

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ»
КАФЕДРА ЕЛЕКТРОННОЇ ТЕХНІКИ

Методичні вказівки
до практичних робіт
з дисципліни
«МЕТОДИ ДОСЛІДЖЕННЯ ТА АНАЛІЗУ ДИСКРЕТНИХ СИГНАЛІВ»
ДЛЯ СТУДЕНТІВ СПЕЦІАЛЬНОСТІ 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ
ДЕННОЇ ТА ЗАОЧНОЇ ФОРМИ НАВЧАННЯ

Луцьк
2023

УДК
К

Методичні вказівки до практичних робіт з дисципліни «Методи дослідження та аналізу дискретних сигналів» для студентів спеціальності 123 Комп'ютерна інженерія усіх форм навчання [Електронний ресурс] / уклад. : С.О. Ковальов, В.В. Шамаєв. – Луцьк : ДонНТУ, 2023. – 66 с.

Методичні вказівки містять основні положення щодо вимог, виконання, структури та оформлення звітів з практичних робіт з дисципліни «Методи дослідження та аналізу дискретних сигналів» магістрами ДонНТУ, що навчаються за спеціальністю 123 Комп'ютерна інженерія усіх форм навчання. Наведено зразки виконання окремих алгоритмів, видів текстового матеріалу, схем, таблиць, формул та ілюстрацій.

Методичні вказівки до практичних робіт з дисципліни «Методи дослідження та аналізу дискретних сигналів» розроблено за методичними матеріалами О.В. Самощенко. Автори щиро вдячні йому за співпрацю та надані матеріали.

Призначено для магістрів спеціальності 123 Комп'ютерна інженерія

Укладачі: С.О. Ковальов, доц., к.т.н., доц. каф. ЕТ.
В.В. Шамаєв, доц., к.т.н., доц. каф. ЕТ.

Рецензент: Н.О. Маслова, доц., к.т.н., доц. каф. ПМІ.

Відповідальний за випуск: О.В. Вовна, проф., д.т.н., зав. каф. ЕТ.

Затверджено навчально-методичним відділом ДонНТУ
Протокол НМВ № 9 від 27.06.2023 р.

Розглянуто на засіданні кафедри електронної техніки
Протокол № 11 від 8.06.2023 р.

Зміст

Вступ	4
1. Основні положення дисципліни «Методи дослідження та аналізу дискретних сигналів»	5
2. Практична робота № 1. Перетворення Фур'є безперервних сигналів	12
3. Практична робота № 2. Первинні перетворення графічних зображень..	20
4. Практична робота № 3. Дискретизація сигналів.	28
5. Практична робота № 4. Інтерполяція дискретних сигналів.	35
6. Практична робота № 5. Квантування аудіо сигналів.	40
Список рекомендованої літератури.	47
Додаток А. Приклад виконання і порядок оформлення результатів роботи	48
Додаток Б. Алгоритми цифрового опрацювання сигналів.	62

ВСТУП

Основна мета виконання практичних робіт – отримання практичних навичок вирішення базових прикладних задач цифрової (комп'ютерної) обробки дискретних сигналів.

Виконання завдань з практичних робіт зумовлює поглиблене вивчення наступних тем: первинне перетворення графічних зображень (методика керування рівнем яскравості монохромних зображень), перетворення Фур'є безперервних сигналів (перетворення періодичних та неперіодичних безперервних сигналів до частотної форми, тобто формування амплітудного і частотного спектрів, відтворення сигналу до часової форми із частотної), дискретизація сигналів (методика перетворення безперервного сигналу до дискретної форми подання згідно положень теореми Котельникова (критерію Найквіста) та дослідження умов проявлення аліасінгу), інтерполяція сигналів (застосування методів поліноміальної інтерполяції при перетворенні одновимірних та двовимірних сигналів, дослідження методу ідеалізованого відтворення дискретного сигналу до безперервної форми та використання для цього положень поліноміальної інтерполяції (першого, другого, третього рівнів) і інтерполяції сплайнами), квантування сигналів (рівномірне та нерівномірне квантування одновимірних та двовимірних сигналів за визначеними параметрами експерименту, оцінка похибки квантування).

Кожна тема має мінімально необхідний попередній опис функцій системи MATLAB, основних положень з їх практичного впровадження, формулювання завдань, порядок і приклади їх виконання, склад звіту з виконання практичної роботи, перелік контрольних питань, перелік додаткових навчальних джерел.

Практичні роботи, опис яких наведено, є початковими в тематичному циклі з комп'ютерного дослідження, аналізу та перетворення дискретних сигналів, зокрема, звуку, зображень, відео, і допомагають студентам засвоїти важливі відповідні основні тематичні поняття.

1. Основні положення дисципліни «Методи дослідження та аналізу дискретних сигналів»

Особливістю аналогових сигналів є те, що сукупність їхніх миттєвих значень на довільному часовому інтервалі описується неперервною (нескінченною) множиною точок, кожна з яких відображає миттєве значення сигналу у відповідний момент часу і може набувати неперервну (нескінченну) множину значень. Отже, уся множина точок містить у собі нескінченну кількість інформації, яка малопридатна для сприйняття, аналізу та ефективної обробки і потребує стискання первинної інформації без істотної втрати корисної інформації [2,3].

Дискретизація – подання сигналу скінченною множиною миттєвих значень. Перетворення аналогового сигналу в дискретний у часі чи цифровий сигнал еквівалентне поданню його скінченною множиною точок, що полегшує зберігання та оброблення інформації, дає змогу збільшувати кількість сигналів, які поширюються по одному й тому самому каналу зв'язку (ущільнювати канали зв'язку), стискати первинну інформацію тощо [2,3].

Дискретизацією називають таке перетворення аналогового сигналу $s(t)$, за якого його описують множиною миттєвих значень $s(t_i)$ у фіксовані моменти часу $t_1, t_2, \dots, t_n$ (рис.1.1) [3,8]

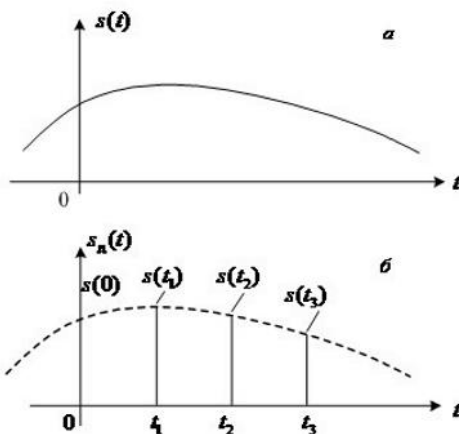


Рис.1.1. Дискретизація аналогового сигналу [3]

Миттєві значення $s(t_i)$ дискретизованого сигналу називають **відліками**, а часовий інтервал $\Delta t = t_i - t_{i-1} = T_0$ між двома сусідніми відліками – **інтервалом дискретизації** [3].

Збільшення інтервалу дискретизації збільшує пропускну здатність каналу зв'язку, але одночасно збільшує і похибку відтворення первинного сигналу $s(t)$. Тому інтервали дискретизації вибирають з урахуванням того, щоб на підставі наявних відліків $s(t_i)$ можна відтворити із заданою точністю первинну функцію $s(t)$ на приймальному пункті [3].

Здебільшого для сигналів, які утворюються у джерелах інформації, характерні порівняно повільні зміни в часі миттєвих значень та велика тривалість. Аналіз спектральних властивостей таких сигналів показує, що їхній спектральний склад зосереджується в обмеженому діапазоні (смузі) частот $0 \dots \omega_B$, а спектральні складові з частотами, вищими від ω_B , практично відсутні. Такі сигнали прийнято називати **сигналами з обмеженим спектром**. Саме ця властивість дає змогу замінити, а відтак знову відновити із заданою точністю неперервний сигнал, заданий скінченною множиною відліків, узятих у дискретні моменти часу [3].

Умови вибору інтервалу дискретизації сигналу з обмеженим спектром визначає теорема відліків (теорема Котельникова – Шеннона).

Умови вибору інтервалу дискретизації Δt , який забезпечує відновлення із заданою точністю первинного аналогового сигналу $s(t)$, формулює теорема відліків (теорема Котельникова – Шеннона): неперервний сигнал $s(t)$, у спектрі якого відсутні частоти, вищі від ω_B , повністю визначається послідовністю своїх миттєвих значень, узятих через інтервал часу $\Delta t \leq \pi/\omega_B$, і може бути поданий рядом [3,8]:

$$s(t) = \sum_{k=-\infty}^{\infty} s(k\Delta t) \varphi_k(t) \quad (1.32)$$

в якому базисні функції мають вигляд:

$$\varphi_k(t) = \sin[\omega_B(t - k\Delta t)] / [\omega_B(t - k\Delta t)] \quad (1.33)$$

Ряд (1.32) - це ряд Котельникова, він дозволяє визначити первинний сигнал у довільний момент часу на підставі відліків його миттєвих значень, які слугують ваговими коефіцієнтами при базисних функціях.

Базисні функції мають таку властивість [3]:

$$\varphi_k(t - n\Delta t) = \begin{cases} 1 & \text{if } n = k \\ 0 & \text{if } n \neq k \end{cases}$$

де n – ціле додатне або від'ємне число.

На рис.1.2 наведено графіки базисних функцій для різних значень k , де кожна базисна функція $j_k(t)$ зсунута стосовно найближчої сусідньої функції $j_{k-1}(t)$ або $j_{k+1}(t)$ на час $T_0 = \pi/\omega_B$, який відповідає інтервалу дискретизації Dt [3].

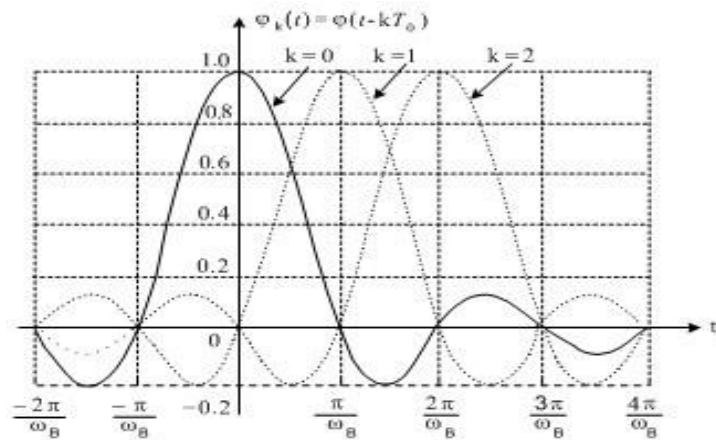


Рис.1.2. Графіки базисних функцій $j(t-kT_0)$ [3]

Відновлення аналогового сигналу $s(t)$ на підставі ряду (1.32) зображено на рис.1.8, звідки бачимо, що точні значення сигналу $s(t)$ отримуємо в точках відліку $\dots -2T_0, -T_0, 0, T_0, 2T_0$ і т. д. Між точками відліку сигнал визначається точно лише тоді, коли додаються усі члени ряду, кількість яких нескінченно велика [2,3].

Точне відновлення аналогового сигналу можливе лише за умови урахування нескінченної кількості відліків [3].

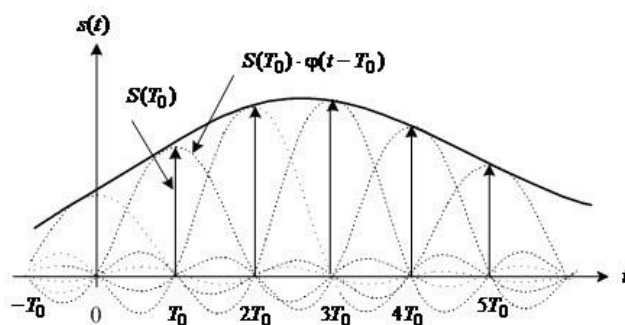


Рис.1.3. Відновлення аналогового сигналу на підставі послідовності відліків

Математична модель дискретизованого сигналу вимагає використання функції Дірака (d-функції) [2,3].

Якщо неперервний в часі сигнал $s(t)$ з найвищою частотою спектра ω_B подати через ряд (1.32) на деякому обмеженому інтервалі T_C , то кількість відліків N дорівнюватиме: $N = T_C/Dt = T_C\omega_B/\pi$. Цю кількість відліків називають **кількістю ступенів свободи сигналу $s(t)$** , або **базою сигналу**. [3]

Сигнал наближено описують рядом, який складається із скінченної кількості членів, і відтворюється точно лише в точках відліку kD_t . У проміжках між відліками з'являється похибка апроксимації, яка збільшується біля країв інтервалу T_0 , де відкинуті члени ряду мають найбільше значення [3,8].

Математичну модель дискретизованого сигналу можна подати у вигляді добутку первинного неперервного аналогового сигналу $s(t)$ і періодичної послідовності σ -імпульсів [3].

Правильний вибір частоти дискретизації забезпечує відтворення первинного сигналу за допомогою фільтру частот [3].

$$S_d(t) = s(t) \sum_{k=-\infty}^{\infty} \delta(t - k\Delta t) \quad (1.34)$$

Для спектрального опису дискретизованого сигналу $s_d(t)$ застосовують до функції (1.34) пряме перетворення Фур'є [2].

Пропускаючи відповідні математичні викладки, зазначимо, що модуль спектральної функції $G_d(\omega)$ дискретизованого сигналу $s_d(t)$ має вигляд модуля спектральної функції $G(\omega)$ первинного неперервного аналогового сигналу і повторюється з частотою дискретизації $\omega_0 = 2\pi/D_t = 2\omega_B$. Формування модуля спектральної функції $G_d(\omega)$ дискретизованого сигналу показано на рис.1.4 [3].

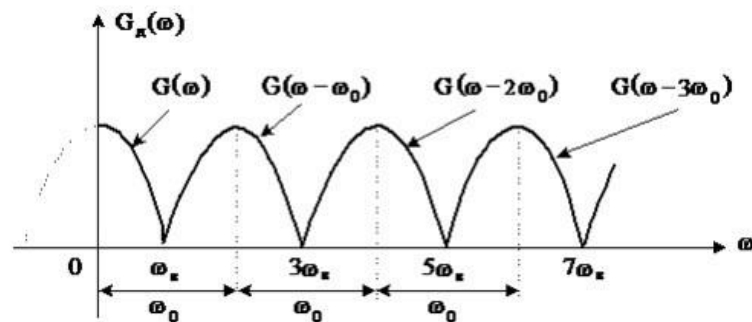


Рис.1.4. Модуль спектральної функції $G_d(\omega)$ дискретизованого сигналу [3]

На практиці частоту дискретизації ω_d вибирають дещо вищою, ніж $2\omega_B$ (тобто $Dt < \pi/\omega_B$), оскільки у такому разі сусідні спектри не перекриваються між собою, що впливає із рис.1.9. Тоді можна відтворити первинний сигнал із дискретизованого, виділивши перший пелюсток спектра за допомогою фільтру нижніх частот [3].

Практично важливим є питання спектрального опису дискретизованого сигналу $s_d(t)$, який заданий на часовому інтервалі $[0, T_c]$ своїми відліками $s(0), s(Dt), s(2Dt), \dots, s((N-1)Dt)$ [3].

Загальна кількість відліків дорівнює $N=T_c/Dt$. Спектральний опис такого сигналу ґрунтується на припущенні, що сигнал $s_d(t)$ є періодичним з періодом T_c , тому застосовують його розклад у ряд Фур'є і знаходять відповідні амплітудні коефіцієнти. Спектральний опис дискретизованого сигналу ґрунтується на припущенні про періодичність цього сигналу.

Часовий опис дискретизованого сигналу має вигляд [3]:

$$s_d(t) = \sum_{n=0}^{N-1} s(n\Delta t) - \delta(t - n\Delta t) * \Delta t \quad (1.35)$$

Комплексний ряд Фур'є такого сигналу описує формула [4]:

$$s_d(t) = \sum_{k=-\infty}^{\infty} \bar{A} \exp\left(\frac{j2\pi kt}{T_c}\right) \quad (1.36)$$

де комплексні амплітуди визначаємо за формулою [4]:

$$A_k = \frac{1}{T_c} \int_0^{T_c} S_d(t) - \exp(-j2\pi kt/T_c) dt \quad (1.37)$$

Підставивши вираз (1.35) в (1.37) та враховуючи фільтрувальну властивість дельта-імпульсу, отримуємо [3]:

$$A_k = \frac{1}{N} \sum_{n=0}^{N-1} s(n\Delta t) - \exp(-j2\pi kn/N) \quad (1.38)$$

Формулу (1.38) називають **дискретним перетворенням Фур'є (ДПФ) розглянутого сигналу**. Вона визначає сукупність коефіцієнтів ряду Фур'є, які утворюють спектр дискретизованого сигналу [3].

Кількість відліків дискретизованого сигналу дорівнює подвоєній кількості гармонічних складових, які можна визначити на підставі ДПФ [3].

Аналіз властивостей ДПФ показує, що кількість різних комплексних амплітуд (крім постійної складової), які можна визначити за формулою (1.38), дорівнює половині кількості відліків за період T_c , тобто $N/2$ [3,5].

Постійна складова дискретизованого сигналу дорівнює середньому арифметичному усіх відліків сигналу [3]:

$$A_0 = \left[\sum_{n=0}^{N-1} s(n\Delta t) \right] / N$$

На практиці досить часто розглядають зворотну задачу дискретного спектрального аналізу: за відомими коефіцієнтами $\bar{A}_k$ ДПФ треба визначити відліки дискретизованого сигналу [3].

Для цього у формулі (1.36) приймають $t=nDt$ і підсумовують скінченну кількість членів ряду, які відповідають гармонікам, що входять до ряду [3].

Отже, відліки дискретизованого сигналу визначають за формулою [3]:

$$s(n\Delta t) = \sum_{k=0}^{N-1} A_k \exp\left(\frac{j2\pi kn}{n}\right) \quad (1.39)$$

яка описує зворотне (обернене) дискретне **перетворення Фур'є**.

Цифрові сигнали подають у вигляді n -розрядних двійкових чисел.

Дискретні перетворення Фур'є широко застосовують, використовуючи цифрові сигнали для подання та оброблення різноманітної інформації.

Цифрові сигнали – це дискретизовані у часі та квантовані за рівнем сигнали. Кожен з рівнів подається, як правило, у вигляді n - розрядного двійкового числа [3,5]:

$$A_n = a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \dots + a_1 2^1 + a_0 = \sum_{i=0}^{n-1} a_i (2^i) \quad (1.40)$$

a_i – цифра в i -му розряді двійкового числа, може набувати значення 0 або 1.

Множник 2^i відіграє роль вагового коефіцієнта. Отже, двійкові числа подають у вигляді комбінацій нулів та одиниць, які відповідають значенням цифр у відповідних розрядах [3].

Цифровий сигнал можна сформулювати за принципом «додатної» або «від'ємної» логіки у вигляді паралельного або послідовного коду [3].

В цифровому сигналі роль «нулів» та «одиниць» відіграють звичайно сигнали напруги, які можуть бути сформовані за двома принципами [3]:

а) додатної логіки – коли логічній «одиниці» відповідає високе значення потенціалу, а логічному «нулеві» – низьке;

б) від'ємної логіки – коли логічній «одиниці» відповідає низьке значення потенціалу, а логічному «нулеві» – високе.

Значення потенціалів, що відповідають «нулеві» та «одиниці», називають стандартними значеннями логічних сигналів.

Таким чином, *цифровий сигнал – це послідовність комбінацій стандартних значень логічних сигналів, які утворюють n -розрядний цифровий код*. Зауважимо, що стандартні сигнали відповідних розрядів можуть з'являться одночасно (паралельний код) або послідовно (послідовний код).

Найпоширеніший спосіб перетворення аналогового сигналу на цифровий полягає у дискретизації аналогового сигналу в часі та у квантуванні відліків за рівнем. Коротко зупинимось на особливостях квантування за рівнем [2,3].

При квантуванні сигналу за рівнем неперервна множина значень аналогового сигналу $s(t)$ замінюється множиною дискретних значень. Враховуючи це, вибирають у заданому діапазоні значень $s_{\min} \dots s_{\max}$ скінченну кількість дискретних значень (дискретних рівнів) і в моменти відліків $t_k = kDt$ значення сигналу $s(kDt)$ замінюється найближчим дискретним значенням, як показано на рис.1.5 [3].

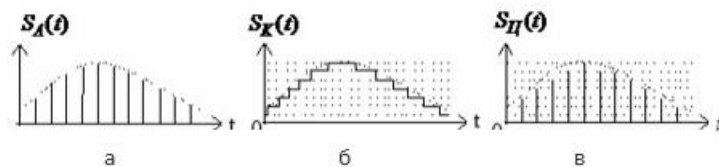


Рис.1.5. Формування цифрового сигналу [3]

Відстань Ds між сусідніми дискретними рівнями називають **кроком квантування**. Загальна кількість дискретних рівнів у заданому діапазоні значень сигналу дорівнює $M = (s_{\max} - s_{\min}) / \Delta s$. Оскільки кожному значенню дискретного рівня ставиться у відповідність певний n -розрядний двійковий код, то кількість необхідних розрядів визначають за формулою [3,9]:

$$n \geq \log_2 M \quad (\text{або } 2^n \geq M) \quad (1.41)$$

Кількість розрядів двійкового коду визначає кількість кроків квантування. Найчастіше двійковий код відображає двійковий номер дискретного рівня.

Оскільки при квантуванні миттєве значення сигналу замінюється найближчим значенням дискретного рівня, то з'являється методична (систематична) похибка - **шум квантування**. Похибка квантування має випадковий характер, її абсолютне значення у будь-який момент часу дорівнює різниці між квантованим значенням $s_{k\delta}(t)$ та миттєвим значенням сигналу $s(t)$ і не перевищує половини кроку квантування. Наявність шуму квантування – особливість процесу квантування [3].

Наявність методичної похибки означає, що істотною відмінністю квантування за рівнем від дискретизації у часі є те, що після реалізації квантування аналоговий сигнал не можна відновити з похибкою, меншою, ніж половина кроку квантування [5].

Практична робота № 1.

ПЕРЕТВОРЕННЯ ФУР'Є БЕЗПЕРЕРВНИХ СИГНАЛІВ [9].

Мета практичної роботи – дослідження методики перетворення Фур'є безперервних сигналів в середовищі системи MATLAB.

