



НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
КОРАБЛЕБУДУВАННЯ
ІМЕНІ АДМІРАЛА МАКАРОВА



Ш. М. Іхсанов В. М. Рябенський О.С. Дьяконов

ДОСЛІДЖЕННЯ СИГНАЛІВ РЕАЛЬНИХ ІНФОРМАЦІЙНИХ СИСТЕМ З ВИКОРИСТАННЯМ ПРИЙМАЧІВ RTL-SDR

Навчальний посібник



Міністерство освіти та науки України

Національний університет кораблебудування
імені адмірала Макарова

Ш. М. Іхсанов, В. М. Рябенський, О.С. Дьяконов

**ДОСЛІДЖЕННЯ СИГНАЛІВ
РЕАЛЬНИХ ІНФОРМАЦІЙНИХ СИСТЕМ
З ВИКОРИСТАННЯМ ПРИЙМАЧІВ RTL-SDR**

Навчальний посібник

Миколаїв
2018

УДК 654.1:621.396.7
ББК 32.884.8-5
І95

Рецензенти:

Докт. технічних наук, професор Лісовий Іван Павлович
Докт. технічних наук, професор Фісун Микола Тихонович

*Схвалено Вченою Радою НУК
Рекомендовано до друку в якості навчального посібника
(протокол № 6 від 27.06 2018 р.)*

І95

Іхсанов Ш. М., Рябенський В. М., Дьяконов О.С.

І95 Дослідження сигналів реальних інформаційних систем з використанням приймачів RTL-SDR : навч. посіб. / Ш. М. Іхсанов, В. М. Рябенський, О. С. Дьяконов. — Миколаїв, 2019. — 182 с.

ISBN 978-617-7421-61-5

Посібник присвячений використанню в навчальному процесі приймачів RTL-SDR, які дозволяють використовувати замість цифрових моделей реальні сигнали різних інформаційних систем.

В першому розділі досліджуються сигнали FM-радіостанцій, аналізуються алгоритми їх обробки. Наведена цифрова модель сигналів FM-радіостанцій і показана її відповідність реальним сигналами.

Другий розділ присвячений дослідженню бортових сигналів повітряних суден в технології ADS-B. Детально розглянуті методи передачі бортових даних, алгоритми їх виділення із зазначенням напрямів можливого вдосконалення, наведені статистичні дані по прийому бортових сигналів.

В третьому розділі розглянута система автоматичної ідентифікації морських суден (AIS). Наведено теоретичні та експериментальні дані по сигналам GMSK, що використовуються в системі для передачі інформації. Наведено результати виявлення сигналів AIS в акваторії Бузького лиману. Оцінені характеристики алгоритму виділення бортових сигналів корпорації MathWorks, експериментально показано, що його вдосконалення призводить до збільшення частоти виявлення сигналів AIS більш ніж в 3 рази.

В четвертому розділі розглянуто алгоритм і програмна реалізація демодулятора QPSK-сигналів. Розглянуті характеристики алгоритму при обробці моделюємої і реальної супутникової інформації.

Наведені індивідуальні завдання з лабораторних та курсових робіт, а також теми курсових проектів. Програмна реалізація алгоритмів прив'язана до MATLAB версії R2015a і вище.

Посібник рекомендується для студентів ЗВО напряму «Електроніка та телекомунікації».

ISBN 978-617-7421-61-5

**УДК 654.1:621.396.7
ББК 32.884.8-5**

© Іхсанов Ш. М., Рябенський В.
М., Дьяконов О.С., 2019
© ФОП Швець В.М., 2019

ВСТУП	5
1. ПРИЙОМ І ОБРОБКА СИГНАЛІВ FM-РАДІОСТАНЦІЙ.....	11
1.1. Історія та теоретичні основи FM-сигналів	11
1.2. Дискримінантний алгоритм виділення FM-сигналів.....	14
1.3. Короткий огляд основ теорії фільтрації сигналів	18
1.4. Цифрова модель сигналів FM-радіостанцій.....	28
1.5. Вдосконалення некогерентного дискримінантного алгоритму обробки FM-сигналів	36
1.6. Завдання для лабораторних, курсових робіт і дипломних проектів.....	44
2. ВІДСТЕЖЕННЯ ТРАЄКТОРІЙ ПОЛЬОТУ ПОВІТРЯНИХ СУДЕН	48
2.1. Загальна інформація про методи передачі повідомлень повітряними суднами	48
2.2. Структура повідомлень, що передаються	49
2.3. Інформація про місцезнаходження повітряного судна	54
2.4. Інформація про швидкість повітряного судна	57
2.5. Алгоритм виявлення бортових сигналів повітряних суден.....	61
2.6. Огляд експериментів по прийому бортових сигналів	78
2.7. Завдання для лабораторних, курсових робіт і дипломних проектів.....	84
3. ВІДСТЕЖЕННЯ РУХУ МОРСЬКИХ СУДЕН.....	87
3.1. Загальні відомості про GMSK сигнали	87
3.2. Дослідження сигналів GMSK в MATLAB.	94
3.3. Загальна інформація про методи передачі повідомлень морськими судами	110
3.4. Дослідження реальних сигналів AIS	113
3.4.1. Алгоритм виявлення сигналів AIS	113
3.4.2. Дослідження записів сигналів AIS від судів в околиці м. Бостона.....	115
3.4.3. Дослідження сигналів судів в акваторії Бузького лиману.....	146
3.5. Завдання для лабораторних, курсових робіт і дипломних проектів.....	159
4. ДОСЛІДЖЕННЯ РЕАЛЬНИХ СУПУТНИКОВИХ СИГНАЛІВ	162
4.1. Стандарт передачі Інтернет-даних через супутникові канали зв'язку.....	162
4.2. Короткий опис запису супутникового сигналу	169
4.3. Математична модель сигналів з фазовою модуляцією	169
4.4. Результати порівняння реального супутникового сигналу з моделлю QPSK-сигналів.....	173
4.5. Завдання для лабораторних, курсових робіт і дипломних проектів.....	180
ОСНОВНІ СКОРОЧЕННЯ.....	181
ЛІТЕРАТУРА	182

ВСТУП

Абревіатура RTL-SDR розшифровується як Real Time Desktop Software Defined Radio – українською перекладається приблизно як «Програмно-конфігурована радіосистема (радіо) реального часу». Насамперед, це приймач, що приймає сигнали в дуже широкому діапазоні частот, а саме від 25 МГц до 1,75 ГГц. Далі прийнятий сигнал переводиться в цифровий по обох квадратурних каналах за допомогою високошвидкісного 8-розрядного АЦП з частотою до 2,8 МГц (частота квантування може бути доведена до 3,2 МГц, проте не рекомендується перевищувати величину 2,8 МГц, оскільки це гарантує відсутність втрат даних). На рисунку 1 зображені деякі види приймача і приймальна антена:

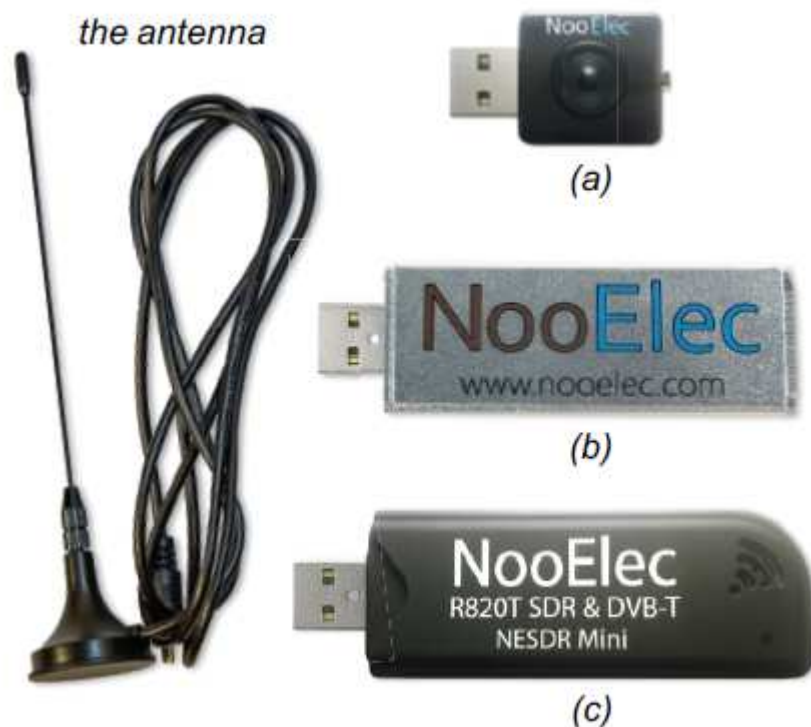


Рис. 1. Деякі види приймача RTL-SDR та антена

Як видно з рисунка, приймач стикується з ПК по USB-порту (USB-2.0).

Зазначені характеристики приймача дозволяють приймати сигнали багатьох служб. На рисунку 2 приведені деякі зі служб Великобританії (приблизно таке ж завантаження частот використовується на території України та Росії), радіосигнали яких можуть бути прийняті на приймач RTL-SDR:

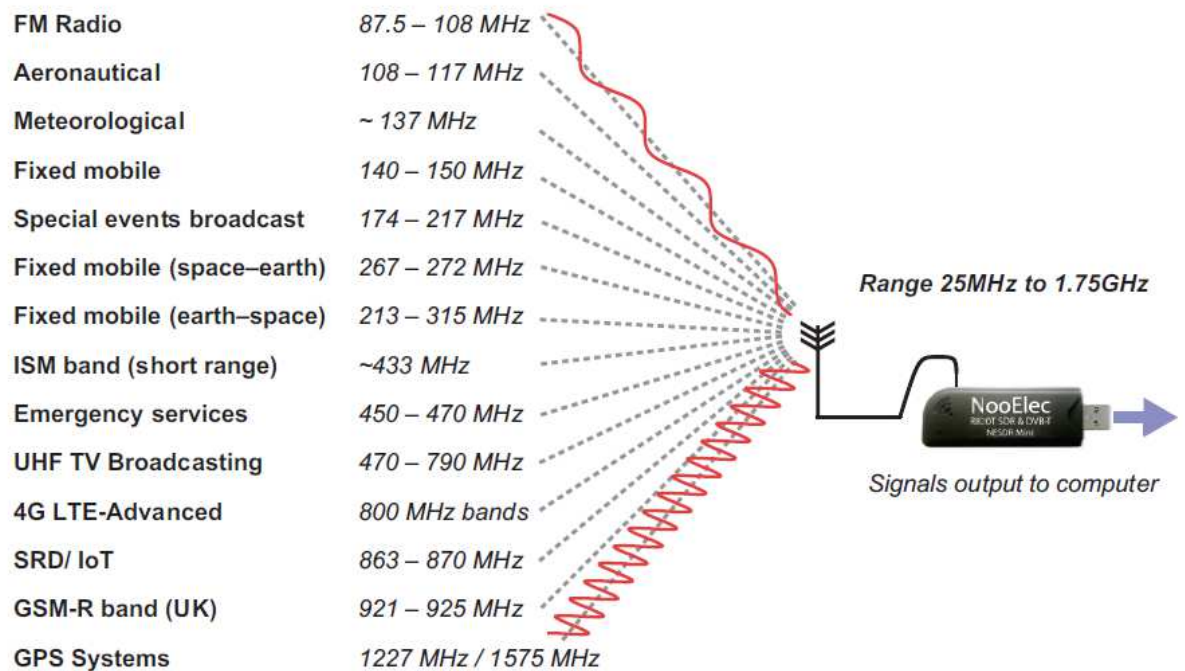


Рис. 2. Служби Великобританії, радіосигнали яких можуть бути прийняті на приймач RTL-SDR

Після установки драйверів приймача і забезпечення можливості подачі сигналів приймача в MATLAB ми можемо починати аналізувати програми по обробці різних сигналів. Наведемо текст програми, що забезпечує прийом квадратурної інформації в заданий масив:

```
%% Функція для приймання інформації з RTL-SDR

function ReceiveFromRTLSDR
%% PARAMETERS (edit)
rtlsdr_id      = '0';      % stick ID - порт приймача
rtlsdr_fc      = 623.225e6; % tuner centre frequency in Hz -
                           % центральна частота приймача в Гц
rtlsdr_gain    = 60;      % tuner gain in dB -
                           % підсилення приймача в дБ
rtlsdr_fs      = 2.4e6;    % tuner sampling rate, max 3.2e6 -
                           % частота квантування сигналу в Гц
rtlsdr_ppm     = 0;      % tuner parts per million correction
                           % - параметр, який використовується при
                           % додатковій корекції частоти
rtlsdr_frmrlen = 5*256;   % output data frame size(multiple of 5)
                           % - кількість комплексних відліків
                           % при одній видачі
rtlsdr_datatype='single'; % output data type - тип даних, що
                           % видаються, аналог float (4 байта)

%% SYSTEM OBJECTS (do not edit)

% link to a physical rtl-sdr
```

```

% підключаємося до фізичного приймача:
obj_rtlsdr = comm.SDRRTLReceiver(...
    rtlsdr_id,...
    'CenterFrequency', rtlsdr_fc,...
    'EnableTunerAGC', false,...
    'TunerGain', rtlsdr_gain,...
    'SampleRate', rtlsdr_fs, ...
    'SamplesPerFrame', rtlsdr_frmlen,...
    'OutputDataType', rtlsdr_datatype,...
    'FrequencyCorrection', rtlsdr_ppm);

% Перевіряємо активність RTL-SDR:
if isempty(sdrinfo(obj_rtlsdr.RadioAddress))
    error(['RTL-SDR failure. Please check connection to ',...
        'MATLAB using the "sdrinfo" command.']);
end

% Створюємо спектр-аналізатор:
h = dsp.SpectrumAnalyzer('SampleRate',rtlsdr_fs);

% calculate time for 1 frame of data – розраховуємо час
% прийому однієї порції (одного фрейму) інформації:
rtlsdr_frmtime = rtlsdr_frmlen/rtlsdr_fs;

% set rec_time and reset run_time to 0 (secs):
rec_time = 3600; % час роботи програми
run_time = 0;

f = figure; % вікно для програми

% Create a constellation diagram and set the SamplesPerSymbol
% property – створення об'єкту для відображення сигнального
% сузір'я (фазового портрета сигналу) та завдання кількості
% відліків на символ (при Rup=1 всі комплексні відліки
% використовуються для відображення, при Rup=2 – через 1 і т.д.)
Rup = 1;
hScope = comm.ConstellationDiagram('SamplesPerSymbol',Rup);

% loop while run_time is less than rec_time – цикл до закінчення
% заданого часу роботи програми:
while run_time < rec_time

    % fetch a frame from obj_rtlsdr –
    % отримання однієї порції інформації:
    rtlsdr_data = step(obj_rtlsdr);

    % відображення квадратур початкового сигналу:
    Nn=1; % Начальна точка на відображення
    dN=245; % кількість точок на відображення
    Nk=Nn+dN;
    t=Nn:Nk;

```

```

plot(t, rtl_sdr_data(Nn:Nk), t, imag(rtl_sdr_data(Nn:Nk)));

% відображення спектру сигналу:
step(h, rtl_sdr_data);

% Display the constellation diagram of the received signal –
% - відображення поточного сигнального сузір'я:
step(hScope, rtl_sdr_data);

% update run_time after processing another frame –
% збільшення поточного часу на час отримання однієї
% порції інформації:
run_time = run_time + rtl_sdr_frmttime;

% Пауза на задане число секунд
% (можна задавати частки секунди)
pause(1.0);

% Переривання роботи програми після натискання клавіші Q при
% активному вікні програми (яке створено командою f =
figure):
if get(f, 'CurrentCharacter') == 'q'
    break;
end
end
close(f);
end

```

У програмі наведено докладне роз'яснення функцій, що використовуються. Додатково зазначимо одну з угод MATLAB по можливості розміщення аргументів функції в кількох рядках. На відміну від мови C++, в якій можна будь-які вирази переносити на новий рядок (винятком є значення рядкових змінних, що укладені в подвійні або одинарні лапки), в MATLAB перед перенесенням необхідно поставити три крапки.

Це зроблено у наведеній програмі при з'єднанні та видачі на фізичний приймач RTL-SDR керуючих команд функцією `comm.SDRRTLReceiver()`. Крім прийому квадратурної інформації з приймача у програмі наведені функції для відображення квадратур у часі, спектра сигналу в смузі частот `rtl_sdr_fs` (2,4 МГц) і сигнального сузір'я (фазового портрета сигналу). Реалізації перерахованих даних наведені для сигналу з радіомовного діапазону на частоті 623,225 МГц:

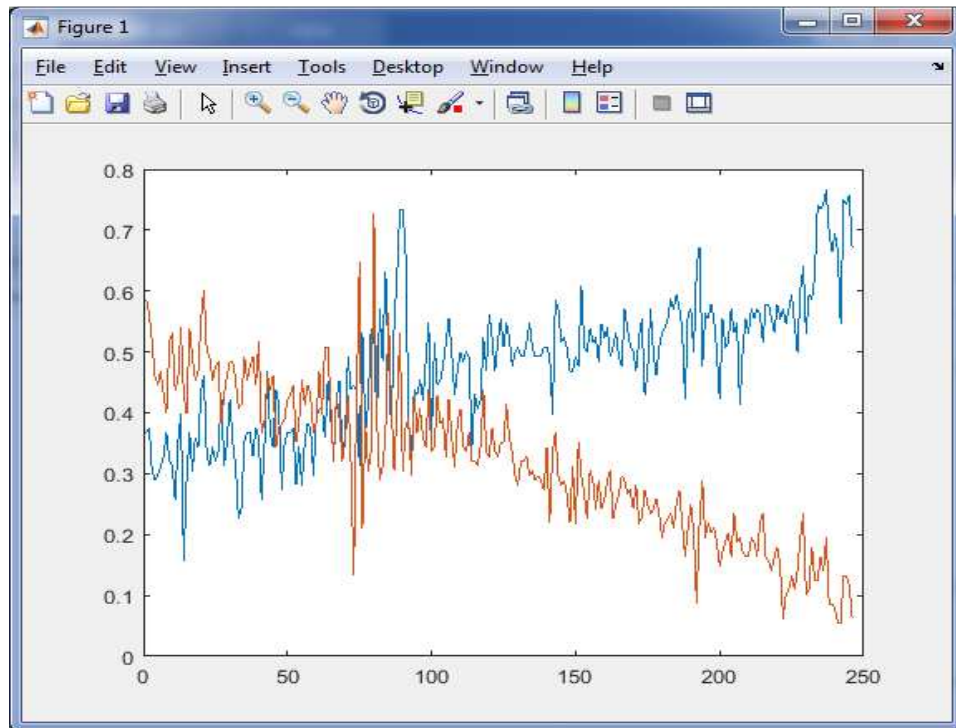


Рис. 3. Приклад інформації в квадратурних каналах

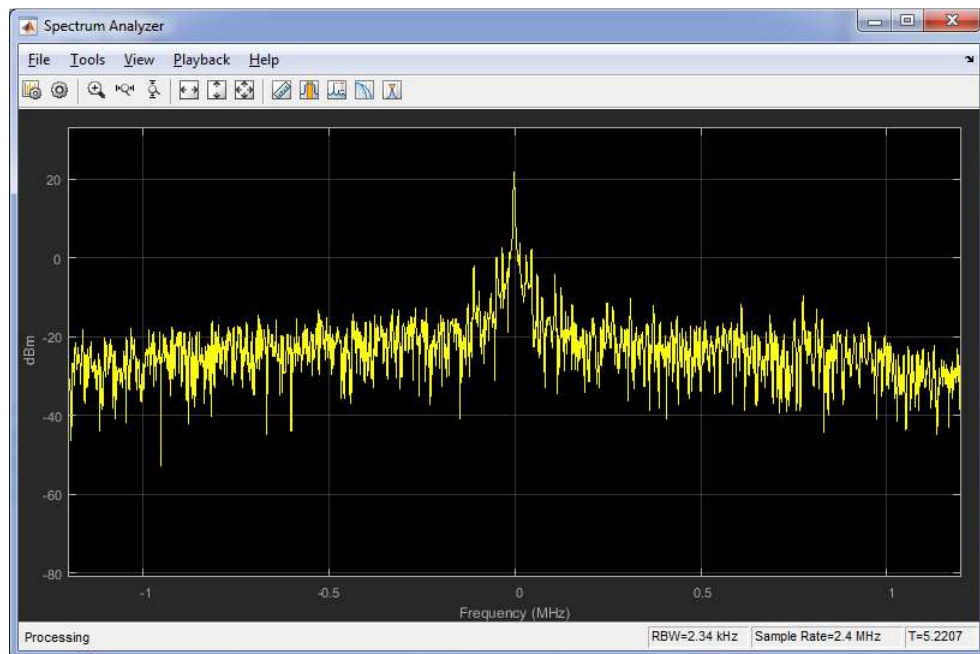


Рис. 4. Приклад спектра сигналу

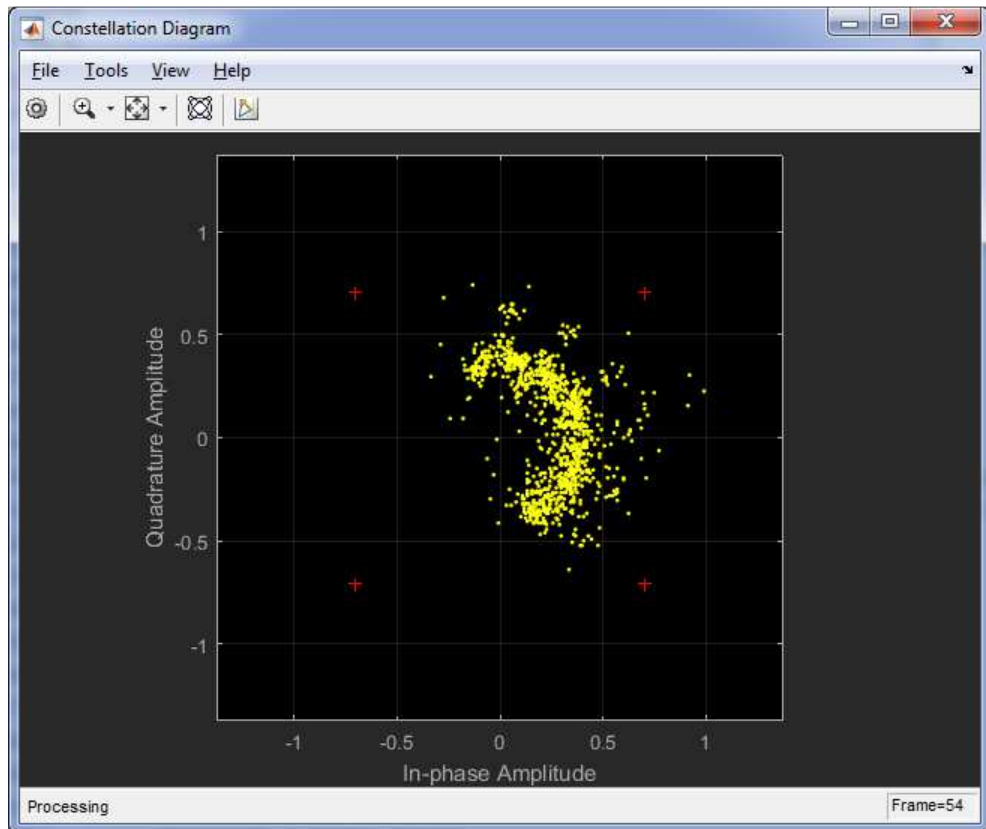


Рис. 5. Приклад сигнального сузір'я

1. ПРИЙОМ І ОБРОБКА СИГНАЛІВ FM-РАДІОСТАНЦІЙ

1.1.	Історія та теоретичні основи FM-сигналів	11
1.2.	Дискримінантний алгоритм виділення FM-сигналів	14
1.3.	Короткий огляд основ теорії фільтрації сигналів	18
1.4.	Цифрова модель сигналів FM-радіостанцій.....	28
1.5.	Вдосконалення некогерентного дискримінантного алгоритму обробки FM сигналів	36
1.6.	Завдання для лабораторних, курсових робіт і дипломних проектів.....	44

1.1. Історія та теоретичні основи FM-сигналів

Докладний аналіз різних реальних сигналів і алгоритмів їх обробки почнемо з сигналів FM-радіостанцій. Спочатку коротко зупинимося на принципах роботи FM-станцій. Аббревіатура FM взята від поєднання Frequency Modulation — частотна модуляція. Не плутати з режимом FSK – (Frequency Shift Keying) для передачі цифрової інформації.

Режим FM був запропонований американським інженером Едвіном Армстронгом (Edwin Howard Armstrong) в 1933 році як результат його зусиль по боротьбі зі статичними перешкодами для режиму амплітудної модуляції (AM – amplitude modulation), що вже широко використовувався в той час. Він відкрив, що при модулюванні несучої частоти замість амплітуди випромінений сигнал стає набагато більш стійким до впливу завад та сигналів ближніх радіостанцій. Отримавши 4 патенти на свій винахід, він почав роботи по створенню мережі FM-радіостанцій. В горах Нью-Джерсі Армстронг побудував ажурну вежу висотою 400 футів (122 м). Вежа була побудована в 1937 році і досі експлуатується!

Найбільш проста схема генерування FM-сигналу виходить в режимі VCO (voltage-controlled oscillator – генератор, керований напругою) (рис. 1.1):

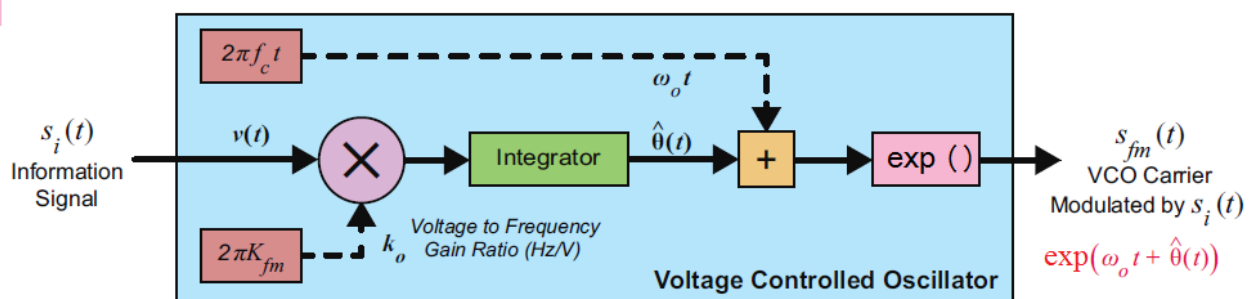


Рис. 1.1. Схема генерування FM-сигналу в режимі VCO

В аналітичному вигляді такий FM-сигнал записується наступним чином:

$$s_{fm}(t) = A_0 \exp \left(j \left(2\pi f_c t + 2\pi K_{fm} \int_{-\infty}^t s_i(\tau) d\tau \right) \right) \quad (1.1)$$

Тут f_c – несуча радіочастота, K_{fm} – коефіцієнт девіації частоти, $s_i(t)$ – інформаційний (звуковий) сигнал. Зверніть увагу, що модуляція несучої частоти проводиться не самим звуковим сигналом, а інтегралом від нього.

За еталон цифрової обробки сигналів FM-радіостанцій візьмемо програму SDRSharp. Опис елементів управління і налаштувань SDRSharp [1] показує широкі можливості програми і складність застосовуваних алгоритмів обробки.

У північно-східній частині Миколаєва ($46,96596^0$, $32,07628^0$), де автори проводили експерименти з приймачем SDR-RTL, програма SDRSharp при підключенні до приймача маленької антени, зображеної на рис. 1, забезпечує хорошу якість звуку по 8-ми FM-радіостанціям (таблиця 1.1):

Таблиця 1.1.

№ з/п	Частота FM-станції (МГц)	Назва
1	99,2	Мелодія
2	102,1	Kiss FM
3	102,8	Наше радіо
4	103,3	Авто радіо
5	104,1	Стильное радио
6	105,1	Радио Пятница
7	106,4	Ретро FM
8	107,8	Ера FM

В частотній області в районі 102 МГц сигнали FM-станцій виглядають таким чином (рис 1.2):

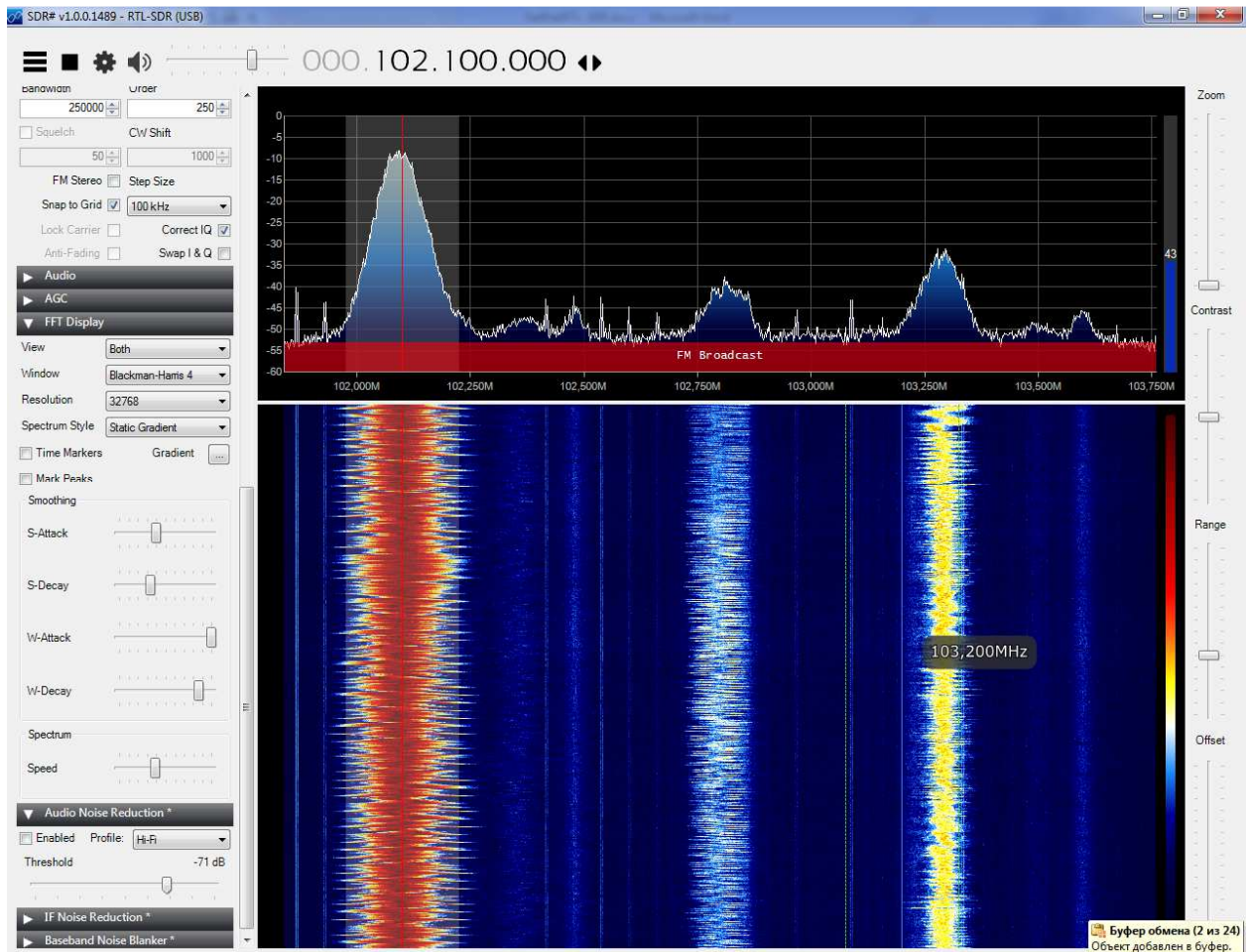


Рис. 1.2. Спектри FM-радіостанцій в програмі SDRSharp

На самому початку діапазону ми бачимо дуже потужний сигнал від **Kiss FM** на частоті 102,1 МГц, далі досить слабкий сигнал від **Нашого радіо** на частоті 102,8 МГц і трохи кращий сигнал від **Авто радіо** на частоті 103,3 МГц. У нижній частині спектр відображається в так званому режимі Waterfall (водоспад), в якому вісь часу йде вниз, а спектральна потужність кожної частоти відображається кольором відповідно наведеною в правій частині шкали. Зазначимо, що сигнали FM-радіостанцій займають смугу до 250 кГц (освітлена смуга на сигналі Kiss FM), відповідно їх не можна розміщувати ближче. Дальність розповсюдження хвиль в діапазоні FM, за винятком рідкісних випадків аномального поширення радіохвиль, строго обмежена так званою «прямою видимістю». Вона залежить від висоти розташування передавальної і приймальної антен. Зрозуміло, що в умовах міської забудови ні про яку пряму видимість говорити не можна, але радіохвилі проходять крізь міські забудови з деяким ослабленням. Чим вище частота, тим вище загасання в міських забудовах. Тому зрозуміло, що необхідно в кожному регіоні мати ретранслятори FM-радіостанцій, при цьому їх частоти в різних регіонах, як правило, відрізняються.

1.2. Дискримінантний алгоритм виділення FM-сигналів

Обробку сигналів FM-радіостанцій почнемо з досить простого некогерентного дискримінантного алгоритму для виділення моносигналу. Програму наведено у відкритому вигляді у бібліотеці файлів чудової книги [2]:

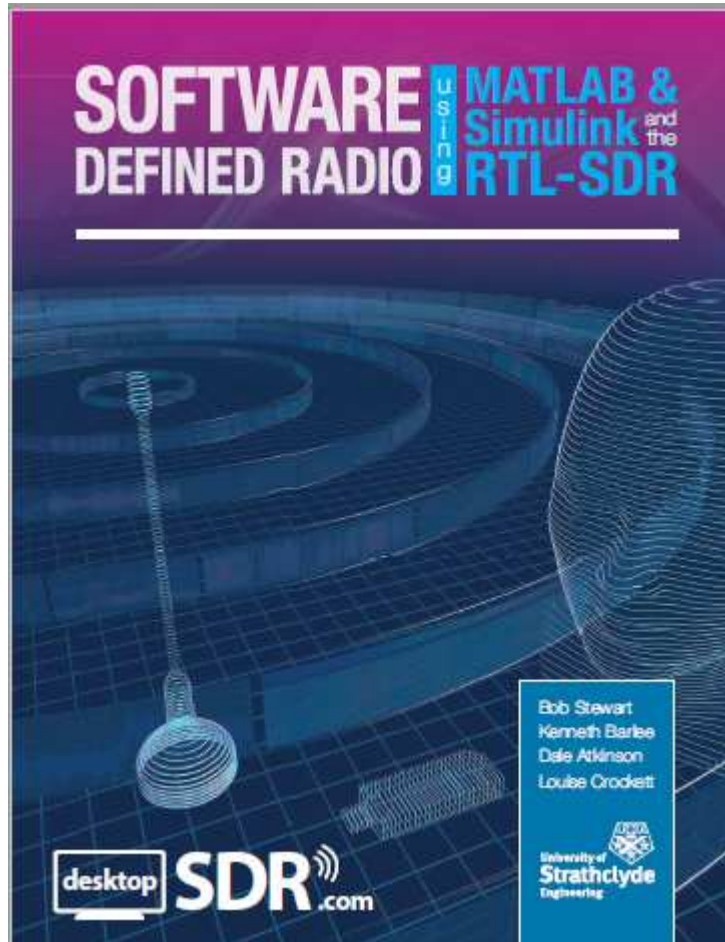


Рис. 1.3. Обкладинка однієї з кращих книг по обробці сигналів приймачів RTL-SDR

Наведемо повний текст програми:

```
% RTL-SDR(rx) FM Mono Non Coherent Discriminator Demodulator
% MATLAB Script

function rtlcdr_fm_discrim_demod_matlab

%% PRINT FILE INFORMATION HYPERLINK TO COMMAND WINDOW
disp(['View file information for <a href="matlab: mfileinfo('',mfilename, '')">',mfilename,'</a>']);

%% PARAMETERS (edit)
offline           = 0;           % 0 = use RTL-SDR, 1 = import data
offline_filepath = 'rec_data\wfm_mono.mat'; % path to signal
rtlsdr_id         = '0';         % stick ID
rtlsdr_fc         = 102.8e6; % tuner centre frequency in Hz
```

```

rtlsdr_gain      = 50;          % tuner gain in dB
rtlsdr_fs        = 2.4e6;      % tuner sampling rate
rtlsdr_ppm       = 0;          % tuner parts per million correction
rtlsdr_frmlen    = 256*25;     % output data frame size
                                % (multiple of 5)
rtlsdr_datatype  = 'single';   % output data type
deemph_region    = 'eu';       % set to either eu or us
audio_fs         = 48e3;       % audio output sampling rate
sim_time         = 3600;       % simulation time in seconds

%% CALCULATIONS (do not edit)
% calculate time for 1 frame of data:
rtlsdr_frmtime = rtlsdr_frmlen/rtlsdr_fs;
% find de-emphasis filter coeff
if deemph_region == 'eu'
    [num,den] = butter(1,3183.1/(audio_fs/2));
elseif deemph_region == 'us'
    [num,den] = butter(1,2122.1/(audio_fs/2));
else
    error('Invalid region for de-emphasis filter -
          must be either "eu" or "us"');
end

%% SYSTEM OBJECTS (do not edit)

% check if running offline
if offline == 1

    % link to an rtl-sdr data file
    obj_rtlsdr = import_rtlsdr_data(...
        'filepath', offline_filepath,...
        'frm_size', rtlsdr_frmlen,...
        'data_type',rtlsdr_datatype);

    % reduce sampling rate
    rtlsdr_fs = 240e3;

    % fir decimator - fs = 240kHz downto 48kHz
    obj_decmttr = dsp.FIRDecimator(...
        'DecimationFactor', 5,...
        'Numerator',
        firpm(100,[0,15e3,20e3,(240e3/2)]/(240e3/2),...
            [1 1 0 0], [1 1], 20));

else

    % link to a physical rtl-sdr
    obj_rtlsdr = comm.SDRRTLReceiver(...
        rtlsdr_id,...
        'CenterFrequency', rtlsdr_fc,...

```

```

        'EnableTunerAGC', false,...
        'TunerGain', rtl_sdr_gain,...
        'SampleRate', rtl_sdr_fs, ...
        'SamplesPerFrame', rtl_sdr_frm_len,...
        'OutputDataType', rtl_sdr_datatype,...
        'FrequencyCorrection', rtl_sdr_ppm);

    % fir decimator - fs = 2.4MHz downto 48kHz
    obj_decmt = dsp.FIRDecimator(...
        'DecimationFactor', 50,...
        'Numerator',
        firpm(350, [0, 15e3, 48e3, (2.4e6/2)]/(2.4e6/2), ...
            [1 1 0 0], [1 1], 20));

end;

% iir de-emphasis filter
obj_deemph = dsp.IIRFilter(...
    'Numerator', num,...
    'Denominator', den);

% delay
obj_delay = dsp.Delay;

% audio output
obj_audio = dsp.AudioPlayer(audio_fs);

% spectrum analyzers
obj_spectrummod = dsp.SpectrumAnalyzer(...
    'Name', 'Spectrum Analyzer Modulated',...
    'Title', 'Spectrum Analyzer Modulated',...
    'SpectrumType', 'Power density',...
    'FrequencySpan', 'Full',...
    'SampleRate', rtl_sdr_fs);
obj_spectrumdemod = dsp.SpectrumAnalyzer(...
    'Name', 'Spectrum Analyzer Demodulated',...
    'Title', 'Spectrum Analyzer Demodulated',...
    'SpectrumType', 'Power density',...
    'FrequencySpan', 'Full',...
    'SampleRate', audio_fs);

%% SIMULATION

% if using RTL-SDR, check first if RTL-SDR is active
if offline == 0
    if ~isempty(sdrinfo(obj_rtl_sdr.RadioAddress))
    else
        error(['RTL-SDR failure. Please check connection to
', ...
            'MATLAB using the "sdrinfo" command.']);
    end
end

```

```

end

% reset run_time to 0 (secs)
run_time = 0;

% loop while run_time is less than sim_time
while run_time < sim_time

    % fetch a frame from obj_rtlsdr (live or offline)
    rtlsdr_data = step(obj_rtlsdr);

    % update 'modulated' spectrum analyzer window with new data
    step(obj_spectrummod, rtlsdr_data);

    % implement frequency discriminator
    discrim_delay = step(obj_delay, rtlsdr_data);
    discrim_conj   = conj(rtlsdr_data);
    discrim_pd     = discrim_delay.*discrim_conj;
    discrim_arg    = angle(discrim_pd);

    % decimate + de-emphasis filter data
    data_dec = step(obj_decmtr, discrim_arg);
    data_deemph = step(obj_deemph, data_dec);

    % update 'demodulated' spectrum analyzer window with new data
    step(obj_spectrumdemod, data_deemph);
    % output demodulated signal to speakers
    step(obj_audio, data_deemph);

    % update run_time after processing another frame
    run_time = run_time + rtlsdr_frmttime;

end
end

```

Перш ніж детально аналізувати алгоритм обробки, подивимося, наскільки програма успішно декодує сигнали наших FM-радіостанцій.

З заданими параметрами навіть найпотужніша станція прослуховується погано, але причина тут не в обробці, а в недостатній швидкодії комп'ютера авторів. Відключення обчислення та відображення обох спектрів (вхідного сигналу і демодульованого звуку) або збільшення буфера прийому в 6 разів призводить до нормальної якості. Однак це тільки для двох, найбільш потужних станцій, інші станції або взагалі не чути, або трохи чути, але на тлі сильних перешкод.

Розглянемо сам алгоритм обробки. Він досить простий. Прийняті з приймача комплексні відліки затримуються на один часовий такт (при зазначених параметрах, з урахуванням зауваження у попередньому абзаці, за один

раз приймається 38400 комплексних відліків, з інтервалом $\Delta t = 1/(2,4 \text{ МГц}) = 0,417 \text{ мкс}$. Для цього попередньо створюється екземпляр об'єкта `dsp.Delay()`, і при обробці створений об'єкт викликається командою:

```
discrim_delay = step(obj_delay,rtlsdr_data) .
```

Функція `Delay()` дозволяє задавати величину затримки, за замовчуванням зсув робиться на один такт. Далі за прийнятим масивом створюється комплексно-сполучений йому масив функцією `conj()` і здійснюється комплексне покоординатне множення двох масивів (перед операцією множення стоїть крапка).

Розглянемо, що при цьому відбудеться з сигналом, заданим формулою 1.1:

$$\begin{aligned} s_{\text{оброб}}(t_k) &= A_0 \exp\left(j\left(2\pi f_c t_k + 2\pi K_{fm} \int_{-\infty}^{t_k} s_i(\tau) d\tau\right)\right) \times \\ &A_0 \exp\left(-j\left(2\pi f_c t_{k-1} + 2\pi K_{fm} \int_{-\infty}^{t_{k-1}} s_i(\tau) d\tau\right)\right) = \\ &A_0^2 \exp\left(j\left(2\pi f_c \Delta t + 2\pi K_{fm} \int_{t_{k-1}}^{t_k} s_i(\tau) d\tau\right)\right). \end{aligned} \quad (1.2)$$

Зазначимо насамперед, що несуча радіочастота f_c залишена тут тільки як величина неузгодженості частот передавача і приймача. Після прийому сигналу на частоті радіостанції сигнал зміщується на нульову частоту, як це видно по спектру сигналу на рис. 4. Перший доданок під експонентою у підсумковій частині формули 1.2 є константою, причому дуже маленькою, і може не враховуватися. Оскільки амплітуда звуку на інтервалі квантування Δt змінюється на мізерну величину, інтеграл у другому доданку $\int_{t_{k-1}}^{t_k} s_i(\tau) d\tau = s_i t_k \Delta t$, і остаточно на цьому етапі обробки ми маємо сигнал:

$$s_{\text{оброб}}(t_k) \approx A_0^2 \exp(j 2\pi K_{fm} s_i(t_k) \Delta t).$$

Фаза цього сигналу і є наш початковий звук з деяким коефіцієнтом посилення і дуже високою частотою квантування. Тому на наступному кроці функцією `angle()` в програмі здобувається шукана фаза. У більшості випадків для оцифрування звуку використовується частота квантування 48 кГц. При вибраних параметрах нам треба знизити частоту квантування в 50 разів (2,4 МГц/48 кГц). Обробку по зниженню частоти називають децимацією. Зазвичай її об'єднують з фільтрацією сигналу.

1.3. Короткий огляд основ теорії фільтрації сигналів

З урахуванням вивчення теорії фільтрації з інших дисциплін, наведемо лише основні відомості цієї теорії.

У радіотехніці в основному широке застосування знайшли лінійні фільтри, для яких розроблені ефективні методи оптимізації їхніх характеристик. Лінійні фільтри поділяються на два основних класи: *фільтри зі скінченною імпульсною характеристикою* (СІХ-фільтри), в англійській термінології *filters with finite impulse response (FIR Filters)*. Перетворення вхідного сигналу $x(t)$ СІХ-фільтром задається формулою:

$$y(n) = \sum_{i=0}^P b_i x(n-i), \quad (1.3)$$

де P – порядок фільтра.

Другий клас фільтрів – це фільтри з *нескінченною імпульсною характеристикою* (НІХ-фільтри), в англійській термінології *filters with infinite impulse response (IIR Filters)*. Перетворення вхідного сигналу $x(t)$ НІХ-фільтром задається формулою:

$$y(n) = \sum_{i=0}^P b_i x(n-i) - \sum_{k=1}^{Q+1} a_k y(n-k), \quad (1.4)$$

де P – як і досі порядок фільтра, Q – порядок зворотного зв'язку.

Як ми бачимо з назв функцій і коментарів, у наведеній програмі обробки ФМ-сигналів використовуються обидва типи фільтрів.

Докладно розглянемо вибраний фільтр. Задається він при створенні об'єкта дециматор (`obj_decmttr`) функцією `dsp.FIRDecimator()`, першим аргументом якого є `DecimationFactor` (у скільки разів необхідно понизити частоту квантування, в нашому випадку в 50 разів), другим аргументом є вектор коефіцієнтів СІХ-фільтра. СІХ-фільтр розраховується функцією `firpm()`. Ця функція для синтезу фільтра використовує алгоритм Паркса-Макклеллана (*Parks-McClellan algorithm*). Даний метод синтезу фільтрів заснований на алгоритмі заміни Ремеза і теорії апроксимації Чебишева. Його суть полягає в ітераційному пошуку фільтру, АЧХ якого максимально точно співпадає із заданою АЧХ. Ступінь збігу в даному випадку визначається максимальним (по модулю) відхиленням АЧХ фільтра від заданої АЧХ. Таким чином, тут вирішується мінімаксна оптимізаційна задача – мінімізується максимальне абсолютне відхилення АЧХ конструйованого і заданого фільтрів.

Фільтри, синтезовані таким методом, характеризуються рівномірними пульсаціями АЧХ, тому їх іноді називають фільтрами з рівномірними пульсаціями (*equiripple filters*). Через рівномірність пульсацій АЧХ імпульсна характеристика синтезованого фільтра може різко обриватися по краях (на відміну від фільтрів, синтезованих з використанням вікон, де кінці імпульсної характеристики завжди спадають плавно). Розглянемо вибрані для синтезу фільтра аргументи:

```
firpm(350, [0, 15e3, 48e3, (2.4e6/2)] / (2.4e6/2), ...
[1 1 0 0], [1 1], 20));
```

Перший параметр – це порядок фільтра n , другим аргументом функції задається вектор частот f , що нормований на частоту Найквіста (половина

частоти квантування), з координатами з інтервалу $[0;1]$, третій аргумент – вектор a , що містить значення бажаної АЧХ для частот з вектора f . Причому, бажана АЧХ на частотах між $f(k)$ і $f(k+1)$ при *непарному* k визначається лінійною функцією, що з'єднує точки $(f(k), a(k))$ і $(f(k+1), a(k+1))$. Бажана АЧХ на частотах між $f(k)$ і $f(k+1)$ при парному k вважається невідзначеною. Ці смуги частот вважаються перехідними або неважливими.

Стосовно до нашого випадку, в діапазоні від нуля до частоти 15 кГц ми хочемо мати максимальний сигнал, без заглушення (звичайна людина здатна чути звукові коливання в діапазоні частот від 16–20 Гц до 15–20 кГц), в діапазоні від 48 кГц до 1,2 МГц ми хочемо повного пригнічення сигналів. В діапазоні від 15 кГц до 48 кГц – як вийде. Далі для відзначаючих діапазонів вказуються вагові коефіцієнти, згідно з якими забезпечується наближення до бажаної АЧХ в кожному діапазоні. Останній аргумент повинен бути цілим числом та визначає ступінь наближення до рівномірної пульсації. Мінімальне значення 16. Функція повертає $n+1$ симетричних комплексних коефіцієнтів фільтра, модуль яких графічно зображено на рис. 1.4:

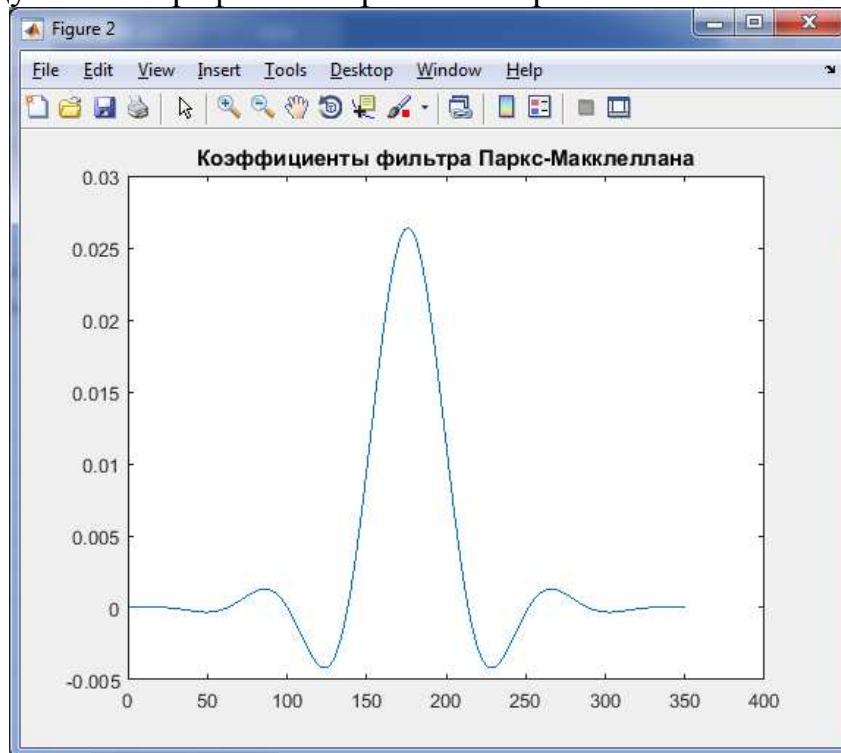


Рис. 1.4. Коефіцієнти фільтра Паркса-Макклеллана

Побудуємо також амплітудно-частотну характеристику (АЧХ) застосовуваного фільтра. Роботу нам полегшить функція `freqz()`, яка за заданими коефіцієнтами фільтра обчислює його АЧХ в 512 значеннях частот рівномірно розташованих у заданому діапазоні:

```
[h1,f]=freqz(numIn,1,[],2);
figure
plot(f,20*log10(abs(h1)));
```

Тут **numIn** – коефіцієнти фільтра, другий аргумент задає максимальне значення АЧХ (1 означає, що АЧХ нормовано на свій максимум), четвертий аргумент є коефіцієнтом розтягування частотній осі (значення 2 відповідає нормалізованій частоті).

