазалық кеңістікке тиесілі болғандықтан, $\mathrm{T_i}$ операторы жүйенің фазалық кеңістігін өзіне қарай бейнелейді. Сондықтан бұл операторды жиі бейнелеу операторы деп атайды.

Динамикалық жүйелерді бейнелеу операторының түріне және фазалық кеңістік құрылымына байланысты жіктеуге болады. Егер оператор бастапқы күйдің тек қана сызықтық түрленуін қарастыратын болса, ол *сызықтық* деп аталады. Сызықтық оператор суперпозиция қаситіне ие: $T\{x(t) + y(t)\} = Tx(t) + Ty(t)$. Егер оператор бейсызық болса, онда сәйкес динамикалық жүйе *бейсызық* деп аталады.

Динамикалық жүйенің маңызды тобын тербелістер мүмкін жүйелер көрсетеді. Тербелмелі жүйе *сызықты* немесе *бейсызық* деп аталуы оны сипаттайтын дифференциалдық теңдеулер жүйесінің сызықты немесе бейсызық болуына тәуелді.

Қарапайым дифференциалдық теңдеулердің ақырғы санымен үлгіленетін динамикалық жүйелер *жинақталған* немесе *нүктелік* жүйелер деп аталады.

Энергетикалық белгісі бойынша динамикалық жүйелер *консерваторлық* және *консерваторлық емес* деп бөлінеді. Егер ондағы тербелмелі жүйелер динамикалық айнымалылар деп қабылданған шамалармен байланысты толық механикалық энергия (кинетикалық плюс потенциалдық) уақыт өтуімен тұрақты болып қалатындай өтетін болса, жүйе консерваторлық (гамильтонды) болады. Уақытта энергия қоры өзгеретін динамикалық жүйелер консерваторлық емес деп аталады. Шашырау немесе үйкелістің (яғни энергияның жылуға айналуы, немесе диссипация) әсерінен энергия шығыны бар жүйелер диссипативті деп аталады.

Консерваторлық жүйелерге бастапқы күй туралы «жадының» шексіз ұзақ сақталуы тән. Мысалы, осциллятор тербелістері әрқашан жүйенің бастапқы күйін беру кезінде пайда болған амплитуданы сақтайды. Басқа бастапқы күйге басқасы, бірақ өзгеріссіз қалатын амплитуда сәйкес болады.

Диссипативті жүйелерге бастапқы күй туралы «естен шығару» тән. Жүйеде пайда болатын, өзіне-өзі ұзақ уақыт ішінде беретін динамика режимі бастапқы күйден тәуелсіз болады (тым болмағанда кейбір ақырғы шектердегі бастапқы шарттардың вариациясы кезінде).

Динамикалық жүйелерді фазалық траекторияларды зерттеу көмегімен сипаттау

Динамикалық жүйенің фазалық траекториясын зерттеу көмегімен тербелмелі процестердің анализ әдісін тербелістер теориясына Л.И. Мандельштам мен А.А. Андронов енгізді. Тербелістер туралы айта отырып, біз ең болмаса уақытта шамамен қайталану қасиетіне ие процестер, құбылыстар, қозғалыстарды айтып жатырмыз.

Фазалық кеңістік деп жүйенің барлық күйінің жиынтығы жүйенің әрбір мүмкін күйіне фазалық кеңістіктің нүктесі сәйкес келетіндей көрсетілген кеңістікті түсінеді.

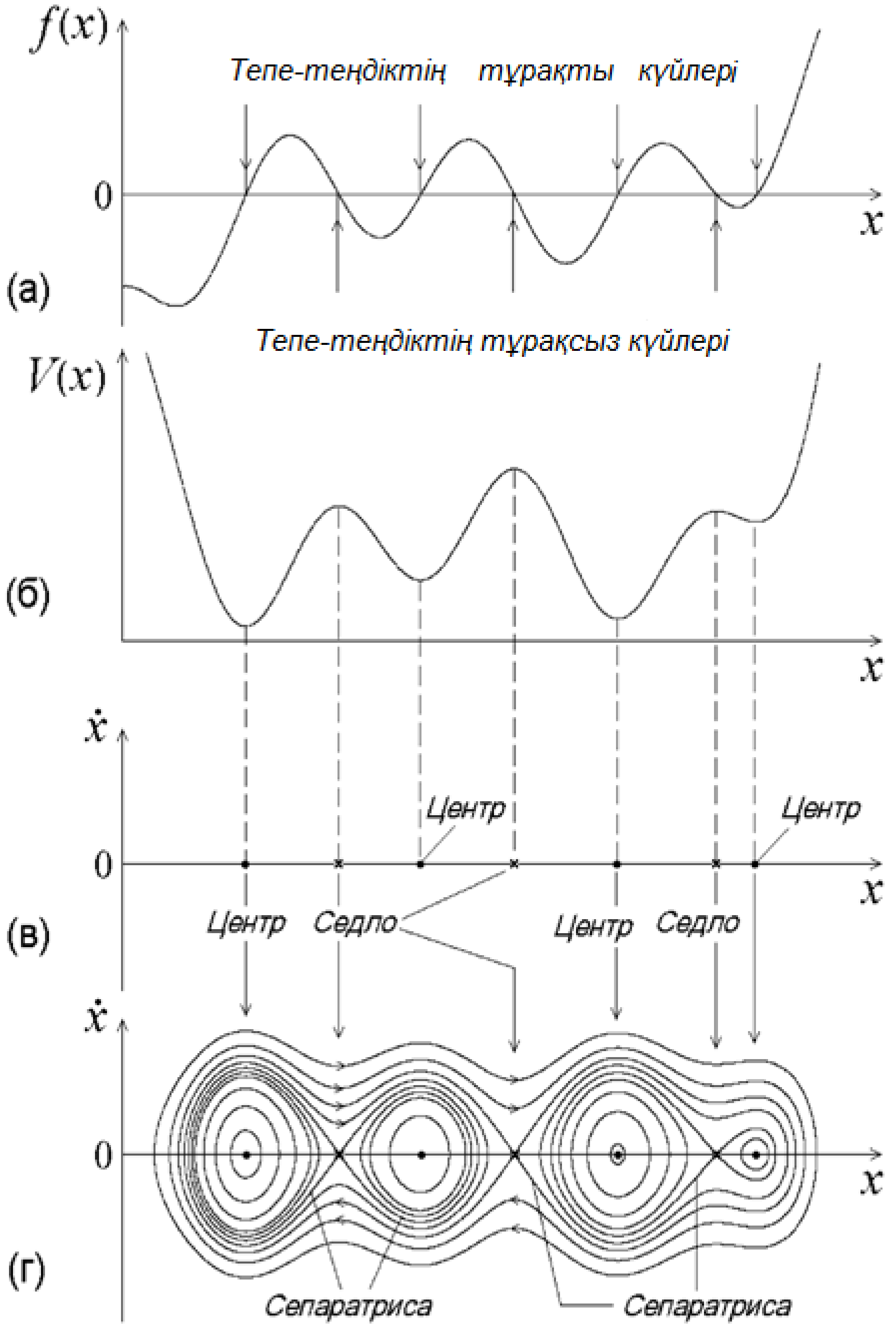
Фазалық кеңістікті көрсету үшін координата ретінде әдетте қарапайым кеңістіктік координаталар (немесе жалпыланған коорндинаталар, яғни саны механикалық жүйенің еркіндік деңгей санына тең, өзара тәуелсіз кез-келген өлшемдегі және жүйенің жағдайын бірмәнді анықтайтын параметрлер) және импульстердің сәйкес мәндері.

Бейнелеу нүктесінің фазалық кеңістіктегі траекториясы түрінде динамикалық процесс көрінісінің мүмкіндігін консерваторлық бейсызық осцилляторды сипаттау мысалында қарастырайық. Мұндай осциллятордың тербеліс теңдеуі $\frac{d^2x}{dt^2} + f(x) = 0$ түрінде болады.

4.1-суретте фазалық жазықтықта ерекше нүктелер көрсетілген. Ерекше нүктеге тән ерекшеліктер:

- Вектор модулі нөлге тең болғандықтан, ерекше нүктеде векторлық өрістің бағыты анықталмаған. (Фазалық жазықтықтың әрбір нүктесіне белгілі жылдамдық векторы жауап береді, яғни динамикалық жүйемен жазықтықтағы белгілі векторлық өріс байланысты деп айтуға болады).
- Ерекше нүктені бөлек фазалық траектория деп мойындау керек: бастапқы шартпен уақыттық эволюция процесінде жүйенің болатын күйлер жиынтығы осы нүктенің біреуінен тұрады.

Физиканың көзқарасынан, ерекше нүктелер динамиканың стационар режимдеріне немесе жүйенің тепе-теңдік күйіне сәйкес келеді.



(а) – f(x) функциясы, (б) – потенциалдық функция, (в) – фазалық жазықтықтағы ерекше нүктелер, (г) – фазалық кескін.

4.1 Сурет. Бейсызық осциллятор мысалы

Сепаратриса $t \to \infty$ (тұрақты сепаратриса) немесе $t \to -\infty$ (тұрақсыз сепаратриса) уақытында тепе-теңдіктің седло күйіне ұмтылатын, екіөлшемді фазалық кеңістігі бар динамикалық жүйенің траекториясы. Егер сепаратриса седлоға $t \to \pm\infty$ кезінде ұмтылатын болса, онда оны (седломен бірге) сепаратриса ілмегі деп атайды.

Диссипативті динамикалық жүйелерде сепаратриса ілмегінен шекті цикл туылуы мүмкін. *Шекті цикл* – периодтық қозғалысты бейнелейтін, динамикалық жүйенің фазалық кеңістігіндегі оқшауланған тұйық траектория. Шекті цикл айналасында фазалық траекториялар одан не алынып тасталады (тұрақсыз шекті цикл), не оған шексіз жақындайды – «оралады» (тұрақты шекті цикл). Консерваторлық динамикалық жүйелерде сепаратриса ілмектері фазалық кеңістікті беталысы әртүрлі траекторияларға бөле алады.

Фазалық кеңістікте тұрақты траекториялардың бар болуы сипатталып жатқан жүйеде динамикалық хаостың бар екенін көрсетеді. Фазалық кескінде тұйық траекториялардың болуы тербелістің периодты сипатын көрсетеді, шексіздікке кететін фазалық траекториялар жүйеде стохастылықтың бар екенін көрсетеді.

Диссипативті жүйеде фазалық көлем элементі уақытта эволюция процесінде сығылады. Динамикалық жүйенің фазалық траекториясының шекті жиынтығы әрқашан нөлдік көлемге ие. Сонымен қатар, шекті жиынтық нүкте, сызық, бет немесе Пуанкаре қимасында канторлық құрылым құратын бет жиынтығы болуы мүмкін. Физикалық көзқарастан маңызды болып еліктіргіш шекті жиынтық – аттракторлар болып табылады. «Оғаш» сөзі аттрактордың екі қасиетін көрсетеді. Бұл, біріншіден, оның геометриялық құрылымының өзгешелігі. Оғаш аттрактордың өлшемділігі бөлшекті (фракталды) болып табылады. Екіншіден, оғаш аттрактор – төңіректегі аймақтардан траектория үшін еліктіргіш аймақ. Сонымен бірге оғаш аттрактор ішінде барлық траекториялар динамикалық тұрақсыз, бұл траекторияның бастапқы мезетіне жақындардың қатты (экспоненциалды) таралғыштығында байқалады.

Динамикалық жүйелердің фазалық кескіндерін құру әдістерін компьютерлік моделдеу көмегімен Matlab ортасында қарастырайық.

Динамикалық жүйені сипаттайтын дифференциалдық теңдеулерді шешу негізінде фазалық кескіндерді құру

Жоғарыда айтылғандай, динамикалық жүйелер дифференциалдық теңдеулер көмегімен сипатталады. Осылайша, фазалық кескіндерді құру әдістемесі дифференциалдық теңдеулерді шешуге негізделеді.

Дифференциалдық теңдеулер – туынды функция мәнін функцияның өзімен, тәуелсіз айнымалы мәндерімен, сандармен (параметрлермен) байланыстыратын теңдеу. Теңдеуге кіретін туындылар реті әртүрлі болуы мүмкін (ресми ол ештеңемен шектелмеген). Туындылар, функциялар, тәуелсіз айнымалылар және параметрлер теңдеуге әртүрлі комбинацияларда кіре алады немесе ең болмағанда бір туындыдан басқа барлығы тіпті болмауы мүмкін. Бір реттіден жоғары дифференциалдық теңдеулер саны бастапқы теңдеу ретіне тең бірінші ретті теңдеулер жүйесіне түрлендіруге болады. *Дифференциалдық теңдеудің реті* немесе деңгейі – оған кіретін туындылардың ең үлкен реті.

Барлық дифференциалдық теңдеулерді тек бір аргументтен болатын функциялардан (және олардың туындыларынан) тұратын қарапайым және кіріс функциялар көптеген айнымалыларға тәуелді болатын дербес туындысы бар теңдеулер деп бөлуге болады. Сондай-ақ кездейсоқ процестерден тұратын стохасты дифференциалдық теңдеулер бар.

Ең қарапайым бірінші ретті дифференциалдық теңдеу – шешімге және зерттеуге ең оңай бейімделген бірінші ретті фдифференциалдық теңдеулер тобы. Оған толық дифференциалдағы, бөлінетін айнымалылары бар, бірінші ретті біртекті теңдеулер және бірінші ретті сызықты теңдеулер жатады. Барлық осы теңдеулерді соңында интегралдауға болады.

$P(t,x)dt + Q(t,x)dx = 0$ симметриялық делінетін формада бірінші ретті дифференциалдық теңдеуді жазайық, мұндағы $P(t,x)$ және $Q(t,x)$ анықталған және кейбір аймақта үздіксіз функциялар.

Matlab-та дифференциалдық теңдеулерді шешу функциялары бар.

Мысалы, $\frac{dy}{dx} = x + y$ дифференциалдық теңдеуін шешу қажет. Шектік шарттар: $x \in (0.2 \div 0.6)$, бастапқы шарт: $y_0 = 2$.

Бұл үшін ‘`function`’ типті m-файлды келесі түрде құрамыз.

```
function  F=xxx22(x,y)
F=x+y
```

Бұл файлды `xxx22. m` деп атаймыз және оған жол орнатамыз, мысалы: `File/SetPath/E:/Matlab/work` және `Save` басамыз. Жол орнатылды. Бұл әрекетті 6.5 және одан төмен Matlab ортасы нұсқасында жұмыс істегенде қолдану қажет. Matlab 7 және соңғы нұсқаларда жұмыс жасағанда m-файлға жол автоматты түрде орнатылады.

Енді бұл m-файлға не командалық терезеден, не басқа m-файлдан жүгіну керек.

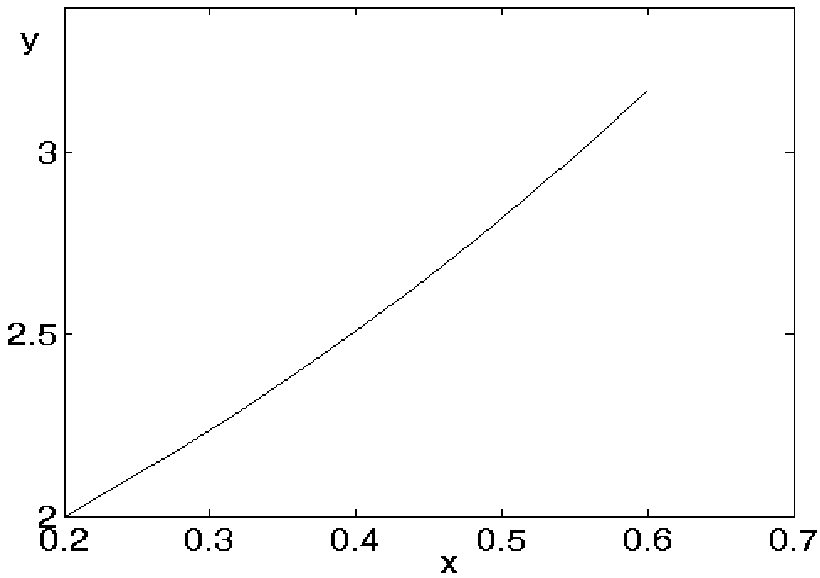
```
>> [X, Y] = ode45 ('xxx22', [0.2  0.6], 2)
```

x бойынша интервал автоматты үрде 40 тең бөлікке бөлінеді және әрбір x үшін өзінің y мәні есептеледі. Дифференциалдық теңдеудің шешімі төртінші ретті Рунге-Кутта әдісі бойынша орындалады. Шешім алгоритмі бірқадамды – әрбір келесі y мәнін есептеу үшін оның алдыңғы мәні қолданылады. Жоғарыда келтірілген жазудың өзі дифференциалды теңдеуді шешу үшін сольвер деп аталады. Рунге-Кутта әдісі

```
>> [X, Y] = ode23 ('xxx', [0.2  0.6], 2)
```

сольвері қолданылғанда да қолданылады. Бірақ бұл жағдайда x бойынша бөлу қадамы көбірек, тура сол интервалда нәтижелер тек 11 болады.

Алынған нәтижені бейнелейік.



4.2 Сурет. $\frac{dy}{dx} = x + y$ дифференциалдық теңдеуін шешу визуализациясы.

Дифференциалдық теңдеулерді шешу үшін басқа да әдістер қолданылуы мүмкін, мысалы, Адамс-Бэшфорт-Милтон әдісі. Сонымен бірге x және y бойынша отыз нәтиже алынады. Matlab-та бұл әдісті қолдану үшін `ode113` функциясы қолданылады:

```
>> [X, Y] = ode113 ('xxx', [0.2  0.6], 2)
```

Басқа да әдістер бар: Розенблох әдісі (x және y бойынша он бір нәтиже қолданылады, әдісті жүзеге асыру үшін `ode23s` функциясы қолданылады), трапеция әдісі (x және y бойынша он екі нәтиже қолданылады, әдісті жүзеге асыру үшін `ode23t` функциясы қолданылады).

Динамикалық жүйені сипаттайтын дискретті кескін негізінде фазалық кескіндерді құру

Динамикалық жүйе түсінігі эволюция операторын кез-келген әдіспен беру мүмкіндігін айтады, дифференциалдық теңдеумен міндетті емес.

Сонымен қатар, соңғы уақытта теориялық зерттеулерде де, қолданбалы сипатты жұмыстарда да рекуррент кескіндермен суреттелетін дискретті уақыты бар жүйелерді өте жиі қарастырады. Бұл жағдайда фазалық траектория деп фазалық кеңістіктегі нүктелердің кейбір дискретті реттілігін түсіну қажет.

Динамикалық жүйені математикалық зерттеу кезінде *кескін* деп $\{x(t_1), x(t_2), \dots, x(t_n), \dots, x(t_N)\}$ деректерді уақыттық сұрыптауды айтады, бұл үшін $x_n = x(t_n)$ белгі енгізеді. Қарапайым анықталған кескінде x_{n+1} шамасын x_n мәні бойынша табуға болады. Мұны жиі келесі түрде жазады:

$$x_{n+1} = f\left(x_n\right). \tag{4.5}$$

Мұндай жазуда айырымдық теңдеуді білуге болады. Кескін түсінігі айнымалылардың көп санына да жалпыланады. Осылай, x_n M компоненті бар вектор бола алады; $x_n = (Y_{1n}, Y_{2n}, \dots Y_{Mn})$, және (4.5) теңдеу M теңдеуден тұратын жүйе болады.

Мысалы, $[x(t), \dot{x}(t)]$ фазалық жазықтықта бейнеленген бөлшектердің қозғалысына талдау жасап жатырмыз деп болжайық. Егер қозғалыс хаосты болса, онда траектория фазалық кеңістіктің кейбір аймағын толтыруға ұмтылады. Егер, алайда, қозғалысты үздіксіз бақылаудың орнына, біз динамикалық сипаттамаларды тек жеке мезеттерде белгілейтін болсақ, онда қозғалыс фазалық жазықтық нүктелерінің реттілігімен көрсетіледі. Егер $x_n = x(t_n)$ және $y_n = \dot{x}(t_n)$, онда бұл фазалық кеңістік нүктелерінің реттілігі екіөлшемді кескінді көрсетеді

$$x_{n+1} = f(x_n, y_n), \;\; y_{n+1} = g(x_n, y_n). \tag{4.6}$$

Егер t_n іріктеме мезеттері белгілі ережелерге бағынатын болса, бұл кескін *Пуанкаре кескіні* деп аталады.

Динамикалық жүйенің фазалық кескінін құру үшін осындай кескін бойынша $x(t_n)$ шамасын абцисса өсі бойынша, ал ордината өсі бойынша - $x(t_{n+1})$ мәнін қою қажет.

Динамикалық хаостың үш өлшемді суретін деректердің бірөлшемді реттілігі бойынша қайта құру әдісі (Такенс әдісі)

Бұл әдіске сәйкес, $x_{i+1} = f(x_i)$ бірөлшемді белгілі реализация бойынша хаосты құбылыстың көпөлшемді суреті қалпына келуі мүмкін. Бұл реттілікті τ кейбір белгіленген кешігуге еселі шамалар ретінде анықталған, ретті өсетін жылжулары бар жиынтықтар қатарына бағыттау керек. Осылайша, біз дискретті айнымалылардың келесі жиынын жазсақ болады:

$$
\begin{gathered}
x_1 : x_1(t_1), \ldots\ldots\ldots\ldots\ldots x_1(t_N) \\
x_2 : x_1(t_1+\tau), \ldots\ldots\ldots\ldots\ldots x_1(t_N+\tau) \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
x_{j-1} : x_1\big(t_1+(j-1)\tau\big), \ldots\ldots\ldots x_1\big(t_N+(j-1)\tau\big)
\end{gathered}
\tag{4.7}
$$

τ тиісті таңдауы кезінде бұл айнымалылар сызықты тәуелсіз болады деп күтуге болады, ал фазалық кеңістікті анықтау үшін бар қажеті осы. Және бұл барлық айнымалыларды $x_{i+1} = f(x_i)$-ға қатысты жалғыз реттіліктен алуға болады. Осылайша, сипатталған алгоритмді қолдану бастапқы реттіліктің бірөлшемді кеңістіктің шегінен шығуға және жүйенің динамикасын көпөлшемді кеңістікте жаюға мүмкіндік береді.

Жоғарыда айтылғанды мысалда қарастырайық. Динамикалық жүйені сипаттайтын теңдеу ретінде өлшемнің фракталды эволюция теңдеуін (аддитивті шаманың) теңдеуін қолданамыз, ол келесі түрде жазылады:

$$
\begin{gathered}
X_{k,i+1} = \left(\frac{1}{C_k} + \sum_{k=1}^{3} \mu_{k,i}\right) \left|\frac{X_{k,i}}{X_{k,0}}\right|^{-\frac{1}{\gamma_k}}, \\
\mu_{k,i+1} = -\frac{1}{\gamma_k}\left(\frac{1}{C_k} + \sum_{k=1}^{3} \mu_{k,i}\right) \left|\frac{X_{k,i}}{X_{k,0}}\right|^{-\frac{1}{\gamma_k}-1}.
\end{gathered}
\tag{4.8}
$$

Сондай-ақ динамикалық хаос заңдарын сипаттайтын, наноқұрылымдалған жартылай өткізгіштегі электрондар, кемтіктер және қоспаларды үлестіру сипаттамасына қарай, бұл теңдеуге кіретін шамалар келесі мағынаға ие. $k = (1,2,3) = (n, p, a)$ параметрі, n, p, a белгілері сәйкесінше электрондарды, кемтіктерді және қоспалардың үлестіруін сипаттайды; $C_к$ – рұқсаттың дәлдік деңгейі; $\gamma_к$- фракталды және топологиялық өлшемділік арасындағы айырмашылық; $X_{к,0}$ – электрондардың, кемтіктердің және қоспалардың тепе-теңдік концентрациясы; μ – белгі функциясы.

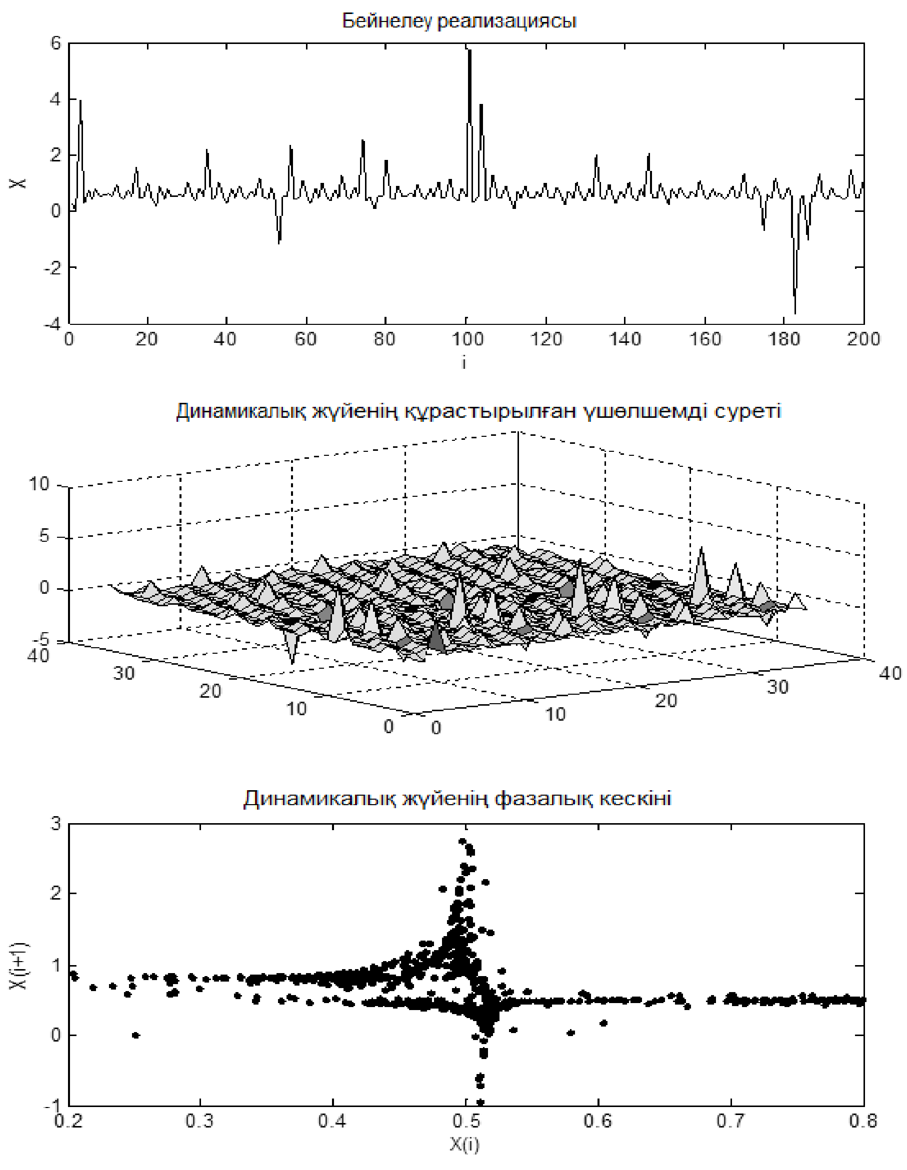
Бұл кескіннің реализациясын бейнелейміз, осы динамикалық жүйені сипаттайтын үшөлшемді суретті қалпына келтіреміз және оның фазалық кескінін құрамыз.

Өлшемнің фракталды эволюциясының кескінін бейнелеу үшін, динамикалық жүйені сипаттайтын үшөлшемді суретін және оның фазалық кескінін құру программасының листингі

```
clear; clc;
gam=3.806;C=1.001;Xo=0.1; % Өлшемнің фракталды эволюция
% теңдеуіне кіретін параметрлер
X(1)=0.25; % X шамасының бастапқы мәні
M=1000; % Есеп жүретін нүктелер саны
mu(1)=-1; % Белгі функциясының бастапқы мәні

% Фракталды өлшем эволюциясы кескінінің реализациясын құру
for i=1:M-1;
% Кескінді құру үшін координатты есептеу
X(i+1)=(1/C+(mu(i))).*(abs((X(i))/Xo))^(-1/gam);
mu(i+1)=(-1/gam)* (1/C+mu(i).*(abs(X(i)/Xo))).^(-1/gam-1);
end
subplot(311);i=1:200;plot(i,X(i)); % алғашқы 200 кескін
% нүктелерін бейнелеу
xlabel('i'); ylabel('X'); % Өстерге жазу
title ('Кескінді жүзеге асыру') % Графикке жазу

% Динамикалық жүйені сипаттайтын, үшөлшемді суретті құру
Z=[]; % Z өсі бойынша координат мәндері үшін матрица жасау
% Такенс әдісін есепке алумен, үшөлшемді суретті
% құру үшін координатты есептеу
M1=35; % Әрбір реализациядағы нүктелер саны,
% олар бойынша график құрамыз
tau=10; % Такенс әдісіндегі жылжыту шамасы

for j=1:1;
     T1=1;
     XX=X(T1:T1+M1);
     Z=[Z; XX];
end
for j=2:M1+1;
     T(j)=(j-1)*tau;
     T1=T(j);
     XX=X(T1:T1+M1);
     Z=[Z; XX];
end

[x,y]=meshgrid(0:M1,0:M1); % Каркасты бет құру
% үшін тор жасау

subplot(312)
surf(x,y,real(Z)); % Үшөлшемді суретті бейнелеу
title ('Динамикалық жүйенің қалпына келтірілген үшөлшемді суреті')
```

```
% Графикке жазу
% Динамикалық жүйенің фазалық кескінін құру
i=1:M-1; % Фазалық кескінді құру үшін
% жүзеге асыру нүктелерінің нөмері
subplot(313);
plot(X(i),X(i+1),'.') % Фазалық кескінді бейнелеу
xlim([0.2 0.8]);
ylim([-1.0 3.0]) % Координаттар өсінің шектері
xlabel('X(i)'); ylabel('X(i+1)');% Өстерге жазу
title (Динамикалық жйенің фазалық кескіні) % Графикке жазу
```

Бұл программаның орындалу нәтижесі 4.3-суретте көрсетілген.



4.3 Сурет. Өлшемділіктің фракталды эволюция бейнесін жүзеге асыру, динамикалық жүйенің қалпына келтірілген үшөлшемді суреті және оның фазалық кескіні

3 суретте көрсетілген жүйенің фазалық кескінінде тұрақты фазалық траекториялардың болуы бұл жүйеде динамикалық хаостың бар екенін көрсетеді.

Тапсырма

1. Келесі дифференциалдық теңдеулер жүйесіне сәйкес, динамикалық жүйе тербелісінің фазалық кескінін құрыңыз (инерциялық бейсызықтығы бар Анищенко-Астахов генераторы):

$$\dot{x} = x(m - z) + y,\ \dot{y} = -x,\ \dot{z} = g(J(x) - z).$$

Келесі параметрлер мәнін қолданыңыз: : $g<1$, $m>1$. $J(x)$ бейсызық функцияны білдіреді (функция түрін өз бетінше беруге болады).