1. СПЕКТР БЕЗПЕРЕРВНОГО СИГНАЛУ

Безперервний періодичний сигнал, що складається із декількох сигналів синусоїдальної форми, в середовищі системи MATLAB може бути сформований наступним чином [5,10]:

```
clear all
f1=1; f2=2; f3=7; f4=4; % Частоти гармонік у Гц fmax=max([f1;f2;f3;f4]);
phi1=-pi/2; phi2=pi/4; phi3=pi/6; phi4=-pi/3; % Фази гармонік
% Генерація сигналу з гармонік:
t=-1:0.00001:2; % вектор моментів безперервного часу
s1=cos(2*pi*f1.*t+phi1); % Перша гармоніка
s2=0.5*sin(2*pi*f2.*t+phi2); % Друга гармоніка
s3=-0.25*cos(2*pi*f3.*t+phi3); % Третя гармоніка
s4=-0.18*cos(2*pi*f4.*t+phi4); % Четверта гармоніка
s0=-0.15; % Постійна сигналу
s=s0'+s1'+s2'+s3'+s4'; % Безперервний сигнал
subplot(2,2,1); plot(t,s,t,s1,t,s2,t,s3,t,s4); title('Аналоговий сигнал и гармоніки');
xlabel('Час (сек.)') ylabel('Значення сигналу (мВ)')
legend('Сигнал', '1 гарм', '2 гарм', '3 гарм', '4 гарм')
```

Сигнал, який сформовано, представлений в декількох періодах (в наведеному прикладі наявні 3 періоди по 1 секунді). Для формування спектрів таких періодичних сигналів достатньо аналізувати один період. Один період даного сигналу можна виділити наступним чином [10]:

```
% ---- Виділення одного періоду:
sp=s(1:100000); % 100000 тактів
tp=0:0.00001:(1-0.00001); % Один період сигналу (1сек) - в періоді 100000
значень сигналу (тактів) з інтервалом 10мкс
subplot(2,2,2); plot(tp,sp); title('Один період сигналу');
xlabel('Время (сек.)')
ylabel('Значення сигналу (мВ)') % -----
```

Формули прямого і зворотного (оберненого) перетворення Фур'є безперервних сигналів мають наступний вид [10]:

$$\dot{S}(f) = \int_{-\infty}^{\infty} s(t) \cdot e^{-j2\pi ft} dt, \quad s(t) = \int_{-\infty}^{\infty} \dot{S}(f) e^{j2\pi ft} df.$$

Модуль $|S(f)|$ спектральної функції називають амплітудним спектром, а її аргумент $\varphi(f)$ – фазовим спектром.

У системі MATLAB сигнал завжди дискретизований. Для формування амплітудного та фазового спектрів таких сигналів застосовується спеціальна вбудована функція $fft(x)$, тобто функція прямого перетворення Фур'є для вектору x (якщо x - матриця, то перетворення для кожного стовбця матриці роздільно). Функція $fft(x)$ формує деякий набір комплексних значень спектральної щільності. Результат залежить від значень вхідного сигналу, кількості відкликів (розміру інтервалу спектрального аналізу), параметрів дискретного подання в системі MATLAB. Обернене перетворення Фур'є реалізовано функцією $ifft(x)$ [1,8].

Для наведеного прикладу сигналу амплітудний та фазовий спектри сформовано наступним чином [10]:

```
N=length(sp);
YdFs=roundn(fft(sp),-2); % Комплексні спектральні коефіцієнти
mdF(2:N) = 2*abs(YdFs(2:N))/N; % Модуль змінних гармонік сигналу
mdF(1)=abs(YdFs(1))/N; % Модуль постійної складової
pdF = angle(YdFs)/pi; % Кут
% Вибірка необхідних коефіцієнтів (по числу гармонік + нульова):
mg=mdF(1:(fmax+1));
pg=pdF(1:(fmax+1));
subplot(2,2,3); stem((0:fmax),mg); title('Амплітуда гармонік одного періоду
сигналу');
set(gca, 'xtick', 0:1:fmax)
xlabel('Частоти гармонік (Гц)')
ylabel('Модулі амплітуд (мВ)')
subplot(2,2,4); stem((0:fmax),pg); title('Фаза гармонік одного періоду сигналу');
set(gca, 'xtick', 0:1:fmax)
set(gca, 'ytick', -1:0.25:1)
xlabel('Частоти гармонік (Гц)')
ylabel('Фази (частина от pi)')
```

Амплітудний спектр вказує на амплітуди сигналів, які надходять до складу сумарного сигналу, зокрема постійної частини (нульової гармоніки). Амплітуди завжди позитивні.

Фазовий спектр вказує на фазовий зсув сигналів, які надходять до складу сумарного сигналу, для випадку їх косинусоїдального подання. Таким чином, фаза кожної гармоніки складного сигналу (у косинусоїдальній формі подання) дорівнюється із фазою косинусоїди відповідної частоти, але при нульовому фазовому зсуві.

Визначити значення дійсної та мнімої частин спектра по наявним комплексним коефіцієнтам (спектральної щільності) можливо наступним чином [5,9]:

% ---- Перевірка відповідності результатів:

rg(2:(fmax+1))=2*real(YdFs(2:(fmax+1)))/N; % Дійсна частина гармонік

rg(1)=real(YdFs(1))/N; % Дійсна частина нульової гармоніки

ig(2:(fmax+1))=2*imag(YdFs(2:(fmax+1)))/N; % Мніма частина гармонік

ig(1)=imag(YdFs(1))/N; % Мніма частина нульової гармоніки

Проконтролювати достовірність перетворень можливо розрахунком значень амплітуд та фаз згідно отриманим дійсної та мнімої частин спектру.

%-- Перевірка:

cm=sqrt(rg.^2+ig.^2); % Модуль амплітуд гармонік

ph=(atan2(ig,rg))/pi; % Фаза гармонік

%-----

Рівень похибки вимірюється нормою різності векторів [20]:

delta1=norm(sm-mg)

delta2=norm(ph-pg).

Графічно результати пояснюються на рис.2.1.

Зміна складу сигналу, параметрів його гармонік, зсуву, масштабування й інші дії приводять до відповідних змін спектру. MATLAB надає можливості для моделювання багаточисельних варіантів перетворень [14].

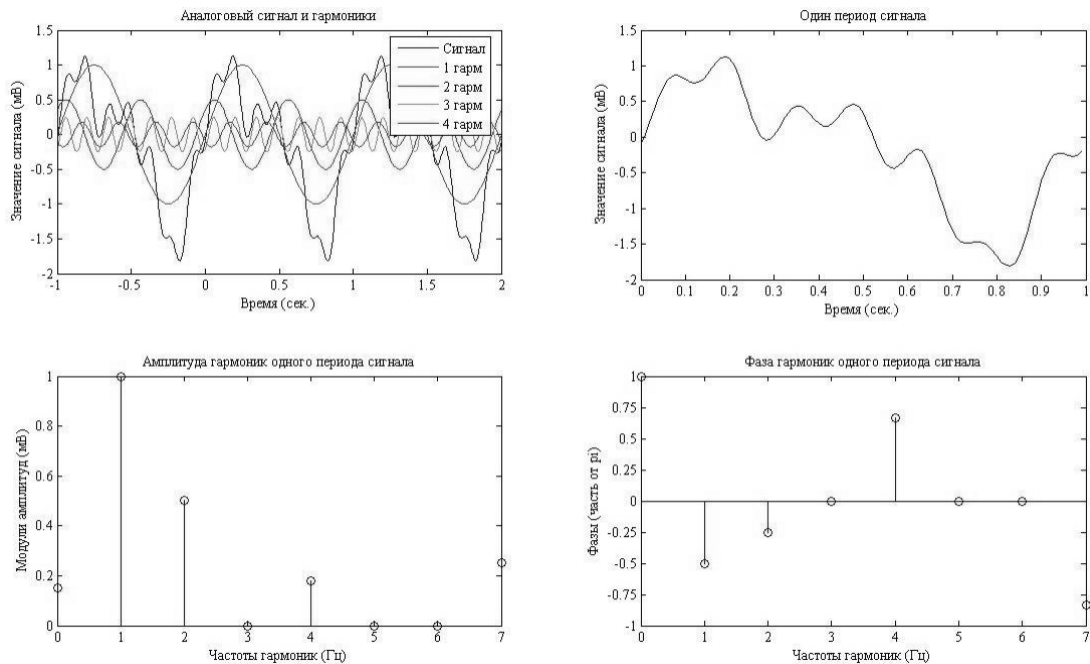


Рис.2.1 Безперервний сигнал, амплітудний та фазовий спектри [5]

2. СПЕКТР ЗВУКОВОГО СИГНАЛА

Аналогічні дії по відношенню до неперіодичного, наприклад звукового сигналу, реалізуються наступною послідовністю команд [5]:

```
y = wavread('c:\windows\media\tada'); N=size(y);
y1=y(:, 1);
y2=y(:, 2);
% Виділення частини сигналу
N1=101;
N2=150;
NF=N2-N1+1;
y11=roundn(y1(N1:N2),-4);
subplot(4,1,1); plot(y11);
title(['Звуковий сигнал «tada» (відліки ',int2str(N1),'...',int2str(N2),')']);
% Спектр каналу 1
YdFs=roundn(fft(y11,NF),-3);
rdF=real(YdFs);
idF=imag(YdFs);
mdF = abs(YdFs);
pdF = angle(YdFs)/pi;
subplot(4,1,2); stem(mdF); title('Амплітуда');
subplot(4,1,3); stem(pdF); title('Фаза');
set(gca, 'ytick', -1:0.25:1)
```

```

% ----- Відновлення -----
yi11=roundn(iff(YdFs),-4);
subplot(4,1,4); plot(yi11);
title(['Відновлення сигналу «tada» (відліки ', int2str(N1), '...', int2str(N2), ')']);
% ---- Похибка -----
delta=norm(y11-yi11)
% -----

```

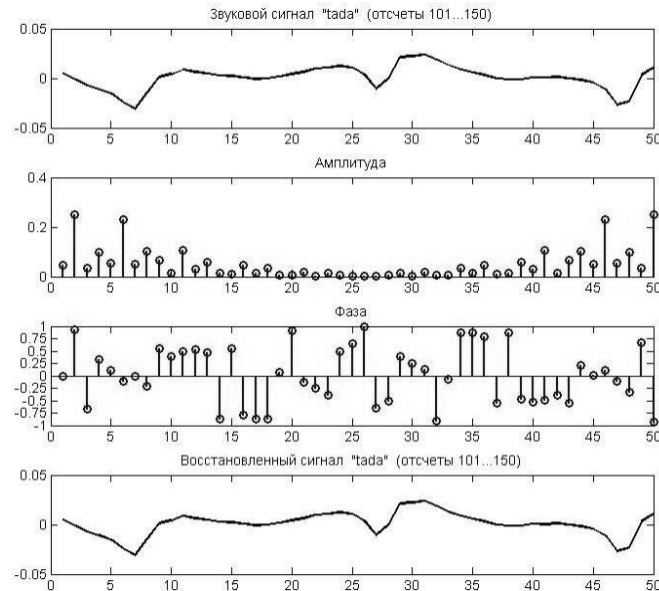


Рис.2.2 Фрагмент звукового начального та відтвореного сигналу “tada”, амплітудний та фазовий спектри [5]

3. ЗАВДАННЯ ДО ПРАКТИЧНОЇ РОБОТИ

1. Згідно вказаної вище методики сформувати спектр сигналу із заданими параметрами (табл.2.1), взявши форму гармонік із табл.2.2.
2. Змінити значення амплітуд A_0 та A_1 . Повторити формування спектра. Ви-значити відмінності.
3. Змінити форми гармонік згідно заданому варіанту (табл.2.2). Повторити формування спектра. Визначити відмінності.
4. Зсунути початковий сигнал вліво згідно заданому варіанту (табл.2.2). Сформувати спектри зсунутого сигналу. Визначити відмінності.
5. Зсунути початковий сигнал вправо згідно заданому варіанту (табл.2.2). Сформувати спектр. Визначити відмінності.
6. Сформувати спектр сигналу без 4-й гармоніки. Визначити відмінності.
7. Розробити програму відтворення сигналів. Показати приклад відтворення сигналів. Розрахувати похибку перетворення.

4. ВАРІАНТИ ЗАВДАНЬ.

Таблиця 2.1 [10]

№ вар	Частота, Гц				Амплітуда, мВ					Фаза, рад			
	f ₁	f ₂	f ₃	f ₄	A ₀	A ₁	A ₂	A ₃	A ₄	φ ₁	φ ₂	φ ₃	φ ₄
1	8	3	2	4	1,1	-1	0,7	0,35	0,5	π/3	-π/2	π/6	π/4
2	2	6	11	5	-0,9	0,8	1	0,25	-1,6	π/6	π/3	-π/9	-π/5
3	4	5	12	6	-0,5	0,7	-0,6	-0,3	1	π/2	-π/4	π/5	0
4	7	6	9	3	-0,4	0,6	1,1	-0,8	-1,3	-π/3	π/4	π/7	π/2
5	10	6	3	8	-0,6	1	-1,4	0,4	0,4	-π/4	0	-π/6	π/3
6	4	9	8	6	0,6	-0,9	1	0,7	1,3	π/5	-π/2	π/9	π/4
7	9	11	3	5	-0,5	0,6	-1,6	1,1	1	π/2	-π/3	π/4	π/5
8	6	3	9	7	0,22	-0,5	1,1	-1,2	-0,8	-π/2	π/4	-π/6	0
9	3	12	6	8	-0,12	1	-1,3	-0,9	-0,4	-π/3	π/5	π/5	0
10	14	11	7	5	-0,13	0,6	-1	0,34	0,9	π/4	0	-π/7	-π/3
11	9	10	2	6	0,41	-0,5	1,1	0,12	1	π/5	-π/2	π/4	0
12	9	7	5	10	-0,5	0,4	-1,5	-0,1	1,3	0	-π/3	-π/9	π/4
13	2	13	5	8	-0,9	1	-0,6	0,6	-1,2	-π/2	π/4	-π/3	0
14	8	10	3	5	0,3	-0,8	-1	0,5	1,4	-π/3	0	π/4	-π/2
15	7	3	2	6	0,3	-0,6	1,4	0,4	1	-π/4	π/3	π/5	π/3
16	9	4	6	7	-0,4	0,3	-1,3	-0,3	1,9	π/4	π/4	-π/6	-π/2
17	10	5	11	3	-0,12	1	-0,5	0,55	-1,3	0	π/1	π/7	-π/3
18	2	6	7	4	-0,13	1,2	-0,4	0,7	-1	π/4	π/2	-π/8	-π/2
19	11	6	5	4	-0,15	0,4	-0,6	0,8	1,5	π/8	0	π/9	π/3
20	10	5	8	7	0,34	-0,9	-0,5	0,44	1,2	π/7	-π/2	-π/9	-π/4
21	10	2	4	8	0,35	-1,1	1	-0,3	0,6	π/2	π/6	-π/8	-π/5
22	6	11	3	5	0,63	-0,7	-0,5	0,4	1	π/4	-π/3	π/7	0
23	7	10	6	8	-0,4	0,8	0,3	-0,5	-1,1	0	-π/4	-π/6	-π/2
24	4	9	5	7	0,6	-0,4	0,7	-0,3	1,4	π/2	π/5	π/5	-π/3
25	8	6	2	12	-0,7	0,6	-1	0,4	-1,3	-π/5	π/2	-π/5	π/4
26	5	10	7	9	-0,1	1,30	0,7	0,6	-1	π/3	0	-π/6	-π/5
27	6	11	3	5	0,1	-0,3	-0,6	0,7	1,2	π/2	-π/4	-π/7	π/2
28	6	10	3	7	-0,2	1,5	-0,5	0,8	1,4	-π/3	0	π/8	π/4
29	5	11	2	8	-0,3	0,6	1	0,9	-1,6	-π/2	0	-π/9	π/3
30	7	12	4	5	0,6	-0,3	-0,8	1,1	1	0	-π/2	π/9	π/4

Таблиця 2.2 [10]

№ вар	Форма гармонік				Змінити гармоніки	Кількість періодів сигналу	Такти зсуву		
	1	2	3	4			Кількість зсувів	вліво	вправо
1	sin	cos	sin	cos	1 і 2	3	4	8	13
2	sin	sin	sin	cos	1 і 4	4	5	5	14
3	cos	cos	sin	sin	1 і 3	5	3	10	13
4	cos	cos	cos	sin	1 і 4	5	3	11	8
5	cos	sin	cos	cos	3 і 2	4	5	7	12
6	cos	cos	sin	sin	3 і 2	3	4	14	8
7	sin	cos	cos	sin	1 і 2	3	4	12	9
8	sin	sin	sin	cos	1 і 4	4	5	4	10
9	sin	sin	cos	sin	3 і 4	5	3	7	16
10	sin	sin	cos	cos	2 і 3	5	3	9	12
11	sin	cos	cos	cos	1 і 4	4	5	5	10
12	cos	sin	sin	sin	3 і 2	3	4	7	12
13	cos	sin	sin	cos	3 і 4	3	4	15	13
14	sin	cos	cos	sin	1 і 2	4	5	8	10
15	sin	cos	sin	cos	1 і 4	5	3	14	9
16	cos	sin	cos	sin	3 і 4	5	3	12	8
17	cos	cos	sin	sin	2 і 3	4	5	4	15
18	sin	sin	cos	sin	1 і 3	3	4	12	10
19	sin	sin	cos	cos	3 і 2	3	4	6	9
20	sin	cos	cos	cos	1 і 4	4	5	7	13
21	cos	sin	sin	sin	1 і 2	5	3	13	11
22	cos	sin	sin	cos	3 і 4	5	3	10	8
23	sin	cos	cos	sin	2 і 4	4	5	4	7
24	sin	cos	sin	cos	2 і 3	3	4	8	9
25	cos	sin	cos	sin	1 і 4	3	4	15	9
26	cos	cos	sin	sin	3 і 2	4	5	10	16
27	sin	sin	cos	sin	1 і 3	5	3	9	17
28	sin	sin	cos	cos	1 і 4	5	3	16	18
29	sin	cos	cos	cos	1 і 3	4	5	11	9
30	cos	sin	sin	sin	1 і 2	3	4	14	10

5. ЗМІСТ ЗВІТУ З ПРАКТИЧНОЇ РОБОТИ

1. Формулювання завдання.
2. 2. Текст програми.
3. Результати виконання всіх розділів завдання.
4. Файл із результатами виконання завдання.
5. Пояснення отриманих результатів.
6. 6. Відповіді на контрольні питання.

6. КОНТРОЛЬНІ ПИТАННЯ

1. З якою метою застосовується псевдо безперервний сигнал.
2. Параметри псевдо безперервного сигналу.
3. Алгоритм формування амплітудного та фазового спектру.
4. Порядок визначення значення дійсної та мнімої частин спектра.
5. Обґрунтуйте відмінність між спектрами періодичного та неперіодичного сигналів.
6. Охарактеризуйте як змінюється спектр при зсуві сигналу.
7. Охарактеризуйте як змінюється спектр при масштабуванні сигналу.
8. Охарактеризуйте як змінюється спектр при зміні складу сигналу.

З навчальними матеріалами з питань дослідження перетворення Фур'є безперервних сигналів в середовищі системи MATLAB можна детальніше ознайомитись також в [1 (с.122-129), 2 (с.454-459), 3].

ПЕРВИННІ ПЕРЕТВОРЕННЯ ГРАФІЧНИХ ЗОБРАЖЕНЬ [5]

Мета практичної роботи - дослідження методів первинного перетворення монохромних зображень у середовищі системи MATLAB.

1. ЗАСОБИ ПІКСЕЛЬНИХ ПЕРЕТВОРЕНЬ СИСТЕМИ MATLAB

1.1 Зміна яскравості пікселів

У найпростішому варіанті перетворення зображень має розмір 1×1 , тобто складається з одного центрального пікселя. У такому випадку значення яскравості пікселя у вихідному зображенні залежить тільки від значення яскравості відповідного пікселя вхідного зображення, що стосовно до монохромних зображень припускає градаційне перетворення [5].

Для перетворення яскравості напівтонових зображень у системі MATLAB використовується функція `imadjust`:

$$g = \text{imadjust}(f, [\text{low_in} \ \text{high_in}], [\text{low_out} \ \text{high_out}], \text{gamma})$$

Функція перетворить значення яскравості з інтервалу $[\text{low_in}, \text{high_in}]$ в інтервал $[\text{low_out}, \text{high_out}]$. Значення, менші граничного рівня low_in , прирівнюються значенню low_out , а більші граничного рівня high_in - прирівнюються значенню high_out . Вхідне зображення може мати клас `uint8`, `uint16` або `double`, а клас вихідного збігається із класом вхідного [9].

Всі вхідні параметри функції `imadjust`, за винятком f , повинні бути речовинними числами в інтервалі від 0 до 1, незалежно від класу f . Якщо f належить класу `uint8` (або `uint16`), то функція `imadjust` множить ці параметри на 255 (або 65535). [14]

Якщо замість вектора зазначений порожній вектор, то використовуються величини за замовчуванням, рівні $[0 \ 1]$.

Якщо high_out менше, ніж low_out , то вихідні яскравості симетрично перевертаються.

Параметр *gamma* служить для вказівки закону перетворення (рис.2.1а,б,в).