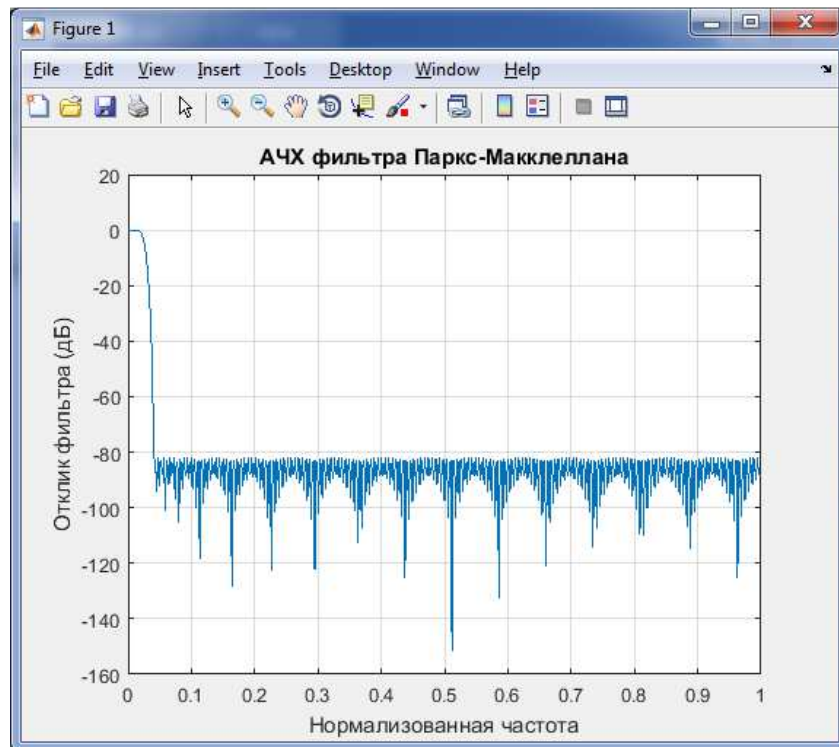


Рис. 1.5. АЧХ фільтра Паркса-Макклеллана

Як бачимо з графіка АЧХ, фільтр пригнічує сигнали з частотою вище 48 кГц більш ніж на 80 дБ. Це дуже гарне приглушення. Розглянемо більш докладно вершину АЧХ. Для цього побудуємо графік АЧХ в діапазоні 0–48 кГц в реальних частотах (рис.1.6):

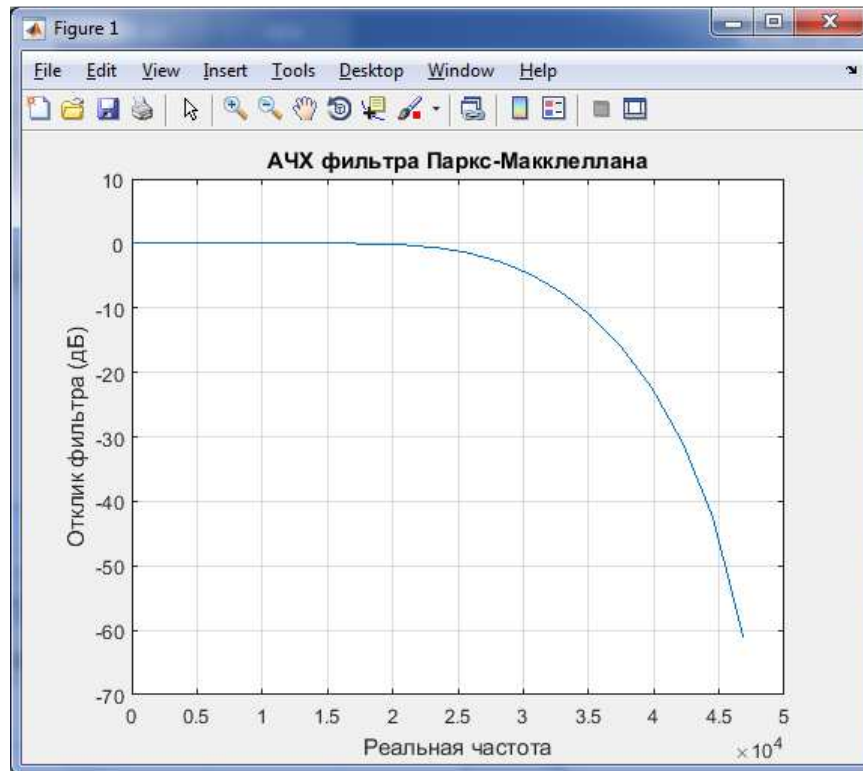


Рис. 1.6. Вершина АЧХ фильтра Паркса-Макклеллана

Як бачимо, хоча було замовлено відсутність пригнічення тільки в діапазоні 0–15 кГц, реально до частоти 21 кГц (там, де ми ще щось можемо чути), пригнічення не перевищує 1 дБ. Нагадаємо ще раз, що зниження частоти і розглянутий СІХ-фільтр застосовуються до фази прийнятого сигналу після частотного дискримінатора.

Отриманий сигнал з частотою квантування 48 кГц, що найчастіше використовується для оцифрування звукових сигналів, проходить ще одну фільтрацію, але тепер вже застосовується НІХ-фільтр Баттерворта низьких частот (Low-pass Butterworth IIR filter). Характеристики фільтра розрізняються частотою зрізу для FM-радіостанцій Європи та США:

```
if deemph_region == 'eu'      % find de-emphasis filter coeff
    [num,den] = butter(1,3183.1/(audio_fs/2));
elseif deemph_region == 'us'
    [num,den] = butter(1,2122.1/(audio_fs/2));
```

Побудуємо АЧХ запропонованого фільтра для Європейського стандарту (рис 1.7):

```
[h1,f]=freqz(num,den);
nf=length(f);
f1=f*audio_fs/2/pi;
q1=20*log10(abs(h1));
figure
plot(f1,q1);
title('АЧХ НІХ-фільтра Баттерворда: ф-я freqz()');
xlabel('Реальна частота');
```

```
ylabel('Відгук фільтра (дБ)');  
grid on
```

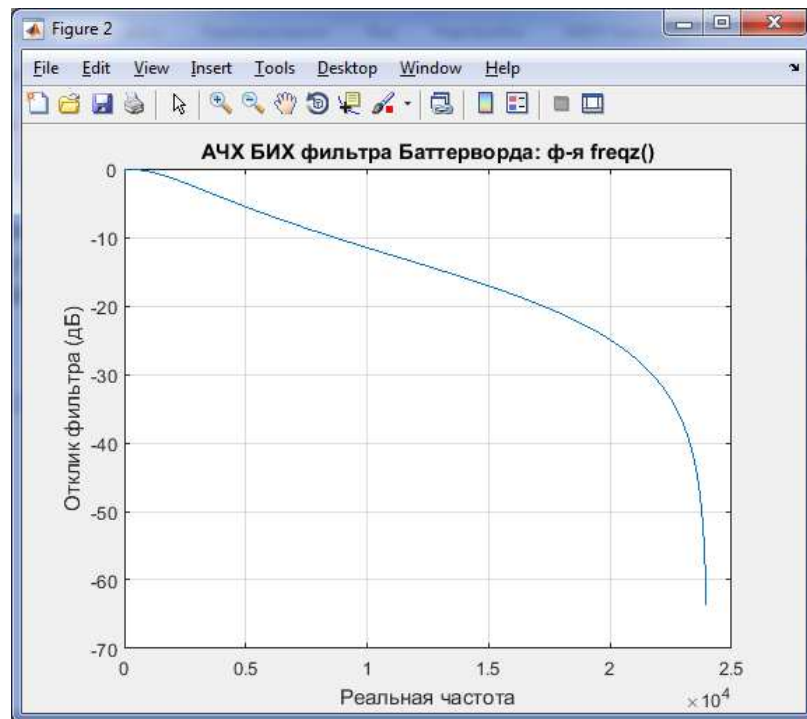


Рис. 1.7. АЧХ НЧ-фільтра Баттерворта 1-го порядку

Звернемо увагу, що використовується найпростіший фільтр 1-го порядку (кількість обох типів коефіцієнтів фільтра при цьому дорівнює 2 у відповідності з формулою 1.4), тому на частоті зрізу ми вже маємо пригнічення близько 3 дБ. З тією ж частотою зрізу АЧХ фільтр 6-го порядку цього недоліку не має (рис.1.8):

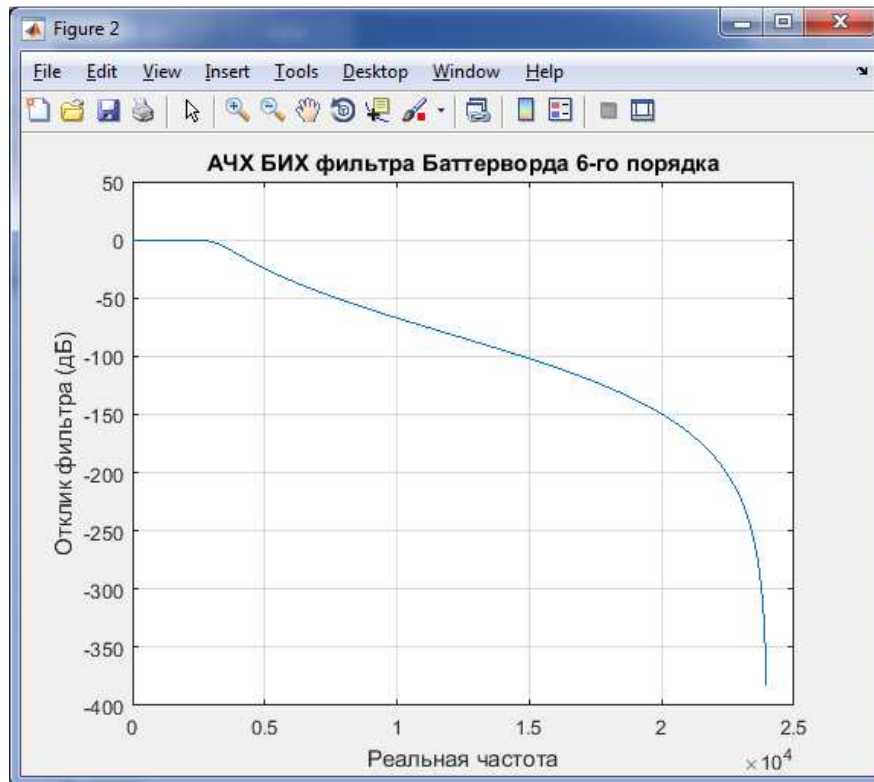


Рис. 1.8. АЧХ НЧ-фильтра Баттерворда 6-го порядка

Для перевірки та для більш детального вивчення фільтра Баттерворта побудуємо його АЧХ самостійно. Для цього будемо подавати на вхід фільтра гармонійний сигнал на різних частотах і дивитися спектр сигналу на виході фільтра:

```
% Власна побудова АЧХ фільтру:
Fs=audio_fs;
dt=1/Fs;
T=dt*1023; % час генерації у с
t=0:dt:T;

L=length(t);
NFFT = 2^nextpow2(L); % кількість фільтрів ШПФ
dFFT=Fs/NFFT; % Розміщення фільтрів ШПФ (Гц):
f0=0:dFFT:Fs-dFFT; % розмітка частотної вісі
for i=1:nf
    Fc=f1(i);
    OMc=2*pi*Fc;
    dataIn = exp(1i*OMc*t);
    dataOut = filter(num,den,dataIn);

    % Амплітудний спектр вхідного сигналу:
    FIn=fft(dataIn,NFFT);
    FmodIn=20*log10(abs(FIn)/max(abs(FIn)));
    % Амплітудний спектр вихідного сигналу:
    FOut=fft(dataOut,NFFT);
```

```

FmodOut=20*log10(abs(FOut)/max(abs(FIn)));
figure
plot(f0,FmodIn,f0,FmodOut);
grid on

[AmaxIn,fmax]=max(abs(FIn));
AmaxOut=abs(FOut(fmax));
q(i)=20*log10(AmaxOut/AmaxIn);
end
figure
plot(f1,q1,f1,q);
title('АЧХ НЧ-фільтра Баттерворда');
xlabel('Реальна частота');
ylabel('Відгук фільтра (дБ)');
legend('Ф-я freqz() Matlab','Спектр-аналізатор');

```

Наприклад, при частоті 3 кГц спектр гармонічного сигналу на вході і виході нашого фільтра Баттерворта виглядає таким чином (рис.1.9):

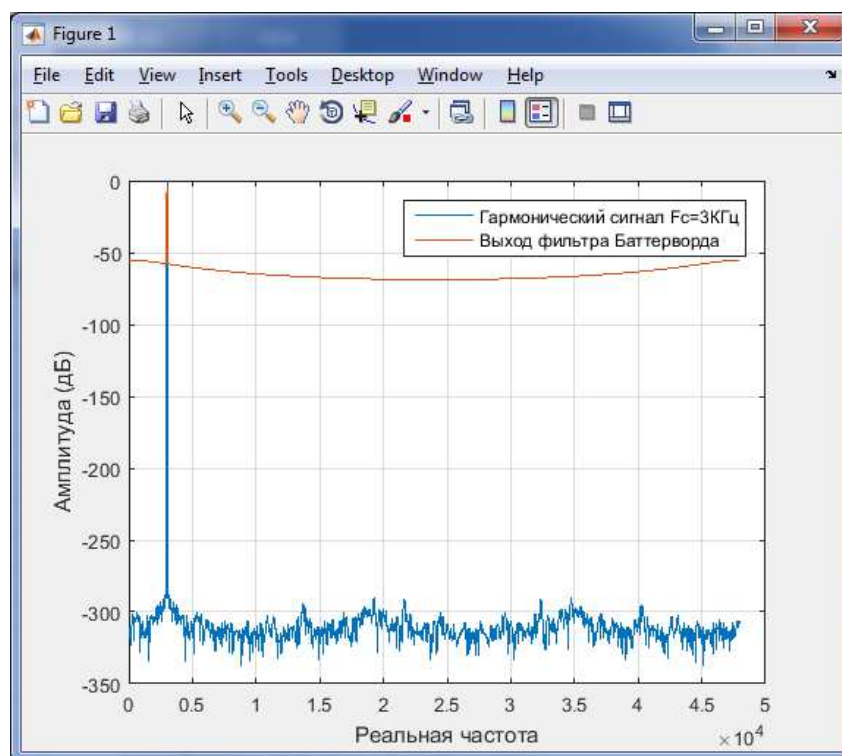


Рис. 1.9. Проходження гармонічного сигналу через фільтр Баттерворта

Рівень бічних пелюсток значно підвищується. Оскільки ми генеруємо ідеальний сигнал, різниця в рівні бічних пелюсток становить близько 250 дБ (перевищення амплітуди одного сигналу над іншим за амплітудою в $2 \cdot 10^{13}$ разів!). Зрозуміло, що для реальних сигналів такого бути не може. Частота 3000 Гц точно відповідає налаштуванню 65 фільтра ШПФ. Зсув частоти сиг-

налу від центру фільтра всього на 0.1 Гц вже дає зовсім іншу спектральну картину (рис. 1.10):

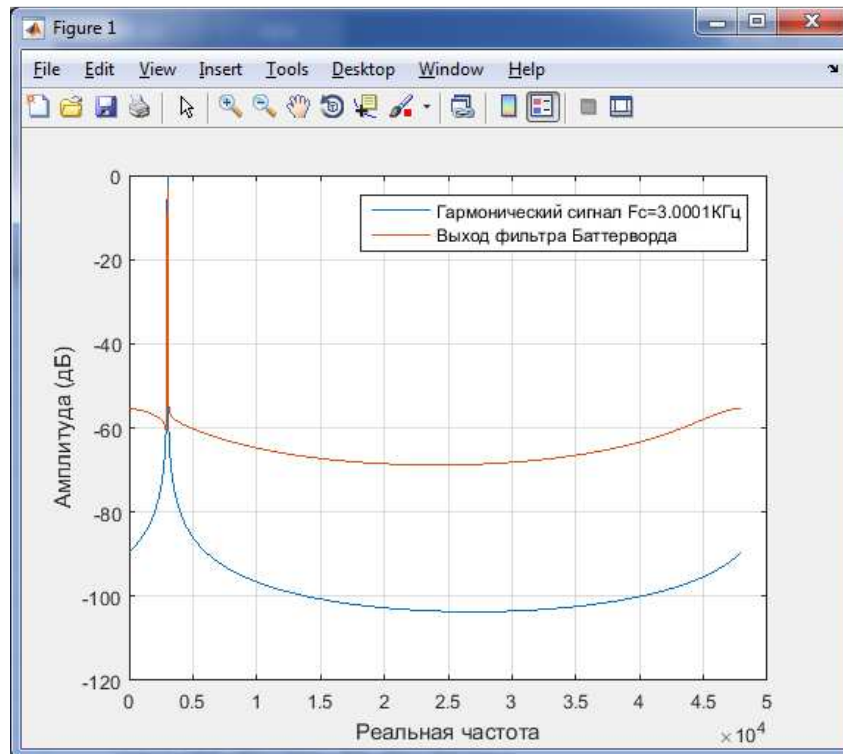


Рис. 1.10. Проходження зміщеного гармонічного сигналу через фільтр Баттерворта

Тепер ми бачимо різницю приблизно в 40 дБ, оскільки на спектр, який спостерігається, впливають бічні пелюстки фільтрів ШПФ. Більш реальну картину ми побачимо, якщо накладемо на гармонійний сигнал білий шум з відношенням сигнал/перешкода 12 дБ (рис. 1.11):

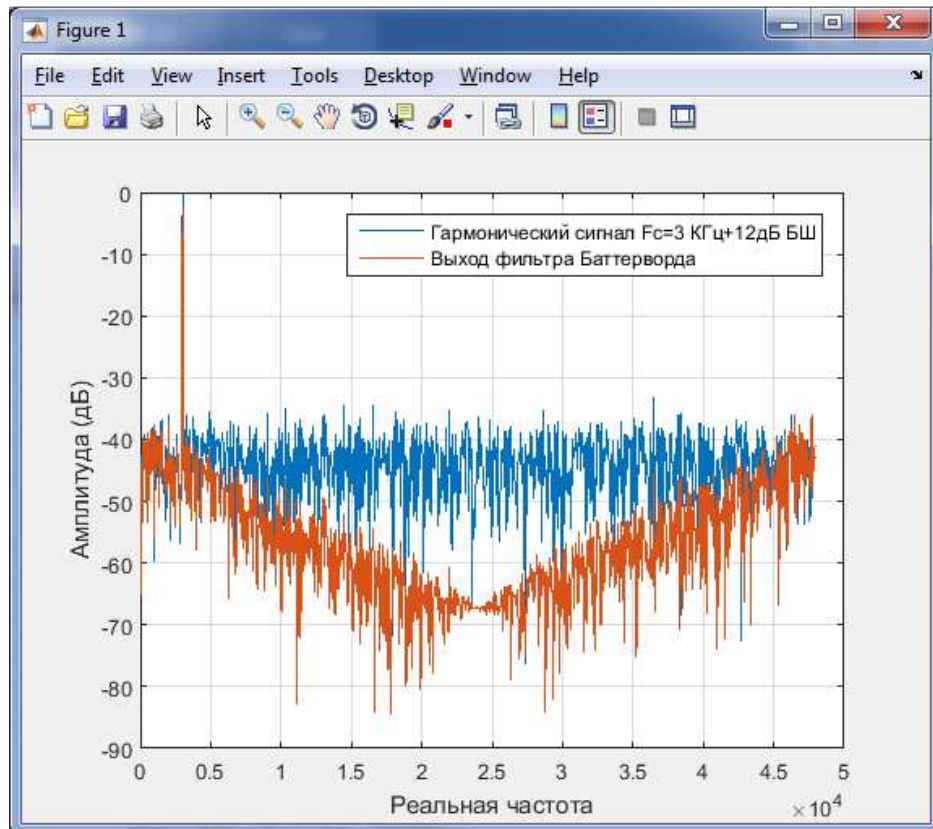


Рис. 1.11. Проходження зашумленого гармонійного сигналу через фільтр Баттерворта

Як бачимо, наш фільтр Баттерворта реально пригнічує шуми після частоти зрізу. Природно, максимальне приглушення коло частоти Найквіста (24 кГц) і становить близько 25 дБ, хоча на АЧХ (рис. 1.7) пригнічення на цій частоті понад 60 дБ. Пояснюється різниця різним проходженням через фільтр Баттерворта гармонійного сигналу і нормального білого шуму.

У висновку відзначимо, що АЧХ фільтра, побудована порівнянням спектрів гармонійного сигналу до і після проходження фільтра Баттерворта, повністю збігається з АЧХ, побудованою функцією `freqz()` (використовуються результати теоретичних розрахунків), причому АЧХ не залежить від накладення розумного рівня завад (рис. 1.12):

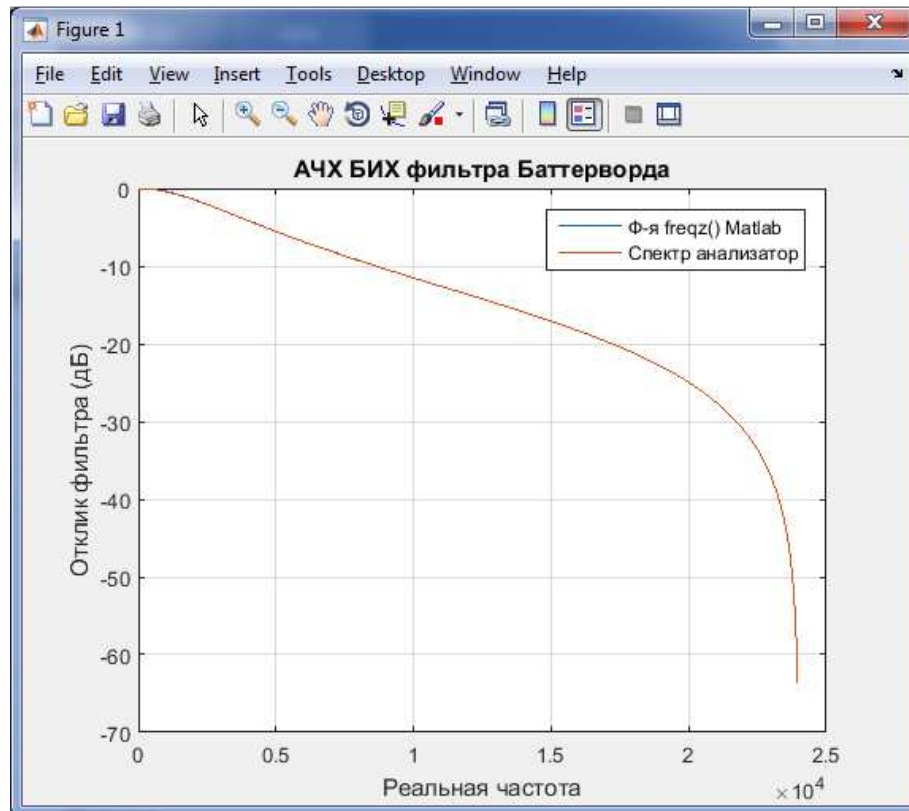


Рис. 1.12. Порівняння теоретичної і експериментальної АЧХ фільтра Баттерворта

Що стосується оптимальності застосовуваного фільтра Баттерворта, ми постараємося відповісти на це питання після реалізації цифрової моделі FM-сигналу і при аналізі сигналів реальних FM-радіостанцій.

1.4. Цифрова модель сигналів FM-радіостанцій

Моделювати ми будемо вихідний сигнал RTL-SDR на частоті 2,4 МГц. Звуковий сигнал будемо брати з будь-якого файлу, в якому звук записаний зі стандартною частотою квантування 48 кГц, тому перш за все доведеться інтерполювати його на частоту приймача. Оскільки сусідні відліки звуку при частоті дискретизації 48 кГц змінюються незначно, досить використовувати лінійну апроксимацію. В основний цикл прийому інформації з RTL-SDR вставимо наступний текст:

```
model=1;
noise=1;
snr=-4; % відношення сигнал/перешкода (signal/noise ratio) в дБ
ks=0;
Fc = 0e3; % Відхилення несучої частоти приймача
        % та передавача (Гц)
dev = 250; % Девіація частоти при частотній модуляції (Гц)
t_rtl = [0:rtlsdr_frmlen-1]'/rtlsdr_fs; % Вектор моментів
        % часу для частоти квантування приймача
```

```

Krtl_s = rtl_sdr_fs/audio_fs;
n_s=rtl_sdr_frmlen/Krtl_s+1; % кількість відліків звуку на інтер-
вально rtl_sdr_frmlen
dt_rtl=1/rtl_sdr_fs; % дискрет квантування приймача
dt_s=1/audio_fs; % дискрет квантування звуку
t=0:dt_rtl:dt_s;
OMc=2*pi*Fc;
FIs=2*pi*dev;
Ki= 1525;
Nl=rtl_sdr_frmlen;
sigma=10^(-snr/20)/sqrt(2);

if model == 0
    % Отримати реальні дані з RTL-SDR:
    rtl_sdr_data = step(obj_rtl_sdr);
else
    % Прочитати дані з аудіо-файлу:
    samples = [ks*n_s+1, (ks+1)*n_s];
    x = audioread('music3_mono48kHz.wav', samples);
    ks=ks+1;

    % Переклад звуку на частоту квантування rtl_sdr_fs лінійною
    % апроксимацією амплітуд:
    for i=1:n_s-1
        dx=(x(i+1)-x(i))/dt_s;
        for j=1:Krtl_s
            x_rtl((i-1)*Krtl_s+j)=x(i)+dx*t(j);
        end
    end
end
end
end

```

Як вже зазначалося (формула 1.1), для модуляції частоти в FM-сигналах використовується не сам звуковий сигнал, а його інтеграл. При цифровому моделюванні інтеграл замінюють кумулятивною сумою:

```

int_x = cumsum(x_rtl)/Ki;
rtl_sdr_data = exp(1i*(OMc*t_rtl - FIs*int_x'));
% Наложение помехи:
if noise
    gaussnoise=(randn(Nl,1)+1i*randn(Nl,1))*sigma;
    rtl_sdr_data = rtl_sdr_data+gaussnoise;
end

```

Кумулятивну суму обчислює функція **cumsum()**, яка повертає вектор тієї ж розмірності, що і вхідний, у якому перший елемент збігається з першим елементом вихідного вектору, другий дорівнює сумі першого і другого і т. д. Коефіцієнт **к_i** будемо підбирати виходячи з близькості декодованого і вихідного звуку.

Прослуховування декодованого звуку показує, що він добре відповідає вихідному звуку. Також можна нанести їх на один графік (рис. 1.13):

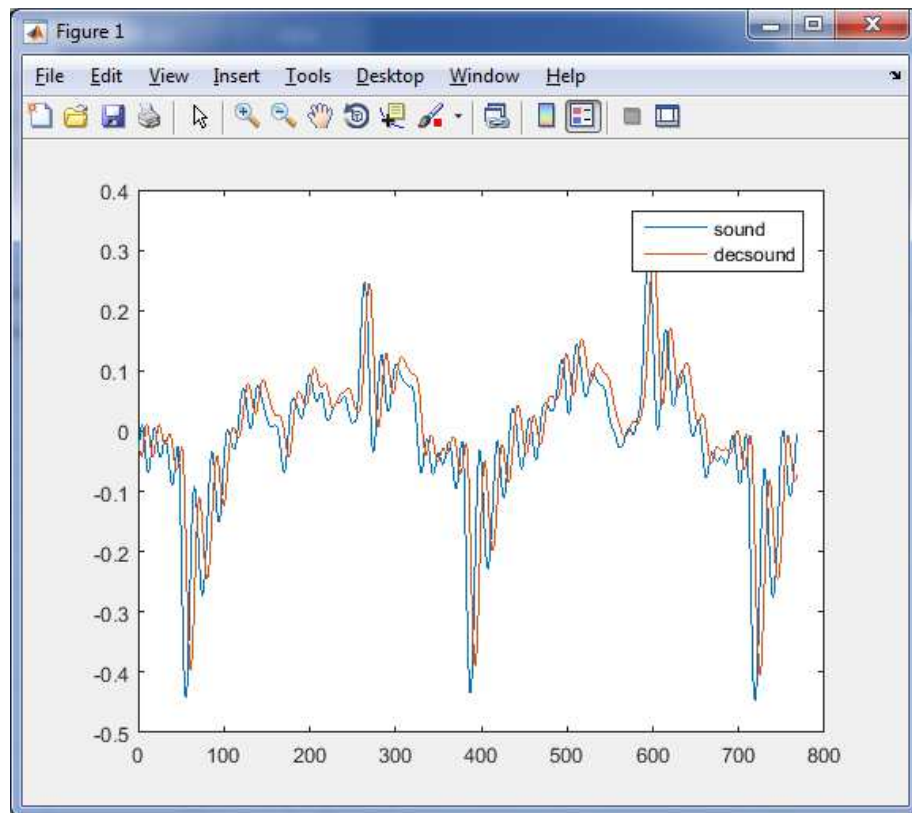


Рис. 1.13. Порівняння вихідного звуку зі звуком, отриманим з моделі FM-сигналу

Ми бачимо, що по амплітуді коливань вихідний і декодований звуки практично збігаються, що свідчить про правильний підбір коефіцієнта підсилення k_i . Цікаво, що зміна знаку перед кумулятивною сумою призводить до зміни знаку коливань звуку (на сприйняття звуку це не впливає) (рис. 1.14):

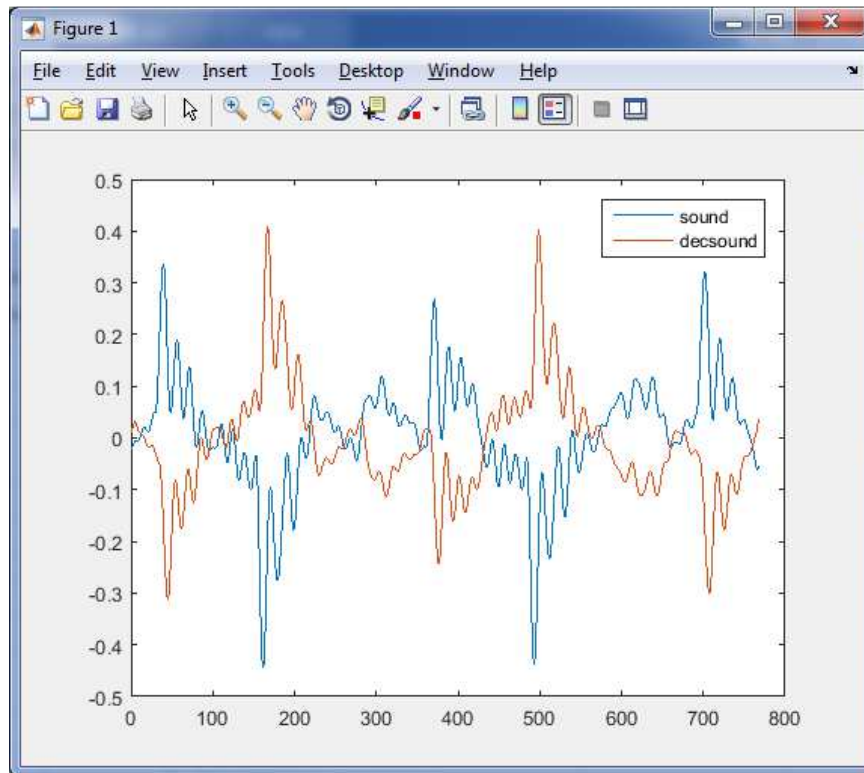


Рис. 1.14. Порівняння вихідного звуку зі звуком, декодованим з моделі FM-сигналу зі зміненим знаком інтегралу

Видно, що декодований звук трохи відстає в часі. Для подальших цілей при відображенні необхідно провести відповідну синхронізацію:

```
Nsd=5; % Зсув демодульованого звуку в моделі
      % відносно вихідного звуку
if Fplot
    Nn=1; % Начальна точка на відображення
    dN=n_s-1-Nn; % кількість точок на відображення
    Nk=Nn+dN-Nsd;
    tgr=Nn:Nk;
    if model
        plot(tgr,x(Nn:Nk),tgr,data_deemph(Nn+Nsd:Nk+Nsd));
        legend('sound','decsound');
    end
    pause(1);
end
```

Паузу в 1 секунду зроблено для можливості перегляду графіка.

Для ідеального підбору зсуву будемо обчислювати СКВ (середнє квадратичне відхилення) різниці демодульованого звуку від вихідного:

```
if model
    n1=length(data_deemph);
    D(m)=var(data_deemph(1+Nsd:n1)-x(1:n1-Nsd));
```

```

m=m+1;
if m == 500
    SKO=mean(D)^0.5;
    fprintf(Nsd=%d, Ki=%d, md=%.3f\n', Nsd, Ki, md*1e3);
    break;
end
end
end

```

В кожному такті отримання порції інформації з SDR-RTL обчислюється дисперсія відхилення і усередненням дисперсії за заданою кількістю порцій обчислюється СКВ.

Обчислення СКВ при декількох зсувах дає наступний результат:

```

Nsd=4, Ki=305, SKO=0.041
Nsd=5, Ki=305, SKO=0.026
Nsd=6, Ki=305, SKO=0.030

```

Звідки випливає, що декодований звук затриманий на п'ять 48-кілогерцових відліків. При компенсації зсуву декодований і вихідний звуки практично збігаються (рис. 1.15):

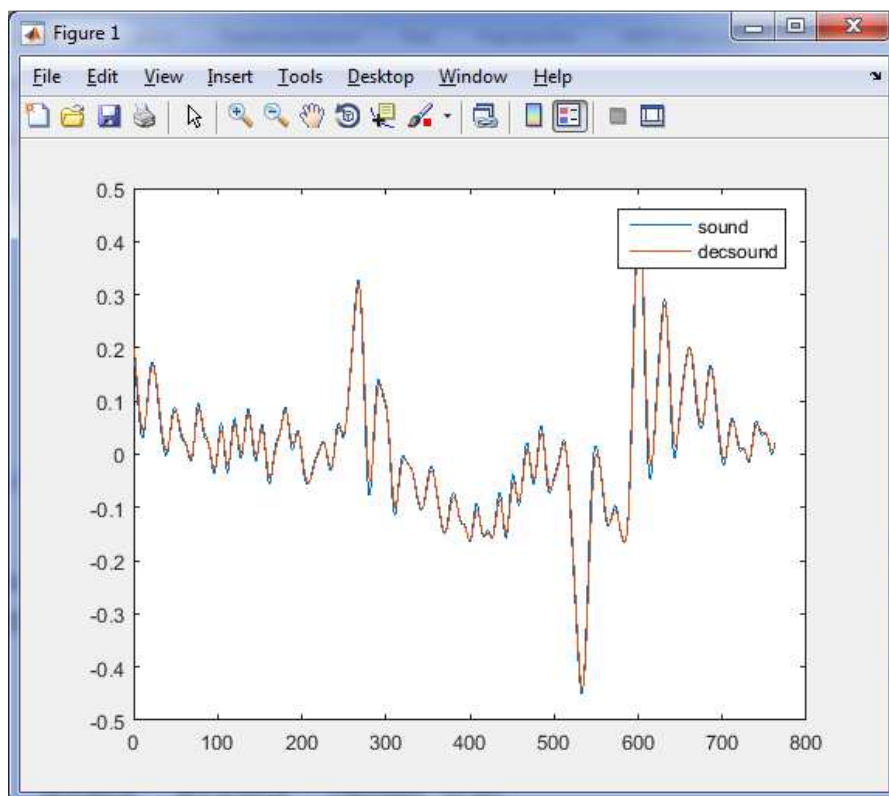


Рис. 1.15. Початковий звук і звук, отриманий з моделі ФМ-сигналу після часової синхронізації

Відзначимо, що в реальності спочатку було підібрано зміщення і потім, знову ж таки за СКВ, підібраний коефіцієнт підсилення k_i .

Коротко розглянемо вплив на демодуляцію звуку нормального білого шуму. Залежність СКВ відхилення від відношення сигнал/перешкода наступна:

snr=6.0, cko=0.026

snr=4.0, cko=0.030

snr=0.0, cko=0.066

Вже при 6 дБ відношенні сигнал/перешкода якість декодованого звуку відповідає вихідному звуку. При 4 дБ чути перешкоди, які на амплітудному графіку виглядають так (рис. 1.16):

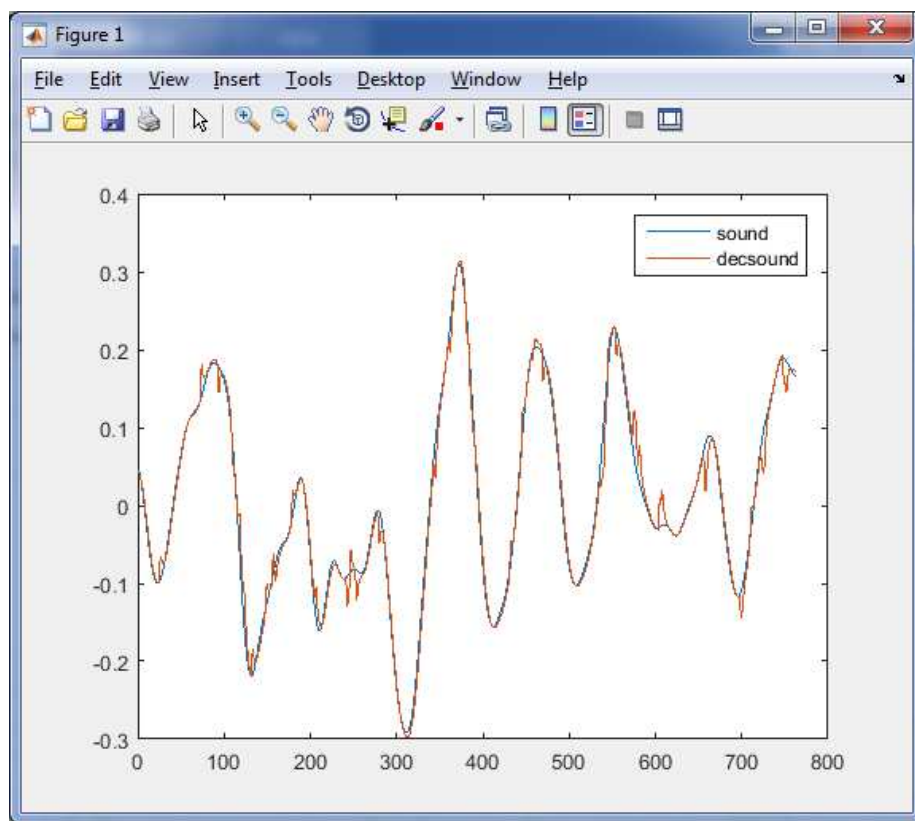


Рис. 1.16. Вихідний і демодульований звук при впливі білого шуму з відношенням с/п 4 дБ

При відношенні сигнал/перешкода 0 дБ записану музику ще можна слухати, але з серйозними шумами (рис. 1.17):

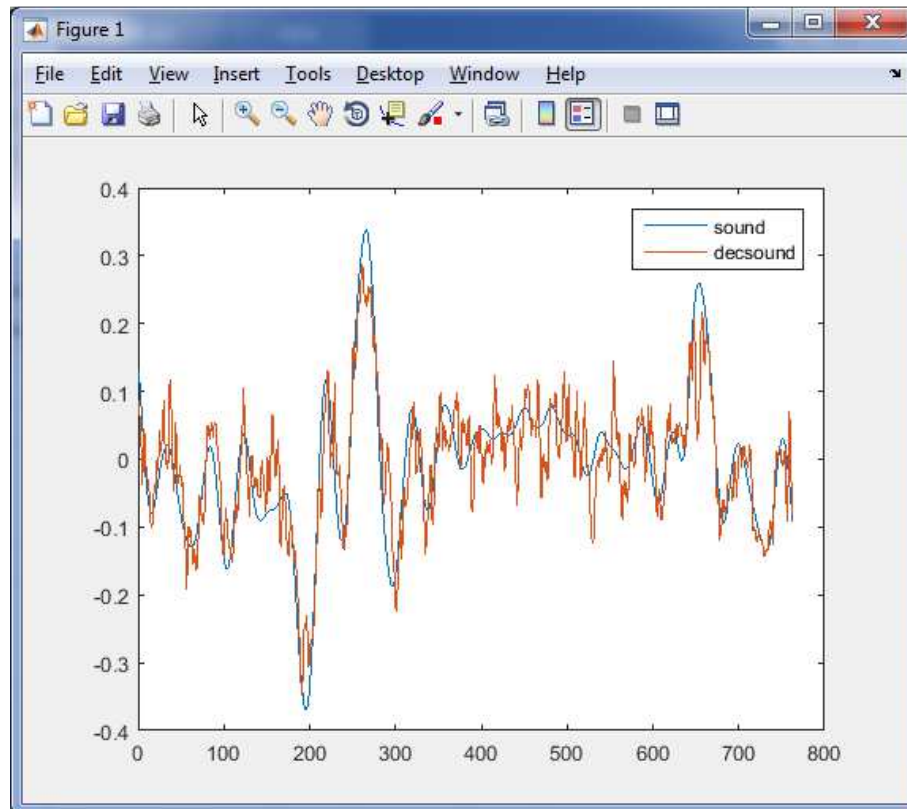


Рис. 1.17. Вихідний і демодульований звук при впливі білого шуму з відношенням с/п 0 дБ

Зазначимо, що ми вимірюємо відношення сигнал/перешкода на виході приймача у всій смузі пропускання 2,4 МГц, тому і отримали невеликі значення. Після обробки сигналу, відношення сигнал/перешкода за піком сигналу у власній смузі 250 кГц для реальних FM-радіостанцій зазвичай варіюється в діапазоні 35–45 дБ, як це видно на рисунку 1.2.

Розглянемо тепер вплив на якість звуку наявності різниці в частотах FM-станції і приймача. Установка значного зміщення в 700 кГц (з таким розподілом вже можуть працювати сусідні радіостанції, наприклад, Kiss FM та Наше радіо) абсолютно не змінює вимірюване нами СКВ. Правда, декодований звук при цьому зміщується на помітну константу вниз (рис. 1.18):

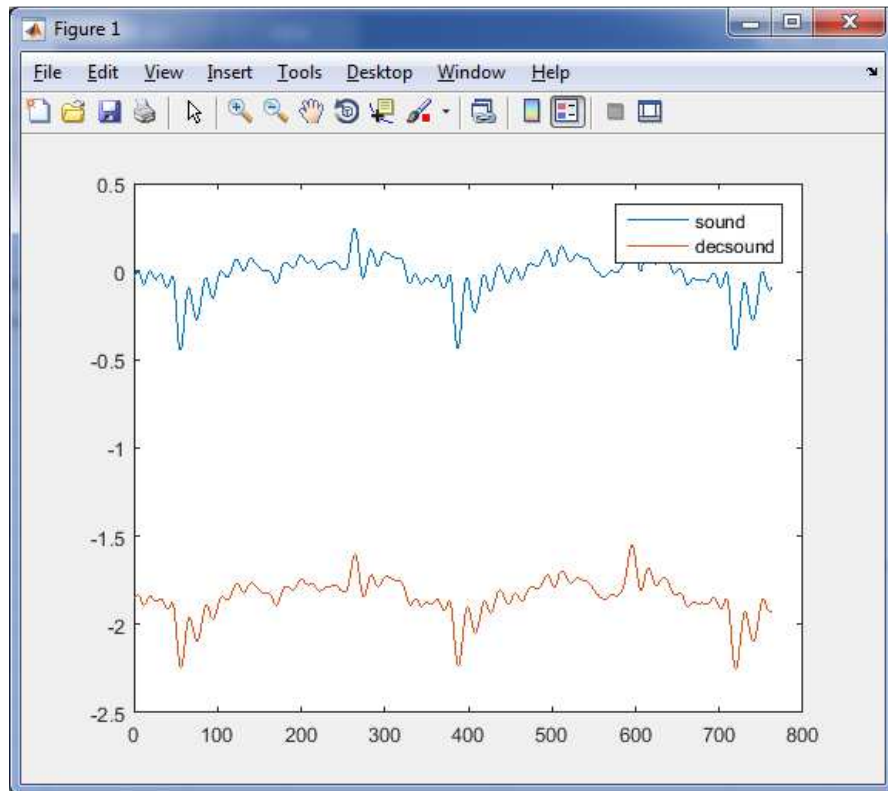


Рис. 1.18. Зсув амплітуди звуку, модельованого при зсуві по частоті

Це зміщення не відображається на сприйнятті звуку і може бути легко усунено наступною процедурою:

```
data_deemph=data_deemph-mean(data_deemph) ;
```

Виявлена властивість нашого простого алгоритму свідчить про його дуже великий недолік, а саме майже повну відсутність вибіркової. У цьому легко переконатися і за сигналами реальних FM-радіостанцій. На частоті Нашого радіо 102,8 МГц ми продовжуємо чути більш потужну станцію Kiss FM, що працює на частоті 102,1 МГц! Причому декодований звук теж зміщений відносно нуля (правда, в позитивну сторону) (рис. 1.19):

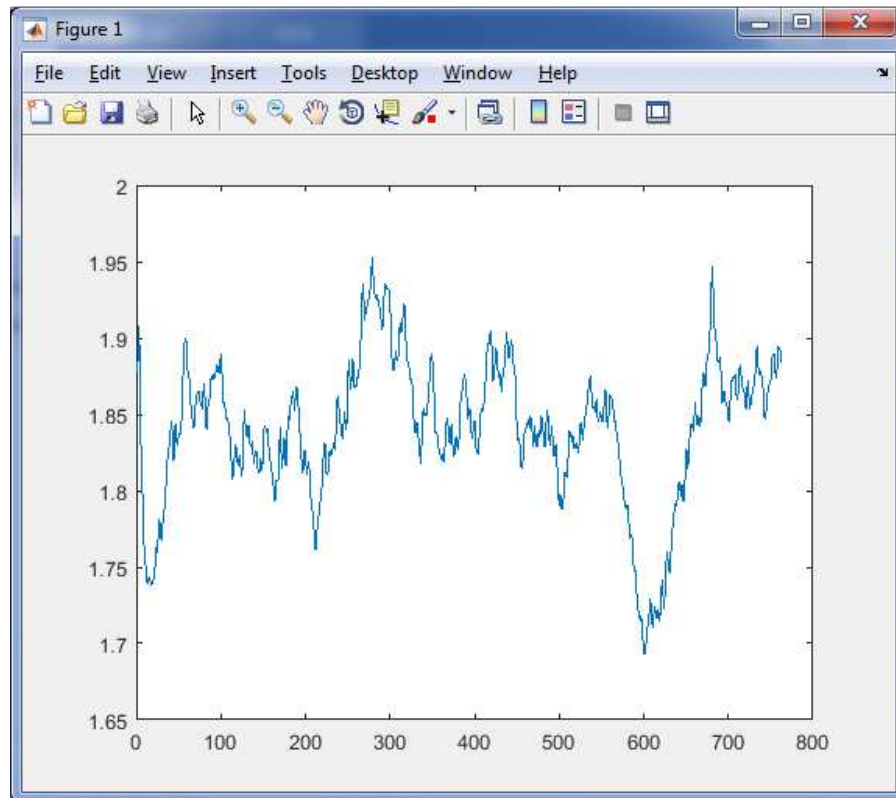


Рис. 1.19. Зсув амплітуди реальної FM-станції при прийомі на зміщеній частоті

Зауважте, що величина зміщення по моделі і реальній FM-радіостанції приблизно однакова ($\sim 1,85$) і відрізняється тільки знаком.

1.5. Вдосконалення некогерентного дискримінантного алгоритму обробки FM-сигналів

Тепер, коли у нас є інструмент для вивчення впливу різних методів обробки FM-сигналів на якість звуку, що декодується, ми можемо приступити до вдосконалення запропонованого нам алгоритму.

Перш за все, сигнал з виходу приймача повинен пройти фільтрацію в смузі, прийнятій для FM-сигналів (250 кГц) і, відповідно, при цьому частоту квантування знижують до 240 кГц. Наприклад, така обробка пропонується при стереоприйомі FM-сигналів у вже знайомій нам книзі [2] (рис. 1.20):

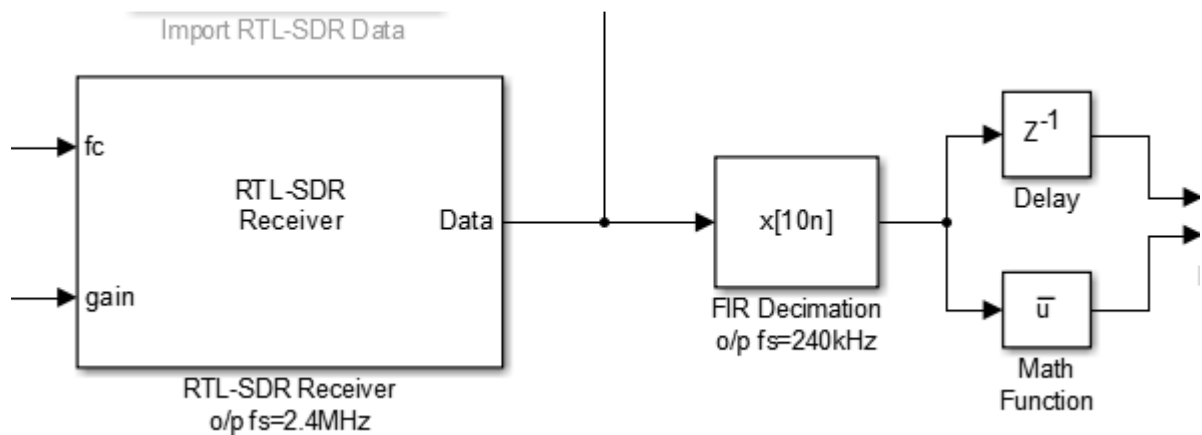


Рис. 1.20. Схема стереоприйому FM-радіостанцій

Зверніть увагу, що в розглянутій нами програмі передбачений варіант читання значень амплітуд в квадратурних каналах з файлу, в який запис проведено з частотою дискретизації 240 кГц.

Розглянемо три варіанти СІХ-фільтрів – з вікном Хеммінга, Кайзера, і вже знайомий нам фільтр, що синтезується алгоритмом Паркса-Макклеллана.

Для створення фільтра з вікном Хеммінга можна використовувати функцію `fir1(n,W)`, в якій це вікно використовується за замовчуванням. Функція призначена для конструювання фільтрів заданого порядку низьких і високих частот, смугових і багатосмугових фільтрів.