1. Келесі бейне жүйесімен сипатталатын днамикалық жүйенің фазалық кескінін құрыңыз (нанокұрылымдалған жартылай өткізгіш қабықшадағы электрондар, кемтіктер және қоспалардың стохасты үлестіруі):

$$\begin{cases} n_{i+1} = \left\{ n_i + sign\left(\xi_i\right) cos^2\left(i \cdot p_i\right) cos^2\left(i \cdot a_i\right) \right\} \left| \dfrac{n_i}{n_0} \right|^{-\frac{1}{\gamma_n}} \\ p_{i+1} = \left\{ p_i + sign\left(\xi_i\right) cos^2\left(i \cdot n_i\right) cos^2\left(i \cdot a_i\right) \right\} \left| \dfrac{p_i}{p_0} \right|^{-\frac{1}{\gamma_p}} \\ a_{i+1} = \left\{ a_i + sign\left(\xi_i\right) cos^2\left(i \cdot n_i\right) cos^2\left(i \cdot p_i\right) \right\} \left| \dfrac{a_i}{a_0} \right|^{-\frac{1}{\gamma_a}} \end{cases},$$

3. Электрондардың, кемтіктердің және қоспалардың бастапқы мәндерін, сондай-ақ олардың тепе-теңдік концентрацияларының мәнін оңайлық үшін бірге тең деп қабылдауға болады: $n_1 = p_1 = a_1 = n_0 = p_0 = a_0 = 1$, деңгей көрсеткіштері $\gamma_n = \gamma_p = \gamma_a = 1.806.$ тең деп қабылдана алады. ξ_i параметрі кездейсоқ сан (оң немесе теріс болуы тең ықтималды). Алынған нәтижелерге сәйкес зерттеліп жатқан жүйелерде динамикалық хаостың болуы немесе болмауының, сондай-ақ процесс сипаты (стохасты, периодты) туралы қорытынды жасаңыз.

4. 2 тапсырмада келтірілген теңдеулермен сипатталатын динамикалық жүйенің үшөлшемді суретін қалпына келтіріңіз.

Бақылау сұрақтары

1. «Динамикалық жүйе» түсінігінің мәні неде? Динамикалық жүйелерге мысал келтіріңіз. Динамикалық жүйелерді сипаттау кезіндегі эволюция операторы деген не?

2. Сызықты, бейсызық және жинақталған жүйелерге қандай динамикалық жүйелер кіреді?

3. «Консерваторлық жүйе», «консерваторлық емес жүйе», «диссипативті жүйе» терминдерінің мағынасы қандай?

4. Динамикалық жүйенің математикалық моделі нені көрсетеді?

5. Бейнелеу, фазалық нүкте, фазалық траектория, фазалық кеңістік, фазалық кескін, динамикалық жүйенің еркіндік деңгей саны деген не? Еркіндік деңгей саны мен динамикалық жүйенің фазалық кеңістігінің өлшемділігі өзара қалай байланысты?

6. Фазалық жазықтықта қандай нүктелер ерекше болып есептеледі?

7. Фазалық жазықтықта тұйық және шексіздікке кететін траекториялардың болуы нені көрсетеді? Сепаратриса деген не?

8. Шекті цикл деген не?

9. «Аттрактор» және «оғаш аттрактор» терминдері қандай мағынаға ие?

10. Динамикалық жүйенің фазалық кескінін оны сипаттайтын дифференциалдық теңдеулер бойынша және дискретті кескіндердің көмегімен құру әдістерін сипаттаңыз.

Қолданылған әдібиеттер тізімі

1. Анищенко В.С. Сложные колебания в простых системах. – М.: Наука, 2005. – 354 с.
2. Жанабаев З.Ж., Гревцева Т.Ю. Фрактальные структуры и оптические явления в наноструктурированных полупроводниках. – Алматы, Қазақ университеті, 2014. – 162 с.
3. Жанабаев З.Ж., Иманбаева А.К., Алмасбеков Н.Е. Компьютерное моделирование в радиофизике и электронике. – Алматы: Қазақ университеті, 2005. – 144 с.
4. Кузнецов С.П. Динамический хаос. – М.: Физматлит, 2004. – 256 с.
5. Дмитриев А.С., Панас А.И. Динамический хаос. – М.: Изд. физ.-мат. лит., 2002. – 278 с.
6. Дьяконов В.П. Matlab и Simulink для радиоинженеров. – М.: ДМК Пресс, 2011. – 976 с.

Зертханалық жұмыс №5

СИГНАЛДАРДЫ КОРРЕЛЯЦИЯЛЫҚ ТАЛДАУ

Жұмыстың мақсаты: Matlab ортасында сигналдардың автокорреляция және өзара корреляциялық функцияларын есептеу әдісін игеру.

Қысқаша теориялық кіріспе

Сигналдарды корреляциялық талдаудың мәні әр түрлі сигналдардың *ұқсастық дәрежесін* сандық өлшеуден тұрады. Бұл үшін корреляциялық функция қызмет етеді.

Корреляциялық функция детерминирленген сигналдың соңғы энергиясымен интеграл (шексіз шекте) өзімен бірге сигналдың екі көшірме көбейтіндісін көрсетеді, бір-біріне қатысты қозғалады τ уақытта:

$$B(\tau)=\int_{-\infty}^{\infty} s(t)s(t-\tau)dt. \tag{5.1}$$

Корреляциялық функция сигналдың және оның қозғалған көшірмесі арасындағы ұқсастық деңгейін көрсетеді, корреляциялық функцияның мәні көп болған сайын, ұқсастық деңгей күштірек.

Корреляциялық функция *жалпы жағдайда* (қатаң периодты сигнал үшін басқа құрылым) келесі қасиеттерге ие:

Корреляциялық функцияның мәні $\tau=0$ болса сигнал энергиясына тең, яғни квадратының интегралына: $B(0)=\int_{-\infty}^{\infty} s^2(t)dt=E$.

Энергия және сигнал қуаты туралы ұғымды толық қарастырайық.

Мысалы, егер резисторға R кедергімен қоса тұрақты кернеу U берілсе, онда резисторда бөлінген қуат $P=U^2/R$ тең болады. Уақыт өте келе T бұл резисторда жылулық энергия бөлінеді, ол $E=U^2T/R$ тең болады. Енді сол резисторға тұрақсыз кернеу берілсе, ал сигнал $s(t)$. Резистордан тараған қуат уақытқа тәуелді, яғни лездік қуатт туралы тап осы жағдайда айтамыз: $p(t)=s^2(t)/R$. T уақыт ішінде бөлінген энергияны есептеу үшін, лездік қуатты келесідей интегралдау керек: $E=\int_0^T p(t)dt=\frac{1}{R}\int_0^T s^2(t)dt$. Берілген уақыт аралығында *орташа қуат* туралы түсінік келтіруге болады, энергияны уақытша ұзақтық интервалына бөлгенде: $\langle P\rangle=\dfrac{E}{T}=\dfrac{1}{RT}\int\limits_0^T s^2\left(t\right)dt$.

Барлық осы формулалар жүктеменің кедегісін R қамтиды. Бірақ энергия және қуат бізді физикалық өлшем ретінде емес, әртүрлі сигналдарды салыстыру амалымен қызықтырса, бұл параметрді формуладан жойып жіберсе болады (R=1 қабылдау)

$E=\int\limits_0^T s^2\left(t\right)dt$ – сигнал энергиясы.

$p\left(t\right)=s^2\left(t\right)$ – сигналдың лездік қуаты.

$\langle P\rangle=\dfrac{1}{T}\int\limits_0^T s^2\left(t\right)dt$ – сигналдың орташа қуаты.

Корреляциялық функция қандай қасиеттерге ие екенін анықтайық.

1. Корреляциялық функция өз аргументінің жұп функциясы болады $B\left(\tau\right)=B\left(-\tau\right)$.
2. Корреляциялық функцияның мәні $\tau=0$ кезінде, оның максималды мүмкін мәні болады: $B\left(\tau\right)\le B\left(0\right)$.
3. Абсолюттік мәнінің $\tau=0$ өсуімен соңғы энергиясы бар сигналдың корреляциялық функциясы өшеді: $\lim\limits_{|t|\to\infty} B\left(\tau\right)=0$.
4. Егер сигналда дельта-функция болмаса, оның корреляциялық функциясы үзіліссіз болады.
5. Периодты сигналдың корреляциялық функциясының өлшемі – сигнал өлшемінің квадратына тең.

Периодты сигналдың корреляциялық функциясы келесідей сипатталады. Периодты сигналдың корреляциялық функциясы T периодпен және қозғалған көшірме шегінің көбейтіндісін бір периодта орташалағанда есептеледі:

$$B\left(\tau\right)=\frac{1}{T}\int\limits_{-T/2}^{T/2} s\left(t\right)s\left(t-\tau\right)dt. \qquad (5.2)$$

Мұндай корреляциялық функцияның қасиеттер жиынтығы бірнеше рет өзгереді.

1. Периодты сигналдың корреляциялық функциясының мәні $\tau = 0$ болғанда энергияға тең болмайды, анализдік сигналдың орташа қуатына тең:

$$B(0) = \frac{1}{T} \int_{-T/2}^{T/2} s^2(t) dt = \langle P \rangle .$$

2. Корреляциялық функцияның жұптық қасиеті сақталады, егер қарастырылған сигнал периодты болса: $B(\tau) = B(-\tau)$.

3. Периодты сигналдың корреляциялық функциясының мәні $\tau = 0$ болса, максималды мүмкін болады: $B(\tau) \le B(0)$.

4. Периодты сигналдың корреляциялық функциясы дәл сондай периодпен периодты функция болады, не сигнал өзі.

5. Егер сигналда дельта-функция болмаса, оның корреляциялық функциясы үзіліссіз болады.

6. Периодты сигналдың корреляциялық өлшемі – сигнал өлшемінің квадратына тең.

Сол сигналдың әртүрлі бөліктері үшін есептелген корреляция функциясы *автокорреляция* функциясы деп аталады (формула (1)). Екі әр түрлі сигнал бөліктері үшін есептелеген корреляциялық функция $s_1(t)$ және $s_2(t)$, бұл *өзара корреляциялық* функция деп аталады, келесі қатынас көмегімен есептелуі мүмкін:

$$B(\tau) = \int_{-\infty}^{\infty} s_1(t) s_2(t-\tau) dt . \qquad (5.3)$$

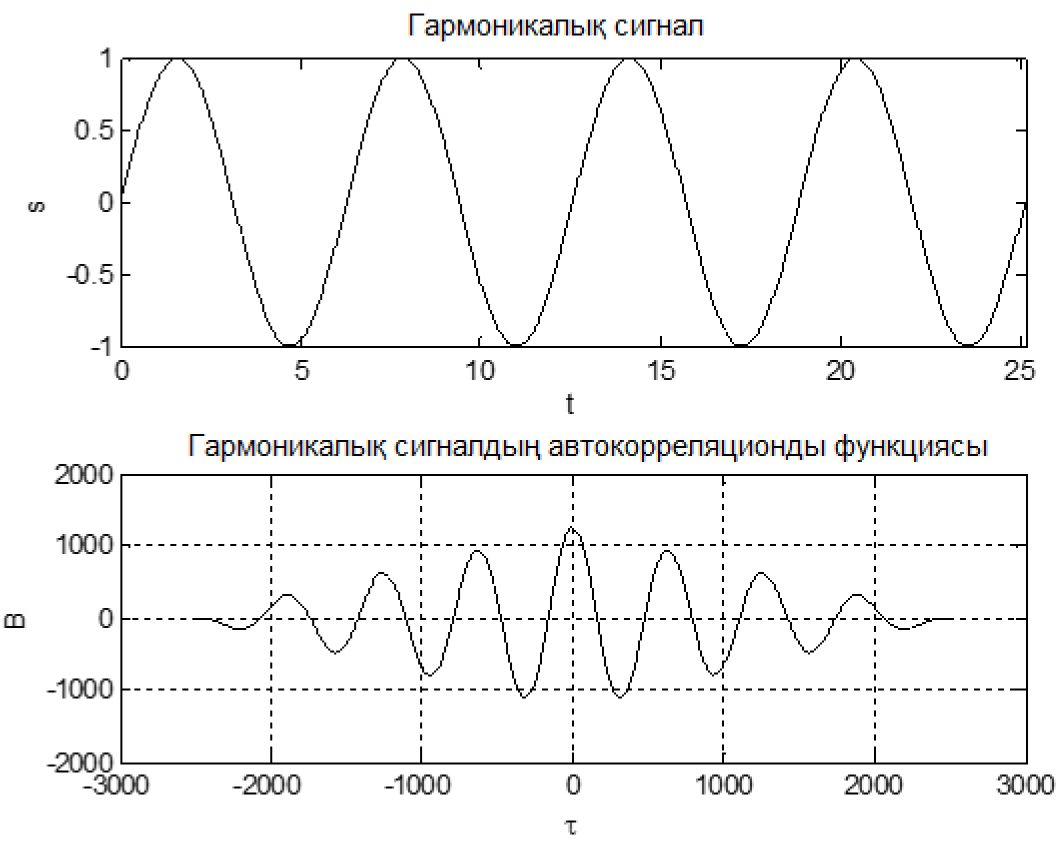
Matlab-та корреляциялық функцияны есептеу үшін `xcorr` командасы қолданылады. Сигналдың $s(t)$ автокорреляциялық функциясын есептеу үшін `xcorr(s)` командасы қолданылады, ал сигналдың автокорреляциялық функциясын табу үшін $s_1(t)$ және $s_2(t)$ `xcorr(s1,s2)` командасы орындалады.

Әр түрлі сигналдардың корреляциялық функциясының есептелуімен құрылуына бірнеше мысалдар қарастырайық.

Гармоникалық сигналдың автокорреляциялық және корреляциялық функциясын құрайық.

Гармоникалық сигналды нөлдік бастапқы кезеңмен бейнелеу және оның автокорреляция функциясын құру бағдарламасы.

```
clear;clc;
% Гармоникалық сигналдың визуализациясы
t=0:0.01:8*pi; % Гармоникалық сигналдың анықталу аймағы
w=1;A=1; % параметрлердің сандық мәндері
s=A*sin(w*t); % Гармоникалық сигналдың сипаттамасы
subplot(211);plot(t,s) % Сигналдың визуализациясы
xlim([0 8*pi]); % абсциссалар осьтері бойынша шектер
xlabel('t'); ylabel('s') % осьтер атауы
title ('Гармоникалық сигнал');

% гармоникалық сигналдың автокоррекциялы функциясының есебі

N=length(t); % сигналдың санақ санын табу
B=xcorr(s); % s сигналының автокорреляуиялық функциясын анықтау
N_B=length(B); % автокорреляциялы функцияның мәндерін анықтау
t1=[1:N_B]; % қолданатын нүктелер санын анықтау
tau=t1-N; % tau массивының есептеуі (смещение)
subplot(212); plot(tau,B) % автокорреляциалы функцияның
% визуализациясы
grid on % графикте тордың орналасуы
xlabel('\tau'); ylabel('B') % осьтер атаулары
title ('Гармоникалық сигналдың автокрреляциялық функциясы');
```



5.1 Сурет. Гармоникалық сигнал және оның автокорреляциялық функциясы

5.1 суретте көрсетілгендей, синусоидалық қисықтың төрт максимумы автокорреляция функциясының графигіндегі төрт максимумға сәйкес келеді, олардың максималды мәні абсцисс осінің нөліне сәйкес келеді. Сондай-ақ, байқағанымыздай гармоникалық сигналдың корреляциялық функциясы τ мәні өсуімен төмендейді.

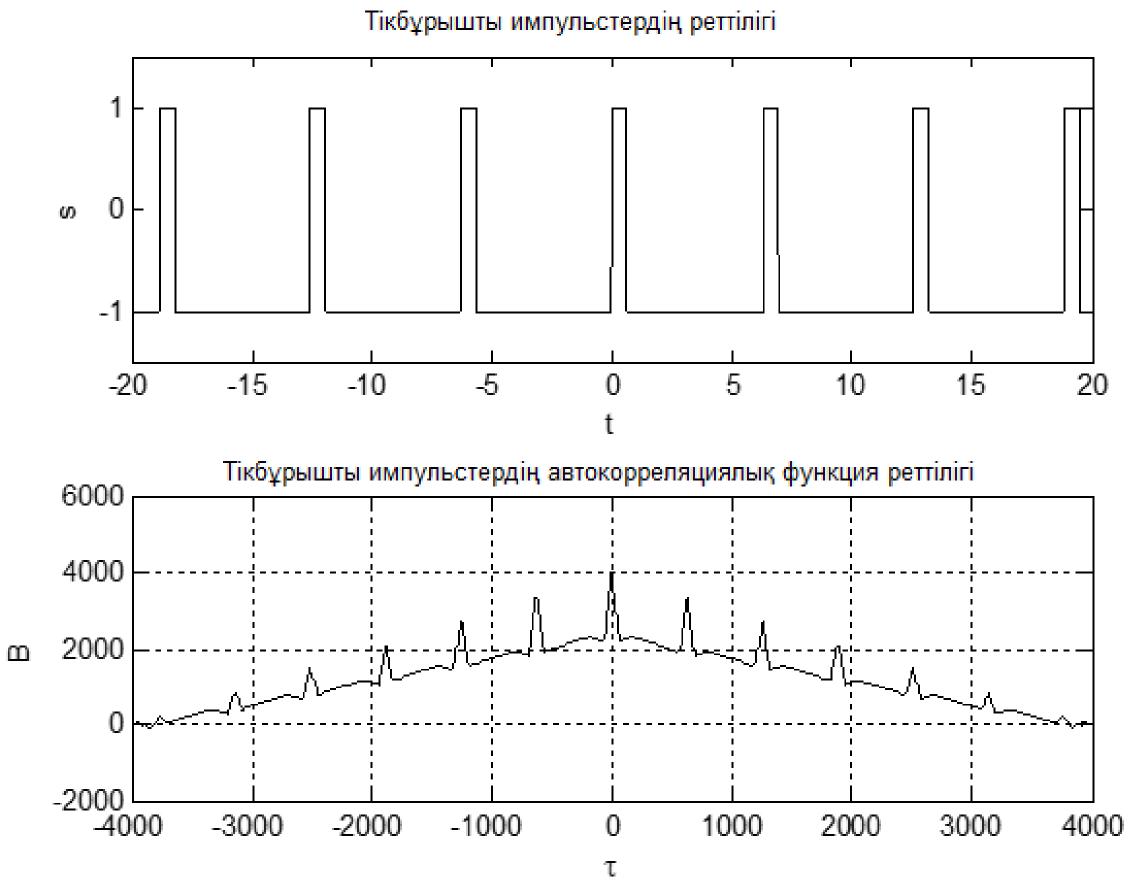
Сол сияқты, біз тікбұрышты импульстардың тізбегімен автокорреляция функциясын құруға болады. Жоғарыда сипатталғандай, тікбұрышты импульстардың тізбегі `square` функциясының көмегімен қалыптасады. Белгілі болғандай, `square` функциясы жалпы жағдайда екі енгізу параметрін қабылдайды, уақыт векторы `t` және параметр `duty`, сол арқылы алынған тізбекті басқаруға болады.

10% интервалда аргумент осьтері -20-дан 20-ға дейін аралықта тікбұрышты сигналдың жетуі және автокорреляция функциясын құрамыз. Осы сигналдың және автокорреляциялық функциясын модельдеу программасы төменде келтірілген.

Тіктөртбұрышты импульстерді және оның автокорреляция функциясын реттейтін бағдарламаның листингі.

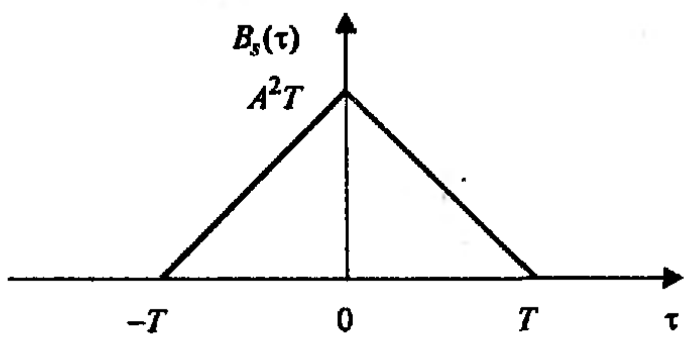
```
clear;clc;
t=-20:0.01:20;
s=square(t,10);
subplot(211);
plot(t,s)
ylim([-1.5 1.5]);
xlabel('t'); ylabel('s')
title ('Тікбұрышты импульстердің реттілігі');
N=length(t);
B=xcorr(s); N_B=length(B);
t1=[1:N_B]; tau=t1-N;
subplot(212); plot(tau,B)
grid on;
xlabel('\tau'); ylabel('B')
title ('Тікбұрышты импульстердің автокорреляциялық функция реттілігі');
```

Тікбұрышты сигналдың алдын-ала белгіленген тізбегі және оның автокорреляциялық функциясының модельдеу нәтижесі 5.2 суретте көрсетілген. Сигналды іске асыру және автокорреляция функциясының графигін салыстырудан бастап, автокорреляция функциясының шашыраулар саны белгілі аралықта байқалатын тікбұрышты сигнал импульстарының санына сәйкес келетінін көруге болады.



5.2 Сурет. Тік бұрышты импульстардың тізбегі
және оның автокорреляциялық функциясы

Жалпы, бір тікбұрышты импульстің корреляциялық функциясының кестесі үшбұрышты пішінге ие (5.3-сурет).



5.3 Сурет. Бір тікбұрышты импульстің автокорреляциялық функциясы

Біздің жағдайымызда, белгіленген интервалда импульстер тізбегінде 7 шың байқалады. Демек, корреляция функциясының әрбір жағында (сол және оң) 7 шыңнан (7 үшбұрыш құрылым) байқалады.

Тапсырма

1. Бір импульсті трапеция тәрізді автокорреляция функциясының графигін тұрғызыңыз (трапеция тәрізді сигналдың модельдеу әдісі № 1 зертханалық жұмыста жазылған). Графикті тұрғызу барысында трапеция тәрізді импульс симметриялы екенін ескеру керек, амплитудасы 10В, сәйкесінше жоғарғы және төменгі табандарының өлшемдері 20 және 60мс. Дискретизация жиілігін 1кГц-ге тең деп аламыз.
2. Шуылды сигналдың автокорреляция функциясының графигін тұрғызыңыз, екі синусоидалы компонентті құрайтын бірінші және үшінші гармоника теңдеуі $y = \sin\left(2\pi \cdot t\right) + 0.25\sin\left(2\pi \cdot 3t\right)$.бойынша. Сигналдың шу деңгейі сигналдың корреляциялық функциясына қалай әсер ететінін қарастырыңыз.
3. Шуылсыз және ақ гаус шуылын қосу арқылы ара тәрізді сигналдың автокореляциялық функциясының графигін тұрғызыңыз. Автокорреляциялық функцияға шу амплитудасының әсерін зерттеңіз.
4. Сигналдардың өзара автокорреляциялық функциясының $s_1\left(t\right) = \sin t$ және $s_2\left(t\right) = 2\sin(2t + \varphi)$, мұнда $\varphi = 0.2$ графигін тұрғызыңыз. Анықтау облысын қарастырыңыз $t \subset \left\{-4\pi \div 4\pi\right\}$.

Бақылау сұрақтары

1. Сигналдарды корреляциялық талдаудың мәні қандай?
2. Автокорреляция мен өзара корреляциялық функциялар арасындағы айырмашылық қандай?
3. Автокорреляция және өзара корреляциялық функциялар қандай мақсаттар үшін қолданылады?
4. Соңғы энергиясы бар детерминделген сигналдың автокорреляциялық функциясы қандай формула бойынша есептеледі?
5. Соңғы энергиялы сигналдарға қандай сигналдар жатады?
6. Өзара корреляциялық функцияны есептейтін формуланы жазыңыз?
7. Периодтық сигналдардың корреляциялық функциясын қалай анықтауға болады?
8. Соңғы энергиясы бар детерминделген сигналдың корреляциялық функциясының қасиеттері қандай?
9. Периодты сигналдың корреляциялық функциясы қандай қасиеттерге ие?
10. Сигналдардың автокорреляция және өзара корреляциялық функцияларын есептеу және бейнелеу үшін Matlab-тың қандай функциялары қолданылады?
11. Автокорреляциялық функцияның Matlab ортасында есептеу алгоритімін сипаттаңыз?
12. Өзара корреляциялық функцияның Matlab ортасында есептеу алгоритімін сипаттаңыз?
13. Шуылсыз және ақ гаусс шуылының қосылуы арқылы сигналдардың корреляциялық функциясының графиктері арасындағы айырмашылық неде? Бұл айырмашылықты қалай түсіндіруге болады?

Қолданылған әдібиеттер тізімі

1. Дьяконов В.П. Matlab 6.5 SP1/7+Simulink 5/6. Основы применения. – М.: СОЛОН-Пресс, 2005. – 800 с.
2. Дьяконов В.П. Matlab и Simulink для радиоинженеров. – М.: ДМК Пресс, 2011. – 976 с.
3. Сергиенко А.Б. Цифровая обработка сигналов. СПб.: БХВ_Петербург, 2013. – 768 с.
4. Сато Ю. Без паники! Цифровая обработка сигналов. – М. Додека-XXI, 2010. – 178 с.
5. Жанабаев З.Ж., Иманбаева А.К., Алмасбеков Н.Е. Компьютерное моделирование в радиофизике и электронике. – Алматы, Қазақ университеті, 2005. – 144 с.
6. Поршнев С.В. Компьютерное моделирование физических процессов в пакете Matlab. – М.: Горячая линия – Телеком, 2008. – 593 с.

Зертханалық жұмыс №6

СИГНАЛДЫҢ ФУРЬЕ-ТАЛДАУЫ

Жұмыстың мақсаты: сигналдардың Фурье-талдауы негіздерін зерттеу, сигналдың Фурье қатарына жіктелу әдісін ұғыну және Matlab ортасында компьютерлік модельдеу көмегімен олардың спектрлік талдауын жасау.

Қысқаша теориялық кіріспе

Сигналдың Фурье қатарына жіктеу

Фурье түрлендіруі сигналдың спектрлік талдаудың құралы болып табылады.

Периодты сигнал $s(t)=s(t+nT)$ арқылы анықталады. Кез келген периодты синусоидалды емес сигналды синусоидалды компоненттердің бірнеше санына жіктеуге болады, оның ішінде негізгі құраушының жиілігі f шығыс сигналы секілді болады, сонымен қатар жиіліктері 2w, 3w, 4w және т.б. болатын гармониктердің бірнешеуі. Тікбұрышты сигнал өте жоғары реттіліктегі гармониктерден тұрады, тік аймақтар мен сынулар шексіз қатардағы гармониктердің барын көрсетеді. Шындығында, тікбұрышты сигналдың жүзеге асуы жақсы болып шығады, әрине, егер ол оныншы гармониктегі жиілікке дейін көрсетілген болса.

Сигналдардың *Фурье қатарына жіктелуі* кезінде олар гармоникалық функциялардың қосындысы немесе арифметикалық прогрессияны құратын жиіліктері бар комплексті экспонентті көрсетеді. Осыны жүзеге асыру үшін, ұзақтығы бір период болатын сигнал бөлігін *Дирехле шарттарын* қанағаттандыруы қажет:

1. Екінші ретті үзілулер болмау қажет (функция бөлігінің шексіздікке кетуі);
2. Бірінші ретті үзілістер (секіру) саны шексіз болуы керек;
3. Экстремумдар саны шексіз болуы қажет (мысал ретінде, соңғы интервалда шексіз экстремумдар санына ие болатын функцияның нөлдік аймақта sin(1/x) әкеп соқтыруы мүмкін).

Базисті функциялардың нақты формаларына қарағанда Фурье қатарының жазылу формалары бірнеше түрге жіктеледі.

S(t) сигналы уақытқа тәуелсіз құраушыдан және гармоникалық тербелістердің шексіз құрамынан тұрады, олардың реттілігі негізгі жиілікке еселі (өзінің жиілігінен төменірек болатын бірінші гармоникке) болатын $w_n = nw\,(n=1,2,...)$ жиіліктері бар гармоник деп атайды.

Қатар коэффициенттері мына қатынас арқылы анықталады:

$$a_0 = \frac{2}{T}\int_0^T s(t)dt,$$
$$a_n = \frac{2}{T}\int_0^T s(t)\cos\left(2\pi\frac{n}{T}\right)dt, \qquad (6.1)$$
$$b_n = \frac{2}{T}\int_0^T s(t)\sin\left(2\pi\frac{n}{T}\right)dt.$$

Реттіліктің негізгі жиілігі: $w = 2\pi/T$. Онда периодты сигнал үшін Фурье қатары:

$$s(t) = \frac{a_0}{2} + \sum\left(a_n\cos(n\omega t) + b_n\sin(n\omega t)\right). \qquad (6.2)$$

Осыдан көретініміз, $s(t)$ функциясы уақытқа тәуелсіз тұрақты құраушы мен гармоникалық тербелістердің шексіз құрамнан тұрады, реттілігі негізгі жиілікке еселі $\omega_n = n\omega\ (n = 1, 2, ...)$ жиілігі бар гармоник.

$\frac{1}{T}\int_0^T s(t)^2 dt = \frac{a_0^2}{2} + \sum_{n=1}^{\infty}\left[a_n^2 + b_n^2\right].$ теңдігі орын алады. Фурье қатарының коэффициенттері $a_n = A_n\cos\varphi_n$, $b_n = A_n\sin\varphi_n$ түрінде жазылуы мүмкін, мұндағы $A_n = \sqrt{a_n^2 + b_n^2}$, $tg\varphi_n = \frac{b_n}{a_n}$. Ондай болса, Фурье қатары $s(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} A_n\cos(nwt - \varphi_n)$ түрінде жазылуы мүмкін.

Периодты функцияның спектралды ыдырауын ортогональды болып табылатын саналы көрсеткіштері бар экспонент түріндегі базисті функциялар жүйесін қолдану арқылы жасауға болады. Ондай жағдайда Фурье қатары $c_n = \frac{1}{\sqrt{T}}\int_0^T s(t)e^{-inwt}dt$ коэффициенттері бар $s(t) = \frac{1}{\sqrt{T}}\sum_{-\infty}^{\infty} c_n e^{inwt}$ түрінде болады. Практикада Фурье қатарының басқа түрде жазылуын да қолдансақ болады: $s(t) = \sum_{-\infty}^{\infty} C_n e^{inwt}$, мұндағы $C_n = \frac{1}{T}\int_0^T s(t)e^{-inwt}dt$.

Фурье қатары периодты сигналдарды көрсетумен қатар, соңғы ұзақтықтағы сигналдардың көрінісі үшін де қолдануға болады. Осы жерде Фурье қатарын құру үшін қажет уақыт интервалы көрсетіледі, ал басқа уақыт кезінде сигнал нөлге тең деп саналады. Қатар коэффициенттерін есептеудің мұндай тәсілі қарастырылатын интервал шекарасынан тыс сигналдың *периоды жалғасы* болып саналады.

Периодты емес сигналдарды Фурье-талдау

Спектральді сипаттамасын алуға болатын периодты емес сигналдардың Фурье қатарының жалпы түрін қарастырайық. Мұндай мәселе *талдау мәселесі* деп аталады.

$s(t)$ функциясы соңғы ұзақтығы $[0,T]$ болатын уақыт интервалында делік. Берілген функцияны T уақыт интервалы периодты қайталанатын осындай сигналдармен толықтырып, комплексті Фурье қатары түріндегі периодты реттілікті аламыз:

$$s(t)=\sum_{n=-\infty}^{\infty} C_n e^{in\omega t} \tag{6.3}$$

Бірлік интервалға қайта оралу үшін T қайталаушы периодын шексіздікке ұмтылдырайық, сондай-ақ көрші гармониктердің жиіліктері $n\omega$ және $(n+1)\omega$ бір-біріне жақын болады, сондықтан $s(t)$ формуласы үшін $n\omega$ дискретті айнымалыны ω үздіксіз айнымалыға ауыстыруға болады, ал келтірілген формулада T мәні үлкен болғандықтан C_n коэффициентері өте аз мәнге ие болады

$$C_n=\frac{1}{T}\int_0^T s(t)e^{-inwt}dt\,. \tag{6.4}$$

Шекті түрге (6.3) ие болу үшін қуаттың спектрлік тығыздығы деген түсінік енгізейік, мұнда Фурье қатарының коэффициенттері комплексті біріккен жұп құрады: $C_n=A_n e^{i\varphi_n}$, $C_{-n}=A_n e^{-i\varphi_n}$, ал әр жұпқа $2A_n e^{i\varphi_n}=2C_n$ комплексті амплитудасы бар қарапайым гармоникалық тербеліс $A_n e^{i(n\omega t+\varphi_n)}+A_n e^{-i(n\omega t+\varphi_n)}=2A_n\cos(n\omega t+\varphi_n)$ жауап береді.