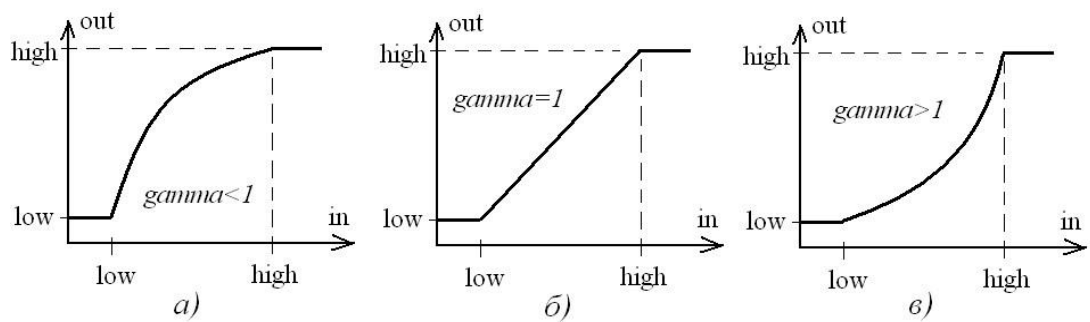


Рис.2.1 Варіанти перетворень у функції *imadjust* [10]

Наступні команди формують негативне зображення (рис.2.2):

```
f=imread('moon.tif');
imshow(f, [ ]) % Позитив
fneg=imadjust(f, [0 1], [1 0]);
figure; imshow(fneg, [ ]) % Негатив
```

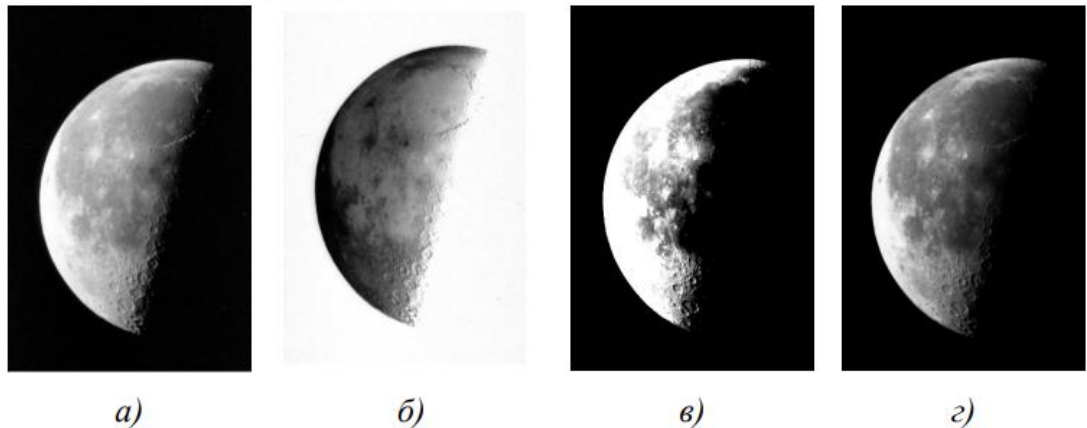


Рис.2.2 Перетворення зображення функцією *imadjust* [5]

На рис.1.2 в показаний результат виконання команди

g2=imadjust(f, [0.5 0.75], [1 0]);

На рис.1.2г показаний результат виконання команди

g3=imadjust(f, [], [], 2);

Негативне зображення формується також функцією *imcomplement*:

```
figure; imshow(imcomplement(f), [11])
```

1.2 Логарифмічне перетворення яскравості

За допомогою вираження [5]:

$$g = c * \log(1 + \text{double}(f))$$

де *c* — деяка константа, у системі MATLAB можливе виконання логарифмічних перетворень.

Закон перетворення аналогічний закону перетворення у функції `imadjust` при $\gamma < 1$, але із зафіксованою кривизною графіка та при граничних вихідних значеннях, рівних 0 і 1.[9]

Логарифмічне перетворення успішно застосовується при стиску динамічно-го діапазону з метою поліпшення візуалізації зображень. Наприклад, наступні ко-манди переводять зображення з діапазону $[0 \dots 1.5 \times 10^6]$ у зображення із діапазоном $[0 \dots 255]$ (рис.2.3) [5]:

```
g = im2uint8(mat2gray(log(1 + double(f))));  
imshow(g)
```

Функція `mat2gray` переводить величини в діапазон $[0,1]$, а функція `im2uint8` перетворить їх до діапазону $[0, 255]$.

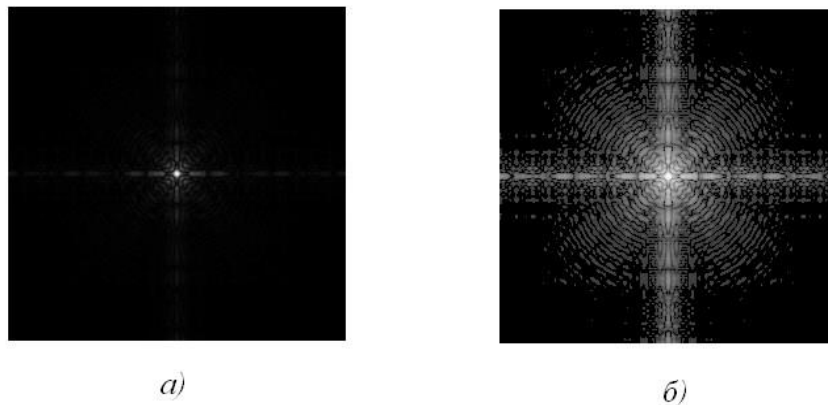


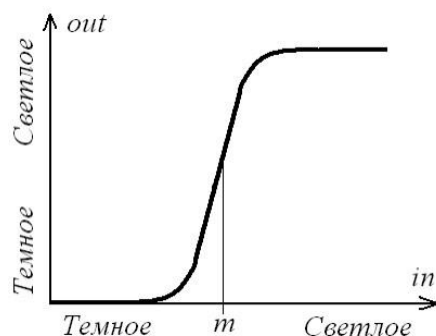
Рис.2.3 Зображення до й після логарифмічного перетворення [9]

1.3. Зміна контрастності

Команда

$$g = 1 ./ (1 + (m ./ (double(f) + eps)).^E)$$

дозволяє змінити контрастність зображення. У результаті виконання команди формується зображення із більшою контрастністю.

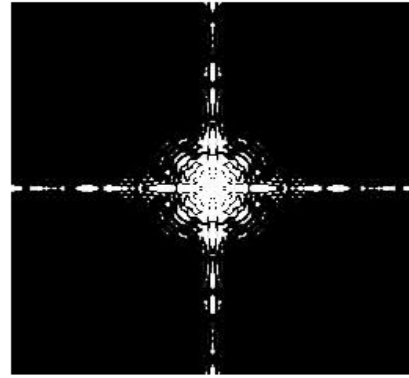


Команда стискає вхідні величини, менші m , у більше вузький піддіапазон темних рівнів на вихідному зображенні, і величини, більші m - у більш вузьку смугу яскравих рівнів (рис.2.4). Параметр E встановлює нахил функції, наприклад $E=20$ або 5 (рис.2.5). [8]

Рис.2.4 Закон перетворення контрастності



а) $m=180, E=5$



б) $m=10, E=20$

Рис.2.5 Приклади перетворення контрастності [5]

2. ЕКВАЛІЗАЦІЯ ГІСТОГРАМ

Результат еквалізації зображення полягає в збільшенні динамічного діапазону рівнів яскравості, що звичайно означає більшу контрастність вихідного зображення. У випадку дискретного подання даних методика зводиться до еквалізації гістограм зображень [4].

Нехай $p_r(r_j)$, $j=1, \dots, L$, позначає гістограму рівнів яскравості деякого вихідного зображення. Величини нормованої гістограми є наближеннями ймовірностей появи кожного рівня яскравості на зображенні. Перетворення еквалізації має вигляд [9]:

$$S_k = T(r_k) = \sum_{j=1}^k p_r(r_j) = \sum_{j=1}^k \frac{n_j}{n}$$

при $k=1, \dots, L$, де s_k – значення яскравості обробленого зображення, що відповідає значенню яскравості r_k вихідного зображення.

Еквалізація гістограм реалізується функцією `histeq` [14]:

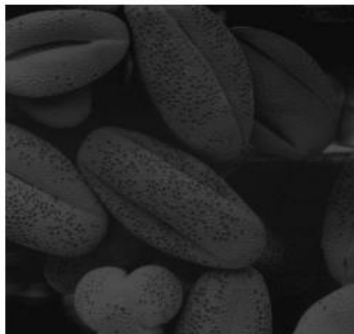
$g = \text{histeq}(f, \text{nlev})$,

де f – вихідне зображення, nlev – число рівнів інтенсивності, установлене для вихідного зображення. Якщо $\text{nlev}=L$, тобто загальному числу можливих рівнів вихідного зображення, то реалізується функція перетворення $T(r_k)$. Якщо $\text{nlev} < L$, то рівні перерозподіляються таким чином, щоб забезпечити плоску гістограму. За замовчуванням nlev дорівнює 64 [20].

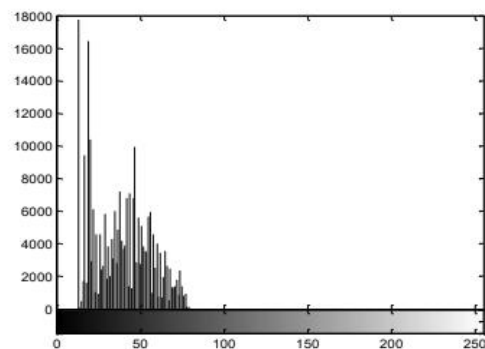
Проілюструємо це на прикладі пилка. На рис.2.6а представлена фотографія пилка, зроблена на електронному мікроскопі (збільшення приблизно в 700 разів), а на рис.2.6б - його гістограма. Зображення є дуже темним і має вузький динамічний діапазон значень яскравості (гістограма повністю розташована в темній ділянці спектра) [10].

Зображення на рис.2.6в є результат гістограмної еквалізації вихідного зображення. У результаті перетворення підвищилася середня яскравість і покращилася контрастність (рис.2.6г). Зміна контрастності викликана істотним розширенням динамічного діапазону на всю шкалу яскравості [5].

```
imshow(f)
figure, imhist(f)
ylim('auto')
g=histeq(f,256);
figure, imshow(g)
figure, imhist(g)
ylim('auto')
```



а)

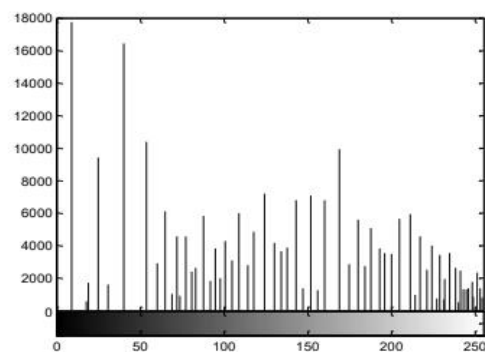


б)

Рис.1.6 а,б Приклад еквалізації гістограми



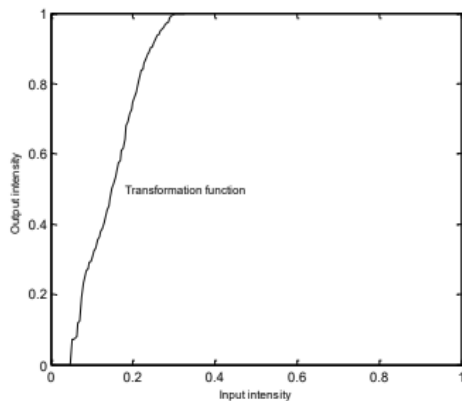
в)



г)

Рис.2.6 в,г Приклад еквалізації гістограми [5]

Вид функції перетворення попереднього зображення (наведено на рис.2.7), отриманий при виконанні послідовності наступних команд [7]:



```
hnorm=imhist(f)./numel(f);  
sdf=cumsum(hnorm);  
x=linspace(0, 1, 256);  
plot(x, sdf);  
axis([0 1 0 1]);  
set(gca, 'xtick', 0:.2:1)  
set(gca, 'ytick', 0:.2:1)  
xlabel('Input intensity', 'fontsize', 9)  
ylabel('Output intensity', 'fontsize', 9)  
text(0.18, 0.5, 'Transformation function',  
    'fontsize', 9)
```

Рис.2.7 Функція перетворення

У багатьох додатках необхідно одержати зображення із заданою формою гістограми. Такий процес називається гістограмним припасуванням або гістограмною специфікацією й реалізований у функції `histeq` із наступним варіантом синтаксису [4]:

```
g=histeq(f, hspec),
```

де `hspec` - задана гістограма (вектор-рядок), що складається із цілих чисел, які відповідають однаковим розбивкам діапазону рівнів.

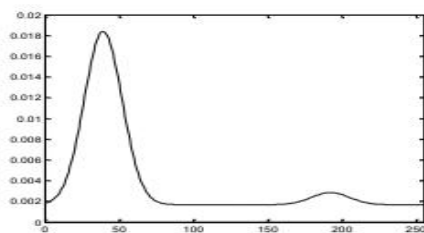
На рис.2.8а показане зображення супутника Марса Фобоса, а на рис.2.8б – його гістограма, отримана функцією `imhist(f)`. Обробка командою `histeq(f, 256)` істотно не поліпшує якість (рис.1.8в). Тому необхідно задати гістограму, що дозволяє якісно поліпшити зображення [5].

Для даного прикладу гістограма повинна мати меншу концентрацію компонентів у нижній частині градаційної шкали, але при цьому зберігати загальну форму гістограми вихідного зображення, яка має дві чітко виділені активних ділянки - біля нульового значення та у верх-ній частині градаційної шкали [10].

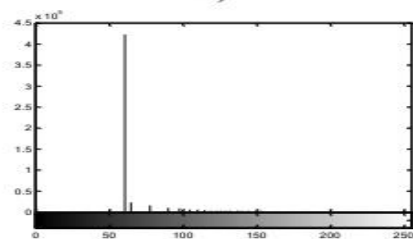
Такий тип гістограм можна змоделювати за допомогою нормованих мультимодальних гаусових функцій. У кожному разі, спочатку завдання зводиться до формування вектор-рядка `hspec` [14].

0.0018	0.0018	0.0018	0.0018	0.0018	0.0019	0.0019	0.0019	0.0020
0.0020	0.0021	0.0021	0.0021	0.0022	0.0022	0.0023	0.0024	0.0024
0.0025	0.0025	0.0026	0.0026	0.0027	0.0027	0.0028	0.0028	0.0028
0.0028	0.0028	0.0028	0.0028	0.0028	0.0028	0.0028	0.0028	0.0027
0.0027	0.0026	0.0026	0.0025	0.0025	0.0024	0.0024	0.0023	0.0023
0.0022	0.0022	0.0021	0.0021	0.0020	0.0020	0.0019	0.0019	0.0019
0.0019	0.0018	0.0018	0.0018	0.0018	0.0018	0.0017	0.0017	0.0017
0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017
0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017
0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017
0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017

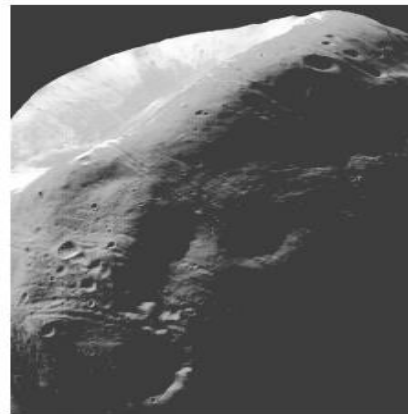
Результат перетворення наведено на рис.2.9б, а гістограму - на рис.2.9в.



а)



в)



б)

Рис.2.9 Еквалізація за заданою гістограмою [10]

3. ЗАВДАННЯ ДО ПРАКТИЧНОЇ РОБОТИ

1. Визначити характеристики заданого зображення.
2. Зрівняти характеристики і встановити особливості подання зображення в повнокольоровому, палітровому, напівтоновому і бінарному варіантах. Реалізувати декілька варіантів формування палітрового зображення із використанням різних колірних карт і параметрів перетворення.
3. Показати процедуру виділення окремих шарів повнокольорового зображення, їхньої роздільної обробки (зміну формату даних) і компонування в єдине зображення.

Примітка: Перед виконанням завдань п.1-3 рекомендується засвоїти відповідний тематичний матеріал згідно «Методичних вказівок для самостійного вивчення».

4. Стосовно заданого зображення виконати наступні перетворення:

- яскравості (функцією `imadjust` при різних значеннях `gamma`),
- у негативне зображення,
- контрастності (при різних параметрах).

Пояснити результати перетворень, використовуючи для цього характерні фрагменти зображення, представлені у вигляді вихідної та результуючої матриць.

5. Перетворити зображення методом еквалізації гістограми.

4. ЗМІСТ ЗВІТУ З ПРАКТИЧНОЇ РОБОТИ

1. Формулювання завдання.
2. Текст програми.
3. Результати виконання всіх розділів завдання.
4. Файл із результатами виконання завдання.
5. Пояснення отриманих результатів. 6. Відповіді на контрольні питання.

5. КОНТРОЛЬНІ ПИТАННЯ

1. Призначення та застосування функції `imadjust`.
2. Параметри функції `imadjust`.
3. Охарактеризуйте переваги та недоліки присутні гамма-перетворенню.
4. Охарактеризуйте переваги та недоліки присутні логарифмічному перетворенню.
5. Яким чином можна змінити контрастність зображення.
6. Обґрунтуйте необхідність застосування еквалізації гістограм.
7. Методи керування динамічним діапазоном.
8. Дайте пояснення поняттю гістограмного припасування.

З навчальними матеріалами з питань дослідження первинного перетворення графічних зображень в середовищі системи MATLAB можна детальніше ознайомитись також в [1(с.79-103), 3].

Практична робота № 3.

ДИСКРЕТИЗАЦІЯ СИГНАЛІВ [11]

Мета практичної роботи — дослідження методики перетворення сигналів до дискретного виду в середовищі системи MATLAB.

1. СПЕКТР БЕЗПЕРЕРВНОГО СИГНАЛА

Дискретний еквівалент безперервного сигналу в середовищі системи MATLAB формується наступним чином (рис.3.1а,б,в) [11]:

```
clear all
A=[1 0.5 -0.25 -0.18]; %Амплітуди гармонік
F=[7 2 5 4]; % Частоти гармонік в Герцах
fmax=max(F);
Phi=[-pi/2 pi/4 pi/6 -pi/3]; % Фази гармонік
s0=-0.15;
% Генерація сигналу:
Fd=1000; % Псевдо безперервність (частота дискретизації 1кГц)
t=0:1/Fd:(1-1/Fd);
    s=s0;
    for k=1:length(F)
        s=s+A(k).*cos(2*pi*F(k).*t+Phi(k));
    end
subplot(5,2,1); plot(t,s); title('Псевдо безперервний сигнал');
xlabel('Время (сек.)')
ylabel('Значення сигналу (мВ)')
subplot(5,2,2); stem(s(1:1000)); title('Псевдо безперервний сигнал (1 сек)');
xlabel('Номер відліку')
subplot(5,2,3); stem(s(1:200)); title('Псевдо безперервний сигнал (на інтервалі 0,2 сек)');
xlabel('Номер відліку')
```

Псевдо безперервність сигналу в середовищі системи MATLAB підтверджується дискретністю параметра моментів часу. Тому для переводу «безперервного» сигналу до дискретної форми достатньо збільшити інтервал вибірки відліків (шагу дискретизації), тобто визначити частоту семплування, менше за частоту начального сигналу. Процес встановлення зниженої частоти дискретизації називається децимацією й є різновидом передискретизації сигналу [5].

Для зниження частоти сигналу в середовищі системи MATLAB використовуються функції `downsample` та `decimate` [9].

Функція $y = \text{downsample}(x, n)$ зменшує частоту дискретизації сигналу x (вектор або матриця) вибором кожного n -го відліку, починаючи з першого. При матричному поданні сигнал оброблюється за стовпцями. Додатковий параметр phase (ціле число) дозволяє задати затримку вибірки першого відліку: $y = \text{downsample}(x, n, \text{phase})$ [14].

Наприклад, зменшення частоти дискретизації в три рази:

```
x=[1 2 3 4 5 6 7 8 9 10 11 12];  
y1=downsample(x,3)  
y2=downsample(x,3,2)
```

Результат:

```
y1= 1 4 7 10  
y2= 3 6 9 12
```

Дискретизація сигналу, що сформований раніше на частотах 100Гц, 25Гц, 10Гц, відповідає результатам, які показано на рис.1г,д,е. Дискретизація у всіх випадках супроводжується децимацією в 10, 40, 100 раз відповідно, тому що частота дискретного подання начального сигналу дорівнює 1кГц [9].

```
sd100=downsample(s,10); %Дискретизація (децимація) на частоті 100Гц  
subplot(5,2,4); stem(sd100(1:20)); title('Дискретний сигнал (fd=100Гц) (інтервал 0,2 сек)');  
xlabel('Номер відліку')  
sd25=downsample(s,40); % Дискретизація (децимація) на частоті 25Гц  
subplot(5,2,5); stem(sd25(1:25)); title('Дискретний сигнал (fd=25Гц) (інтервал 1 сек)');  
xlabel('Номер відліку')  
sd10=downsample(s,100); % Дискретизація (децимація) на частоті 10Гц  
subplot(5,2,6); stem(sd10(1:10)); title('Дискретний сигнал (fd=10Гц) (інтервал 1 сек)');  
xlabel('Номер відліку')
```

При децимації необхідно дотримуватись положень теореми Котельникова, тобто перед вказанням нової частоти дискретизації необхідно перевірити, що най-велика частота в спектрі сигналу не перевищує половини частоти дискретизації (частоти Найквіста). Якщо ця вимога не виконується, то необхідно попередньо обмежити спектр сигналу за допомогою низькочастотної фільтрації. В іншому ви-падку, за результатом децимації буде перекриття частот, тобто аліасинг, та, як наслідок, сигнал буде відтворено невірно [7,4].