В якості початкового наближення розглянемо фільтр Хеммінга відносно невисокого порядку з наступними параметрами:

```
n = 45;           % порядок фільтру Хемминга
Df=125e3;        % половина смуги FM-сигналів
W=Df/(rtlsdr_fs/2);
numIn=fir1(n,W); % Вікно Хемминга
```

Фільтр з вікном Кайзера більш універсальний і задається наступним чином:

```
Df=125e3;        % половина смуги FM-сигналів
fcuts = [Df Df+Df/5]/(rtlsdr_fs/2);
mags = [1 0];
devs = [0.05 0.03];
[n,Wn,beta,ftype] = kaiserord(fcuts,mags,devs);
numIn = fir1(n,Wn,ftype,kaiser(n+1,beta),'noscale');
```

Порядок фільтра не вказується, а обчислюється, виходячи із заданих вимог до характеристик фільтра. Зріз визначається частотою, після якої необхідно почати пригнічення, і частотою, після якої необхідно забезпечити максимальне пригнічення. Як завжди, частоти повинні бути нормовані на частоту Найквіста. У векторі **mags** вказується рівень амплітуд на інтервалі пропус-

ку та пригнічення сигналу. У нашому випадку ми даємо фільтру 25 кГц для виходу на максимальне пригнічення. У векторі **devs** вказується в частках ступінь наближення до заданих амплітуд. Для заданих параметрів порядок фільтра дорівнює **215**.

С фільтром Паркса-Макклеллана ми вже знайомі, використаємо його з наступними параметрами:

```
numIn = firpm(300, [0, Df, Df+Df/5, rtl_sdr_fs/2]/(rtl_sdr_fs/2), ...  
              [1 1 0 0], [1 1], 20);
```

Порядок фільтра дорівнює 300, і вимагаємо виходу на максимальне пригнічення за ті ж 25 кГц, що і в фільтрі Кайзера.

АЧХ запропонованих фільтрів, що нанесені на один графік, виглядають наступним чином (рис. 1.21, 1.22):

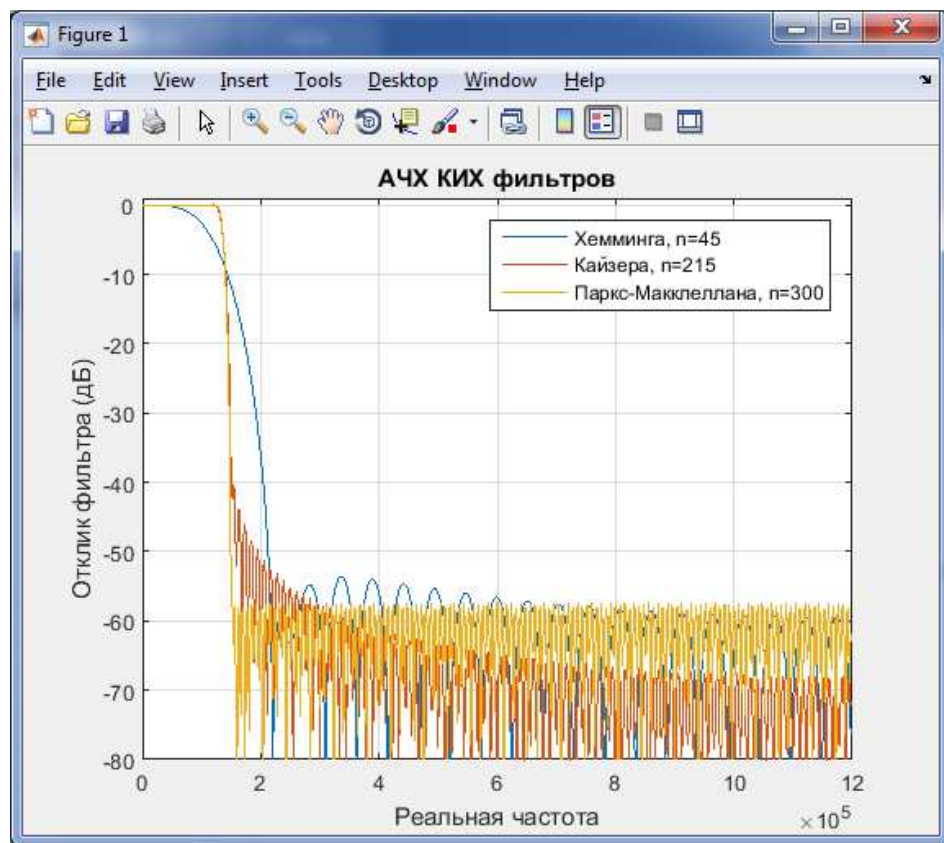


Рис. 1.21. АЧХ фільтрів Хеммінга, Кайзера та Паркса-Макклеллана в смузі 1,2 МГц

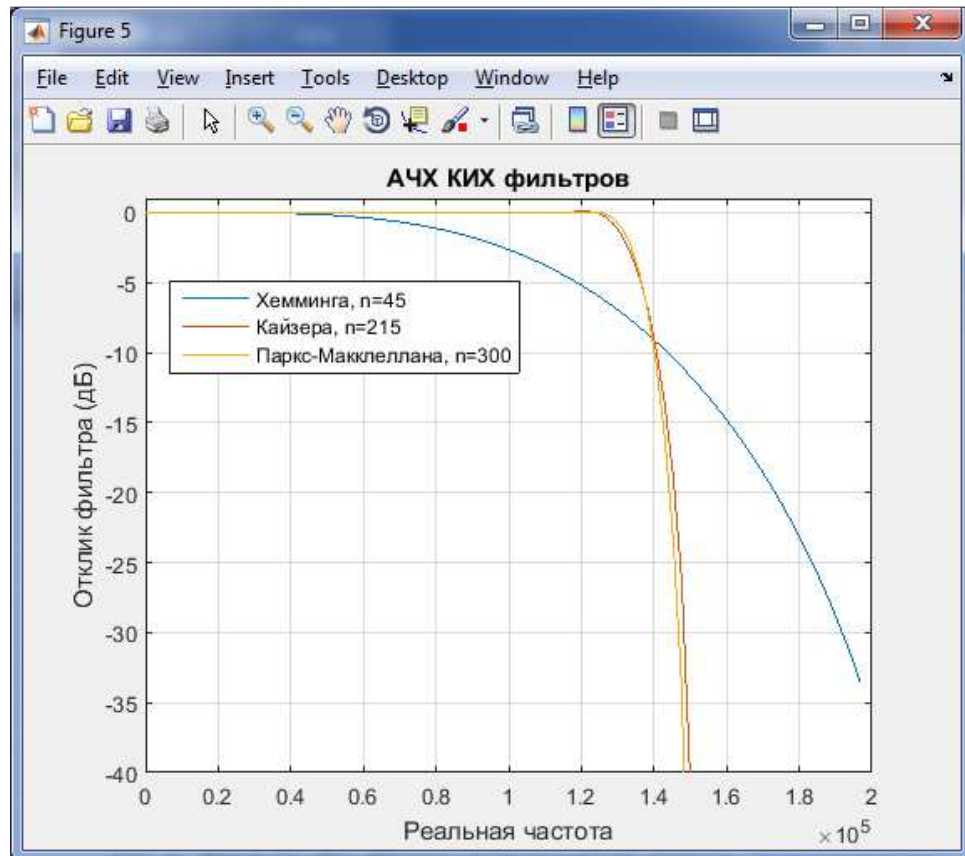


Рис. 1.22. АЧХ фільтрів Хеммінга, Кайзера та Паркса-Макклеллана в смузі 200 кГц

Фільтри Кайзера, Паркса-Макклеллана з великим порядком забезпечують практично вертикальний зріз і трохи відрізняються рівнем бічних пелюсток. У фільтра Кайзера вони починаються на рівні -40 дБ і поступово опускаються до -70 дБ, в той час як у фільтра Паркса-Макклеллана вони одразу виходять на постійний рівень -58 дБ. У фільтра Хеммінга з істотно меншим порядком рівень бічних пелюсток практично такий же, але значно більш полого основна пелюстка (втрати на частоті зрізу досягають 6 дБ, а бічні пелюстки починаються з частоти ~ 220 кГц).

Подальшу обробку зробимо такою ж, як обробку сигналу, записаного у файлі з частотою квантування 240 кГц. СІХ-дециматор, застосовуваний до фази сигналу приймемо з параметрами:

```
obj_decmt = dsp.FIRDecimator(...
    'DecimationFactor', 5, 'Numerator', ...
    firpm(100, [0, 15e3, 20e3, (240e3/2)] / (240e3/2), ...
    [1 1 0 0], [1 1], 20));
```

Змінивши порядок фільтра Паркса-Макклеллана з 350 до 100, повне пригнічення високих частот починається з 20 кГц замість 48 кГц і, природно, фактор децимації дорівнює $240/48=5$. АЧХ такого фільтра виглядає так (рис. 1.23):

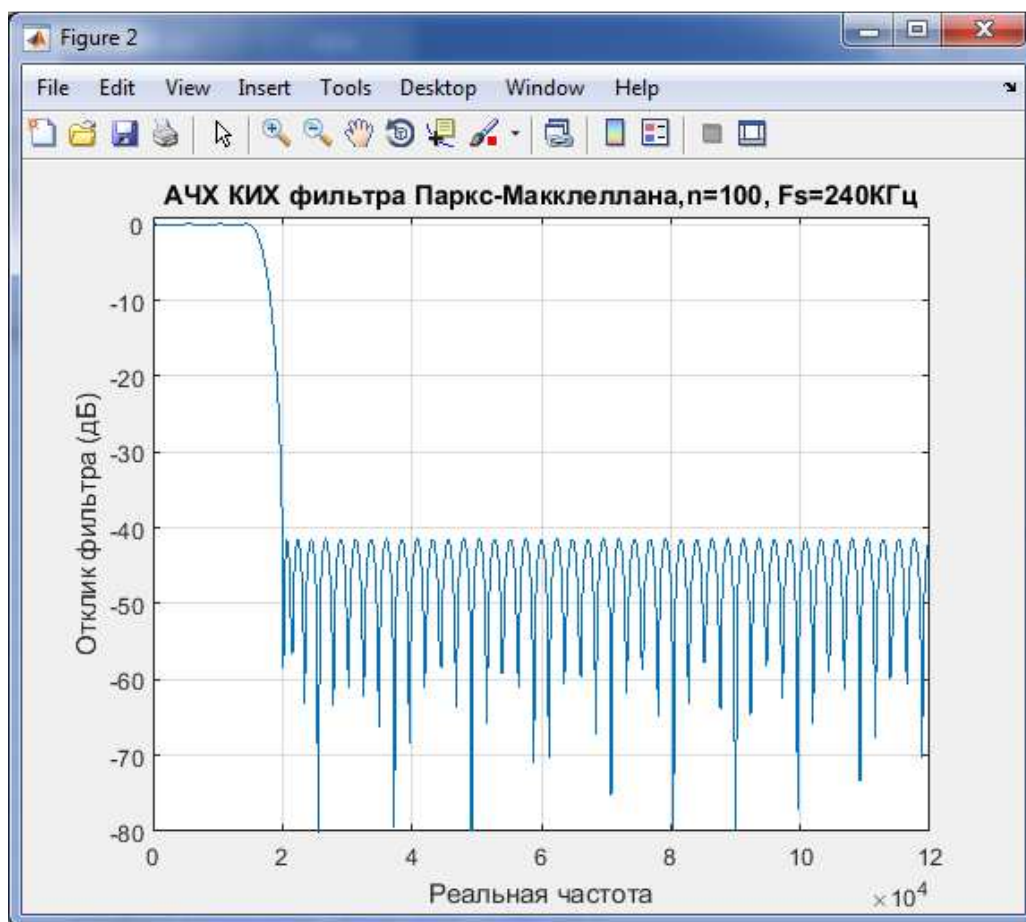


Рис. 1.23. АЧХ фільтра Паркса-Макклеллана 100-го порядку у смузі 120 кГц

Як бачимо, починаючи з 20 кГц, забезпечується пригнічення близько 40 дБ.

Тепер оцінимо, як відобразилися наші доробки на якості прийому сигналів реальних FM-станцій. Результати поміщені в таблицю 1.2. За 10 балів прийнято якість звуку, що забезпечується програмою SDRSharp. Три стовпці в графі з СІХ-децимацією відповідають розглянутим нами варіантам фільтрації (таблиця 1.2):

Таблиця 1.2

№ з/п	Частота FM-станції (МГц)	Обробка на $F_s=2,4$ МГц	СІХ-децимація до $F_s=240$ кГц		
			1	2	3
1	99,2	3	10	10	10
2	102,1	9	10	10	10
3	102,8	0	7	7	7
4	103,3	0	1	1	1
5	104,1	0	10	10	10
6	105,1	0	9	9	9
7	106,4	0	10	10	10
8	107,8	5	10	10	10

Доопрацювання істотно поліпшили можливості прослуховування FM-радіостанцій. Якщо у вихідному варіанті програми комфортно прослуховувалася одна радіостанція і ще дві від сили, то після попередньої фільтрації і зниження частоти квантування з 2,4 МГц до 240 кГц хороша чутність забезпечується на 6 радіостанціях, ще одна прослуховується задовільно і тільки одна, як і раніше, практично не прослуховується. Зрозуміло, що на слух вловити різницю між варіантами СІХ-фільтрів не вдається.

Для більш тонкого вивчення впливу різних елементів обробки введемо в розглянуту модель FM-сигналів два сигнали з різницею в частотах 700 кГц, імітуючи ситуацію з Kiss FM і Нашим Радіо, коли прийом сигналу потрібно забезпечити з однієї радіостанції на тлі іншої потужної радіостанції, розташованої поруч по частоті. Обидва сигнали будемо адитивно змішувати між собою і з генерованою нормальною перешкодою. Амплітуду другого сигналу задамо параметром A_1 , на відміну від першого сигналу, амплітуду якого ми припускаємо одиничною. Забезпечуване відношення сигнал/перешкода у всій смузі прийому для другого сигналу буде обчислюватись за формулою $snr1=snr+20 \cdot \lg(A_1)$. Для аналізу будемо використовувати мелодії різного жанру, які забезпечують більш широке використання звукового діапазону в порівнянні з мовленням.

СКВ різниці демодульованого звуку та вихідного наведено в таблиці 1.3. Перевищення заважаючого сигналу на частоті -700 кГц над прослуховуємим складає 20 дБ ($A_1=10$) (таблиця 1.3):

Таблиця 1.3

snr (дБ)	snr1 (дБ)	Фільтр		
		Хеммінга	Кайзера	Паркса-Макклеллана
20	40	0,026617	0,027171	0,025297
6	26	0,026738	0,027277	0,025425
0	20	0,027151	0,027717	0,025866
-4	16	0,030004	0,034175	0,033216

Якщо за критерій якості фільтра взяти величину, зворотну СКВ і за 100% взяти фільтр Паркса-Макклеллана, отримаємо більш зручну таблицю для аналізу (таблиця 1.4):

Таблиця 1.4

snr (дБ)	snr1 (дБ)	Фільтр		
		Хеммінга	Кайзера	Паркса-Макклеллана
20	40	95%	93%	100%
6	26	95%	93%	100%
0	20	95%	93%	100%
-4	16	111%	97%	100%

Таким чином, у всьому діапазоні відношення сигнал/перешкода, яке забезпечує комфортне прослуховування музичного контенту (0-20 дБ), фільтр Паркса-Макклеллана виявився кращим. Фільтр Хеммінга з відносно невеликим порядком поступається йому на 5%, а фільтр Кайзера – на 7%. Для слабких сигналів фільтр Хеммінга виявляється кращим, проте слухати музику і навіть мовлення з великими хрипами навряд чи зараз хтось захоче (за радянських часів у короткохвильовому діапазоні заборонені «голоси» так і доводилося слухати).

Повторення дослідження з двома іншими музичними композиціями призвело до ще більшого підкреслення переваги фільтра Паркса-Макклеллана (таблиця 1.5):

Таблиця 1.5

snr (дБ)	snr1 (дБ)	Фільтр		
		Хеммінга	Кайзера	Паркса-Макклеллана
20	40	68%	65%	100%
6	26	69%	65%	100%
0	20	71%	68%	100%
-4	16	104%	88%	100%

Для графічної інтерпретації отриманих результатів наведемо СКВ по 500 порціям прийнятої інформації (рис. 1.24):

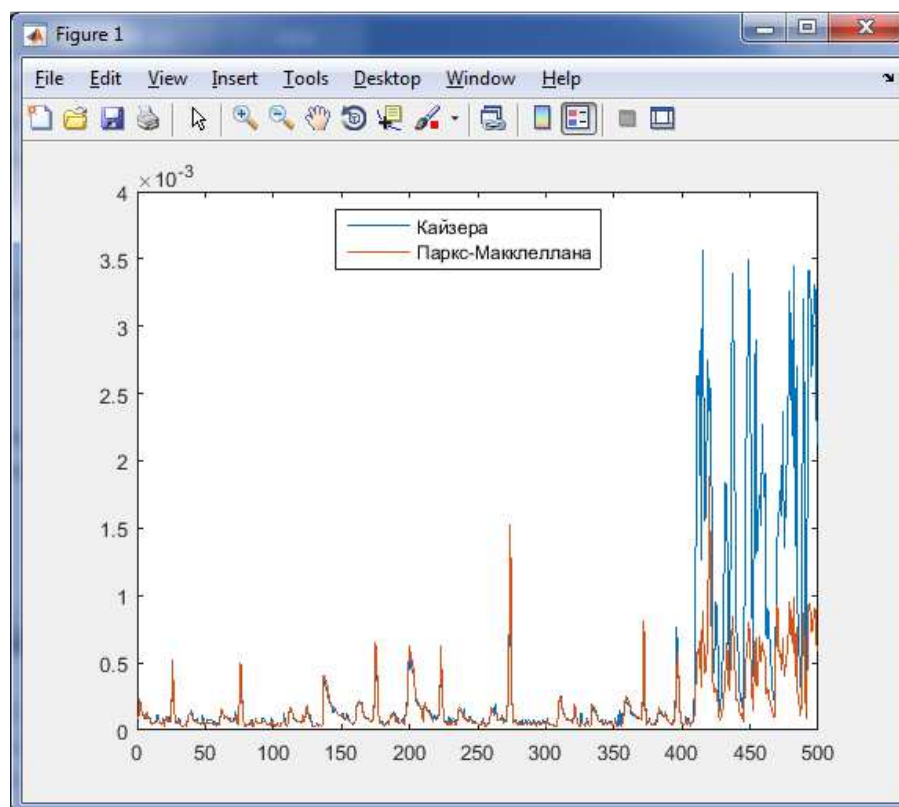


Рис. 1.24. СКВ різниці демодульованого звуку та вихідного для фільтрів Кайзера і Паркса-Макклеллана

Відзначимо, що, як і у випадку реального прийому, без децимації ми демодулюємо звук на частоті 700 кГц.

На закінчення наведемо тимчасову реалізацію вихідного звуку, в якому спостерігаються коливання не нижче 48 кГц, які для людського слуху абсолютно недоступні (рис. 1.25). Можливо, такі ультразвуки видають деякі му-

зичні інструменти, можливо, це просто збої. На графіку добре видно, як запропонований алгоритм обробки FM-сигналів їх повністю усуває (рис. 1.25):

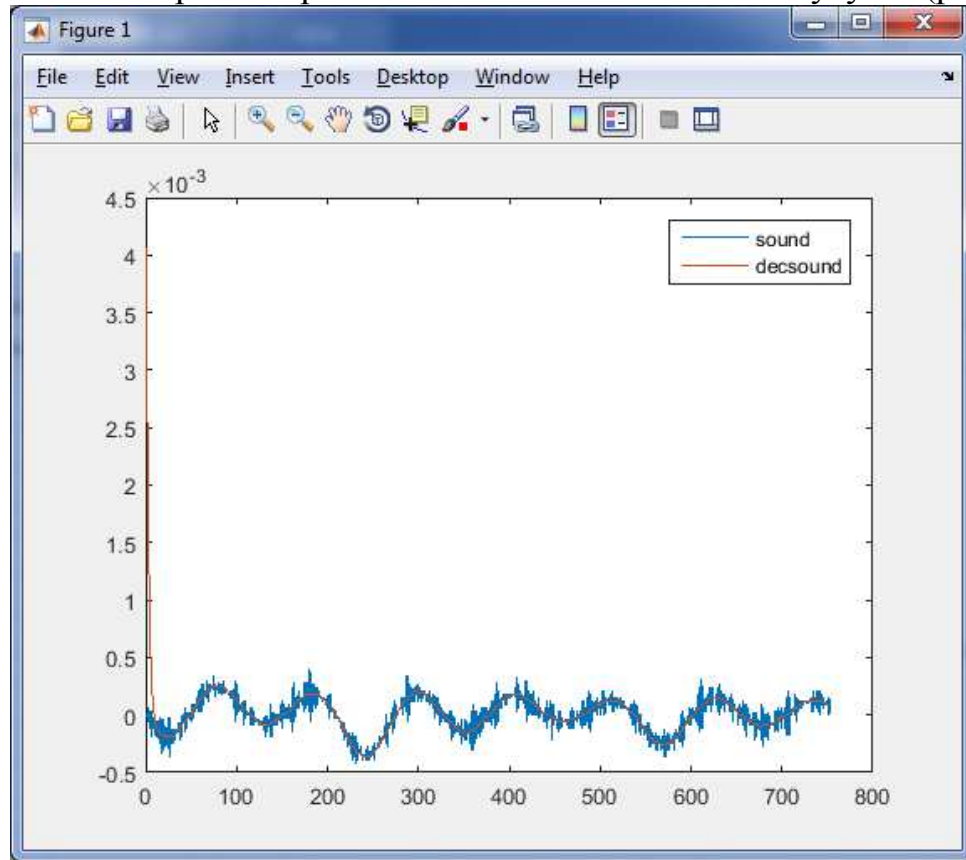


Рис. 1.25. Усунення високочастотного шуму при поліпшеній обробці

1.6. Завдання для лабораторних, курсових робіт і дипломних проектів

В MATLAB створити модель FM-радіосигналів і канал обробки некогерентного дискримінантного алгоритму для виділення моносигналу. В якості модулюючого звуку використовувати будь-який запис монозвуку у форматі WAV з частотою квантування 48 кГц.

Варіант № 1.

Дослідити вплив на якість демодульованого звуку порядку фільтра Чебишева, що використовується при зниженні частоти квантування з 2,4 МГц до 240 кГц для декількох значень відношення сигнал/перешкода. Знайдене оптимальне значення порядку фільтра випробувати при прийомі реальних FM-радіостанцій приймачем RTL-SDR, навести для фільтра графік АЧХ.

Варіант № 2.

Дослідити вплив на якість демодульованого звуку частоти зрізу фільтра Чебишева, що використовується при зниженні частоти квантування з 2,4 МГц до 240 кГц для декількох значень відношення сигнал/перешкода. Знайдене оптимальне значення частоти зрізу випробувати при прийомі реальних FM-радіостанцій приймачем RTL-SDR, навести для фільтра графік АЧХ.

Варіант № 3.

Дослідити вплив на якість демодульованого звуку частоти зрізу фільтра Кайзера, що використовується при зниженні частоти квантування з 2,4 МГц до 240 кГц для декількох значень відношення сигнал/перешкода. Знайдене оптимальне значення частоти зрізу випробувати при прийомі реальних FM-радіостанцій приймачем RTL-SDR, навести для фільтра графік АЧХ.

Варіант № 4.

Дослідити вплив на якість демодульованого звуку частоти виходу на повне пригнічення фільтра Кайзера, що використовується при зниженні частоти квантування з 2,4 МГц до 240 кГц для декількох значень відношення сигнал/перешкода. Привести залежність порядку фільтра від зазначеної частоти. Знайдене оптимальне значення зазначеної частоти випробувати при прийомі реальних FM-радіостанцій приймачем RTL-SDR, навести для фільтра графік АЧХ.

Варіант № 5.

Дослідити вплив на якість демодульованого звуку вектора **devs** (в частках ступінь наближення до заданих амплітуд) фільтра Кайзера, що використовується при зниженні частоти квантування з 2,4 МГц до 240 кГц для декількох значень відношення сигнал/перешкода. Привести залежність порядку фільтра від значень вектору. Знайдене оптимальне значення вектору **devs** випробувати при прийомі реальних FM-радіостанцій приймачем RTL-SDR, навести для фільтра графік АЧХ..

Варіант № 6.

Дослідити вплив на якість демодульованого звуку порядку фільтра Паркса-Макклеллана, що використовується при зниженні частоти квантування з 2,4 МГц до 240 кГц для декількох значень відношення сигнал/перешкода. Знайдене оптимальне значення порядку фільтра випробувати при прийомі

реальних FM-радіостанцій приймачем RTL-SDR, навести для фільтра графік АЧХ.

Варіант № 7.

Дослідити вплив на якість демодульованого звуку частоти зрізу фільтра Паркса-Макклеллана, що використовується при зниженні частоти квантування з 2,4 МГц до 240 кГц для декількох значень відношення сигнал/перешкода. Знайдене оптимальне значення частоти зрізу випробувати при прийомі реальних FM-радіостанцій приймачем RTL-SDR, навести для фільтра графік АЧХ.

Варіант № 8.

Дослідити вплив на якість демодульованого звуку частоти виходу на повне пригнічення фільтра Паркса-Макклеллана, що використовується при зниженні частоти квантування з 2,4 МГц до 240 кГц для декількох значень відношення сигнал/перешкода. Знайдене оптимальне значення зазначеної частоти випробувати при прийомі реальних FM-радіостанцій приймачем RTL-SDR, навести для фільтра графік АЧХ.

Варіант № 9.

Дослідити вплив на якість демодульованого звуку ступінь наближення до рівномірної пульсації фільтра Паркса-Макклеллана, що використовується при зниженні частоти квантування з 2,4 МГц до 240 кГц для декількох значень відношення сигнал/перешкода. Знайдене оптимальне значення ступеня наближення випробувати при прийомі реальних FM-радіостанцій приймачем RTL-SDR, навести для фільтра графік АЧХ.

Варіант № 10.

Дослідити вплив на якість демодульованого звуку порядку фільтра Паркса-Макклеллана, що використовується при зниженні частоти квантування з 240 кГц до 48 кГц для декількох значень відношення сигнал/перешкода. Знайдене оптимальне значення порядку фільтра випробувати при прийомі реальних FM-радіостанцій приймачем RTL-SDR, навести для фільтра графік АЧХ.

Варіант № 11.

Дослідити вплив на якість демодульованого звуку частоти зрізу фільтра Паркса-Макклеллана, що використовується при зниженні частоти квантування з 240 кГц до 48 кГц для декількох значень відношення сигнал/перешкода. Знайдене оптимальне значення частоти зрізу випробувати при прийомі реальних FM-радіостанцій приймачем RTL-SDR, навести для фільтра графік АЧХ.

Варіант № 12.

Дослідити вплив на якість демодульованого звуку частоти виходу на повне пригнічення фільтра Паркса-Макклеллана, що використовується при зниженні частоти квантування з 240 кГц до 48 кГц для декількох значень відношення сигнал/перешкода. Знайдене оптимальне значення зазначеної частоти випробувати при прийомі реальних FM-радіостанцій приймачем RTL-SDR, навести для фільтра графік АЧХ.

Варіант № 13.

Дослідити вплив на якість демодульованого звуку ступінь наближення до рівномірної пульсації фільтра Паркса-Макклеллана, що використовується при зниженні частоти квантування з 240 кГц до 48 кГц для декількох значень відношення сигнал/перешкода. Знайдене оптимальне значення ступеня наближення випробувати при прийомі реальних FM-радіостанцій приймачем RTL-SDR, навести для фільтра графік АЧХ.

Варіант № 14.

Дослідити вплив на якість демодульованого звуку порядку НІХ-фільтра Баттерворта, що використовується для фільтрації звуку на частоті 48 кГц для декількох значень відношення сигнал/перешкода. Знайдене оптимальне значення порядку фільтра випробувати при прийомі реальних FM-радіостанцій приймачем RTL-SDR, навести для фільтра графік АЧХ.

Варіант № 15.

Дослідити вплив на якість демодульованого звуку частоти зрізу НІХ-фільтра Баттерворта, що використовується для фільтрації звуку на частоті 48 кГц для декількох значень відношення сигнал/перешкода. Знайдене оптимальне значення частоти зрізу випробувати при прийомі реальних FM-радіостанцій приймачем RTL-SDR, навести для фільтра графік АЧХ.

2. ВІДСТЕЖЕННЯ ТРАЄКТОРІЙ ПОЛЬОТУ ПОВІТРЯНИХ СУДЕН

2.1.	Загальна інформація про методи передачі повідомлень повітряними суднами	48
2.2.	Структура повідомлень, що передаються	49
2.3.	Інформація про місцезнаходження повітряного судна	54
2.4.	Інформація про швидкість повітряного судна	57
2.5.	Алгоритм виявлення бортових сигналів повітряних суден.....	61
2.6.	Огляд експериментів по прийому бортових сигналів	78
2.7.	Завдання для лабораторних, курсових робіт і дипломних проектів.....	84

2.1. Загальна інформація про методи передачі повідомлень повітряними суднами

В даний час для спостереження за рухом повітряних суден з хорошою точністю (близько 30 м за місцем розташування) використовується технологія ADS-B – (automatic dependent surveillance-broadcast, у перекладі українською: радіомовне автоматичне залежне спостереження).

Незалежно від надходження запиту від наземного радару, кілька разів на секунду літаком передається коротке або розширене повідомлення (замість терміна «повідомлення» використовується також термін «сквіттер» — «squitter»).

Інформація завжди передається на частоті 1090 МГц з використанням амплітудної модуляції. Час передачі одного інформаційного біта складає 1 мкс. Одна посилка складається з 56 або 112 бітів даних. Посилка починається з синхроімпульсу (преамбули) тривалістю 8 мкс, і потім проходить інформаційна частина. Значення біта «0» передається відсутністю випромінювання на першій половині мікросекунди і наявністю його у другій половині, значення біта «1» — навпаки, наявністю в першій половині і відсутністю сигналу у другій половині. Загальна схема передачі наведена на рис. 2.1:

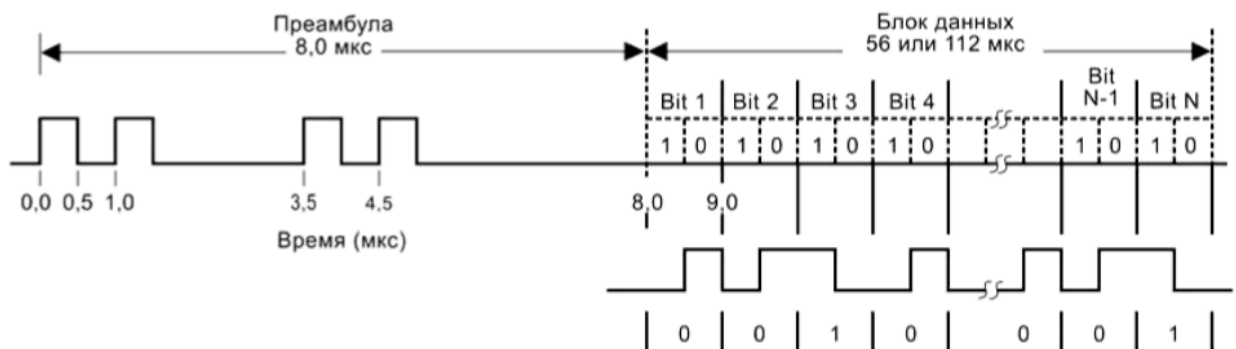


Рис. 2.1. Загальна схема передачі повідомлення в технології ADS-B

56 бітів короткого повідомлення розподіляються наступним чином: 5 бітів — номер формату, потім йдуть 27 інформаційних бітів, ці 32 біта закриваються 24-бітним кодом CRC (рис. 2.2):

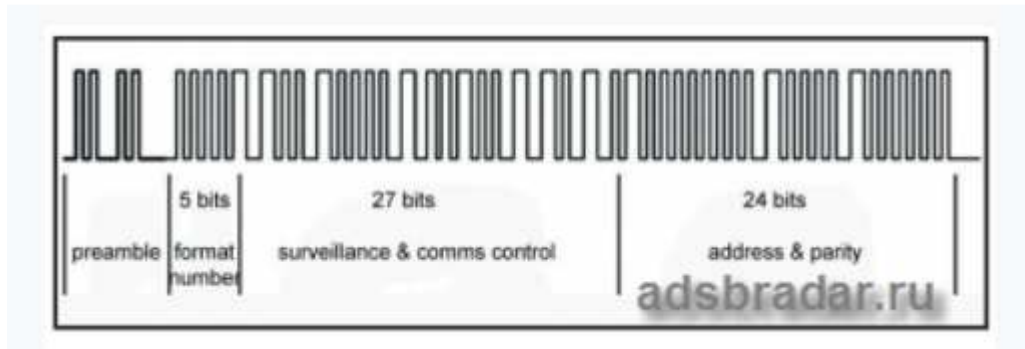


Рис. 2.2. Структура короткого повідомлення

Довге повідомлення відрізняється від короткого тим, що інформаційних бітів стає 83 (рис. 2.3).



Рис. 2.3. Структура довгого повідомлення

Перераховані параметри передачі інформації відносяться до режиму Mode-S (режими Mode-A і Mode-C використовуються для зв'язку з наземними службами).

2.2. Структура повідомлень, що передаються

Найбільш повний опис інформації, що передається наведено в [3,4]. Для довгого повідомлення номер формату дорівнює **10001** у двійковому поданні, для короткого – **01011**. У десятковому поданні DF = 17 і DF = 11. Наступні 3 біта несуть додаткову інформацію, зміст якої може змінюватися в залежності від конкретної системи, і рівні зазвичай **101** або **111**. Таким чином, 8 бітів повного формату можуть бути рівними або **8D**, або **8F** у шістнадцятковому поданні для довгого повідомлення і **5D** або **5F** – для короткого.

Коротке повідомлення містить тільки номер судна у відповідності з реєстрацією ICAO (International Civil Aviation Organization – Міжнародна організація цивільної авіації), який передається відразу за форматом (біти **9-32**, відповідно півбайти **3-8**). Номер ICAO у документах завжди записується в 16-ковому вигляді.

У довгому повідомленні номер ICAO передається на цих же позиціях, але структура повідомлення є значно складнішою. Воно може містити інформацію про розташування судна відносно поверхні землі або швидкісні характеристики борту або номер рейсу. Висота польоту видається як по барометру, так і за даними GNSS (Global Navigation Satellite Systems). Все повідомлення можна поділити на 5 інформаційних груп (таблиця 2.1):

Таблиця 2.1

Номера бітів повідомлення	Кількість бітів	Умовне позначення	Передана інформація
1-5	5	DF	Формат типу зв'язку (Downlink format, для ADS-B Mode-S 10001 або 10011)
6-8	3	CA	(Capability – additional identifier)
9-32	24	ICAO	Номер борту за реєстрацією ICAO (ICAO aircraft address)
33-88	56	DATA	Дані (Data)
89-112	24	PI	Код CRC-24 (Parity/Interrogator ID)

Для визначення типу повідомлення використовуються біти з 33 по 37 (9-ий півбайт і старший біт 10-го півбайта), відповідність значень яких конкретним типам повідомлень задається таблицею 2.2:

Таблиця 2.2

Значення бітів з 33 по 37	Інформація, що надходить
Від 00001 до 00100	Номер рейсу (Flight identification)
Від 00101 до 01000	Немає опису (Surface position)
Від 01001 до 10010	Географічні координати та висота за барометром (Airborne position – w/Baro Altitude)
10011	Швидкісні характеристики (Airborne velocities)
Від 10100 до 10110	Географічні координати та висота за даними GNSS (Airborne position – w/GNSS Height)
Від 10111 до 11111	Резерв для інших потреб (Reserved for other uses)

У бібліотеці файлів відзначеної книги [2] наведена з відкритим кодом програма **ADSBAirplaneTrackingwithRTLSDR**, яка дозволяє отримувати і декодувати повідомлення від повітряних суден, що пролітають в радіусі до 100 км. Зону прийому бортових сигналів у Миколаєві щодня перетинає більше 2 десятків повітряних суден.

Докладно розглянемо, яким чином передається номер рейсу на прикладі повідомлення, виданого 27 квітня 2017 року:

```
Time = 19:30:36
Decoded Message = 8D4BB84623407534CB7820 (код CRC видалено)
AircraftID = 4BB846
FlightID = PGT427
```

Бачимо, що номер ICAO присутній в явному вигляді. Починаючи з 33 розряду, в двійковому вигляді повідомлення виглядає наступним чином:

```
00100 011 010000 000111 010100 110100 110010 110111 100000 100000
```

Бачимо, що передається номер рейсу. Далі слідують 3 біта, що несуть інформацію про тип борту (важкий, великий, маленький, легкий, планер і т. д., розшифровка в [2] відсутня). Решта бітів розбиваються по 6 і несуть інформацію у відповідності з наступним кодуванням літерів:

Латинські літери **A – Z** десятковий код **1 – 26**, цифри **0 - 9**: код **48 – 57**, пробіл – код **32**.

В десятковому вигляді шістки бітів мають значення:

16 7 20 52 50 55 32 32, що відповідає номеру рейсу **PGT427**.

Зазначимо, що за обома номерами сайт **flightradar24.com** видає додаткову інформацію про повітряне судно та рейс. Введення номера рейсу PGT427 дозволяє дізнатися, що це рейс турецької авіакомпанії Pegasus Airlines Харків – Стамбул, який значиться тепер під номером PC427 (мабуть, відбувся перехід на нову нумерацію рейсів, але бортові передавачі залишилися на старій нумерації) (рис. 2.4):

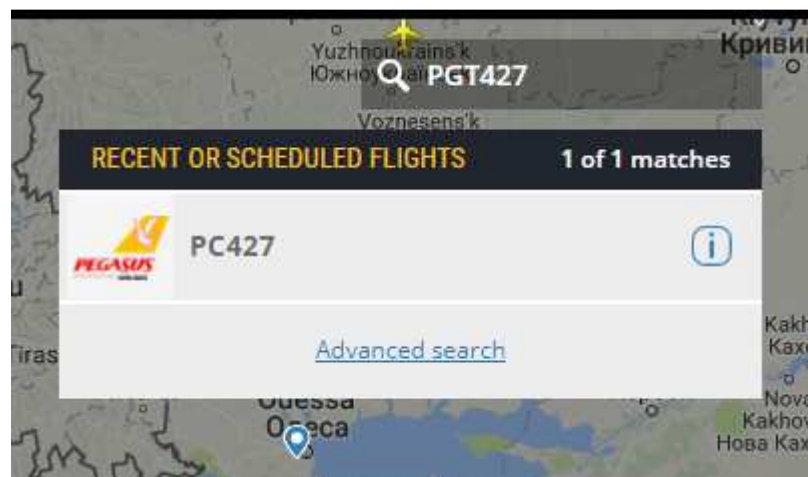


Рис. 2.4. Відповідність номера рейсу, що передається в режимі Mode-S номеру рейсу в розкладі польотів

Також можна довідатися про інформацію щодо майбутніх і раніше виконаних польотів з цього рейсу (рис. 2.5):

Flight history for Pegasus Airlines flight PC427							
DATE	FROM	TO	AIRCRAFT	FLIGHT TIME	STD	ATD	STA
19 Nov	Kharkiv (HRK)	Istanbul (SAW)	738	-	4:40 PM	-	7:55 PM
17 Nov	Kharkiv (HRK)	Istanbul (SAW)	738	-	4:40 PM	-	7:55 PM
16 Nov	Kharkiv (HRK)	Istanbul (SAW)	738	-	4:40 PM	-	7:55 PM
14 Nov	Kharkiv (HRK)	Istanbul (SAW)	738	-	4:40 PM	-	7:55 PM
12 Nov	Kharkiv (HRK)	Istanbul (SAW)	738	-	4:40 PM	-	7:55 PM
10 Nov	Kharkiv (HRK)	Istanbul (SAW)	B738 (TC-CPB)	1:55	4:40 PM	5:03 PM	7:55 PM

Рис. 2.5. Розклад польотів за маршрутом PGT427 (PC427)

Після введення номера борту дізнаємося, що судно літає з Туреччини в різні міста світу (рис. 2.6):

AIRCRAFT	TYPE CODE	MODE S						
Airbus A320-251N	A20N	4BB846						
AIRLINE	Code	SERIAL NUMBER (MSN)						
Pegasus Airlines	PC / PGT							
OPERATOR	Code	AGE						
Pegasus Airlines	PC / PGT							

DATE	FROM	TO	FLIGHT	FLIGHT TIME	STD	ATD	STA	STATUS
03 Dec	Istanbul (SAW)	Antakya (HTV)	PC4134	-	10:20 AM	11:24 AM	12:00 PM	Delayed 12:40 PM
03 Dec	Adana (ADA)	Istanbul (SAW)	PC2081	1:13	8:55 AM	9:11 AM	10:25 AM	Landed 10:25 AM
03 Dec	Istanbul (SAW)	Adana (ADA)	PC2080	1:07	6:50 AM	7:07 AM	8:20 AM	Landed 8:14 AM
02 Dec	Brussels (CRL)	Istanbul (SAW)	PC804	3:06	3:50 PM	4:21 PM	9:15 PM	Landed 9:27 PM
02 Dec	Istanbul (SAW)	Brussels (CRL)	PC803	3:03	1:35 PM	2:01 PM	3:10 PM	Landed 3:05 PM
02 Dec	Erzurum (ERZ)	Istanbul (SAW)	PC2553	2:00	11:00 AM	11:00 AM	12:55 PM	Landed 1:00 PM
02 Dec	Istanbul (SAW)	Erzurum (ERZ)	PC2552	1:31	8:45 AM	9:03 AM	10:35 AM	Landed 10:35 AM
02 Dec	Samsun (SZF)	Istanbul (SAW)	PC2797	1:09	6:50 AM	6:50 AM	8:15 AM	Landed 8:00 AM

Рис. 2.6. Розклад польотів борту 4BB846 (TC-NBF)

Також можна побачити траєкторію польоту рейсу при отриманні повідомлень. Видно, що політ проходив дуже близько над Миколаєвом (рис. 2.7):

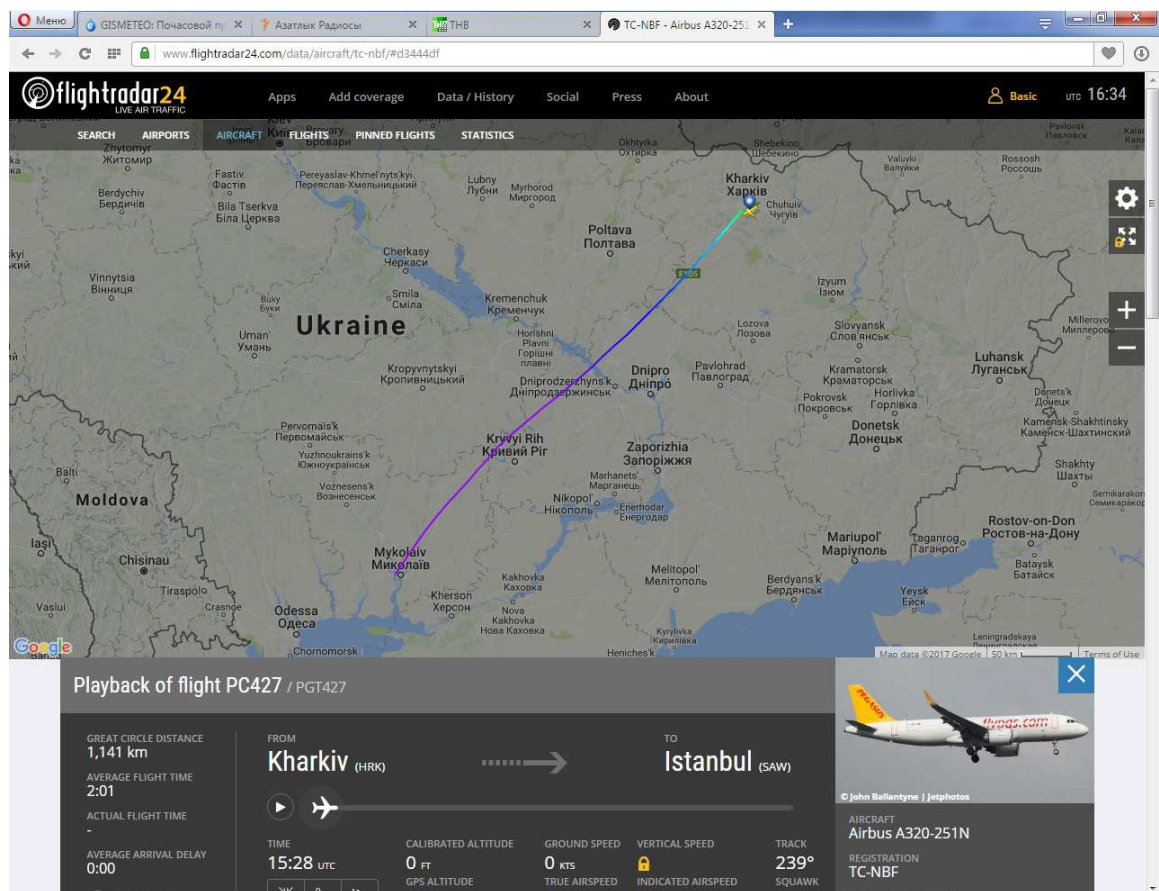


Рис. 2.7. Маршрут польоту рейсу PGT427 27 квітня 2017 року

2.3. Інформація про місцезнаходження повітряного судна

Інформація про місцезнаходження повітряного судна в повідомленнях ADS-B кодується у форматі Compact Position Reporting (CPR – компактний формат передачі розташування). Мета CPR полягає в тому, щоб кодувати більше число знаків в географічних координатах, використовуючи меншу кількість бітів. Видається, що зараз жорстка економія бітів не актуальна, але так історично склалося.

Основна ідея CPR полягає у розбитті земної поверхні на зони і передачі номера зони замість географічних координат. Кількість зон по широті від екватора до полюса **NZ=15** для режиму Mode-S. Кількість зон по довготі NL залежить від широти (lat) і задається формулою:

$$NL(lat) = \left\lceil \frac{2\pi}{\arccos\left(1 - \frac{1 - \cos\left(\frac{\pi}{2 \cdot NZ}\right)}{\cos^2\left(\frac{\pi}{180} \cdot lat\right)}\right)} \right\rceil \quad (0.1)$$

Кількість зон по довготі змінюється від 1 для полярних широт і до 59 для екваторіальних широт.

Розташування інформації в режимі передачі координат визначається за таблицею 2.3:

Таблиця 2.3

Номери бітів повідомлення	Кількість бітів	Умовне позначення	Інформація, що передається
1-5	5	DF	Формат типу зв'язку (Downlink format, для ADS-B 10001 або 10010)
33-37	5	TC	Тип повідомлення
38-39	2	SS	Статус відстеження стану (Surveillance status)
40	1	NICsb	Точність оцінки розташування (Navigation Integrity Category – NIC supplement B)
41-52	12	ALT	Висота (Altitude)
53	1	T	Час (Time)
54	1	F	Парність повідомлення у CPR форматі (CPR odd/even frame flag)
55-71	17	LAT-CPR	Широта у CPR-форматі (Latitude in CPR format)

Номери бі- тів повідо- млення	Кіль- кість бітів	Умовне по- значення	Інформація, що передається
72-88	17	LON-CPR	Довгота у CPR-форматі (Longitude in CPR format)

Зазначимо насамперед, що для початкової прив'язки судна до географічних координат воно генерує два повідомлення – парне і непарне. Надалі, знаючи попереднє місце розташування судна, інформацію про географічні координати можна отримувати з кожного повідомлення.

Спочатку розглянемо глобальну прив'язку судна до географічних координат.

1. Отримані дані $Lat_{cprEven}$, Lat_{cprOdd} , $Lon_{cprEven}$, Lon_{cprOdd} переводяться у формат з плаваючою точкою і нормуються на 2^{17} ;
2. Обчислюється *індекс широти*:

$$j = \text{floor} \left(59 \cdot Lat_{cprEven} - 60 \cdot Lat_{cprOdd} + \frac{1}{2} \right).$$

3. Обчислюється *відносна широта*:

$$Lat_{even} = dLat_{even} \cdot (\text{mod}(j, 60) + Lat_{cprEven});$$

$$Lat_{odd} = dLat_{odd} \cdot (\text{mod}(j, 59) + Lat_{cprOdd}),$$

де використані дві константи

$$dLat_{even} = \frac{360}{4 \cdot NZ} = \frac{360}{60};$$

$$dLat_{odd} = \frac{360}{4 \cdot NZ - 1} = \frac{360}{59}.$$

4. Для забезпечення приналежності широти діапазону $[-900, 900]$ при необхідності потрібно скорегувати обчислені значення:

$$Lat_{even} = Lat_{even} - 360 \text{ if } (Lat_{even} \geq 270);$$

$$Lat_{odd} = Lat_{odd} - 360 \text{ if } (Lat_{odd} \geq 270).$$

5. В якості остаточного значення широти використовується більш пізнє повідомлення:

$$Lat = \begin{cases} Lat_{even} & \text{if } (T_{even} \geq T_{odd}); \\ Lat_{odd} & \text{else.} \end{cases}$$

6. Необхідно обчислити номери зон по довготі для обох значень широти $NL(Lat_{Even})$ і $NL(Lat_{Odd})$ за формулою 2.1. Якщо вони різні, то обчислення глобальної довготи неможливо, і треба чекати наступної пари повідомлень.
7. При збігу номерів зон по довготі для парного і непарного повідомлення можна приступати до обчислення глобальної довготи. Якщо

час парного повідомлення більше часу непарного повідомлення, то обчислення наступні:

$$\begin{aligned} ni &= \max(NL(Lat_{even}), 1); \\ dLon &= \frac{360}{ni}; \\ m &= \text{floor}\left(Lon_{cprEven} \cdot [NL(Lat_{even}) - 1] - Lon_{cprOdd} \cdot NL(Lat_{even}) + \frac{1}{2}\right); \\ Lon &= dLon \cdot (\text{mod}(m, ni) + Lon_{cprEven}). \end{aligned}$$

В іншому випадку:

$$\begin{aligned} ni &= \max(NL(Lat_{odd}), 1); \\ dLon &= \frac{360}{ni}; \\ m &= \text{floor}\left(Lon_{cprEven} \cdot [NL(Lat_{odd}) - 1] - Lon_{cprOdd} \cdot NL(Lat_{odd}) + \frac{1}{2}\right); \\ Lon &= dLon \cdot (\text{mod}(m, ni) + Lon_{cprOdd}). \end{aligned}$$

8. На закінчення коректуємо значення довготи, що перевищують 180°:

$$Lon = Lon - 360 \text{ if } (Lon \geq 180).$$

Інформація про висоту судна над рівнем моря надходить в тому ж повідомленні, що і географічні координати, і обчислити по ній значення висоти досить просто. Обчислення не залежать від парності повідомлення і завжди наводяться у футах. У даних, що надійшли, міститься інформація про точності вимірювання висоти на судні. Цю інформацію несе 48-й, так званий Q-біт, якщо його значення дорівнює 0, то дискретність висоти *discr* дорівнює 100, інакше 25. Q-біт необхідно видалити і отримане значення *N* перерахувати у фути:

$$Alt = N \cdot discr - 1000 \text{ (футів)}.$$

Згідно з наведеною формулою та кількістю розрядів, що несуть інформацію про висоту польоту (11 бітів), діапазон висот, який може бути закодований, становить від -1000 до $(2^{11}-1) \cdot 25 - 1000 = 50175$ футів або від -305 до 15293 метрів для Q біта=1.