$\Delta\omega$ жиілігінің аз интервалында ω_0 жиілігінің бірнеше мәнінің аймағында спектральді құраушылардың бөлек жұптары $N=\Delta\omega/\omega=\Delta\omega T/2\pi$ құрамында болады, мұндағы жиіліктердің шамалары өте аз, сондықтан құраушыларды бірдей жиілік шамасында және бір комплексті амплитудамен сипатталады деп алсақ та болады. Осылайша,

$\Delta\omega$ интервалына кіретін барлық спектралді құраушыларды сипаттайтын эквивалентті гармоникалық сигналдың комплексті амплитудасы мынаған тең:

$$\Delta A_{\omega_0} = \frac{2N}{T}\int_{-\infty}^{\infty} s(t)e^{-i\omega_0 t}dt = \frac{\Delta\omega}{\pi}\int_{-\infty}^{\infty} s(t)e^{-i\omega_0 t}dt.$$

Сигнал функциясының спектралді $s(t)$ тығыздығы келесі қатынас көмегімен анықталады: $S(\omega) = \int_{-\infty}^{\infty} s(t)e^{-i\omega t}dt.$ Бұл формула Фурьенің тура түрлендірілуінің математикалық теңдеуі болып табылады. $S(\omega)$ спектрлік тығыздығы жиілік осі бойымен энергия таралуының реализациясын сипаттайды.

Талдау мәселесіне кері болатын, сигналды өңдеу теориясындағы *синтез мәселесі* деп аталатын спектральды тығыздық $s(t)$ функциясының түрін анықтау мәселесі болып табылады. Оның шешімі үшін, периодты емес сигнал периодты реттіліктен алынады деп болжайық, мұндағы оның периоды $T\to\infty$. (6.3) және (6.4) формулалар көмегімен алатынымыз

$$s(t) = \lim_{T\to\infty}\sum_{n=-\infty}^{\infty}\frac{1}{T}S(n\omega)e^{in\omega t}. \qquad (6.5)$$

Осы формулаға енетін $1/T$ коэффициенті кез келген n-нің мәнінде көрші гармониктер жиіліктерінің айырымына тура пропорционал:

$$\frac{1}{T} = \frac{\omega}{2\pi} = \frac{1}{2\pi}\left[n(\omega+1) - n\omega\right]. \qquad (6.6)$$

(6.5)-ті (6.6)-ға қойып, алатынымыз:

$$s(t) = \lim_{T\to\infty}\frac{1}{2\pi}\sum_{n=-\infty}^{\infty}S(n\omega)e^{in\omega t}\left[n(\omega+1) - n\omega\right]. \qquad (6.7)$$

Көрші гармониктер арасындағы жиіліктік интервалдар шексіз қысқаратындықтан, соңғы формуладағы сумманы интегралмен ауыстырсақ болады

$$s(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty}S(\omega)e^{i\omega t}d\omega. \qquad (6.8)$$

(6.8) формула Фурьенің кері түрлендіруін көрсетеді.

S(t) функциясы мен $S(w)$ спектральді тығыздығы Фурьенің тура және кері түрлендіруімен байланысты. Бұл қатынас тек интегралды функцияға, яғни $\int_{-\infty}^{\infty}\left|s(t)dt\right|<\infty$ шартын қанағаттандыратын функцияға тең. Бұл шарт мүмкін болатын функциялар классын ерекше шектейді, бірақ физикалық экперименттерде тіркелетін барлық реалды сигналдар шекті ұзақтыққа ие болады, сондықтан осы шартты автоматты түрде қанағаттандырады.

Тез Фурье түрлендіруі

Дискретті Фурье түрлендіруінің бір коэффициентін анықтау үшін комплексті көбейту мен бөлудің *N* операциясын орындау қажет. Осылайша, *N* коэффиценттен тұратын барлық дискретті Фурье түрлендіруін есептеу үшін, N^2 жұп «көбейту-бөлу» операциясын орындау қажет.

Операция саны дискретті Фурье түрлендіруінің квадратты өлшеміне пропорционал өседі. Бірақ, егер *N* параметр жай сан емес және көбейткіштерге жіктелетін болса, зерттелетін есептеу жинағын бөлімдерге бөліп, олардың дискретті Фурье түрлендіруін есептеп, нәтижелерді біріктіру арқылы есептеу процессін тездетуге болады.

Бұндай есептеу әдісі *тез Фурье түрлендіруі* деп аталады және практикада жиі қолданылады.

Тез Фурье түрлендіруінің Matlab функциясы

Matlab пакетінде тез Фурье түрлендіруі тура және кері тез Фурье түрлендіруін орындайтын функция жұбы ретінде іске асырылады: `fft` және `ifft`. Берілген функциялар нақты әрі комплексті реттілік үшін қолданылады, сондай-ақ, реттіліктің ұзындығы әр түрлі болуы мүмкін.

`fft(v)` функциясы 2^m-ретті вектордың дискретті Фурье түрлендіруін іске асырады, мұндағы аргумент кейбір функциялардың тең уақыт аралығы көмегімен алынатын дискретизация нәтижесі болып табылады. Бағдарлама жұмысының нәтижесі 2^{m+1} тең өлшемділікке ие комплексті вектор. fft функциясы қайтаратын вектор элементі $c_n=\frac{1}{N}\sum_{k=0}^{N-1}v_k e^{i2\pi nk/N}$, формула арқылы анықталады, мұндағы N – v векторының элементтер саны.

`ifft(v)` функциясы Фурьенің кері дискретті түрлендіруін орындау үшін қызмет етеді. v векторы 2^{m+1} элементтен тұру қажет. Бағдарлама жұмысының нәтижесі - 2^{m+1} өлшемділікке ие нақты вектор. Вектор элементтері $c_n = \dfrac{1}{N}\sum_{k=0}^{N-1} v_k e^{-i2\pi nk/N}$, формуласы арқылы есептеледі, мұндағы N – ν векторының элементтер саны. Барлық векторлар үшін `ifft(fft(ν))=ν` теңдігі дұрыс.

Matlab ортасында Фурье-талдауын жүргізу мысалдары

Тікбұрышты импульсті 25 гармоник бойынша Фурье қатарына жіктеуді қарастырайық. Импульс ұзақтығы 1 с. Төменде сәйкес келетін бағдарламалар листингі көрсетілген.

Фурье қатарына жіктелетін функцияны сипаттау үшін файл-функция листингі

```
function z=FF(t,T)
N=length(t);
% Тікбұрышты импульстің координаталар сипаттамасы
  for i=1:N
      if t(i)<0;
         z(i)=0;
      end;
      if (t(i)>=0)&(t(i)<=T/2);
         z(i)=-1/2;
      end;
      if (t(i)>T/2)&(t(i)<=T);
         z(i)=1/2;
       end;
      if t(i)>T;
         z(i)=0;
      end;
  end;
```

a_0 и a_n үшін (1) формулаға сәйкес косинустар бойынша Фурье қатарының жіктелуіндегі көрсетілген коэффициент мәні қайтарылатын файл-функция листингі

```
function z=AF(k,T)
% a0 и an коэффициенттерін есептеу
dt=T/1000; % абсцисса осі бойынша қадам шамасын есептеу
t=0:dt:T; % уақыт интервалының сипаттамасы
F=FF(t,T).*cos(2*pi*k/T*t); % Фурье қатары
% коэффициентінің есептелуі
z=2/T*trapz(t,F); % трапеция әдісі бойынша интегралды есептеу
```

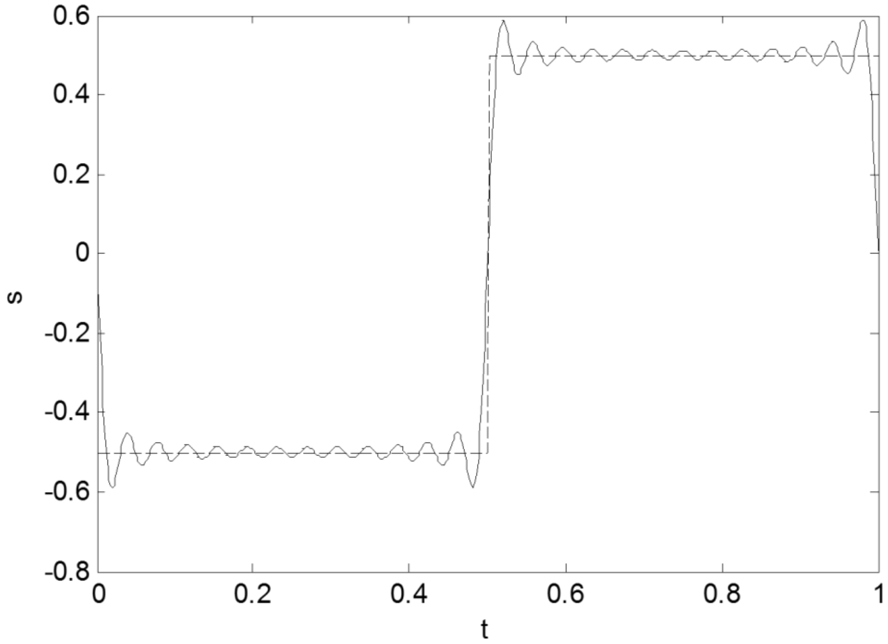
b_n үшін (1) формулаға сәйкес синустар бойынша Фурье қатарының жіктелуіндегі көрсетілген коэффициент мәні қайтарылатын файл-функция листингі

```
function z=BF(k,T)
% bn коэффициентін есептеу.
dt=T/1000; % абсцисса осі бойынша қадам шамасын есептеу
t=0:dt:T;      %      уақыт      интервалының      сипаттамасы
F=FF(t,T).*sin(2*pi*k/T*t); % Фурье қатары
% коэффициентінің есептелуі
z=2/T*trapz(t,F) % трапеция әдісі бойынша интегралды есептеу
```

Тікбұрышты импульстің визуализациясы және оны 25 гармоник бойынша Фурье қатарына жіктеудің файл-функция листингі

```
clear;clc;
Nf=25; % гармониктер саны
k=1:Nf;
T=1; % импульс ұзақтығы
A0=AF(0,1); % A(0) коэффициентін есептеу
  for k=1:Nf
     A(k)=AF(k,T); % A(k) коэффициентін есептеу
     B(k)=BF(k,T); % B(k) коэффициентін есептеу
  end;
% Фурье қатарының мәнін анықтау
% нөлден бірлікке дейінгі уақыт интервалында
Np=512; % t осінің интервалдарға бөліну саны
t=0:T/Np:1; % t осі бойынша есептеу
  for i=1:Np+1
      S=A0/2;
      for k=1:Nf
         S=S+A(k)*cos(2*pi*k/T*t(i))+B(k)*sin(2*pi*k/T*t(i));
      end;
      s(i)=S;
  end;
% Сигнал визуализациясы және оның Фурье қатарына жіктелуі
plot(t,s); hold on
plot(t,FF(t,T),'k--')
% Остерді белгілеу
xlabel('t'); ylabel('s')
```

Осы бағдарламаларды іске асырғанда алынатын есептеулердің нәтижесі 6.1-суретте көрсетілген график арқылы көруге болады



6.1 Сурет. 25 гармоникті қолданған кездегі тікбұрышты импульстің Фурье қатарына жіктелуі

Енді шуыл қосылған сигналдың спектральді функциясының тұрғызылу әдісін қарастырайық. 0-ден 0,6 секунд уақыт интервалында $x = \sin(2\pi \cdot 50t) + \sin(2\pi \cdot 120t)$, тасымалдаушы сигнал болсын делік, бұл сигналға $x_* = 2 \cdot \xi$ қатынасымен сипатталатын Гаусстың ақ шуылы қосылады, мұндағы ξ - кездейсоқ шама (яғни, нәтижесінде орташа мәні нөлге тең болатын шуылдық сигналмен зақымдалған сигнал алынады).
Керек:

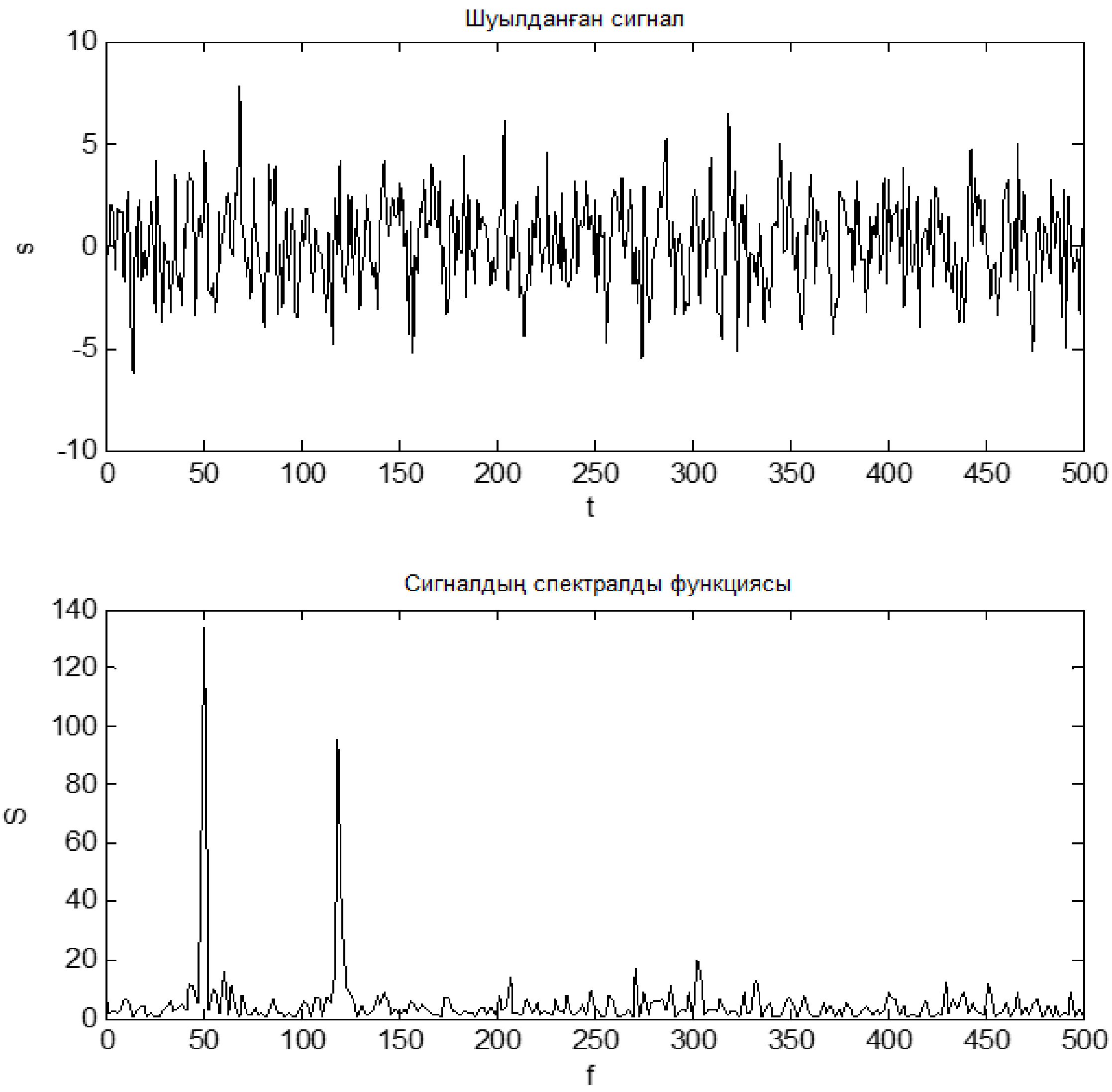
1) шуыл қосылған сигналдың реализациясын құру;
2) осы сигналдың спектральді функциясын құру.
3) төменде осы шарттарды реализациялайтын бағдарламалар листингі көрсетіледі.

Шуыл қосылған сигналдың визуализациясы және оның спектральді функциясын құру бағдарламасының листингі

```
clear;  clc;
% Шуыл қосылған сигнал визуализациясы
t = 0:0.001:0.6;  % Уақыт интервалы
% Тасушы сигнал
x = sin(2*pi*50*t)+sin(2*pi*120*t);
% Шуылдық сигнал
s = x + 2*randn(size(t));
subplot(211);
% s сигналының бастапқы 500 санағының визуализациясы
```

```
plot(1000*t(1:500),s(1:500))
xlabel('t'); ylabel('s');title('Шуылданған сигнал')
% s сигналының спектральді функциясының құрылуы.
Y = fft(s,512); % 512 нүкте бойынша сигналдың Фурье-
% түрлендіруі
S = Y.* conj(Y) / 512;
% conj функциясы комплексті-ұштасқан массивті қайтарады
% Y элементтерінің мәні
% Жиілік массивін формалау
f = 1000*(0:256)/512;
subplot(212);
% S спектральді функциясының визуализациясы
plot(f, S(1:257)); xlabel('f'); ylabel('S')
title(' Сигналдың спектралды функциясы ')
```

Осы бағдарламаның орындалу нәтижесі 6.2-суретте көрсетілген.



6.2 Сурет. Шуылдық сигнал мен оның спектральді функциясының реализациясы

Суреттен көретініміздей, шуылдың амплитудасы үлкен болғаны соншалықты тасушы сигнал шуылдық сигналда тіпті «көрінбейді». Бірақ тасушы сигналдың гармониктерін спектральді талдау арқылы анықтауға болады (төмендегі сурет). Спектральді функция екі үлкен шекке ие екенін – нақтырақ айтқанда, тасушы сигналдың гармониктерінде (50 және 120 Гц) екенін көру қиын емес. Графиктегі спектральді функцияның басқа да ерекшеліктері сәйкесінше аз және шуылдық компоненттерге байланысты.

Тапсырма

1. Жұмыстағы мысалдарда келтірілген нәтижелерді Matlab ортасында алыңыз.
2. Ара тәріздес және трапеция типтес сигналдарды, сонымен қатар тікбұрышты импульстердің еркіті санының реттілігін Фурье қатарына жіктеңіз.
3. Спектральді функция құрыңыз
   1) Синусоидалды сигнал;
   2) Шуыл қосылған синусоидалды сигнал;
   3) Шуыл қосылған ара тәріздес сигнал.

Бақылау сұрақтары

1. Сигналдарды Фурье қатарына жіктеген кезде олар қандай түрде келтіріледі?
2. Дирехле шартын формалаңыз.
3. Фурье қатарының коэффициенттерін анықтайтын формулаларды жазыңыз.
4. Периодты және периодты емес сигналдардың Фурье түрлендіруі.
5. Функцияның спектральді тығыздығының мағынасы қандай?
6. Функцияның спектральді тығыздығын есептейтін формуланы жазып, мағынасын түсіндіріңіз.
7. Фурьенің кері және тура түрлендіру формулаларын жазып, мағынасын түсіндіріңіз.
8. Тез Фурье түрлендіруі нені сипаттайды?
9. Тез Фурье түрлендіруі үшін Matlab-тың қандай функциялары қолданылады?
10. fft, ifft, trapz , conj, size және randn функциялары не үшін қолданылады?

Қолданылған әдібиеттер тізімі

1. Дьяконов В.П. Matlab 6.5 SP1/7+Simulink 5/6. Основы применения. – М.: СОЛОН-Пресс, 2005. – 800 с.

2. Мэтьюз Дж.Г., Финк К.Д. Численные методы. Использование Matlab. – М.: Изд. дом «Вильямс», 2010. – 720 с.

3. Жанабаев З.Ж., Иманбаева А.К., Алмасбеков Н.Е. Компьютерное моделирование в радиофизике и электронике. – Алматы: Қазақ университеті, 2005. – 143 с.

4. Дьяконов В.П. Matlab и Simulink для радиоинженеров. – М.: ДМК Пресс, 2011. – 976 с.

Сато Ю. Без паники! Цифровая обработка сигналов. – М.: Додека-XXI, 2010. – 178 с.

Зертханалық жұмыс №7

СИГНАЛДАРДЫҢ ВЕЙВЛЕТТІК АНАЛИЗІ

Жұмыстың мақсаты: сигналдардың вейвлеттік анализінің негізін оқып білу және Matlab ортасында оны жүзеге асырудың дағдыларын алу.

Қысқаша теориялық кіріспе

Егер функцияның жергілікті ерекшеліктері болса, соның ішінде, импульсті сигналдар мен суреттер үшін еркін функциялар мен синалдарды Фурье қатары түрінде көрсетудің тиімділігі аз. Бұл Фурье қатарының базистік функциясы болып табылатын синусоиданың -∞-тен +∞-ке дейінгі кеңістікте анықталғандығы мен тегіс және периодты функция болғандығымен байланысты. Сондықтан сигналдардың анализі кезінде бұл сигналдарды Фурье қатарына жіктеуге негізделген әдістермен оларды біркелкі көрсету туралы сұрақты шешуге мүмкіндік бермейтін теориялық шектеулерге негізделген қиындықтар туады.

Бұл мәселені шешу үшін сигналдың *вейвлеттік анализін* қолдануға болады. Сигналдар анализінің бұл түрі әртекті процесстер құылымын зерттеудің тиімді амалы болып табылады. «Wavelet» ағылшын сөзі (француз тілінен «ondelette») қысқа, кішкентай толқын деп аударылады. Вейвлет-анализінде s(t) сигнал гильберттік кеңістік функциясы ретінде (яғни ақырлы-өлшемдік) түсіндіріледі, ал гармоник орнына Фурье-түрлендіруінде келесі функция жүйесі қолданылады:

$$\psi_{a,b}(t)=\frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right). \qquad (7.1)$$

Бұл жүйе $\psi(t)$ бекітілген функциядан жақындаған сигналдардың әртүрлі созылуы мен жылжытулардан алынады, мұны Фурье қатарындағы гармоник жиіліктерінің өзгерісімен ұқсатуға болады. a параметрі вейвлет масштабын, ал b – оның орнын береді. $1/\sqrt{a}$ көбейткіші бұл функциялардың a масштабтаушы санынан норма тәуелсіздігін қамтамасыз етеді. a және b параметрлерінің берілген мәндері үшін $\psi_{a,b}(t)$ функциясы *вейвлет* болып табылады, ол $\psi(t)$ бас вейвлеттен туылады.

Осылайша, вейвлеттер нөлдік интегралдық мәні бар қысқа толқындар (шолп) түріндегі және осы өс бойынша жылжу мен масштабтауға (созылу мен сығылу) қабілетті тәйуелсіз айнымалылардың өс бойынша шектеуі бар ерекше функциялар болып табылады. Ең жиі қолданылатын вейвлет түрінің кез келгені функцияның толық ортогональды жүйесін тудырады. Сигналдың вейвлет-анализ жағдайында масштабтың өзгеруіне байланысты вейвлеттер әртүрлі шкалада процесс сипаттамасындағы айырмашылықты анықтауға қабілетті, ал жылжытудың көмегімен барлық зерттеліп жатқан интервалда әртүрлі нүктелердегі сигнал қасиеттерін талдауға болады. Бұл жүйенің дәл толықтық қасиетінің арқасында, процессті қалпына келтіруді (реконструкция немесе синтез) кері вейвлет-түрлендіруінің көмегімен жүзеге асыруға болады.

Вейвлеттерді Фурье түрлендіруін барлық уақыттық өс бойынша емес, өзінің орналасуы бойынша локалды жүзеге асыруға қабілетті кейбір толқындық функциялар ретінде жеткілікті дөрекі көрсетуге болады.

Вейвлеттер олардың түрін және қасиетін беретін арнайы базалық функция – прототиптің көмегімен жасалады.

Фурье түрлендіруіне қарағанда, вейвлет-түрлендіруі бір мәнді емес анықталған: әрбір вейвлетке өзінің түрленуі сәйкес келеді. Сигналды ажырату кезінде қолданылатын вейвлеттер саны сигнал декомпозициясының деңгейін анықтайды. Әрі декомпозицияның нөлдік деңгейін жиі сигналдың өзі қабылдайды, ал декомпозицияның келесі деңгейлері әдетте бір түрде жалбыраған вейвлет-ағашын құрады. Декомпозицияның төменірек деңгейлеріне өтуіне қарай сигналдың көрініс дәлдігі төмендейді, бірақ есесіне сигналдардың вейвлет-фильтрациясы, сигналдан шуларды алып тастау және сигналдың тиімді компрессиясы мүмкін. Осылайша, сигналдарды вейвлет-өңдеу мүмкін болады.

Практикалық қозғалыс үшін $\psi(t)$ функциясы оны вейвлет деп санау үшін ие болу керек негізгі белгілерді білу қажет.

Функция егер төменде айтылатын қасиеттерге ие болса вейвлет болып табылады.

1) *Үздіксіздік*. $\psi(t)$ функциясы анықтаудың барлық аймағында үздіксіз болу қажет.

2) *Шектілік*. $\psi(t)$ функциясы анықтаудың барлық аймағында интегралданатын болуы, ал оның норма квадраты ақырғы болуы қажет

$$\|\Psi\|^2 = \int_{-\infty}^{\infty} |\psi(t)|^2 dt < \infty \ . \tag{7.2}$$

3) *Нөлдік орташа*. $\psi(t)$ функция графигі абцисс өсінде нөлдің айналасында тербелуші болуы керек, яғни осы өске қатысты айнымалы таңбалы, және нөлдік ауданға ие болу керек:

$$\int_{-\infty}^{\infty} \psi(t)\, dt = 0. \tag{7.3}$$

$\psi(t)$ функция графигімен шектелген ауданның нөлге теңдігі осы функцияның фурье-түрлендіруі $w = 0$ жиілігінде нөлге тең және жолақтық фильтр, яғни кейбір ерекшеленген жолақта жиілікті өткізетін фильтр түріне ие екеніне сәйкес. а параметрінің әртүрлі мәндерінде бұл жолақтық фильтрлер жиынтығы болады.

Сигналдардың дәлірек вейвлет-анализі үшін тек $\psi(t)$ функциясының интегралы нөлге тең болуы ғана емес, келесі ара қатынастың орындалуы қажет

$$\int_{-\infty}^{\infty} t^n \psi(t) dt = 0\,. \tag{7.4}$$

n-ретті вейвлеттер, ереже бойынша, жіңішкерек құрылымды сигнал анализі үшін қолданылады, әрі оның баяу өзгеретін құраушылары басылады.

4) *Өзіне-өзі ұқсастығы*. Вейвлеттерге тән белгі олардың өзіне-өзі ұқсастығы болып табылады. Бұл қасиет кейбір тектің барлық вейвлеттері бас вейвлет ие осцилляция санына ие екендігінен көрінеді, себебі a параметрімен сипатталатын масштабты түрлендірулер және b параметрімен сипатталатын жылжу арқылы алынған.

Сигналдардың вейвлет-анализі Matlab Wavelet Toolbox кеңейту пакетінде жүргізіледі. Wavelet Toolbox-тағы вейвлеттерді түрі бойынша және жасаушы функцияның ерекшеліктері бойынша және вейвлетті алғаш ұсынған ғалымның аты бойынша жіктеу қабылданған. 1 кестеде осы пакетте бар вейвлеттер тізімі келтірілген. Сол жақта вейвлеттің атауы, ал оң жақта – осы вейвлетті сипаттау үшін қолданылатын Matlab функциясы көрсетілген.

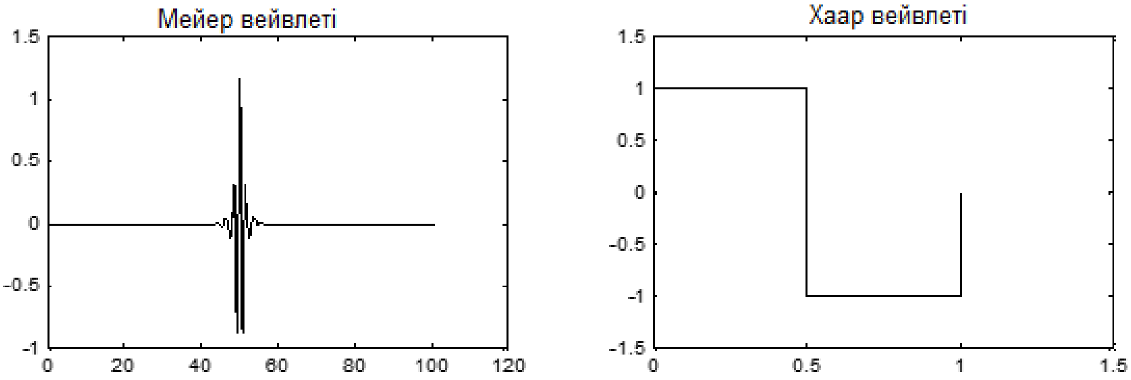
Кесте 1. Вейвлеттер түрі

Вейвлет түрінің атауы	
Вейвлеттің атауы	Вейвлетті сипаттау үшін Matlab функциясы
Haar (Хаар)	haar
Daubechies (Добеши)	db
Symlets (Симлет)	sym
Coiflets (Койфлетс)	coif
BiorSplines (биортогональды)	bior
ReverseBior (кері биортогональды)	rbio
Meyer (Мейер)	meyr
Dmeyer (Мейер вейвлетінің дискретті аппроксимациясы)	dmey
Gaussian (Гаусс)	gaus
Mexican_hat (мексикандық қалпақ)	mexh
Morlet (Морле)	morl
Complex Gaussian (Гаусс комплекстісі)	cgau
Shannon (Шеннон)	shan
FrequencyB-Spline (жиілікті В-сплайндық)	fbsp
Complex Morlet (Морле комплекстісі)	cmor

Matlab ортасында вейвлет визуализациясы үшін `wavefun` функциясы қолданылады. Төменде мысал ретінде Морлен және Хаар вейвлеттерінің визуализациясы үшін программа листингі, ал 7.1-суретте осы вейвлеттердің графигі келтірілген.

Морле вейвлетінің визуализациясы үшін программа листингі

```
clear; clc % Workspace тазартуы және Command Window
[phi,psi,x]=wavefun('dmey'); % Мейер вейвлетін құру
subplot(121); plot(x,psi); % Мейер вейвлетінің визуализациясы
title ('Мейер вейвлеті') % графикке атау жасау

[phi,psi,x]=wavefun('haar');  % Хаар вейвлетін құру
subplot(122); plot(x,psi)  % Хаар вейвлетінің визуализациясы
title ('Хаар вейвлеті') % графикке атау жасау
```



7.1-сурет. Мейер және Хаар вейвлеттері

Сигналдардың анализі үшін үздіксіз, дискретті және жылдам вейвлет-түрлендірулер қолданыла алады.