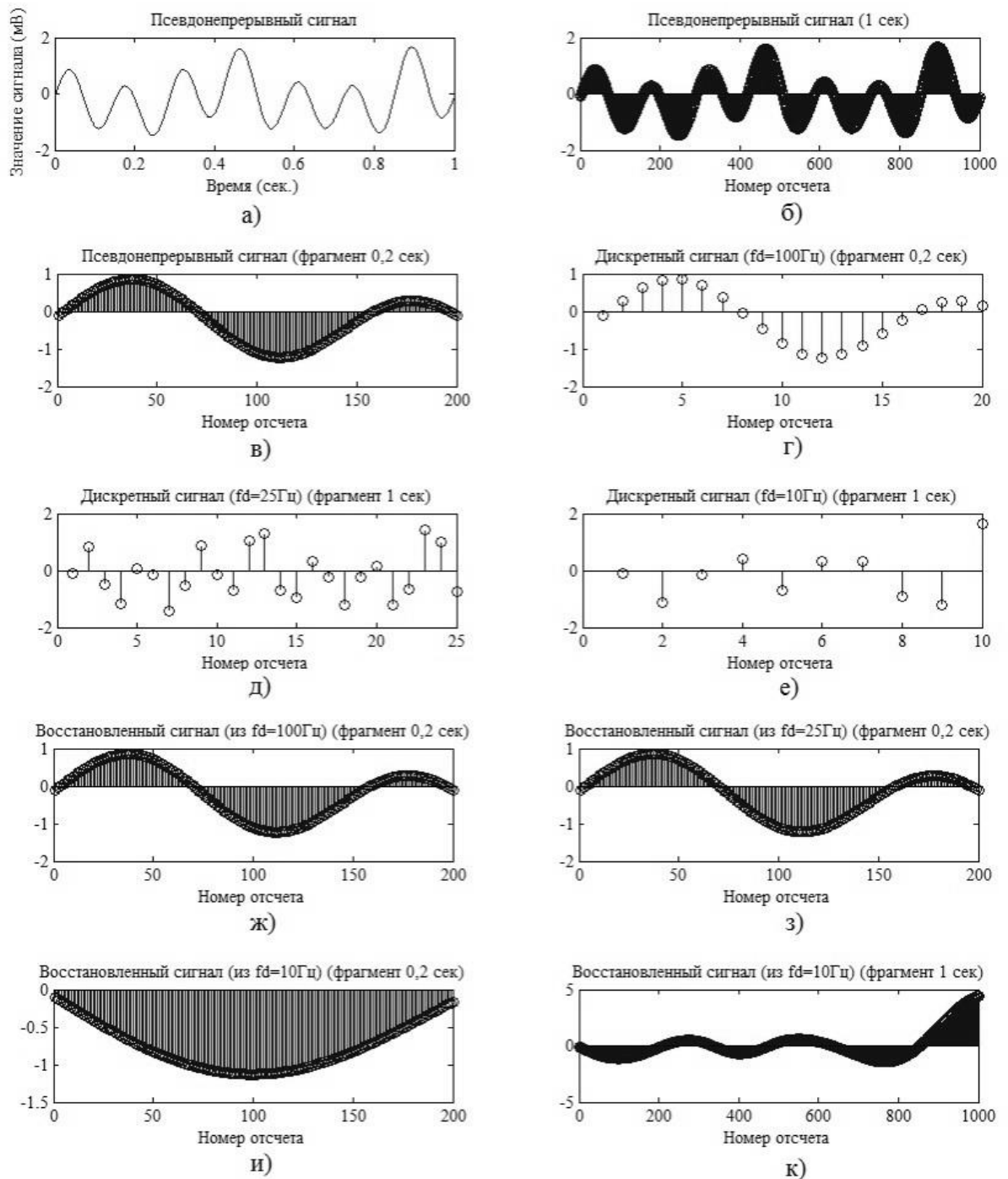


Рис.3.1 Начальный сигнал (а,б,в), сигналы після дискретизації (г,д,е), сигналы після вірного відтворення (ж,з), сигналы після помилкового відтворення (и,к) [11]

На рис.3.1ж,з показано результат процесу відтворення сигналу по дискретним відлікам, які взято на частотах 100Гц і 25Гц відповідно. Результат отримано вірно, тому що при дискретизації дотримання вимоги Котельникова (Найквіста) забезпечено (максимальна частота в початковому сигналі цього прикладу дорівнює 7Гц) [11].

Відтворення сигналу по відлікам, які взято на частоті 10Гц, завершується помилкою, тому що частота дискретизації не перевищує як мінімум вдвічі максимальну частоту в сигналі [11].

У всіх випадках, для відтворення використана функція інтерполяції $\text{interp}(x,r)$, де x – дискретні відліки, r – ціле чисельний коефіцієнт підвищення частоти [5].

В додаток до децимації, в функції $\text{decimate}(x,r)$ виконується попередня низькочастотна фільтрація, де r - коефіцієнт зниження частоти дискретизації. За замовчуванням використовується фільтр Чебишева. Однак, ігнорування умови Котельникова (Найквіста) також створює передумову для відтворення невірної частоти, але іншої форми [9].

Для передискретизації застосовується також функція $\text{resample}(x, p, q)$, де p і q – чисельник та знаменник дробного коефіцієнта [14].

2. ЗАВДАННЯ ДО ПРАКТИЧНОЇ РОБОТИ

1. Сформулювати сигнал із заданими параметрами (табл.3.1).
2. Дискретизувати сигнал на заданих частотах та мінімально допустимій частоті із використанням функцій downsample та decimate .
3. Відтворити сигнал по дискретним відлікам із використанням функції interp .
4. Зіставити візуально й оцінити математично навчальний та відтворений сигнали. Знайти випадки помилкового відтворення.
5. Сформулювати спектр початкового і помилково відтворених сигналів (згідно навичок виконання попередньої практичної роботи). Визначити та пояснити відмінності.

3. ЗМІСТ ЗВІТУ З ПРАКТИЧНОЇ РОБОТИ

1. Формулювання завдання.
2. Текст програми.
3. Результати виконання всіх розділів завдання.
4. Файл із результатами виконання завдання.
5. Пояснення отриманих результатів.
6. Відповіді на контрольні питання.

4. ВАРІАНТИ ЗАВДАНЬ

Таблиця 3.1 [9]

№ вар	Частота, Гц			Амплітуда, мВ				Фаза, рад			Частота дискретизації, Гц		
	f_1	f_2	f_3	A_0	A_1	A_2	A_3	φ_1	φ_2	φ_3	f_{s1}	f_{s2}	f_{s3}
1	7	3	2	1,1	-1	0,7	0,35	$\pi/3$	$-\pi/2$	$\pi/6$	120	45	5
2	2	6	11	-0,9	0,8	1	0,25	$\pi/6$	$\pi/3$	$-\pi/9$	120	48	10
3	4	5	12	-0,5	0,7	-0,6	-0,3	$\pi/2$	$-\pi/4$	$\pi/5$	120	80	10
4	7	6	14	-0,4	0,6	1,1	-0,8	$-\pi/3$	$\pi/4$	$\pi/7$	120	90	20
5	10	6	3	-0,6	1	-1,4	0,4	$-\pi/4$	0	$-\pi/6$	120	96	10
6	4	9	8	0,6	-0,9	1	0,7	$\pi/5$	$-\pi/2$	$\pi/9$	120	96	5
7	9	11	3	-0,5	0,6	-1,6	1,1	$\pi/2$	$-\pi/3$	$\pi/4$	120	90	10
8	6	3	9	0,22	-0,5	1,1	-1,2	$-\pi/2$	$\pi/4$	$-\pi/6$	120	80	10
9	3	12	6	-0,12	1	-1,3	-0,9	$-\pi/3$	$\pi/5$	$\pi/5$	120	48	20
10	14	11	7	-0,13	0,6	-1	0,34	$\pi/4$	0	$-\pi/7$	120	45	25
11	9	10	2	0,41	-0,5	1,1	0,12	$\pi/5$	$-\pi/2$	$\pi/4$	120	45	10
12	9	7	5	-0,5	0,4	-1,5	-0,1	0	$-\pi/3$	$-\pi/9$	120	48	5
13	2	15	5	-0,9	1	-0,6	0,6	$-\pi/2$	$\pi/4$	$-\pi/3$	120	80	25
14	8	12	3	0,3	-0,8	-1	0,5	$-\pi/3$	0	$\pi/4$	120	90	20
15	7	3	2	0,3	-0,6	1,4	0,4	$-\pi/4$	$\pi/3$	$\pi/5$	120	96	10
16	9	4	6	-0,4	0,3	-1,3	-0,3	$\pi/4$	$\pi/4$	$-\pi/6$	120	96	5
17	10	5	11	-0,12	1	-0,5	0,55	0	$\pi/1$	$\pi/7$	120	90	10
18	2	6	7	-0,13	1,2	-0,4	0,7	$\pi/4$	$\pi/2$	$-\pi/8$	120	80	10
19	12	6	5	-0,15	0,4	-0,6	0,8	$\pi/8$	0	$\pi/9$	120	48	20
20	9	5	8	0,34	-0,9	-0,5	0,44	$\pi/7$	$-\pi/2$	$-\pi/9$	120	45	10
21	10	2	4	0,35	-1,1	1	-0,3	$\pi/2$	$\pi/6$	$-\pi/8$	120	45	10
22	6	12	3	0,63	-0,7	-0,5	0,4	$\pi/4$	$-\pi/3$	$\pi/7$	120	48	20
23	7	13	6	-0,4	0,8	0,3	-0,5	0	$-\pi/4$	$-\pi/6$	120	80	25
24	4	9	5	0,6	-0,4	0,7	-0,3	$\pi/2$	$\pi/5$	$\pi/5$	120	90	10
25	7	6	2	-0,7	0,6	-1	0,4	$-\pi/5$	$\pi/2$	$-\pi/5$	120	45	5
26	5	9	7	-0,1	1,30	0,7	0,6	$\pi/3$	0	$-\pi/6$	120	80	10
27	6	11	3	0,1	-0,3	-0,6	0,7	$\pi/2$	$-\pi/4$	$-\pi/7$	120	96	20
28	6	15	3	-0,2	1,5	-0,5	0,8	$-\pi/3$	0	$\pi/8$	120	90	25
29	5	16	2	-0,3	0,6	1	0,9	$-\pi/2$	0	$-\pi/9$	120	80	25
30	7	12	4	0,6	-0,3	-0,8	1,1	0	$-\pi/2$	$\pi/9$	120	96	20

5. КОНТРОЛЬНІ ПИТАННЯ

1. Опишіть сутність процедури децимації сигналів.
2. В яких випадках використовуються функції `downsample` та `decimate`.
3. Як при децимації забезпечити виконання критерія Найквіста.
4. Охарактеризуйте поняття «аліасинг».
5. Обґрунтуйте відмінності між функціями `downsample` та `decimate`.
6. Як змінюється відтворений сигнал при різних параметрах децимації.
7. Порядок використання функції `resample`.
8. Охарактеризуйте зміни спектрів почального та відтвореного сигналів при різних параметрах децимації.

З навчальними матеріалами з питань дискретизації сигналів в середовищі системи MATLAB можна детальніше ознайомитись в [2 (с.460-464), 3].

ІНТЕРПОЛЯЦІЯ ДИСКРЕТНИХ СИГНАЛІВ [9]

Мета практичної роботи – дослідження в середовищі системи MATLAB методики інтерполяції дискретних сигналів.

1. ІНТЕРПОЛЯЦІЯ ЗАСОБАМИ СИСТЕМИ MATLAB

Функції інтерполяції:

$y_i = \text{interp1}(x, y, x_i, \text{method})$ – одновимірна поліноміальна інтерполяція за методом *method* для точок x_i по відомим значенням y в точках x .

Методи інтерполяції: «nearest» – по значенню найближчої сусідньої вибірки (поліномом нульового рівня), «linear» – лінійна (поліномом першого рівня), використовується за замовчуванням, «cubic» – кубічна (поліномом третього рівня), «spline» – кубічними сплайнами [2,7,8].

$ZI = \text{interp2}(X, Y, Z, XI, YI, \text{method})$ – двовимірна поліноміальна інтерполяція за методом *method* для нових точок, що задаються матрицями XI, YI , по відомим значенням Z в точках X, Y . Методи інтерполяції: «nearest» – по значенню най-ближчої сусідньої вибірки (поліномом нульового рівня), «linear» – білінійна (поліномом першого рівня), «cubic» – бікубічна (поліномом третього рівня), «spline» – кубічними сплайнами [7].

Для коректної роботи функції обов'язковим є монотонність X, Y (вони повинні постійно збільшуватись або постійно зменшуватись), що реалізуємо функцією *meshgrid*. В якості альтернативи допустимо задавати в якості параметрів вектор-строку XI та вектор-стовбець YI , но в відповідності із форматом, що формується функцією *meshgrid*. Кожний із методів інтерполяції автоматично відображає дані на рівномірно розподілену сітку. Якщо дані рівномірні спочатку, то обчислення прискорюються при наступному варіанту вказання методу – «*cubic» (із додатковою зірочкою) [6].

Варіанти використання функції *interp2* [14]:

$ZI = \text{interp2}(Z, XI, YI)$ – при цьому $X=1:n$ та $Y=1:m$, де $[m, n] = \text{size}(Z)$.

$ZI = \text{interp2}(Z, \text{ntimes})$ – інтерполяція рекурсивним методом із кількістю шагів, яке дорівнює *ntimes*.

Пояснимо інтерполяцію зображення методами на прикладі. Між відомими вибірками додатково розраховуються до 10 точок в обох напрямках. Зображення за результатом формується відповідно збільшеним (рис.4.1) [10].

```
clear all f=imread('roseg.tif');
g = imresize(f,0.05);

l=size(g,1); m=size(g,2);
[X,Y]=meshgrid(1:1:m, 1:1:l);
gd=im2double(g);
[XI,YI]=meshgrid(1:0.1:m, 1:0.1:l);
ZI1=interp2(X,Y,gd,XI,YI,'nearest');
ZI2=interp2(X,Y,gd,XI,YI,'linear');
ZI3=interp2(X,Y,gd,XI,YI,'cubic');
ZI4=interp2(X,Y,gd,XI,YI,'spline');

subplot(2,2,1); subimage(ZI1); title('За найближчим');
subplot(2,2,2); subimage(ZI2); title('Білінійна');
subplot(2,2,3); subimage(ZI3); title('Бікубічна');
subplot(2,2,4); subimage(ZI4); title('Сплайнова');
```

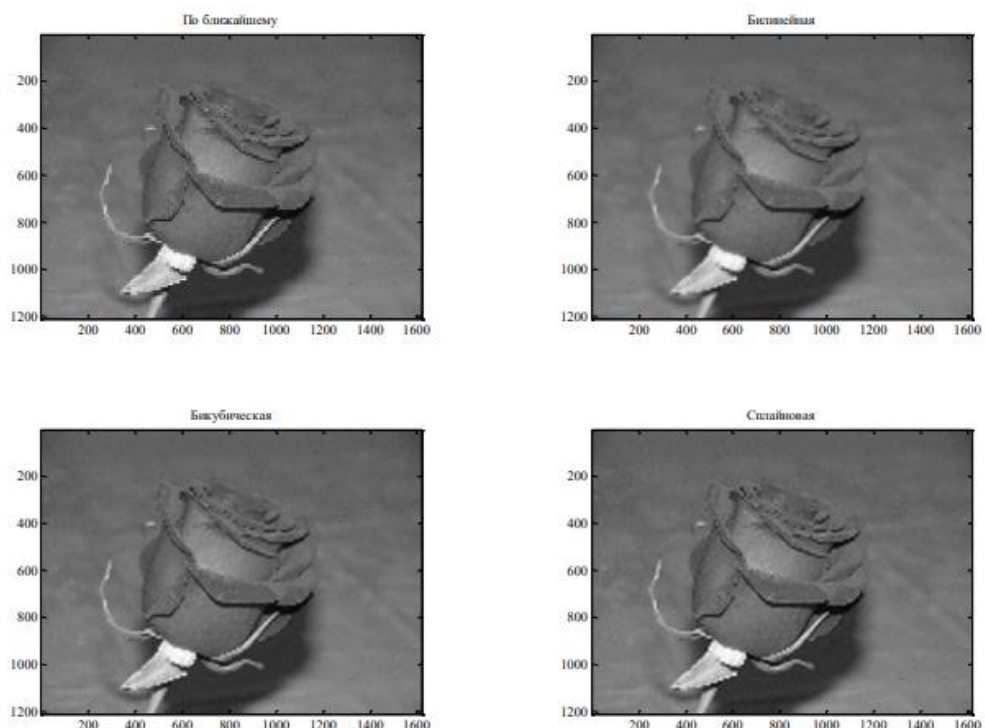


Рис.4.1 Приклад інтерполяції функцією interp2 [10]

$V = \text{interp3}(X,Y,Z,V,XI,YI,ZI,\text{method})$ – тримірна поліноміальна інтерполяція за методом `method` для нових точок, які задаються матрицями XI , YI , ZI , по відомим значенням V в точках X,Y,Z . [9]

Методи інтерполяції: 'nearest' – поліномом нульового рівня, 'linear' – поліномом першого рівня, 'cubic' – поліномом третього рівня, 'spline' – сплайнами. При рівномірному розподіленні даних для прискорення обчислень допустимо вказання методу із додатковою зірочкою - '*cubic' [6].

Варіанти використання функції *interp3*:

$Vl=interp3(V,Xl,Yl,Zl)$ – при цьому $X=1:N$, $Y=1:M$, $Z=1:P$, де $[M,N,P]=size(V)$.
 $Vl=interp3(V,ntimes)$ – інтерполяція рекурсивним методом із кількістю шагів, яке дорівнює $ntimes$. [9]:

- $Vl=interp_n(X1,X2,X3,...,V,Y1,Y2,Y3,...,method)$ – n -мірна поліноміальна інтерполяція за методом *method*.
- $yi=spline(x,y,xi)$ – кубічна сплайнова інтерполяція сигналу для точок xi за відомими y в точках x . Два варіанти застосування цієї функції показано нижче.

Отримані результати мають відмінності (рис.4.2) [11].

```
x=[2, 8, -3, 5, -7, 1, 3, -2];
t=1:length(x);
%-----
ys1 = spline(t,[0 x 0]);
tt = linspace(1,8,2807);
figure, plot(t,x,'o',tt,ppval(ys1,tt),'-');
%-----
ti=1:0.0025:length(x);
ys2 = spline(t,x,ti);
figure, plot(t,x,'o',ti,ys2,'-');
```

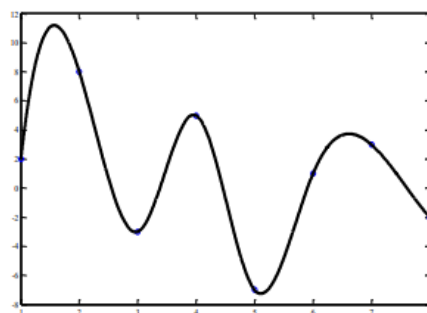
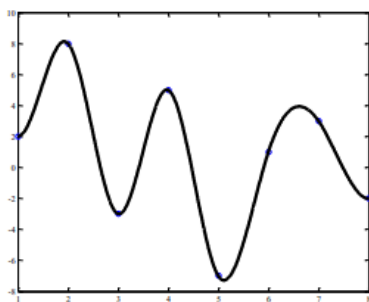


Рис.4.2 Приклади інтерполяції функцією *spline* [10]

Далі показано приклади інтерполяції при збільшенні зображення (рис.4.3) [10].

```
f=imread('roseg.tif');
g = imresize(f,0.05);
ZI1 = imresize(g,10,'nearest');
```

```
ZI2 = imresize(g,10,'bilinear');
```

```
ZI3 = imresize(g,10,'bicubic');
```

```
subplot(2,2,1); subimage(ZI1); title('За найближчим');
```

```
subplot(2,2,2); subimage(ZI2); title('Білінійна');
```

```
subplot(2,2,3); subimage(ZI3); title('Бікубічна');
```

```
subplot(2,2,4); subimage(g); title('Початкове');
```

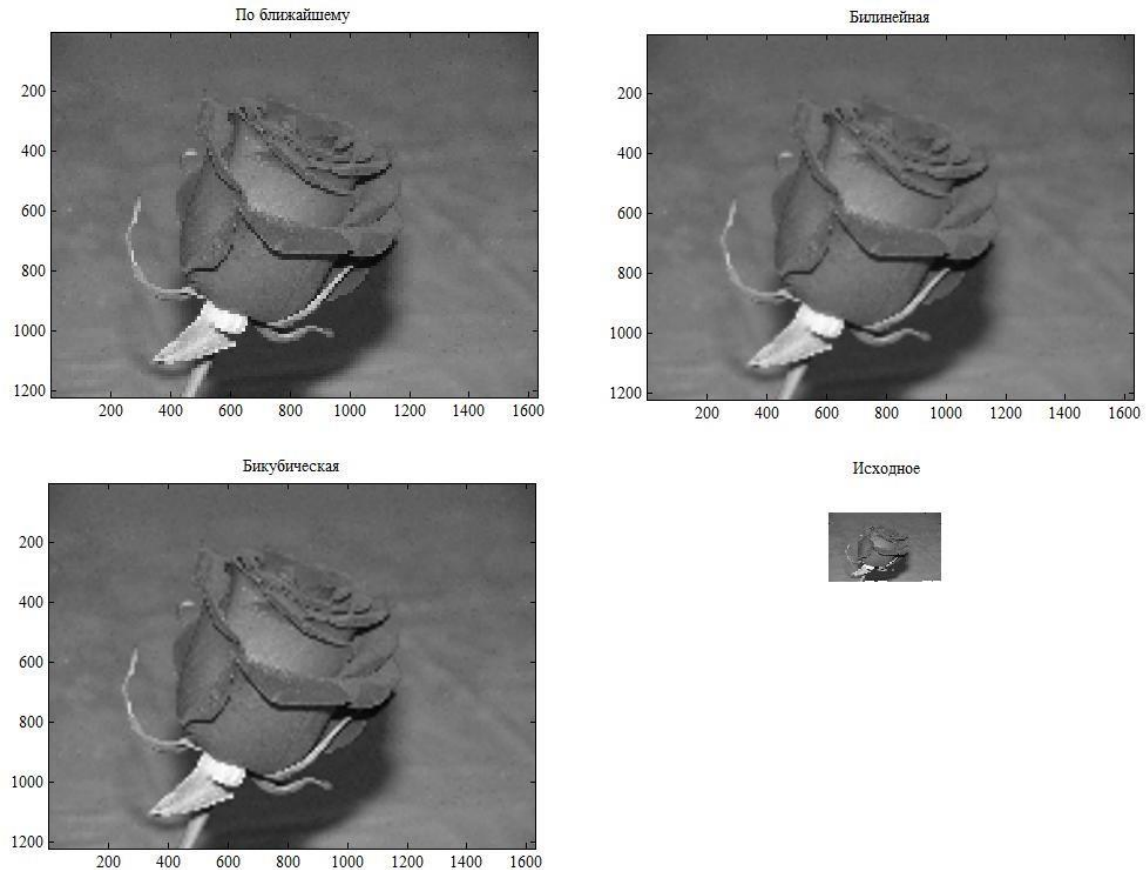


Рис.4.3 Приклад збільшення функцією `imresize` з інтерполяцією

2. ЗАВДАННЯ ДО ПРАКТИЧНОЇ РОБОТИ

Для одновимірного псевдо безперервного сигналу:

1. Сформувати дискретний сигнал.
2. Виконати інтерполяцію дискретного сигналу всіма методами функції `interp1`.
3. Виконати інтерполяцію дискретного сигналу функцією `spline`.
4. Реалізувати програмно `sinc`-інтерполяцію дискретного сигналу.
5. Візуально й математично порівняти отримані результати та початковий сигнал.
6. Пояснити отримані результати.

Для зображення обраного з попередніх лабораторних робіт:

1. Сформуванати зменшений дискретний варіант зображення (функцією `imresize`).
2. Виконати інтерполяцію дискретного сигналу всіма методами функції `interp2`.
3. Виконати інтерполяцію дискретного сигналу функцією `spline`.
4. Реалізувати програмно `sinc`-інтерполяцію дискретного зображення.
5. Візуально й математично порівняти отримані результати та початковий сигнал.
6. Пояснити отримані результати.