Локальна прив'язка судна до географічних координат передбачає знання опорної точки, в околиці якої завідомо знаходиться судно. Це можуть бути дані, що коректно отримані з попередніх повідомлень, або дані про місцезнаходження приймальної антени. При цьому може бути використано повідомлення з будь-якою парністю.

1. Обчислити широту:

$$Lat = dLat \cdot (J + Lat_{cpr}),$$

де

$$dLat = \begin{cases} \frac{360}{4 \cdot NZ} = \frac{360}{60} & \text{if even message;} \\ \frac{360}{4 \cdot NZ - 1} = \frac{360}{59} & \text{if odd message,} \end{cases}$$

$$j = \text{floor}\left(\frac{Lat_{ref}}{dLat}\right) + \text{floor}\left(\frac{\text{mod}(Lat_{ref}, dLat)}{dLat} - Lat_{cpr} + \frac{1}{2}\right),$$

Lat_{ref}, Lon_{ref} – широта і довгота опорної точки.

2. Обчислити довготу:

$$Lon = dLon \cdot (m + Lon_{cpr}),$$

де

$$dLon = \begin{cases} \frac{360}{NL(Lat)} & \text{if } NL(Lat) > 0; \\ 360 & \text{if } NL(Lat) = 0, \end{cases}$$

$$m = \text{floor}\left(\frac{Lon_{ref}}{dLon}\right) + \text{floor}\left(\frac{\text{mod}(Lon_{ref}, dLon)}{dLon} - Lon_{cpr} + \frac{1}{2}\right).$$

2.4. Інформація про швидкість повітряного судна

Згідно з таблицею 2.2. тип повідомлення, що містить швидкісні характеристики судна, дорівнює 10011 або 19 в десятковому поданні. Ці повідомлення можуть бути 4 підтипів, при підтипі 1 і 2 передається швидкість щодо Землі (surface velocity ground or speed), при підтипі 3 і 4 передається повітряна швидкість (airspeed). Причому підтипи 2 і 4 призначені для надзвукових літаків, і їх структурою найближчим часом можна не цікавитися. В [2] зазначається, що повідомлення 3-го підтипу складають всього 0.3% від повідомлень 1-го типу. Підтип передається в бітах 38-40. Розміщення численних складових вектору швидкості для підтипу 1 наведено в таблиці 2.4:

Таблиця 2.4

Номери бітів повідомлення	Кількість бітів	Умовне позначення	Інформація, що передається
33-37	5	TC	Тип повідомлення (Type code)
38-40	3	ST	Підтип повідомлення (Subtype)
41	1	IC	Прапор зміни наміру (Intent change flag)
42	1	RESV_A	Резерв (Reserved-A)
43-45	3	NAC	Невизначеність швидкості (Velocity)

Номери бітів повідомлення	Кількість бітів	Умовне позначення	Інформація, що передається
			uncertainty – NAC)
46	1	s_{ew}	Знак швидкості в напрямку Схід-Захід (East-West velocity sign)
47-56	10	v_{ew}	Швидкість у напрямку Схід-Захід (East-West velocity)
57	1	s_{ns}	Знак швидкості в напрямку Північ-Південь (North-South velocity sign)
58-67	10	v_{ns}	Швидкість у напрямку Північ-Південь (North-South velocity)
68	1	VrSrc	Метод отримання вертикальної швидкості (Vertical rate source)
69	1	s_{vr}	Знак вертикальної швидкості (Vertical rate sign)
70-78	9	Vr	Вертикальна швидкість (Vertical rate)
79-80	2	RESV_B	Резерв (Reserved-B)
81	1	s_{Dif}	Відмінність від даних по барометру, знак (Diff from baro alt, sign)
82-88	7	Dif	Відмінність від даних по барометру (Diff from baro alt)

З наведеної таблиці ми бачимо, що вектор швидкості розкладений на компоненти Північ-Південь і Схід-Захід. Знак швидкості **s_{ns} = 1** означає політ з Півночі на Південь, **s_{ns} = 0** – політ з Півдня на Північ, аналогічно, **s_{ew} = 1** – політ зі Сходу на Захід, **s_{ew} = 0** – політ із Заходу на Схід.

Обчислення вектора горизонтальної швидкості здійснюється за такими формулами:

$$\begin{aligned}
 V_{we} &= \begin{cases} -1 \cdot (V_{ew} - 1) & \text{if } s_{ew} = 1; \\ V_{ew} - 1 & \text{if } s_{ew} = 0; \end{cases} \\
 V_{sn} &= \begin{cases} -1 \cdot (V_{ns} - 1) & \text{if } s_{ns} = 1; \\ V_{ns} - 1 & \text{if } s_{ns} = 0; \end{cases} \\
 v &= \sqrt{V_{we}^2 + V_{sn}^2};
 \end{aligned}$$

$$h = \arctan2(V_{we}, V_{sn}) \cdot \frac{360}{2\pi} \text{ (deg)};$$

$$h = h + 360 \text{ if } h < 0.$$

Швидкість вимірюється у вузлах (1 вузол = 1,852 км/год), напрям швидкості h – в градусах, $\arctan2()$ – відома функція, реалізована в багатьох програмних середовищах [5].

Вертикальну швидкість прийнято міряти в футах/хвилину, і вона пов'язана з прийнятим значенням простою формулою:

$$V_r(\text{футів/хв}) = (V_r \text{ bits} - 1) \cdot 64.$$

Якщо $s_{vr}=1$, судно збільшує висоту польоту, інакше – зменшує.

Розряд **VrSrc** визначає метод вимірювання вертикальної швидкості, якщо **VrSrc=1**, то швидкість отримана вимірами геометричної висоти, інакше – вимірами висоти по барометру.

У разі якщо швидкість щодо Землі недоступна, у повідомленнях борту передається повітряна швидкість. Підтип повідомлення, як зазначалося вище, при цьому дорівнює 3. Розміщення складових цієї швидкості багато в чому аналогічні підтипу 1 і наведені в таблиці 2.5:

Таблиця 2.5

Номери бітів повідомлення	Кількість бітів	Умовне позначення	Інформація, що передається
33-37	5	TC	Тип повідомлення (Type code)
38-40	3	ST	Підтип повідомлення (Subtype)
41	1	IC	Прапор зміни наміру (Intent change flag)
42	1	RESV_A	Резерв (Reserved-A)
43-45	3	NAC	Невизначеність швидкості (Velocity uncertainty – NAC)
46	1	s_hdg	Статус напрямку польоту (Heading status)
47-56	10	Hdg	Напрямок польоту – пропорція (Heading – proportion)
57	1	AS-t	Тип повітряної швидкості (Airspeed Type)
58-67	10	AS	Повітряна швидкість (Airspeed)

Номери бітів повідомлення	Кількість бітів	Умовне позначення	Інформація, що передається
68	1	VrSrc	Метод отримання вертикальної швидкості (Vertical rate source)
69	1	S_vr	Знак вертикальної швидкості (Vertical rate sign)
70-78	9	Vr	Вертикальна швидкість (Vertical rate)
79-80	2	RESV_B	Резерв (Reserved-B)
81	1	S_Dif	Відмінність від даних по барометру, знак (Diff from baro alt, sign)
82-88	7	Dif	Відмінність від даних по барометру (Diff from baro alt)

Розряд **s_hdg=1** показує доступність даних по напрямку польоту, і їх можна перевести в градуси за формулою:

$$heading = Decimal(Hdg)/1024 \cdot 360^{\circ}.$$

Повітряна швидкість може бути двох типів: індикаторна (Indicated airspeed – IAS), що зчитується безпосередньо з індикатора, і справжня (true airspeed – TAS), щодо повітряних мас навколо літака. При **AS-t=0** передається індикаторна швидкість. Значення швидкості у вузлах виходить простим перекладом в десяткову систему 10 бітів значення **AS**.

Обчислення вертикальної швидкості залишається таким же, як і для підтипу 1.

На закінчення розділу зазначимо, що у повідомленні також доступна точність вимірювання параметрів руху літака. Для цього використовуються служби навігаційної системи GPS:

- NIC – Navigation Integrity Category – цілісність коду навігації;
- NAC – Navigation Accuracy Category – точність коду навігації.

Наприклад, у даних про місцезнаходження судна в таблиці 2.3 є розряд **NICsb**, який разом з типом повідомлення **TC** показує точність вимірювань в метрах (таблиця 2.6):

Таблиця 2.6

ТС	NICsb	Помилка виміру (м), не більше
9	0	7.5
10	0	25
11	1	74
	0	185
12	0	370
13	1*	556
	0	925
	1**	1111
14	0	1852
15	0	3704
16	1	7408
	0	14800
17	0	37000
18	0	Понад 37 км або невідома

* NIC Supplement-A = 0

** NIC Supplement-A = 1

В таблиці використовується додаткова інформація про розряді NIC Supplement-A, який разом з розрядом NIC Supplement-C передається при **ТС=31** (таблиця 2.2 – рядок «Резерв для інших потреб»), таке повідомлення отримало назву «Повідомлення про експлуатаційний статус літака» (Aircraft Operational Status Message). На жаль, в [3,4] структура такого повідомлення не наведена.

2.5. Алгоритм виявлення бортових сигналів повітряних суден

Після детального розгляду методів передачі і кодування інформації у повідомленнях, ми можемо перейти до найцікавішої частини розділу, а саме до розгляду алгоритму виявлення таких сигналів з виходу приймача RTL-SDR.

Як вже зазначалося у вступній частині розділу, в додатку до книги [2] наведена програма з відкритим кодом **ADSBAirplaneTrackingwithRTLSDR**, яка вирішує поставлену задачу (авторські права на програму належать корпорації MathWorks).

Розглянемо, насамперед, які програма встановлює параметри на приймач:

```
%% PARAMETERS (edit)
% ідентифікатор пристрою (stick ID):
rtlsdr_id          = '0';
% Робоча частота приймача в Гц (tuner center frequency in Hz):
rtlsdr_fc          = 1090e6;
% Підсилення приймача в дБ (tuner gain in dB):
rtlsdr_gain        = 60;
% Частота квантування вхідного сигналу – тактова частота (tuner
sampling rate), максимальне значення 3.2e6
rtlsdr_fs          = 2.4e6;
% Коригування частоти (tuner parts per million correction):
rtlsdr_ppm         = 0;
% Розмір запитуваного за 1 раз об'єму вхідної інформації фрей-
му(output data frame size):
rtlsdr_frmrlen     = 2^18;
% Формат вихідних квадратур (output data type single) – аналог
float (4 байта):
rtlsdr_datatype    = 'single';
```

Як зазначалося вище, частота випромінювання сигналів для всіх літаків дорівнює 1090 МГц. Наступний дуже важливий параметр — це частота квантування вхідних сигналів. У штатних системах прийому бортових повідомлень ADSB використовується частота квантування 12 МГц, яка забезпечує отримання 12 відліків на тривалості передачі 1 біта. У нас же встановлена частота квантування 2,4 МГц, це відповідає 2,4 відліку на біт. По-перше, дрібна кількість відліків призведе до неоднакової кількості відліків на біт (2 або 3 відліку). По-друге, це запевне мало. Однак можливості збільшення частоти для даної моделі RTL-SDR обмежені величиною 3,2 МГц, причому попереджається, що при перевищенні частоти квантування 2,8 МГц можливі втрати відліків. Звичайно, є приймачі, що дозволяють знімати інформацію з більш високою частотою квантування, наприклад, приймач BladeRF – the USB 3.0 Super speed Software Defined Radio коректно працює до частоти 40 МГц. Однак його вартість становить від \$420 до \$650, тоді як найпростіші RTL-SDR доступні за ціною близько \$20. У розглянутій програмі після прийому інформації її інтерполюють на ту ж стандартно прийняту частоту 12 МГц. Інтерполяцію, безумовно, неможливо зробити без відчутної втрати якості алгоритму. Ці втрати надалі будуть оцінені. Довжина фрейму (розмір запитуваного за 1 раз обсягу вхідної інформації) досить велика і становить 262144 комплексних відліків. Один комплексний відлік займає 8 байтів у приймальному буфері.

Як зазначалося у вступі до посібнику, прийняті від АЦП приймача RTL-SDR 8-бітні значення квадратур перетворюються в тип `single` – змінну з плаваючою комою, яка займає 4 байти на кожен квадратур. Таким чином, один фрейм займає 2048 Кб у вхідному буфері. Час його формування становить близько 0.109 с.

Програма **ADSBAirplaneTrackingwithRTLSDR** досить об'ємна. Текст програми розміщений у 18 файлах і містить близько 50 функцій. Складається вона з трьох основних частин. У першій частині щодо виявленого сигналу формується повідомлення у 16-ковому вигляді, у другій частині з нього витягуються параметри польоту і третя частина забезпечує текстове і графічне відображення отриманих даних.

Найбільш цікава, природно, перша частина, оскільки оптимальність алгоритмів виявлення реальних сигналів завжди може бути поставлена під сумнів і запропоновано удосконалення алгоритму, що покращують його характеристики виявлення. Пов'язано це з тим, що оптимальні алгоритми виявлення вдається синтезувати для дуже вузького класу імовірнісних розподілів корисного сигналу і перешкод, що впливають на нього. Ще більш істотно, що будь-яка запропонована статистична модель сигналів і перешкод ніколи не буде повністю адекватною реальності.

Для докладного аналізу запропонованого алгоритму виділення бортових сигналів ADSB, перепишемо його у вигляді одного модуля з мінімальними змінами самого алгоритму:

```
function ReceiveAirplaneInf
%% PARAMETERS (edit)
rtlsdr_id = '0';           % stick ID
rtlsdr_fc = 1090e6;        % tuner centre frequency in Hz
%1.6570293e9; 623.225e6
rtlsdr_gain = 60;         % tuner gain in dB
rtlsdr_fs = 2.4e6;        % tuner sampling rate, max 3.2e6
rtlsdr_ppm = 0;           % tuner parts per million correction
rtlsdr_frmrlen = 2^18; % output data frame size (multiple of 5)
rtlsdr_datatype = 'single'; % output data type single - аналог
%float (4 байта)
ADCRate=rtlsdr_fs;
%% SYSTEM OBJECTS (do not edit)

RealInf=0; % признак прийому вхідної інформації 0 - з файлу, 1 -
%RTL-SDR
% link to a physical rtl-sdr
if RealInf
    obj_rtlsdr = comm.SDRRTLReceiver(...
        rtlsdr_id,...
        'CenterFrequency', rtlsdr_fc,...
        'EnableTunerAGC', false,...
        'TunerGain', rtlsdr_gain,...
```

```

'SampleRate', rtlsdr_fs, ...
'SamplesPerFrame', rtlsdr_frmlen,...
'OutputDataType', rtlsdr_datatype,...
'FrequencyCorrection', rtlsdr_ppm);

% Перевіряємо активність RTL-SDR:
if isempty(sdrinfo(obj_rtlsdr.RadioAddress))
    error(['RTL-SDR failure. Please check connection to ',...
          'MATLAB using the "sdrinfo" command.']);
end
end

% Інтерполяція:
interpolationFactor = 5;
hInterp = dsp.FIRInterpolator(interpolationFactor);
samplingRate = interpolationFactor*ADCRate;
searchThr = 5.0; % поріг по амплитуді відносно середнього рівня
                % перешкод для пошуку синхроімпульсу

% Обчислюємо час прийому одного фрейму
% (calculate time for 1 frame of data):
rtlsdr_frmtime = rtlsdr_frmlen/rtlsdr_fs;

% set rec_time and reset run_time to 0 (secs):
rec_time = 3600; % час роботи програми (с)
run_time = 0;    % поточний лічильник часу роботи (с)

dataDuration = 112e-6; % Тривалість повного повідомлення в
% секундах (112 мкс)
% Параметри сигналу и обробки після інтерполяції
% кількість відліків повного повідомлення (1344=112*12):
messageLength = dataDuration * samplingRate;
% відступ перед синхроімпульсом (5 мкс, 60 відліків):
samplesBeforeSyncPulse = round(samplingRate*5e-6);
% відступ після повідомлення (10 мкс, 120 відліків):
samplesAfterMessage = round(samplingRate*10e-6);
% загальна довжина зони пошуку сигналу (1524 відліка):
totalSegmentLength =
samplesBeforeSyncPulse+messageLength+samplesAfterMessage;
% індекс початку пошуку у фреймі (61):
startIdx = samplesBeforeSyncPulse+1;
messageLength1 = messageLength + samplesAfterMessage - 1;

corrThreshold = 0.60; %0.68; % поріг по коефіцієнту кореляції

% синхроімпульс, прив'язаний до половини часу випромінювання
% одного бита (0.5 мкс - 6 відліків):
syncSeq = 2*([1 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0])-1;
% половина часу випромінювання одного бита (0.5 мкс):
halfSymbolTime = 0.5e-6;

```

```

% кількість відліків на половині часу випромінювання бита
% (6 відліків):
samplesPerHalfSymbol = halfSymbolTime * samplingRate;
% синхроімпульс, прив'язаний до одного відліку:
sync = rectpulse(syncSeq, samplesPerHalfSymbol);
% кількість відліків на синхроімпульсі (96 відліків):
syncLength = length(sync);
% нормувальний коефіцієнт синхроімпульсу:
syncNorm = norm(sync);
% інтервал пошуку синхроімпульсу (180 відліків):
numSamples = 15*samplingRate*1e-6;
% індекс останнього відліку від початку зони пошуку
% для обчислення кореляції з синхроімпульсом:
jEnd = numSamples-syncLength+1;
% масив для коефіцієнтів кореляції:
normCorr = zeros(jEnd,1);
longMessageLength = 112; %us - мкс
shortMessageLength = 56; %us - мкс
% кількість відліків на біт (12) :
samplesToBit = samplingRate*1e-6;
% масив для зберігання декодованих бітів:
decodedBits = false(longMessageLength,1);

% шаблон повідомлення в 16-ому вигляді:
rxMsg = '8D00000000000000000000000000';
% структура для декодованих параметрів польоту:
Data.ReceivedMessage = rxMsg;
Data.AircraftID = '';
Data.Speed = [];
Data.Direction = '';
Data.UpDownVelocity = [];
Data.Altitude = [];
Data.Latitude = [];
Data.Longitude = [];
Data.FlightID = '';
Data.CRC_OK = false;
Data.normCorr = 0;

messageTypeFlag.Detected = 0;
messageTypeFlag.Long = 0;
messageTypeFlag.Short = 0;
numMessageLong=0;
numMessageShort=0;

% шлях до файлу з записом даних RTL-SDR:
fileID = fopen('d:\MatLab\RTL-SDR\AIRPLANES\20171123-
24\Frame20171124.bin');
% номер фрейму у запису:
nFrame=0;

% цикл до закінчення часу роботи програми

```

```

% (loop while run_time is less than rec_time)
while run_time < rec_time
    % прийом фрейму від RTL-SDR
    % (fetch a frame from obj_rtlsdr):
    tic
    if RealInf
        rawData = abs(step(obj_rtlsdr));
    else
        % читання фрейму з файлу:
        [rawData,count]=fread(fileID,rtlsdr_frmlen,'single');
        if count < rtlsdr_frmlen
            fprintf('nFrame=%d, numMessageLong=%d,...
                numMessageShort=%d\n',nFrame,...
                numMessageLong,numMessageShort);
            error('End of file!');
        end
    end
end

% інтерполяція прийнятої інформації фрейму:
capturedData = abs(step(hInterp,rawData));
% кінцевий індекс оброблення всередині фрейму:
endIdx = length(capturedData) - messageLength1-1;

if mod(nFrame,100) == 0
    fprintf('numMessageLong=%3d,numMessageShort=%3d\n',...
        numMessageLong,numMessageShort);
end
nFrame=nFrame+1;

% Шукаємо амплітуди, які перевищують середній рівень перешкод
% в задану кількість разів (встановлено в 5 разів - 14 дБ)
% (Find values in the data set that stick up above the
% noise floor)
Athr= searchThr*mean(capturedData);
i=startIdx;
while i <= endIdx
    if capturedData(i) >= Athr
        % Відбираємо амплітуди з зони можливої
        % присутності повідомлення:
        msgSamples = capturedData(i -
samplesBeforeSyncPulse:...
        i + messageLength1);
        % Операція не впливає на коефіцієнт
        % кореляції normCorr!!!
        % msgSamples = msgSamples - mean(msgSamples);

        % Перевіряємо можливість присутності синхроімпульсу
        % в перших 180 відліках відібраної інформації -
        % розрахунок коефіцієнту кореляції
        % (Check the first 180 samples to see if the sync
        % pulse is present.

```

```

% Normalized correlation computation):
for j = 1:jEnd
    msg = msgSamples(j:j+syncLength-1);
    msg = msg - mean(msg);
    num = sync*msg;
    msgNorm = norm(msg);
    den = syncNorm*msgNorm;
    normCorr(j) = num/den;
end
if any(normCorr > corrThreshold)
    [~,ind] = max(normCorr);
    decodeIndex = ind+syncLength;
    if decodeIndex > numSamples
        decodeIndex = numSamples;
    end

    % Від відліку з максимальною кореляцією обираємо
    % амплітуди на інтервалі довгого повідомлення,
    % відступивши від синхроімпульсу:
    msgBuffer =
    msgSamples(decodeIndex:decodeIndex+messageLength-1);
    fprintf('normCorr=%.2f\n',normCorr(ind));

    % Витяг бітової інформації - «1» кодується 6
    % відліками (0.5 мкс) з увімкненим сигналом та
    % потім 6 відліками з вимкненим сигналом,
    % «0» - навпаки (Bit «1» is represented with 1/2
    % usec ON and 1/2 usec OFF,
    % while bit «0» is vise versa.
    k1=0;
    for k = 1:samplesToBit:messageLength
        k1=k1+1;
        if sum(msgBuffer(k:k+5)) <
            sum(msgBuffer(k+6:k+11))
            decodedBits(k1) = false;
        else
            decodedBits(k1) = true;
        end
    end

    % Об'єднання отриманих даних по 4 бита и переклад
    % в десятковий и 16-оковий вигляди:
    c = reshape(decodedBits,4,numel(decodedBits)/4);
    d = bin2dec(num2str(c));
    rxBytes = dec2hex(d)';
    fprintf('AllMessage=%s\n',rxBytes);

    msgType = 0;
    Data.CRC_OK=false;
    messageTypeFlag.Detected = 0;
    messageTypeFlag.Long = 0;

```

```

messageTypeFlag.Short = 0;
% Перевірка коду CRC у припущенні отримання
% довгого повідомлення:
[decodedBits,err] = crc24f(decodedBits);
if err == 0
    msgType = 1;
    messageTypeFlag.Long = 1;
    numMessageLong = numMessageLong+1;
    rxBytes = rxBytes(1:22);
else % якщо код CRC не співпав, перевіряємо
    % гіпотезу об отриманні короткого
    % повідомлення:
    [decodedBits,err] = crc24f(decodedBits(1:
        shortMessageLength));
    if err == 0
        msgType = 2;
        messageTypeFlag.Short = 1;
        numMessageShort = numMessageShort+1;
        rxBytes = rxBytes(1:8);
    end
end
decodeIndexInFrame=i-samplesBeforeSyncPulse
    +decodeIndex;
fprintf('nFrame=%d, decodeIndexInFrame=%d\n',
    nFrame,decodeIndexInFrame);
fprintf('msgType=%d\n',msgType);
if msgType == 1 || msgType == 2
    messageTypeFlag.Detected = 1;
    Data.ReceivedMessage = rxBytes;
    Data.AircraftID = rxBytes(3:8);
    Data.CRC_OK=true;
    Data.normCorr =normCorr(ind);
    adsbLogDataToFile1(Data);
    i=i+totalSegmentLength-1;
end
i=i+jEnd+1;
end % if any(normCorr > corrThreshold)
end % if capturedData(i) >= Athr
i=i+1;
end % if i <= endIdx
toc
% update run_time after processing another frame
run_time = run_time + rtlsdr_frmttime;

end

end

% Функція для перевірки коду CRC:
function [y,err] = crc24f(m)

```

```

persistent hcrc;
if isempty(hcrc)
    g=logical([1 0 0 1 0 0 0 0 0 0 1 0 1 1 1 1 1 1 1 1 1 1]);
    gflip = fliplr(g);
    hcrc = comm.CRCDetector(gflip);
end

reset(hcrc);
[y,err] = step(hcrc,m);
end

```

Звернімо увагу насамперед на те, що інформація з виходу приймача відразу ж пропускається через квадратичний детектор (комплексні відліки $z = x + jy$ замінюються на амплітуди $abs(z) = \sqrt{x^2 + y^2}$). Детектування сигналу в самому початку обробки не є безперечною. У багатьох системах виявлення сигналів з відомою модуляцією (в нашому випадку необхідно виявити синхроімпульс) проводять кореляційну обробку в квадратурах. Для розглянутих сигналів доцільність квадратичного детектування на вході вимагає окремого дослідження.

У програмі проведено доопрацювання, що дозволяє проводити запис амплітуд фреймів, у яких відбулося перевищення порогу по кореляції з синхроімпульсом, відповідно, у наведеній програмі є можливість отримувати амплітуди з файлу. Безумовно, для аналізу алгоритму прийому бортових повідомлень ця можливість абсолютно необхідна. Зверніть увагу, що запис проводиться після детектування, а не вихідних квадратур. Обсяг фрейму при цьому зменшується в 2 рази і складає 1024 КБ.

Амплітуди на виході квадратичного детектора надходять на інтерполяцію для підвищення частоти квантування з 2,4 МГц до 12 МГц, тобто між кожними двома відліками вставляються ще 4 обчислених відліка. Займається цим досить цікава функція **dsp.FIRInterpolator()**. Додаткові відліки будуються на базі реконструкційного або відновлювального СІХ-фільтра (reconstruction or anti-imaging FIR-filter). При подачі на вхід інтерполятора звичайної синусоїди ми отримуємо невеликий зсув у часі, стиснення сигналу по амплітуді (коефіцієнт стиснення в даному випадку збігається з фактором інтерполяції, що дорівнює 5) і невеликі високочастотні биття (рис. 2.8):

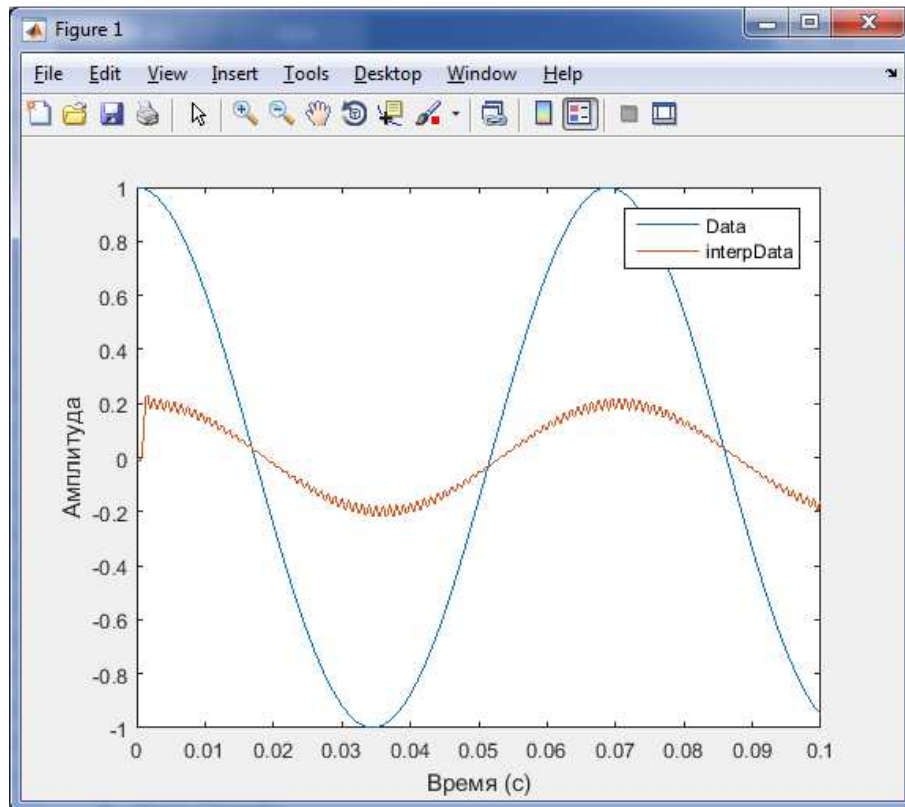


Рис. 2.8. Результат інтерполяції гармонічного сигналу

При інтерполяції реального сигналу ми бачимо схожу картину (рис. 2.9, 2.10):

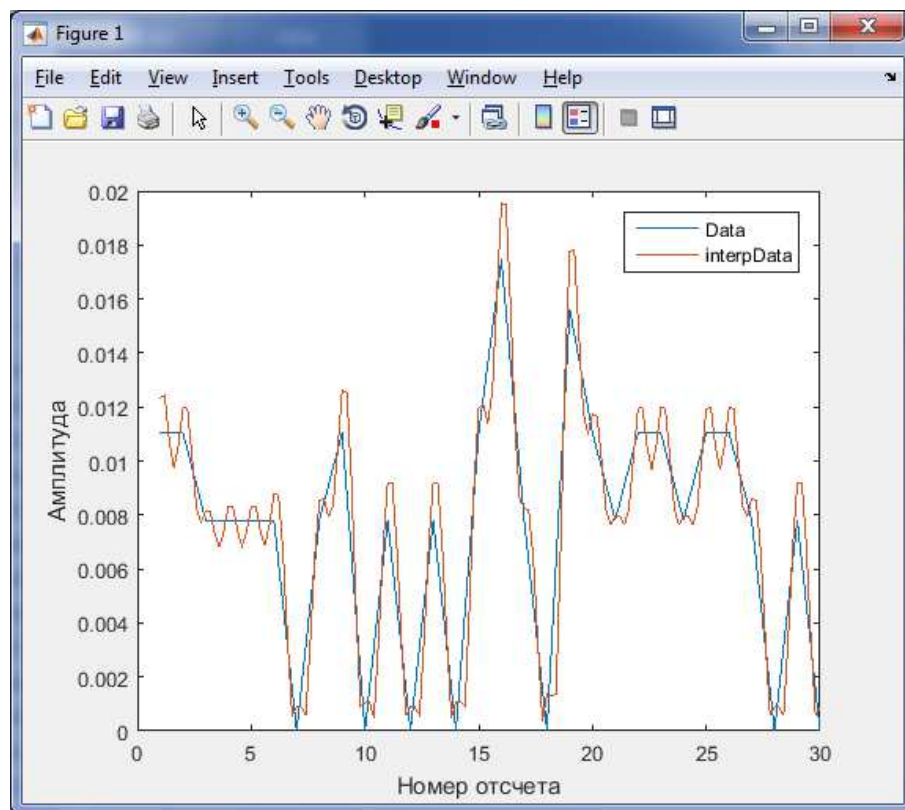


Рис. 2.9. Результат інтерполяції реального сигналу (вибірка 1)

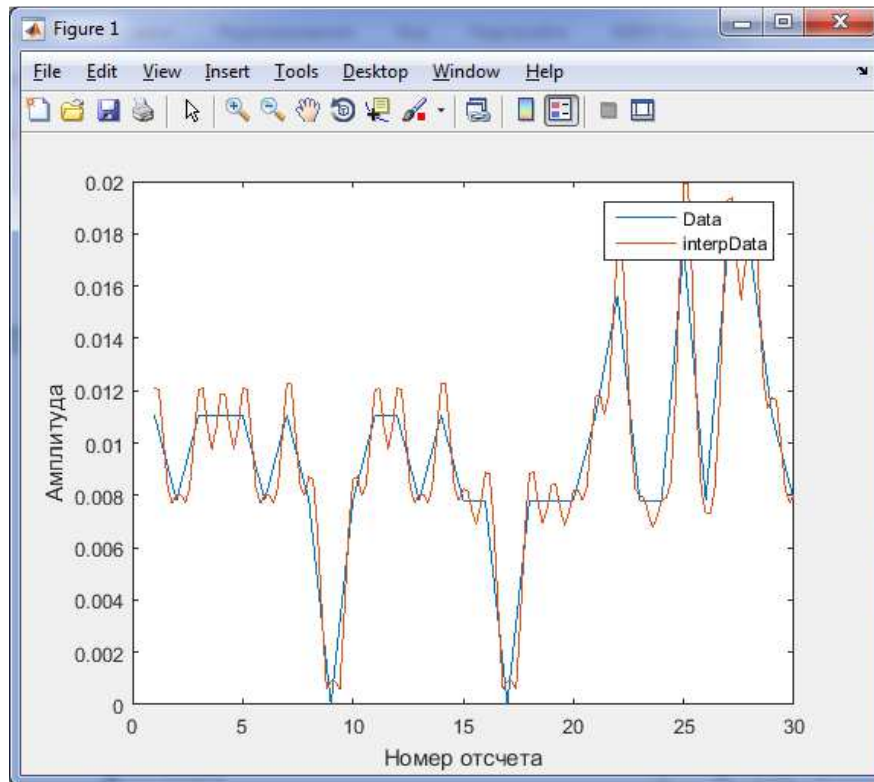


Рис. 2.10. Результат інтерполяції реального сигналу (вибірка 2)

При інтерполяції виникає затримка вихідного сигналу відносно вхідного приблизно на 7 відліків (на графіках ця затримка прибрана, сигнали також приведені до одного масштабу амплітуди). Бачимо, що на інтервалах сталості вихідного сигналу інтерполяція проводиться коливальним процесом. Ясно, що спроба вгадати проміжні значення вхідного модульованого сигналу (сигнал постійно гаситься на половині тривалості передачі одного біта) заздалегідь приречена на серйозні помилки. Тим не менш, алгоритм, побудований на такій інтерполяції, дає цілком задовільні характеристики виявлення бортових повідомлень. Природно, виникає питання про ефективність інших варіантів інтерполяції, наприклад, набагато більш простої лінійної інтерполяції або інтерполяції невідомих значень найближчим виміряним значенням сигналу.

Як вже вказувалося вище і пояснено в коментарях до тексту програми, після інтерполяції ми маємо частоту квантування 12 МГц, кількість відліків у фреймі дорівнює 1310720, на інтервалі 1 мкс передачі одного біта ми маємо 12 відліків. Зображена графічно інформація фрейма виглядає, наприклад, так (рис. 2.11–2.13):

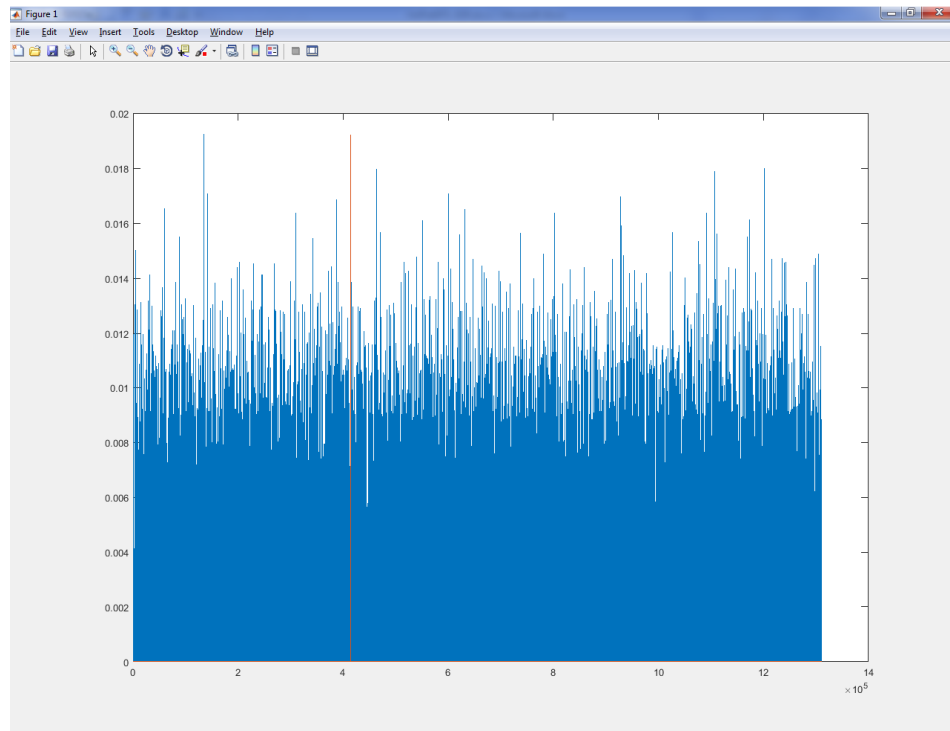


Рис. 2.11. Один з фреймів сеансу за 13.07.2017 року

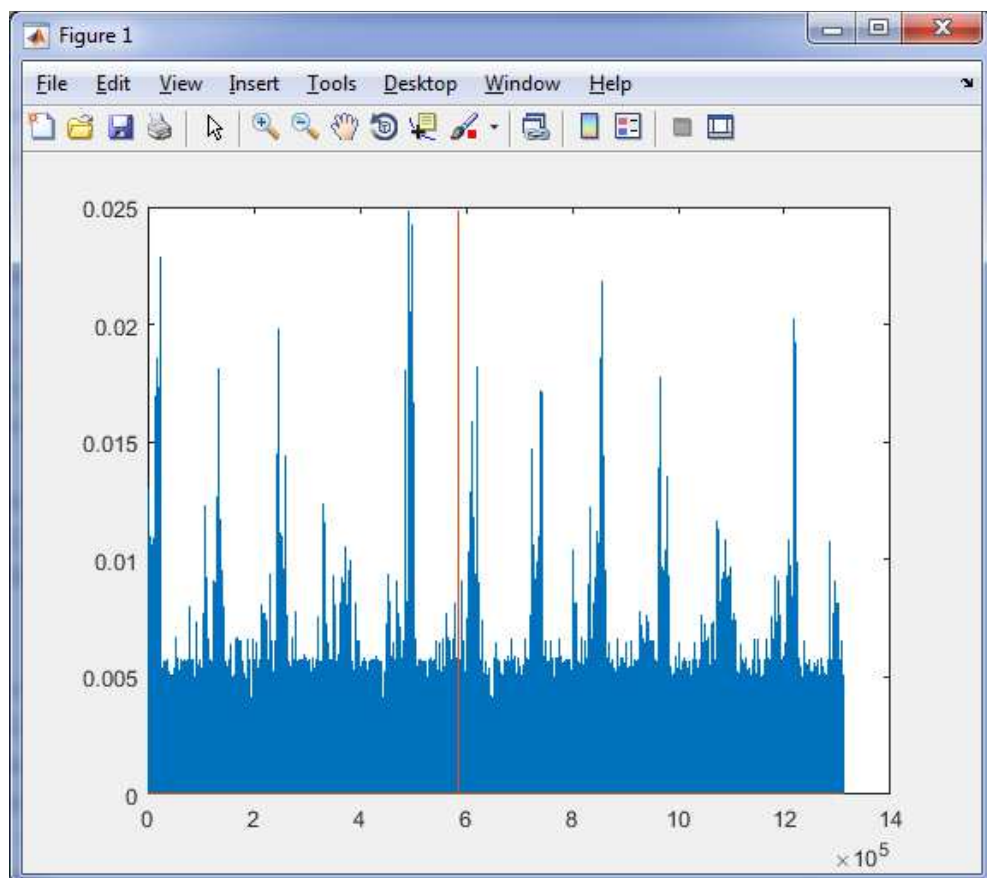


Рис. 2.12. Один з фреймів сеансу за 22.11.2017 року

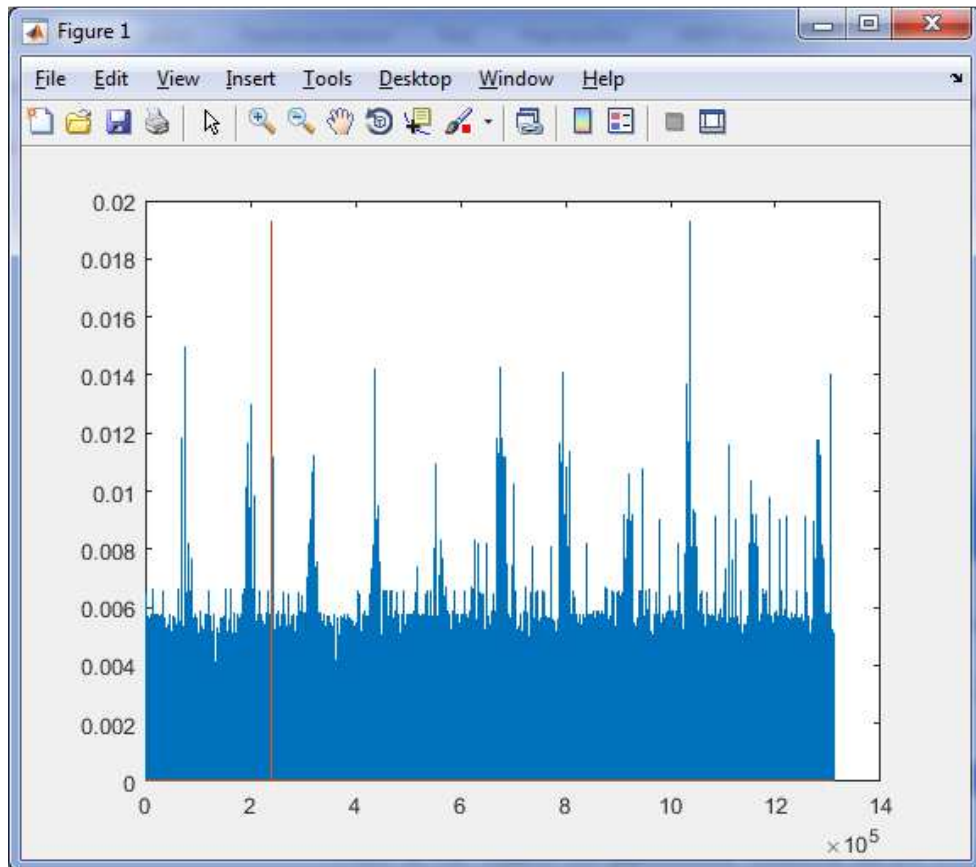


Рис. 2.13. Один з фреймів сеансу за 24.11.2017 року

На графіках кадрів червоною вертикальною лінією відзначено початок бортового сигналу, який було успішно декодовано (збігся код CRC). Як бачимо, наші сигнали за амплітудою не дуже примітні, рівень перешкод різко нерівномірний. Неможливо сказати, яка природа потужних сигналів, які ми спостерігаємо на графіках. Можливо, це теж сигнали ADS-B того ж борта, сигнал якого достовірно присутній, інших бортів або якихось інших систем.

Виявлення бортових сигналів, як ми бачимо з тексту програми, починається з виділення відліків, що перевищують середній завадовий фон на 14 дБ – обчислюється середній рівень амплітуди по всьому фрейму і вибираються відліки, амплітуда яких не менше ніж в 5 раз перевищує середнє значення (параметр **searchThr**). Безумовно, оптимальність обраного параметра **searchThr** потребує дослідження.

Для таких відліків формується зона можливої присутності сигналу з відступом на **samplesBeforeSyncPulse=60** відліків перед обраним відліком і на **samplesAfterMessage=120** відліків після очікуваного довгого повідомлення. Кількість відліків у розширеній зоні 1524 (довжина синхроімпульса 96 відліків, в довгому повідомленні 1248 відліків). Для перших 85 відліків цієї зони обчислюється стандартний вибіркового коефіцієнт кореляції з відомим синхроімпульсом. Випромінені і реально прийняті синхроімпульси виглядають так (рис. 2.13–2.15):

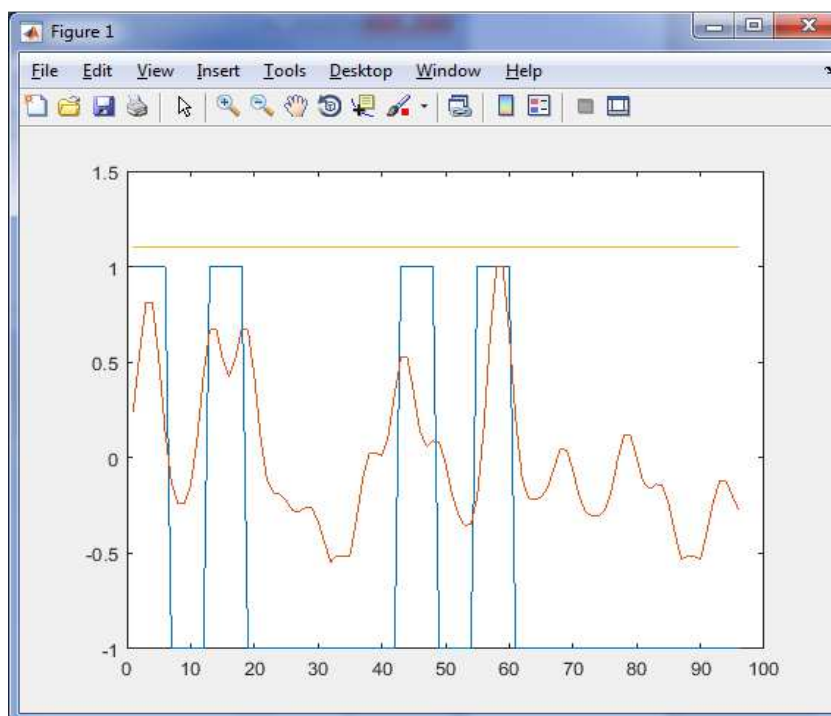


Рис. 2.14. Випромінені і реально прийняті синхроімпульси в сеансі за 24.11.2017 року (фрейм 1, коефіцієнт кореляції=0,64, повідомлення декодовано)

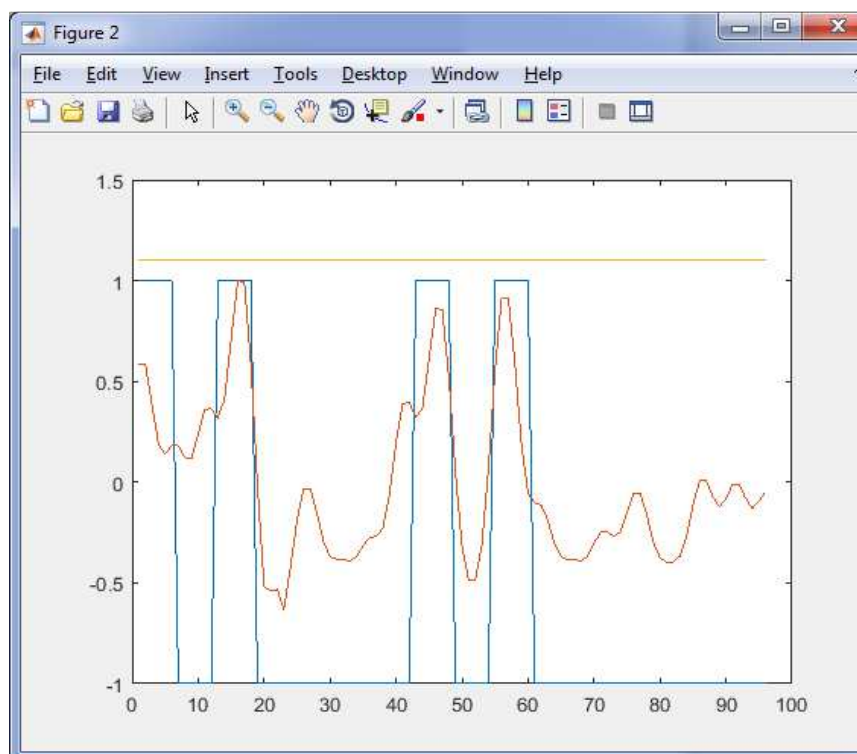


Рис. 2.15. Випромінені і реально прийняті синхроімпульси в сеансі за 24.11.2017 року (фрейм 2, коефіцієнт кореляції=0,67, повідомлення декодовано)

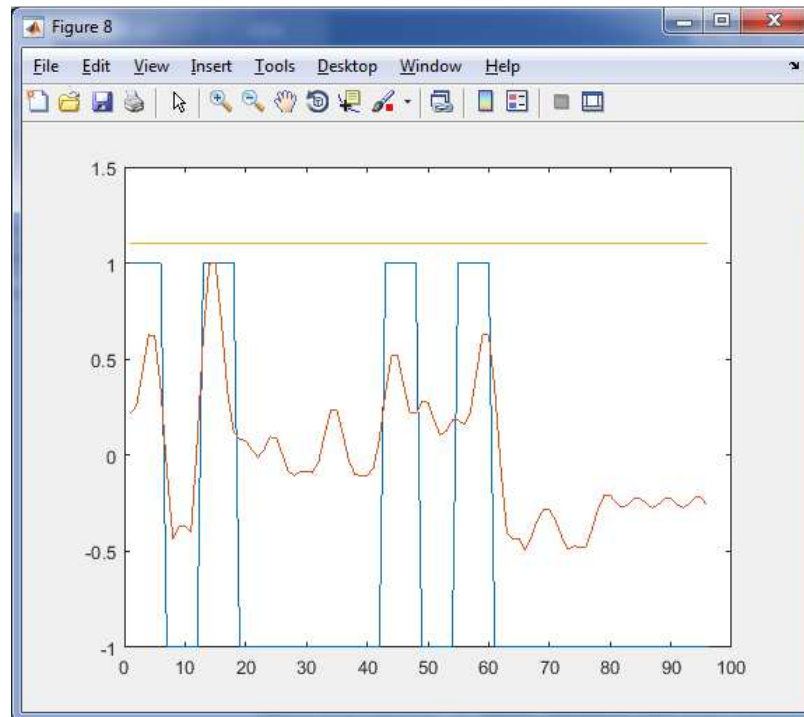


Рис. 2.16. Випромінені і реально прийняті синхроімпульси в сеансі за 24.11.2017 року (фрейм 3, коефіцієнт кореляції=0,68, збій по CRC)

Бачимо, що прийняті синхроімпульси досить сильно спотворені (зазвичай, коефіцієнт кореляції не перевищує 0,7). Чи є зображений на рис. 2.16 сигнал синхроімпульсом чи це випадковий збіг – відповісти важко. Зазначимо, що доцільність виявлення синхроімпульсу за коефіцієнтом кореляції не є очевидною. У багатьох системах замість обчислення коефіцієнта кореляції використовують згортку прийнятого сигналу з опорним імпульсом, і, як зазначалося вище, згортка проводиться окремо у кожному квадратурному каналі.