Үздіксіз вейвлет-түрлендіру

S(t) сигналының үздіксіз вейвлет-түрлендіруі келесі түрде сипатталады:

$$W_s(b,a) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} s(t)\psi\left(\frac{t-b}{a}\right)dt. \tag{7.5}$$

Осылайша, вейвлеттер вейвлет-функцияға берілген сигналдың скаляр көбейтіндісінің интеграл мәнімен анықталады. Кері үздіксіз вейвлет-түрлендіру бастапқы сигналды қалпына келтіруге мүмкіндік береді және келесі ара қатынаспен сипатталады

$$s(t) = \frac{1}{C_\psi} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} W_s(a,b)\psi(t)\frac{dadb}{a^2}, \tag{7.6}$$

мұндағы C_ψ – нормалаушы коэффициент.

Wavelet Toolbox пакетінде зерттеліп жатқан сигналдардың вейвлет-спектрограммаларын алуға болады. Вейвлет-спектрограммалар «масштаб-уақыт» жазықтығындағы вейвлет-коэффициенттер мәнін көрсетеді. Спектрограмманың төменгі бөлігі коэффициенттердің аз нөмеріне сәйкес сигналдардың егжей-тегжейлі суретін, ал жоғарғысы – бірнеше дөрекіленгенін көрсетеді. Спектрограмманың жоғарғы бөлігі a параметрінің үлкен мәндеріне сәйкес. Гармоникалық сигнал визуализациясы үшін программа листингі және оның вейвлет-спектрограммасы төменде келтірілген.

Спектрдің континуал түрлендіруін орындайтын және түрлендірудің спектралды коэффициенттер массивін қайтаратын `cwt` функциясын “s” сигналының вейвлет-спектрограммасын құру үшін қолданылады. Бұл функцияның синтаксисі келесідей:

```
c = cwt (s, scales, 'wname', mode)
```

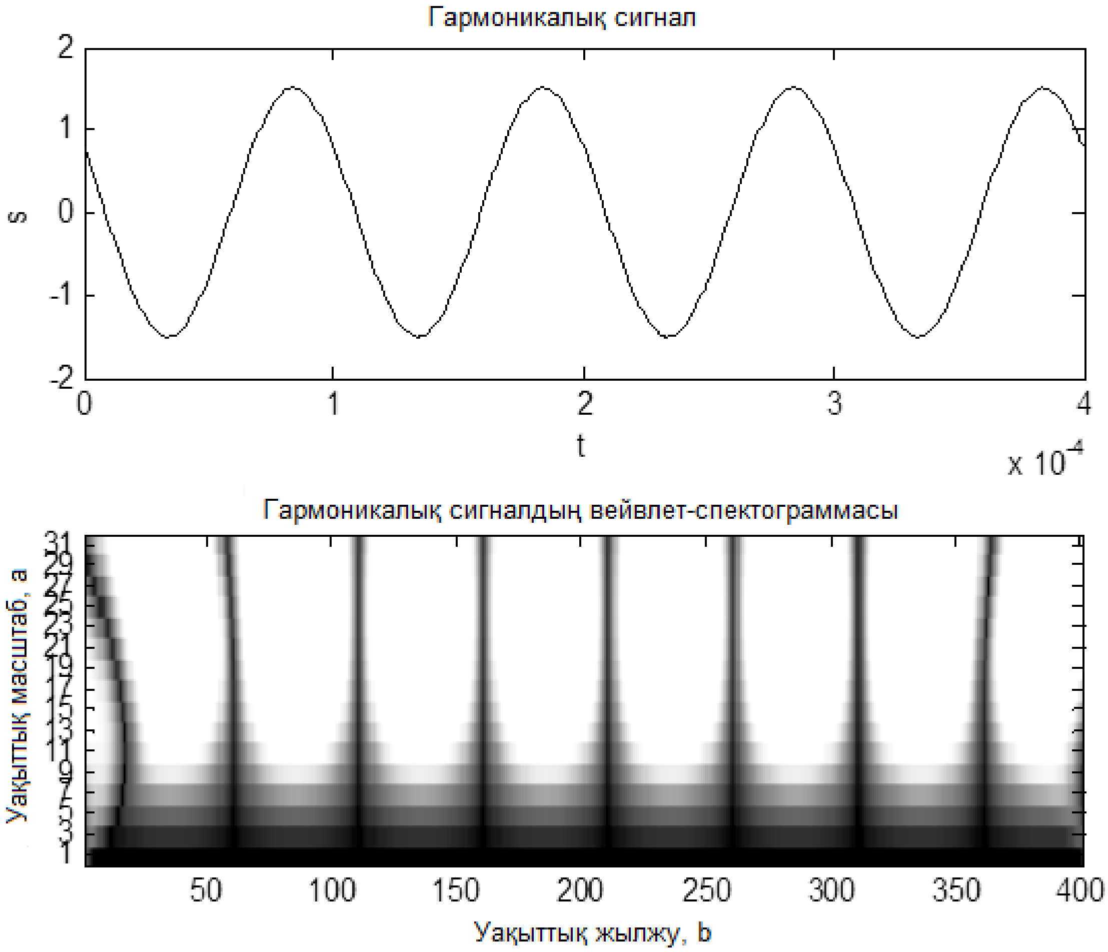
Мұндағы `scales` – масштабтаушы айнымалы базистік а вейвлет мәнінің векторы. wname параметрі вейвлет-спектрограмманы құру үшін қолданылатын вейвлет атауын анықтайды. `mode` функциясы вейвлет-спектрограмманың контрасттылығын және тегістеуін реттеуге мүмкіндік береді. Осылай, `‘lvl’` мәнін mode ретінде жазу сигналдың спектралды коэффициенттерінің сурет контрасттылығын жақсартуға, `‘glb’` мәні – түрлендіру коэффициентінің спектралды шуылын тегістеуге мүмкіндік береді, бұл тұтас алғанда спектралды суреттің түстілігін тегістейді. Сурет контрасттылығын реттеу үшін `'abslvl'` және `'lvlabs'` функциялары, ал спектрдің түстік гаммасын тегістеу үшін - `'absglb'` немесе `'glbabs'` функциялары қолданылады.

Төменде мысал ретінде $s = A\cos(2\pi wt + \varphi)$ гармоникалық сигналдың вейвлет-спектрограммасын құру үшін «мексикандық қалпақ» вейвлетін қолданумен программа келтірілген. Сигнал графигі және оның вейвлет-спектрограммасы 7.2-суретте кетірілген.

Гармоникалық сигналды және оның вейвлет-спектрограммасын жүзеге асыру визуализациясы үшін программа листингі

```
clear; clc
t = 0:0.000001:0.0004;  % Уақыттық интервал
A= 1.5; % Сигнал амплитудасы
w = 10000;  % Жиілік
phi = 0; % Бастапқы фаза

s = A*cos(2*pi*w*t+phi); % Гармоникалық сигнал
axis([0 0.0004 -3 3]); % Координат өстері бойынша
% шекті мәндерді беру

grid on;  % Графикке тор түсіру
subplot(2,1,1) ;
plot(t,s);  % Гармоникалық сигнал визуализациясы
title('Гармоникалық сигнал')
xlabel('t');  ylabel('s');

% Вейвлет-спектрограмма құру
subplot(2,1,2);
```

```
c = cwt(s, 1:2:32, 'mexh', 'abs1vl', [0 10]);
title('Гармоникалық сигналдың вейвлет-спектрограммасы');
xlabel('Уақыттық жылжу, b');
ylabel('Уақыттық масштаб, a');
```



7.2 Сурет. Гармоникалық сигнал және оның вейвлет-спектрограммасы

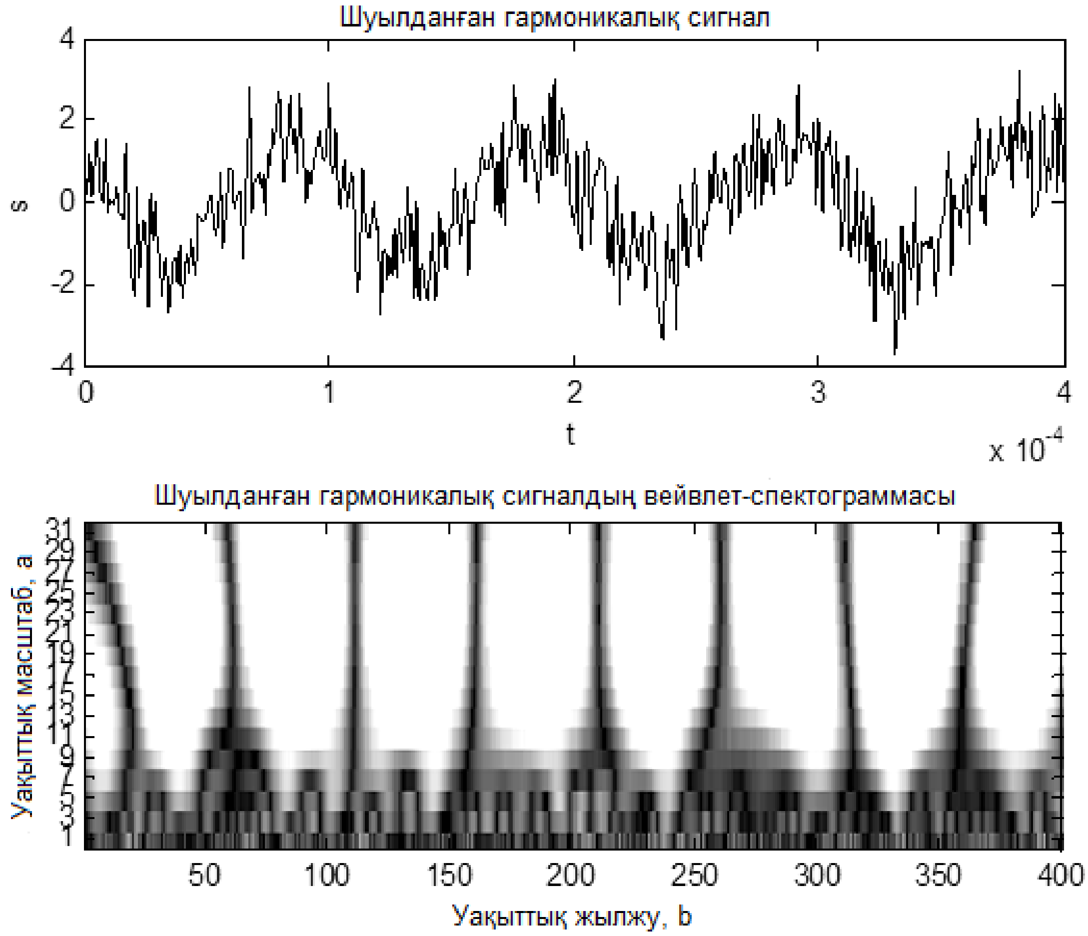
Вейвлет-спектрограмманың түрі бойынша сигналда шуылдың бар екені туралы жорамалдау оңай. Мұны көрнекі көрсету үшін алдыңғы суретте көрсетілген $s = A\cos(2\pi wt + \varphi)$гармоникалық сигналға амплитудасы An ақ шуыл қосамыз және шуылды сигналдың вейвлет-спектрограммасын құрамыз. Осы мақсатпен жоғарыда келтірілген программаны келесі түрде қайта жазамыз.

Шуылды гармоникалық сигналды және оның спектрограммасының реализациясын құру үшін программа листингі

```
clear; clc;
t = 0:0.000001:0.0004;  A = 1.5; w = 10000;  phi = 0;
An=0.8;% Шуыл амплитудасы
```

```
s = A*cos(2*pi*w*t+phi)+An*randn(size(t));
axis([0 0.0004 -3 3]); grid on;
subplot(211) ;plot(t,s);
title('Шуылды гармоникалық сигнал')
xlabel('t');  ylabel('s');
% Шуылды гармоикалық сигналдың
% вейвлет-спектрограммасын құру
subplot(212);
c = cwt(s, 1:2:32, 'mexh', 'abs1vl', [0 10]);
title('Шуылды гармоникалық сигналдың вейвлет-спектрограммасы');
xlabel('Уақыттық жылжу, b');
ylabel('Уақыттық масштаб, a');
```

Бұл программаның нәтижесі 7.3-суретте келтірілген графиктер болып табылады.



7.3 Сурет. Шуылды гармоникалық сигнал және оның вейвлет-спектрограммасы

Жоғарыда келтірілген графиктерді салыстырудан шуылсыз сигналдың және шуылды сигналдың вейвлет-спектрограммалары бір-

бірінен едәуір ерекшеленетіні шығады. Сондықтан сигналдардың вейвлет-анализі оларда шуылдың барын бағалау үшін қолданыла алады.

Вейвлет-анализ тек сигналда шуылдың бар екенін анықтау үшін ғана емес, сигналдардың басқа да – ажыраулар, туынды белгісінің өзгеруі және т.б. әртүрлі ерекшеліктерін анықтау үшін қолданылады. Сигналдың қандай да бір ерекшелігі анығырақ байқалған сайын, ол оның вейвлет-спектрограммасында күштірек ерекшеленеді. Осылайша, вейвлет-анализ сигналдардың сезімтал ерекшеліктерінің толық анализін жүргізуге мүмкіндік береді. Бұл әсіресе дыбысты сигнал үшін, сондай-ақ сурет сигналдар үшін маңызды, себебі мұндай ерекшеліктер олардың қалпына келтіру қасиетін анықтайды.

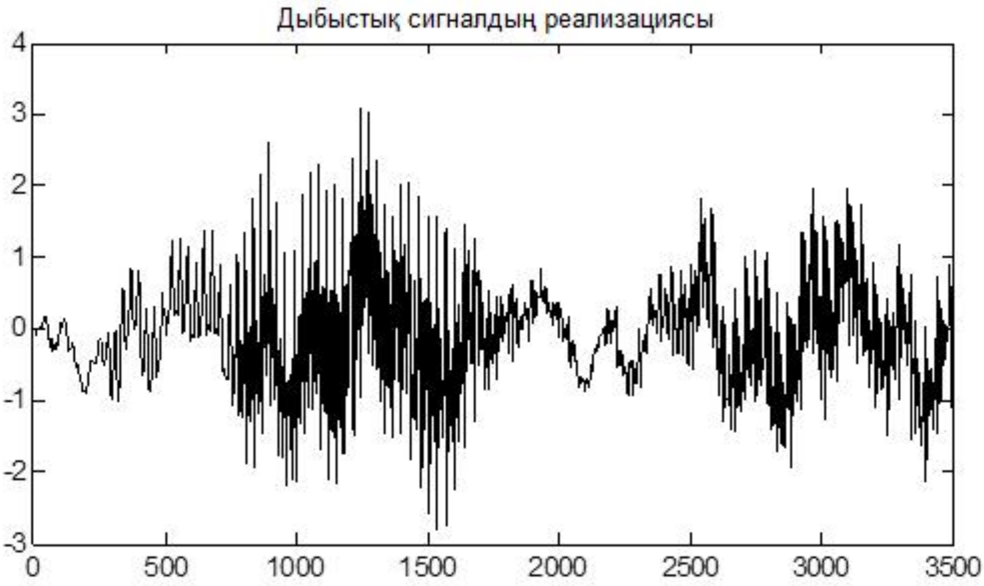
Дыбысты сигналдың вейвлет-сректрограммасын құру алгоритмінің біршама ерекшеліктері бар. Осындай спектрограмманы тестілік дыбысты сигналды өңдеу мысалында құру әдісін қарастырайық.

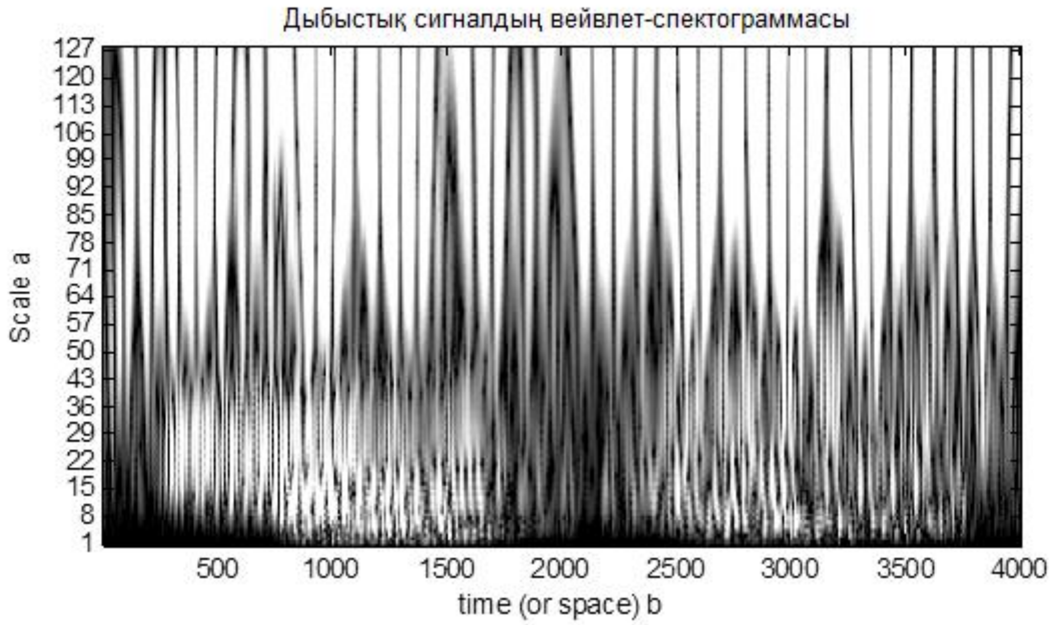
Дыбысты сигналды және оның вейвлет-спектрограммасын құруды жүзеге асыру визуализациясы үшін программа листингі

```
clear; clc
load mtlb; % Тестілік дыбысты файлды жүктеу
v=mtlb(1:4000); % Сигналды өңдеу үшін 1-ден 4000-ға
% дейін санау нөмерін таңдау
lv = length(v); % Дыбысты сигналдың отсчетынан тұратын
% массив элементтерінің санын анықтау
subplot(211); plot(v); % Дыбысты сигналды
% жүзеге асыру визуализациясы
title('Дыбысты сигналды жүзеге асыру'); % Графикке
% атау жасау
set(gca,'Xlim',[0 3500]);
% set функциясы графикалық объекттердің қасиеттерін
% беру үшін қолданылады
% gca функциясы (get current axes)ағымдағы өстерге
% көрсеткіш алу үшін қолданылады
% [0 3500] жазуы абсциссаның максималды және
% минималды мәндерін анықтайды
[c,l] = wavedec(v,5,'sym2');
% wavedec функциясы вейвлет базисі (бұл жағдайда Симлет
% вейвлеті) бойынша v векторымен берілген деректерді
% көпдеңгейлі дискретті жіктеу процедурасын жүзеге асырады
% 5 – бастапқы деректерді жіктеудің максималды деңгейінің
нөмері
cfd = zeros(5,lv); % 5 x 1v өлшемді
% массивтің нөлдерін қалыптастыру
subplot(212)
ccfs = cwt(v,1:128,'sym4','plot'); % Дыбысты сигналдың
% вейвлет-спектрограммасын құру
```

```
title('Дыбысты сигналдың вейвлет-спектрограммасы')
colormap(pink(64)); % Вейвлет-спектрограмма түстер
% палитрасын беру
ylabel('Scale a')
```

Программаны іске қосу нәтижесінде 7.4-суретте көрсетілген графикті аламыз.





7.4 Сурет. Дыбысты сигналды және оның вейвлет-спектрограммасын жүзеге асыру

Сондай-ақ Wavelet Toolbox кеңейту пакетінде үшөлшемді вейвлет-спектрограмманы құруға болады. Ол үшін cwt функциясы қолданылды, келесі түрде жазылады:

```
c = cwt (s, scales, 'wname', '3Dplot')
```

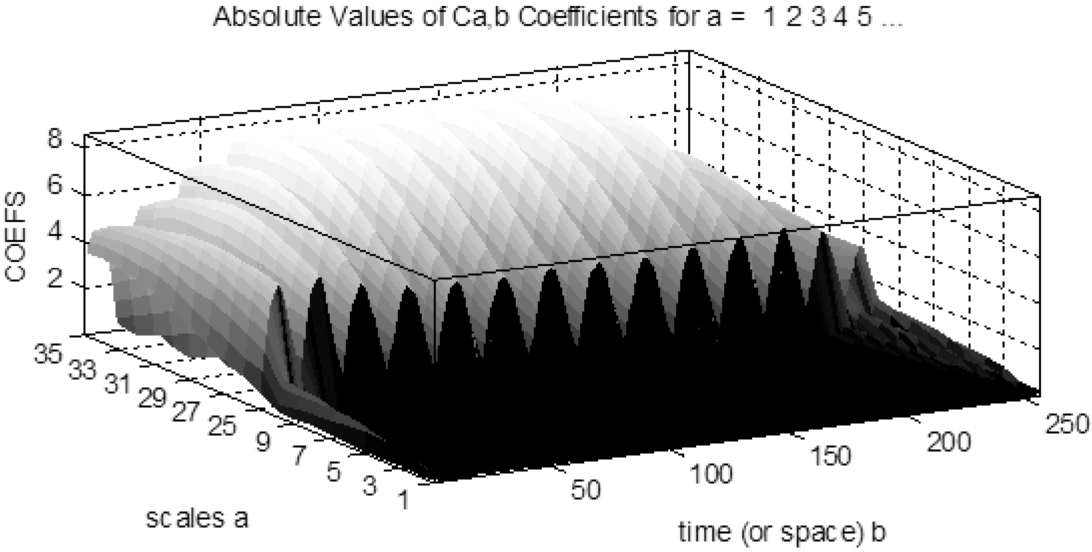
Сигналдың үшөлшемді вейвлет-спектрограммасын екі компоненттен тұратын және келесі $s = A_1\cos\left(0.05\pi t + \varphi_1\right) + A_2\cos\left(0.02\pi t + \varphi_2\right)$ теңдеумен сипатталатын, мұндағы $A_1 = 1.2$, $A_2 = 1.6$, $\varphi_1 = \pi/3, \varphi_1 = \pi/4.$, Морле вейвлетінің көмегімен құруды қарастырайық.

Бұл сигналды жүзеге асыру визуализациясы және көлемді вейвлет-спектрограмма құру үшін келесі программаны жазамыз.

Сигналдың үшөлшемді вейвлет-спектрограммасын құру үшін программа листингі

```
clear; clc;

% t торында сигналдың анықтау облысы
t = 1:256;
phi1=pi/3; phi2=pi/4; % сигнал құраушыларының
% бастапқы фазалары
A1=1.2; A2=1.6; % сигнал құраушыларының амплитудалары

% Сигналдың сипаты:
s = A1*sin(0.05*pi*t+phi1)+A2*sin(0.02*pi*t+phi2);

% Сигнал визуализациясы:
subplot(211); plot(t,s);

title('Сигналды жүзеге асыру')

% Масштабтаушы айнымалы мәнінің векторы
scales = [1: 10, 25: 35];

% Спектрдің үшөлшемді визуализациясымен континуал түрлендіру
subplot(212);
c=cwt (s, scales, 'morl', '3Dplot');
```

Бұл программаның орындалу нәтижесі 7.5-суретте көрсетілген. Горизонталь өстер бойынша вейвлет-түрлендірудің *a* және *b* параметрлерінің мәндері қойылған.



7.5 Сурет. Зерттеліп жатқан сигналдың және оның үшөлшемді вейвлет-спектрограммасын жүзеге асыру

Сигналды шуылдан тазарту және компрессия

Вейвлет-анализ сигналдарды шуылдан тазарту және сығу (компрессия), фильтрация мәселелері үшін де қолданыла алады.

Вейвлеттерді қолдану вейвлет-түрлендіру коэффициенттерінің деңгейін шектеуге мүмкіндік береді. Олардың деңгейі үшін белгілі шек беріп және мәндері осы шектен төмен коэффициенттерді шығарып, шуылдың деңгейін төмендетуге, сондай-ақ сигналды сығуға болады. Шектік деңгейді әрбір коэффициентке жеке орнатуға болады.

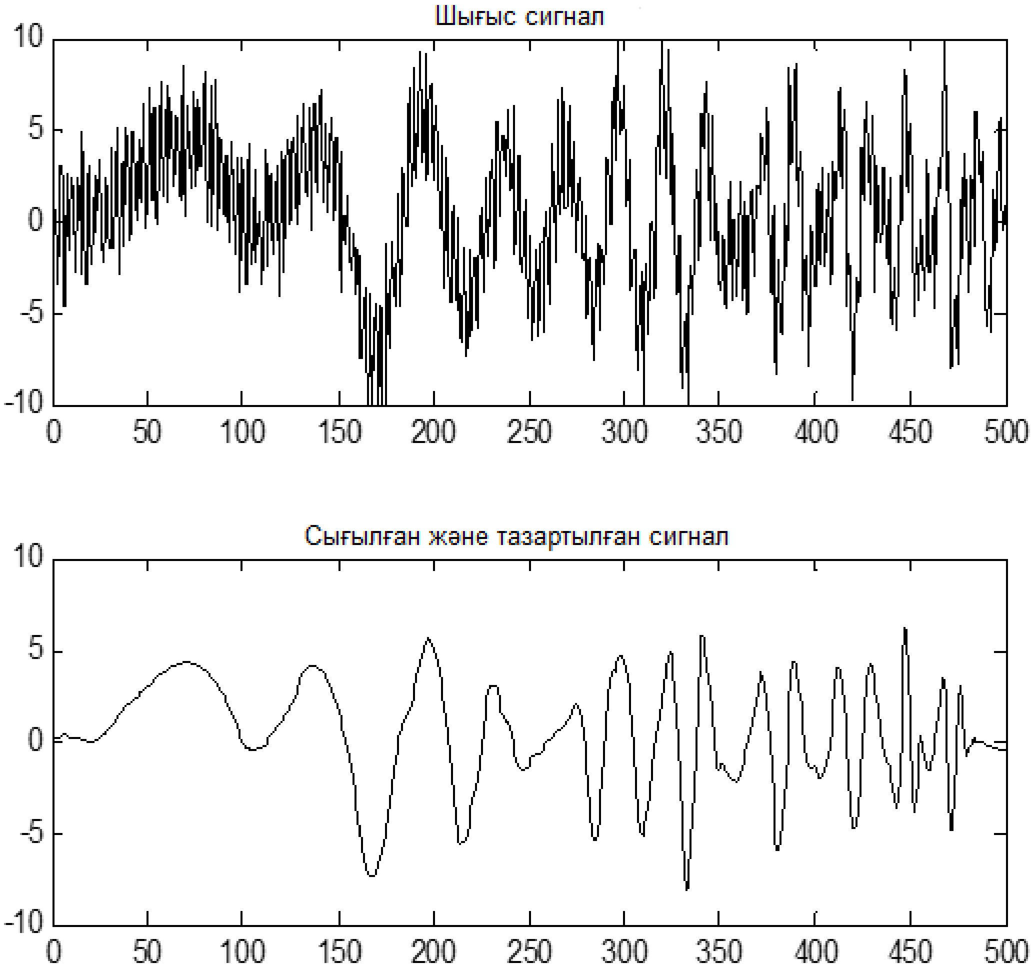
Сигналды сығу және шуылды жою процедурасы үш кезеңнен тұрады. Бірінші кезең `lev` деңгейі үшін сигналды жіктеу болып табылады, әрі вейвлет түрі және декомпозиция деңгейі таңдалады. Бұл үшін `wavedec` функциясы қолданылады. Екінші кезеңде талдап тексеруші коэффициенттер үшін белгілі шек таңдау жүзеге асырылады. Осы мақсатпен орнатылған деңгейге қатысты шекті және сақталған коэффициенттер санын қайтаратын `wdcbm` функциясы қолданыла алады. Үшінші кезең `wdencmp` функциясының көмегімен сигналдың вейвлет-қалпына келуі болып табылады.

Сигналды шуылдан компрессиялау және тазалау процедурасын *noismima* функциясымен тудырылған тестілік шуылды сигналды өңдеу мысалында қарастырайық.

Сигналды, оның компрессиясы мен шуылдан тазалауды іске қосу, сондай-ақ бастапқы және тазаланған сигналдарды құру листингі

```
clear;
clc;
load noismima; % Тестілік сигналды жіктеу
x = noismima; % x массивіне сигнал есептеулерін енгізу
wname = 'db4';  % Вейвлет таңдау (бұл жағдайда Добеши
% вейвлеті)
lev = 5; % Декомпозиция деңгейін беру
[c,l] = wavedec(x,lev,wname);  % Сигнал декомпозициясы
alpha = 2; % Ереже бойынша, alpha параметрі сығу үшін 1,5
% және шуылды жою үшін 3,0 тең деп есептеледі
m = 2*l(1);  % Шекті баптау параметрі
[thr,nkeep] = wdcbm(c,l,alpha,m); % Орнатылған деңгейге
% қатысты шекті беру
[xd,cxd,lxd,perf0,perfl2]=…                              =…
…wdencmp('lvd',c,l,wname,lev,thr,'h');% Сигналдың
% вейвлет-қалпына келуі
subplot(211);
plot(x); % Бастапқы сигнал визуализациясы
title('Бастапқы сигнал');
axis([0,500,-10,10]); % Абсцисс және ординат өстері
% бойынша шектер
subplot(212);
plot(xd); % Сығылған және тазаланған сигнал визуализациясы
title('Сығылған және тазаланған сигнал');
axis([0,500,-10,10]);
```

Осы программаға сәйкес есептеу нәтижесі төменде келтірілген суретте көрсетілген.



7.6 Сурет. Бастапқы сигнал мен сығылуға және шуылдан тазартуға ұшыраған сигнал

Осылайша, сигналдарлың вейвлет-анализі сигналды сығуды және оны шуылдан тазартуды жүзеге асыруға мүмкіндік береді.

Тапсырма

1. Шуылсыз ара тәрізді сигналдың және шуылды ара тәріздес сигналдың вейвлет-спектрограммаларын құрыңыз және салыстырыңыз.
2. Шуылсыз және шуылды тікбұрышты импульстер реттілігінің вейвлет-спектрограммасын құрыңыз және салыстырыңыз.
3. Ақ шуылдың үшөлшемді вейвлет-спектрограммасын құрыңыз.
4. Берілген дыбысты сигналдың вейвлет-спектрограммасын құрыңыз.
5. Вейвлет-түрлендірудің көмегімен шуылды гармоникалық, шуылды тікбұрышты және шуылды ара тәріздес сигналдарды шуылдан тазартуды және компрессияны жүзеге асырыңыз.

Бақылау сұрақтары

1. Қандай функция вейвлет деп аталады?
2. Вейвлеттің негізгі қасиеттерін атаңыз.

3. Үздіксіз вейвлет-түрлендіру деген не?
4. Matlab-тың қандай функциялары вейвлеттер визуализациясы үшін қолданылады?
5. Сигналдардың вейвлет-анализі не үшін қолданылады?
6. Вейвлет-спектрограмма нені көрсетеді?
7. Шуылды сигналдың және шуылсыз сигналдың вейвлет-спектрограммалары қалай ерекшеленеді?
8. Matlab-та екіөлшемді және үшөлшемді вейвлет-спектрограманы қалай құруға болады?
9. Вейвлет-түрлендірудің көмегімен сигналды сығу және шуылды жою процедурасы қандай кезеңдерден тұрады?
10. *wavedec, wdcbm, wdencmp* функцияларының міндетін сипаттаңыз.
11. Сигналдарды шуылдан тазарту және компрессия үшін WaveletToolbox пакетінің қандай функциялары қолданылады?