3. ЗМІСТ ЗВІТУ З ПРАКТИЧНОЇ РОБОТИ

1. Формулювання завдання.
2. Текст програми.
3. Результати виконання всіх розділів завдання.
4. Файл із результатами виконання завдання.
5. Пояснення отриманих результатів.
6. Відповіді на контрольні питання.

5. КОНТРОЛЬНІ ПИТАННЯ

1. Охарактеризуйте сутність інтерполяції сигналів.
2. В яких випадках доцільно використовувати функцію `meshgrid`.
3. Обґрунтуйте відповідь на питання чому при інтерполяції сигнали змінюють свій розмір.
4. Обґрунтуйте відповідь на питання чому різні методи інтерполяції формують різні за значенням сигнали.
5. Поясніть відмінність інтерполяції сплайнами від інших методів.
6. Як змінюється відтворений сигнал при різних параметрах інтерполяції.
7. В яких випадках доцільно використовувати функцію `spline`.
8. Охарактеризуйте сутність і особливості `sinc`-інтерполяції.

З навчальними матеріалами з питань дослідження інтерполяції сигналів в середовищі системи MATLAB можна детальніше ознайомитись також в [2 (с.465-474), 3].

Практична робота № 5.

КВАНТУВАННЯ АУДІО СИГНАЛІВ [11]

Мета практичної роботи – дослідження способів квантування звукових сигналів у середовищі системи MATLAB.

1. КВАНТУВАННЯ СИГНАЛІВ У MATLAB

У загальному випадку, при квантуванні діапазон можливих значень сигналу розділюється на інтервали квантування [2,7,8]:

$$x_0 \dots x_1, x_1 \dots x_2, x_2 \dots x_3, \dots, x_{k-1} \dots x_k, \dots, x_{N-1} \dots x_N.$$

Інтервалам квантування зіставляються вихідні квантуванні значення. Якщо значення сигналу міститься у діапазоні $[x_{k-1} \dots x_k]$, то значення результату його квантування дорівнюватиме y_k [7,8].

Відстань між сусідніми значеннями квантування називають рівнем (кроком) квантування. Якщо крок постійний, квантування є рівномірним (лінійним). При цьому число рівнів рівномірно розподілене в динамічній області можливих вхідних рівнів $\pm U_{\max}$, тобто [8]:

$$b = \frac{2U_{\max}}{N} = \frac{2U_{\max}}{2^q}$$

де $N = 2^q$ - кількість рівнів квантування, а q – розрядність квантування.

Наприклад, у випадку 16-розрядного перетворення кількість рівнів квантування дорівнює $2^{16}=65536$. При цьому похибка (шум) квантування, що представляє собою різницю між початковим сигналом і результатом його квантування $e(t)=s(t)-s_k(t)$, знаходиться в межах: $-b/2 < e(t) \leq b/2$ [11].

При нелінійному (нерівномірному) квантуванні в процесі командування може використовуватися характеристика стиснення, за наступним законом [4,7]:

$$y = \begin{cases} y_{\max} \frac{A \left(\frac{|x|}{x_{\max}} \right)}{1 + \ln A} \operatorname{sgn} x, & 0 < \frac{|x|}{x_{\max}} \leq \frac{1}{A} \\ y_{\max} \frac{1 + \ln \left[A \left(\frac{|x|}{x_{\max}} \right) \right]}{1 + \ln A} \operatorname{sgn} x, & \frac{1}{A} < \frac{|x|}{x_{\max}} \leq 1 \end{cases}$$

де $A > 0$ – константа, x та y – значення на вході та виході квантователя.

Однією з основних функцій квантування в системі MATLAB є `quantiz`, яка реалізує нерівномірне квантування (при цьому рівномірне квантування трактується як окремий випадок нерівномірного) і має наступний синтаксис:

`[index, quants, distort] = quantiz(sig, partition, codebook),`

де `sig` – вектор відліків сигналу, `partition` – вектор границь рівнів квантування вхідного сигналу, `codebook` – вектор рівнів квантування вихідного сигналу. Довжина вектору `codebook` повинна бути на одиницю більшою за довжину вектору `partition`. Вихідні параметри `index` та `quants` містять індекси та значення квантованого сигналу і мають розміри, які дорівнюють вектору сигналу [6].

Значення `index` встановлюються таким чином [8]:

- `index(k) = 0` якщо `sig(k) ≤ partition(1)`,
- `index(k) = m` якщо `partition(m) < sig(k) ≤ partition(m+1)`,
- `index(k) = length(partition)` якщо `sig(k) > partition(end)`.

Значення `quants` розраховується як `codebook(index+1)`. Отже, величина `codebook(1)` задає рівень квантування для всіх значень сигналу, менших, ніж `partition(1)`, величина `codebook(m)` відповідає зоні квантування `partition(m-1) ... partition(m)`, а значення `codebook(end)` використовується у якості рівня квантування для всіх значень сигналу, вищих за `partition(end)`.

Вихідний параметр `distort` містить значення середнього квадрату помилки квантування [9].

Якщо параметри `quants` та `distort` не вказані, то параметр `codebook` при виклику функції допустимо не вказувати [9].

Приклад застосування функції `quantiz` для лінійного квантування [11]:

```
clear all;
[xt, Fs, bits] = wavread('c:\windows\media\tada.wav', [10001 11000]);
xt1 = xt(:,1);
xt2 = xt(:,2);
% -----
x=xt1;
xmax=max(abs(x)); x=x/xmax; %нормування
T=length(x)/Fs; disp('Тривалість сигналу T='); disp(T);
t=1/Fs:1/Fs:length(x)/Fs; figure; subplot(311); plot(t,x);
title(['Перший канал початкового сигналу ', num2str(bits), ' бит ', num2str(Fs), ' Гц']);
xlabel('Час, c'); grid on;
```

```

%-----
q=10; % Кількість біт квантування
N=2^q; % Кількість рівней квантування
b=2/(N-1); % Шаг квантування
partition=-1+b/2:b:1-b/2;
codebook=-1:b:1;
[ind, y, dis]=quantiz(x, partition, codebook);
subplot(312); plot(t,y);
title(['Квантований сигнал ', num2str(q),' бит ',num2str(Fs),' Гц']);
xlabel('Час, с'); grid on;
%-----
--err=x'-y;
subplot(313); plot(t,err);
title(['Шум
квантування']);
xlabel('Час, с'); grid on;
dis

```

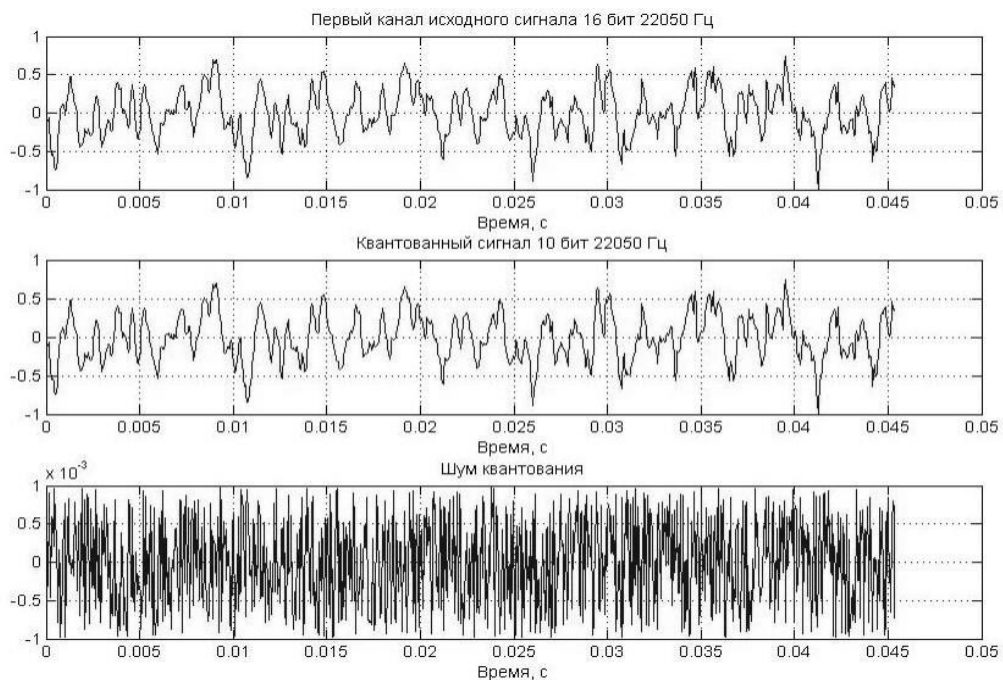


Рис.5.1 Приклад рівномірного квантування [11]

Наведемо приклад нелінійного квантування [11]:

```

clear all;
[xt, Fs, bits] = wavread ('c:\windows\media\tada.wav', [10001 10100]);
x1 = xt(:,1);
x2 =
xt(:,2); %
-----
x=x1; %Один з каналів
xmax=max(abs(x)); x=x/xmax;

```

```

T=length(x)/Fs; disp('Тривалість сигналу T='); disp(T);
t=1/Fs:1/Fs:length(x)/Fs; figure; subplot(511); plot(t,x);
title(['Перши канал початкового сигналу ',num2str(bits),'бит',num2str(Fs),' Гц']);
xlabel('Время, с'); grid on;
%----- перед спотворений сигнал -----
A = 50;          % значення параметра A
if 0<abs(x)<=1/A
    yc = A*abs(x).*sign(x)/(1+log(A));
else
    if x == 0
        x = 1e-3;
    end
    yc = (1+log(A*abs(x))).*sign(x)/(1+log(A)); end
%sum_nan = sum(isnan(yn)); % кількість NaN
%yn = yn(~isnan(yn));      % видалення NaN
tc = 1/Fs:1/Fs:(length(yc))/Fs;
subplot(512); plot(tc,yc);
title(['Пред спотворений сигнал ', num2str(bits), ' бит ', num2str(Fs), ' Гц']);
xlabel('Час, с'); grid on;
%----- квантування -----
q = 10;          % нова кількість біт
N = 2^q;         % нова кількість рівней
b = 2/(N-1); % шаг рівня
partition = -1+b/2:b:1-b/2;
codebook = -1:b:1;
[ind, ynq, dis] = quantiz(yc, partition, codebook);
subplot(513); plot(tc,ynq);
title(['Квантований пред спотворений сигнал ',num2str(q), ' біт ',
num2str(Fs), ' Гц']);
xlabel('Час, с'); grid on;
% ----- експандування квантованого сигналу -----
if 0<abs(ynq)<=1/(1+log(A))
    yv = (1+log(A))*abs(ynq).*sign(ynq)/A;
else
    yv = exp((1+log(A))*abs(ynq)-1).*sign(ynq)/A;
end
subplot(514); plot(tc,yv);
title(['Результат нелінійного квантування',num2str(q),'бит',num2str(Fs),' Гц']);
xlabel('Час, с'); grid on;
err=x'-yv;
subplot(515); plot(t,err);
title(['Шум квантування']);
xlabel('Время, с'); grid on;
dis

```

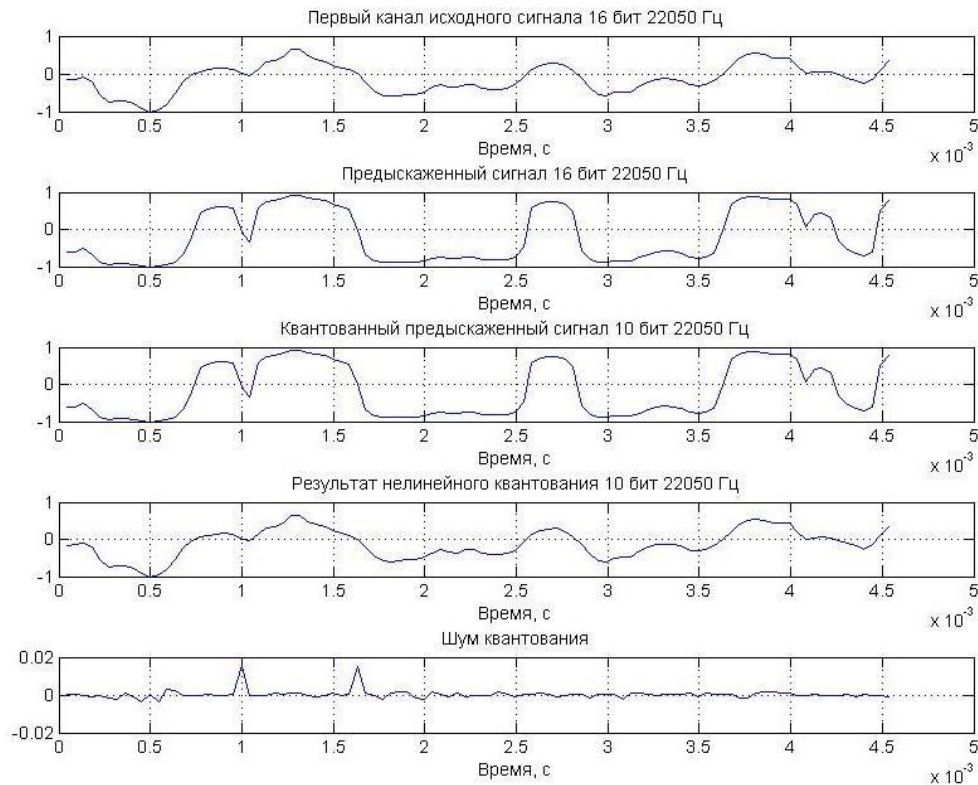


Рис.5.2 Приклад нерівномірного квантування [11]

Для рівномірного квантування також можна використовувати функції `uencode` та `udocode`, а функція `lloydс` призначена для оптимізації параметрів не-рівномірного квантування за тестовим набором даних.

2. ЗАВДАННЯ ДО ПРАКТИЧНОЇ РОБОТИ

1. Визначити характеристики заданого звукового сигналу.
2. Виконати рівномірне квантування заданого сигналу, змінюючи розрядність квантователя (параметр `q`). Встановити залежність між розрядністю (`q`) та похибкою (`dis`). (Кількість вимірювань повинна бути більше п'яти).
3. Виконати нелінійне квантування, змінюючи розрядність квантування (параметр `q`) та коефіцієнт `A` (до значення `10A`). Встановити залежність між розрядністю (`q`), коефіцієнтом `A` та похибкою (`dis`). (Кількість вимірювань повинна бути більше п'яти).
4. Забезпечити графічне виведення форми сигналу (допускається – фрагменту сигналу) до і після квантування, а також форму відповідного шуму.
5. Забезпечити звукове відтворення сигналу повністю до і після квантування з метою встановлення прийнятних параметрів квантування.

3. ЗМІСТ ЗВІТУ З ЛАБОРАТОРНОЇ РОБОТИ

1. Формулювання завдання.
2. Текст програми.
3. Результати виконання всіх розділів завдання.
4. Файл з результатами виконання завдання.
5. Пояснення отриманих результатів.
6. Відповіді на контрольні запитання.

4. ВАРІАНТИ ЗАВДАНЬ [11]

Вар	WAV - файл	Вар	WAV - файл
1	Balloon	19	Windows - акорд
2	Battery Critical	20	Windows - заблоковані спливаючі вікна
3	Battery Low	21	Windows – завершення роботи
4	Critical Stop	22	Windows - дзвіночок
5	Default	23	Windows - вилучення (видалення) обладнання
6	Ding	24	Windows - вихідний дзвінок
7	Error	25	Windows - повідомлення
8	Hardware Insert	26	Windows - корзина
9	Logoff Sound	27	Windows - критична помилка
10	Logon Sound	28	Windows - низький заряд батарей
11	Notify	29	Windows - відмова обладнання
12	Windows - оклик	30	Windows - помилка
13	Windows - відновлення	31	Вхід у Windows
14	Windows - спливаюче повідомлення	32	Windows - печать завершена
15	Windows - вставка обладнання	33	Вихід із Windows
16	Windows - майже повна розрядка батарей	34	Windows - стандартний звук
17	Windows - згорнення	35	$s(t) = \frac{a_0}{2} + \sum_{i=1}^M A_i \cdot \cos(\omega_i t + \varphi_i)$
18	Windows - вхідний дзвоник		

5. КОНТРОЛЬНІ ПИТАННЯ

1. Обґрунтуйте переваги нелінійного квантування перед лінійним.
2. Охарактеризуйте вплив коефіцієнту A на результат квантування.
3. Порядок оцінки похибки квантування.
4. Обґрунтуйте залежність якості квантування від форми сигналу.
5. Охарактеризуйте термін «експандування».
6. Охарактеризуйте поняття «командування».
7. Обґрунтуйте, чому довжина вектору codebook повинна бути на одиницю більшою за довжину вектору partition.
8. Охарактеризуйте сутність і функції параметрів `quants` та `distor`.

З навчальними матеріалами з питань дослідження квантування сигналів в середовищі системи MATLAB можна детальніше ознайомитись також в [3].

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Digital Image Processing Using MATLAB, 2nd ed. by Rafael C. Gonzalez, 2009.
2. Richard G. Lyons. Understanding Digital Signal Processing. – Prentice Hill, Professional Technical Reference, 2011.
3. Кобилін О.А., Творошенко І.С. Методи цифрової обробки зображень: навч. посібник. – Харків: ХНУРЕ, 2021. – 124 с.
4. Коваль о.А. Конспект лекцій. Методи синтезу і аналізу вимірних сигналів. – ХНАДУ, Харків, 20182, 367 с.
5. Самощенко О.В. Конспект лекцій з курсу «Комп'ютерні засоби обробки графічної, звукової та відеоінформації». – ДонНТУ, 2017. Рег.№196, прот.№2 від 21.03.2017.
6. Steven W.Smith. Digital Signal Processing. A Practical Guide for Engineers and Scientists. – California Technical Publishing San Diego, California, 2013
7. Rafael C. Gonzalez, Richard E. Woods. Digital Image Processing - 4th Edition, 2017.
8. Bocharnikov V., Bocharnikov I. Optimal discrete fuzzy filter of UAV's flight parameters. ISSN 1813-1166. Proceedings of the NAU. 2012. № 2. P. 22-29.
9. Методичні вказівки до виконання індивідуального завдання з дисципліни «Методи дослідження та аналізу дискретних сигналів» для студентів магістратури спеціальності 123 Комп'ютерна інженерія / уклад. О. Самощенко. – Покровськ: ДонНТУ, 2020 – 16 с.
10. Методичні вказівки до виконання лабораторних робіт з дисципліни «Цифрова обробка сигналів» для студентів напряму «Комп'ютерна інженерія» денної та заочної форм навчання / О.В.Самощенко. – Покровськ: ДонНТУ, 2016.- 40 с.

Інформаційні ресурси

11. Самощенко О.В. «Методичні вказівки до самостійної роботи студентів "Пакети прикладних програм системного аналізу. Комп'ютерна обробка даних в середовищі системи MatLab». - 2019. https://www.dut.edu.ua/uploads/l_1757_17123724.pdf
12. Самощенко О.В. Лекція: Швидке перетворення Фур'є. [Ел.ресурс], <http://ea.donntu.edu.ua/bitstream/123456789/7721/1/FFT%20basic.pdf>
13. Доступ до програмного забезпечення MatLab. <https://it.sumdu.edu.ua/dostup-do-prohramnoho-zabezpechennia-matlab>
14. Навчальні курси MatLab. <https://www.nobleprog.com.ua/kursy-matlab>
15. Еквалізація [Ел.ресурс], 03.05.2021 [https://en.wikipedia.org/wiki/Equalization_\(audio\)](https://en.wikipedia.org/wiki/Equalization_(audio))
16. Цифровий звук [Ел. ресурс], 18.04.2021 https://en.wikipedia.org/wiki/Digital_audio
17. Joe Wolfe, What is a Sound Spectre <http://newt.phys.unsw.edu.au/jw/sound.spectrum.html>
18. Introduction to Digital Filters with Audio Applications', by Julius O. Smith III, What is a Filter https://ccrma.stanford.edu/~jos/filters/What_Filter.html
19. FIR filter basics <https://dspguru.com/dsp/faqs/fir/basics/>
20. Обробка зображень в MatLab. <https://www.mathworks.com/videos/image-processing-made-easy-96823.html>
21. Основи теорії сигналів. Дослідження і обробка цифрових сигналів. <https://org2.knuba.edu.ua/mod/book/tool/print/index.php?id=23040>

Приклад виконання і порядок оформлення результатів роботи

ПРАКТИЧНА РОБОТА №1 (варіант N)

ПЕРЕТВОРЕННЯ ЦИФРОВИХ ОДНОВИМІРНИХ СИГНАЛІВ
В ЧАСТОТНОЇ ОБЛАСТІ

Мета практичної роботи - дослідження методики одновимірного дискретного перетворення Фур'є в середовищі системи MATLAB.

Завдання:

1. Для інтервалів різної довжини (не менше трьох - пересічних, вкладених, непересічних), що характеризують властивості заданого звукового сигналу, знайти амплітудний і фазовий спектри за трьома варіантами перетворень.
2. Відновити сигнали, оцінити похибку перетворення.
3. Виконати дослідження перетворень, змінюючи:
 - точність перетворень (параметр округлення в функції roundn),
 - розмір фрагментів,
 - розмір блоку ШПФ (параметр N в функції fft),
 - розташування сигналу (зрушення вліво і вправо),
 - склад сигналу (додавання фрагментів).
4. Виконати перетворення сигналу в цілому, відтворити звуковий сигнал до і після перетворення і порівняти звучання.
5. При перетворенні сигналу в цілому виконати часткове обнуління коефіцієнтів (рівень обнуління задається відповідно до індивідуального варіанту, наприклад, спочатку обнуляють 10% коефіцієнтів, потім 18% і т.д.).
6. Для кожного випадку обнуління відновити сигнал.
7. Оцінити похибку для кожного з випадків відновлення.
8. Підібрати оптимальне, на свою думку, обнуління коефіцієнтів, яке забезпечує прийнятне стиснення без істотної втрати якості.

Параметри обнуління: початковий рівень обнуління 10%, шаг обнуління 5%, кількість операцій обнуління-відновлення 12.