У розглянутому алгоритмі, якщо максимальний серед обчислених коефіцієнтів кореляції перевищує заданий поріг `corrThreshold=0.68`, то цей відлік приймається за початок синхроімпульсу, і проводиться виділення бітів на інтервалі довгого повідомлення. Формується «1», якщо сума перших 6 відліків перевершує суму наступних 6 відліків, в іншому випадку формується «0». Нульові та одиничні біти на прийнятій інформації виглядають так (рис. 2.17):

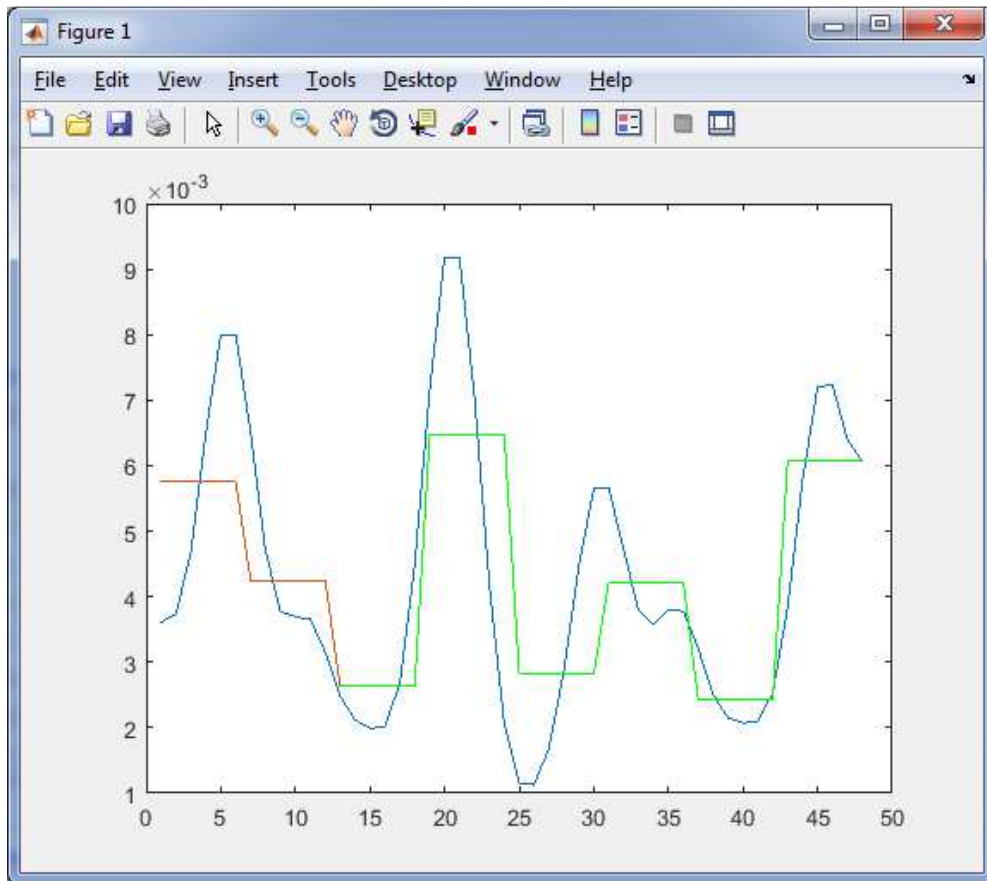


Рис. 2.17. Кодування перших чотирьох бітів одного з повідомлень у сеансі за 24.11.2017 року

На рис. 2.17 синім кольором зображені перші 48 відліків прийнятого бортового сигналу, що йдуть безпосередньо за синхроімпульсом. Ламана крива відповідає середнім значенням, обчисленим за 6 відліками. Колір лінії червоний, якщо середнє значення на перших 6 відліках більше середнього значення на наступних 6 відліках (передано «1»), і зелений, якщо навпаки (передано «0»). Бачимо, що передана комбінація бітів «1000», що відповідає цифрі «8». Всі повідомлення декодовано як «8D50824A90A572F3A95B3A**BC02DB**» з збігом коду CRC (останні 6 символів).

Ще раз зазначимо, що незалежно від довжини переданого повідомлення, воно декодується як довге. Повідомлення переводиться в 16-ковий вигляд, і перевіряється код CRC. Якщо він збігся, то гарантується прийняття довгого повідомлення, якщо ж не збігається, то прийняті символи обрізаються до 56 бітів (14 символів), і перевіряється гіпотеза про отримання короткого повідомлення новим обчисленням коду CRC. Для перевірки коду CRC використовується функція `crc24f()`, що реалізує 24-бітний CRC на базі універсальної функції `comm.CRCDetector()`.

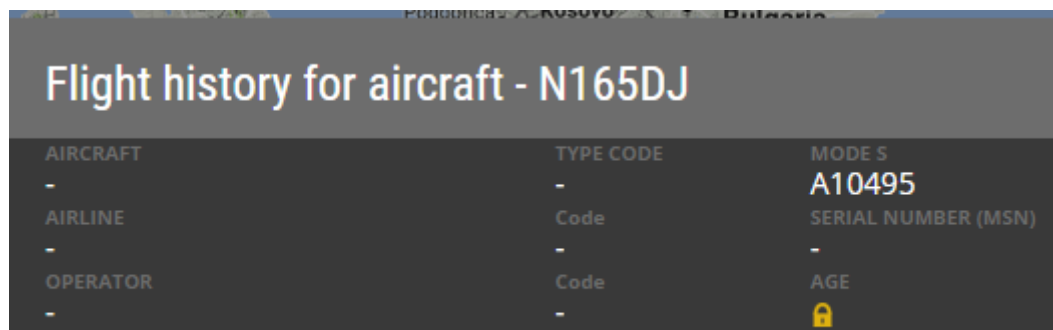
Підкреслимо основоположну роль коду CRC у забезпеченні працездатності розглянутої і всіх інших систем передачі інформації. Ймовірність випадкового збігу 24-бітного коду CRC, незалежно від довжини захищеного пові-

домлення, дорівнює $\frac{1}{2^{24}} \approx 6 \cdot 10^{-8}$, що практично гарантує безпомилкове декодування усіх повідомлень. Реально ця ймовірність ще нижче, оскільки для збігу коду CRC необхідна повна зміна переданої інформації (BER – bit error ratio близько 0,5). Зрозуміло, що реальні системи передачі інформації з таким BER не працюють. Один з авторів брав участь у створенні систем зв'язку, зокрема на базі КХ-радіостанцій [5], і навіть для 16-бітного CRC бути свідком збійного блоку інформації, для якого код CRC не детектував помилку, не довелося.

Однак розглянута система істотно відрізняється від систем безперервної передачі великих обсягів інформації. Код CRC доводиться рахувати і по завадовій ситуації, для якої коефіцієнт кореляції перевищив поріг, причому частка таких ситуацій велика. Наприклад, в сеансі за 12.01.2018 року частка обчислених коефіцієнтів кореляції, які перевищили поріг 0,68 (значення встановлено в штатній програмі), але код CRC детектував помилку, становить близько 75%. Тим самим можна стверджувати, що попереднє виявлення синхроімпульсу не суттєво знижує наведене значення вірогідності випадкового збігу коду CRC. Авторам здалося, що такий випадковий збіг коду CRC на реальній інформації системи ADS-B вдалося зафіксувати. За записом інформації сеансу за 24.11.2017 року була запущена програма, яка обчислює код CRC за відліками фрейма, що перевищує раніше заданий поріг, незалежно від кореляції з синхроімпульсом. В одному з фреймів було виявлено коротке повідомлення з правильним кодом CRC:

```
Time = 16:45:56
CRC OK = 1.000000
Decoded Message = BAA10495
AircraftID = A10495
```

Як вже зазначалося у вступі до розділу, в режимі Mode-S короткі повідомлення зобов'язані починатися з 5D або 5F. Перевірка бортового номера на www.flightradar24.com не надто прояснила ситуацію:



Sorry, but we could not find data for specified flight

Рис. 2.18. Відповідь від flightradar24 на пошук бортового номера у повідомленні, яке не відповідає режиму Mode-S

Вірогідність наявності випадкового бортового номера в БД реальних літаків становить 0,03 (на flightradar24 заявлено, що в БД близько 500000 літаків, всього номерів 16^6). Ця ймовірність не надто велика, але і не така, щоб дивуватися, якщо подія відбудеться. Все вирішилося трохи пізніше, коли було виявлено, що повідомлень з цим бортовим номером на невеликому інтервалі часу 19 штук (надалі багато таких повідомлень було виявлено у всіх записах, початок повідомлення «ВА» або «ВЕ»). Зрозуміло, що це повністю виключає випадковий збіг коду CRC. Швидше за все, фіксуються сигнали літаків малої або військової авіації.

2.6. Огляд експериментів по прийому бортових сигналів

Радіус прийому сигналів повітряних суден на маленьку 15 см штиркову антену (рис. 1), як зазначалося вище, близько 100 км. В зв'язку з помітним загасанням гігагерцових хвиль у приміщенні, реальна зона прийому з боку стін зменшується. Частина експериментів проводилася при підключенні приймача RTL-SDR до домашнього ПК автора з вікном на захід, частина — при підключенні до сервера в корпусі Інституту автоматики і електротехніки з вікнами на схід. У першому випадку більше повідомлень приймалося від літаків, що пролітали між Миколаєвом і Одесою, у другому випадку — між Миколаєвом і Херсоном. Щодооби в зоні контролю виявляються 2-3 десятки літаків. Для прикладу в таблиці 2.7 наведені дані за сеанс спостереження 12-13 січня 2018 року (дислокація приймача в навчальному корпусі) (таблиця 2.7):

Таблиця 2.7

№ з/п	№ борта (AircraftID)	Кількість пові- домлень (рейс прямий/ зво- ротний)	Кількість спосте- режень	Номер рейсу (FlightID)	Перше по- відомлен- ня	Останнє повідом- лення	Час спосте- режень	Швид- кість (км/год.)	Шлях за час спостере- ження (км)
1	740738	447	1	RJA177	12:43:25	12:51:19	0:07:54	817	108
2	508340	161	1	ANR5117	12:59:17	13:02:48	0:03:31	765	45
3	50818D	84	2		13:51:35	13:56:00	0:04:25		
4		181			16:58:29	17:07:12	0:08:43		
5	4B9065	523	1	PGT427	17:32:34	17:38:53	0:06:19	800	84
6	5081EE	7	1	AUI780	17:44:47	17:49:14	0:04:27		
7	4B84E9	709	2	UJX251	15:50:28	15:59:36	0:09:08	835	127
8		561		UJX250	18:26:52	18:31:48	0:04:56	835	69
9	50839B	7	1		16:29:15	16:32:40	0:03:25	793	45
10	51407D	1	1		16:46:09	16:46:09	0:00:00	837	0
11	8990BF	9	1		18:37:55	18:40:48	0:02:53	883	42
12	4BAA7A	443	2	THY1475	18:58:58	19:04:10	0:05:12	817	71
13		125		THY1476	21:38:52	21:44:03	0:05:11	817	71
14	50816F	170	2		21:25:24	21:30:08	0:04:44		66
15		116			7:11:35	7:22:03	0:10:28		135
16	5082EA	260	2		22:14:59	22:18:19	0:03:20	837	47
17		132			3:41:38	3:47:25	0:05:47	837	81
18	4BB846	776	2	PGT432	0:00:49	0:09:02	0:08:13	828	113

№ з/п	№ борта (AircraftID)	Кількість пові- домлень (рейс прямий/ зво- ротний)	Кількість спосте- режень	Номер рейсу (FlightID)	Перше по- відомлен- ня	Останнє повідом- лення	Час спосте- режень	Швид- кість (км/год.)	Шлях за час спостере- ження (км)
19		616		PGT433	2:53:39	3:02:06	0:08:27	828	117
20	4B906C	820	1	PGT429	2:33:15	2:41:06	0:07:51	789	103
21	508371	10	1		6:27:33	6:30:37	0:03:04	800	41
22	508379	11	1		6:57:22	7:00:26	0:03:04	811	41
23	508279	1	1		7:23:30	7:23:30	0:00:00		
24	5083C9	18	1	AUI722	7:24:19	7:28:05	0:03:46	785	49
25	508366	3	1		7:39:32	7:40:04	0:00:32	806	7
26	508395	27	1		7:48:50	7:50:26	0:01:36	798	21
	Всього	6218	26						

В таблиці дані наведено тільки у відповідності з отриманими повідомленнями, тому номери рейсів вказані не для всіх літаків. Вони можуть бути отримані через сайт www.flightradar24.com за бортовим номером і часом польоту. Те ж саме стосується швидкості польоту. Від частини літаків отримані тільки короткі повідомлення. Зокрема, це стосується бортів 50818D і 50816F, які є представниками літаків Embraer ERJ 145 — сімейства регіональних реактивних літаків виробництва Embraer, Бразилія. Пасажирів всього 50, максимальна швидкість 834 км/год. Мабуть, їх бортова апаратура влаштована простіше, і такі параметри, як географічні координати, швидкість руху, напрямок польоту і інші параметри або не вимірюються, або не включаються в повідомлення ADS-B.

Усього за неповну добу спостереження отримано **6218** повідомлень. Найбільш тривалий інтервал спостереження зафіксовано для борта 50816F, що виконував рейс Київ-Херсон (4 хвилини 44 секунди) і Херсон-Київ (10 хвилин 28 секунд). Дані за цими рейсами наведені на рис. 2.19, 2.20:

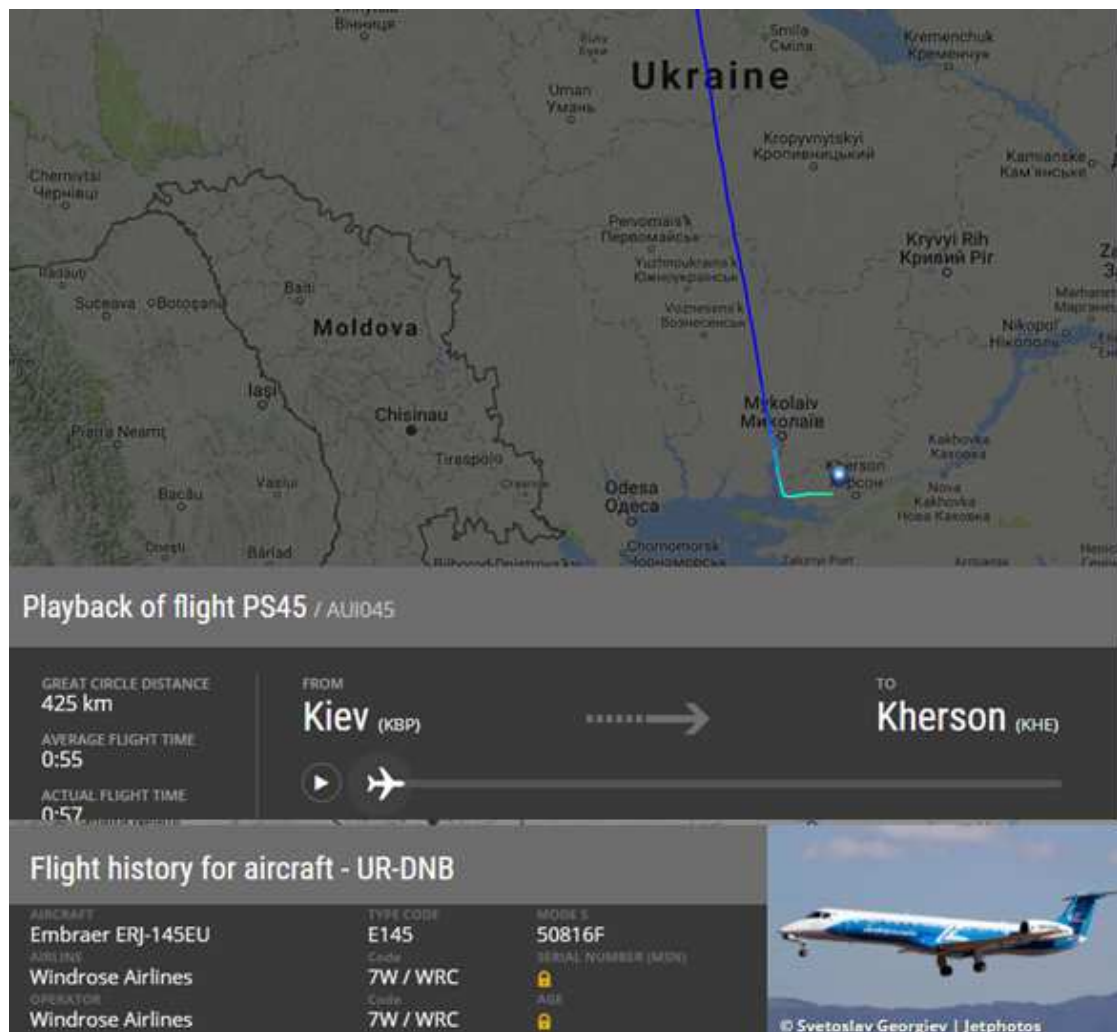


Рис. 2.19. Дані flightradar24 про рейс Київ-Херсон (борт 50816F)

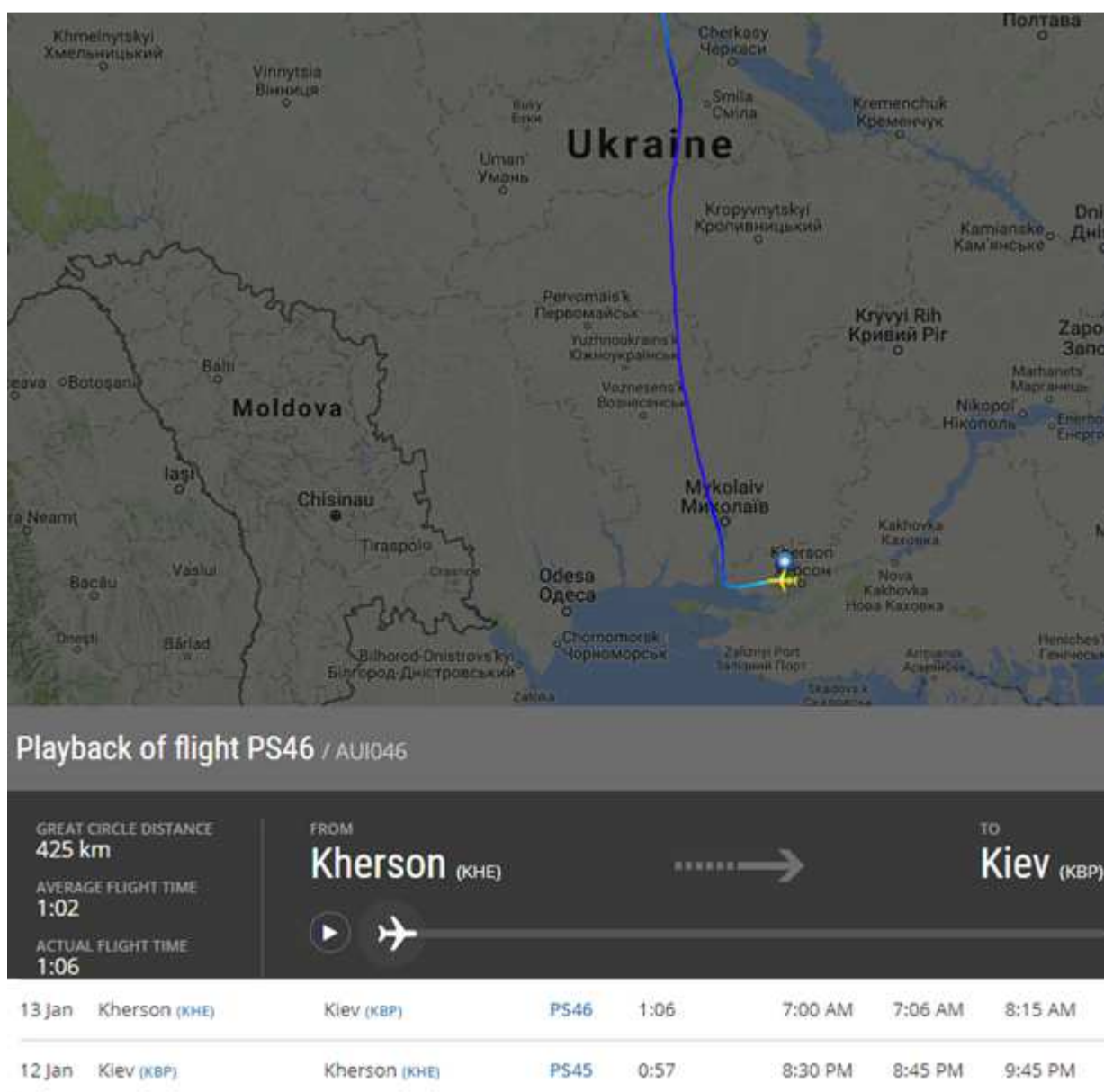


Рис. 2.20. Дані flightradar24 про рейс Херсон- Київ (борт 50816F)

Рекордна тривалість спостереження, мабуть, пов'язана з обльотом Миколаєва з півдня і заходу. Взявши швидкість польоту 775 км/год. (реальна швидкість невідома, максимальна швидкість 834 км/год.), отримуємо, що шлях за час спостереження склав близько **135 км**.

Найбільшу кількість повідомлень (**820**) отримано від борту 4V906С, що виконував рейс PGT429 (PC429) Харків-Стамбул. Дані по ньому наведено на рисунку 2.21:

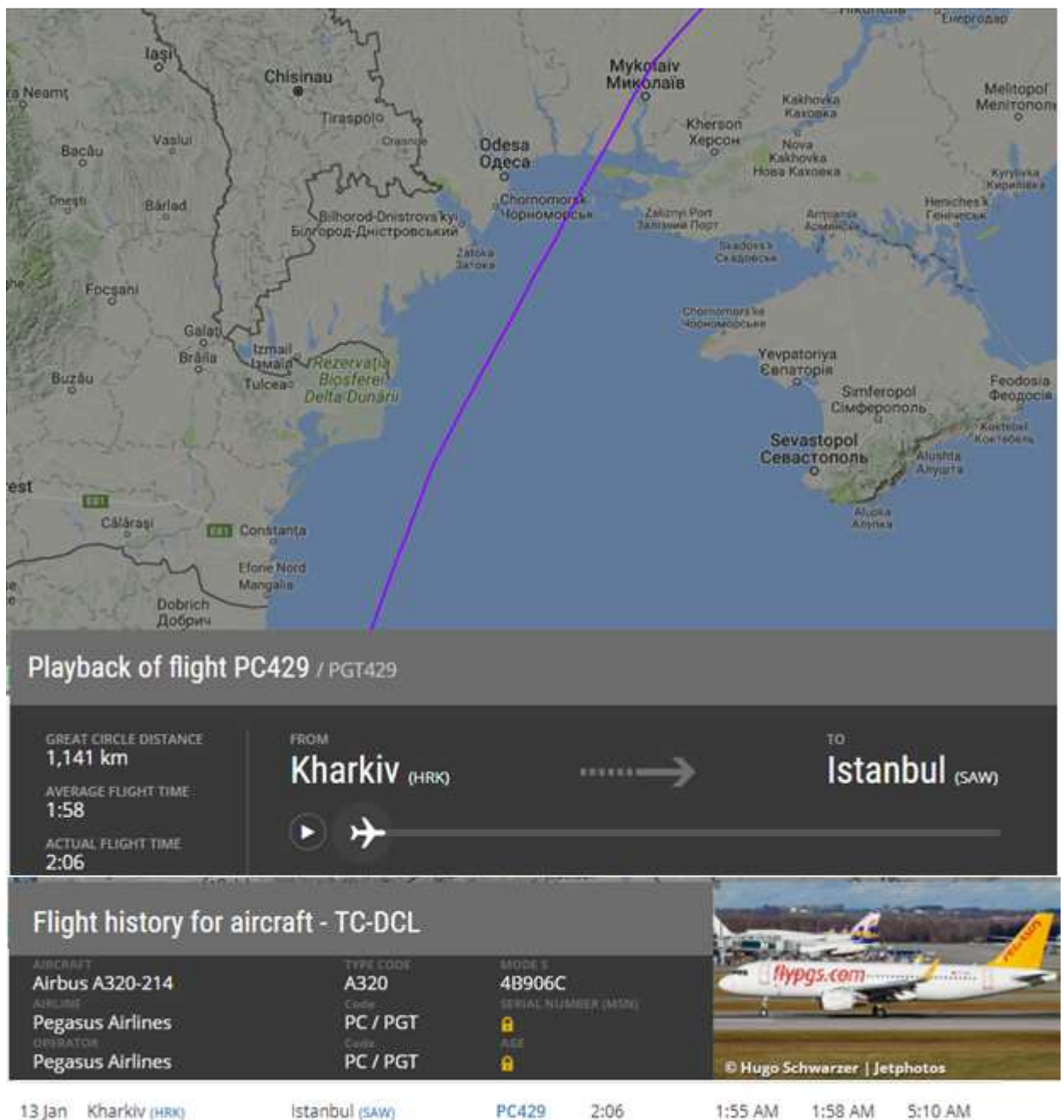


Рис. 2.21. Дані flightradar24 про рейс PGT429 (PC429) Харків-Стамбул (борт 4B906C)

Як бачимо, літак дійсно пролетів дуже близько від Миколаєва.

Як зазначалося вище, найважливішим параметром програми є поріг по коефіцієнту кореляції **corrThreshold**, оптимальне значення якого необхідно підібрати експериментально. Дійсно виявилось, що значення 0,68, встановлене авторами програми, не є оптимальним. Всі наведені вище дані в реальному часі отримані при порозі **0,60** (отримано 6218 повідомлень). Відтворення запису амплітуд з порогом 0,68 дозволило отримати тільки 4767 повідомлень, тобто 1455 повідомлень були втрачені, що становить 23,4%. Таким чином, для значної кількості повідомлень з коректним кодом CRC коефіцієнт коре-

ляції з опорним синхроімпульсом знаходиться в межах 0,60-0,68. Подальше зниження порогу по коефіцієнту кореляції до значення 0,55 дає незначний приріст кількості прийнятих коректних повідомлень. Так під час прийому бортових сигналів за 17-18.01.2018 року (з 20:00:32 до 10:08:02) зі значенням `corrThreshold = 0.55` було отримано **6616** повідомлень. Обробка запису зі значенням `corrThreshold = 0.60` дозволило виділити **6508** повідомлень. Таким чином, вказане зниження порогу призвело до збільшення кількості повідомлень на 108, або на 1,6%. Зрозуміло, що зниження розглянутого порогу в середньому збільшує обсяг обчислень. Для наведеної вище переписаної програми отримані наступні оцінки часу роботи. При порозі 0,68 середній час обробки одного фрейму, при явній присутності хоча б одного сигналу ADS-B, займає приблизно 2,5 с (на ПК середньої потужності), зменшення порогу до 0,60 збільшує час рахунку до 3,5 с. Звідси передусім випливає, що в реальному часі будуть оброблятися далеко не всі фрейми. Дещо несподівано з'ясувалося, що штатна програма обчислює швидше. Обробка одного фрейму в реальному часу в середньому займає близько 0,2 с (поріг 0,60). У цій же ситуації доопрацьована програма на фрейм у середньому витрачає близько 2 с. При цьому кількість виділених повідомлень з однієї і тієї ж інформації приблизно однаково (за записом сеансу 12-13.01.2018 року доопрацьованою програмою до 6218 виявленим в реальному часі повідомленнями додано лише 4 нових). Таким чином, вимагають дослідження 2 завдання. Знайти причини, по яким доопрацьована програма працює на порядок повільніше, і вивчити можливість скорочення втрат фреймів в реальному часі штатної програми (зараз, поки обробляється один фрейм, інформація на довжині двох фреймів пропускається).

2.7. Завдання для лабораторних, курсових робіт і дипломних проектів

Варіант 1.

В MATLAB або на мові C++ написати програму, що визначає за повідомленням у 16-ковому вигляді номер рейсу. Програма має брати дані з текстового файлу протоколу повідомлень штатної і доопрацьованої програм.

Варіант 2.

В MATLAB або на мові C++ написати програму, що визначає за повідомленням у 16-ковому вигляді його парність і непарність. Програма має брати дані з текстового файлу протоколу повідомлень штатної і доопрацьованої програм.

Варіант 3.

В MATLAB або на мові C++ написати програму, що визначає по парному і непарному повідомленням в 16-ковому вигляді поточні **глобальні** географічні координати і висоту польоту судна. Програма має брати дані з текстового файлу протоколу повідомлень штатної і доопрацьованої програм.

Варіант 4.

В MATLAB або на мові C++ написати програму, що визначає за повідомленням у 16-ковому вигляді поточні **локальні** географічні координати і висоту польоту, припускаючи, що відома опорна точка, в околі якої завідомо знаходиться судно. Програма має брати дані з текстового файлу протоколу повідомлень штатної і доопрацьованої програм.

Варіант 5.

В MATLAB або на мові C++ написати програму, що визначає за повідомленням у 16-ковому вигляді поточну швидкість судна (щодо Землі або повітряну). Програма має брати дані з текстового файлу протоколу повідомлень штатної і доопрацьованої програм.

Варіант 6.

В MATLAB або на мові C++ написати програму, що визначає за повідомленням у 16-ковому вигляді, який містить дані про місцезнаходження судна, точність вимірювання. Програма має брати дані з текстового файлу протоколу повідомлень штатної і доопрацьованої програм.

Варіант 7.

Дослідити ефективність лінійної інтерполяції та інтерполяції невідомих значень найближчим виміряним значенням сигналу.

Варіант 8.

Провести доопрацювання програми запису інформації з приймача RTL-SDR і програми читання цих файлів, що забезпечують прив'язку фреймів до поточного реального часу.

Варіант 9.

В MATLAB або на мові C++ написати програму, що створює з текстового файлу протоколу повідомлень штатної програми інтегральну таблицю аналізу повідомлень за сеансом спостережень повітряних суден (за прикладом таблиці 2.7).

Варіант 10.

Провести дослідження ефективності пошуку синхроімпульсу в повідомленнях ADS-B за квадратурними каналами прийому інформації.

Варіант 11.

Знайти причини, по яким доопрацьована програма працює на порядок повільніше штатної програми, і усунути їх.

Варіант 12.

Дослідити можливість скорочення втрат фреймів у реальному часі штатної програми.

Варіант 13.

Провести доопрацювання програми для прийому нестандартних повідомлень (режим ADS-B, відмінний від Mode-S), що починаються з символів «BA» або «BE» (імовірно повідомлення літаків малої або військової авіації).

3. ВІДСТЕЖЕННЯ РУХУ МОРСЬКИХ СУДЕН

Технологія для відстеження руху морських суден отримала назву AIS (Automatic Identification System – автоматична ідентифікаційна система морських суден). В ній для передачі інформації вибрані сигнали з гаусовою частотною модуляцією з мінімальним зсувом (GMSK – Gaussian Minimum Shift Keying), які також широко використовується в інших системах зв'язку.

- 3.1. Загальні відомості про GMSK сигнали **Ошибка! Закладка не определена.**
- 3.2. Дослідження сигналів GMSK в Matlab. **Ошибка! Закладка не определена.**
- 3.3. Загальна інформація про методи передачі повідомлень морськими судами **Ошибка! Закладка не определена.**
- 3.4. Дослідження реальних сигналів AIS **Ошибка! Закладка не определена.**
 - 3.4.1. Алгоритм виявлення сигналів AIS **Ошибка! Закладка не определена.**
 - 3.4.2. Дослідження записів сигналів AIS від судів в околиці м Бостона..... **Ошибка! Закладка не определена.**
 - 3.4.3. Дослідження сигналів судів в акваторії Бузького лиману..... **Ошибка! Закладка не определена.**
- 3.5. Завдання для лабораторних, курсових робіт і дипломних проектів.....160

3.1. Загальні відомості про GMSK сигнали

Насамперед, відзначимо, що крім технології AIS, модуляція GMSK використовується в сучасних системах цифрового радіозв'язку GSM (Global System for Mobile Communications), GPRS (General Packet Radio Service) та інших і забезпечує високу якість передачі у відносно вузькій смузі, що займає сигнал.

GMSK-сигнали – це сигнали мінімальної частотної маніпуляції з гаусовою огинаючою. Тому спочатку коротко розглянемо сигнали мінімальної частотної маніпуляції (MSK).

Основним параметром при частотній маніпуляції є індекс частотної маніпуляції m , який визначає девіацію частоти маніпуляції:

$$\omega_d = \pi \cdot Br \cdot m,$$

де Br – швидкість передачі інформації. MSK-модуляція повинна забезпечити ортогональність сигналів, що передають «0» і «1» цифрової інформації при мінімально можливому індексі m . Іншими словами, MSK-сигнал – сигнал з мінімальним розносом частот «0» і «1» (тобто з мінімальною девіацією), при якому ці частоти можна розрізнити на інтервалі T передачі одного символу. З умови ортогональності можна вивести, що мінімальне допустиме значення $m = 0,5$.

Схема MSK-модулятора виглядає наступним чином (рис. 3.1):

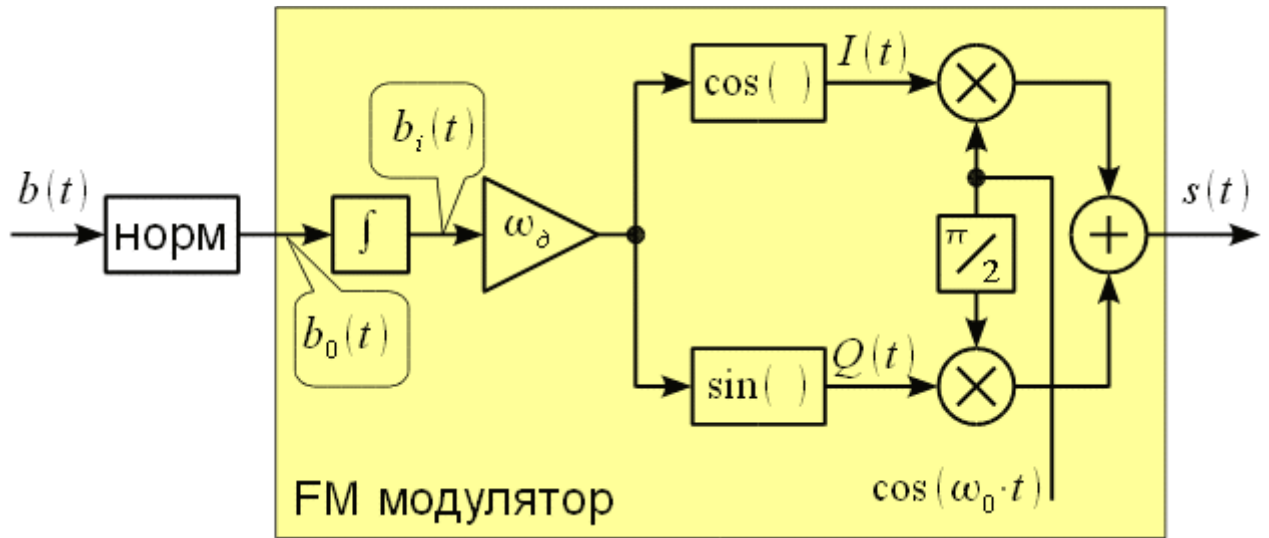


Рис. 3.1. Структурна схема формування MSK на основі FM-модулятора

Повна фазова діаграма представлена на рисунку 3.2:

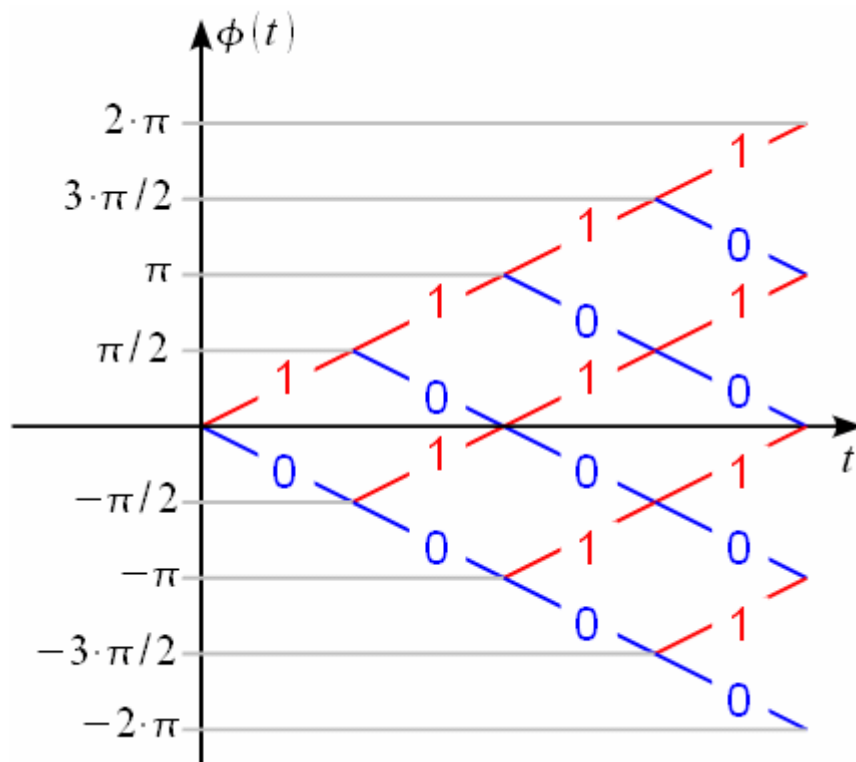


Рис. 3.2. Повна фазова діаграма при MSK для 4-х біт інформації

Таким чином, передача «1» призводить до збільшення фази сигналу на $\frac{\pi}{2}$, а передача «0» — до її зменшення на $\frac{\pi}{2}$.

Сигнали з мінімальною частотною маніпуляцією (MSK) забезпечують мінімальну ширину головної пелюстки спектру, рівну $1,5Br$. Однак постійне

ущільнення радіочастотного спектру призвело до необхідності додаткового зменшення рівня бічних пелюсток спектра сигналу. Цього можна досягти згладжуванням фронтів імпульсів модулюючого цифрового сигналу. На рис. 3.3 показано фазову діаграму MSK без згладжування і зі згладжуванням фронтів вхідного цифрового сигналу. Згладженому сигналу $b_g(t)$ відповідає фазова діаграма $\Phi_g(t)$, яка не має точок перелому.

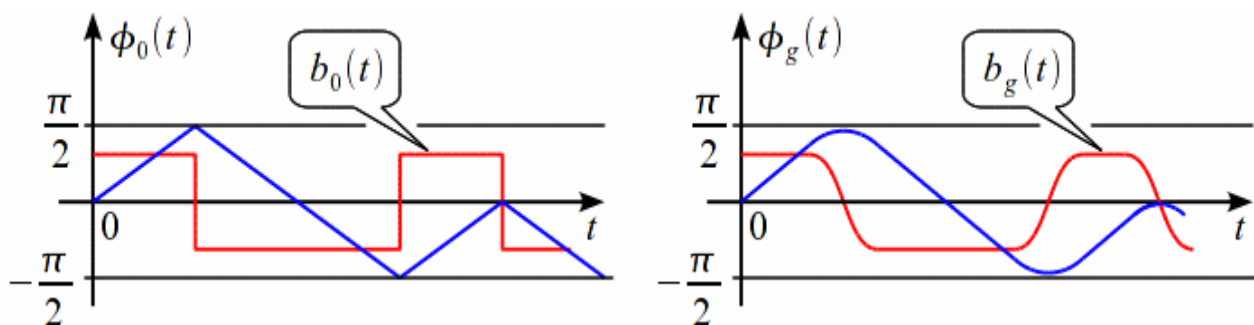


Рис. 3.3. Фазова діаграма MSK без згладжування і зі згладжуванням фронтів вхідного цифрового сигналу

Для GMSK-сигналів згладжування вхідного цифрового сигналу здійснюється гаусовим фільтром [6]. ФНЧ (фільтр нижніх частот) Гауса задається імпульсною характеристикою виду:

$$g(t) = \frac{BT}{T} \sqrt{\frac{2\pi}{\ln 2}} \cdot \exp\left(-\frac{2 \cdot \pi^2 \cdot BT^2}{\ln 2 \cdot T^2} \cdot t^2\right),$$

де BT – безрозмірна величина, яка дорівнює $B_{-3\text{дБ}} \cdot T$; $B_{-3\text{дБ}}$ – смуга фільтра Гауса за рівнем -3дБ , T як і раніше, – тривалість передачі одного символу. Таким чином, параметр BT показує, у скільки разів смуга фільтра Гауса $B_{-3\text{дБ}}$ відрізняється від швидкості передачі інформації Br , вираженої в одиницях вимірювання частоти. Легко показати, що

$$\int_{-\infty}^{\infty} g(t) dt = 1.$$

Тим самим, ФНЧ Гауса на нульовій частоті має коефіцієнт передачі, що дорівнює 1 для будь-яких BT . На рис. 3.4 показано імпульсні характеристики $g(t_n)$ фільтра Гауса при $T = 1$ с і різних параметрах BT .

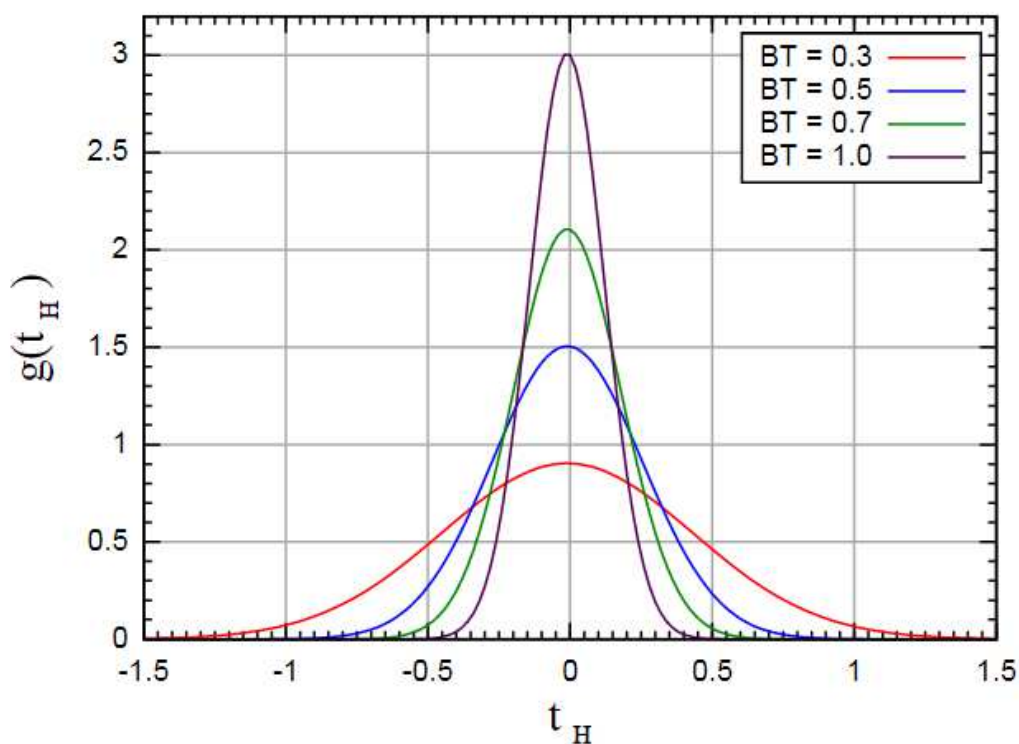


Рис.3.4. Імпульсні характеристики фільтра Гауса при $T=1$ с і різних BT

На рис. 3.5 наведено нормовану АЧХ фільтра Гауса з нормуванням частоти $f_H = f \cdot T$. Тим самим нормована частота $f_H = 1$ відповідає частоті $Br = 1/T$ Гц.

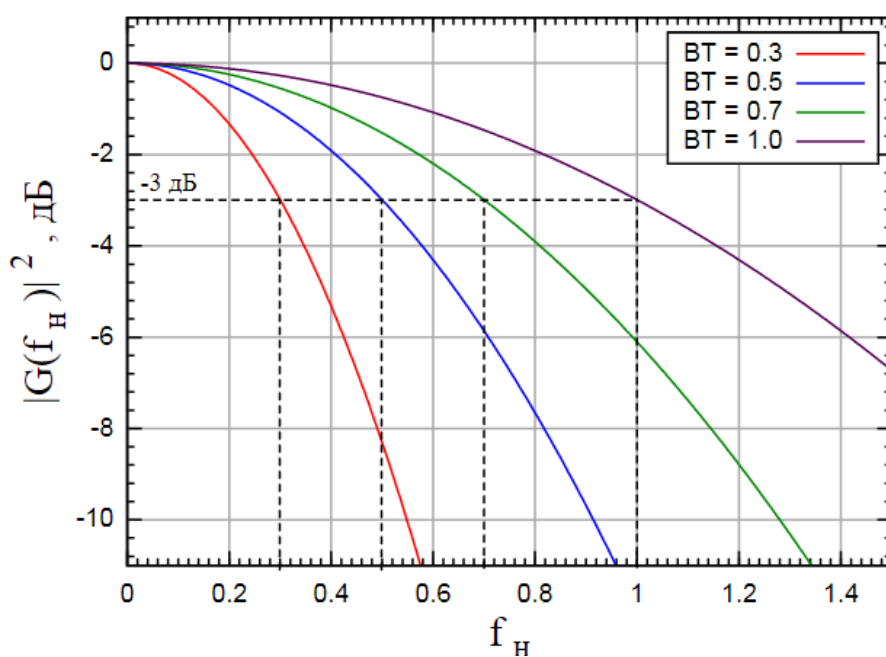


Рис.3.5. Нормована АЧХ фільтра Гауса при різних параметрах BT

З рисунку добре видно (позначено пунктирними лініями), що нормована смуга фільтра Гауса за рівнем -3дБ дорівнює BT .

Розглянемо структурну схему GMSK-модулятора, показану на рисунку 3.6:

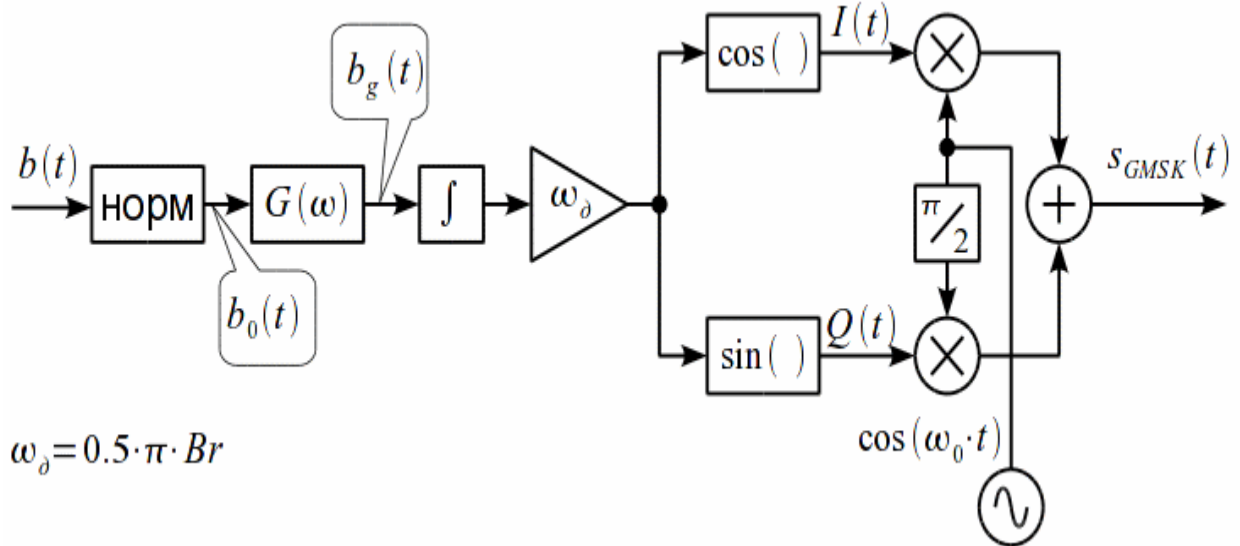


Рис. 3.6. Структурна схема модулятора GMSK

Вхідний цифровий сигнал $b(t)$ нормується по амплітуді і виходить сигнал $b_0(t)$ з нульовим середнім і не перевищує за модулем 1. Після $b_0(t)$ подається на згладжуючий фільтр Гауса $G(\omega)$, на виході якого маємо згладжений сигнал $b_g(t)$. Цей згладжений сигнал буде модулюючим сигналом частотного модулятора. Частота девіації при модуляції відповідає частоті девіації MSK $\omega_d = 0,5 \cdot \pi \cdot Br$ рад/с. В результаті отримаємо сигнал $S_{GMSK}(t)$ на несучій частоті ω_0 .

Негативною стороною застосування Гаусового фільтра є межсимвольна інтерференція, яка тим більше, чим менше BT . На рисунках 3.7–3.10 наведено осцилограми згладженого сигналу $b_g(t)$ при різних значеннях BT і швидкості передачі даних $Br = 20$ кбіт/с.