Қолданылған әдібиеттер тізімі

1. Дьяконов В.П. Matlab и Simulink для радиоинженеров. – М.: ДМК Пресс, 2011. – 976 с.
2. Яковлев А.Н. Введение в вейвлет-преобразования. – Новосибирск: Изд-во НГТУ, 2009. – 104 с.
3. Жанабаев З.Ж., Иманбаева А.К., Алмасбеков Н.Е. Компьютерное моделирование в радиофизике и электронике. – Алматы: Қазақ университеті, 2005. – 143 с.
4. ДьяконовВ.П. Matlab 6.5 SP1/7+Simulink 5/6. Основы применения. – М.: СОЛОН-Пресс, 2005. – 800 с.
5. Сергиенко А.Б. Цифровая обработка сигналов. СПб.: БХВ_Петербург, 2013. – 768 с.
6. Сато Ю. Без паники! Цифровая обработка сигналов. – М. Додека-XXI, 2010. – 178 с.

Зертханалық жұмыс №8

СИГНАЛДАРДЫҢ АҚПАРАТТЫҚ-ЭНТРОПИЯЛЫҚ АНАЛИЗІ

Жұмыстың мақсаты: «Ақпарат» және «энтропия» түсініктерін қарастыру, сигналдың ақпараттық-энтропиялық анализінің негіздерін қарастыру, ақпараттық энтропиялық түрлі сигналдарды есептеудің бағдарламасын құрудың дағдысын алу.

Қысқаша теориялық кіріспе

Ақпараттық-энтропиялық анализ ашық жүйені сипаттаудың тиімді құрамы болып табылады, мұндағы аталған ашық жүйе қоршаған ортамен энергия, заттар және ақпарат көмегімен жаңғыртылып отырылады.

Ақпарат

Ақпарат сөзінің мағынасына үңіліп көрейік. Ықтималдылық теориясында ақпарат деп аддитивті өлшемді қарастырамыз (өлшем), ол түрлі құбылыстардың ықтималдылығын салыстыру үшін қызмет етеді. Мұндағы маңызды нәрсе ақпараттың сандық тұрғыдан бағалануы болып табылады.

Ақпарат объекттің табиғатына тәуелсіз түрде симметрияның, құрылымның және зерттеліп отырылған процестің ықтималдылық бұзылуынан пайда болады. Құрылым сәйкесінше өздігінен құрылатын процестер үшін сипатталатындықтан, ақпараттық сандық мәні осындай процестерді сипттау үшін қолданылуы мүмкін.

Саны $i = 1, 2, 3...$ болатын бірнеше оқиға қарастырылатын болсын. Оқиға $i -$жылын P_i ықтималдылығы арқылы белгілеп аламыз. Онда

$$I_i = -\ln P_i \tag{8.1}$$

мәні осы оқиға туралы ақпараттар санын көрсететін болады. Оқиғаның ықтималдылығы дегеніміз 0-ден 1-ге дейінгі аралықтағы өлшем болғандықтан, ақпараттық сандық мәні қашан да теріс емес болады. Ақпарат санының өлшем бірлігі екілік, ондық және натуралды логарифмдер бойынша (8.1) формулаға сәйкес биттер, диттер және наттар болып табылады.

Қос тәуелсіз оқиғаның бір уақытта іске асу ықтималдылығы осы оқиғалардың ықтималдылық қосындысына тең. Сондықтан да аталған оқиғалардың бір уақытта іске асуы туралы ақпарат осы оқиғаға сәйкес келетін ақпараттың сандық мәнінің қосындысына тең болады. Бұл ақпараттың аддитивтілік қасиетіне сәйкес келетіндігін көрсетеді. Аддитивтілік- қосуға қатысты өлшемдердің қасиеті болып табылады, толық объектіге сәйкес келетін өлшем мәнінің қосындысына тең. Кез-келген аддитивті өлшем анықталуы бойынша өлшем болып саналады.

(8.1) формулаға сәйкес ықтималдылықпен сипатталатын нөлге ұмтылатын оқиға туралы ақпарат максималды мәнді қабылдайды, сонымен қатар ықтималдылығы бар бірге тең (100%) оқиғаға сәйкес ақпарат ретінде нөлге ұмтылады. Демек, қайта-қайта қайталанатын оқиғаға қарағанда сирек кездесетін оқиғаның ақпараты көбірек болады, сәйкесінше, мұнда үлкен ықтималдылық пен оқиғалар болады.

Энтропия (грек тілінен аударғанда айналыс, бұрылыс деген мағына береді) ашық жүйелердің маңызды фактілік сипаттамасы болып табылады және мынадай қызмет атқарады:

1) жүйенің статистикалық сипаттамсы кезінде белгісіздік өлшемі ретінде;

2) ашық жүйелердің тепе-теңдік жағдайындағы қатыстық реттелу деңгейінің өлшемі ретінде;

3) эволюция теориясындағы түрлілік ретінде қызмет атқарады.

Энтропия туралы алғашқы түсінікті Клаузиус 1865 жылы термодинамикада энергияның қайтымсыз таралуын сипаттау үшін қолданды. Термодинамикалық процестерді сипаттаған кезде жүйе жағдайы болатын S функциясын энтропия ретінде қарастырамыз, оның дифференциялы элементарлы қайту процесінде хабарланған жүйе жылуының термодинамикалық температурасында Т:

$$dS = \frac{\delta Q}{T} \qquad (8.2)$$

шексіз кіші қатынасының $\delta Q,$ мәніне тең болады.

Күрделі жүйенің энтропиясы аддитивті өлшем болып табылады, яғни ол барлық біртекті бөліктердің энтропиясының қосындысына тең.

(8.2) формуласын интегралдаған кезде энтропияның сандық мәні бірнеше тұрақтығы дейінгі дәлдікпен ғана анықталуы мүмкін. Қайтымды процесс іске асатын оқшауланған жүйенің энтропиясы тек жоғарылауы мүмкін, егер жүйеде жүріп жатқан процесс қайтымды болса, энтропия тұрақты болып қалады.

Энтропия жүйенің ретсіздік өлшемін сипаттайды: жүйе реттілігі неғұрлым төмен болса, оның энтропиясы соғұрлым жоғары болады.

Больцман бойынша қарастыратын болсақ, энтропияны статистикалық түрде түсіндіруге болады, сондықтан да, статистикалық физикада энтропия оның қандай да бір макроскопиялық жағдайын жүзеге асыру өлшемі болып табылады. Статистикалық физикада энтропия қосалқы жүйенің $\Delta\Gamma$ макроскопиялық күйінің статистикалық салмағының логарифмі арқылы мына түрде анықталады:

$$S = \ln \Delta\Gamma \tag{8.3}$$

Кванттық механика бойынша статистикалық салмақ деп бірдей энергиясы бар, яғни энергетикалық деңгейдің пайда болу дәрежесі жүйенің түрлі кванттық күйдің санын анықтайтын физикалық өлшемді айтамыз. Статистикалық физикада статистикалық салмақ жүйенің аталған макроскопиялық күйі жүзеге асырылуы мүмкін болған әдістер санын анықтайды.

К. Шеннон энтропия түсінігін беріліп жатқан хабарламадағы белгісіздік деңгейін сипаттау үшін қолдану керек деп ұсынды. Бұл жағдайда энтропия мына түрде жазылады

$$S = -\sum_{i=1}^{N} P_i \ln P_i ,\; i = 1, 2, 3... \tag{8.4}$$

және *ақпараттық энтропия* деп аталады.

Ақпараттық энтропия ақпараттың ортаықтималды мәнін анықтайды және жүйедегі біз қарастырып отырған мәнді сипаттайды. Егер жүйе тепе-тең күйде болатын болса, онда оның ақпараттық энтропиясы максималды болады. Демек, тепе-тең жүйе ақпаратты сақтай алмайды. Жүйе туралы ақпарат алу ол туралы белгісіздіктің төмендеуі арқылы түсіндіріледі. Осылайша, ақпарат келесі қатынас арқылы жазылуы мүмкін:

$$I = S_1 - S_2 . \tag{8.5}$$

Мұнда S_1 – априорлық (тәжірибеге дейінгі) энтропия, ал S_2 – апостериорлық (тәжірибеден кейін). Жүйе туралы белгісіздік деңгейін арттырған кезде бұл жүйенің ақпараттық энтропиясы артады.

Күрделі жүйелердің өздігінен құрылу деңгейінің критерийлері. Ашық жүйелердің өздігінен құрылу деңгейін сипаттау үшін қолданылатын сандық критериі профессор З.Ж. Жаңабаев тарапынан ұсынылған.

Осы параметрлерді енгізу идеясы мен олардың физикалық мағынасын қарастырайық.

Ақпаратты тәуелсіз айнымалы ретінде қарастырайық. (1) формулаға сәйкес ақпараттың I іске асуының ықтималдығы $P(I)$

$$P(I)=e^{-I}. \tag{8.6}$$

Ықтималдылықтың тығыздығы $f(I)$ мына қатынас арқылы анықталады

$$0\le P(I)\le 1,\ 0\le I\le\infty,\ \int_0^{\infty} f(I)dI=1,$$

$$P(I)=\int_I^{\infty} f(I)dI,\ f(I)=P(I)=e^{-I}. \tag{8.7}$$

Ақпараттың іске асу ықтималдылығының $P(I)$ функциясы ықтималдылық тығыздығының $f(I)$ таралу функциясына сәйкес келеді. (1) формула арқылы анықталатын ақпарат масштабтық инварианттылық қасиетімен сипатталады, яғни бөлік пен толық ақпарат таралудың бір заңы арқылы сипатталады. Ақпарат мәнінің таралуының $S(I)$ ақпараттық энтропиясын ақпараттық орта мәні ретінде анықтаймыз:

$$S(I)=\int_I^{\infty} If(I)dI=(1+I)e^{-I}. \tag{8.8}$$

$0\le I\le\infty$ үшін $1\ge S\ge 0$болады, яғни энтропия бірге нормаланған болады. Үзіліссіз мәндердің энтропиясы айнымалылардың секірісті өзгерісі кезінде шексіз болатыны анық, сондықтан да интеграл Лебег түсінігі бойынша бірнеше өлшемдерді енгізу арқылы анықталады. Өлшем ретінде біз ақпараттың өзін алдық және (8.8) нәтижесін алдық.

Осыған дейін белгілі болған функционалдық теңдеуді қолданамыз, мұнда кейбір физикалық өлшемдердің x іске асуы масштабтық-инвариантты қасиет арқылы кей сипаттамалық $g(x)$функцияны қанағаттандырады:

$$g(x)=\alpha g\left(g(x/a)\right), \tag{8.9}$$

мұндағы $\alpha-$ масштабтық көбейткіш. Кез-келген үздіксіз функция өзінің қозғалмайтын нүктесінде (8.9) теңдеуін қанағаттандырады. Сипаттамалық функция ретінде $f(I)$ және $S(I)$ қабылдай отырып, олардың қозғалыссыз нүктесін анықтаймыз :

$$f\left(I\right)=I,\ e^{-I}=I, I=I_1=0.567, \tag{8.10}$$

$$S\left(I\right)=I,\ (1+I)e^{-I}=I,\ I=I_2=0.806. \tag{8.11}$$

Бұл қозғалыссыз нүктелер бірегей тұрақтылар болып табылады, себебі, олар I_0 кез-келген бастапқы мәнінде қол жеткізілетін шексіз көріністердің шегі болып саналады:

$$I_{i+1}=f\left(I_i\right),\ \lim_{i\to\infty}\exp\Big(-\exp\Big(\ldots-\exp\left(I_0\right)\ldots\Big)\Big)=I_1, \tag{8.12}$$

$$I_{i+1}=S\left(I_i\right),\ \lim_{i\to\infty}\exp\Big(-\exp\Big(\ldots-\exp\Big(\ln\left(I_0+1\right)-I_0\Big)\ldots\Big)\Big)=I_2, \tag{8.13}$$

мұндағы жақшалар саны $i+1$ тең.

$I_1=0.567$ және $I_2=0.806$ сандарының физикалық түсінігін түрліше түсіндіріп қарастырады. Ықтималдылықтың тығыздығы локалды (бір сәттік) сипаттама болып табылады. Сондықтан да ол түрлі айнымалының мәнінде түрліше болуы мүмкін және I_1 санын өзаффиндік критерийі ретінде қабылдауға болады. (8.10) теңдеуі бойынша $I<1$ үшін ең нашар жақындауды $1-I_{10}=I_{10}, I_{10}=0.5$ көруге болады. Энтропия орташаландырылған сипаттама болып табылады, сондықтан да I_2 саны өз-өзіне ұқсастық критерийі болуы мүмкін. Басқаша қарастыратын болсақ, I_1, I_2 сандары Фиббоначчи $I_{20}=0.618$ (динамикалық өлшемнің "алтын кесуі") сандарының ұқсасы ретінде қарастырылуы мүмкін, сәйкесінше бұл статистикалық өзаффинді және өз-өзіне ұқсас жүйелер үшін қарастырылады. Расында да (8.9) формулада $I \lesssim 1$ кезінде

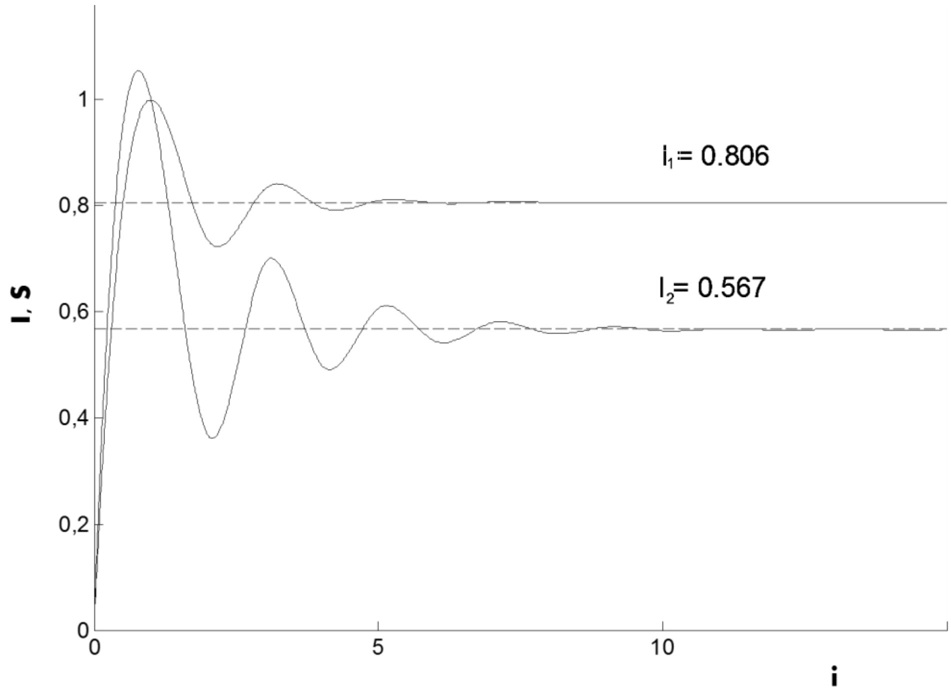
$$\left(1+I\right)\left(1-I\right)=I,\ I^2+I-1=0, I=I_{20}=0.618, \tag{8.14}$$

сондай-ақ (8.11) теңдеуі бойынша $I<<1$ кезде $e^{-I}=I,\ I=I_1=0.618$ аламыз. Өзаффиндік, динамикалық тепе-теңдік, өз-өздігіне ұқсастық секілді жүйенің заңдылықтары (8.11) формуласымен сипатталады.

Жоғарыда айтылған жағдайлар бойынша ақпарат пен ақпараттық энтропияның эволюциялық графигін тұрғызайық. Бұл үшін листингі төменде көрсетілген программа қолданылуы мүмкін.

Визуалды қисық ақпарат және ақпараттық энтропияға арналған бағдарлама листингі.

```
clear; clc; %  Workspace және Command Window тазалау
N=15; %  итерация саны

%  I1 мағынасының есебі.
I1(1) = 0;
for i = 1:N;
    I1(i+1) = exp(-I1(i)); % ағымдағы мәні I1.
end;

x=0:0.01:N; % координата нүктесі (абсцисс осьі бойынша)
% интерполяция мағынасы  I1
i = 0:N;
I1= spline(i,I1,x); % Сплайн-интерполяция мағынасы I1
plot(x,I1) % Визуализация мәні I1.
hold on;

% есептеу мәні I2.
I2(1) = 0;
for i = 1:N
    I2(i+1) = (I2(i)+1)*exp(-I2(i)); % ағымдағы мәні I2.
end;

x=0:0.01:N; % Координата нүктесі  (абсцисс осі бойынша)
% интерполяция мәні I2
i = 0:N;
I2= spline(i,I2,x); % Сплайн-интерполяция мәні I2
plot(x,I2) % Визуализация мәні I2.

hold on;
%  горизонтальді асимптотикалық түзу орнату прямой
% ордината I1.
L=length(x);

X=[0 N];
Y=[I1(L) I1(L)];
plot(X,Y,'r--');

hold on;
% горизонтальді асимптотикалық түзу орнату
% ординатамен I2.
X=[0 N];Y=[I2(L) I2(L)]; plot(X,Y,'r--');
```



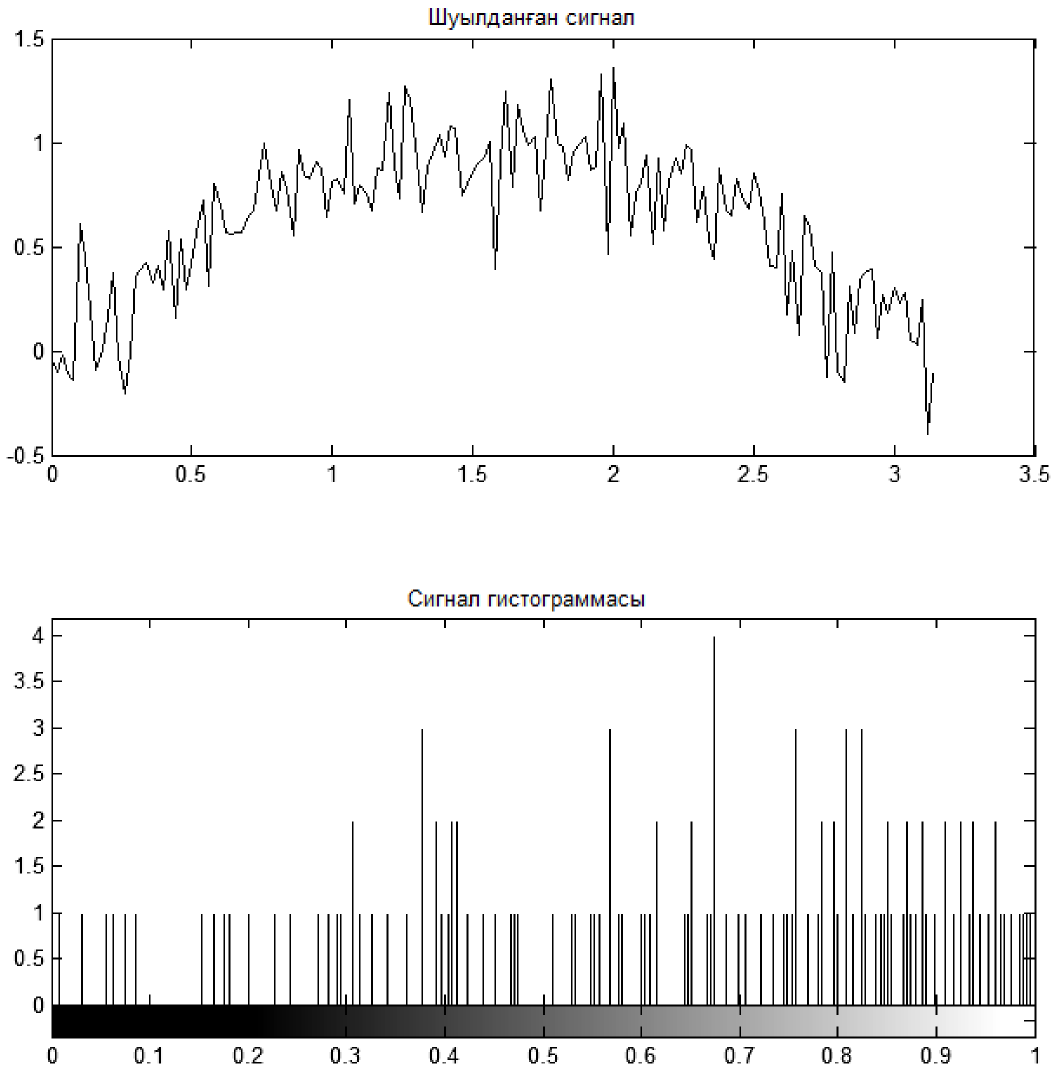
8.1 Сурет. *Ақпарат және ақпараттық энтропияның эволюциясы*

Мультифракталдардың спектрлік теориясынан көрініп тұрғанындай, өз-өзіне ұқсас жинақтың энтропиясы кезектесусіз ұяшықардың фракталдық өлшемділіктеріне тең болады. Сондықтан да I_1 және I_2 сандарын сәйкесінше қарапайым және күрделі түрдегі нүктелер жиынының өзаффиндік және өз-өзіне ұқсастық ұяшығы ретінде қарастыруға болады. Тәжірибеден $I_{20} < S \leq I_2$ кезінде жергілікті құрылымдардың өзаффиндігін күтуге болады. Ал жүйенің өздігінен құрылуын (статистикалық өз-өзіне ұқсастық) $I_{10} < S \leq I_1$ кезінде байқауға болады. Бұған $\gamma_0 = I_{10}$, I_1, I_{20}, I_2 мәндері сәйкес келеді.

Matlab ортасында энтропияны есептеу үшін скаляр мәні I элементінің интенсивтілігін көрсететін E параметрін қайтарып отыратын Image Processing Toolbox пакетінің `E=entropy(I)` функциясы қолданылады. Энтропия қарастырып отырылған көріністің сипаттамасы ретінде қарастыруға болатын белгісіздіктің статистикалық өлшемін көрсетеді. Ақпараттық энтропияның мәні (8.4) формуласымен анықталады. Мұнда P_i мәні `imhist` функциясы арқылы қайтып отыратын гистограмманың есептеулеріне сәйкес келеді. I массиві нақты сандармен көрсетілуі керек. Төменде ұзақтығы жарты периодқа тең болатын шуыл қосылған гармоникалық сигналдың құрылымына қатысты мысал және бұл сигнал үшін гистограмманың визуалдану есебі және энтропияның есептелу мәні көрсетілген. Осы программаға сәйкес есептеулердің нәтижесі 8.2 суретте көрсетілген.

Гистограмманы есептеуге және ақпараттық энтропия есебін, гармоникалық шуылданған сигналдың визуалзациясына арналған бағдарлама листингі.

```
clear; clc;
x=(0:0.02:pi)'; % сигналды анықтау аймағы
y=sin(x);yn=y+0.2*randn(size(x)); % координата есебі
% шуылды гармоникалық сигнал тұрғызу
figure  % графикалық терезені құрастыру
subplot(2,1,1); plot(x,yn)
subplot(2,1,2);
imhist (yn) % шуылданған сигнал гистограммасын тұрғызу
S=entropy(yn) % энтропия есебі
```



8.2 Сурет. Сигналдың кері санағы кері санағы бойынша алынған гистограмма және шуылданған сигнал

MatLab ортасында сонымен қатар ақпараттық энтропия санағының бейнесі де алынады. Ол үшін MatLab ортасында өлшемі міндеті түрде tif, jpg және png болатын түсті емес немесе түрлі-түсті суретті жүктеп алу қажет.

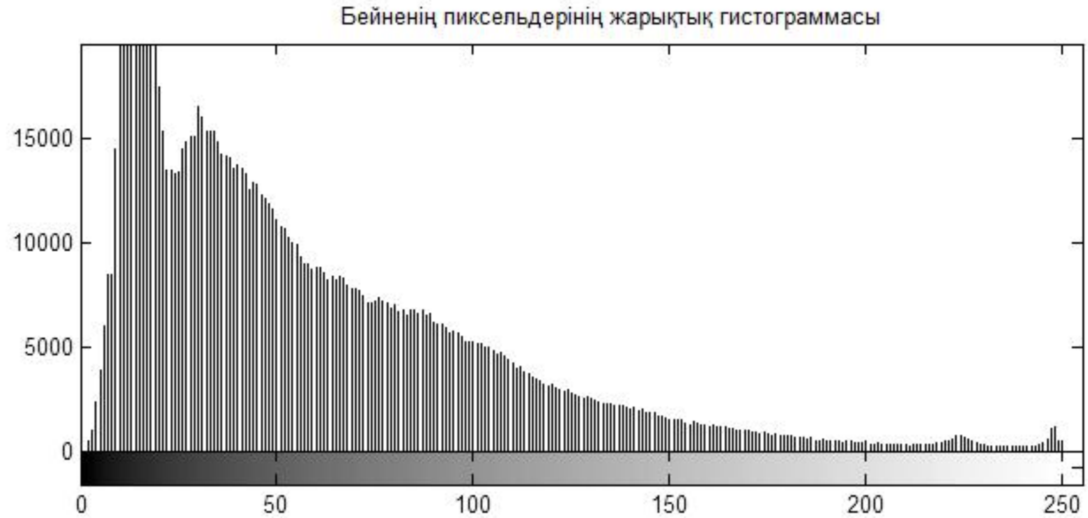
Ақпараттық-энтропиялық өлшем алгоритмі жоғарғы сипаттамаларға ұқсас. Ең алдымен түсті емес суретті қалай өңдеу қажет екенін қарасытрамыз.

MatLab ортасында өңделетін бейнелерді компьютердің кез-келген папкасынан алуға болады. Сурет өңдеуді бастау бастау үшін программада суреттің орналасқан жерін көрсету қажет. Мысал ретінде жұмыс үстелінде тұрған суретті MatLab ортасына жүктеуді қарастырайық. Суретті жүктеу `imread` командасының көмегімен жүзеге асады. Содан кейін жақша мен тырнақшаға файлдың орналасқан жерін көрсету қажет. Қарастырылған мысалда сурет файлы roza.png жұмыс үстеліне орналасады. Жүктелген суретті визуализациялау үшін `imshow` функциясы қолданылады. Төменде көрсетілген программа листингі арқылы MatLab-тағы суреттерді орналастыру, мысалы жұмыс үстеліне гистограмма орналастыра аламыз, ал сол суреттің жарықтандыру пикселін және ақпараттық энтропияның сандық мағынасын сипаттайды.

Программаға қатысты есептеулер 8.3 суретте көрсетілген.

```
clear; clc;

% түсті емес бейнені ашамыз
I = imread('C:\Users\qwe\Desktop\roza.png');

figure; % графикалық терезе құрастыру
subplot(2,1,1);
imshow(I) % бейнені визуализациялау
title ('шығыс бейне')
subplot(2,1,2);

imhist(I) % жарықтандыру пикселінің гистограммасын құрастыру
% сурет
title ('Бейненің пиксельдерінің жарықтық гистограммасы')
J = entropy(I) % энтропия есебі
```

Бұл жағдайда ақпараттық энтропия 7.1592 тең. Сонымен қатар осы алгоритм негізінде MatLab ортасында түрлі-түсті бейнелерді өңдеу мүмкіндігі бар. Төменде көрсетілген программа листингі гистограмманы интенсивтілігі бойынша оның пикселінің және ақпараттық энтропия есебінің түрлі-түсті бейнелерін визуализациялау үшін арналған. Программаға сәйкес, ең алдымен `imread` функциясының көмегімен түрлі-түсті суретті ашып оны `imshow` функциясы арқылы визуализациялау қажет. Одан кейін шығыс суреттің интенсивтілігі мәнін визуализациялап, қарастырып және алынған мәндер бойынша жарықтадыруды бөліп, гистограмма құрастыру қажет. Бұл жағдайда бейне түрлі-түстіден түсті емеске өзгереді. Бұл бейнені де визуализациялау үшін де `imshow` командасы қолданылады. Гистограмманы қарастыру кезінде жарықтандыруға түсті емес суреттен алынған сұр түсті реңктері қолданылады. Гистограмманы құрастыру кезінде бейнені жарықтандыруға imhist функциясын қолданамыз. Ақпараттық энтропияны есептеу үшін entropy командасы қолданылады.



8.3 Сурет. Түсті емес суретті визуализациялау және жарықтандыру пикселінің гистограммасы

Бұл программа листінгі суреттің пиксел интенсивтілігіннің гистограммасын құрастыру үшін, түрлі-түсті суретті визуализациялау үшін және энтропия есебі үшін арналған.

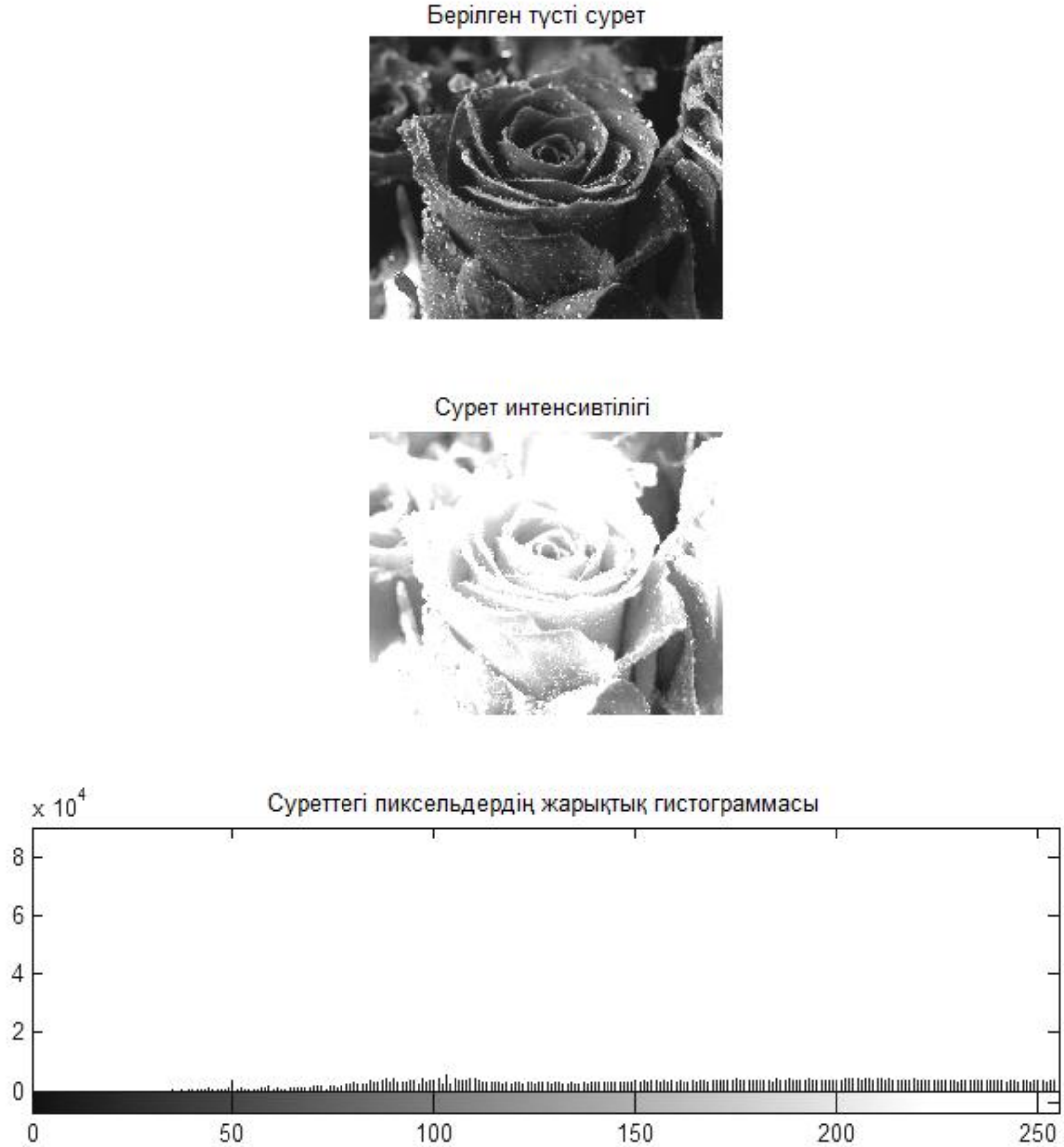
```
clear;
clc;

% Түсті бейнені ашамыз
L = imread('C:\Users\qwe\Desktop\roza_color.jpg');
figure; % графикалық терезенің құрылуы
subplot(311);
imshow (L) % түсті бейненің визуализациясы
title('Берілген түсті бейне')

% Берілген бейненің мәндерін визуализациялаймыз және
% құрастырамыз, гистограмма тұрғызамыз
I=L(:,:,1)+L(:,:,2)+L(:,:,3);
subplot(312)
imshow(I); % берілген бейненің мәндер интенсивтілігн
% визуализациялаймыз
title('Бейненің интенсивтілігі');

subplot(313);
imhist(I); % берілген бейненің мәндерінің
% интенсивтілігі бойынша гистограмма тұрғызу
title ('Бейненің пиксельдерінің жарықтық гистограммасы')

j=entropy(I) % энтропия есептеуі
```

Ақпараттық энтропияның сандық мәнін алу үшін `Command Window` енгізіледі.