Код:

```
sd = [0.3536, 0.3536, 0.6464, 1.0607, 0.3536, -1.0607, -1.3536, -0.3536];
f1=1; f2=2; % Частоти гармонік
phi1=-pi/2; phi2=pi/4; % Фази гармонік
% Генерація сигналу з двох гармонік
t=-1:0.01:2; % вектор моментів безперервного часу
s1=cos(2*pi*f1.*t+phi1); % Перша гармоніка
s2=0.5*cos(2*pi*f2.*t+phi2); % Друга гармоніка
s=s1+s2; % Безперервний сигнал
subplot(4,2,1); plot(t,s,t,s1,t,s2); title('Аналоговий сигнал и гармоніки');
%Дискретизація

td=0.5:0.125:1.5;
sd1=cos(2*pi.*f1.*td+phi1);
sd2=0.5*cos(2*pi.*f2.*td+phi2);
sd=sd1'+sd2'; % Дискретний сигнал
subplot(4,2,2); stem(td,sd); title('Дискретний сигнал 1');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
set(gca, 'xtick', td)
N=length(sd); n=[0:N-1]; k=[0:N-1];
rd = roundn(cos((2.*pi.*n/length(n))*k)*sd,-2);
id = -roundn(sin((2.*pi.*n/length(n))*k)*sd,-2);
md = abs((exp((-1i.*2.*pi.*n/length(n))*k))*sd);
pd=(atan2(id,rd))/pi;
subplot(4,2,3); stem(md); title('Амплітуда дискретного сигналу');
subplot(4,2,4); stem(pd); title('Фаза дискретного сигналу');
set(gca, 'ytick', -1:0.25:1)
% -----
% Спектр методом FFT
YdFs=roundn(fft(sd),-2);
rdF=real(YdFs);
idF=imag(YdFs);
mdF = abs(YdFs);
pdF = angle(YdFs)/pi;
subplot(4,2,5); stem(mdF); title('Амплітуда FFT дискретного сигналу');
subplot(4,2,6); stem(pdF); title('Фаза FFT дискретного сигналу');
set(gca, 'ytick', -1:0.25:1)
% -----
```

```

% Спектр метод матриці DFT
M=dftmtx(N);          % Матриця ДПФ
Yr=roundn((M*sd(1:N)),-3); % ДПФ с округленням
m = abs(Yr);          % Амплітуда
p = angle(Yr)/pi;     % Фаза
subplot(4,2,7); stem(m); title('Амплітуда DFT дискретного сигналу');
subplot(4,2,8); stem(p); title('Фаза DFT дискретного сигналу');
% -----
% --- Відновлення дискретного сигналу -----
Mi=conj(dftmtx(N))/N; % матриця зворотного ДПФ
sdi=roundn((Mi*Yr),-4);
delta=norm(sd-sdi);
%%%%%%%%%%%%%%

figure;

td=0:0.125:(1-0.125); % Кількість відліків в періоді дорівнює 8
sd1=cos(2*pi.*f1.*td+phi1);
sd2=0.5*cos(2*pi.*f2.*td+phi2);
sd=sd1'+sd2';          % Дискретний сигнал
subplot(4,2,2); stem(td,sd); title('Дискретний сигнал 2');
%%%%%%%%%%%%%%
set(gca, 'xtick', td)
N=length(sd); n=[0:N-1]; k=[0:N-1];
rd = roundn(cos((2.*pi.*n/length(n))*k)*sd,-2);
id = -roundn(sin((2.*pi.*n/length(n))*k)*sd,-2);
md = abs((exp((-1i.*2.*pi.*n/length(n))*k))*sd);
pd=(atan2(id,rd))/pi;
subplot(4,2,3); stem(md); title('Амплітуда дискретного сигналу');
subplot(4,2,4); stem(pd); title('Фаза дискретного сигналу');
set(gca, 'ytick', -1:0.25:1)
% -----
% Спектр методом FFT
YdFs=roundn(fft(sd),-2);
rdF=real(YdFs);
idF=imag(YdFs);
mdF = abs(YdFs);
pdF = angle(YdFs)/pi;
subplot(4,2,5); stem(mdF); title('Амплітуда FFT дискретного сигналу');
subplot(4,2,6); stem(pdF); title('Фаза FFT дискретного сигналу');
set(gca, 'ytick', -1:0.25:1)
% -----

```

```

% Спектр методом матриці DFT
M=dftmtx(N);          % Матриця ДПФ
Yr=roundn((M*sd(1:N)),-3); % ДПФ с округленням
m = abs(Yr);          % Амплітуда
p = angle(Yr)/pi;     % Фаза
subplot(4,2,7); stem(m); title('Амплітуда DFT дискретного сигналу');
subplot(4,2,8); stem(p); title('Фаза DFT дискретного сигналу');
% -----
% --- Відновлення дискретного сигналу -----
Mi=conj(dftmtx(N))/N; % матриця зворотного ДПФ
sdi=roundn((Mi*Yr),-4);
delta1=norm(sd-sdi);
%%%%%%%%%%%%%%

figure;

td=1:0.125:2; % Кількість відліків в періоді дорівнює 8
sd1=cos(2*pi.*f1.*td+phi1);
sd2=0.5*cos(2*pi.*f2.*td+phi2);
sd=sd1'+sd2'; % Дискретний сигнал
subplot(4,2,2); stem(td,sd); title('Дискретний сигнал 3');

%%%%%%%%%%%%%%
set(gca, 'xtick', td)
N=length(sd); n=[0:N-1]; k=[0:N-1];
rd = roundn(cos((2.*pi.*n/length(n))*k)*sd,-2);
id = -roundn(sin((2.*pi.*n/length(n))*k)*sd,-2);
md = abs((exp((-1i.*2.*pi.*n/length(n))*k))*sd);
pd=(atan2(id,rd))/pi;
subplot(4,2,3); stem(md); title('Амплітуда дискретного сигналу');
subplot(4,2,4); stem(pd); title('Фаза дискретного сигналу');
set(gca, 'ytick', -1:0.25:1)
% -----
% Спектр методом FFT
YdFs=roundn(fft(sd),-2);
rdF=real(YdFs);
idF=imag(YdFs);
mdF = abs(YdFs);
pdF = angle(YdFs)/pi;
subplot(4,2,5); stem(mdF); title('Амплітуда FFT дискретного сигналу');
subplot(4,2,6); stem(pdF); title('Фаза FFT дискретного сигналу');
set(gca, 'ytick', -1:0.25:1)
% -----
% Спектр методом матриці DFT
M=dftmtx(N);          % Матриця ДПФ
Yr=roundn((M*sd(1:N)),-3); % ДПФ с округленням

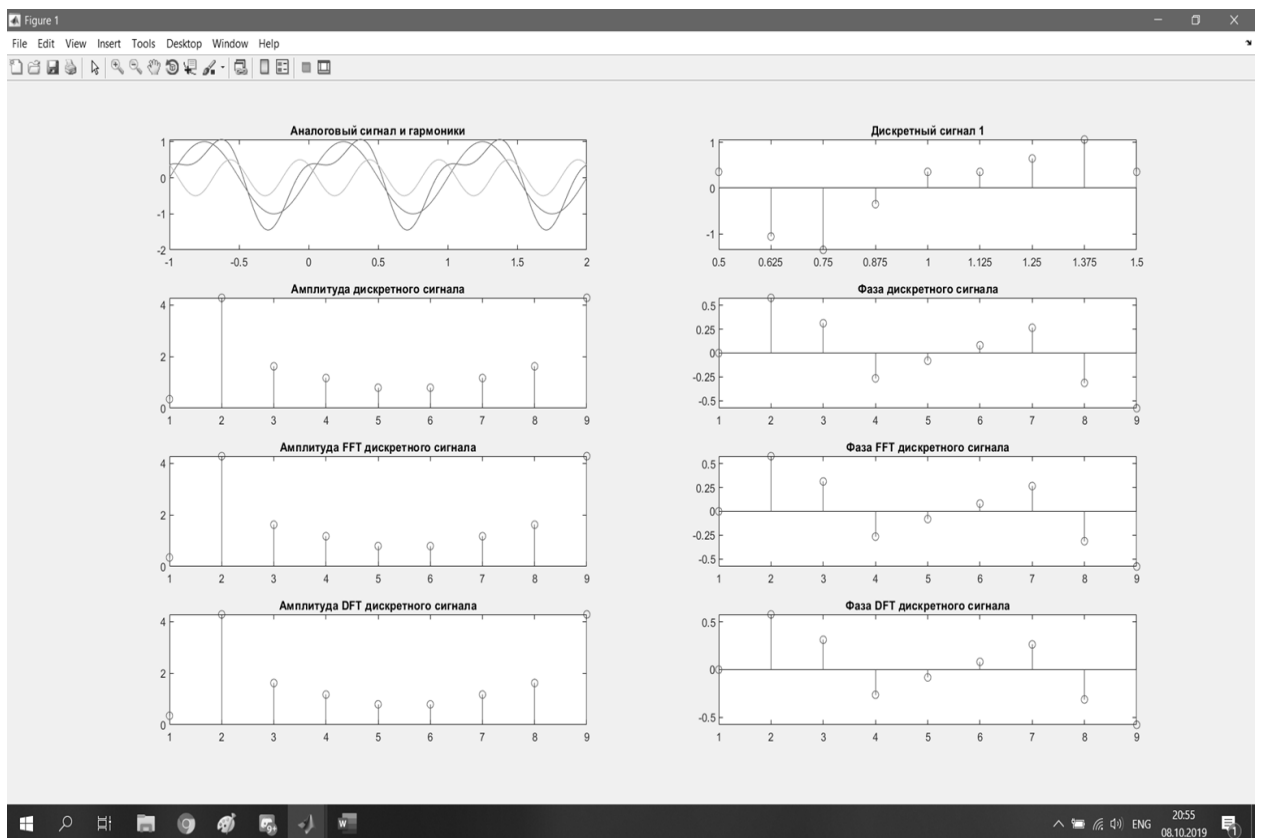
```

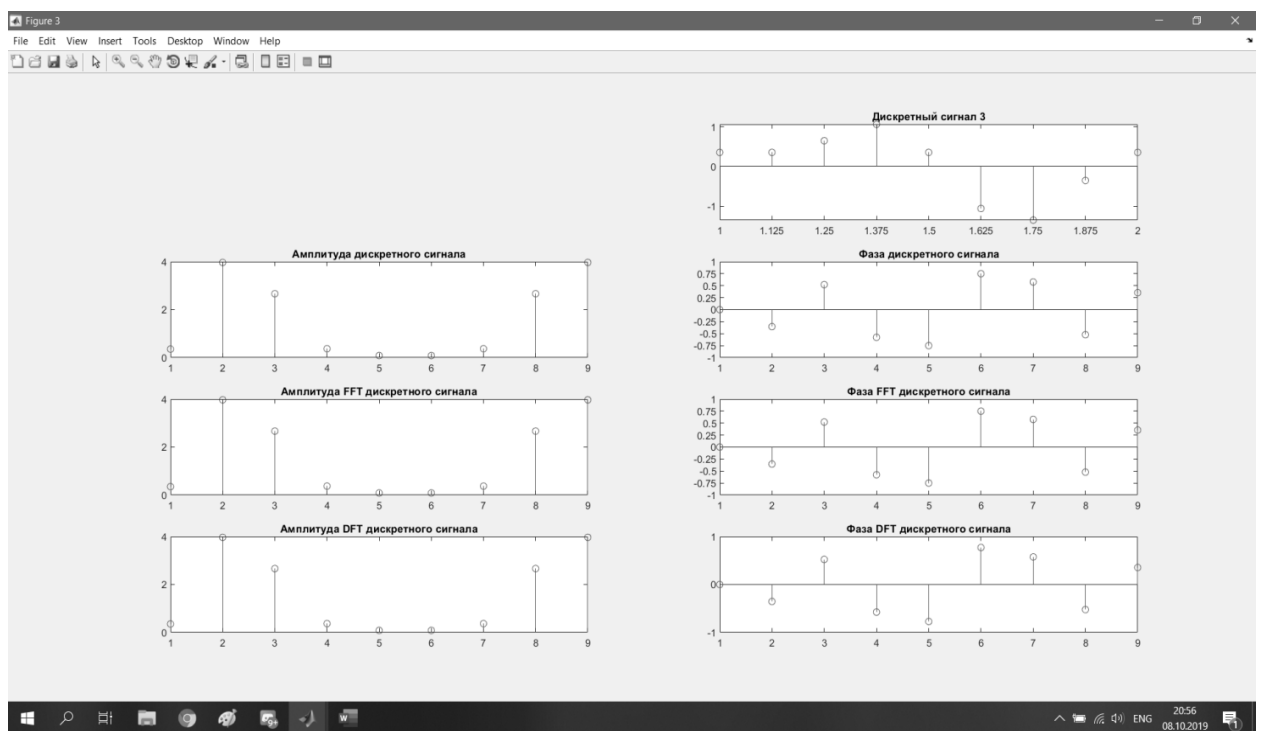
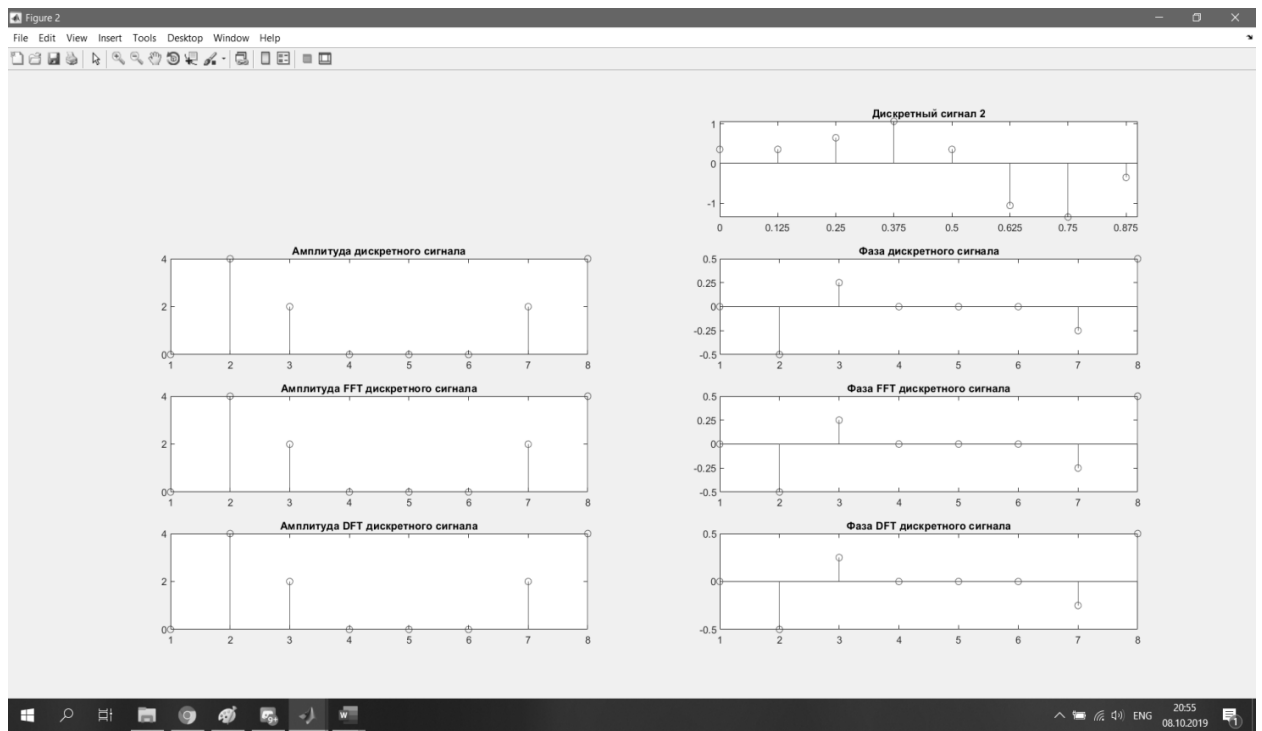
```

m = abs(Yr); % Амплітуда
p = angle(Yr)/pi; % Фаза
subplot(4,2,7); stem(m); title('Амплітуда DFT дискретного сигналу');
subplot(4,2,8); stem(p); title('Фаза DFT дискретного сигналу');
% -----
% --- Відновлення дискретного сигналу -----
Mi=conj(dftmtx(N))/N; % матриця зворотного ДПФ
sdi=roundn((Mi*Yr),-4);
delta2=norm(sd-sdi);
%%%%%%%%%%%%%%

```

1,2 Графік дискретного сигналу на другому скріншоті вкладений у графік дискретного сигналу на першому скріншоті, на третьому скріншоті графік пересікається відносно графіка на першому скріншоті, а графіки дискретного сигналу на третьому і другому скріншоті не пересікаються.





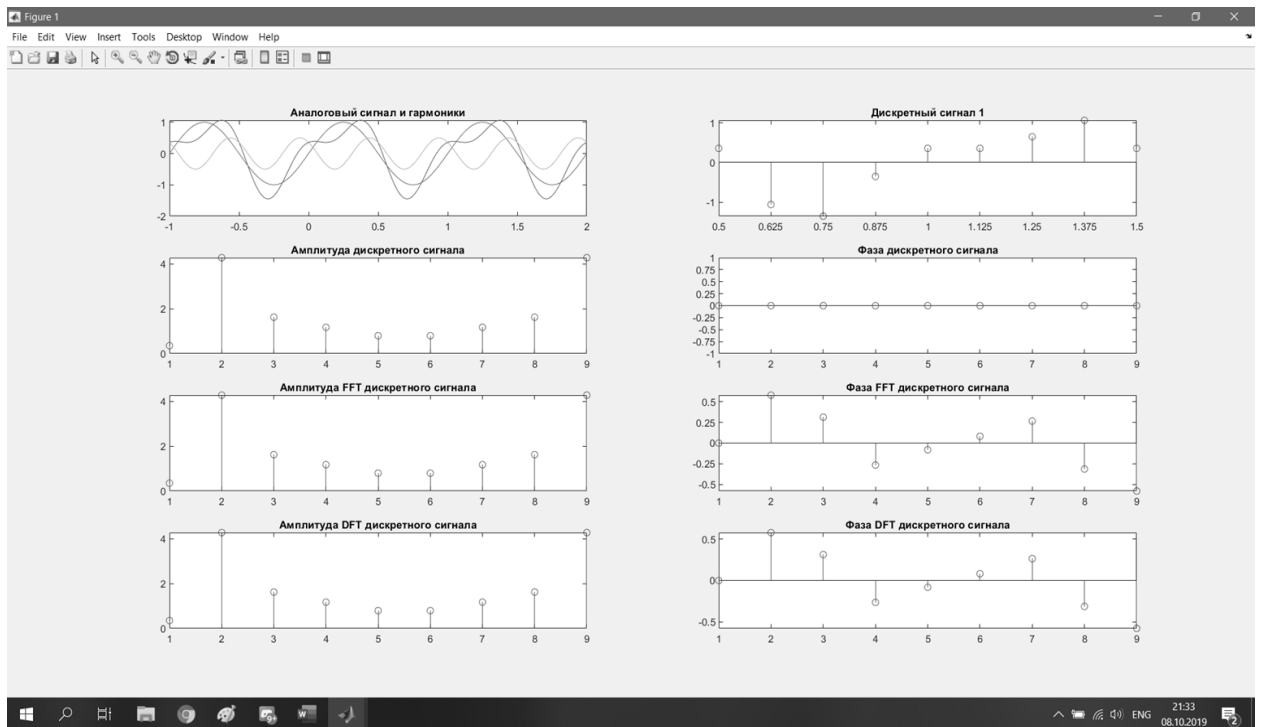
Похибка:

Name	Value
delta	3.2008e-04
delta1	1.5162e-04
delta2	3.4701e-04
Σ	4

Чим довше сигнал тим більша кількість похибок накопичується.

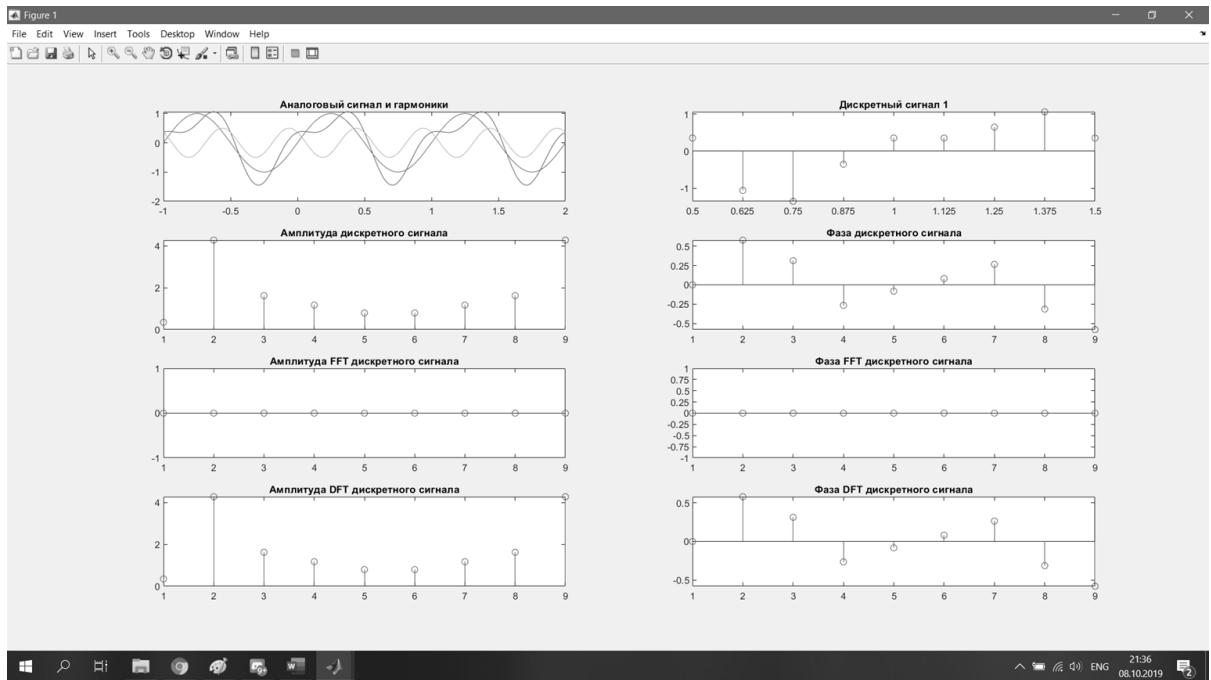
3. Під час зміни параметру округлення у функції `round` у від'ємну сторону, графік фази візуально не змінюється, хоча чим більше ціле від'ємне число, тим менше знаків після коми залишається. Якщо параметр округлення почати збільшувати у додатну сторону, то почнуть округлятися цілі розряди числа.

```
set(gca, 'xtick', td)
N=length(sd); n=[0:N-1]; k=[0:N-1];
rd = roundn(cos((2.*pi.*n/length(n))*k)*sd,1);
id = -roundn(sin((2.*pi.*n/length(n))*k)*sd,1);
md = abs((exp((-1i.*2.*pi.*n/length(n))*k))*sd);
pd=(atan2(id,rd))/pi;
subplot(4,2,3); stem(md); title('Амплітуда дискретного сигналу');
subplot(4,2,4); stem(pd); title('Фаза дискретного сигналу');
set(gca, 'ytick', -1:0.25:1)
```



`% Спектр методом FFT`

```
YdFs=roundn(fft(sd),1);
rdF=real(YdFs);
idF=imag(YdFs);
mdF = abs(YdFs);
pdF = angle(YdFs)/pi;
subplot(4,2,5); stem(mdF); title('Амплітуда FFT дискретного сигналу');
subplot(4,2,6); stem(pdF); title('Фаза FFT дискретного сигналу');
```



Зміна розміру блоку БПФ (параметр N в функції fft).