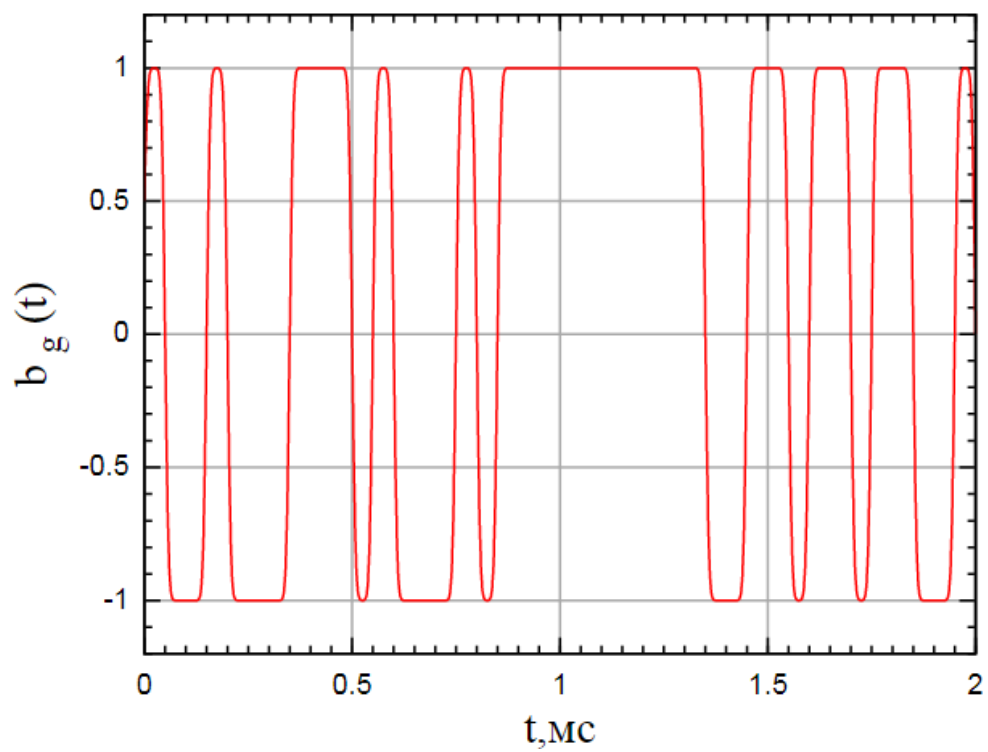


Рис. 3.7. Сигнал на виході фільтра Гауса при $BT = 1$

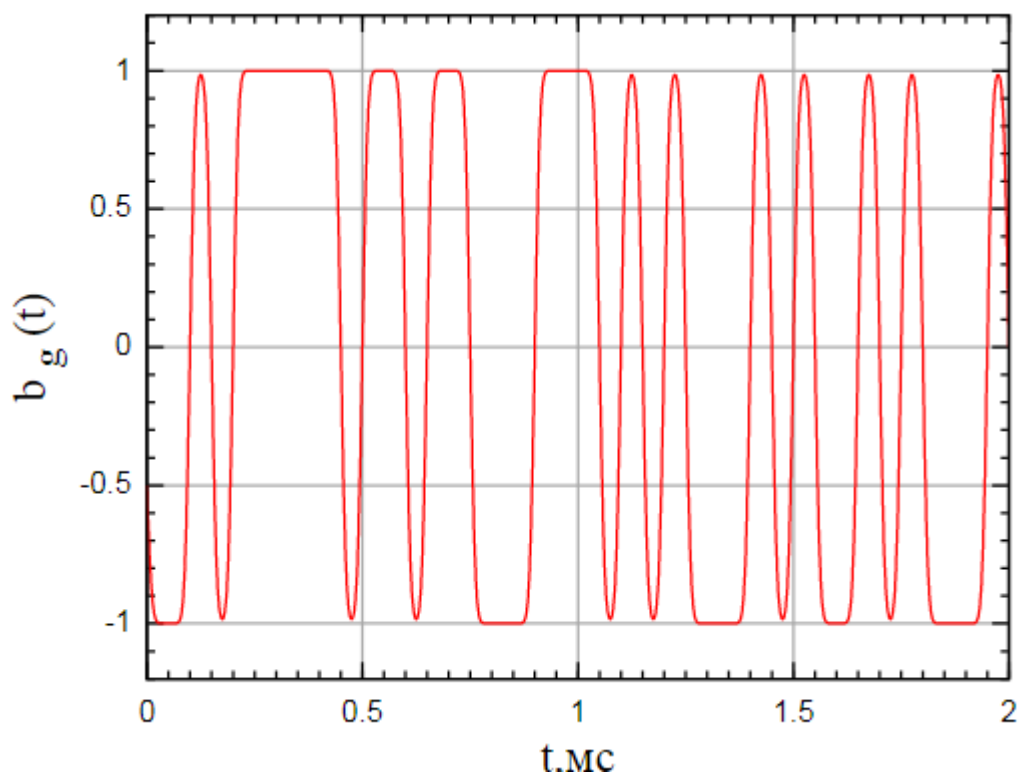


Рис. 3.8. Сигнал на виході фільтра Гауса при $BT = 0,7$

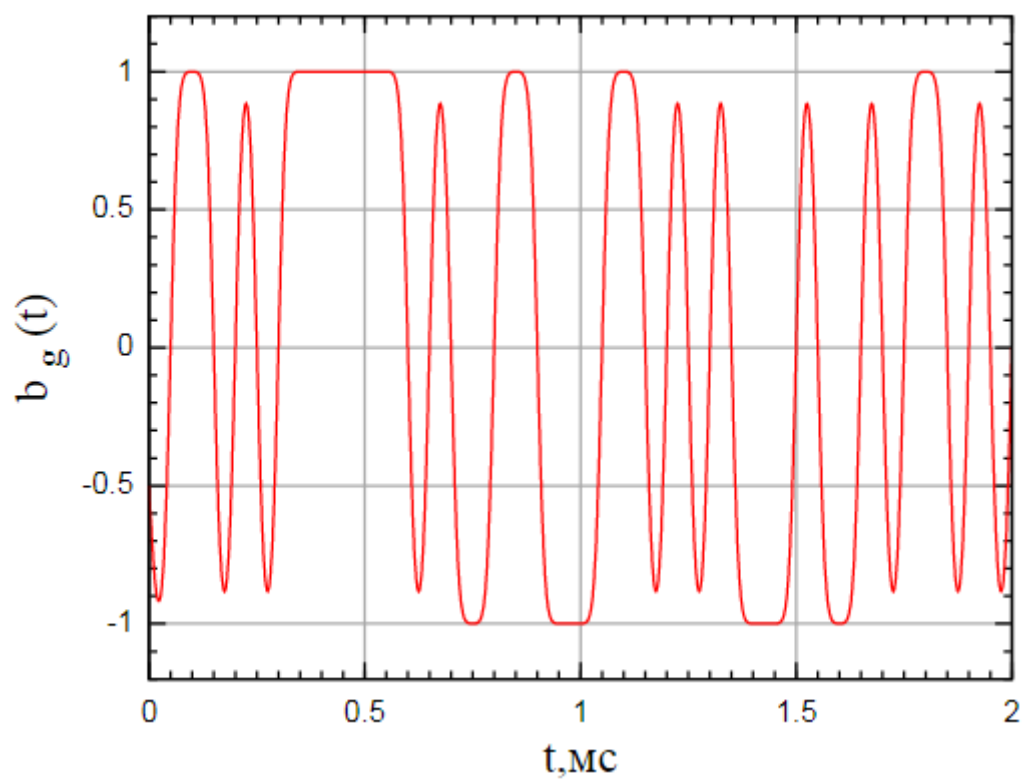


Рис. 3.9. Сигнал на виході фільтра Гауса при $BT = 0,5$

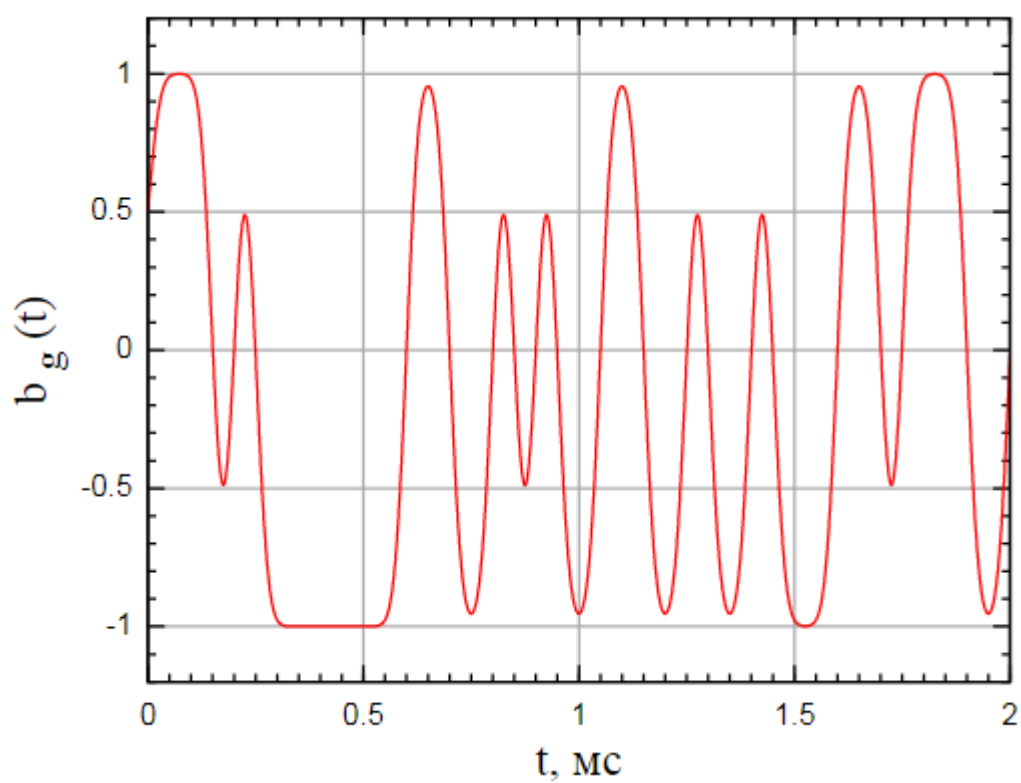


Рис. 3.10. Сигнал на виході фільтра Гауса при $BT = 0,3$

З рис. 3.7 – 3.10 добре видно, що зменшення BT приводить до розширення ефективної тривалості імпульсної характеристики фільтра і збільшення міжсимвольної інтерференції.

На рис. 3.11 показаний спектр MSK-сигналу і GMSK-сигналів при різних BT , при тій же швидкості передачі, що і вище.

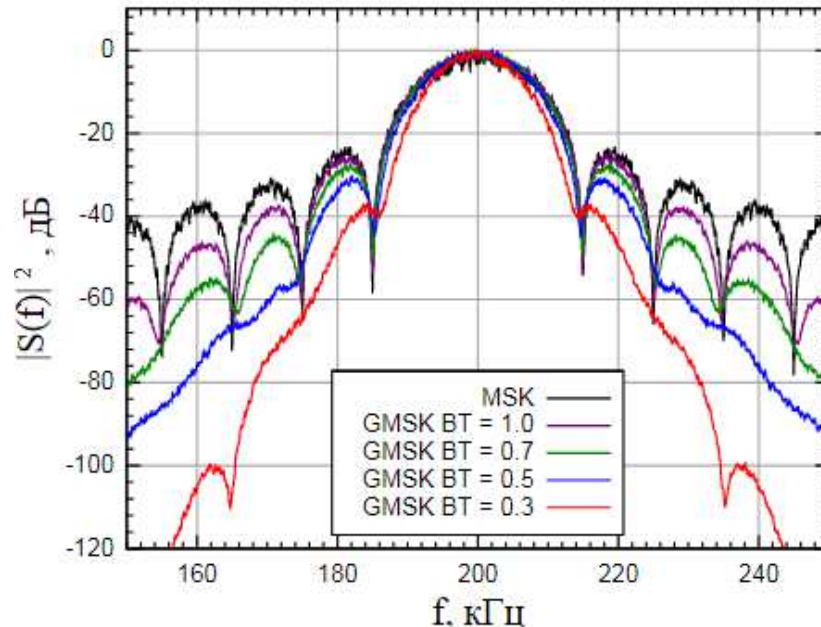


Рис. 3.11. Порівняння спектрів MSK- і GMSK-сигналів при різних BT

З рис. 3.11 добре видно, що в спектрі GMSK при зменшенні BT зменшуються рівні бічних пелюсток, крім того, значно зростає швидкість спадання спектру. Так максимальний рівень бічного пелюстка GMSK при $BT=0,3$ на 15 дБ менше, ніж у MSK-сигналу, а швидкість спадання лінійно залежить від частоти, що обумовлено застосуванням Гаусового фільтра.

3.2. Дослідження сигналів GMSK в MATLAB.

У MATLAB сигнали GMSK генерує comm-функція `comm.GMSKModulator()`. Функція має такі параметри:

- **BitInput** — логічний параметр, що визначає представлення біт у формі 0,1 (**true**) або +1, -1, за замовчуванням **BitInput=false**;
- **BandwidthTimeProduct** — параметр BT , введений в розділі 1 (добуток ширини фільтра Гауса на тривалість передачі одного символу);
- **PulseLength** — довжина Гаусового імпульсу (фільтра) у кількостях інтервалів передачі одного символу. Значення за замовчуванням дорівнює 4;

- **SymbolPrehistory** — передісторія бітів в представленні $+1, -1$ не може перевищувати по довжині **PulseLength-1**, за замовчуванням один біт який дорівнює 1;
- **InitialPhaseOffset** — початковий зсув по фазі в радіанах;
- **SamplesPerSymbol** — кількість формованих відліків на довжині одного символу;
- **OutputDataType** — тип комплексних сигнальних відліків, що повертаються, за замовчуванням використовується тип **double**.

Для демодуляції GMSK-сигналів служить comm-функція **comm.GMSKDemodulator()**. У неї такі ж аргументи, як і у модулюючої функції, з додаванням параметра:

- **TracebackDepth** — глибина повернення назад в алгоритмі динамічного програмування Вітербі.

Як і в більшості comm-функцій, виклик зазначених функцій тільки створює об'єкт відповідного класу. Для реальних операцій служать методи класу, зокрема метод **Y=step(N,X)** у першій функції виконує модулювання. Тут **N** — ім'я створеного об'єкта, **X** — вхідний інформаційний бітовий масив для передачі, обов'язково повинен бути вектором стовпцем (у багатьох comm-функціях MATLAB вхідна інформація з кількома стовпцями сприймається як багатоканальна, у розглянутих функціях багатоканальність не підтримується), **Y** — сформований для передачі сигнал GMSK. У другій функції **X** — прийнятий комплексний GMSK-сигнал, **Y** — декодований потік біт.

Основним недоліком реалізованої моделі GMSK-сигналу є відсутність узгодженості по несучій частоті між передавачем і приймачем, що в реальних системах з використанням GMSK-сигналу завжди присутня. Відсутність блоку оцінки зсуву по частоті та її корекції призводить до повної непрацездатності каналу зв'язку, в чому ми переконаємося при дослідженні реальних сигналів AIS.

Проведемо порівняння GMSK-сигналу з сигналом з подвійною фазовою маніпуляцією (BPSK) для забезпечення однакової максимальної швидкості передачі інформації. Методику порівняння і структуру програми запозичуємо з [5]. Будемо використовувати параметри сигналу GMSK, прийняті в технології AIS для морських суден:

```
% Програма для дослідження характеристик сигналів GMSK
close all;

% Параметри відповідають прийнятими в AIS
Vt=9600;% Швидкість передачі біт/с
```

```

dt=1/Vt; % Тривалість символу в с (104,17 мкс)
dtf=dt/SamPerS; % інтервал дискретизації (с)
SamPerS=24; % кількість відліків на тривалості одного символу
PulseLength=3;
BT=0.3;

snr=-8; % відношення сигнал/перешкода (дБ)
N=10^3; % кількість переданих бітів

% Генерування випадкового інформаційного вектора
data = randi([0 1],N,1);
hMod = comm.GMSKModulator('BitInput', true,
    'SamplesPerSymbol',SamPerS, 'PulseLength', PulseLength,
    'BandwidthTimeProduct',BT);
hDemod = comm.GMSKDemodulator('BitOutput', true,
    'SamplesPerSymbol',SamPerS, ...
    'PulseLength',
    PulseLength, 'BandwidthTimeProduct',BT);
% Створення об'єкту для розрахунку BER (bit error ratio -
% частота збою біт)з урахуванням затримки в алгоритмі Вітербі
hError = comm.ErrorRate('ReceiveDelay', hDemod.TracebackDepth);

modSignal = step(hMod, data);
LS=length(modSignal);

% Підготовка бітів для відображення:
for i=0:N-1
    data1(SamPerS*i+1:SamPerS*(i+1))=data(i+1);
end

t=1:LS/50;
Ph=angle(modSignal(t));
Ph=unwrap(Ph);
figure
plot(t,data1(t),t,Ph(t));
title('Бітова інформація й розгорнута фаза GMSK-сигналу');

% Додатковий зсув сигналу по частоті:
df=0*10^3; % зсув по частоті (Гц)
omeg=2*pi*df;
t=(0:dtf:dt*N-dtf)';
modSignal = modSignal.*exp(1i*omeg*t);

% Амплітудний спектр сигналу:
Fs=1/dtf; % частота квантування (Гц) - полоса сигналу
NFFT = 2^nextpow2(LS); % кількість фільтрів ШПФ
FM=fft(modSignal,NFFT);
dFFT=Fs/NFFT; % Розміщення фільтрів ШПФ (Гц):
f=0:dFFT:Fs-dFFT; % розмітка частотної вісі

FMmod=abs(FM);

```

```

%plot(f,FMmod);

% Розрахунок зсуву спектра:
[~,nfmax]=max(FMmod);
dfest=nfmax*dFFT;
fprintf('%.3f   %.3f   %.3f\n',df,dfest,dfest-df);

% Пропускаємо сигнал через AWGN-канал
% noisySignal = step(hAWGN, modSignal);
sigma=10^(-snr/20)/sqrt(2);
gaussnoise=(randn(LS,1)+1i*randn(LS,1))*sigma;
noisySignal = modSignal+gaussnoise;

% Розрахунок відношення сигнал/перешкода:
Ps=sum(abs(modSignal).^2);
Pn=sum(abs(gaussnoise).^2);
qizm=10*log10(Ps/Pn);
fprintf('snr=%.2f   qizm=%.2f\n',snr,qizm);

% Спектр після накладання перешкод:
FMnoise=fft(noisySignal,NFFT);
FMnoisemod=abs(FMnoise);
figure
plot(f,FMnoisemod);

receivedData = step(hDemod, noisySignal);
errorStats = step(hError, data, receivedData);

fprintf('Кількість помилкових бітів = %d, BER = %f\n', ...
        errorStats(2),errorStats(1));

```

В програмі після генерування GMSK-сигналу по випадковій послідовності бітів вставлено блок зсуву сигналу на задану частоту. Потім обчислюється спектр сигналу за допомогою функції `fft()` (Fast Fourier transform — швидке перетворення Фур'є). На сигнал накладається білий Гаусів шум, при цьому потужність перешкоди обчислюється за заданим відношенням сигнал/перешкода в припущенні одиничної потужності генерованого сигналу GMSK, що дійсно забезпечується функцією `comm.GMSKModulator()`. Для контролю реального співвідношення с/п реалізовано його вимірювання за сумарною потужністю сигналу й завади. Зазначимо, що таке вимірювання обчислює співвідношення с/п не в реальній смузі сигналу, а у всій смузі частоти квантування. Сигнал після накладання перешкод надходить на декодування, і обчислюється кількість збійних біт і частота збоїв (BER – bit error ratio). Наприклад, для співвідношення с/п **-8 дБ** отримаємо:

```
snr=-8.00   qizm=-8.05
```

```
Кількість помилкових бітів = 8, BER = 0.008130
```

Як бачимо, задане і вимірне співвідношення с/п збігаються.

На рис 3.12 наведена розгорнута фаза сигналу разом із вхідною бітовою інформацією до накладення перешкод (відображаються всі 24 відліки на тривалості символу):

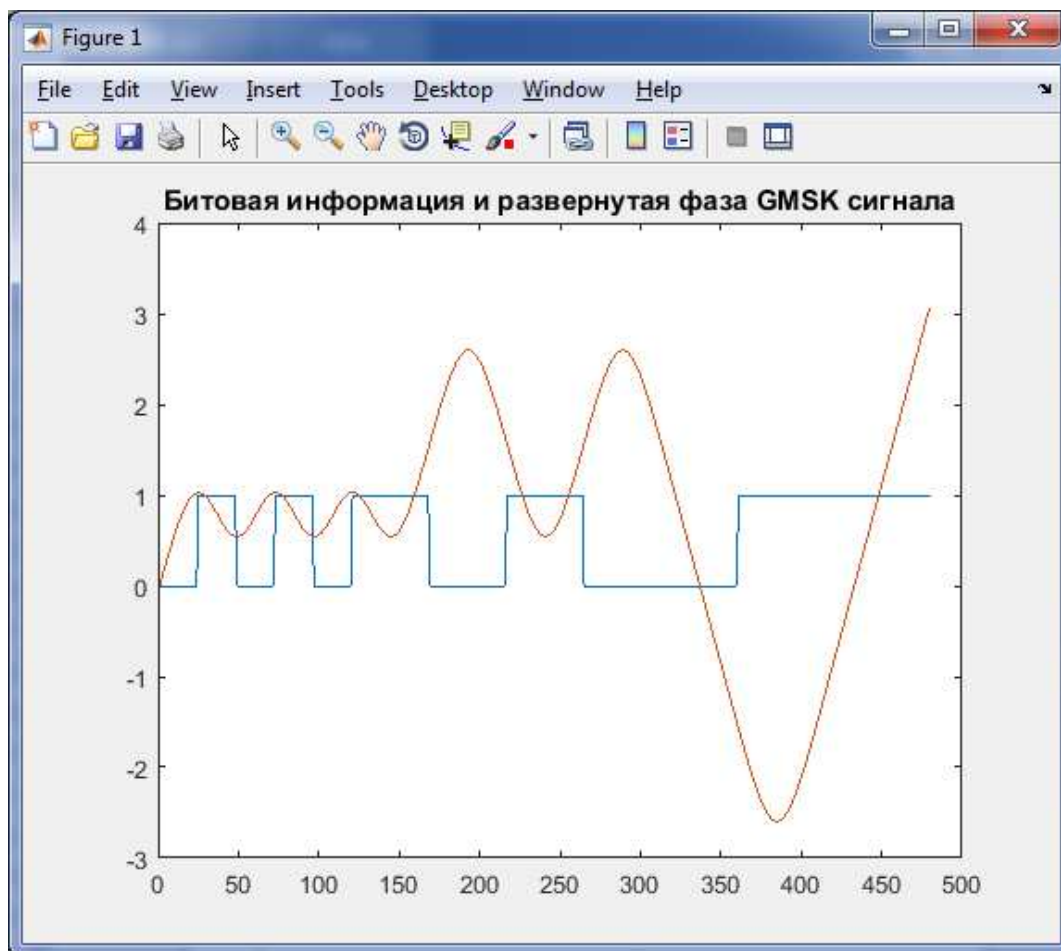


Рис. 3.12. Бітова інформація і розгорнута фаза GMSK-сигналу до накладення перешкод

Тут для усунення стрибків фази, пов'язаних з переходом $\pm\pi$ використана функція `unwrap()`.

Після накладення перешкод (співвідношення с/п **0 дБ**) фаза значно спотворюється (рис. 3.13):

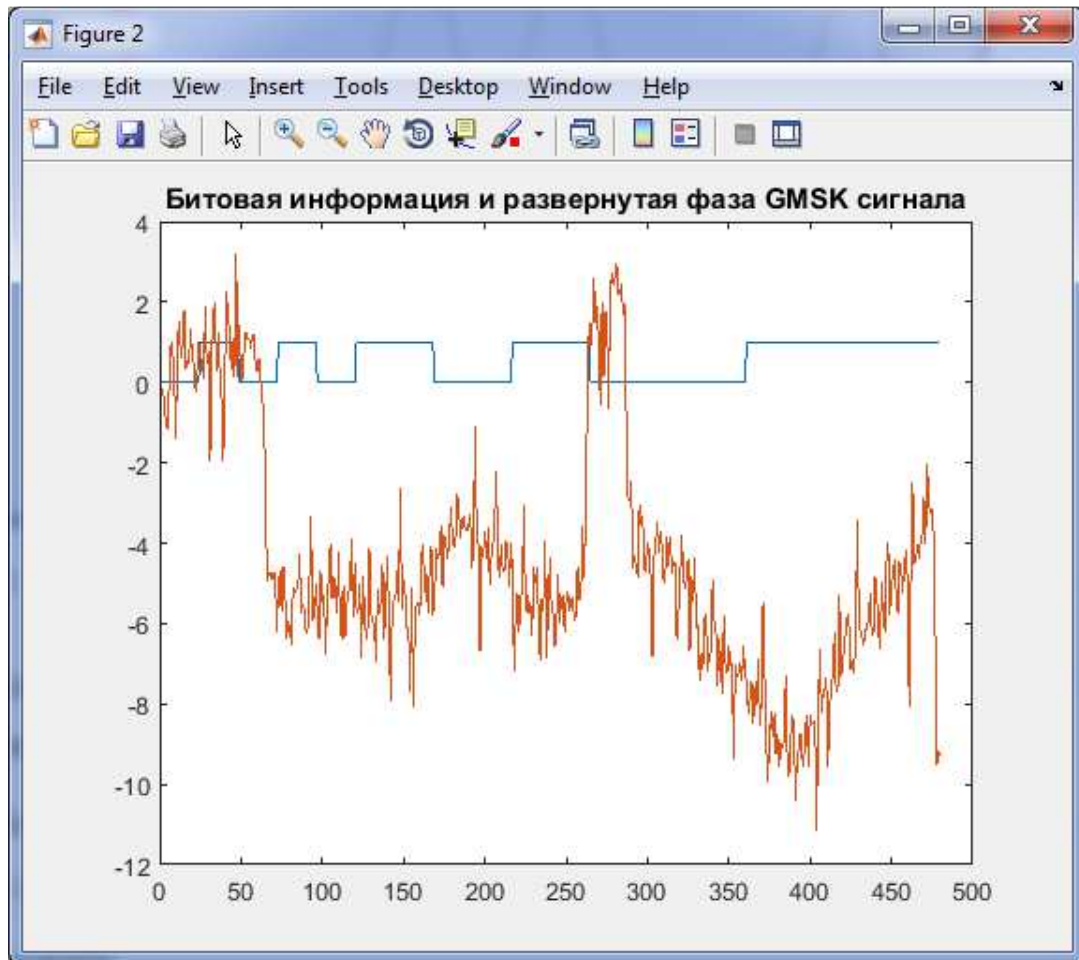


Рис.3.13. Бітова інформація і розгорнута фаза GMSK-сигналу після накладення перешкод

Незважаючи на значні зміни фази після накладення перешкод, забезпечується безпомилкове декодування бітової інформації (BER менше 10^{-6}).

Розглянемо тепер спектр GMSK-сигналу і порівняємо його зі спектром BPSK-сигналу (модуляція фази на два положення).

Для зручності аналізу сигнали змістимо на 50 кГц, а амплітуду сигналу будемо нормувати на максимум і значення відкладати в логарифмічному масштабі:

```
[Amax,~]=max(FMnoisemod);
FMnoisemod=20*log10(FMnoisemod/Amax);
figure
plot(f,FMnoisemod);
ylim([-50 0]);
title('Спектр GMSK-сигналу (зсув на 50 кГц)');
xlabel('Частота (Гц)');
ylabel('Амплітуда (дБ)');
```

Для моделювання сигналу BPSK використовується функція `pskmod()` [5], в якій відсутній параметр, що визначає кількість відліків на тривалості одного символу. Введемо його самостійно, дотримуючись [8]:

```
% Повтор комплексних амплітуд:
for k=1:N
    yp((k-1)*SamPerS+1:k*SamPerS)=y(k); % length(yp)=N*SamPerS
end
```

Тут вектор **y** комплексний сигнал на виході функції `pskmod()`. Відповідно, перед декодуванням необхідно буде залишити один відлік на символі процедурою вибору максимального по амплітуді відліку на довжині кожного символу:

```
% Залишаємо максимальний відлік на символі:
for k=1:N
    [~,imax]=max(abs(WGNY(SamPerS*(k-1)+1:SamPerS*k)));
    y(k)=WGNY(SamPerS*(k-1)+imax);
end
```

Тут **wgny** – BPSK-сигнал після накладення перешкод.

Переконаємося, насамперед, що наші вимірювання спектру узгоджуються з теоретичними даними, наведеними в розділі 3.1. У відповідності з рис. 3.11 ширина спектру GMSK-сигналу при **BT=0.3** за рівнем **-20 дБ** становить приблизно **20 кГц** при швидкості передачі **20 кГц/с**. При цих же параметрах реалізація спектру в нашій моделі виглядає таким чином (рис. 3.14, співвідношення с/п **25 дБ**):

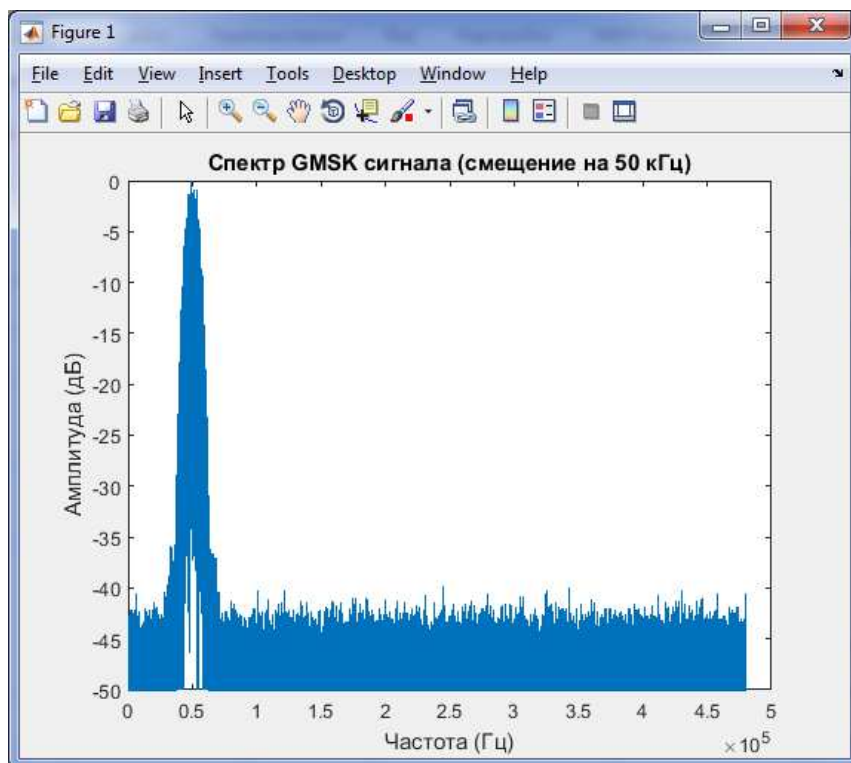


Рис.3.14. Спектр GMSK-сигналу при швидкості передачі 20 Кбіт/с

З рисунка видно, що реалізація спектра в цілому відповідає теоретичному спектру. Ми можемо легко обчислити ширину спектра за заданим рівнем з допомогою чудової функції `find()`:

```
% Вимірювання ширини спектра по заданому рівню:
Q=-20;
idxf=find(FMnoisemod >= Q,1,'first');
idxl=find(FMnoisemod >= Q,1,'last');
df=(idxl-idxf)*dFFT/10^3;
fprintf('Ширина спектра за рівнем %d дБ = %.3f кГц\n',Q,df);
```

Результат вимірювання такий:

Ширина спектра за рівнем **-20 дБ = 20.570 кГц**

Повернемось до параметрів GMSK-сигналу, використовуваним в AIS; спочатку побудуємо спектр обох сигналів фактично без перешкод (співвідношення с/п **25 дБ**) (рис. 3.15 і 3.16):

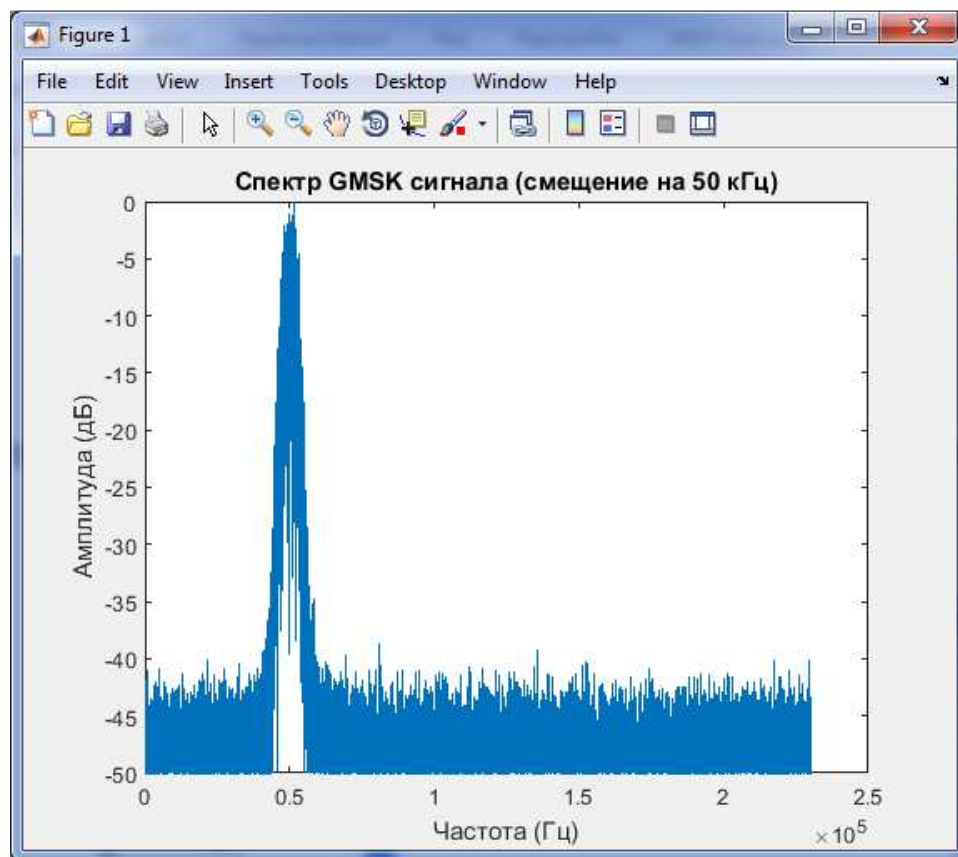


Рис. 3.15. Спектр GMSK-сигналу при співвідношенні с/п 25 дБ

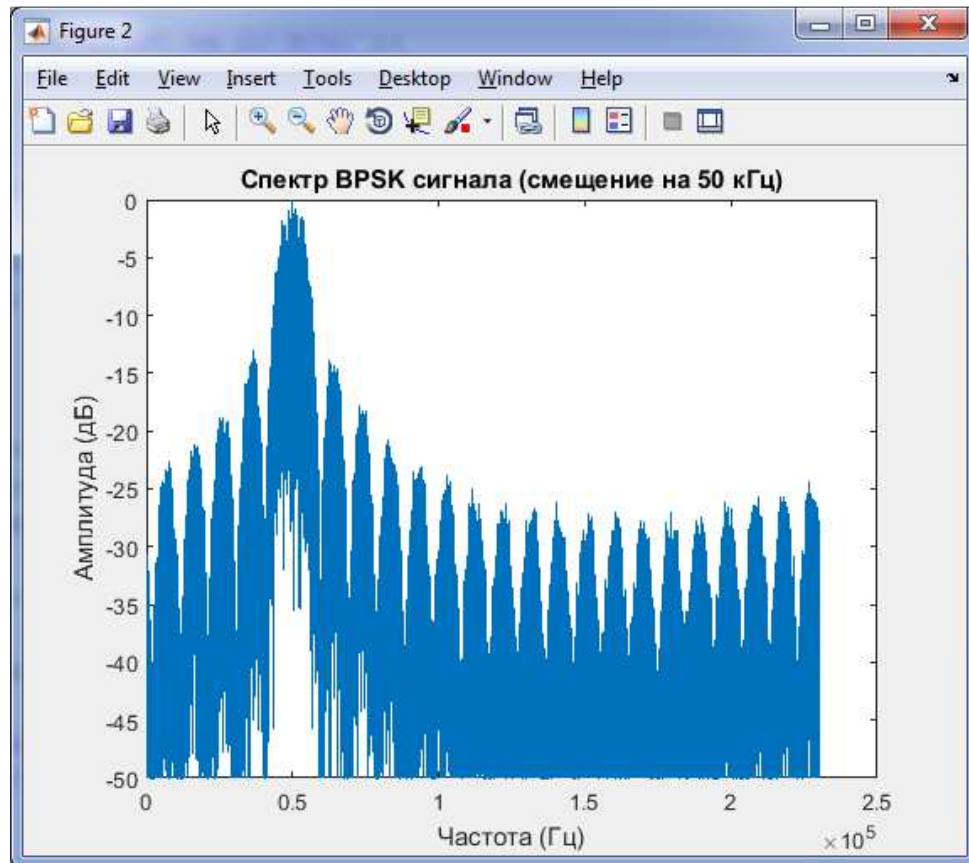


Рис. 3.16. Спектр BPSK-сигналу при співвідношенні с/п 25 дБ

З рисунків добре видно значну перевагу сигналу GMSK над BPSK. По-перше, спектр BPSK-сигналу вже за рівнем **-30 дБ** забиває всю смугу квантування, тоді як зі спектром GMSK-сигналу це відбувається тільки при **-45 дБ**. Порівняльна ширина спектрів обох сигналів за різними рівнями наведена в таблиці 3.1:

Таблиця 3.1

Рівень, дБ	Ширина спектра GMSK-сигналу, кГц	Ширина спектра BPSK-сигналу, кГц
-5	4,729	8,297
-10	7,118	13,711
-20	10,003	67,198
-30	11,703	Смуга квантування
-40	17,685	Смуга квантування

Зрозуміло, що відмінності в спектрі не тільки призведуть до різної можливості заповнення частотного діапазону, але й істотно позначаться на якості передачі інформації.

Спектри обох сигналів для робочих відносин сигнал/перешкода, наприклад, для **-8 дБ** виглядають наступним чином (рис. 3.17, 3.18):

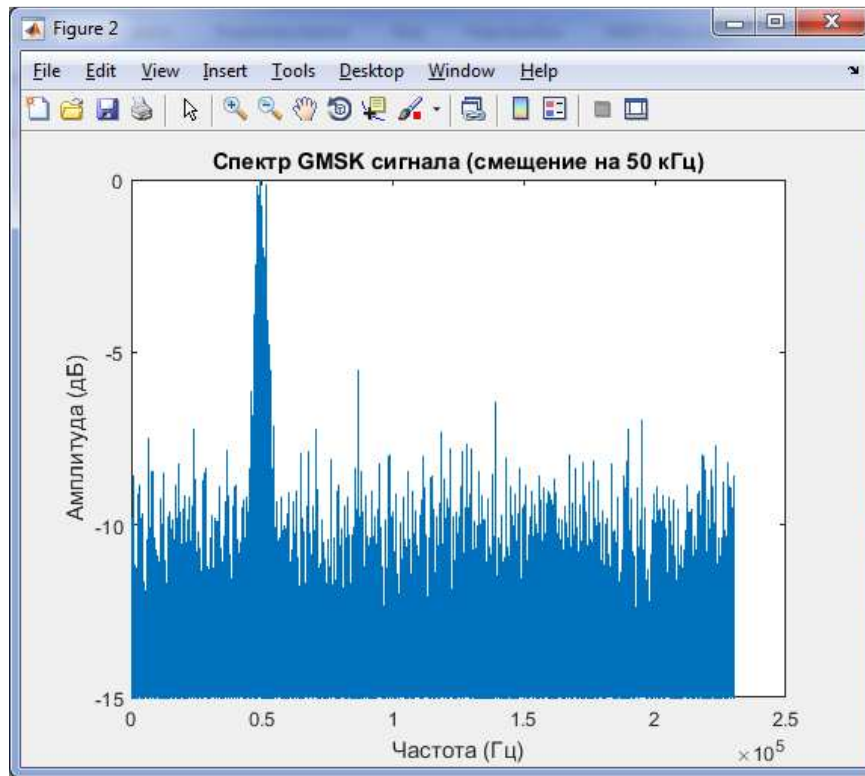


Рис. 3.17. Спектр GMSK-сигналу при співвідношенні с/п -8 дБ

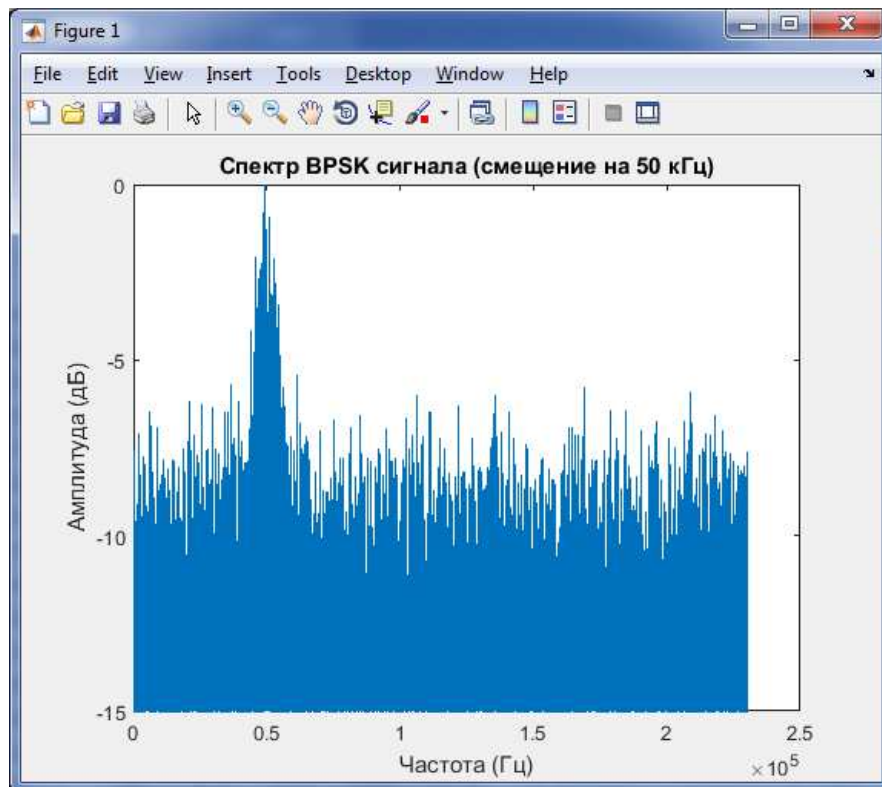


Рис. 3.18. Спектр BPSK-сигналу при співвідношенні с/п -8 дБ

Зрозуміло, що на тимчасовій амплітудній розгортці такі сигнали не будуть виявлятися візуально (сигнал присутній лише на 0,1 розгортки в середині, рис. 3.19):

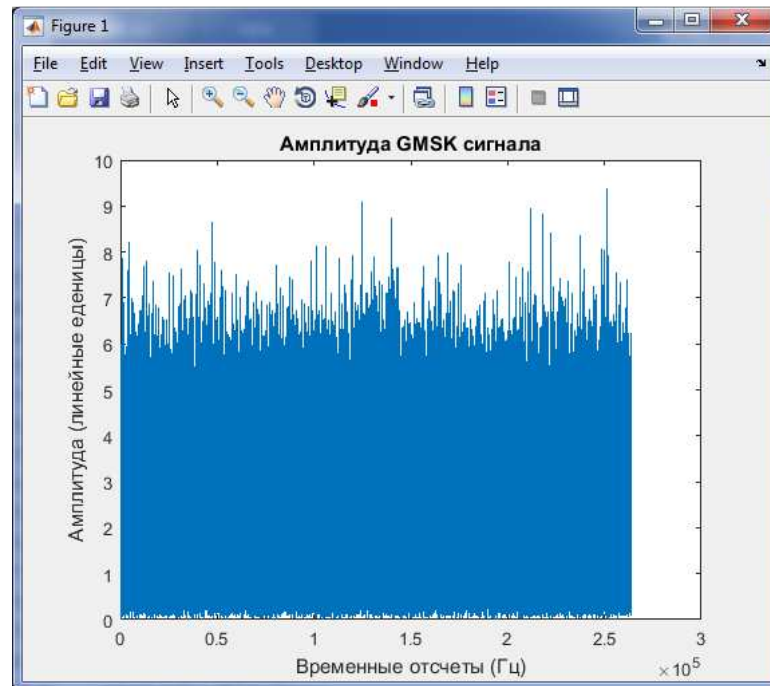


Рис. 3.19. Амплітудна розгортка GMSK-сигналу при співвідношенні с/п -8 дБ

Для візуального виявлення необхідно збільшити співвідношення с/п хоча б до 0 дБ (рис. 3.20):

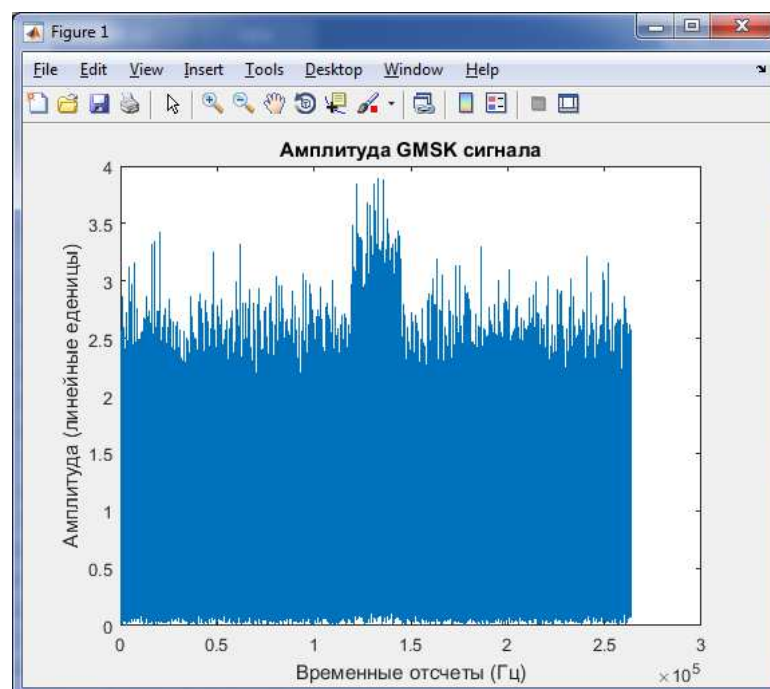


Рис. 3.20. Амплітудна розгортка GMSK-сигналу при співвідношенні с/п 0 дБ

Ми бачимо, що співвідношення с/п в смузі GMSK-сигналу становить близько 10 дБ, для BPSK-сигналу – близько 7 дБ. Однак виграш GMSK-сигналу істотно більше. Для оцінки виграшу необхідно звернутися до основної характеристики каналу зв'язку — залежності BER від співвідношення с/п. Для попередньої оцінки досить провести вимірювання в обох каналах для декількох значень співвідношення с/п. По моделі GMSK-сигналу отримуємо (зсув по частоті прибрано):

```
snr=-8.00  qizm=-8.01
```

```
Кількість помилкових бітів = 4, BER = 0.004065
```

У моделі BPSK-сигналу проведемо вимірювання BER для 2-х значень співвідношення с/п:

```
snr=-8.00  qizm=-7.98
```

```
Кількість помилкових бітів = 117, BER = 0.11700
```

```
snr=0.00  qizm=0.01
```

```
Кількість помилкових бітів = 5, BER = 0.00500
```

Отримали, що при співвідношенні с/п **-8 дБ** GMSK-сигнал забезпечує прийнятний рівень збоїв, який може бути повністю усунений, наприклад, використанням коригувальних кодів, тоді як в каналі зв'язку з BPSK-сигналом передача інформації вже практично не можлива (збоїть кожен 10-ий біт). Для забезпечення приблизно такого ж рівня BER, як в каналі з GMSK, довелося збільшити співвідношення с/п на **8 дБ**.

З отриманих даних не випливає, що у всіх реальних системах зв'язку можна працювати з сигналами меншими по потужності, ніж фонові потужності перешкод. Наприклад, в технології спостереження руху повітряних суден ADS-B та в технології автоматичної ідентифікації морських суден AIS використовується імпульсна передача невеликого обсягу інформації (довжина повідомлення ADS-B – не більше 112 мкс, AIS – 27 мс, розділ 2.1) з частотою до декількох повідомлень в секунду.

У цьому випадку тимчасове положення сигналу на приймальній стороні не відомо, і первинне визначення інтервалу наявності сигналу, як правило, здійснюється за амплітудною розгорткою і вимагає значного співвідношення с/п. Крім того, як зазначалося вище, в MATLAB реалізована спрощена модель сигналу, зокрема не реалізований зсув по частоті між приймачем і передавачем, який завжди присутній у реальних системах і вимагає адаптивної оцінки та усунення.

Завершимо розділ приведенням кривої залежності BER від співвідношення сигнал/перешкода для GMSK- і BPSK-сигналів. Для цього нам достатньо реалізувати циклічне обчислення BER для співвідношення с/п із заданого

діапазону з заданим кроком. Після видалення демонстраційного друку і графіки програма виглядає цілком лаконічно:

```
% Програма для дослідження характеристик сигналів GMSK
clear all;

% Параметри відповідають прийнятими в AIS
Vt=9600;% Швидкість передачі біт/с
dt=1/Vt;% Тривалість символу в с (104,17 мкс)
SamPerS=24;% кількість відліків на тривалості одного символу
dtf=dt/SamPerS;% інтервал дискретизації (с)
PulseLength=3;
BT=0.3;

N=4*10^5;% кількість переданих бітів
LS=N*SamPerS;% загальна кількість відліків сигналу

hMod = comm.GMSKModulator('BitInput', true,
    'SamplesPerSymbol',SamPerS, ...
    'PulseLength',
    PulseLength,'BandwidthTimeProduct',BT);
hDemod = comm.GMSKDemodulator('BitOutput', true,
    'SamplesPerSymbol',SamPerS, ...
    'PulseLength',
    PulseLength,'BandwidthTimeProduct',BT);
% Створення об'єкту для розрахунку BER
% (bit error ratio - частота збою біт)
% з урахуванням затримки в алгоритмі Вітербі
hError = comm.ErrorRate('ReceiveDelay', hDemod.TracebackDepth);

qmin=-12;% мінімальне відношення с/п (дБ)
qmax=-3.5;% максимальне відношення с/п (дБ)
dq=0.5;% крок по відношенню с/п (дБ)
snr=qmin:dq:qmax;
n=length(snr);
BER(1:n)=0;
fprintf('snr (дБ) Кіл-ть помилок BER \n');
for i=1:n
    data = randi([0 1],N,1); % Генерування випадкового
    % інформаційного вектора
    modSignal = step(hMod, data);

    % Пропускаємо сигнал через AWGN-канал
    sigma=10^(-snr(i)/20)/sqrt(2);
    gaussnoise=(randn(LS,1)+1i*randn(LS,1))*sigma;
    noisySignal = modSignal+gaussnoise;

    % Демодуляція:
    receivedData = step(hDemod, noisySignal);
    errorStats = step(hError, data, receivedData);
    reset(hError);
    BER(i)=errorStats(1);
```

```

    if BER(i) == 0
        BER(i)=10^(-5);
    end
    fprintf('%5.1f      %4d      %.5f\n',
snr(i),errorStats(2),BER(i));
end
figure
plot(snr,log10(BER));
title('Залежність BER від відношення с/п');
xlabel('Відношення с/п (дБ)');
ylabel('lg(BER)');
grid on
hold on

```

Для сигналу BPSK аналогічна програма виглядає так:

```

%% Програма для дослідження характеристик сигналів
% з фазовою маніпуляцією (PSK)
%clear all;
M = 2;          % розмір алфавіту повинен бути ступенем числа 2.
symbol_order='gray';% 'bin'(default)/'gray'
ini_phase0=0;% початкова фаза у радіанах до передачі
N=4*10^5;       % кількість символів у вхідному масиві
SamPerS=24;

tic
% Генеруємо N випадкових M-арних символів:
msg = randi([0 M-1],N,1);

% Генеруємо PSK-сигнал:
y = pskmod(msg,M,ini_phase0,symbol_order);

% Повтор комплексних амплітуд:
for k=1:N
    yp((k-1)*SamPerS+1:k*SamPerS)=y(k); % length(yp)=N*SamPerS
end
N1=N*SamPerS;

% Для сигналів AIS:
Vt=9600;% Швидкість передачі біт/с
dt=1/Vt; % Тривалість символу у с (104,17 мкс)

qmin=-4; % мінімальне відношення с/п (дБ)
qmax=5.0; % максимальне відношення с/п (дБ)
dq=0.5; % крок по відношенню с/п (дБ)
snr=qmin:dq:qmax;
n=length(snr);
BER(1:n)=0;
fprintf('snr (дБ) Кількість помилок      BER \n');
for i=1:n
    % Пропускаємо сигнал через AWGN-канал
    sigma=10^(-snr(i)/20)/sqrt(2);

```

```

gaussnoise=(randn(1,N1)+1i*randn(1,N1))*sigma;
WGNy = yp+gaussnoise;

% Залишаємо максимальний відлік на символі:
for k=1:N
    [~,imax]=max(abs(WGNy(SamPerS*(k-1)+1:SamPerS*k)));
    y(k)=WGNy(SamPerS*(k-1)+imax);
end

% Декодування прийнятого сигналу:
ini_phase1=0; % початкова фаза у радіанах після прийому
WGNz = pskdemod(y,M,ini_phase1,symbol_order);

% Розраховуємо кількість и інтенсивність збоїв
% (BER - bit error rate):
[numErr,BER(i)] = biterr(msg,WGNz);
if BER(i) == 0
    BER(i)=10^(-5);
end
fprintf('%5.1f      %4d      %.5f\n',
snr(i),numErr,BER(i));
end
plot(snr,log10(BER));
legend('GMSK','BPSK');

```

Результатом спільної роботи обох програм є рис. 3.21.