Image Processing Toolbox осындай түрде бейнелер өңделеді. Бұл жағдайда бейнені ашу келесі түрде:

```
I = imread('circuit.tif');
```

`circuit.tif` файлынан басқа, пакет Image Processing Toolbox басқа да тесттік суреттерді қамтиды, мысалы, `AT3_1m4.tif`, `autumn.tif`, `bag.tif`, `cell.tif`, `circles.png`, `concordaerial.png`, `greens.jpg`, `testpat1.jpg` және т.б.



8.4 Сурет. Түрлі-түсті бейнені визуализациялау және интенсивті пикселдердің гистограммасы.

Тапсырма

1. Уақыт бойынша эволюциялық ақпарат пен ақпараттық энтропияның графиктерін алу.
2. Ара тәрізді шуылсыз сигнал, ара тәрізді шуылды сигнал, шуылды және шуылсыз қайталағыш тікбұрышты импульстердің сандық мәнін есептеңіз.
3. Түрлі түсті және түсті емес бейнелердің интенсивтілік пикселінің гистограммасын құрастырыңыз. Бейнелерге сәйкес келетін ақпараттық энтропияның өлшемін есептеңіз.

Бақылау сұрақтары

1. Ықтималдық теориясында «ақпарат» терминінің астында қандай мағына жатыр? Ақпарат қандай формула арқылы есептеледі?
2. Аддитивтілік мағынасының мәні неде?

3. Өлшем мағынасын түсіндіріңіз? Өлшем арқылы түсіндірілетін физикалық мағыналарға мысал келтіріңіз.
4. Қандай жағдай ақпараттық болып саналады – аз немесе көп ықтималдықтан?
5. Термодинамикалық процесстер сипаттамасынан энтропияның қандай физикалық мағынасын көре аламыз?
6. Энтропия аддитивті өлшем бола алады ма?
7. «Статикалық салмақ» мағынасын ашып айтыңыз.
8. Ақпараттық энтропия қандай формула бойынша есептеледі?
9. Ақпараттық энтропия нені сипаттайды?
10. Өзін-өзі ұйымдастырудың күрделі жүйесі нені сипатайды?
11. Ақпараттық энтропия бейнесінің алгоритмдік есептеуін сипаттаңыз.
12. Шығыс бейненің интенсивтілік мағынасын MatLab ортасында қалай қалыптастыруға және бейнелеуге болады?
13. Келесі фунциялар қандай мақсатпен және қандай жолдармен қолданылады: *imread, imshow, imhist* және *entropy*?
14. Бір объектінің түрлі түсті және түсті емес бейнелерінің энтропиялық сандық есептеулерін тексеріңіз? (нолдік жарықтандыру кезінде пикселдер санын салыстырыңыз)

Қолданылған әдібиеттер тізімі

1. Жанабаев З.Ж., Иманбаева А.К., Алмасбеков Н.Е. Компьютерное моделирование в радиофизике и электронике. – Алматы: Қазақ университеті, 2005. – 143 с.
2. Zhanabaev Z. Zh. The informational properties of self-organizing systems // Rep. Nat. Acad. of Science RK. – 1996. – № 5. – P. 14-19.
3. Жанабаев З.Ж., Гревцева Т.Ю. Фрактальные структуры и оптические явления в наноструктурированных полупроводниках. – Алматы, Қазақ университеті, 2014. – 162 с.
4. Гонсалес Р., Вудс Р., Эддинс С. Цифровая обработка изображений в среде Matlab. – М.: Техносфера, 2009. – 618 с.
5. Дьяконов В.П. Matlab и Simulink для радиоинженеров. – М.: ДМК Пресс, 2011. – 976 с.
6. Сато Ю. Без паники! Цифровая обработка сигналов. – М. Додека-XXI, 2010. – 178 с.

Зертханалық жұмыс №9

СИГНАЛДАРДЫҢ ФИЛЬТРАЦИЯСЫ

Жұмыстың мақсаты: Matlab ортасын қолдану арқылы фильтрлердің сипаттамаларын тұрғызу және сигналдарды фильтрлеу әдістерімен танысу.

Қысқаша теориялық кіріспе

Фильтрлер және олардың жіктелуі

Сигналдарды өңдеудің ең кең таралған құралы - олардың *фильтрациясы*. Әдетте фильтрлеудің мақсаты, сигнал қоспасынан басқа сигналдармен және шуылдармен пайдалы сигналды шығару болып табылады. Математикалық тұрғыдан фильтрлеу - сигналдардың спектрін үйкеліс кезінде көбейту әсеріне айтылады. (Үйкеліс операциясын бір функцияның екінші бір шағылыстырылған және жылжыған көшірмесі «ұқсастығы» ретінде түсіндіруге болады. Сондай-ақ, үйкелісті бір функцияның салмағы ретінде сипаттауға болады, егер басқа функция шағылысатын және жылжитын болса). Спектрлер комплекс сандар сияқты көбейтілген кезде бастапқы сигналдың гармоника амплитудасы және үйкеліс ядросы көбейтіледі және фазалар бірге қосылады. Осылайша, біз сигнал спектрін өзгертүге мүмкіндік аламыз. Бұл өте пайдалы операция. Мысалы, дыбыстық жазбада сигнал спектрін өзгерту жазбаны шудан тазартуға, түрлі жазу құрылғылары сигнал бұрмалануын өтеуге және құралдардың үнін өзгертуге мүмкіндік береді. Суреттерді өңдеу кезінде, фильтрлеу кескінге әртүрлі әсерлерді қолдануға мүмкіндік береді: бұлыңғырлық, шекараны сызу, өрнектеу және т.б. Басқа салаларда фильтрлеу көбінесе қабаттасып тұрған сигналдарды ажыратуға және де сигналды шудан тазартуға қызмет етеді. Сондай-ақ, фильтрлеу көптеген басқа күрделі процестердің ажыралмас бөлігі болып табылады. Фильтрациядағы үйкелістің ядросы әдетте *сүзгі* деп аталады. Фильтрлеу операциясын орындайтын бүкіл құрылғыларды жиі филтр деп атайды. Фильтрдің ұзындығы (өлшемі) – бұл үйкеліс ядросының ұзындығы. Жалпы алғанда, фильтр сигнал спектрінде және гармоника амплитудасы мен олардың фазаларында өзгертеді. Дегенмен, фильтрлерді сигнал фазасын өзгертпейтіндей етіп жобалауға мүмкіндік бар. Мұндай фильтрлер сызықтық фазасы бар фильтрлер деп аталады. Бұл дегеніміз, сигнал фазасын өзгерткенде, сигналдың барлық гармоникасы уақыт бойынша бірдей шамаға жылжиды. Осылайша, сызықтық фазасы бар фильтрлер сигнал фазасын бұрмаламайды, тек бүкіл сигналды уақыт бойынша жылжытады. Мұндай фильтрдің үйкеліс ядросы өзінің орталық нүктесіне салыстырмалы түрде қатаң симметриялы (бірақ ядросы антисимметриялық болып табылатын, сызықты фазасы бар фильтр түрлері де болады). Кез-келген фильтр жиілік және фазалық сипаттамаларға ие. Олар өңделетін сигналдың әртүрлі гармоникасының амплитудасы мен фазасына фильтрдің қандай әсері бар екенін көрсетеді. Егер фильтр сызықтық фазаға ие болса, онда тек қана фильтрдің жиіліктік сипаттамасы қарастырылады. Әдетте жиілік сипаттамасы, амплитуданың жиілікке тәуелділігіне байланысты график түрінде бейнеленеді (децибелмен). Signal Processing Toolbox (сигналды өңдеу құралдарының) пакетінде белгілі бір жиілік ауқымындағы сигналдар жиілігінің компоненттерін бөлетін, фильтр тізбектерін жобалау үшін кең құралдар бар.

Фильтрлерді жобалау кезінде олардың электрлік параметрлерін анықтау қарастырылған, бұл осы түрдің амплитуда-жиіліктік және фаза-жиіліктік сипаттамаларын және берілген параметрлерді алуға мүмкіндік береді.

Амплитуда-жиіліктік сипаттамасы, фильтрдің жіберу коэффициентінің сол фильтрдің кірісіне берілген сигналдың тұрақты амплитудасы мен жиілік диапазонына тәуелділігін түсіндіреді.

Фаза-жиіліктік сипаттамасы радиандар немесе градустарда анықталатын кіріс және шығыс сигналдары арасындағы фазалық жылжу жиілігінің функциясы болып табылады. Фильтрлер көбінесе фазалық өзгерістердің кең спектріне ие емес, сол себепті фазалық ауысу π модулінен немесе 180° асатын нүктелерде фаза-жиіліктік сипаттамаларының үзілістеріне әкеліп соғады.

Құрастырылған фильтрлер екі топқа бөлінеді - аналогтық (үздіксіз) және сандық. Жиілікке жіберу коэффициенті модулінің тәуелділік формасы бойынша фильтрлер келесі түрлерге бөлінеді:

- lowpass – төмен жиілікті, белгілі бір шекаралық жиіліктерге дейін сигналдарды өткізу;
- highpass – жоғары жиілікті, белгілі бір шекаралық жиіліктен кейін сигнал өткізу;
- bandpass – жолақты, белгілі бір жиілік диапазонында сигналдарды өткізу;
- stoppass – белгілі бір жиілік диапазонында әлсірейтін сигналдар.

Импульстік сипаттамасы түріне қарай (бірлік ауданымен шексіз жұқа бір импульске фильтрлік жауап), сандық фильтрлер екі негізгі түрге бөлінеді:

- шексіз импульстiқ сипаттамасы бар фильтрлер;
- соңғы импульстік сипаттамасы бар фильтрлер.

Шексіз импульстiқ сипаттамасы бар фильтрлерді рекурсивті фильтрлер деп те атайды. Осындай фильтрлер үшін шығыс сигналының келесі саны шығыс сигналының алдыңғы үлгілерімен есептеледі, яғни фильтрлер кері байланысты пайдаланады.

Сондай-ақ, фильтрдің кіріс сигналы бірлік қадам түрінде болса, шығыс сигналы уақытқа тәуелді өтпелі сипаттамалармен сипатталады.

Signal Processing пакеті фильтрлерді жобалаудың белгілі және дәлелденген әдістерінің көпшілігін (классикалық Butterworth, Bessel, Chebyshev 1 және 2 сұрыптау фильтрлері, эллиптикалық фильтрлер және т.б.) жүзеге асырады. Бұл пакетте фильтрлерді тікелей құрастыру құралдарымен бірге прототипті фильтрлер қолданылады, бір фильтрді басқа түрлерге түрлендіруге арналған алгоритмдер жүзеге асырылады, рекурсивті алгоритмдер және т.б.

Төменде фильтрлермен жұмыс істеуге арналған Signal Processing Toolbox пакетінің функцияларын қолданудың кейбір мысалдары қарастырылады.

freqs функциясының көмегімен амплитуда-жиіліктік және фаза-жиіліктік сипаттамаларын құрастыру.

Фильтрлер тасымалдау сипаттамалары арқылы жазылады

$$H(s)=\frac{b(1)s^{nb}+b(2)s^{nb-1}+...+b(nb+1)}{a(1)s^{na}+a(2)s^{na-1}+...+a(na+1)}. \qquad (9.1)$$

Бұл жерде na және nb – полиномның бөлімі және алымының дәрежесі, s – комплекс айнымалы. $H(s)$ табу үшін сызықтық жүйелер мен тізбектерді талдаудың операторлық әдісі қолданылады. Комплексті амплитуда-жиіліктік сипаттамасы $H(jw)$ сияқты бөлімі және алым $H(s)$ -ты есептеп шығару, яғни $s=jw$ (және олардың бөліндiлері) болған кезде анықталады.

`freqs` функциясы төмендегідей нұсқадағы бірқатар амплитуда-жиіліктік сипаттамаларын есептеуді жүзеге асырады. Бұл функцияның дұрыс жазылуы мен қолданылуы келесідей болуы мүмкін. `h=freqs(b,a,w)` функциясы `a` және `b` векторларында берілгенде, фильтр сипаттамасының тасымалдау коэффициенті $H(s)$ аналогты фильтрдің амплитуда-жиіліктік сипаттамасының w жиілік векторына сәйкес келетін `h` векторын анықтайды. `[h,w]=freqs(b,a,w)` функциясы көмегімен амплитуда-жиіліктік сипаттамасының `h` векторы және `w` жиілігі анықталады, сонымен қатар жиілік диапазоны және оның көрінісі өздігінен белгілі болады. `[h,w]=freqs(b,a[,n])` функциясы амплитуда-жиіліктік сипаттамасының n нүктелері үшін амплитуда-жиіліктік сипаттамасының `h` векторын және `w` жиілігін есептеу үшін қызмет етеді. Егер n берілмесе, келісім бойынша 200 таңдалады. `freqs(b,a)` функциясын пайдаланып амплитуда-жиіліктік сипаттамасы есептеледі және амплитуда-жиіліктік және фаза-жиіліктік сипаттамаларының графиктері тұрғызылады. Амплитуда-жиіліктік сипаттамасын есептеу алгоритмі Лаплас түрлендірулеріне негізделген.

Жіберу сипаттамасына ие екінші ретті сызықтық жүйенің амплитуда-жиіліктік және фаза-жиіліктік сипаттамаларын тұрғызуға және есептеуге арналған алгоритмді қарастырайық

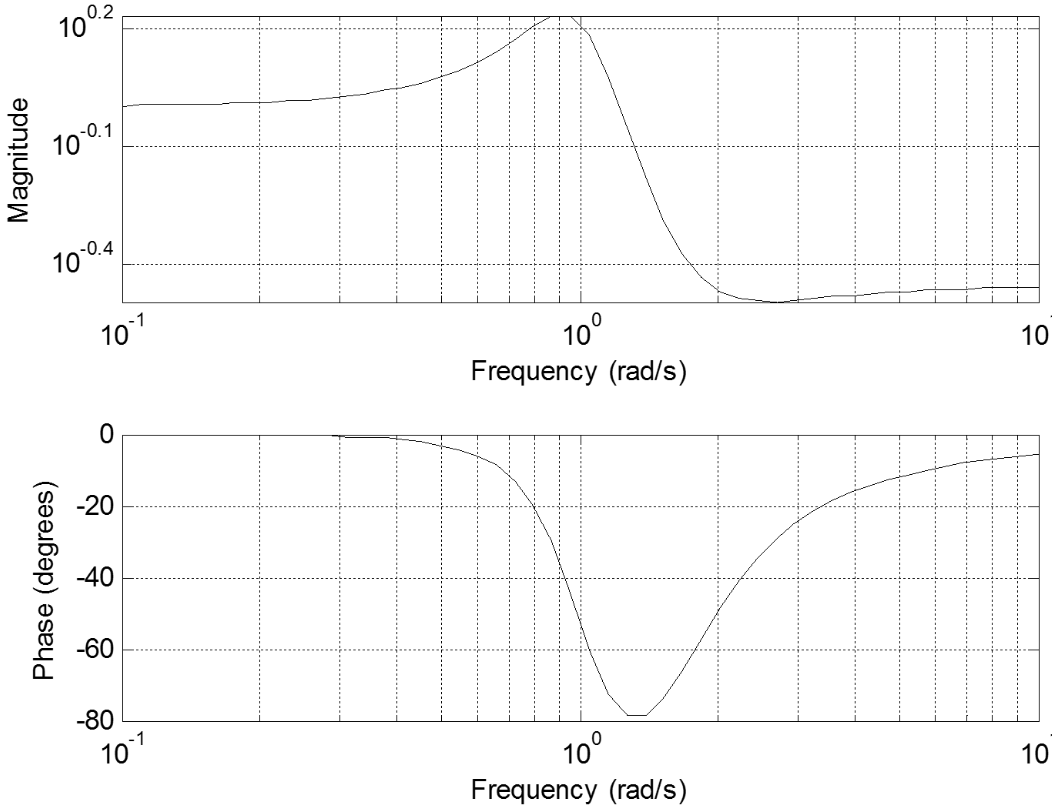
$$H(s)=\frac{0.35s^2+0.5s+1}{s^2+0.5s+1}. \tag{9.2}$$

Бұл үшін келесі бағдарламаны пайдалануға болады.

Белгілі жіберу сипаттамасына ие екінші ретті сызықтық жүйенің амплитуда-жиіліктік және фаза-жиіліктік сипаттамаларын тұрғызуға және есептеуге арналған листинг бағдарламасы

```
clear;
clc;

% H(s) полином бөлімі коэффициентінің векторы
a=[1 0.5 1];

% H(s) полином алымы коэффициентінің векторы
b=[0.35 0.5 1];

% Логарифмдік масштабтағы жиілік векторы
w=logspace(-1,1);

% Амплитуда-жиіліктік және фаза-жиіліктік сипаттамаларының
% графигін тұрғызу

freqs(b,a,w)
```

Нәтижесінде экранда төмендегідей графиктер шығады:



9.1 Сурет. Сызықтық жүйенің амплитуда-жиіліктік сипаттамасы(жоғарыда) және фаза-жиіліктік сипаттамасы (төменде)

grpdelay функциясын пайдаланып, топтық кешігу (кідірілу) уақытын есептеу.

Топтық кешігу уақыты $r(w) = -d\vartheta(w)/dw$ деп анықталады және жиілік функциясы ретінде орташа сигналдың кідірілуінің өлшемі ретінде қызмет етеді. Мұнда w – айналмалы жиілік және θ – фазалық бұрыш. Топтық кешігу уақытын анықтау өте маңызды міндет болып табылады, өйткені сигнал кешігу уақытының айтарлықтай өзгеруі оның импульсінің, спектрдің, фазалық қателердің пайда болуына және т.б. өзгеруіне әкелуі мүмкін. Топтық кешігу уақыты фаза-жиіліктік сипаттамасын түсіндіреді, атап айтқанда, фазаның сызықтығы және өткізу жолағындағы фазалық ауысу сияқты параметрлер.

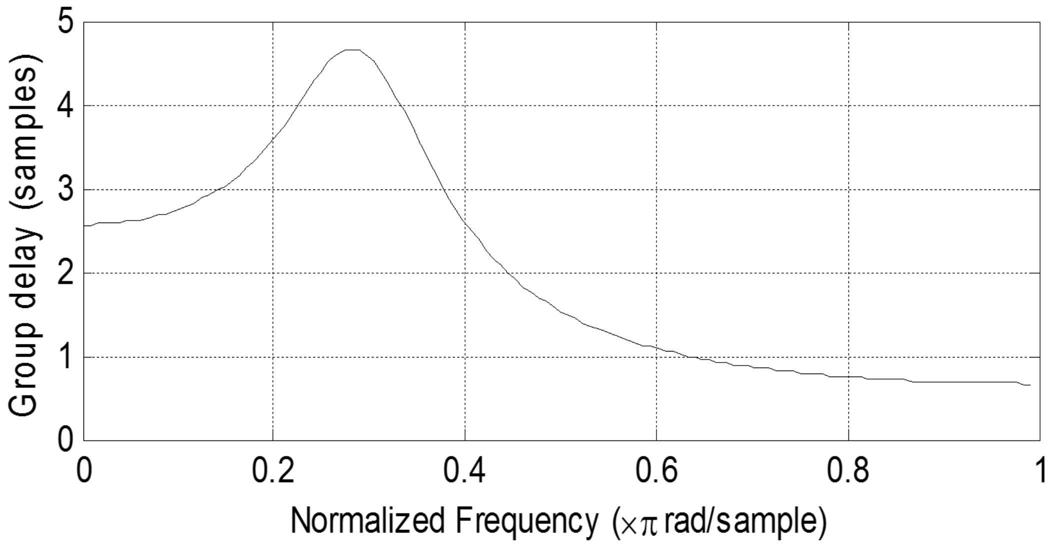
Мысал ретінде амплитудалы-жиіліктік сипаттамасы өткізу жолағындағы жиіліктерде максималды тегіс болатын, топтық уақыттың кешігу есептеулерінің Баттерворд фильтрін қарастыруға болады. Баттерворд фильтрі үшін амплитудалы-жиіліктік сипаттамасының жуықтау функциясы H=$1/\sqrt{1+f^n}$ қатынасымен анықталады. Мұндағы, f – нормаланған жиілік, n – фильтр қатары. Фильтр кесу жиілігімен, электронды схеманың шығыс сигналының қуаты өткізу жолағындағы қуаттың жартысына азаятындай яғни жоғары немесе төмен жиілігімен сипатталады.

Кесу жиілігі 0,3 болатын 4-реттік Баттерворт фильтрінің нормаланған жиілікпен топтық кешігу уақытының есептеулерін мысал ретінде қарастырайық. Ол үшін төмендегі программананы қолдануға болады.

Кесу жиілігі 0,3 болатын 4-реттік Баттерворт фильтрінің нормаланған жиілікпен топтық кешігу уақытын есептеулерінің программалық листингі.

```
clear; clc;
[b,a]=butter(4,.3); % Баттерворт фильтрінің сипаттамасы
grpdelay(b,a,128) % топтық кешігу уақытын есептеу
```

Осы бағдарламаның нәтижесі 9.2-суретте көрсетілген.



9.2 Сурет. Баттерворд фильтрі үшін топтық кешіктіру уақытының жиіліктен тәуелділігі

`impz` *функциясының көмегімен сандық фильтрдің импульстік сипаттамасының есептелуі.*

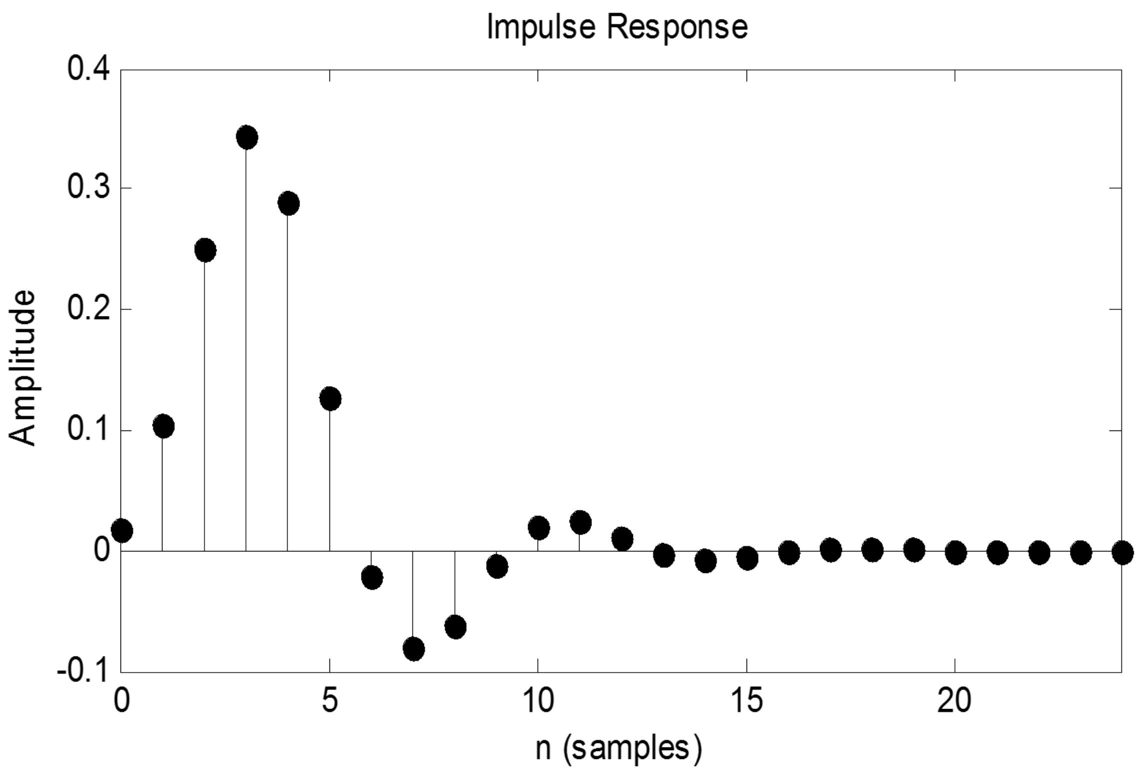
Импульсте шексіз көп амплитудамен шексіз аз ұзақтылық және бірлік ауданы бар жүйенің реакциясы деп жүйенің *импульстік сипаттамасын* айтамыз. Сандық фильтр үшін ол оның алымының және бөлімінің полиномдарының *b* және *a* коэффициенттерінің векторларымен көрсетілген тасымалдау сипаттамасында орналасқан. Бұл операцияны `[h,t]=impz(b,a[,n,fs])` функциясы орындайды. Ол импульстік сипаттамасы `h` және уақыт `t` векторларын қайтарады. Қосымша санау ретін `n` және дискреттеу жиілігі `fs` мәндерін беруге болады. `n`-ді бергенде уақыт векторы `t=[0:n-1]′` өрнегімен анықталады. Шығыс параметрлерінсіз функция импульс сипаттамасының графигін тұрғызады.

Кесу жиілігі 0,3 болатын 4-реттік Баттерворт фильтрінің импульстік сипаттамасының алғашқы 25 мәнін тұрғызайық. Ол үшін листинг программасы төмендегідей болатын программа құру керек.

Кесу жиілігі 0,3 болатын 4-реттік Баттерворт фильтрінің импульстік сипаттамасының алғашқы 25 мәнін тұрғызу үшін құрылған Листинг программасы.

```
clear;
clc;
[b,a]=butter(4,.3);
impz(b,a,25)
```

Нәтижесінде экранға келесі график шығады:



9.3 Сурет. Фильтрдің импульстік сипаттамасы `zplane` *функциясының көмегімен амплитудалы-жиіліктік сипаттамасының нөльдері мен полюстерін тұрғызу.*

Комплексті амплитудалы-жиіліктік сипаттамасы көбінесе өзінің нөльдері мен полюстары арқылы сипатталады. *Нөльдер* – тасымалдау сипаттамасы алымының түбірі, ал *полюстар* – бөлімінің түбірі. Нөльдері мен полюстарын комплексті жазықтықта бейнеленеді. Ол үшін мынадай функциялар қолданылады:

```
zplane(z,p)

zplane(b,a)

[hz,hp,ht]=zplane(z,p)
```

Жазудың бірінші формасында ол `'o'` нөл белгілерін және олар туралы деректерді және `z` және `p` векторларын пайдалану арқылы `'x'` полюстерінің белгілерін құрастырады. Шкаланы өзгерту үшін келесі пәрмендерді пайдалануға болады:

```
axis([xmin xmax ymin ymax])

set(gca,'ylim',[ymin ymax])

set(gca,'xlim',[xmin xmax])
```

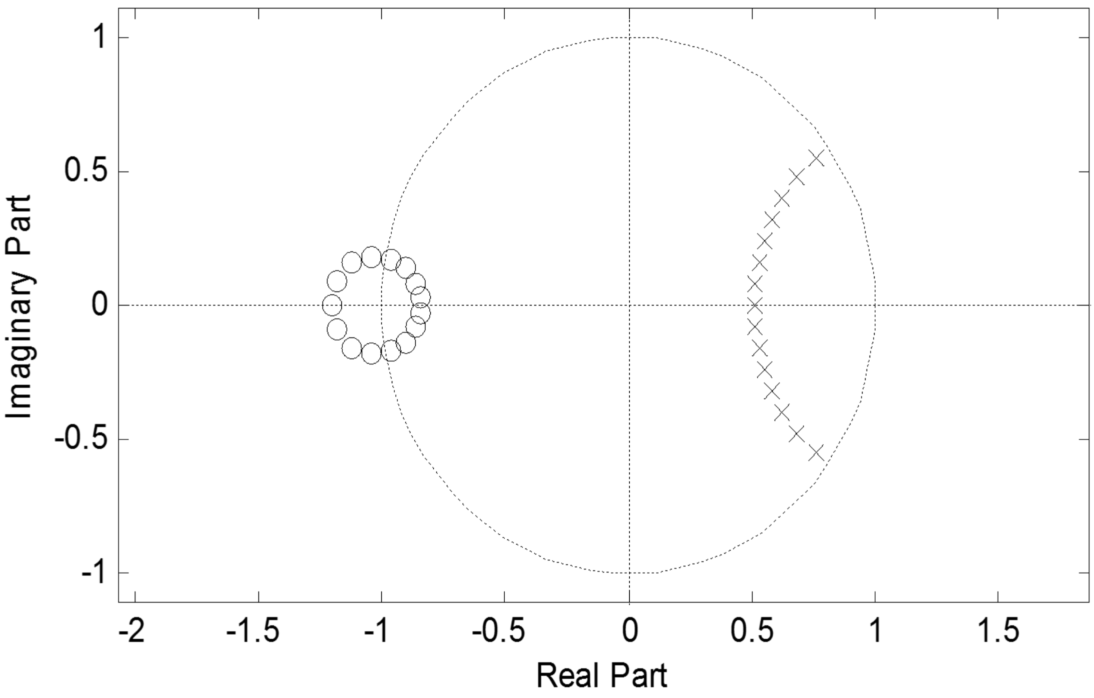
Екінші формада функция – полином алымы және бөлімі коэффициенттерін көрсететін, нөлдер мен полюстерді берілген `b` және `a` векторлармен есептеп береді.

Үшінші формада функция – нөлге(`hz`) дейінгі көрсеткіші бар векторларды, полюстерді(`hp`) және мәтәндік нысандарды(`ht`) жасайды. Нөлдер мен полюстерден басқа, нүктелі сызықтар координат осін және бірлік радиусының шеңберін құрайды.

Мысал ретінде біз 0,2 кесу жиілігіне ие 15-реттік Баттерворт фильтрінің нөлдері мен полюстерін тұрғызамыз.