Зміна блоку БПФ з 9 до 6 (урізання).

`% Спектр методом FFT`

```
YdFs=roundn(fft(sd, 6),-2);
```

```
rdF=real(YdFs);
```

```
idF=imag(YdFs);
```

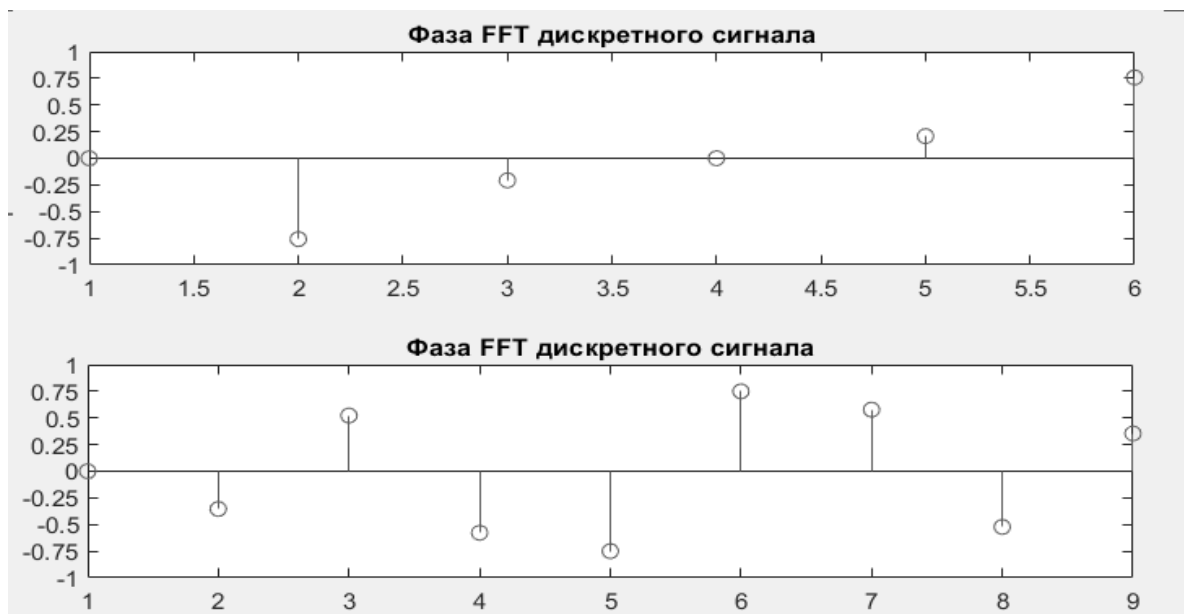
```
mdF = abs(YdFs);
```

```
pdF = angle(YdFs)/pi;
```

```
subplot(4,2,5); stem(mdF); title('Амплітуда FFT дискретного сигналу');
```

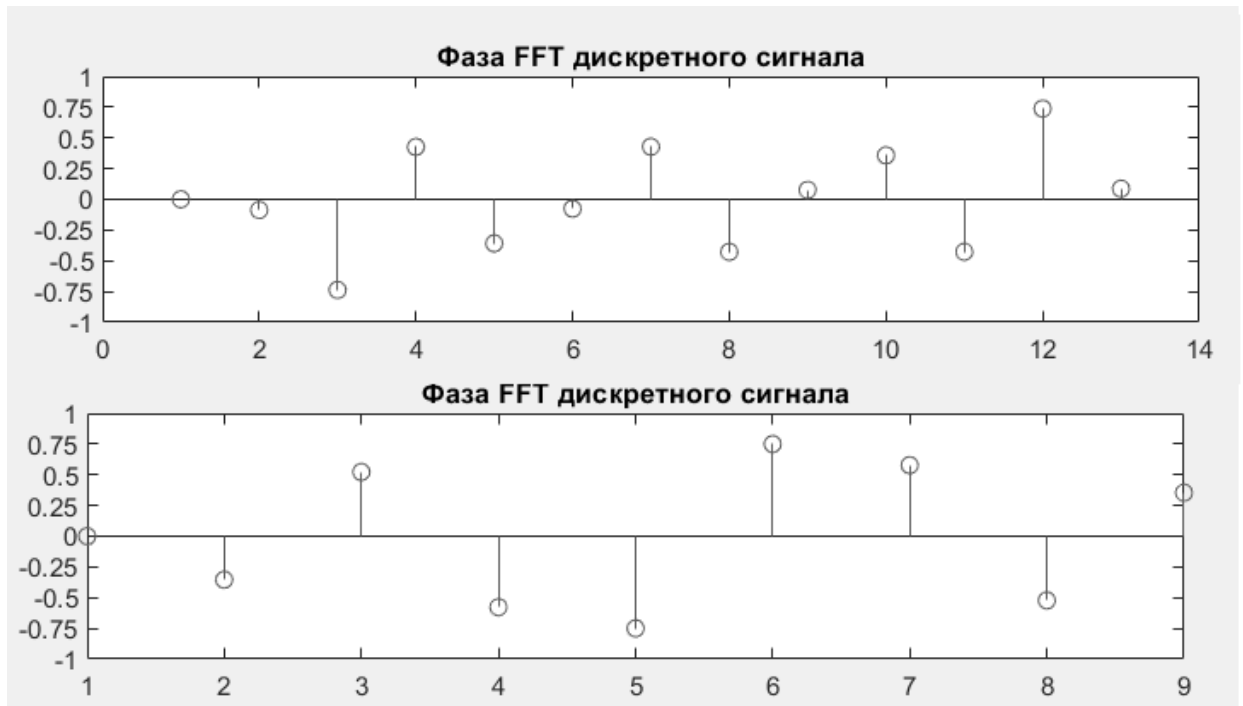
```
subplot(4,2,6); stem(pdF); title('Фаза FFT дискретного сигналу');
```

```
set(gca, 'ytick', -1:0.25:1)
```



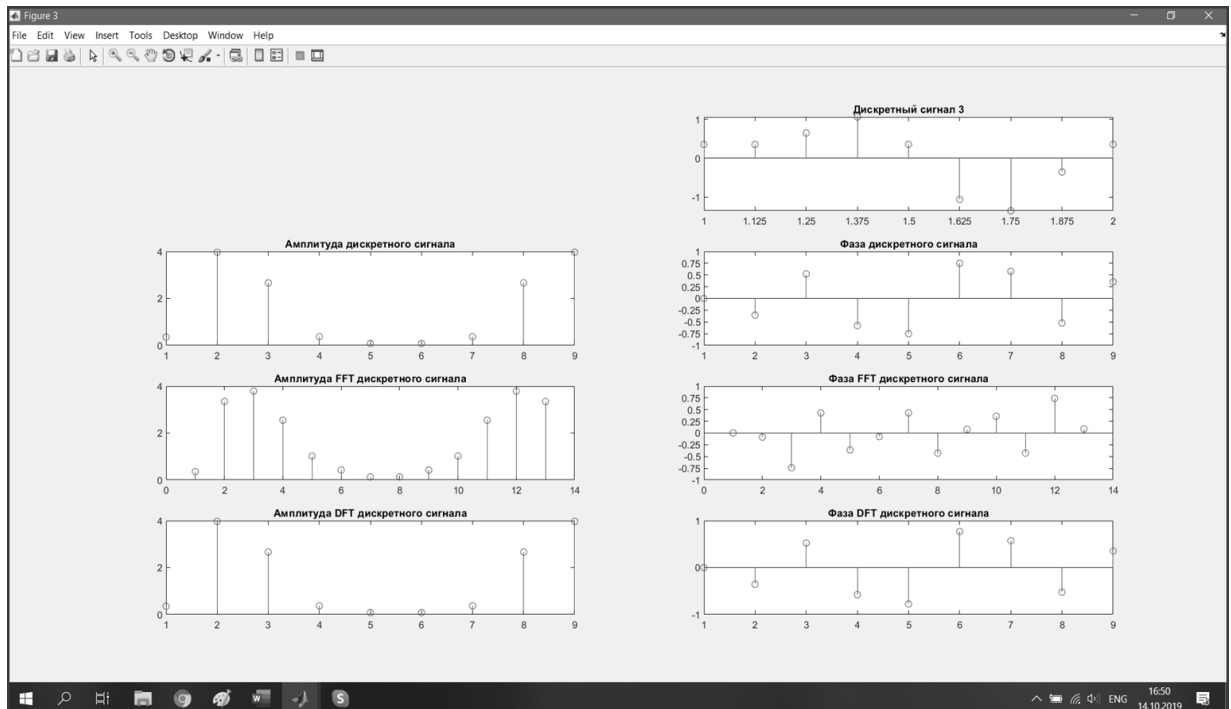
Зміна блоку БПФ з 9 до 14 (розширення нулями).

```
YdFs=roundn(fft(sd, 13),-2);
```

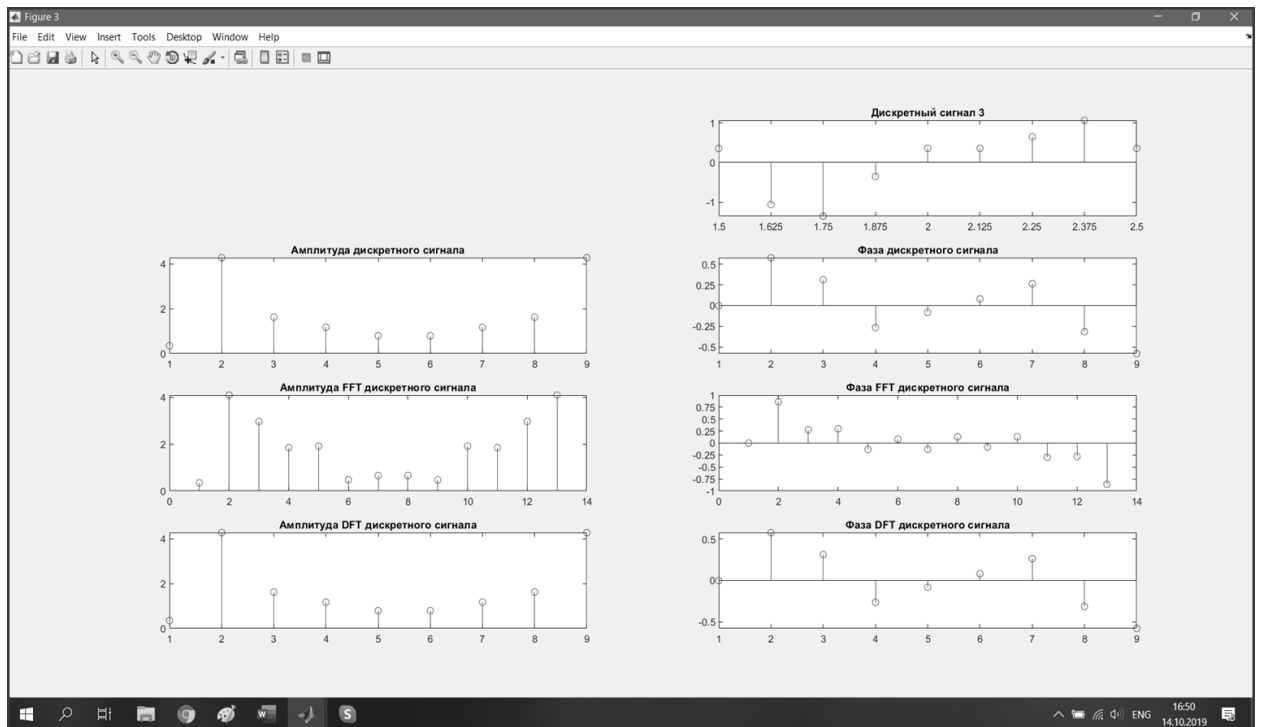


Зміна розташування сигналу (зрушення вліво і вправо).

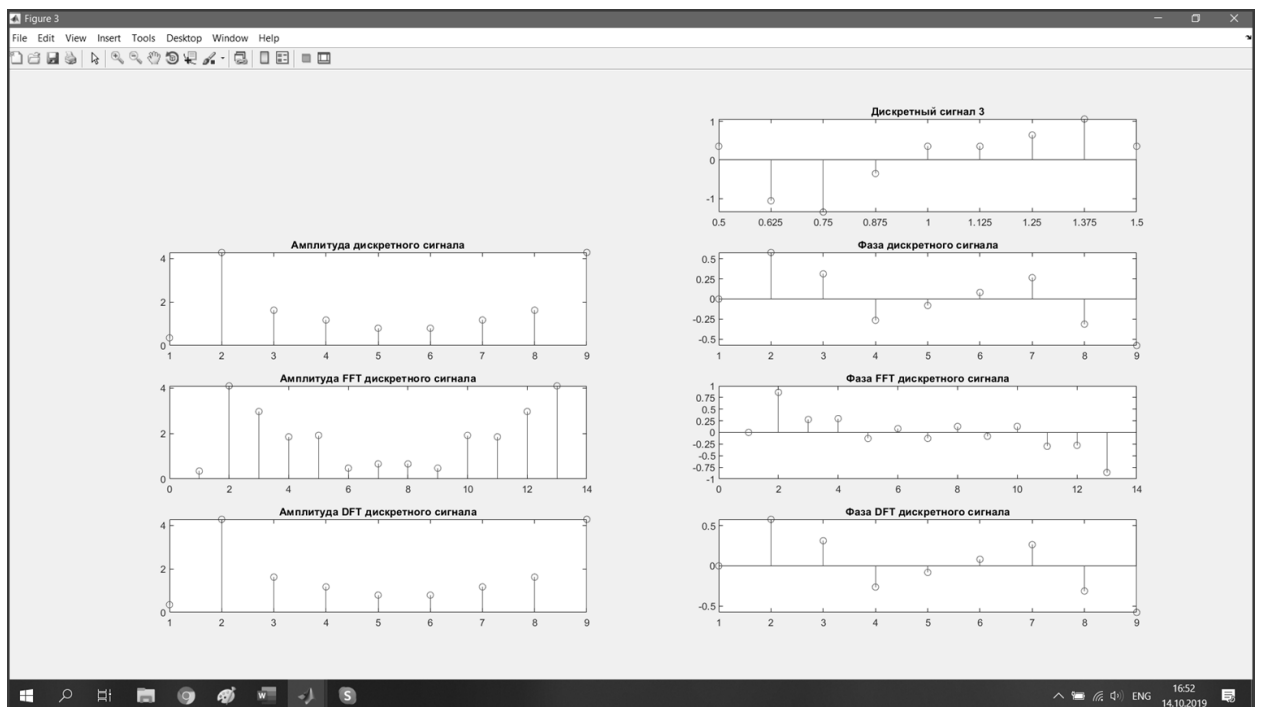
`td=1:0.125:2;` % Кількість відліків в періоді дорівнює 8



`td=1.5:0.125:2.5;` % Кількість відліків в періоді дорівнює 8



`td=0.5:0.125:1.5;` `% Кількість відліків в періоді дорівнює 8`

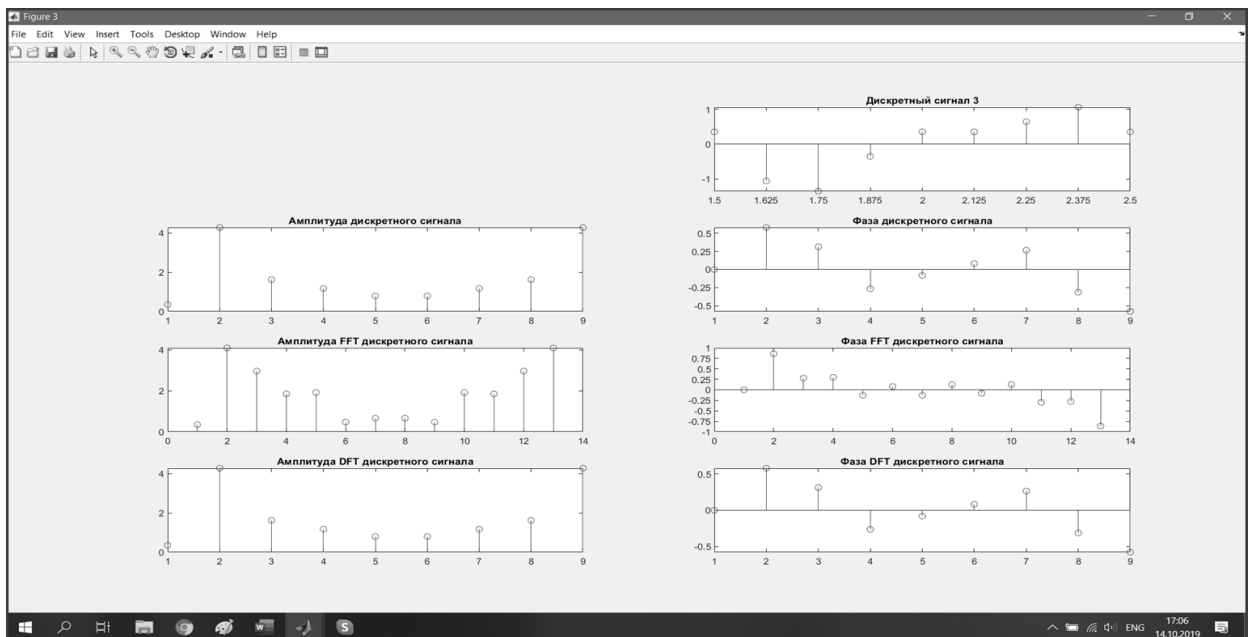


Зміна складу сигналу (додавання фрагментів).

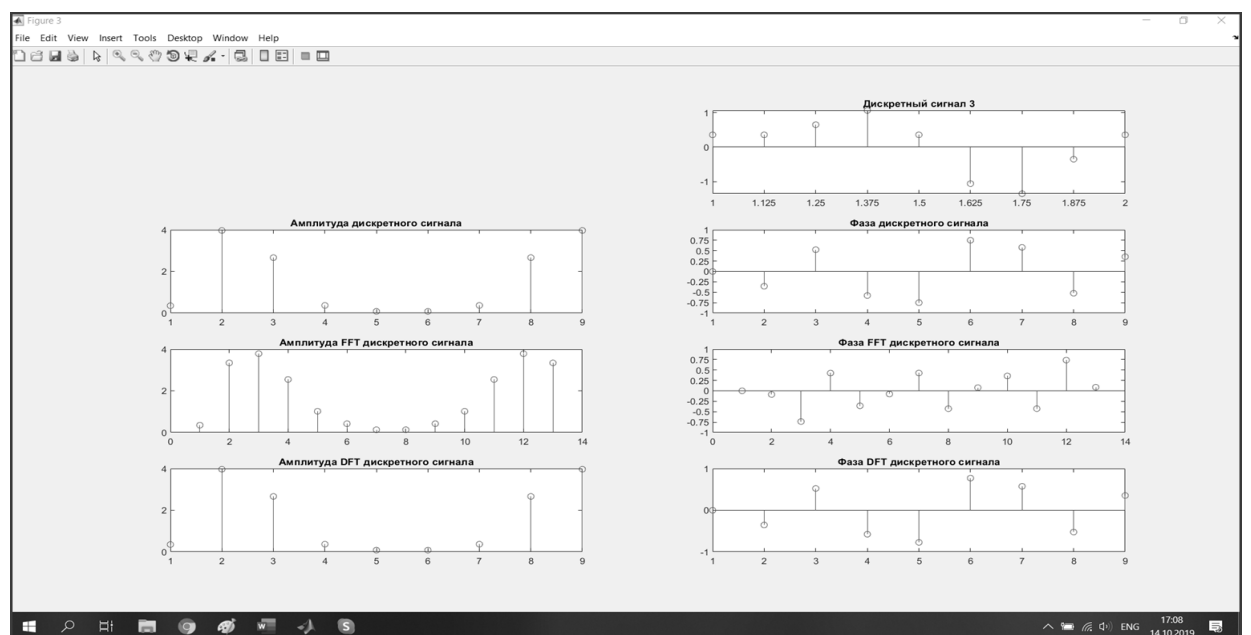
```
td=1:0.125:2;          % Кількість відліків в періоді дорівнює 8
sd1=cos(2*pi.*f1.*td+phi1);
sd2=0.5*cos(2*pi.*f2.*td+phi2);
sd99=sd1'+sd2';

td=1.5:0.125:2.5;      % Кількість відліків в періоді дорівнює 8
sd1=cos(2*pi.*f1.*td+phi1);
sd2=0.5*cos(2*pi.*f2.*td+phi2);
sd=sd1'+sd2';          % Дискретний сигнал
sd=sd+sd99;
subplot(4,2,2); stem(td,sd); title('Дискретний сигнал 3')
```