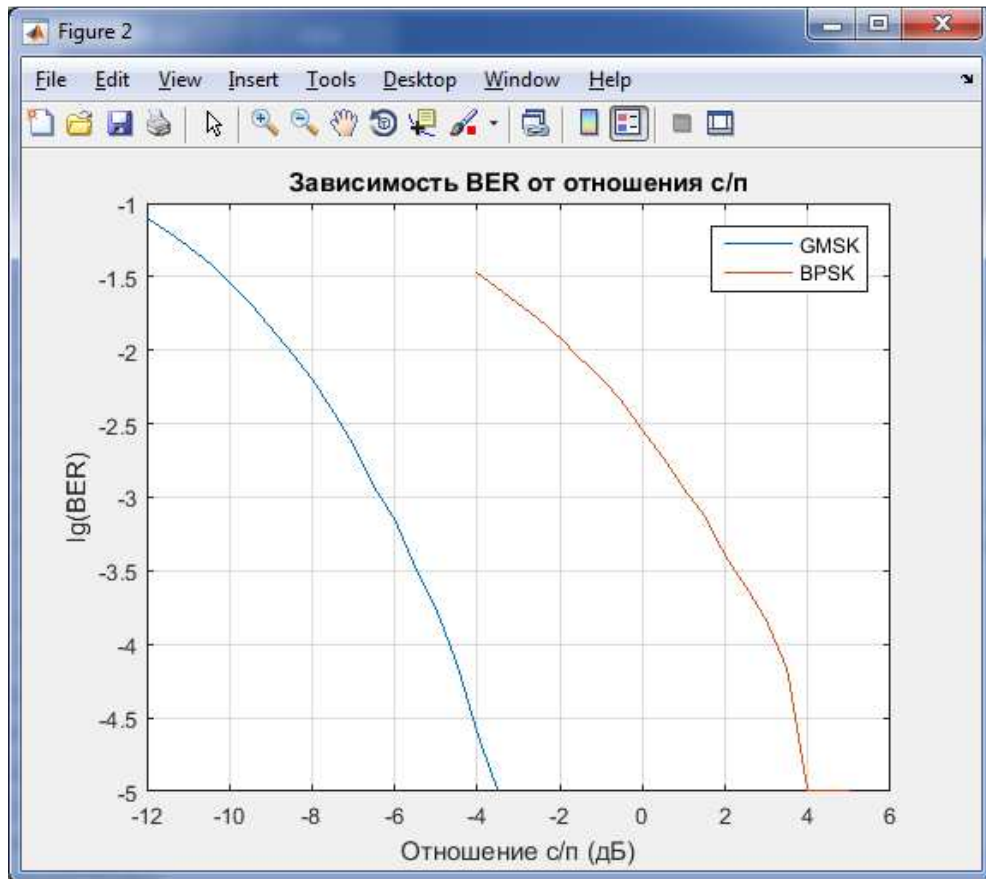


Рис. 3.21. Залежність BER від співвідношення сигнал/перешкода для GMSK- і BPSK-сигналів

Зазначимо, що опція `hold on` в MATLAB дозволяє на один графік нанести дані декількох програм.

Отримані графіки підтверджують вищенаведені дані про виграш близько 8 дБ для малих рівнів BER ($10^{-3} - 10^{-5}$). Для більш високих рівнів BER виграш на 1–2 дБ менше.

3.3. Загальна інформація про методи передачі повідомлень морськими судами

Як і повітряні судна, морські судна постійно передають сигнали зі своїми даними. При русі судна інтервал передачі становить від 2 до 10 секунд і 3 хвилини при стоянні на якорі. Залежно від класу, судна комплектуються двома видами передавальної апаратури. До класу А (AIS А) віднесені судна валовою місткістю понад 300 реєстрових тонн, які виконують міжнародні рейси, судна валовою місткістю 500 реєстрових тонн і більше, які не здійснюють міжнародні рейси, і всі пасажирські судна. Для них виділений канал зв'язку на частоті **161,975 МГц**. Потужність передавача 12,5 Вт. Інші судна віднесені до класу В (AIS В), і вони повинні передавати свої сигнали на частоті **162,025 МГц**. Потужність передавача в п'ять разів нижче. Швидкість передачі завжди 9600 Кбіт/с, відповідно, тривалість передачі одного символу близько 104 мкс.

Для передачі інформації використовується GMSK-модуляція з $BT = 0.3$ і довжиною імпульсу (pulse length) = 3. Основне повідомлення має наступну структуру:

- Час розгону (Ramp-up time) — 8 бітів;
- Синхроімпульс (Training sequence) — 24 біта;
- Прапор початку повідомлення (Start flag) — 8 бітів;
- Інформація про судно — 168 бітів;
- Код CRC — 16 бітів;
- Прапор кінця повідомлення (End flag) — 8 бітів;
- Буфер (Buffer) — 24 біта.

Вся довжина повідомлення складає 256 бітів. Для деяких типів повідомлень загальна довжина може досягати 512 бітів.

Start flag = End flag = 0x7E. Для запобігання утворення хибних прапорів, після кожних 5 послідовних «1» вставляється 0. Синхроімпульс задається послідовністю **1010101010101010101010**. В кожному байті передача починається з молодших біт (transmit LSB first). Додатково використовується кодування NRZI (Non Return to Zero Inverted), що полягає в заміні «1» на попередній символ і «0» на інверсію попереднього символу.

Згідно з [9], структура інформаційної частини повідомлення для AIS класу А наведена в таблиці 3.2.

Таблиця 3.2.

Параметр	Кількість бітів	Короткий опис
Номер повідомлення (Message ID)	6	1,2 або 3
Індикатор повторення (Repeater indicator)	2	Кількість повторень повідомлення 0-3, 3 – не повторювати (використовується репітером)
Ідентифікатор судна (User ID)	30	MMSI (Maritime Mobile Service Identity) number — ідентифікатор морської рухомої служби (IMPC)
Навігаційний статус (Navigational status)	4	0 = в дорозі з використанням двигуна, 1 = на якорі, 2 = не під командою, 3 = обмежена маневреність, 4 = обмеження по тязі, 5 = пришвартоване, 6 = на міліні, 7 = зайнято промислом, 8-15 = резерв та додаткова інформація
Швидкість розвороту (Rate of turn - ROT_{AIS})	8	0 до +126 = поворот праворуч до 708 град/хв.; 0 до -126 = поворот ліворуч до 708 град/хв. ($ROT_{AIS} = 4.733 \sqrt{ROT_{sensor}}$ град/хв., де ROT_{sensor} – показання індикатора повороту); +127 = поворот ліворуч більш ніж 10 град/хв.; -127 = поворот праворуч більш ніж 10 град/хв.; -128 = інформація про розворот недоступна.
Швидкість руху (Speed over ground – SOG)	10	Швидкість руху у 1/10 долях вузла (0-102.2 вузла); 1 023 = не доступно, 1 022 = 102.2 вузла або вище
Точність вимірювання позиції (Position accuracy)	1	1 = висока (≤ 10 м) 0 = низька (> 10 м)
Довгота (Longitude)	28	Довгота в 1/10000 долях хвилини від -180 до +180 градусів, Східна довгота – позитивні числа, Західна довгота – від'ємні числа; 181 градус = 6791 AC0h = недоступне
Широта (Latitude)	27	Широта в 1/10000 долях градуса від -90 до +90 градусів, Північна широта – позитивні числа, Південна широта – від'ємні

Параметр	Кількість бітів	Короткий опис
		числа. 91 градус = 3412140h = недоступне
Курс по землі (Course over ground – COG)	12	Курс в 1/10 долях градуса (0-3599). 3600 = E10h = недоступне
Справжній напрямок руху (True heading)	9	В градусах від 0 до 359, 511 = недоступне
Мітка часу (Time stamp)	6	Секунди формування повідомлення за UTC – часом електронної системи координат (EPFS) в діапазоні від 0-59, 60-63 у нештатних ситуаціях
Індикатор спеціального маневру (Special maneuver indicator)	2	0 = недоступне, 1 = немає спеціального маневру 2 = в режимі спеціального маневру
Резерв (Spare)	3	Не використовується, резерв для майбутніх доробок
RAIM-flag	1	Прапор наявності пристрою електронного позиціонування судна, 0 = не використовується, 1 = використовується
Стан зв'язку (Communication state)	19	Стан зв'язку в повідомленнях AIS класу А. Використовується при плануванні наступної передачі, щоб уникнути взаємних перешкод
Всього бітів (Number of bits)	168	

Основне повідомлення AIS класу В по структурі не сильно відрізняється від наведеної структури для класу А. Довжина повідомлення ті ж 168 бітів, номер повідомлення завжди дорівнює 18. Відсутні поля «Навігаційний статус», «Швидкість розвороту» та «Індикатор спеціального маневру» (всього 14 бітів). Замість них додано 7 прапорців, і резерв збільшено з 3 до 10 бітів. Починаючи з поля «Курс по землі» поля трохи зміщені за номерами бітів [10].

При порівнянні з технологією ADS-B спостереження руху повітряних суден бачимо, що розробники AIS біти не економили, і вся інформація розміщується практично в початковому вигляді. Якщо технологія ADS-B корінням йде в післявоєнні роки, то AIS порівняно нова технологія, практична експлуатація її почалася тільки в 2000 році.

3.4. Дослідження реальних сигналів AIS

3.4.1. Алгоритм виявлення сигналів AIS

Після детального розгляду методів передачі і кодування інформації в повідомленнях, ми можемо перейти до найцікавішої частини розділу, а саме до розгляду алгоритму виявлення таких сигналів з виходу приймача RTL-SDR.

Як і у випадку з сигналами ADS-B, в додатку до книги [2] наведена програма з відкритим кодом **aisLiveData**, яка вирішує поставлену задачу (авторські права на програму належать корпорації MathWorks).

Розглянемо, насамперед, яка програма встановлює параметри на приймач:

```
% Робоча частота приймача в Гц (tuner center frequency in Hz):
rtlsdr_fc          = 161.975e6; % для AIS A
% Підсилення приймача в дБ (tuner gain in dB):
rtlsdr_gain        = 60;
% Частота квантування вхідного сигналу - тактова частота
% (tuner sampling rate), максимальне значення 3.2e6 Гц
rtlsdr_fs          = 9600*24=230400;
% Розмір запитуваного за 1 раз об'єму вхідної інформації -
% фрейму(output data frame size):
rtlsdr_frmrlen     = 2^18;
% Формат вихідних квадратур (output data type single) - аналог
% float (4 байта):
rtlsdr_datatype    = 'single';
```

Як зазначалося у розділі 3.1, швидкість передачі сигналів AIS становить 9600 бітів/с, що відповідає довжині передачі одного символу близько 104 мкс і це на 2 порядки нижче швидкості передачі сигналів ADS-B. Як бачимо, використовується частота квантування, що забезпечує на тривалості символу рівно 24 відліків. Кількість відліків одного фрейму така ж, як при виявленні сигналів ADS-B (тривалість ~1,14 с).

Насамперед, у програмі окремою функцією генерується синхроімпульс GMSK, що відповідає використовуваній послідовності бітів:

```
function syncIdeal = syncGen(samplesPerSymbol)
% Generate ideal AIS sync waveform

% Copyright 2016, The MathWorks, Inc.

% Set up training sequence
tr1=logical([1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0]); % 24 bits

% Set up GMSK Modulator
mod=comm.GMSKModulator;
```

```

mod.BandwidthTimeProduct=.3;
mod.SamplesPerSymbol=samplesPerSymbol;
mod.BitInput=true;
mod.PulseLength=3;
mod.SymbolPrehistory=[1 -1];

% Apply NRZI (Non-return-to-zero inverted) encoding
for ii=2:length(tr1)
    if tr1(ii)==1
        tr1(ii)=tr1(ii-1);
    else
        tr1(ii)=~tr1(ii-1);
    end
end

% Generate GMSK waveform for training sequence
syncIdeal = step(mod,tr1');

```

Як бачимо, генерація проводиться функцією `comm.GMSKModulator()`, яку ми використовували для дослідження характеристик GMSK-сигналів в розділі 3.2. Як зазначалося на початку цього розділу, у AIS використовується кодування NRZI (Non Return to Zero Inverted), що полягає в заміні «1» на попередній символ і «0» на інверсію попереднього символу. Таке перетворення призводить до наступної послідовності бітів, яка і подається на вхід модулятора: 100110011001100110011001. Зверніть увагу, що значення попереднього символу могло бути раніше змінено.

Отриманий в модуляторі сигнал проходить багатоступеневу обробку для отримання фазового образу ідеального синхроімпульсу:

```

syncCalc = syncGen(samplesPerSymbol);
syncIdeal =
unwrap(diff(angle(syncCalc(1:samplesPerSymbol*20+1)))));

```

Насамперед, звернемо увагу, що синхроімпульс обрізається на 4 біта. Як ми побачимо пізніше за наявними експериментальними даними, це дозволяє трохи збільшити кореляцію ідеального і реального синхроімпульсів. Виділяється фаза сигналу, потім вона замінюється на приріст фази (функція `diff()` видає різниці $s(i+1) - s(i)$), розмірність вихідного масиву на 1 зменшується). Функція `unwrap()` повертає фазу сигналу, усуваючи стрибки, викликані переходом фази через $\pm\pi$. У даному випадку операція зайва. На рис. 3.22 зображена фаза ідеального синхроімпульсу з використанням і без використання функції `diff()` (вихід `diff()` помножений на коефіцієнт 20):

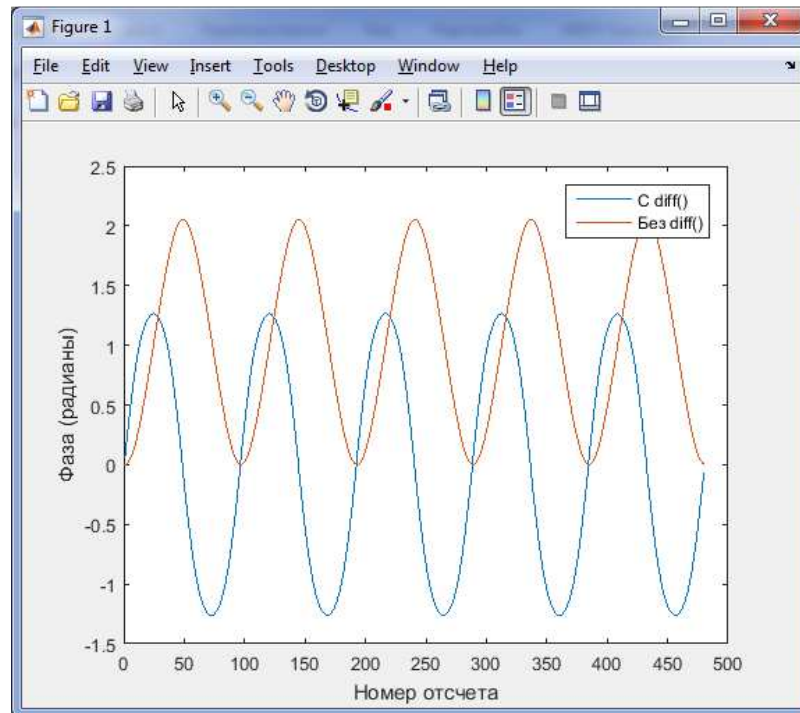


Рис. 3.22. Фаза і приріст фази ідеального синхроімпульсу

Як бачимо, два варіанти фази синхроімпульсу відрізняються тільки тимчасовим зсувом, масштабом і на адитивну константу (коефіцієнт кореляції практично дорівнює 1). Тому цілком можна обійтися без розрахунку приросту фази.

3.4.2. Дослідження записів сигналів AIS від судів в околиці м. Бостона

Паралельно з описом використовуваного алгоритму виділення сигналів AIS розглянемо наявні експериментальні дані. Разом з програмою автори надали записи сигналів AIS судів на Східному узбережжі США поблизу м. Бостон. Записано лише 2 фрейми з сигналами від 6 суден. Амплітудна розгортка фреймів виглядає наступним чином (рис. 3.23, 3.24):

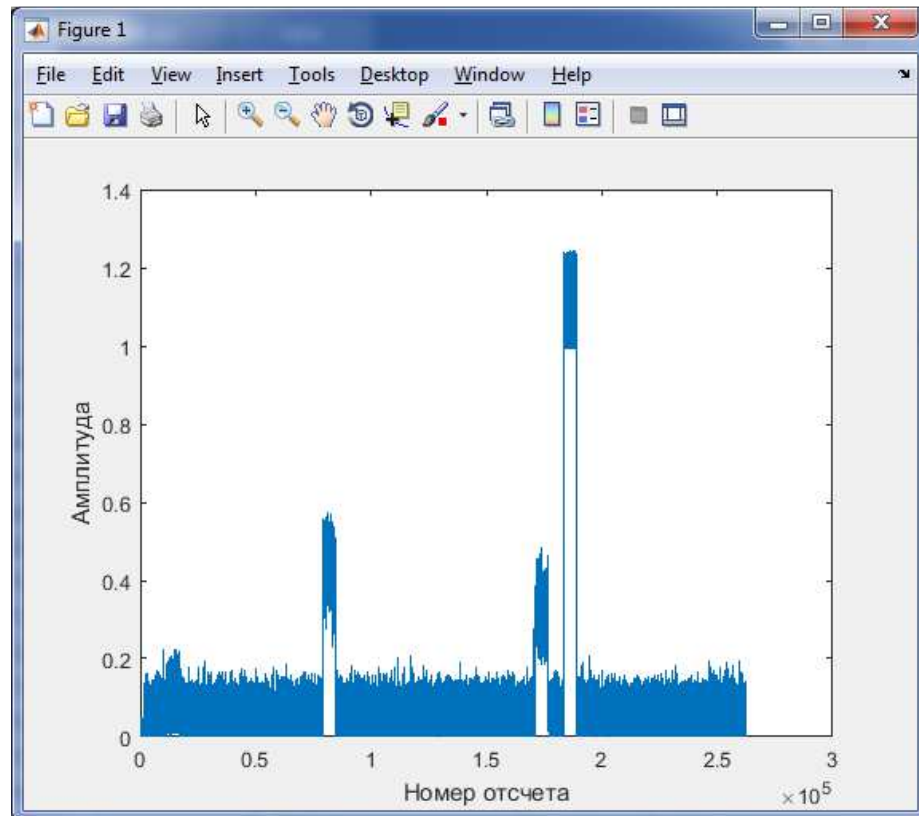


Рис. 3.23. Амплітудна розгортка фрейму 1

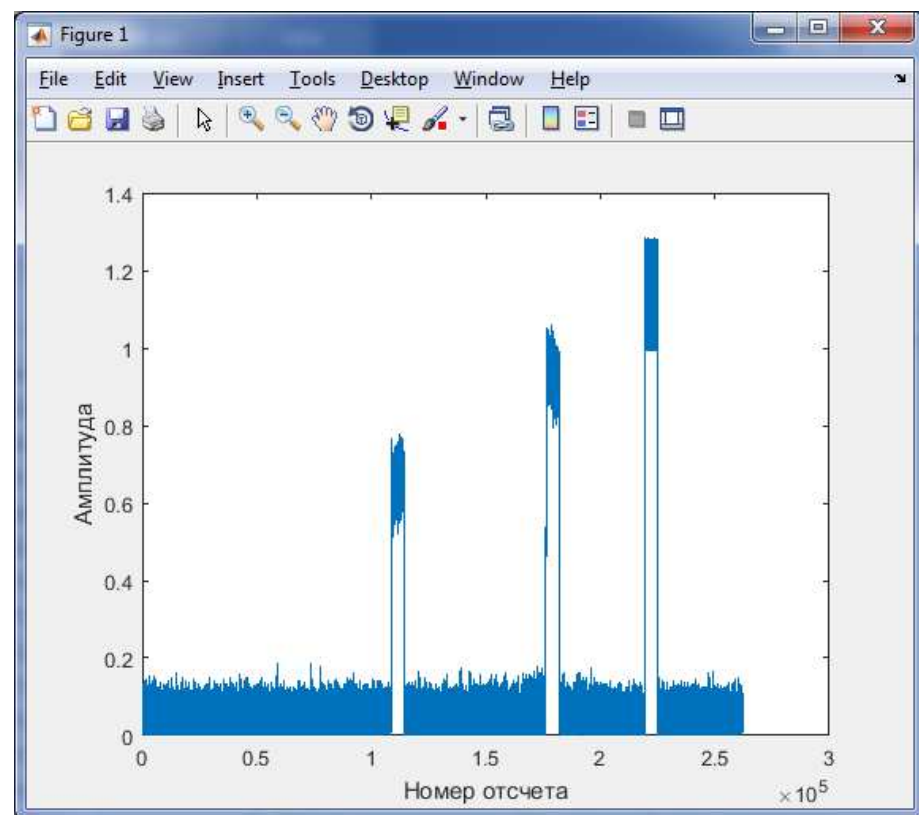


Рис. 3.24. Амплітудна розгортка фрейму 2

Відстань від судів до приймача не зазначено, але судячи по потужності сигналів, вона невелика. Наведемо відразу результати обробки фреймів штатною програмою:

Фрейм 1

```
corr=0.649
msgType = 1
MMSI:367164250 Latitude:42.38 Longitude:-71.0421
045789ED68200006BACB889840016C8BFFDC0860101600
```

```
corr=0.788
msgType = 1
MMSI:367432220 Latitude:42.3749 Longitude:-71.0477
04579A4870200006BAC4FF183F43393EFFDC0906B0187B
```

Фрейм 2

```
corr=0.319
msgType = 1
MMSI:367513020 Latitude:42.3804 Longitude:-71.0421
04579F36F3E00006BACB8C184011001DFFF401401A2593
```

```
corr=0.499
msgType = 1
MMSI:367513050 Latitude:42.3803 Longitude:-71.0425
04579F376B200016BACB0B98400E59A5FF80014031F6DD
corr=0.889
msgType = 1
MMSI:366941020 Latitude:42.3845 Longitude:-71.0596
04577C4D70000000EBAV7131840AC8AC45B760008A08D89
```

У розглянутій програмі роздруковується лише ідентифікатор MMSI судна, географічні координати і отримане повідомлення в 16-ковому вигляді (тільки дані і код CRC – 184 біта). Додатково введено обчислення коефіцієнта кореляції ідеального і прийнятого синхроімпульсу на інтервалі його виявлення програмою, а також виведений номер повідомлення. Сигнали в програмі аналізуються в порядку їх потужності, починаючи з найбільш потужного.

Бачимо, що всі сигнали отримані від суден класу А. На жаль, третій за потужністю сигнал в першому фреймі, який добре видно на амплітудній розгортці (рис. 3.24, співвідношення сигнал/перешкода близько 10 дБ у всій смузі частоти квантування), не виявлений, що може свідчити про помилки в алгоритмі або програмі. Отримані коефіцієнти кореляції для розглянутих потужних сигналів можуть вказувати на ті ж причини.

Аналогічно сайту www.flightradar24.com для відстеження повітряних суден, у США функціонує сайт www.marinetraffic.com для відстеження руху морських суден (можливо, і річкових теж). Ввівши отримані ідентифікатори MMSI, отримуємо наступну інформацію про судна, які передали в ефір досліджувані нами сигнали (рис. 3.26–3.28):



FREEDOM
Tug

© WILLIAM BURNI
MarineTraffic.com

US BOS

ATD: 2018-02-26 05:55 ATA: 2018-02-26 06:17 ⓘ

➔

Past Track Route Forecast

Status: **Stopped** Speed/Course: **0kn / 0°** Draught: **4.3m**

Received: 2 minutes ago (AIS Source: 3194)



SCARLETT ISABEL...
Offshore Supply Ship

© Matt Patnaude
MarineTraffic.com

US WOH US BOS

ATD: 2018-01-28 09:25 ATA: 2018-01-28 16:45 ⓘ

➔

Past Track Route Forecast

Status: **Stopped** Speed/Course: **0kn / 0°** Draught: **3.1m**

Received: 5 minutes ago

Рис. 3.25. Інформація про судна, сигнали яких записані у фреймі 1 (MMSI 367164250 та 367432220)

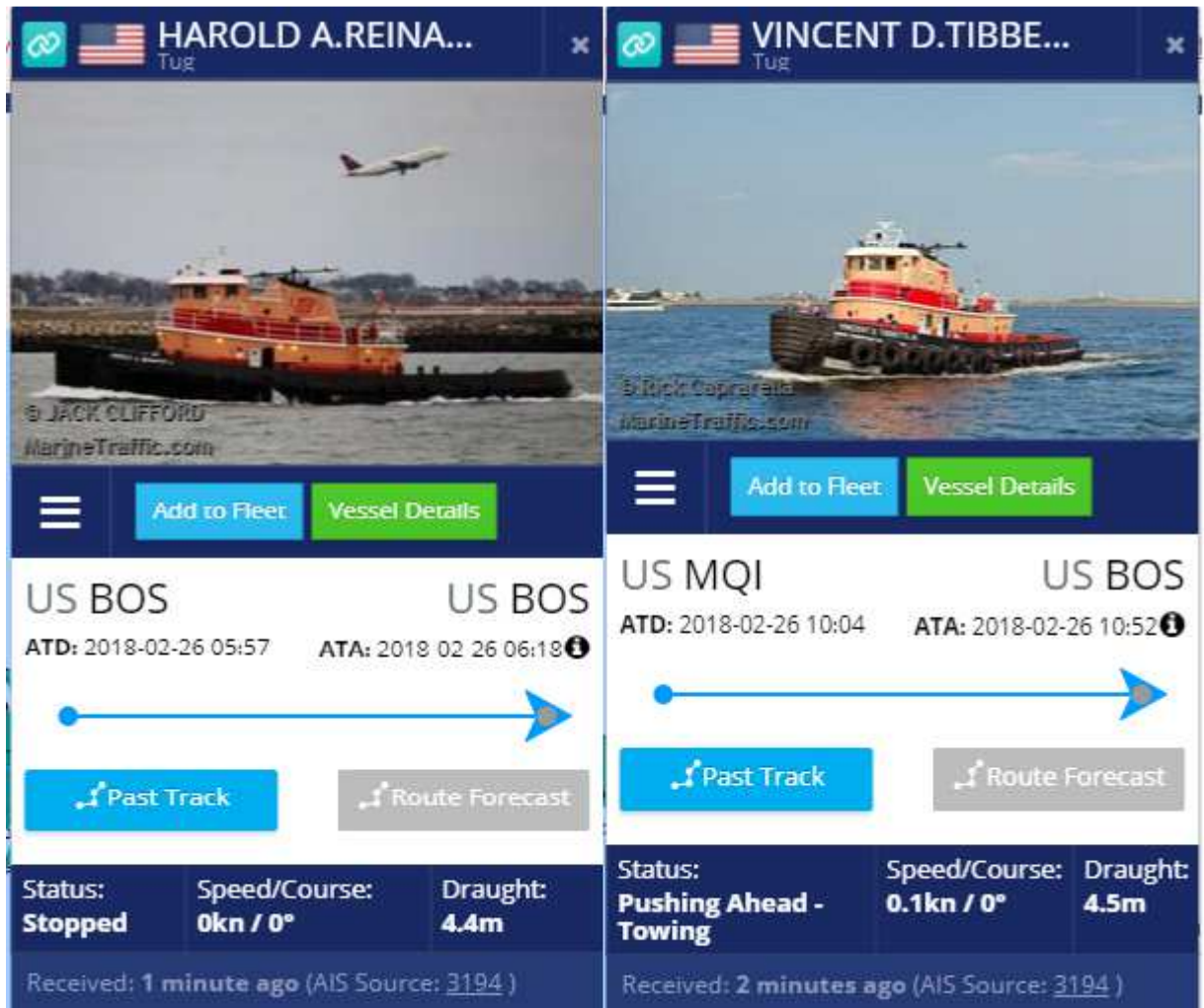


Рис. 3.26. Інформація про судна, сигнали яких записані у фреймі 2 (MMSI 367513020 та 367513050)

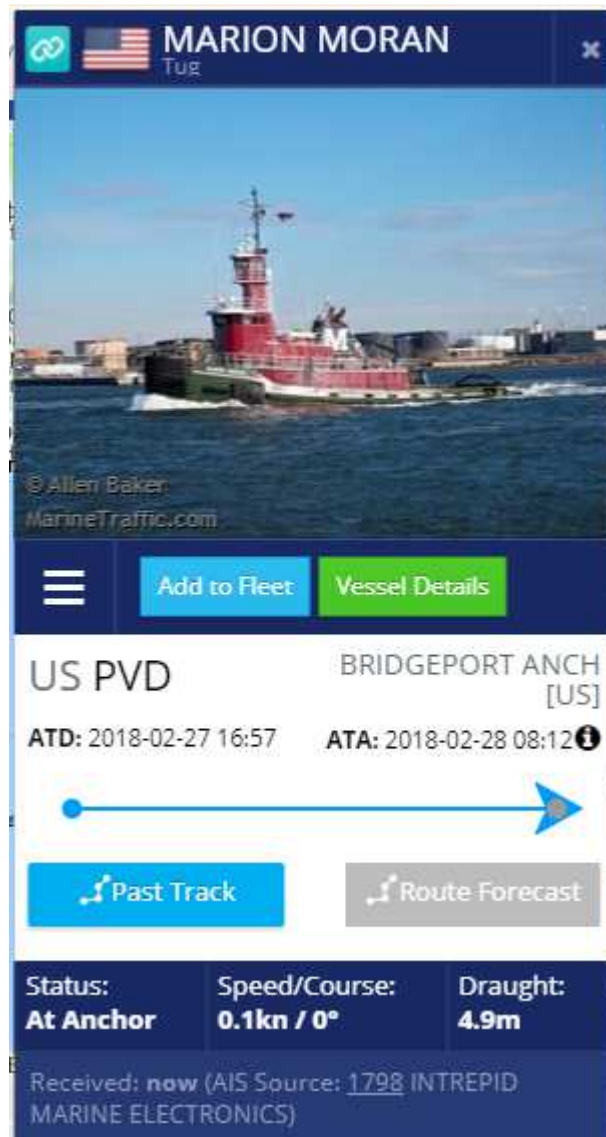


Рис. 3.27. Інформація про судно, сигнал якого записаний у фреймі 2 (MMSI 366941020)

На сайті також можна отримати інформацію на карті про останні переміщення судна.

Продовжимо опис і дослідження розглянутого алгоритму виділення сигналів. Розглянемо основну частину програми, яка завершується знаходженням синхроімпульсу:

```
% Після завдання параметрів на RTL-SDR чи відкриття файлу з
% квадратурною інформацією та розрахунком фази ідеального
% синхроімпульсу:
```

```
% Створення об'єкту для розрахунку коду CRC (16-бітний HDLS FCS)
crcGen = comm.CRCGenerator('Polynomial',
[1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1],...
'InitialConditions',1,'DirectMethod',true,'FinalXOR',1);
```

```

% Створення об'єкту для гаусової фільтрації
% (Gaussian Filter Design)
BT=.3;
pulseLength=3;
gx = gaussdesign(BT,pulseLength,samplesPerSymbol);

% Створення об'єкту для добування розгорнутої фази -
% еквівалентно unwrap((angle(z))
phaseCalc=dsp.PhaseExtractor;

% Цикл за кількістю оброблюваних фреймів (Loop for nCaptures)
for nn=1:nCaptures
    % Цикл до отримання некоректного повідомлення (Loop until
    % find invalid message)
    validCRC=1;
    % Отримання інформації від RTL-SDR чи з
    if liveFlag
        data = step(RX);
    else
        data=double(data);
    end
    while validCRC
        % Знаходження найбільш потужного сигналу у фреймі
        % розбиттям фрейму на сегменти по 128 відліків (Find
        % strongest RX signal in capture)
        windowLen = RX.SamplesPerFrame/128;
        l=floor(length(data)/windowLen);
        m=sum(reshape(abs(data(1:l*windowLen)),windowLen,l));
        d=diff(m);

        [mx,inx]=max(d(1:l-1));
        md=mean(abs(d(1:l-1)));
        inx = max(2,inx);
        inx = min(inx,l-2);

        % Пошук сегменту для подальшої обробки, що має
        % найбільший приріст сумарної амплітуди
        % відносно попереднього сегменту. Максимальний
        % приріст амплітуди сегменту повинен перевищувати середній
        % рівень приростів у 8 разів. Такий сегмент приймається
        % за початок сигналу
        % (Search the data set for a transition from low to high
        % power. The max needs to be 8x greater than the mean in
        % a segment to be a transition. Then trim the waveform
        % from the start of the transition until the end)
        if mx>8*md
            a1=d(inx)-d(inx-1);
            a2=d(inx)-d(inx+1);
            if a1>a2
                startPt = inx*windowLen;
            end
        end
    end
end

```

```

        else
            startPt = max(1, (inx-1)*windowLen);
        end
    else
        startPt = 1;
    end
% Обчислюється довжина вибірки із розрахунку прийому самого
% тривалого сигналу - 480 бітів
% (Trim waveform. Messages can be from 184 bits long
% (including CRC) to 440 bits long. There are also some
% ramp up bits, 24 sync bits and an 8 bit start
% flag. Capture enough samples to get longest message,
% which is the window length plus
% 480bits*samplesPerSymbol)
    endPt = min(length(data),
                startPt+windowLen+480*samplesPerSymbol);
    dataSlice=data(startPt:endPt);
    aisIdx=[startPt endPt];

    % Первинне (грубе) усунення зсуву по частоті
    % між приймачем та передавачем по спектру сигналу
    % (Coarse Frequency Correction)
    Y=abs(fftshift(fft(dataSlice)));
    idx=find(Y==max(Y));
    frShift=(floor(length(dataSlice)/2)-
            idx)*RX.SampleRate/length(Y);
    hc=comm.PhaseFrequencyOffset('FrequencyOffset',
    frShift,'SampleRate',RX.SampleRate);
    dataShifted=step(hc,dataSlice);

    % Уточнення меж сигналу
    % (Find Start and End Points)
    newStart=21;
    newEnd=length(dataShifted)-20;
    m=mean(abs(dataShifted));
    idx1=find(abs(dataShifted)>m,1,'first');
    idx2=find(abs(dataShifted)>m,1,'last');
    idxStart=max(newStart,idx1);
    idxEnd=min(newEnd,idx2);
    aisSig=dataShifted(idxStart:idxEnd);

    % GMSK Filter
    rxf = filter(gx,1,aisSig);

    % Усунення зсуву по частоті за вимірюванням фази.
    % Не реалізовано!
    % Fine Frequency Correction with
    % comm.FrequencySynchronizer (отсутствует в используемой
    % версии MATLAB)
    rxfShifted = rxf;

```

```

% Пошук синхроімпульсу по згортці з ідеальним
% (Find max correlation to preamble)
rxAngles = step(phaseCalc, rxfShifted);
syncCorr=zeros(length(rxAangles)-length(syncIdeal),1);
if (length(rxAangles) > samplesPerSymbol*50 +
    length(syncIdeal))
    for ii=1:samplesPerSymbol*50
        syncCorr(ii)=syncIdeal'*
            rxAngles(ii:ii+length(syncIdeal)-1);
    end
else
    syncCorr(1)=1;
end

% Прив'язка початку сигналу до кінця синхроімпульсу
% (Compute best sample phase for making bit decisions)
 [~,idx]=max(abs(syncCorr));
samplePhase =
    mod(idx, samplesPerSymbol)+floor(samplesPerSymbol/2);

```

Наведені коментарі розробників програми та їх вільний переклад дозволяють розібратися з алгоритмом обробки. Початок алгоритму сильно ускладнено. В алгоритмі обробки сигналів ADS-B кожна амплітуда, що перевищила середній рівень амплітуд фрейму у задану кількість разів, розглядається як можливий початок бортового сигналу, тобто сигнали виділяються послідовно у часі. Тут же запропонована сильно ускладнена обробка, спрямована на виділення сигналів, починаючи з найбільш потужних.

Весь кадр розбивається на 128 сегментів (windows). В якості критерію обрано не відношення амплітуди до середнього значення, а приріст амплітуди щодо попереднього сегмента. Результат обох видів обробки, безумовно, буде близьким, але будь-яке невинуватене ускладнення обробки може нести в собі непередбачувану поведінку в окремих випадках. При виборі інтервалу присутності сигналу AIS розробники виходять з максимально можливої довжини повідомлення, хоча на амплітудних розгортках межі сигналу дуже чіткі і можуть бути легко обчислені. Процедура уточнення меж сигналу у авторів є, але найважливіший блок усунення зсуву частоти між приймачем і передавачем, без якого неможливо виділення інформації з фази сигналу (див. алгоритми адаптації до зміщення по частоті для сигналів з фазовою модуляцією [8]), підключається до уточнення меж.

Оцінка зміщення по частоті у відповідності з класикою проводиться по спектру сигналу. Підключимо відображення спектрів і оцінімо частоту максимуму спектра (рис. 3.28–3.33). Для обчислення спектрів в програмі завжди використовується 13569 відліків, виділені сигнали AIS займають не більше 6144 відліків (разом з часом розгону). Нагадаємо, що смуга сигналу дорівнює $9600 \cdot 24 = 230400$ Гц.

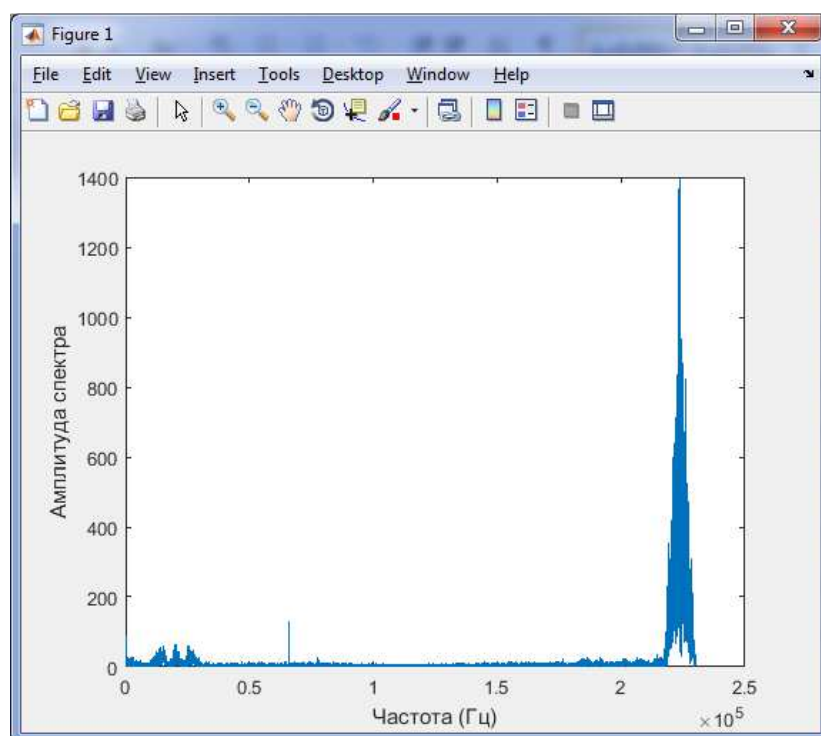


Рис. 3.28. Спектр сигналу від судна 367164250 (фрейм 1),
зміщення -6537 Гц

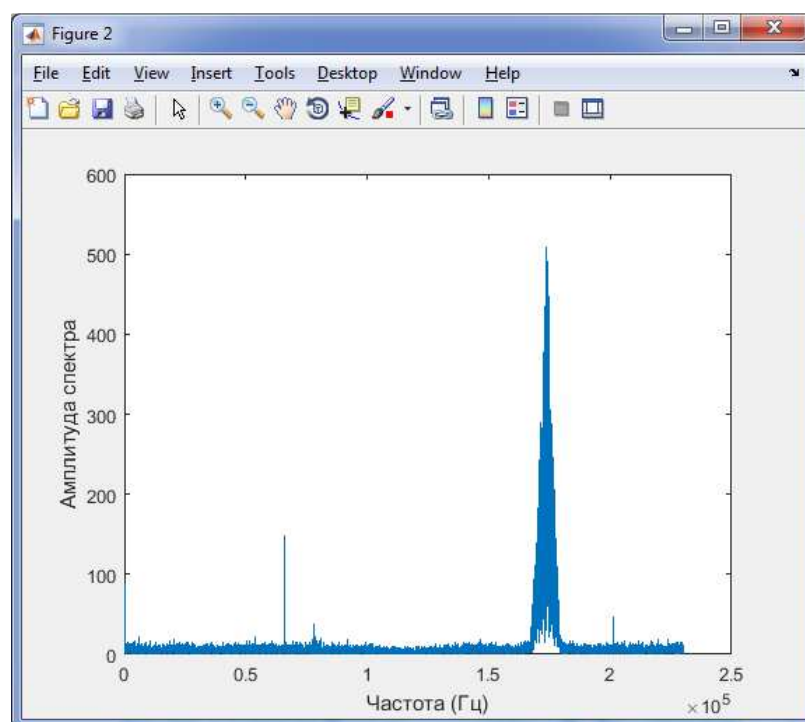


Рис. 3.29. Спектр сигналу від судна 367432220 (фрейм 1),
зміщення -56763 Гц

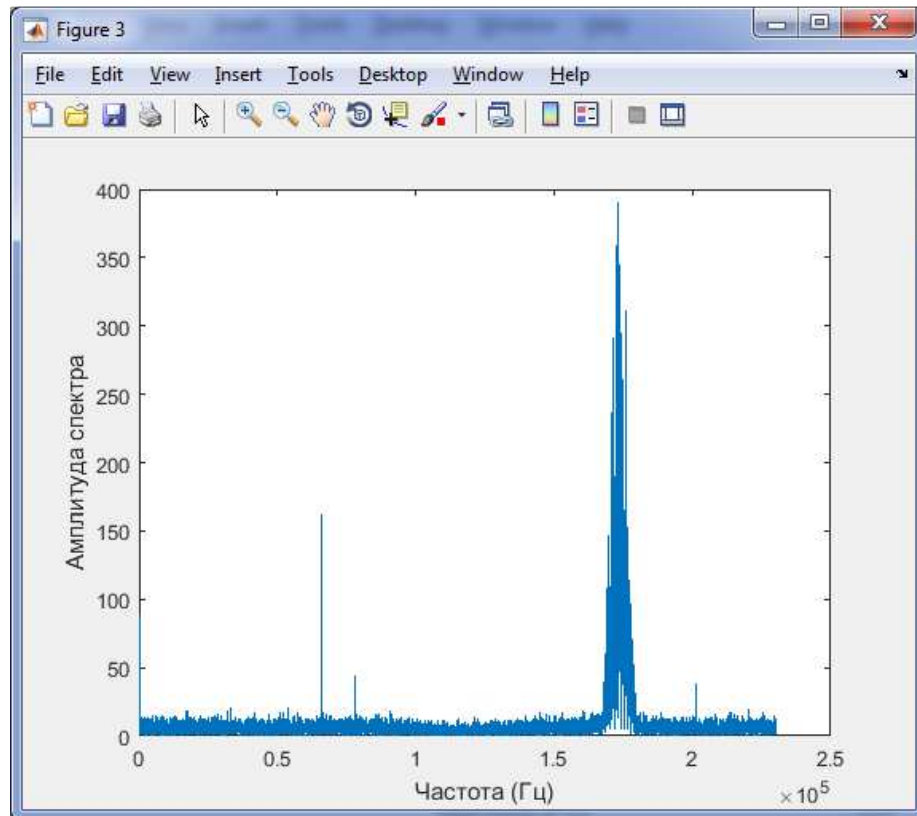


Рис. 3.30. Спектр сигналу від 3-го судна (фрейм 1, збій по CRC),
зміщення -57578 Гц

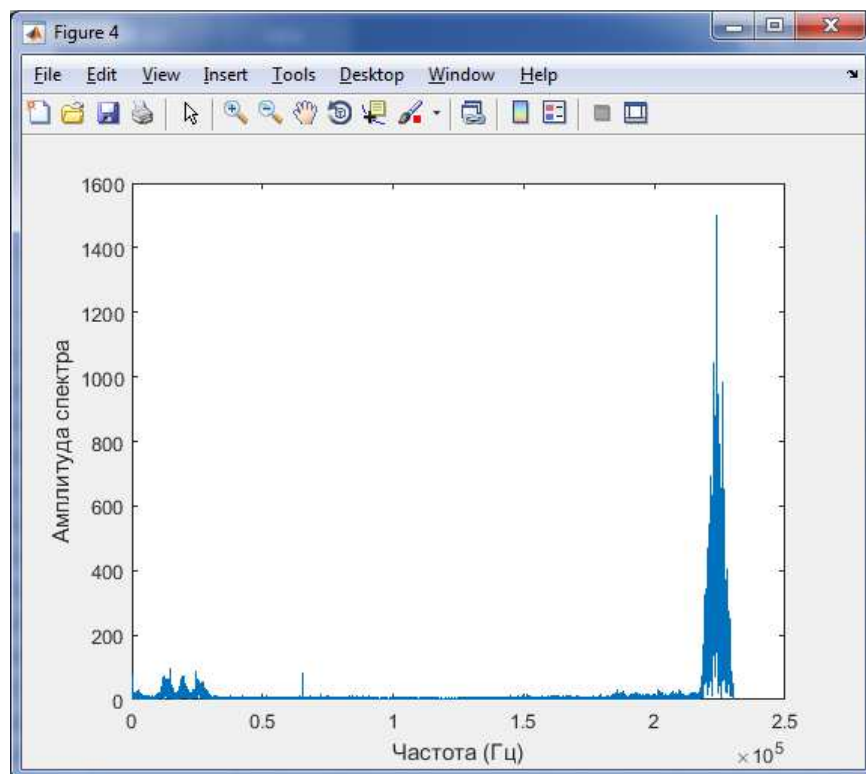


Рис. 3.31. Спектр сигналу від судна 367513020 (фрейм 2) ,
зміщення -6554 Гц

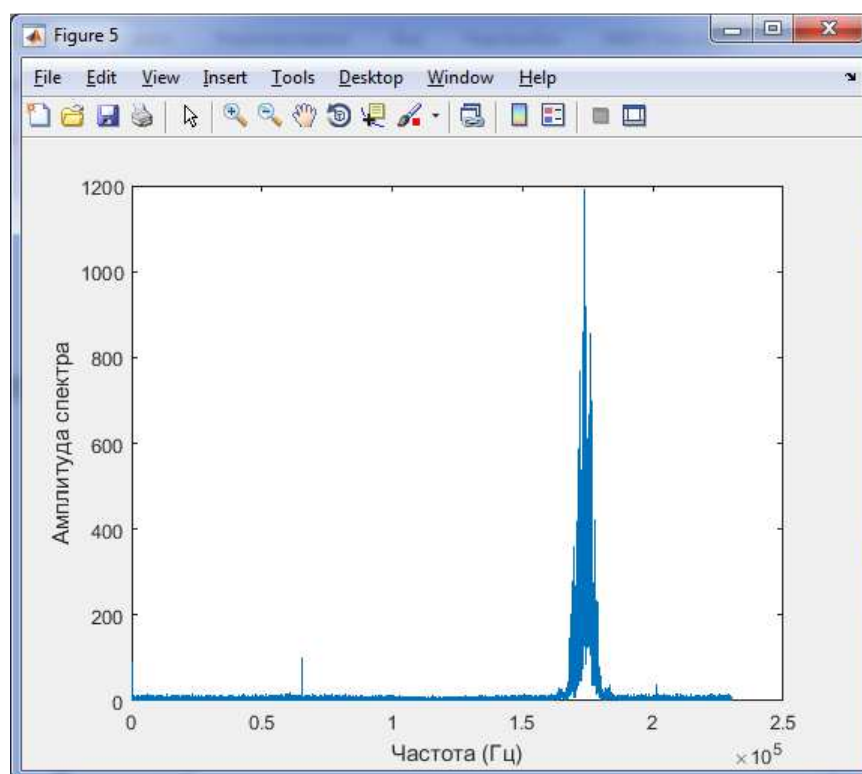


Рис. 3.32. Спектр сигналу від судна 367513050 (фрейм 2) ,
зміщення -56865 Гц

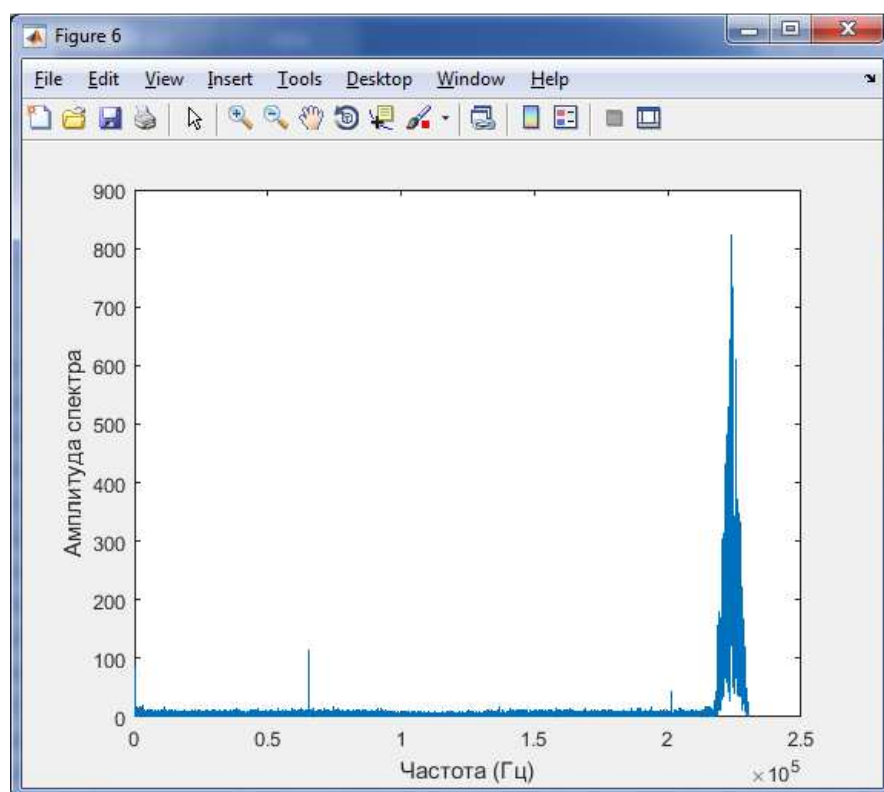


Рис. 3.33. Спектр сигналу від судна 366941020 (фрейм 2) ,
зміщення -6469 Гц

Спостерігаються значні зміщення частоти корабельних передавачів і приймача, які для деяких суден досягають майже **60 кГц**, і у всіх випадках спостерігається негативне зміщення. Судячи по коментарям, в програмі передбачався 2-ий рівень компенсації частотного зсуву по фазі сигналу за допомогою функції `comm.FrequencySynchronizer()` (схоже, її немає в усіх версіях MATLAB, є функція `comm.CarrierSynchronizer()`, але працює вона тільки з видами модуляції BPSK, QPSK, 8PSK, QAM або PAM). Якість компенсації зсуву розглянемо на розгортці фази виділених сигналів, після компенсації вимірюного зсуву, уточнення тривалості сигналу і гаусової фільтрації (рис. 3.34–3.39):