0,2 кесу жиілігіне ие 15-реттік Баттерворт фильтрінің нөлдері мен полюстерін тұрғызуға арналған Листинг бағдарламасы.

```
clear;
clc;
[b,a]=butter(15,.2);
zplane(b,a)
```



9.4 Сурет. Баттерворт фильтрінің нөлдері мен полюстері

Дискретті бірөлшемді фильтрлеу

filter функциясы формуланың соңғы-айырмашылық теңдеулері арқылы сипатталған дискреттік сүзгіні пайдаланып, бір өлшемді массив ретінде (x) берілген сигналды фильтрлейді

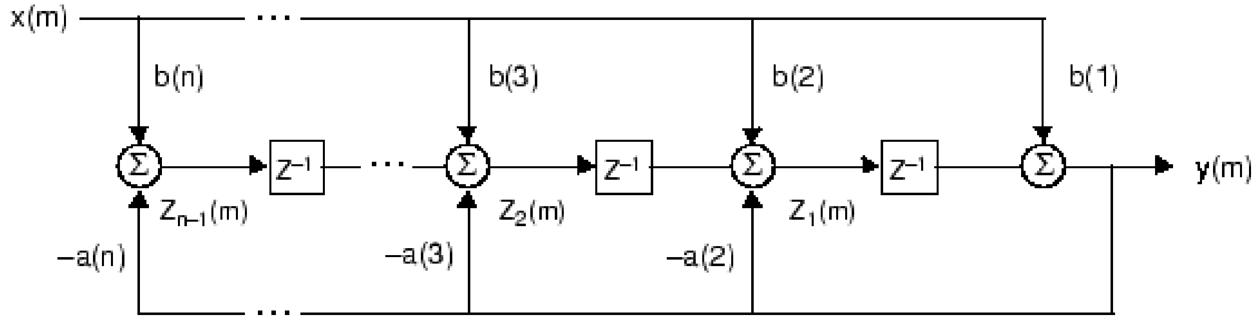
$$y(m) = \big(b(1)x(m) + z_1(m-1)\big)/a(1)$$

$$z_1(m) = b(2)x(m) + z_2(m-1) - a(2)y(m)$$

$$z_{n-2}(m) = b(n-1)x(m) + z_{n-1}(m-1) - a(n-1)y(m) \qquad (9.3)$$

$$z_{n-1}(m) = b(n)x(m) - a(n)y(m)$$

`filter` функциясы фильтрлеуді тура формадағы фильтр арқылы модельдеуге қызмет етеді, оның диаграммасы 9.5-суретте көрсетілген.



9.5 Сурет. Тура формадағы сандық
фильтрдің құрылымы

`Filter` функциясы төмендегідей формада жазылады:

```
y = filter(b,a,x)
```

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

```
[...] = filter(b,a,x[,zi,dim])
```

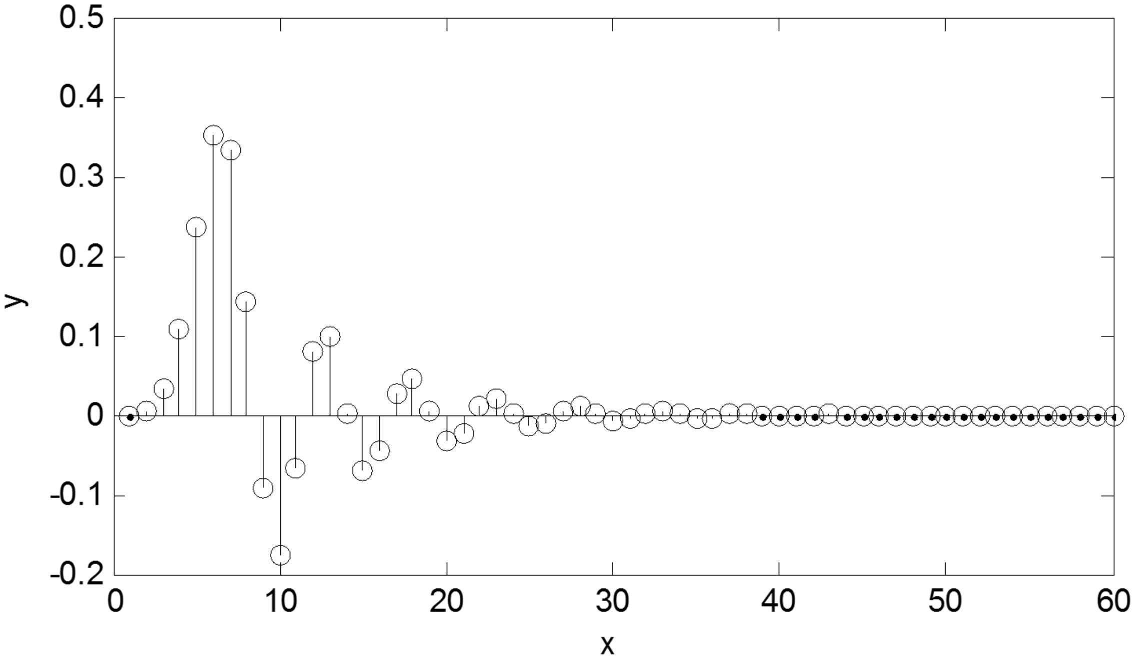
`X` параметрі вектор, матрица немесе көпөлшемді массив бола алады. Егер `x` матрица түрінде болса, онда фильтрлеу баған түрінде болады. Егер `x` көпөлшемді массив түрінде берілсе, фильтірлеу ең жақсы 1-өлшемнен басталады. Фильтрлеу нәтижесі у-ке орналастырылады. `Zf` параметрінде күй векторы енгізіледі. `Zi` параметрі бастапқы шарттарды анықтауға мүмкіндік береді, ал dim параметрі көпөлшемді массивтің өлшемін алу үшін қызмет етеді.

Төмендегі мысалда 10-реттік сандық фильтрдің импульстік сипаттамасының 60 нүктесін есептеу келтірілген.

10-реттік сандық фильтрдің импульстік сипаттамасының 60 нүктесін есептеуге арналған Листинг бағдарламасы.

```
clear;
clc;
x=[1 zeros(1,59)];
[b,a]=butter(10,.4);
y=filter(b,a,x);
stem(y)
```

Нәтижесінде келесі график алынады:



9.6 Сурет. filter функциясы көмегімен алынған сандық фильтрдің импульстік сипаттамасы

Бесселдің аналогтық фильтр прототипінің параметрлерін есептеу

Аналогтық фильтр прототипі әдетте H(s) тасымалдау функциясы түрінде берілген жалпыланған фильтр жүйесін құру үшін қолданылады. Бұл фильтрлер белгілі бір түрлендірулер көмегімен алдағы қарастырылатын нақты аналогтық және сандық фильтрдің құрылымын құруға мүмкіндік береді.

Бесселдің аналогтық фильтр прототипі – бұл төмендегідей тасымалдау функциясына ие төменгі жиілікті фильтр:

$$H(s)=\frac{z(s)}{p(s)}=\frac{k}{(s-p(1))(s-p(2))...(s-p(n))}. \qquad (9.4)$$

Бұл типтегі фильтрдің параметрін есептеуге төмендегідей функция қолданылады.

```
[z,p,k] = besselap(n)
```

Бұл функция берілген `n` реттік фильтрдің (берілген фильтрдің нөлдері жоқ) `z` полюсінің массивін генерациялайды. Бұл жерде `p`-вектор, `k`- жіберу коэффициенті. Фильтрдің `n`- реті 25-тен аспауы керек. `k` және `z` жоғарғы және төменгі жиілікті аймақтарда Бессел фильтірінің тасымалдау функциясы Баттерворт фильтірінің тасымалдау функциясының n-ретіне жақындайтындай етіп орналастырылады. Бессель фильтрінің полюсі ортасы нақты жарты осьтегі шеңберде орналасқан. Амплитуда-жиіліктік сипаттама мәні w_c кесу жиілігінде $1/\sqrt{2}$ аспайды.

Келесі мысалда 10-реттік Бесселдің аналогтық фильтр прототипінің параметрлері көрсетілген.

10-реттік Бесселдің аналогтық фильтр прототипінің параметрлерін анықтауға арналған Листинг бағдарламасы

```
clear; clc;
n=10;
[z,p,k]=besselap(n)
```

Есептеу нәтижелері Command Window-да шығады:

```
z =
     []
p =
  -0.9091 - 0.1140i
  -0.9091 + 0.1140i
  -0.8688 - 0.3430i
  -0.8688 + 0.3430i
  -0.7838 - 0.5759i
  -0.7838 + 0.5759i
  -0.6418 - 0.8176i
  -0.6418 + 0.8176i
  -0.4083 - 1.0813i
  -0.4083 + 1.0813i
k =
     1
```

Баттерворттың аналогтық фильтр прототипінің параметрлерін buttap *функциялары арқылы есептеу*

Баттерворт аналогтық фильтр прототипінің H(s), Бессел фильтрі үшін жазылған түрмен сәйкес. Баттерворт аналогтық фильтр прототипінің параметрлері келесі функциямен есептелінеді

```
[z,p,k]  =  buttap(n)
```

Баттерворт фильтрі өткізу аймағында ерекше көлбеу амплитуда-жиіліктік сипаттамаға және барлық жиіліктік диапазаонда біркелкі төмендеуге ие. Амплитуда-жиіліктік сипаттама $\left|H(w)\right|^2 = 1\Big/\left(1+\left(w/w_0\right)^{2n}\right)$.

Амплитуда-жиіліктік сипаттама мәні w_0 кесу жиілігінде $1/\sqrt{2}$ аспайды. Амплитуда-жиіліктік сипаттамасы модулінің квадратының $2n$- 1 алғашқы туындысы w_0 = 0 болғанда 0-ге тең. 10 реттегі Баттерворт фильтрінің параметрлерін есептеуге мысал төменде көрсетілген.

10-реттік Баттерворт фильтрінің параметрлерін анықтауға арналған программаның листингі

```
clear; clc;
n=10; [z,p,k]=buttap(n)
```

Осы программа бойынша есептеу нәтижелері командалық терезеге шығады:

```
z =
     []

p =
  -0.1564 + 0.9877i
  -0.1564 - 0.9877i
  -0.4540 + 0.8910i
  -0.4540 - 0.8910i
  -0.7071 + 0.7071i
  -0.7071 - 0.7071i
  -0.8910 + 0.4540i
  -0.8910 - 0.4540i
  -0.9877 + 0.1564i
  -0.9877 - 0.1564i

k =
    1.0000
```

Бессель аналогтық фильтрлерін - `besself` *функциялары арқылы жобалау*

Бессель аналогты фильтрлерін жобалау үшін келесі функциялар қызмет етеді

```
[b,a]  = besself(n,Wn)     [b,a] = besself (n,Wn, 'ftype')
```

Бұл функциялар b және a векторларын тасымалдау сипаттамасы түрінде есептейді

$$H(s) = \frac{b(1)s^n + b(2)s^{n-1} + ... + b(n+1)}{a(1)s^n + a(2)s^{n-1} + ... + a(n+1)}. \quad (9.5)$$

Бұл фильтрлер сигналдар түрін аз бұрмалайды, себебі өткізу жолағында әрдайым топтық кідіріс уақытына ие. Сандық фильтрлерде мұндай қасиет жоқ, сол себепті Бессель фильтрлері өзінше бірегей болып келеді.

Wn параметрі төменгі жиілік фильтрінің кесу жиілігін скаляр болған жағдайда белгілейді. Егер қос сыңарлы вектор түрінде Wn=[wl w2] болса, функция

```
besself (n,Wn)
```

2n реттік жолақты фильтрдің тасымалдау сипаттамасы коэффициенттерін өткізу жолағы wl<w<w2 қайтарады. 'ftype' параметрі жоқ болған жағдайда, төменгі жиілік фильтрінің немесе жолақты фильтр сипаттамасын аламыз. Параметр 'ftype' екі мағынаға ие, және белгіленген жағдайда, тағы екі маңызды фильтр үлгісін есептеуге мүмкіндік береді:

high – жоғары жиілік фильтрі;
stop – режекторлық фильтр (берілген Wn=[wl w2]).

Функция

```
[z,p,k]  =  besself(...)
```

нөлдер, полюстер мен күшею коэффициентін есептеуге мүмкіндік береді, ал [A,B,C,D]=besself(...) функциясы күй кеңістігінің параметрлерін анықтайды.

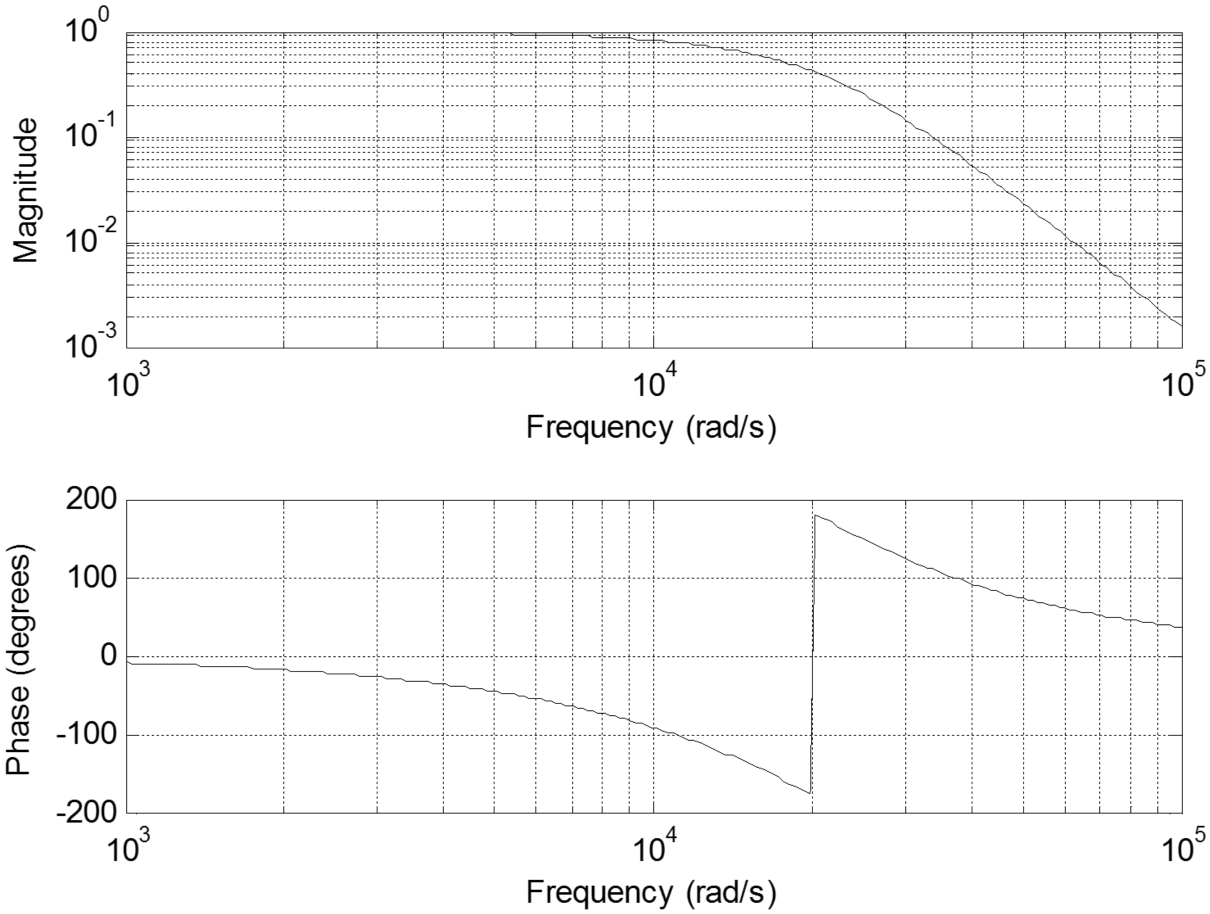
Келесі мысалда шектік жиілігі 20 000 (радианда) 4-реттік Бессель төменгі жиілік фильтрінің коэффициент векторлары есептеледі.

Бессельдің төменгі жиілік фильтрінің коэффициенттерін есептеуге арналған программаның Листингі

```
clear;
clc;
```

```
[b,a] = besself(4,20000);
freqs(b,a);
```

Фильтрдің амплитуда-жиіліктік және фаза-жиіліктік сипаттамасының графигі 9.7- суретте көрсетілген.



9.7 Сурет. 4-реттік Бессель төменгі жиілік фильтрінің амплитуда-жиіліктік және фаза-жиіліктік сипаттамасы

Баттерворт фильтрлерін `butter` *функциялары арқылы жобалау*

Баттерворт фильтрлерін жобалау үшін бірнеше жазба түрі бар butter функциясы қызмет етеді:

```
[b,a] = butter(n,Wn)
[b,a] = butter(n,Wn,'ftype')
[b,a] = butter(n,Wn,'s')
[b,a] = butter(n,Wn,'ftype','s')
[z,p,k] = butter (...)
[A,B,C,D] = butter (. . .)
```

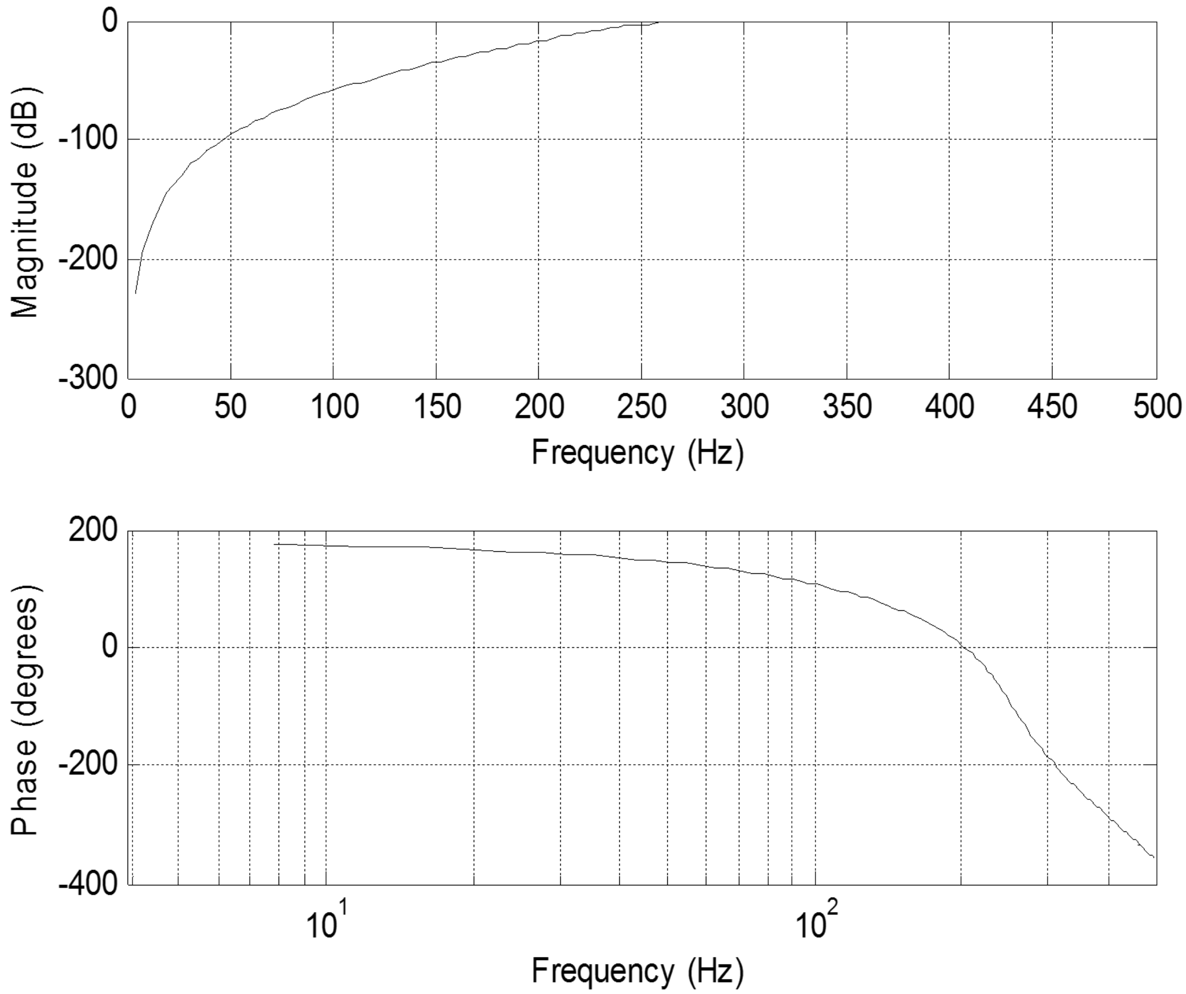
Бұл функция сандық фильтрлерді, сонымен қатар аналогты фильтрлерді есептеуге мүмкіндік береді. Аналогты фильтрлерді есептеу белгісі болып параметр `'s'` табылады. `Butter` функциясының жазба түрінде өзге параметрлерді енгізу, `besself` функциясына енгізгенмен бірдей.

Келесі мысалда шектік жиілігі 0.5 болатын 6-реттік Баттерворт жоғарғы жиілік фильтрінің амплитуда-жиіліктік және фаза-жиіліктік сипаттамасы есептелінеді және тұрғызылады:

Шектік жиілігі 0.5 болатын 6-реттік Баттерворт жоғары жиілік фильтрінің амплитуда-жиіліктік және фаза-жиіліктік сипаттамасын визуалдауға арналған программаның Листингі.

```
[b,a]    = butter(6,.5,'high');
freqz (b,a,128,1000);
```

Берілген фильтрдің амплитуда-жиіліктік және фаза-жиіліктік сипаттамаларының көрінісі 9.8-суретте көрсетілген.



9.8 Сурет. Шектік жиілігі 0.5 болатын 6-реттік Баттерворт жоғарғы жиілікті дискретті фильтрінің амплитуда-жиіліктік және фаза-жиіліктік сипаттамасы

Шексіз импульстік сипаттамасы бар фильтрмен сигналдарды фильтрлеу

Шексіз импульстік сипаттамасы бар фильтр (рекурсивтік фильтр) – екі немесе одан да көп шығысын кіріс ретінде пайдаланатын сызықтық электрондық фильтр, яғни кері байланысы бар.

Мұндай фильтрлердің негізгі қасиеттерінің бірі импульстік өтпелі сипаттамасы уақыттық аймақта шексіз ұзындыққа ие, ал тасымалдау функциясы бөлшек-рационал түрінде. Бұл фильтрлер аналогтық және сандық бола алады.

Шексіз импульстік сипаттамасы бар фильтрлерге мысал Чебышев фильтрі, Баттерворт фильтрі, Калман фильтрі және Бессель фильтрі.

Осы фильтрмен сәйкесінше 5, 15 және 30 Гц жиіліктегі үш құраушысы бар синусоидалық сигналды фильтрлеу мысалын қарастырайық.

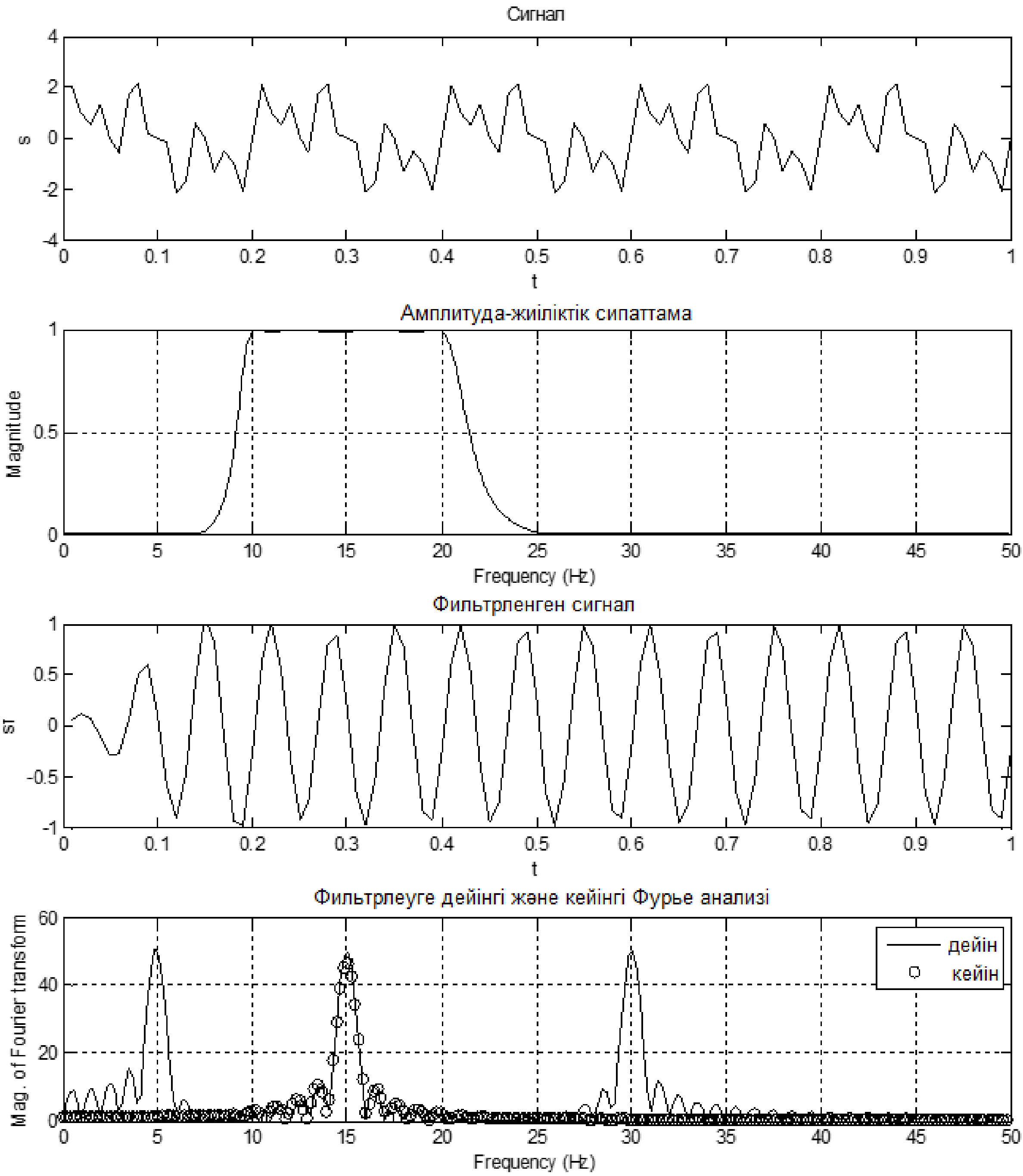
Алдымен сигналды құрастырайық, кейін эллипстік фильтрді модельдеп, сигналдың амплитуда-жиіліктік сипаттамасын зерттейміз. Артынан сигналды фильтрлеу үшін `filter` функциясын қолданамыз. Нәтижесінде алғашқы және фильтрленген сигналдар құраушыларының жиіліктері бойынша қорытынды жасау үшін фильтрлеуге дейінгі және кейінгі сигналдардың Фурье-анализін жүргіземіз.

Шексіз импульстік сипаттамасы бар фильтрмен сигналды фильтрлеу программасының Листингі

```
% Үш синусоидалық құрауышы бар сигналды құрастыру,
% 5, 15 және 30 Гц жиіліктері бар

clear;
clc;
Fs = 100; t = (1:100)/Fs; % Уақыт санағын құру

% Сигналдың синусоидалық құрауыштарын сипаттау
s1 = sin(2*pi*t*5);
s2=sin(2*pi*t*15);
s3=sin(2*pi*t*30);
s = s1+s2+s3; % Нәтижелік сигнал

subplot(411);
plot(t,s); % Сигналды визуалдау
xlabel('t');
ylabel('s');
title ('Сигнал')
```

```
% Өткізу құраушылары жиілігі 15 Гц және кешігу құраушылары
% жиіліктері 5 және 30 Гц болотын фильтр моделін тұрғызу.
[b,a] = ellip(4,0.1,40,[10 20]*2/Fs); % эллиптикалық
% фильтрдің синтезі
[H,w] = freqz(b,a,512); % Комплексті жиілік сипаттамасының
%мәндер векторын және айналмалы  жиіліктердің сәйкес векторын
%есептеу.
subplot(412);
plot(w*Fs/(2*pi),abs(H));
xlabel('Frequency (Hz)');
ylabel('Magnitude');
grid on;
title('Амплитуда-жиіліктік сипаттама')

% filter функциясы арқылы сигналды фильтрлеу
sf = filter(b,a,s); % сигналды фильтрлеу
subplot(413);
plot(t,sf); % Фильтрленген сигналды визуалдау
xlabel('t');
ylabel('sf');
axis([0 1 -1 1]);
title ('Фильтрленген сигнал')

% Фильтрлеуге дейінгі және кейінгі сигналдың Фурье-анализін
% жүргіземіз.
S = fft(s,512); % Фильтрлеуге дейінгі сигналдың Фурье-
% түрлендірілуі
SF = fft(sf,512); % Фильтрлеуге кейінгі сигналдың Фурье-
% түрлендірілуі
w = (0:255)/256*(Fs/2);% Жиіліктер жиынын(массив)қалыптастыру
subplot(414);
plot(w,abs(S(1:256)));
hold on
plot(w,abs(SF(1:256)),'o');
xlabel('Frequency (Hz)');
ylabel('Mag. of Fourier transform');
grid on;
legend({'дейін','кейін'})
title('Фильтрлеуге дейінгі және кейінгі сигналдың Фурье-анализі')
```

Осы бағдарламаға сәйкес есептеулердің нәтижесі 9.9-суретте келтірілген.



9.9 Сурет. Үш синусоидалы құраушылары бар сигналды фильтрлеу нәтижесі

10-суретте көрсетілген графиктерден көріп отырғанымыздай, фильтрленген сигналда 5 және 30 Гц жиілікке ие құраушылары жоқ.

Тапсырма

1. Тұрғызыңыздар: сызықтық жүйенің амплитуда-жиіліктік және фаза-жиіліктік сипаттамаларын; Баттерворт фильтріне арналған топтық кешіктіру уақытының жиілікке тәуелділігін; фильтрдің импульстік сипаттамасын, фильтрдің нөлдері мен полюстерін, 4-ші реттік Бессель төменгі жиілік фильтрінің амплитуда-жиіліктік және фаза-жиіліктік сипаттамаларын; Шектік жиілігі 0.5 болатын 6-

реттік Баттерворт жоғарғы жиілікті дискретті фильтрінің амплитуда-жиіліктік және фаза-жиіліктік сипаттамаларын.
2. Бессель және Баттерворт аналогтық фильтр 20 Гц.-прототиптерінің параметрлерін есептеңіз.
3. 20 Гц-ден жоғары жиіліктегі синусоидальды сигнал құрушыларын шығаруға арналған бағдарламаны жасаңыз.

Бақылау сұрақтары
1. Фильтрлердің мақсаты мен олардың жіктелуін сипаттаңыз.
2. Фильтрдің жиілігі мен фазалық сипаттамалары нені білдіреді?
3. Амплитуда-жиіліктік және фаза-жиіліктік сипаттамаларын тұрғызудың алгоритмін сипаттаңыз.
4. Топтық кешіктіру уақытын есептеудің алгоритмін сипаттаңыз.
5. Сандық фильтрдің импульстік сипаттамасын тұрғызудың алгоритмін сипаттаңыз.
6. Баттерворт фильтрі үшін нөлдер мен полюстер тұрғызудың алгоритмін сипаттаңыз.
7. Сигнал фильтрлерін модельдеу үшін қандай Matlab функциялары қолданылады?
8. Шексіз импулстік сипаттамаға ие фильтрмен сигналды фильтрлеудің алгоритмін сипаттаңыз.

Қолданылған әдібиеттер тізімі

1. Дьяконов В.П. Matlab и Simulink для радиоинженеров. – М.: ДМК Пресс, 2011. – 976 с.
2. Дьяконов В.П. Matlab 6.5 SP1/7+Simulink 5/6. Основы применения. – М.: СОЛОН-Пресс, 2005. – 800 с.
3. Сергиенко А.Б. Цифровая обработка сигналов. СПб.: БХВ_Петербург, 2013. – 768 с.
4. Сато Ю. Без паники! Цифровая обработка сигналов. – М. Додека-XXI, 2010. – 178 с.