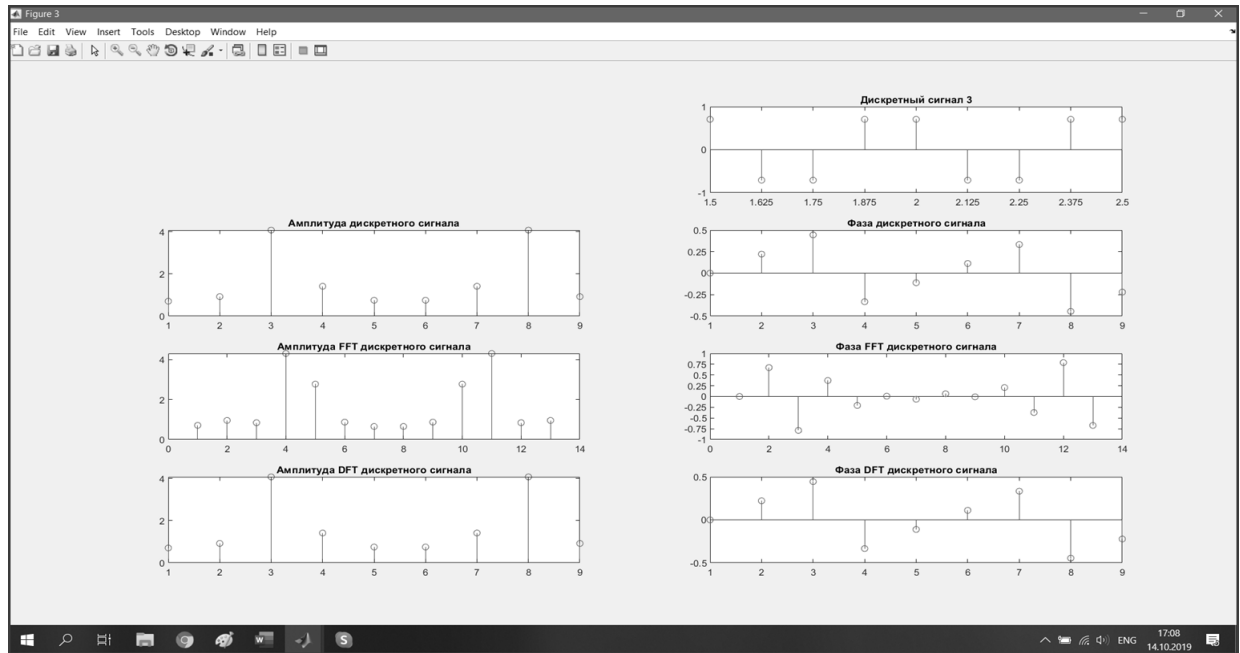
Фрагмент 1:



Фрагмент 2:



Після додавання:



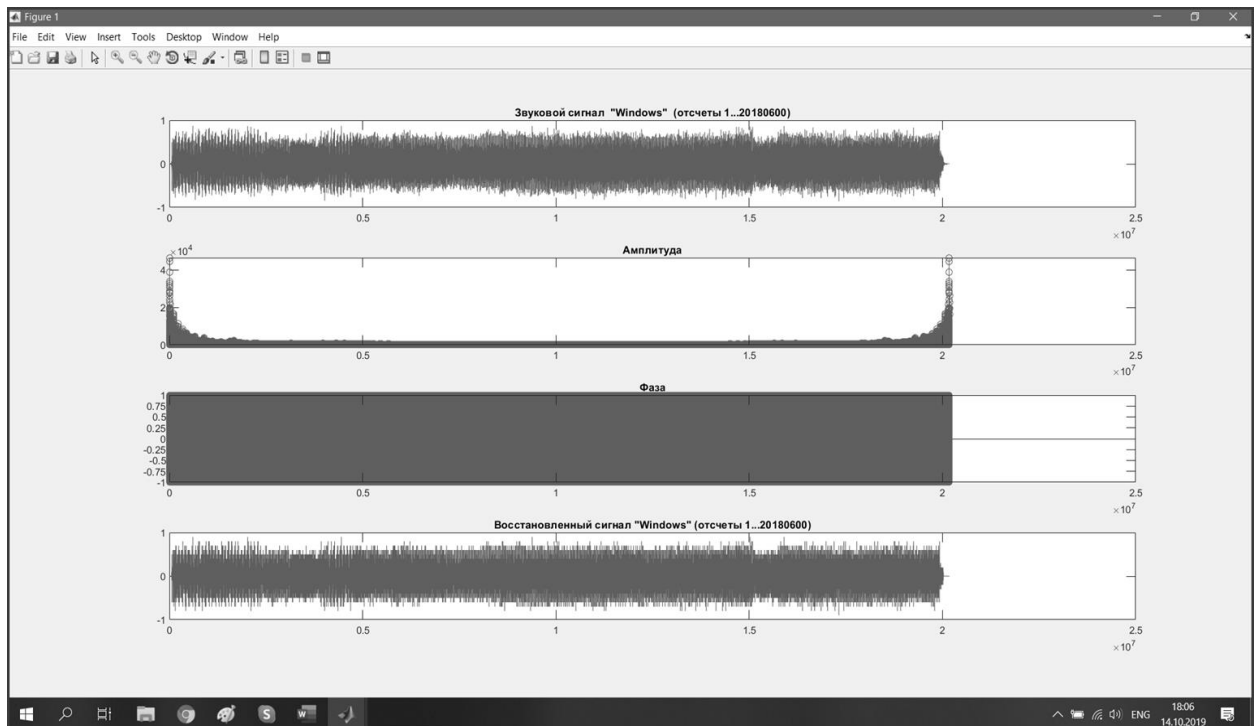
4. Виконати перетворення сигналу в цілому, відтворити звуковий сигнал до і після перетворення і порівняти звучання.

Код:

```
clear y Fs;
[y, Fs] = audioread('C:\Users\Kona\Desktop\3.flac');
N=size(y);
y1=y(:, 1);
y2=y(:, 2);
% Виділення частини сигналу
% N1=147;
% N2=200;
N1=1;
N2=20180600;
NF=N2-N1+1;
y11=roundn(y1(N1:N2),-5);
subplot(4,1,1); plot(y11);
title(['Звуковий сигнал "Windows" (відліки ',int2str(N1), '...',int2str(N2), ')']);
% Спектр каналу 1
YdFs=roundn(fft(y11,NF),-5);
rdF=real(YdFs);
idF=imag(YdFs);
mdF = abs(YdFs);
pdF = angle(YdFs)/pi;
subplot(4,1,2); stem(mdF); title('Амплітуда');
subplot(4,1,3); stem(pdF); title('Фаза');
set(gca, 'ytick', -1:0.25:1)
% ----- Відновлення -----
y111=roundn(iff(YdFs),-5);
subplot(4,1,4); plot(y111);
title(['Відновлений сигнал "Windows" (відліки ', int2str(N1), '...',
int2str(N2), ')']);
% ---- Похибка ----
delta=norm(y11-y111);

audiowrite('C:\Users\Kona\Desktop\2.flac',y11, Fs);
```

Приклад сигналу (скріншот при округленні під час відновлення до 1го знаку після коми).

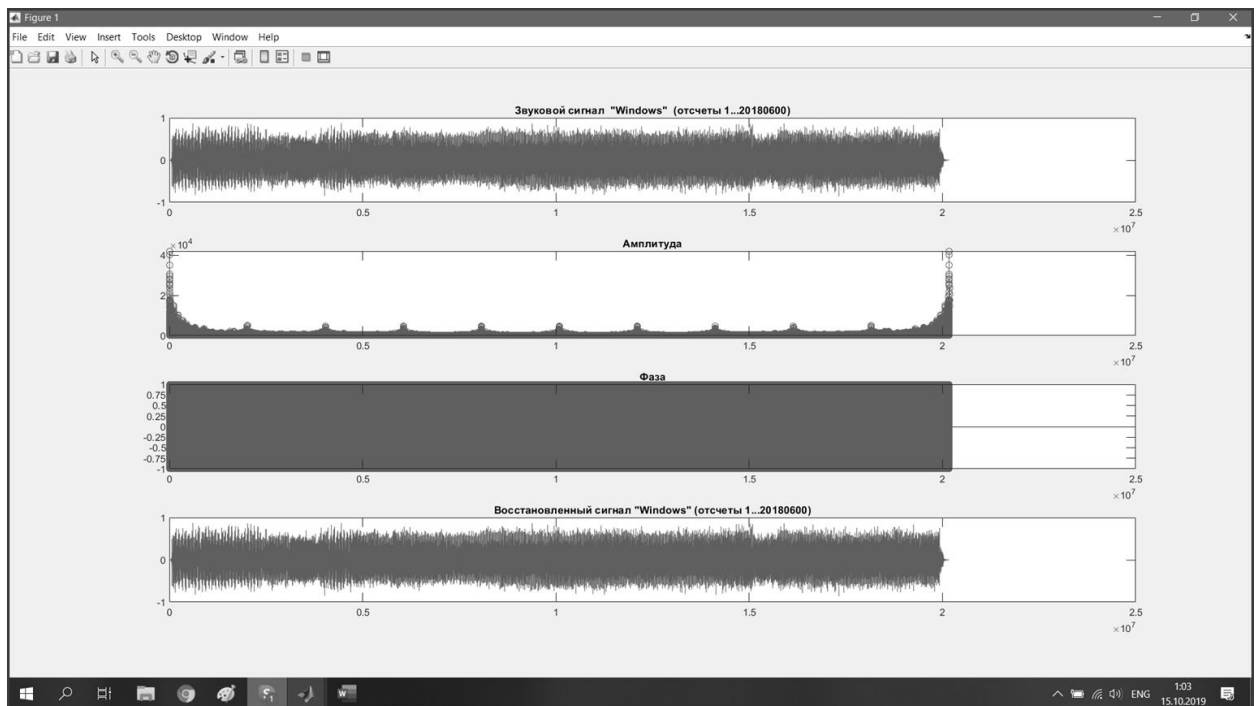


При округленні, до 5ти знаків після коми чітко відчувається погіршення якості, після відновлення, при округленні до 1го знаку після коми сигнал дуже зіпсований у порівнянні з оригіналом.

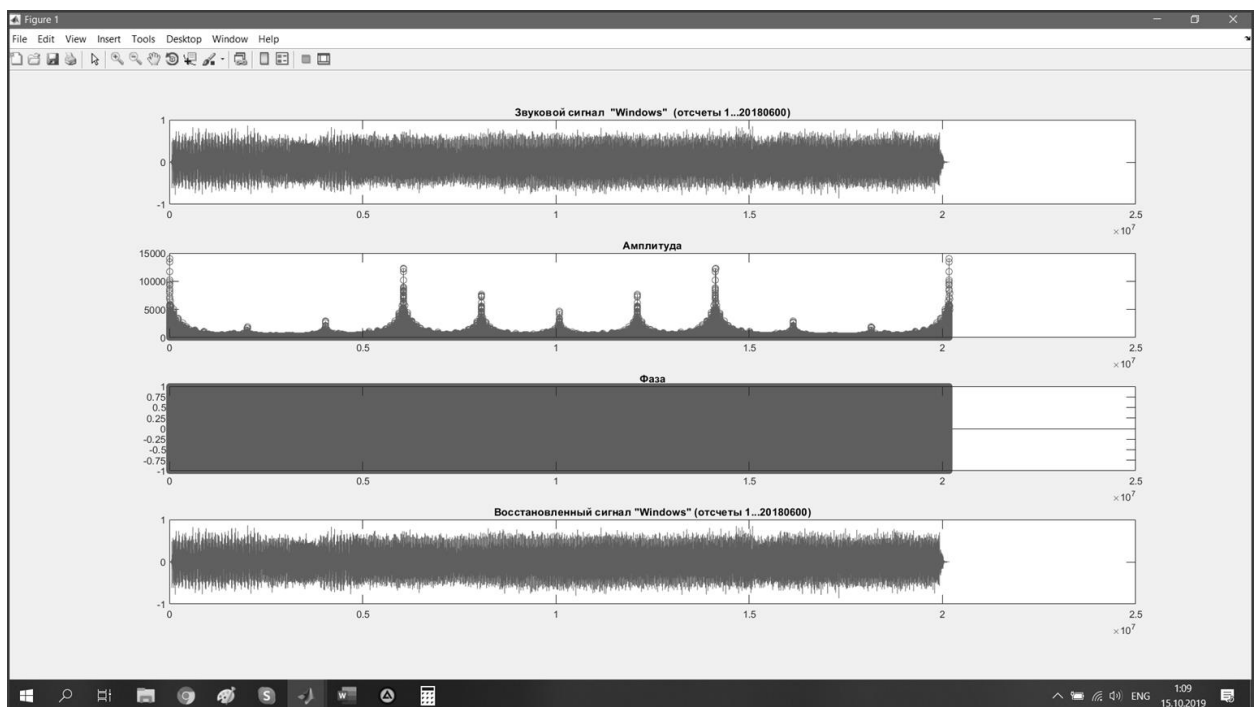
Код обнуління з заданням відсотків:

```
M=(size(y,1)/100)*70;
L=size(y,1)/M;
G=size(y,1);
L1=L-fix(L);
N=size(y);%розмір матриці

is=L1;
is2=L1;
if L>1
    is2=1-L1;
    T=fix(L);
    O=0;
    while O<G
        O=O+T;
        if O>=G
            break;
        end
        y(O,1)=0;
        y(O,2)=0;
        is2=is2+is;
        if is2>=1
            O=O+1;
            is2=is2-1;
        end
    end
end
end
```



10%



70%

Висновки: При обнулінні на 50% це найкращий варіант із всіх обнулінь, але звук дуже поганий. При всіх 12 замірах похибка відновлення була рівна нулю. У форматі flac 16біт 44,1КГц стиснення не відбулося, розмір файлу навпаки трохи збільшився.

Алгоритми цифрового опрацювання сигналів [21].

Лінійна фільтрація. Лінійна фільтрація використовується для подавлення шумів, спектр яких не перетинається зі спектром сигналу і дозволяє виділити в чистому вигляді функцію, що описує явище, яке досліджується. На практиці розрізняють два основних типи лінійних фільтрів: з скінченою (СІХ) і нескінченою (НІХ) імпульсними характеристиками. НІХ фільтри мають рекурсивну структуру, що описується різницевиими рівняннями згідно з формулою (1.1) [21].

$$y(n) = \sum_{i=0}^M b_i x(n-i) - \sum_{i=1}^M a_i y(n-i), n \geq 0; \quad (1.1)$$

де x та y - вхідна та вихідна реалізації процесу; a_i і b_i - постійні коефіцієнти, що характеризують властивості фільтра, причому $a_M \neq 0$; M – порядок фільтру.

СІХ - фільтри реалізуються на основі нерекурсивної структури, яка описується формулою (1.2), і може бути отримана з формули (1.1) якщо a_i прирівняти до нуля, $i = \overline{1, M}$. [21]

$$y(n) = \sum_{i=0}^M b_i x(n-i), n \geq 0; \quad (1.2)$$

b_i - це дискретна імпульсна характеристика системи (фільтра).

Медіанна фільтрація. Медіанна фільтрація є нелінійним способом опрацювання одновірних та двовірних послідовностей (вибірок) і використовується для зменшення рівня імпульсних шумів. В порівнянні з лінійною, медіанна фільтрація має такі переваги: зберігає різкі перепади сигналу і добре згладжує імпульсний шум. Процес виконання фільтрації цього типу виконується в три етапи: сортування вибірок реалізації в ковзаючому «вікні»; вибір середнього значення у «вікні» (медіана); заміна відліку, розташованого в середині вікна значенням медіани [21].

Дискретне перетворення Фур'є. Дискретне перетворення Фур'є (ДПФ) скінченної періодичної послідовності відліків сигналу $\{x(n)\}$, $0 \leq n \leq N-1$, визначається формулою (1.3) [12,21]:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}, k=0,1,\dots,N-1 \quad (1.3)$$

де $W_N^{nk} = e^{-j \frac{2\pi kn}{N}}$ - комплексний дискретний ортонормований базис.

Таким чином, ДПФ представляє собою множину спектральних коефіцієнтів Фур'є, що відповідають N рівномірно рознесеним частотам, починаючи від нульової і до (але не включно) частоти $2\pi/T$.

Обернене дискретне перетворення Фур'є (ОДПФ) дозволяє однозначно перейти від Фур'є-образу $X(k)$ до функції $x(n)$, визначається виразом (1.4) [21]:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn}, n=0,1,\dots,N-1; \quad (1.4)$$

Швидке перетворення Фур'є (ШПФ) є узагальненою назвою множини алгоритмів, що призначені для виконання ДПФ і ОДПФ. Основна ідея алгоритмів ШПФ полягає в розбитті процедури виконання ДПФ на $m=\log_2 N$ етапів, на кожному з яких вхідний дискретний набір розділяється на дві частини. До кожної такої частини вхідного дискретного набору, застосовується операція ДПФ в 2 рази меншої розмірності: в результаті застосувань розбиття m разів на останньому етапі, матимемо тривіальне 2-х точкове ДПФ [21].

Застосування ШПФ можливе лише тоді, коли N є складним числом. Найбільшого розповсюдження набули алгоритми ШПФ для послідовностей довжини N , що є степенем числа 2 ($N=2^m$). Загальне число операцій в алгоритмі ШПФ складає приблизно $N \log_2 N$ додавань і $0,5 N \log_2 N$ множень комплексних чисел. Алгоритми ШПФ з проріджуванням в часі і з проріджуванням по частоті мало відрізняються між собою. Тому розглянемо алгоритм ШПФ з проріджуванням в часі. Формула розкладу алгоритму, в результаті застосування якої отримується шуканий дискретний набір $X(k)$, $k=0,1,\dots,N-1$, що є зображенням вихідного дискретного набору перетворюваної функції $x(n)$, $n=0,1,2,\dots,N-1$ в Фур'є просторі визначається згідно з виразом (1.5) [21]:

$$X(k) = X_0(k) + W_N^k X_1(k), \quad k=0,1,\dots,N-1; \quad (1.5)$$

де $X_0(k)$ та $X_1(k)$ відповідно $N/2$ - точкові ДПФ послідовностей $x(2n)$ та $x(2n+1)$, $n=0, 1, \dots, (N/2-1)$.

При проріджуванні, N -точкове ДПФ зводиться до обчислення двох $N/2$ -точкових ДПФ. Рекурсивно застосовуючи вказаний спосіб розділення до перетворень меншої розмірності, отримуємо алгоритм ШПФ за основою два з проріджуванням в часі [21].

Аналізуючи формулу (1.5), і беручи до уваги, що послідовності $X_0(k)$ та $X_1(k)$, періодичні з періодом $N/2$, та $W_N^{N-k} = -W_N^k$, можна встановити, що при кожному розділенні реалізується однотипна базова операція [15]:

$$A^* = A + W_N^k B; \quad B^* = A - W_N^k B; \quad (1.6)$$

причому $k \in \{0, 1, \dots, (N/2-1)\}$, а величини A, B, A^*, B^* , і W_N^k - комплексні числа. Для виконання базової операції (1.6) необхідно виконати одне множення, одне додавання і одне віднімання комплексних величин.

Кореляція. Для визначення подібності між сигналами в різні моменти часу або виділення сигналу на фоні шуму проводять кореляційну обробку, тобто обчислення функцій взаємної кореляції двох числових послідовностей x і y , кожна з яких містить N відліків, записується у вигляді [21]:

$$R_{xy}(r) = \frac{1}{N-r} \sum_{i=1}^{N-r} X_i Y_{i+r}, \quad r = 0, 1, \dots, N-1; \quad (1.7)$$

Взаємна кореляційна функція двох сигналів обчислюється як сума їх по елементних добутків з відносною затримкою r . [21]

Автокореляційна функція записується у вигляді [18]:

$$R_{xx}(r) = \frac{1}{N-r} \sum_{i=1}^{N-r} X_i X_{i+r}, \quad r = 0, 1, \dots, N-1; \quad (1.8)$$

Обчислення автокореляційної і взаємкореляційної функцій двох сигналів складається з трьох основних операцій: часової затримки, множення і сумування [21].

Згортка сигналів. При згортці сигналів $x(n)$ та $h(m)$, де $n = 0, 1, \dots, N_1-1$, $m = 0, 1, \dots, N_2-1$, виконується обробка згідно з формулою (1.9) [21].

$$y(n) = \sum_{m=0}^n h(m)x(n-m); \quad (1.9)$$

де $h(m)$ і $x(n)$ рівні нулю поза відповідними інтервалами, $h(m)$ - дискретна імпульсна характеристика системи (фільтра).

Нормування сигналів. Одним з прикладів операції нормування сигналів є ділення комплексних чисел, що виконується згідно з формулою (1.10) [17].

$$U_H(n,i) = \frac{U(n,i)}{U'(i)} = \frac{\operatorname{Re}U(n,i)\operatorname{Re}U'(i) + \operatorname{Im}U(n,i)\operatorname{Im}U'(i)}{[\operatorname{Re}U'(i)]^2 + [\operatorname{Im}U'(i)]^2} + j \frac{\operatorname{Im}U(n,i)\operatorname{Re}U'(i) - \operatorname{Re}U(n,i)\operatorname{Im}U'(i)}{[\operatorname{Re}U'(i)]^2 + [\operatorname{Im}U'(i)]^2} \quad (1.10)$$

Сортування. В основі алгоритмів сортування лежать дві основні операції: порівняння і пересилання даних [21].

Більшості з названих алгоритмів властиві регулярність, рекурсивність і локальність, що робить їх придатними для реалізації на НВІС. Зокрема, до локально рекурсивних алгоритмів відносяться алгоритми перемноження матриць, згортки, НІХ-фільтрація, сортування вибором; до глобально-рекурсивних - алгоритми ШПФ, бітове сортування [21].

На основі аналізу алгоритмів в табл. 1.2 виділено набір базових операцій для опрацювання сигналів та зображень [21].

Таблиця 1.2 [21].

Базові операції для задач керування та опрацювання сигналів.

Алгоритм	Базові операції
Реалізація фільтрів, обчислення кореляційної і взаємкореляційної функцій	Додавання, віднімання, множення, обчислення суми парних добутків
Алгоритми перетворення	Множення, додавання комплексних чисел; обчислення операцій ШПФ, ШПХ, множення послідовностей комплексних чисел
Алгоритми обчислення: коефіцієнтів взаємної кореляції, координат, відстаней, виконання вагового множення, тощо	Обчислення тригонометричних функцій, добування квадратного кореня, піднесення до степені, ділення, сортування

НАВЧАЛЬНО-МЕТОДИЧНЕ ВИДАННЯ

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт з дисципліни

«МЕТОДИ АНАЛІЗУ ТА ДОСЛІДЖЕННЯ ДИСКРЕТНИХ СИГНАЛІВ»

до виконання лабораторних робіт

для студентів спеціальності 123 Комп'ютерна інженерія

усіх форм навчання

Комп'ютерний набір і верстка:

Ковальов Сергій Олександрович

Шамасв Віталій Віталійович

Укладачі:

Ковальов Сергій Олександрович

Шамасв Віталій Віталійович

Автори щиро вдячні Самощенко О.В. за співпрацю та надані матеріали.

ДВНЗ «Донецький національний технічний університет»

43003, м. Луцьк, вул. Потебні, 56