Зертханалық жұмыс №10

MATLAB ПРОГРАММАЛЫҚ ОРТАСЫНДА АМПЛИТУДАЛЫҚ МОДУЛЯЦИЯ

Жұмыстың мақсаты: Matlab программалау ортасында сигналды модуляциялау процесін компьютерлік модельдеу.

Қысқаша теориялық кіріспе

Кіріс сигналының спектрі байланыс каналының тиімді өтпейтін жиіліктерінде шоғырлануы - ақпаратты жіберу кезінде жүйелерде жиі кездесетін жағдай. Сондай-ақ бір мезетте бір байланыс каналынан бірнеше сигналдың жіберілуі жиі кездесетін жағдайлардың бірі болып табылады. Бұл қиындықты шешудің бір жолы ретінде каналды жиілікпен бөлу, яғни әр түрлі сигналдар жиілік жолақтарын сәйкесінше толтырады. Бұдан бөлек берілетін сигналдың қысқажолақты, яғни спектрдің эффективті ені орталық енінен едәуір кіші болуы қажет деген талап қойылуы мүмкін.

Бұл қиындықтардың бәрі кіріс сигналының мынадай түрлендіру жасауына негіз болады: жиілік жолақтарына деген талаптардың бәрі орындалып, кіріс сигналын қайтадан қалпына келтіру.

Бұл қиындықтың шешімі сигналды *модуляциялау* болып табылады. Сигналды модуляциялау процесінің негізі тербеліс тасымалдаушы немесе тасымалдаушы деп аталатын қандай да бір тербеліс пайда болады және бұл тербелістің қандай да бір параметрі кіріс сигналына уақыт бойынша пропорциональді түрде өзгеріп отыруында. Кіріс сигналы модуляциялаушы, ал уақыт бойынша параметрі өзгеретін қорытқы тербеліс – модуляцияланған сигнал деп аталады. Модуляцияға кері процесс демодуляция деп аталады. Сандық сигналдар үшін «модуляция» терминінің орнына «манипуляция» термині қолданысқа ие.

Тербеліс тасымалдаушы ретінде қолданылатын гармоникалық сигнал мынадай өрнекпен өрнектеледі.

$$s(t) = A\cos(w_0 t + \varphi). \tag{10.1}$$

Бұл сигнал үш параметрмен сипатталады, амплитудамен A, жиілікпен w_0 және бастапқы фазамен φ. Бұл параметрлердің әрқайсысын модуляциялаушы сигналмен байланыстыру арқылы модуляциялаудың үш түрін: амплитудалық, жиіліктік және фазалық түрлерін алуға болады. Жиіліктік және фазалық модуляцияны көбіне бұрыштық модуляция деп атайды. Егер сигналды модуляциялау барысында амплитудасы және фазасы бір мезетте өзгеретін болса, онда модуляцияның бұл түрі квадратты деп аталады.

Амплитудалық модуляция

Амплитудалық модуляция – тасымалдаушы сигналдың өзгеруші параметрі амплитуда болып табылатын модуляция түрі. Сонымен, амплитудалық модуляция барысында тасымалдаушы тербелістің амплитудасы былайша өзгереді:

$$s_{AM}(t) = A(t)\cos(w_0 t + \varphi_0). \tag{10.2}$$

Музыка мен сөзді амплитудалық модуляция әдісімен радио бойынша алғаш тәжірибе жүргізген американдық инженер Р. Фессенден болып табылады. Радиотаратқыштағы 50 кГц тасымалдағыш жиілік альтернатормен(генератормен) өңделді, ал оны модуляциялау үшін генератор мен антеннаның ортасына тізбектегі сигналдың өшуін өзгертуші бұрыштық микрофон қосылды. 1920 жылдан бастап альтернаторлардың орнына электронды лампалы генераторлар пайдаланыла бастады. 1930 жылдың екінші жартысынан бастап ультрақысқа толқындарды меңгеруге байланысты, радиобайланыс саласында амплитудалық модуляцияны ультрақысқа-толқындық жиіліктік модуляция ығыстырды. XX ғасырдың ортасынан бастап радиобайланыс саласында барлық жиілікте амплитудалық модуляцияның жиынтығынан тұратын біржолақты модуляция енгізілді.

Егер модуляция барысында амплитуданы $A(t)$ модуляциялаушы сигналға тура пропорционал қоятын болсақ, онда мынадай қиындыққа тап болуымыз мүмкін. Ереже бойынша, модуляциялаушы сигнал таңбасы тұрақсыз яғни екіполярлы деп аталады. Мұндай сигналға мысал ретінде келесі өрнекпен өрнектелетін сигналды жатқызуға болады:

$$s_M(t) = 3\cos(2\pi t) - \sin(4\pi t + \pi/6). \tag{10.3}$$

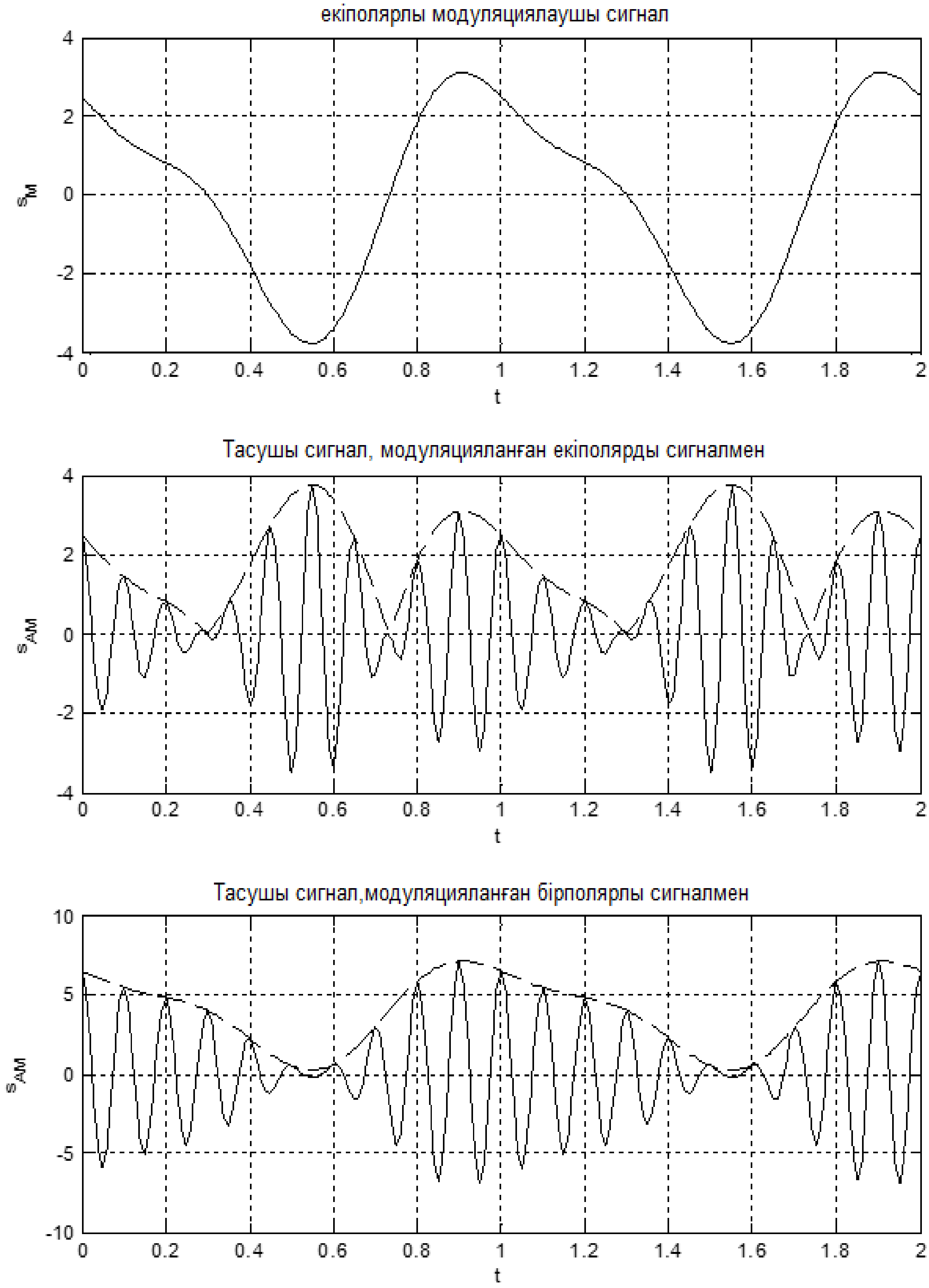
Егер мұны амплитудалық функция $A(t)$ ретінде пайдаланар болсақ, демодуляция барысында кіріс сигналын қалпына келтіруші қисық алынады. Осы себепті амплитудалық модуляцияны іске асыру барысында екіполярлы сигналды бірполярлы сигналға түрлендіруші тұрақты құраушыны қосады. Ал тұрақты құраушыны келесі түрде беретін боламыз

$$A(t) = A_0 + k \cdot s_M(t). \tag{10.4}$$

Сигнал модуляциясына бұл операцияларды қолданудағы нәтижені қарастырайық. Амплитудалық модуляцияланған сигналды шығару үшін программа листингі

```
% Амплитудалық модуляция
clear;
clc;
Fs=100; % Дискреттеу жиілігі
t=0:1/Fs:2; % Уақыт интервалы,
% сигнал реализациясы болған
% екіполярлы модуляциялаушы сигнал:

s_M=3*cos(2*pi*t)-sin(4*pi*t+pi/6);
subplot(311); % Графикалық терезе құру
plot(t,s_M); % Сигнал визуализациясы s_M
grid on % Торды келтіру
xlabel('t');
ylabel('s_M');
title('екіполярлы модуляциялаушы сигнал ')

Fc=10; % Тасушы жиілік
s_AM=s_M.*cos(2*pi*Fc*t); % Сигнал,
% модуляцияланған екіполярлы сигналмен

subplot(312);
plot(t, s_AM, t, abs(s_M), '--');
grid on % Торды келтіру
xlabel('t');
ylabel('s_A_M');
title('Тасушы сигнал, модуляцияланған екіполярлы сигналмен ')

s_AM=(4+s_M).*cos(2*pi*Fc*t); % Бірполярлы
% модуляциялаушы сигнал
% Тұрақты құраушының қосылуы арқылы
% модуляциялаушы сигнал бірполярлы болады.
subplot(313);
plot(t, s_AM, t, 4+s_M, '--');
grid on % торды келтіру

xlabel('t');
ylabel('s_A_M');
title('Тасушы сигнал,модуляцияланған бірполярлы сигналмен')
```

Бұл программаның орындалу нәтижесі 10.1-суретте көрсетілген.



10.1 Сурет. Амплитудалық модуляцияланған сигнал

10.1-суреттің төменгі кескінінен байқағанымыз, амплитудалық жанасудың формасы модуляциялаушы сигналдың соңғы құраушысының, яғни сигнал демодуляциясынан кейін өшіруге болатын құраушының дәлдігімен сәйкес келеді.

Бұл жағдайда, амплитудалық модуляцияланған сигналдың математикалық өрнегі келесі қатынас түрінде жазылуы мүмкін:

$$s_{AM}(t)=\left(A_0+k\cdot s_M(t)\right)\cos\left(w_0t+\varphi\right). \quad (10.5)$$

Біртональді амплитудалық модуляция

Біртональді модуляция амплитудалық модуляцияның қарапайым күйі болып табылады. “Біртональді” термині – тек бір мәнді жиілікті үннің (сигналдың) сәйкестігімен түсіндірілетін “тон” сөзінен алынған.

Амплитудалық модуляцияланған сигналдың құрылымын қарастырайық. Ол үшін модуляциялаушы сигнал гармоникалық болған кездегі жағдайды сипаттайық. Бұл жағдайда оны жазу үшін келесі өрнектер қолданылады:

$$s_M(t)=A_M\cos\left(\Omega t+\Phi_0\right), \quad (10.6)$$

$$s_{AM}(t)=\left(A_0+A_M\cos\left(\Omega t+\Phi_0\right)\right)\cos\left(w_0t+\varphi_0\right). \quad (10.7)$$

Мұндағы $A_м$ – модуляциялаушы сигналдың амплитудасы, ал A_0 – тасушы сигналдың амплитудасы. Бұл амплитудалардың қатынасы

$$m=\frac{A_M}{A_0} \quad \textit{(10.8)}$$

модуляция коэффициенті немесе модуляция тереңдігі деп аталатын параметрді білдіреді. Модуляция тереңдігін амплитуда-модуляциялаушы сигналды тасушы сигналмен салыстыру деңгейі ретінде пайымдауға болады. Егер тасушы және модуляциялаушы сигнал деңгейі бірдей болса, модуляция тереңдігі 100% - ды құрайды. Негізі оптимальді модуляция тереңдігі 30%-ды құрайды, өйткені бұл кезде жақтық гармоника деңгейі минималды болады.

(10.6)-(10.8) қатынастарын есепке ала отырып, жазуға болады:

$$s_{AM}(t)=A_0\left(1+m\cos\left(\Omega t+\Phi_0\right)\right)\cos\left(w_0t+\varphi_0\right). \quad (10.9)$$

Біртональді амплитуда-модуляцияланған сигналда косинус бірлік мәнде болғанда, жанасу максималды мәнге жетеді: $A_{\max} = A(1+m)$. Жанасудың мәні модуляциялаушы сигнал косинусы -1 –мәнге тең болғанда, минималды мәнге ие: $A_{\min} = A(1-m)$. Бұл жағдайда, модуляция коэффициенті келесі түрде жазылады:

$$m = \frac{A_{\max} - A_{\min}}{A_{\max} + A_{\min}}. \qquad (10.10)$$

Негізі $m \in \{0,\ 1\}$. Егер m>1 болса, қайта модуляциялау пайда болады. *Қайта модуляциялау* модуляциялаушы кернеу жоғары болғандағы, яғни оң жағдайда оның жарты толқында модуляциялаушы кернеу амплитудасы орташа мәнмен салыстырғанда 100%-ға өскен кездегі амплитудалық модуляцияның жағдайын түсінеді. Басқа жарты толқында модуляциялаушы кернеу амплитудасы берілетін сигнал орнына модуляцияланған тербелістің бұрмалану формасы қайта модуляциялау кезінде міндетті түрде бақыланғанда кернеу амплитудасы 0-ге дейін төмендеуі мүмкін.

Біртональді амплитуда-модуляцияланған сигнал әр түрлі жиілікті қарапайым гармоникалық тербелістердің суммасы түрінде көрсетілуі мүмкін.

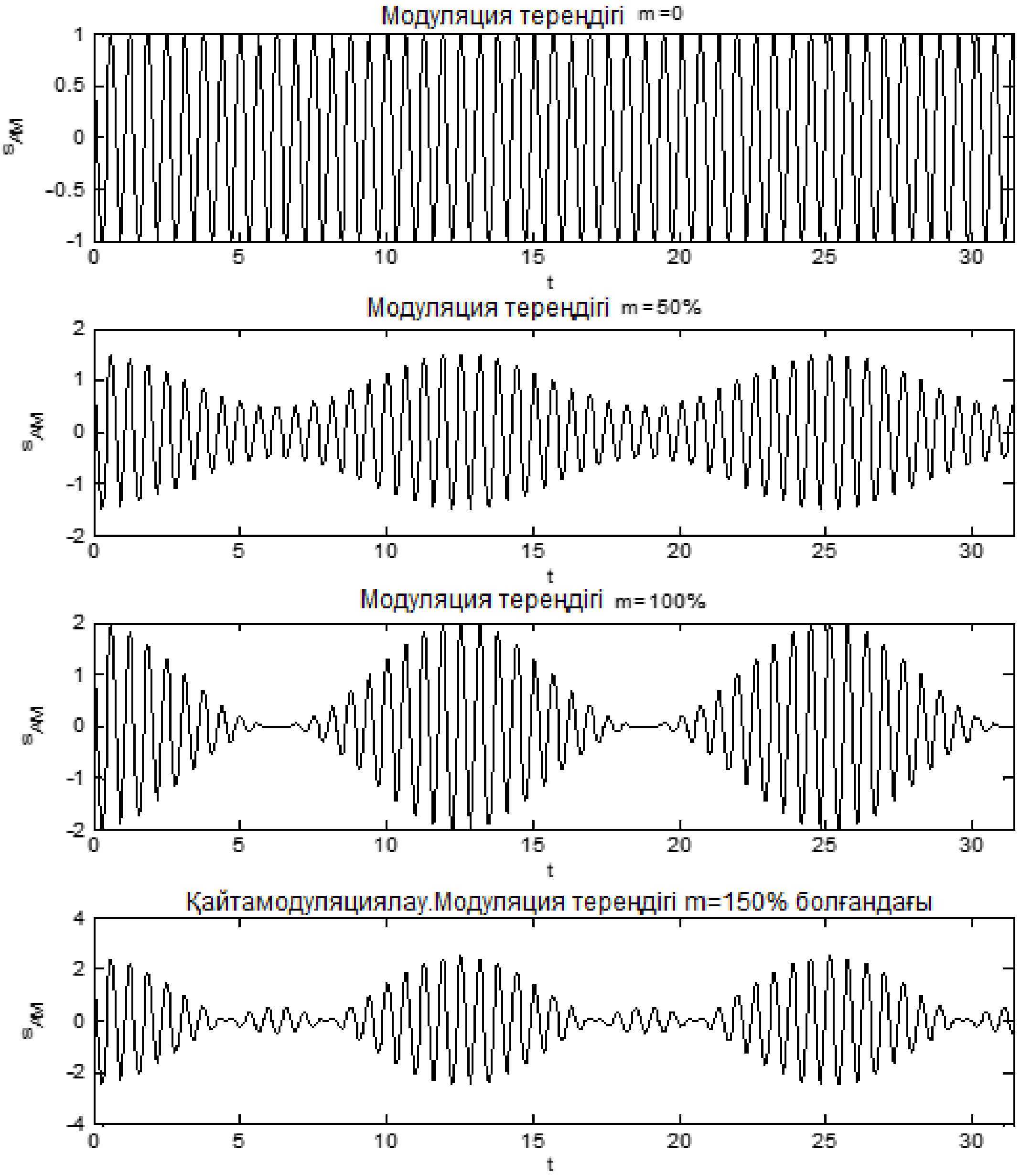
Төменде, түрлі модуляция коэффициентінде амплитудалық-модуляцияланған сигнал алудың листинг программасы көрсетілген.

Модуляция тереңдігінің түрлі мәнінде амплитуда-модуляцияланған сигналды жүргізудегі листинг программасы

```
% Біртональді модуляция
clear;
clc;

Fs=100; % сигналдың дискреттеу жиілігі
t=0:1/Fs:10*pi; % уақыт есебі
omega0=10; % тасушы жиілік
OMEGA=0.5; % модуляцияланатын сигнал жиілігі
m=0; % модуляция тереңдігі
% Біртональді амплитуда-модуляцияланған сигнал
% модуляция тереңдігі ноль болғандағы:
s_AM_0=(1+m*cos(OMEGA*t)).*cos(omega0*t);
subplot(411)% Графикалық терезе құру
plot(t,s_AM_0); % m=0 болғандағы сигнал визуализациясы
xlabel('t');
```

```
ylabel('s_A_M')% Ось бойынша жазу
title (' m=0 болғандағы модуляция тереңдігі')
xlim([min(t) max(t)]) % x осінің аралығы

m=0.5;% Модуляция тереңдігі (50%)
% Біртональді амплитуда-модуляцияланған сигнал
% Модуляция тереңдігі 50% болғандағы:
s_AM_50=(1+m*cos(OMEGA*t)).*cos(omega0*t);
subplot(412) % Графикалық терезе құру
plot(t,s_AM_50);% m=0.5 болғандағы сигнал визуализациясы
xlabel('t');
ylabel('s_A_M')% Ось бойынша жазу
title (' Модуляция тереңдігі50% болғандағы ')
xlim([min(t) max(t)])% x осінің аралығы

m=1; % Модуляция тереңдігі(100%)
s_AM_100=(1+m*cos(OMEGA*t)).*cos(omega0*t);
subplot(413) % Графикалық терезе құру
plot(t,s_AM_100);% m=1 болғандағы сигнал визуализациясы
xlabel('t');
ylabel('s_A_M')% Ось бойынша жазу
title (' Модуляция тереңдігі 100% болғандағы ')
xlim([min(t) max(t)])% x осінің аралығы


% Қайта модуляциялау (m>1)
m=1.5; % Модуляция тереңдігі (150%)
s_AM_150=(1+m*cos(OMEGA*t)).*cos(omega0*t);
subplot(414) % Графикалық терезе құру

plot(t,s_AM_150);% m=1.5 болғандағы сигнал визуализациясы
xlabel('t');
ylabel('s_A_M')% Ось бойынша жазу
title ('Қайта модуляциялау. Модуляция тереңдігі m=150
% болғандағы ')
xlim([min(t) max(t)])% x осінің аралығы
```

Бұл программаның орындалу нәтижесі 10.2-суретте көрсетілген. Бұл суреттен қайта модуляция кезінде сигнал формасының бұрмаланғанын байқауға болады.



10.2 Сурет. Модуляция тереңдігінің түрлі мәніндегі біртональді амплитуда-модуляцияланған сигнал

Амплитудалық модуляцияланған сигнал демодуляциясы

Демодуляция (басқаша сигналды детектрлеу деп аталады) – тасушы жиіліктегі модуляцияланған тербелістен ақпараттық (модуляциялайтын) сигналды бөліп алатын кері модуляция процессі.

Амплитуда-модуляциялық сигнал демодуляциясының кеңінен тараған бір түрі синхронды детектрлеу деп аталады. Бұл процессте амплитудалық модуляцияланған сигнал тасушы тербеліс жиілігіне ие кейбір модуляцияланбаған тербеліске көбейтіледі. Соның негізінде бұл сигнал төменгі жиілікті фильтрден өтеді. Нәтижесінде, екі құрылымнан тұратын сигнал пайда болады. Бұл құрылымның бірі бастапқы модуляциялайтын сигналға тура пропорционалды, ал екіншісі – екі еселенген тасушы жиілікпен амплитудалық модуляцияланған сигнал.

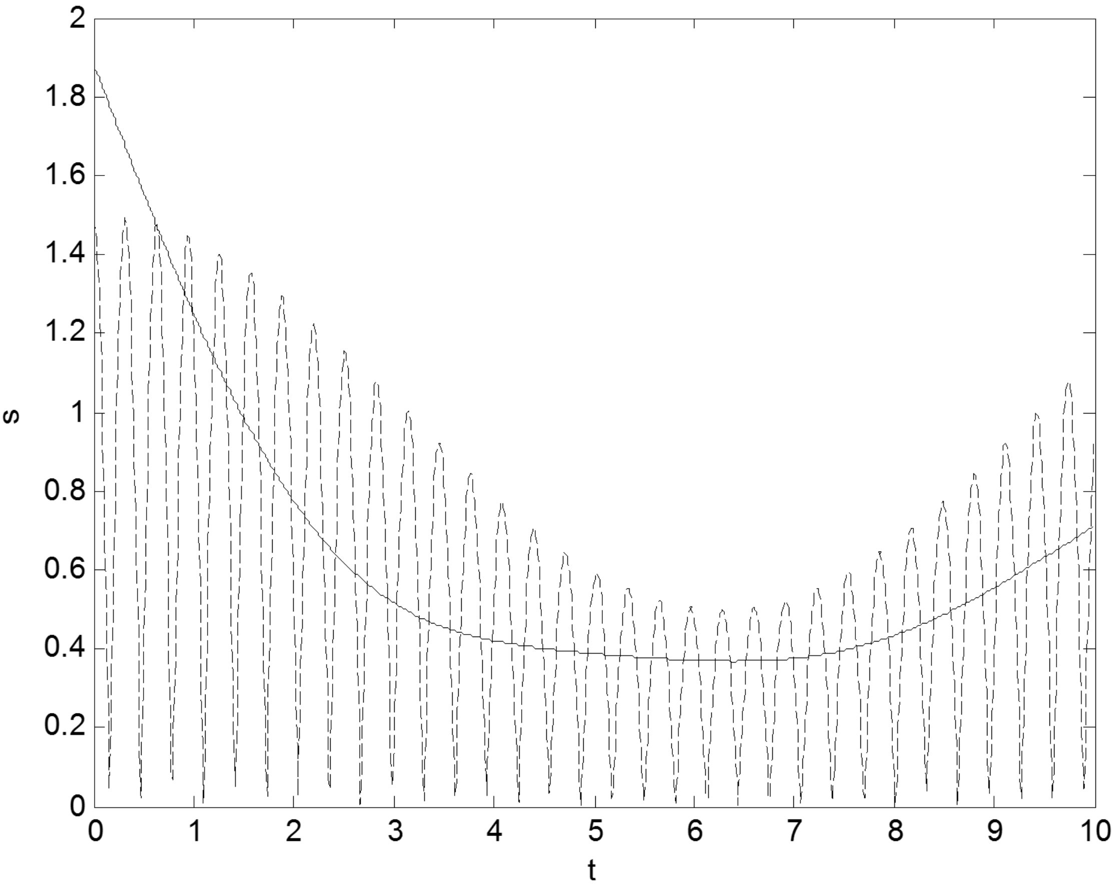
Амплитудалық модуляцияланған сигнал демодуляциясын жүзеге асырудың ең қарапайым әдісінің бірі - аналогты екі жарты периодты детектор жұмысының имитациясы болып табылады. Негізі детектор (демодулятор) құрылғының электронды түйінінен сондай-ақ тасушы құраушыдан тұратын модуляцияланған сигналды көрсетеді. Амплитудалық модуляция жағдайында демодулятор ретінде ең қарапайым түрде диод немесе басқа бейсызық элемент алынады. Оның жұмыс істеу принципі тасушының жиілігі модуляцияланатын сигнал жиілігінен біршама жоғары, ал модуляция коэффициенті 1-ден аз,сондай-ақ сигнал қайта модуляциясы қосылмайтынына негізделген. Бұл жағдайда сигнал құрылғы кірісінде тегістеледі және кесу жиілігінен үлкен төменгі жиілікті фильтр көмегімен фильтрленеді.

Аналогты екіжартыпериодты детекторды модельдеу жұмысында келесі әрекеттерді орындаймыз. Ең алдымен кіріс амплитудалық модуляцияланған сигнал кірісінің модулін есептейміз, содан соң бірполярды косинусты импульсты төменгі жиілікті фильтрден өткізе отырып тегістейміз. Бұл әдіс қайта модуляция кезінде жұмыс жасамайды, басқаша айтқанда модуляция тереңдігі 0 мен 1 аралығында болуы керек.

Компьютерлік модельдеудің MatLab ортасы көмегімен амплитудалық модуляцияланған сигнал демодуляциясының әдіснемесі төменде көрсетілген листинг программада сипатталған.

Амплитудалық модуляцияланған сигнал демодуляциясын жүзеге асырудың листинг программасы

```
% Амплитудалық модуляцияланған сигнал демодуляциясы
clear; clc;
Fs=100; % Сигналдың дискреттеу жиілігі
t=0:1/Fs:10*pi; % Уақыт есебі
omega0=10; % Тасушы жиілік
OMEGA=0.5; % Модуляциялаушы сигнал жиілігі
m=0.5;% Модуляция тереңдігі (50%)
% Біртональді амплитуда-модуляцияланған сигнал
% модуляция тереңдігі 50% болғандағы:
s_AM_50=(1+m*cos(OMEGA*t)).*cos(omega0*t);
y=abs(s_AM_50); % Амплитудалық модуляцияланған сигнал модулі
[b,a]=butter(5, 2*OMEGA/pi/Fs); % Тегістеу
% төменгі жиілікті фильтр
z=filtfilt(b,a,y); % Фильтрлеу
plot(t(1:1000), y(1:1000), '--', t(1:1000),z(1:1000))
% Бірінші 1000 сигналдың визуализациясы
xlabel('t'); ylabel('s')
```



10.3 Сурет. Амплитудалық модуляцияланған сигналдың екіжартыпериодты детектрленуі

Жоғарыдағы жазылған алгоритмдердің негізінде амплитудалық модуляцияны және демодуляцияны, тасушы біржолақты модуляцияны және демодуляцияны, сондай-ақ полярлы модуляцияны жүзеге асыруға болады. Амплитудалық модуляцияда таратқыштың шығысында тасушы жиілікпен тасушы сигнал кірісінде тек модуляциялайтын сигнал түрінде қатысады. Біржолақты модуляцияда тасушы жиілікпен және жолақтың бір жақтық жиілігімен жіберіледі. Ол үшін сигналдың орнын басатын жиілік жолағы екі есеге азаяды, яғни сол жиілік диапазонында екі еселенген байланыс каналдарын орындауға мүмкіндік береді. Полярлы модуляцияда бір уақытта екі сигнал беріледі, мысалы, стерео сигналдарды жіберуде қажет болады.

Осылайша, компьютерлік модельдеудің жоғарыда көрсетілген әдістері амплитудалық модуляцияның әртүрлі түрлерін жүзеге асыруға қолданылуы мүмкін.

Тапсырма

1. Шулы синусоидалды сигналмен амплитудалық модуляцияны жүзеге асыр.
2. Тікбұрышты импульстпен тізбектелген амплитудалық модуляцияны жүзеге асыр.

3. Ара тәрізді амплитудалық-модуляцияланған сигнал құрыңыз.
4. Қайта модуляциясыз амплитудалық модуляцияланған сигнал түріне модуляция коэффициентінің әсерін зерттеңіз.
5. Қайта модуляция қосылған амплитудалық модуляцияланған сигнал түріне модуляция коэффициентінің әсерін зерттеңіз.
6. Демодуляцияланған сигнал моделін алыңыз.

Бақылау сұрақтары

1. Модуляция дегеніміз не? Модуляцияның қандай түрлері бар?
2. Амплитудалық модуляцияны жүзеге асырудағы алгоритмдерін жазыңыз.
3. Манипуляция дегеніміз не?
4. Тасушы тербеліс дегеніміз не?
5. Модуляцияланатын сигнал дегеніміз не?
6. Қандай сигналды екіполярды деп атайды?
7. Сигнал демодуляциясының мәні қандай?
8. Модуляция тереңдігін есептеудің формуласын жазыңыз. Оның мағынасы қандай?
9. Қайта модуляциялау деген не?
10. Біртональді амплитудалық модуляция дегеніміз не?
11. Синхронды детектрлеу процессін сипаттаңыз.
12. Біржолақты модуляцияда жалпы қандай жиіліктер беріледі?
13. Модуляцияның қай түрінде сигнал айналасында, жиілік жолағы екі есеге қысқаауы мүмкін?
14. Полярлы модуляция дегеніміз не және ол қандай жағдайда қолданылады?

Қолданылған әдібиеттер тізімі

1. Сергиенко А.Б. Цифровая обработка сигналов. СПб.: БХВ-Петербург, 2013. – 768 с.
2. Дьяконов В.П. Matlab 6.5 SP1/7+Simulink 5/6. Основы применения. – М.: СОЛОН-Пресс, 2005. – 800 с.
3. Дьяконов В.П. Matlab и Simulink для радиоинженеров. – М.: ДМК Пресс, 2011. – 976 с.
4. Сато Ю. Без паники! Цифровая обработка сигналов. – М. Додека-XXI, 2010. – 178 с.
5. Matlab. Специальный справочник. – СПб: Питер, 2010. – 480 с.
6. Ярмоленко В.И., Приоров А.Л. Сигналы в радиотехнических и телекоммуникационных системах: учеб. Пособие / Ярославль: ЯрГУ, 2009. - 100 с.