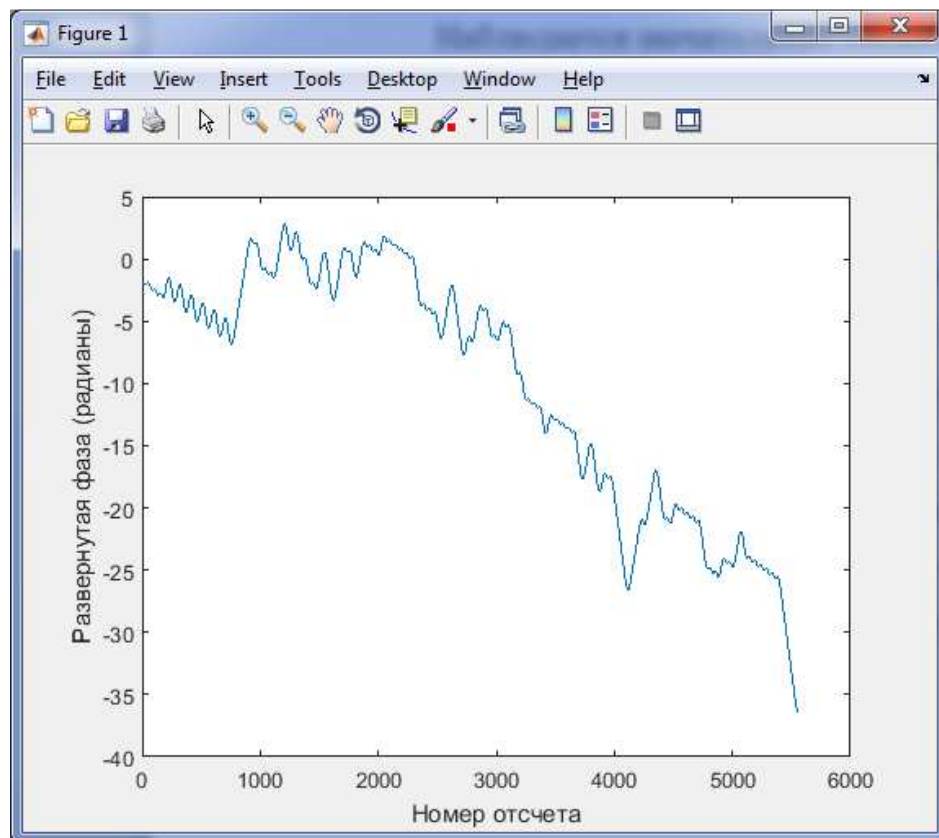


Рис. 3.34. Розгорнута фаза сигналу від судна 367164250 (фрейм 1)

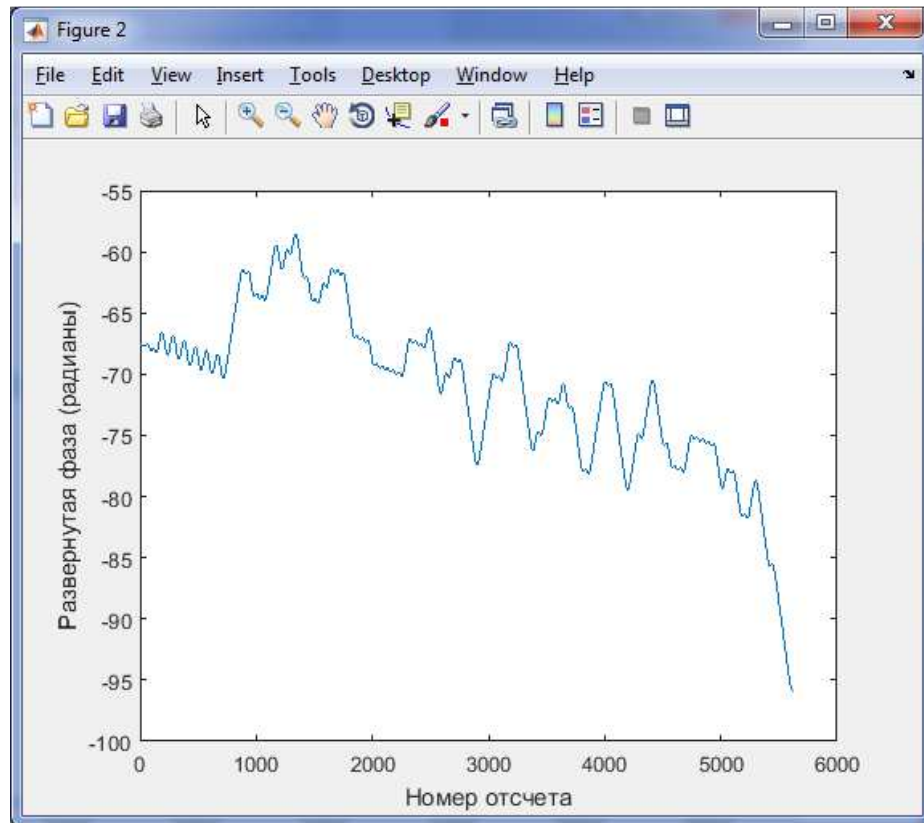


Рис. 3.35. Розгорнута фаза сигналу від судна 367432220 (фрейм 1)

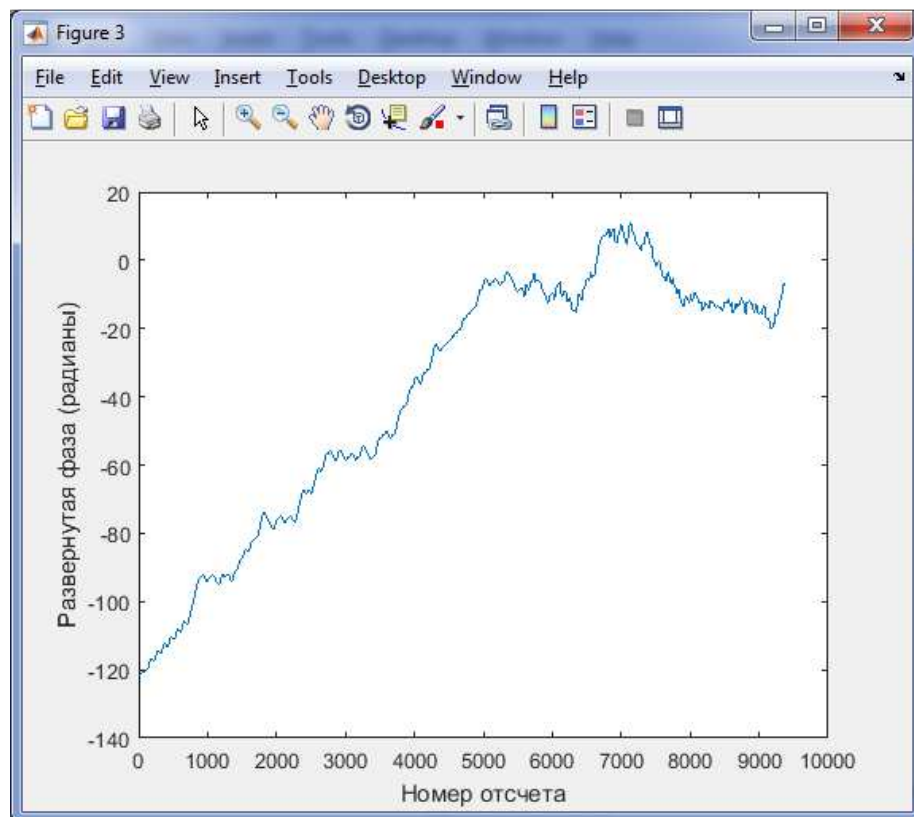


Рис. 3.36. Розгорнута фаза сигналу від 3-го судна (фрейм 1, збій по CRC)

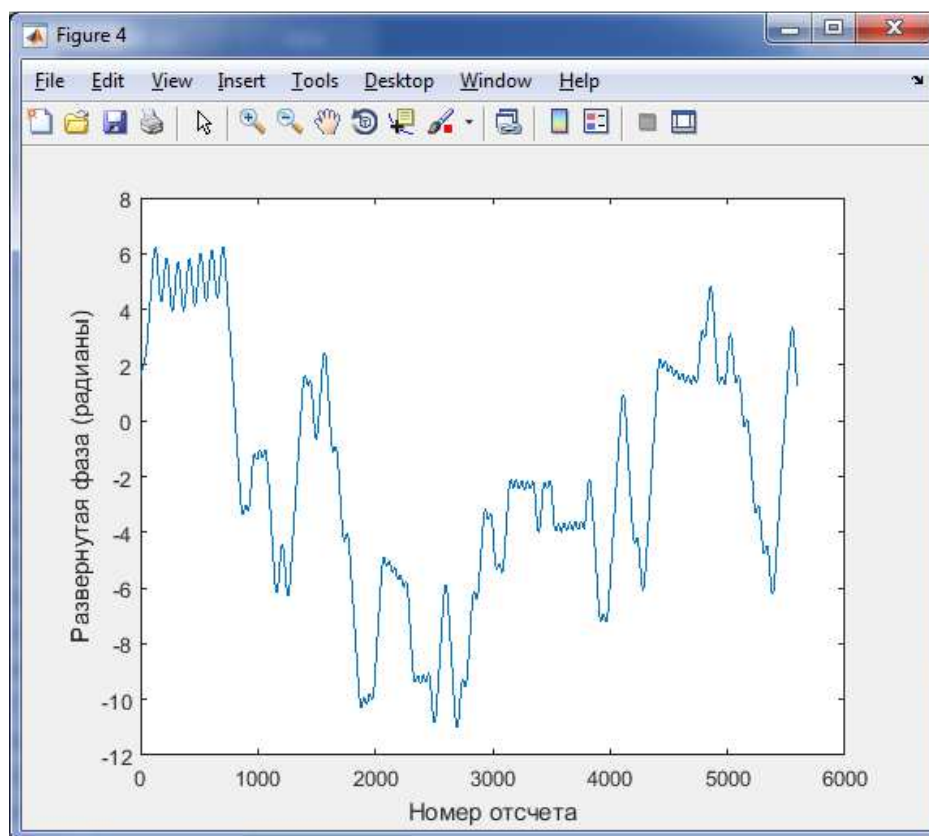


Рис. 3.37. Розгорнута фаза сигналу від судна 367513020 (фрейм 2)

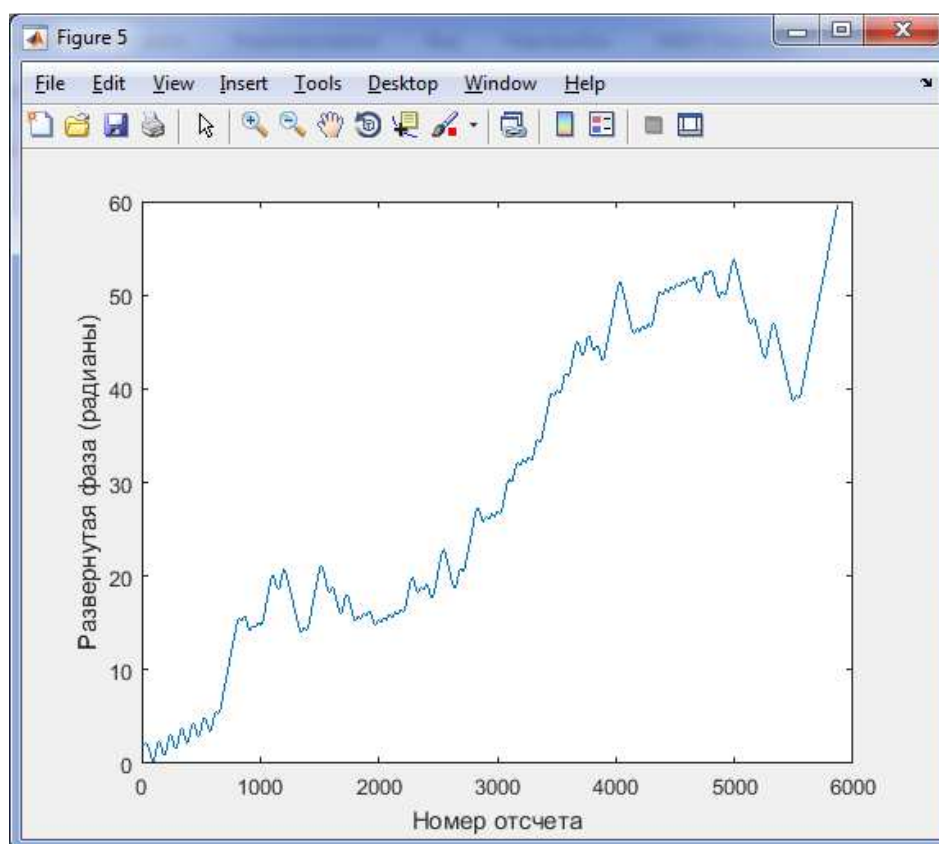


Рис. 3.38. Розгорнута фаза сигналу від судна 367513050 (фрейм 2)

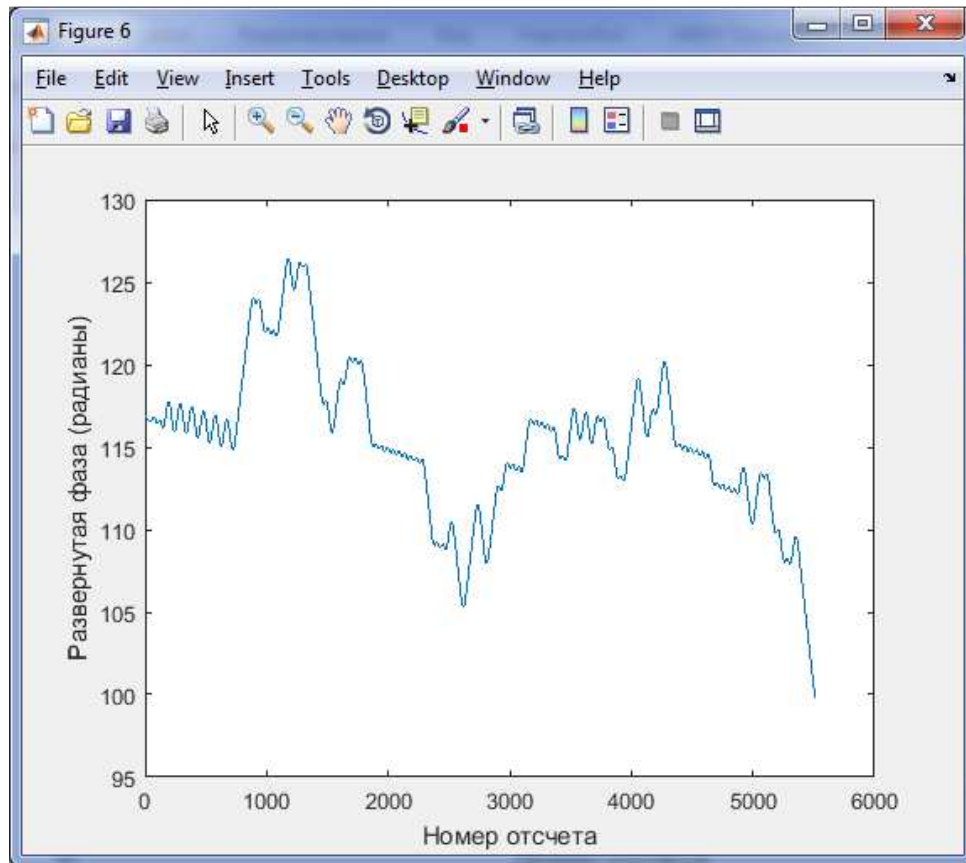


Рис. 3.39. Розгорнута фаза сигналу від судна 366941020 (фрейм 2)

На всіх рисунках добре видно знайомий нам, практично ідеальний, синусоїдальний синхроімпульс, виняток становить перший сигнал фрейму 2, де спостерігаються невелике спотворення (рис. 3.37). Тільки для цього сигналу вдалося майже повністю компенсувати зсув по частоті. Для всіх інших сигналів добре видна лінійна зміна фази на синхроімпульсі та інших частинах сигналу без інформаційних стрибків фази, що говорить про неповну компенсацію зміщення, яке і призвело до низьких значень коефіцієнтів кореляції (див. результати обробки фреймів штатною програмою). Спостерігається залишковий зсув по частоті як позитивний (фаза зростає), так і негативний (фаза убыває). Найбільш невдало виглядає фаза невиявленого сигналу (третій сигнал фрейму 1, рис. 3.36). Тут і інтервал існування сигналу завищений майже в 2 рази і залишкове зміщення вище.

Отримані результати змушують нас зайнятися усуненням виявлених недоліків штатної програми. Насамперед, введемо уточнення меж сигналу до побудови спектру і компенсації зсуву, оскільки побудова спектру з захопленням зони відсутності сигналу, яка перевершує по ширині зону сигналу, може призводити до помітних спотворень спектру:

```
% Уточнення меж сигналу
dataAbs=abs(dataSlice);
c=2.0;
p=c*mean(dataAbs);
```

```
idxStart=find(dataAbs > p,1,'first');
idxEnd=find(dataAbs > p,1,'last');
dataSlice=dataSlice(idxStart:idxEnd);
```

Блок виглядає прозоро і лаконічно при використанні потужної і корисної функції **find()**, яка може знаходити індекси в масиві, амплітуди яких задовольняють заданим умовам. В даному випадку ми шукаємо перший і останній відліки, амплітуди яких перевищують середній рівень в 2 рази. Наприклад, для невиявленого сигналу графічно це виглядає так (рис. 3.40):

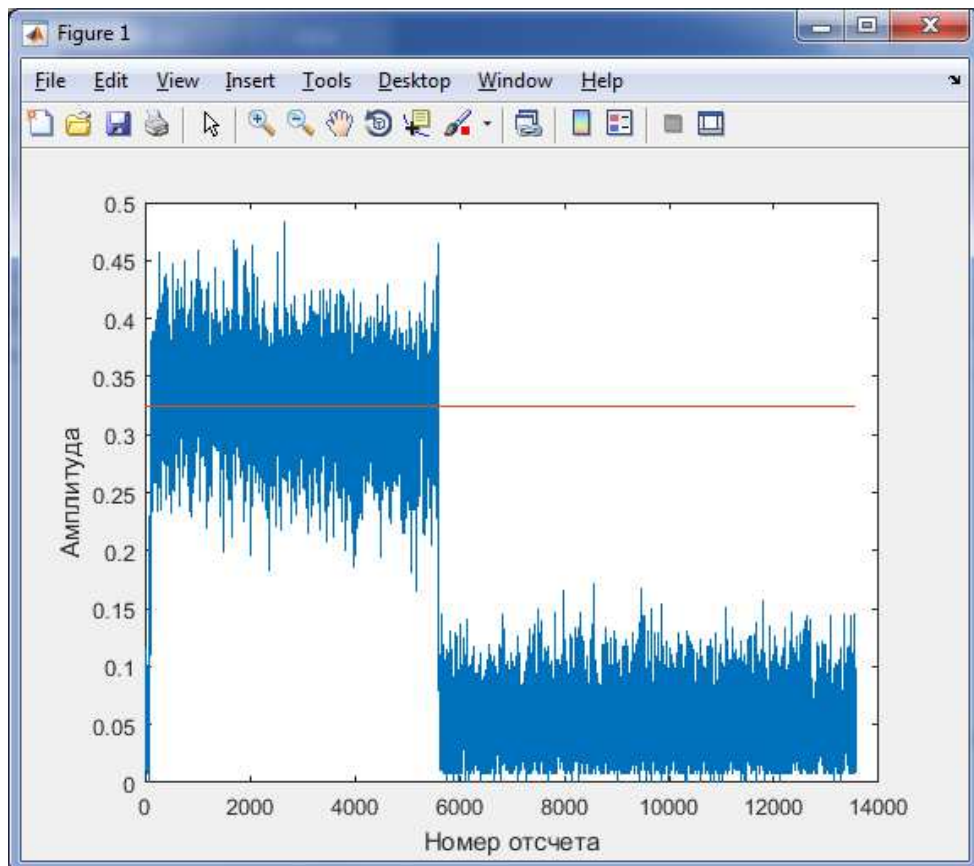


Рис. 3.40. Схема уточнення меж сигналу на прикладі невиявленого сигналу

На рисунку вся розгортка відповідає інтервалу побудови спектру штатної програмою, червона лінія – поріг, за яким оцінюється довжина сигналу. Добре видно, що межі сигналу оцінюються точно. Виключення з інтервалу побудови спектру сигналу задовгих відліків дозволяє трохи краще оцінити частотний зсув сигналу, проте цього не достатньо. Модуляція сигналу інформаційними бітами призводить до значних випадкових флуктуацій максимуму спектра. Якщо для сигналів з фазовою модуляцією та амплітудно-фазовою модуляцією (PSK, PAM, QAM) вдається легко звільнитися від модуляції фази шляхом зведення сигналу в ступінь, що відповідає розміру алфавіту символів [8], то для GMSK-сигналів ця процедура марна. Мабуть, можна трохи поліпшити оцінку зміщення, проводячи її не за одним, а декількома відліками. У

будь-якому випадку очевидна необхідність повторної оцінки зсуву безпосередньо по фазі сигналу. Графіки розгорнутої фази сигналів (рис. 3.34–3.39) наштотують, принаймні, на два способи вимірювання частотного зсуву. Перший спосіб – провести пряму методом найменших квадратів (МНК) на інтервалах незначної флуктуації фази. Візуальний вибір таких інтервалів досить очевидний і призведе до гарних результатів, реалізація алгоритму потребує ускладнення логіки алгоритму. Другий спосіб – провести пряму по всім відлікам і потім, виключивши відліки фази, далеко віддалені від проведеної прямої, повторити МНК. Наведемо реалізацію і результати роботи другого способу:

```
function df = FineFreqSynchr1(ph, SampleRate)
% Функція для оцінки зсуву частоти по фазі для GMSK-сигналів.
% Проводиться лінійний МНК на всьому інтервалі,
% потім повторний МНК
% с виключенням відліків віддалених від отриманої прямої на
% задану величину
% ph - масив фаз після видалення биття в околиці
% плюс(мінус)Pi (розгорнута фаза)
% SampleRate - частота квантування в герцах
% df - зсув в герцах

dph=4; % поріг при виключенні відліків при повторному МНК

% Приведення середньої фази до нуля:
ph=ph-mean(ph);

Lph=length(ph);
dt=1/SampleRate;
t=(0:dt:dt*(Lph-1))';

p=polyfit(t,ph,1); % лінійний МНК
phLS=p(1)*t+p(2); % отримана пряма

% Повторний МНК з виключенням видалених значень фаз:
Ind=find(abs(ph-phLS) <= dph);
p=polyfit(t(Ind),ph(Ind),1); % лінійний МНК
df=p(1)/(2*pi); % оцінка зсуву частоти у Гц
```

Знову ж завдяки функції **find()** реалізація повторного МНК вийшла вельми лаконічною. На рис. 3.41 наведено результат обробки невиявленого сигналу з фрейму 1:

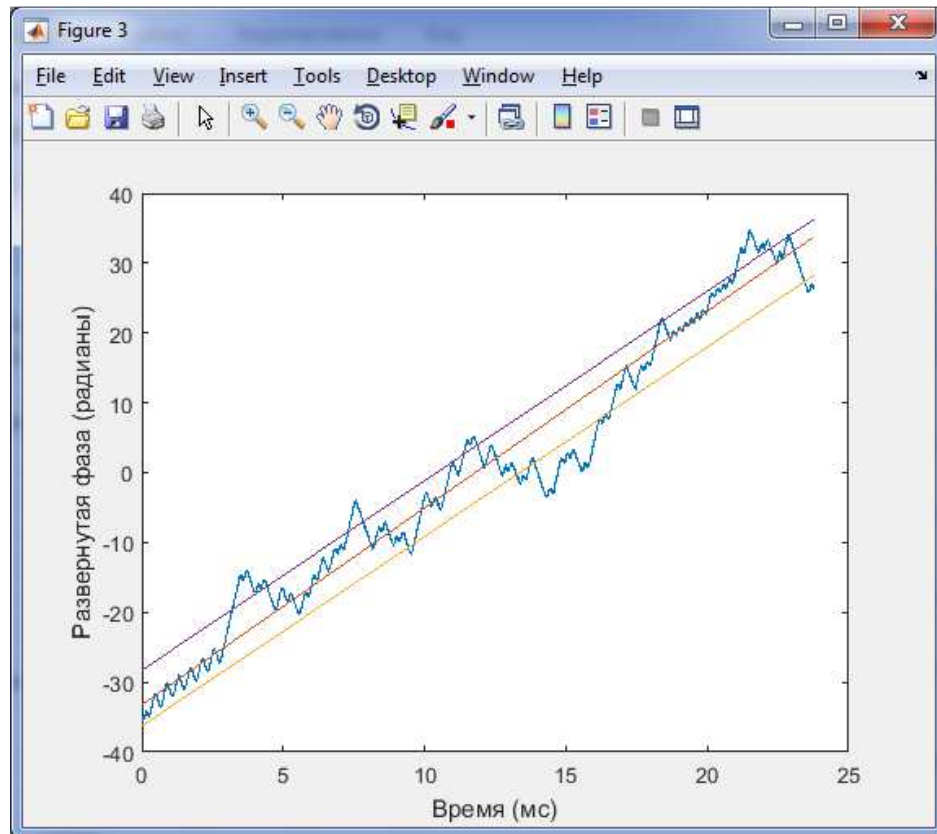


Рис. 3.41. Графічна інтерпретація оцінки частотного зсуву по фазі сигналу від 3-го судна (фрейм 1, збій по CRC)

По-перше, ми бачимо, що оцінка зсуву спектру сигналу спрацювала краще. На рис. 3.36 на тривалості сигналу фаза змінюється на ~ 120 радіан, тут це ~ 60 радіан. Синхроімпульс і сам сигнал тепер виглядають нітрохи не гірше інших 5 виявлених сигналів. Розгорнута фаза того ж сигналу після усунення виміряного частотного зсуву виглядає так (рис. 3.42):

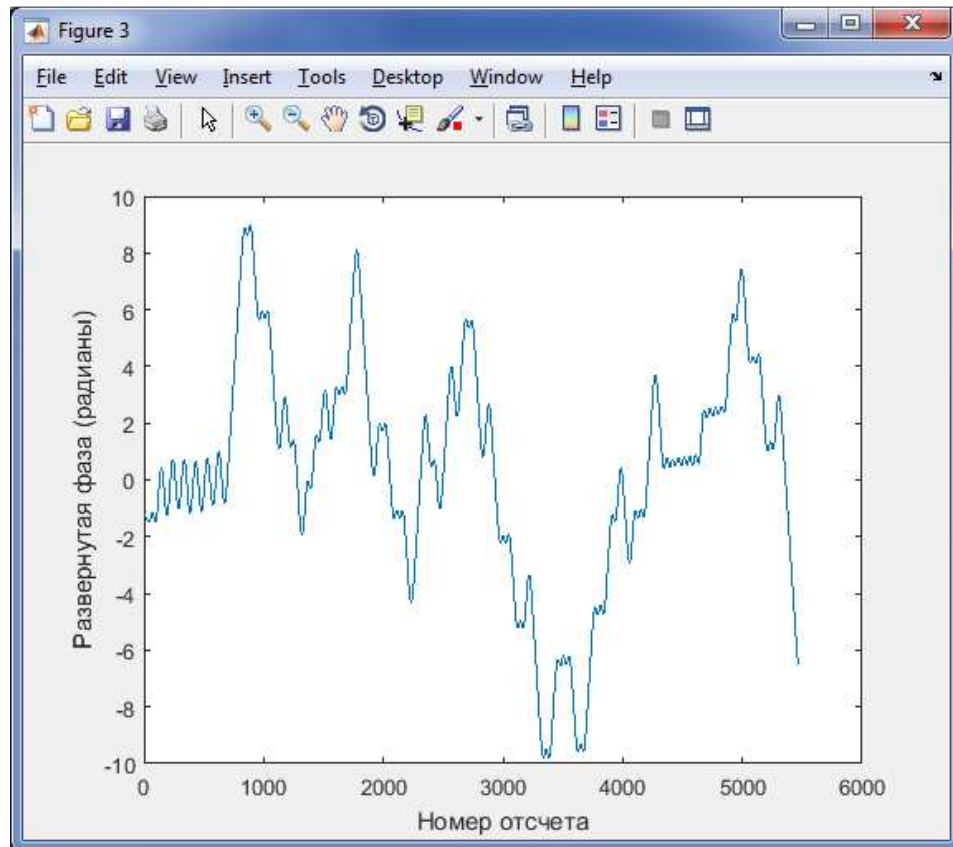


Рис. 3.42. Розгорнута фаза сигналу від 3-го судна після остаточного усунення частотного зсуву (фрейм 1, збій по CRC)

Ясно, що такий сигнал буде успішно декодований. Дійсно за цим сигналом ми отримуємо наступне повідомлення:

```
corr=0.979, msgType = 3
MMSI:367367470 Latitude:42.3597 Longitude:-71.0464
0C579654B81FC35EBAC675183D06E3302DDC0020C0348C
```

Коефіцієнт кореляції синхроімпульсу перевищив раніше зафіксовані коефіцієнти кореляції по всім п'яти іншим виявленим сигналам і впритул наблизився до 1. За виданим ідентифікатором MMSI на сайті www.marinetraffic.com дізнаємося інформацію про судно (рис. 3.43):

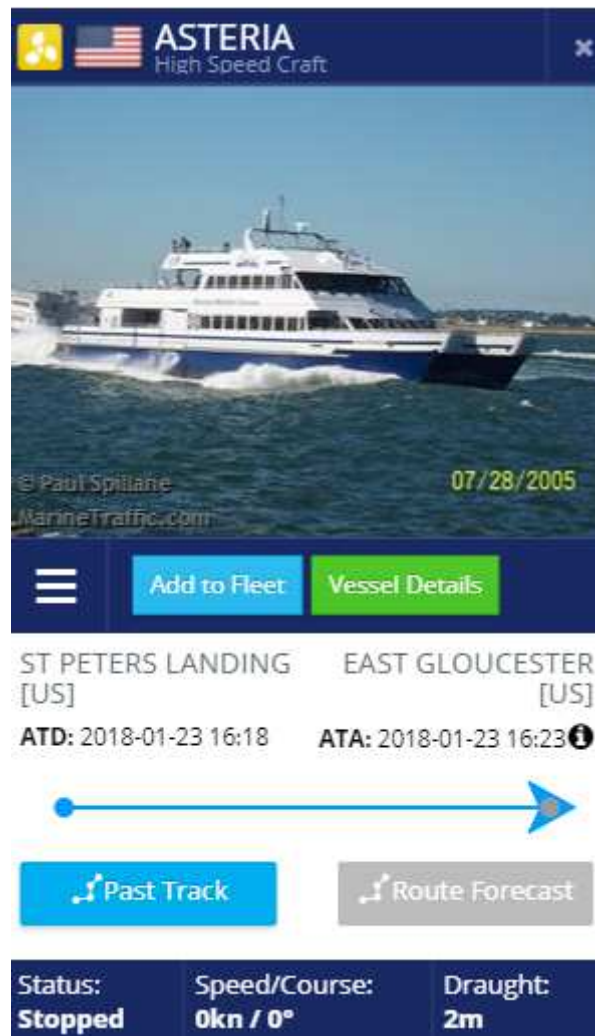


Рис. 3.43. Дані про судно, сигнал якого не виявлений штатною програмою

Це теж судно класу А, номер повідомлення, на відміну від 5 інших суден, третій.

Процедура оцінки та усунення частотного зсуву без використання функцій `fftshift()`, `comm.PhaseFrequencyOffset()`, стає більш прозорою:

```
df = FineFreqSynchr1(rxAangles, SampleRate);
omeg = -2*pi*df;
dt = 1/9600; % інтервал передавання одного символу (с)
dtf = dt/samplesPerSymbol; % інтервал дискретизації (с)
N1 = length(rxAangles);
t = (0:dtf:dtf*(N1-1))';
rxShifted = dataShifted.*exp(1i*omeg*t);
```

Далі йде гаусова фільтрація:

```
% GMSK Filter
rxShifted = filter(gx, 1, rxShifted);
```

Необхідно особливо відзначити, що застосування такого фільтра до сигналів з великими частотними зміщеннями руйнує фазову структуру сигналу. Використання його до оцінки зсуву за спектром сигналу для досліджуваних сигналів повністю їх руйнує (не одне повідомлення не виділяється), використання перед другою ступінню оцінки зсуву незначно зменшує кореляцію з синхроімпульсом.

Розглянемо тепер блок пошуку синхроімпульсу штатної програми. До нього теж є серйозні зауваження. Згідно з наведеним текстом прийняття рішення здійснюється не за коефіцієнтом кореляції, а по згортці, але це не завжди еквівалентно. На рис. 3.44 наведено залежності значення ковзної згортки і коефіцієнта кореляції ідеального синхроімпульсу з реалізаціями розгорнутої фази для одного з виявлених сигналів (залежності приведені до одного масштабу діленням згортки на коефіцієнт 15):

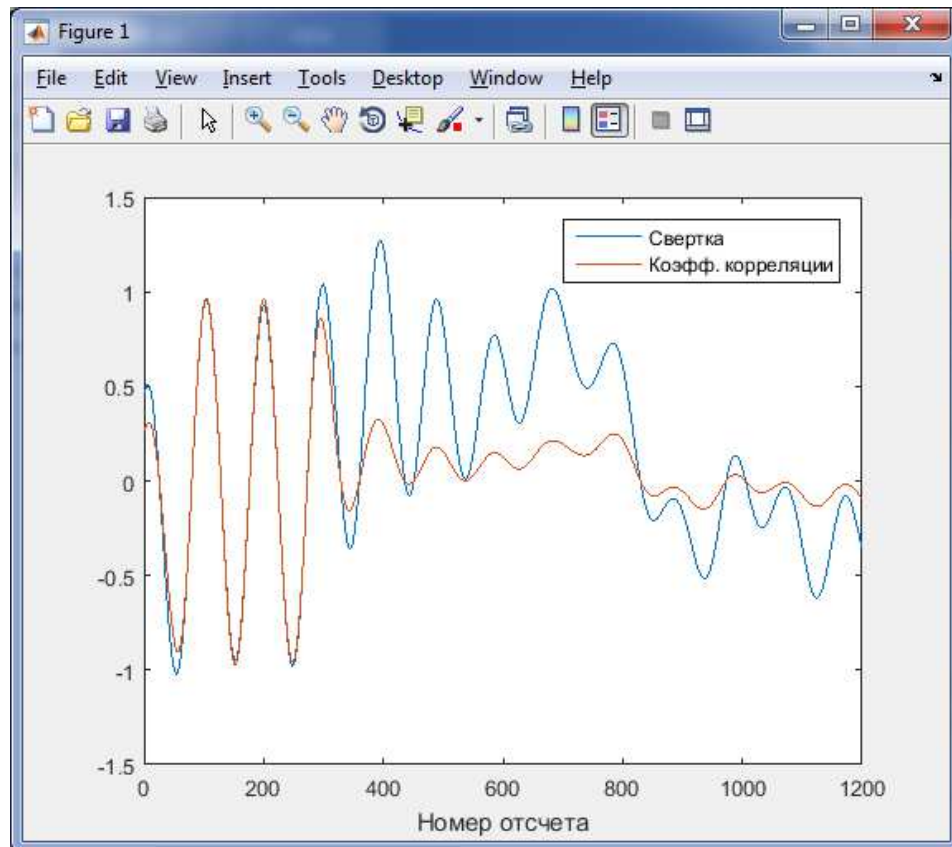


Рис. 3.44. Порівняння результатів згортки і коефіцієнта кореляції для судна 367513020 (1-й сигнал фрейму 2)

Як бачимо, відліки, що відповідають максимальним значенням згортки і коефіцієнта кореляції можуть сильно відрізнятися. Ще більш важливо, що для використовуваного в AIS синусоїдального синхроімпульсу поведінка обох статистик теж має синусоїдальний характер, тому прив'язка до інформаційної частини сигналу тільки за синхроімпульсом дуже часто може приводити до помилок декодування. В блоці пошуку синхроімпульсу є ще помилка, мабуть, механічна — пошук максимуму проводиться за модулем згорт-

ки, тобто у деяких випадках можлива прив'язка до відліку, щодо якої реальний і ідеальний синхроімпульси знаходяться в протифазі. Не дуже зрозуміла також прив'язка до синхроімпульсу тільки всередині кордонів символу (від відліку з максимальною кореляцією обчислюється модуль за кількістю відліків на символі):

```
[~, idx]=max(syncCorr);
samplePhase=mod(idx, samplesPerSymbol)+floor(samplesPerSymbol/2);
```

Видається, що абсолютна прив'язка краща:

```
samplePhase=idx + samplesPerSymbol/2;
```

Зазначимо, що використання коефіцієнта кореляції, оскільки це нормована величина, дозволяє виставляти поріг для подальшого продовження обробки, тим самим зменшуючи час обробки фрейму.

Розглянемо решту операцій програми до отримання всього повідомлення в двійковому вигляді:

```
% Рішення щодо прийнятих бітів - зміну фази більш pi/4
% приймаємо за «1» (Make bit decisions - phase change greater
% than pi/4 is logical 1)
abits=zeros(size(rxfShifted(samplePhase:samplesPerSymbol:end)));
idx=find(abs(diff(step(phaseCalc,
rxfShifted(samplePhase:samplesPerSymbol:end))))>pi/4);
abits(idx)=1;

sb=1; % Номер першого біта після StartByte
% Серед перших 50 бітів шукаємо StartByte flag
% (Search the first 50 bits for the StartByte flag - 0x7E)
if length(abits)>56
    for ii=2:50
        if (sum(abits(ii:ii+5))==6 && abits(ii-1)==0 &&
            abits(ii+6)==0 && sb==1)
            sb=ii+7;
        end
    end
end

% Читання номеру повідомлення та відправлення бітів у відповідну
% функцію декодування (Read the message type and route the
% bits to the correct decode function)
msgType=0;
if length(abits) >= sb+7
    msgType=(2.^(0:5)*abits(sb+2:sb+7));
end
fprintf('msgType = %d\n',msgType);

% Дотримуючись специфікацій AIS видаляємо «0» після 5
% послідовних «1». Це змінить також End flag
```

```

% (Follow AIS spec to unstuff after 5 consecutive 1's. This
% will unstuff everything, including the end flag in the AIS
% message)
ubits=aisUnstuff(ubits(sb:end));

% Декодування повідомлення у відповідності з його номером.
% Перевірка % коду CRC, перевертання байтів и декодування
% повідомлення. (Decode
% message based on detected message type. Compute the checksum,
% compare to the received message bits, and if the checksum
% passes flip the bytes and decode the message.
switch msgType
    case 1
        if length(ubits)>=184
            checksum = step(crcGen,ubits(1:168));
            if isequal(checksum(169:184),ubits(169:184))
                flippedData=aisFlipBytes(ubits(1:184));
                cs=aisDecodeMsg1(flippedData);
                validCRC=1;
            else
                validCRC=0;
            end
        else
            validCRC=0;
        end
    case 2,3,4,5,18,19,21
end
% Видалення обробленого сигналу з початкових квадратур фрейму
% (Remove signal from data set)
data=[data(1:aisIdx(1));data(aisIdx(2):end)];

```

Як бачимо, формування бітового масиву за отриманою розгорнутою фазою сигналу досить просте. Готується нульовий масив необхідної довжини (кількість відліків фази поділена на 24), починаючи від індексу прив'язки до синхроімпульсу (**samplePhase**) з кроком 24 вибирається по одному середньому відліку на кожному символі, і якщо фаза змінилася більше ніж на $\pi/4$, в бітовий масив записується 1. Має сенс дослідити варіанти прийняття рішення по сумі або максимуму серед декількох середніх відліків символу.

Таким чином, ми дійшли до місця алгоритму, в якому буде задіяна паличка-виручалочка технології AIS, а саме до синхронізації бітового масиву по **Start Byte**. Це дозволяє успішно нівелювати огріхи прив'язки за синхроімпульсом. Значення Start Byte дорівнює **0x7E**, що в двійковій системі дорівнює **01111110**, комбінація симетрична щодо молодших і старших бітів байту. Тому без перевертання байт можна шукати **Start Byte** (нагадаємо, що в AIS байт передається, починаючи з молодших бітів). Якщо б він не був симетричний, то виникли б проблеми, оскільки на цьому етапі межі байтів невідомі.

При знаходженні **Start Byte** формується параметр **sb**, що рівний номеру першого біта після **Start Byte**. Наведемо результати пошуку синхроімпульсу і **Start Byte** для штатної і доопрацьованої програм (таблиця 3.3):

Таблиця 3.3

№ фрейму	№ сигналу	Коефіцієнт кореляції	Sb	Коефіцієнт кореляції	Sb
		Штатна програма		Доопрацьована програма	
1	1	0,649	39	0,990	27
	2	0,788	37	0,992	27
	3	Збій CRC	-	0,979	27
2	1	0,319	37	0,966	33
	2	0,499	35	0,866	30
	3	0,889	38	0,965	31

Після доопрацювань коефіцієнт кореляції реального та ідеального синхроімпульсів вийшов на значення, близькі до 1 (не менше 0,97 для всіх сигналів, крім 2-го сигналу у другому фреймі, де значення склало 0,87). Різні значення прив'язки до **Start Byte** (параметр **sb**) говорять про прив'язку до синхроімпульсів зі зміщенням. Тільки для першого фрейму видані доопрацьованою програмою однакові значення **sb**. Таким чином, дійсно, без прив'язки до **Start Byte**, відсоток правильно декодованих повідомлень, навіть для потужних сигналів, значно скоротився б. Дуже можливо, що використання неперіодичного синхроімпульсу дозволило б більш точно прив'язуватися до нього, однак з урахуванням прив'язки до **Start Byte** загальні характеристики декодування навряд чи б помітно покращилися.

Зазначимо, що за наявними записами сигналів AIS потреба в двоступінчастому усунення зміщення по частоті приймача і передавача відсутня. Відключення первинної оцінки зсуву за спектром сигналу з точністю до трьох знаків після коми призводить до тих же коефіцієнтів кореляції, які наведені в таблиці 3.3. У таблиці 3.4 наведено зміщення розглянутих сигналів судів по частоті до і після їх корекції спектральним методом (вказані так само на рис. 3.28-3.33), які успішно усуваються фазовим методом:

Таблиця 3.4

№ фрейму	№ сигналу	Зсув після спектрального методу (Гц)	Зсув без спектрального методу (Гц)
1	1	-580	-6537
	2	-209	-56763
	3	449	-57578
2	1	16	-6554
	2	-8	-56865
	3	-115	-6469

Як бачимо, фазовий метод цілком справляється з частотним зсувом, принаймні, до 57 кГц.

Після прив'язки до **Start Byte** визначається номер повідомлення. Зверніть увагу, що переклад номера повідомлення у десятковий вид здійснюється з перевертанням бітів. Далі сформовані біти, починаючи з першого біта після **Start Byte**, записуються в інший масив з виключенням усіх «0» наступних після п'яти «1», що йдуть поспіль.

Нагадаємо, що при формуванні повідомлення для запобігання утворення хибних прапорів, після кожних 5-ти послідовних «1» вставляється 0. Далі за типом повідомлення визначається довжина повідомлення і перевіряється код CRC, у разі збігу, в кожному байті перевертаються молодші і старші біти і одержаний потік бітів надходить на отримання параметрів судна. Звідси випливає, що при формуванні повідомлення перевертання бітів відбувається до формування коду CRC, а зайві «0» додаються пізніше. Як вже зазначалося вище, розробники програми обмежилися тільки друком основних параметрів судна.

Завершимо розділ дослідженням можливостей виявлення більш слабких сигналів AIS, ніж записи наявних сигналів. Оскільки поки що відсутні записи слабких сигналів на фоні реальних перешкод, будемо зменшувати співвідношення сигнал/перешкода додаванням до реальної завади добре відомого нам нормального білого шуму.

Перш за все, проведемо вимірювання співвідношення сигнал/перешкода за наявними записами. Для цього вставимо в доопрацьовану програму, відразу після уточнення меж сигналу, блок вимірювання співвідношення сигнал/перешкода:

```
% Вимірювання відношення с/п:
didx=idxEnd-idxStart+1;
Ps=sum(abs(dataSlice(idxStart:idxEnd).^2))/didx;
Lds=length(dataSlice);
idxNS1=max(1,idxStart-didx);
idxNE1=max(1,idxStart-1);
idxNS2=min(Lds,idxEnd+1);
idxNE2=min(Lds,idxEnd+didx);
Pn1=sum(abs(dataSlice(idxNS1:idxNE1).^2)/
        (idxNE1-idxNS1+1));
Pn2=sum(abs(dataSlice(idxNS2:idxNE2).^2)/
        (idxNE2-idxNS2+1));
qizm=10*log10(2*Ps/(Pn1+Pn2));
fprintf('qizm=%.2f\n',qizm);
```

Блок вимірює потужність сигналу (квадрат амплітуди) на обчисленому інтервалі існування сигналу, потужність перешкоди оцінюється на двох інтервалах такої ж довжини праворуч і ліворуч від сигналу. Обчислення ведуться з масиву квадратур до звуження сигналу за уточненими межами. Оскільки можливо звуження інтервалів оцінки перешкод довжиною обраного масиву, введена відповідна процедура обчислення індексів, що не дозволяє їм вийти за межі масиву. В результаті роботи блоку отримані наступні значення співвідношення сигнал/перешкода для розглянутих сигналів (таблиця 3.5):

Таблиця 3.5

№ фрейму	№ сигналу	Співвідношення сигнал/перешкода (дБ)
1	1	25,5
	2	17,3
	3	10,5
2	1	19,5
	2	15,7
	3	10,1

Наведені значення в цілому відповідають тому, що ми бачимо на рис. 3.23 і 3.24, якщо врахувати, що при нанесенні великих масивів на графік відліки зливаються, і ми бачимо максимальний рівень відліків, а не середній. Наприклад, з рисунку 3.23 середній рівень перешкод візуально оцінюється приблизно як 0,15. Нанесені на графік 100 завадових відліків амплітуди в околі найбільш потужного сигналу виглядають так:

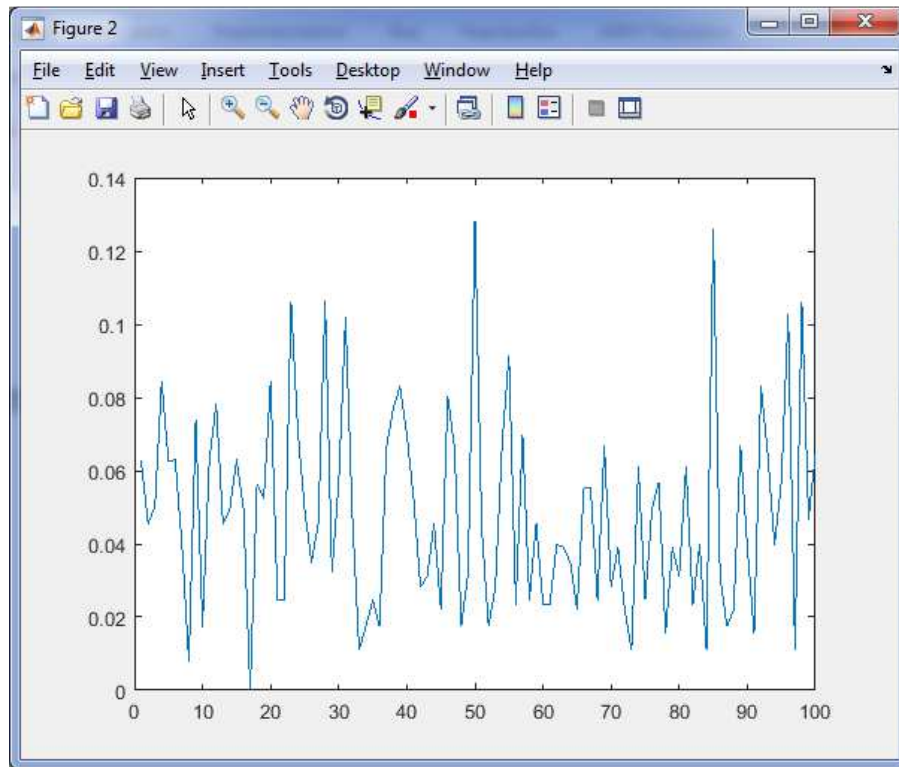


Рис. 3.45. Рівень перешкод біля сигналу 1 фрейму 2 при можливості спостереження величин всіх відліків

З рисунку добре видно, що реальний рівень перешкод близько 0,05.

Тепер розглянемо, при яких рівнях перешкоди алгоритми, які ми досліджуємо, зможуть виділити коректні повідомлення з наявних сигналів.

Накладення перешкод на реальну інформацію реалізуємо відразу після отримання відліків фрейму:

```
% Накладання білого шуму:
snr=6; % відношення сигнал/перешкода (signal/noise ratio) в дБ
sigma=10^(-snr/20)/sqrt(2);
NN=length(data);
gaussnoise=(randn(NN,1)+1i*randn(NN,1))*sigma;
data = data + gaussnoise;
```

Тут наведена формула розрахунку середньоквадратичного відхилення (СКВ) датчика нормальних випадкових чисел, що і при моделюванні інших сигналів [5, 8], виходячи з одиничної потужності сигналу і накладення тільки нормального білого шуму. При накладенні білого шуму на реальну заваду ситуація змінюється, коректною формулою є така:

$$q = 10 \lg \frac{p_s}{p_n + p_g}.$$

Тут p_s – потужність сигналу, p_n – потужність реальної перешкоди, p_g – потужність гаусового білого шуму, все в лінійних одиницях. Звідси СКВ білого шуму для отримання співвідношення с/п q (дБ) повинна обчислюватися за формулою:

$$\sigma_g = \sqrt{0,5p_s(10^{-q} - 10^{-q_0})}. \quad (0.2)$$

Тут q_0 – відношення сигналу до реальної перешкоди в дБ; нормування на $\sqrt{2}$, як і у випадку впливу на сигнал тільки білого шуму, виникає за наявності двох квадратурних каналів. Зрозуміло, що для отримання вихідного співвідношення с/п q_0 СКВ білого шуму повинне дорівнювати 0, і його подальше підвищення неможливо. Це ж добре видно і по наведеній формулі. Для розглянутого найбільш потужного сигналу першого фрейму $p_s = 1,237$, $q_0 = 25,5$ дБ (за результатами роботи блока вимірювання співвідношення с/п).

Результати коректного виділення першого сигналу з першого фрейму для штатної програми (помилка з пошуком синхроімпульсу по модулю згортки виправлена) і доопрацьованої програми, наведені у таблиці 3.6:

Таблиця 3.6

Програма	Співвідношення сигнал/перешкода (дБ)						
	4	5	6	7	14,5	15,5	16,5
	Частота виділення повідомлення						
Штатна	0	0	0	0	0,5	0,6	1
Доопрацьована	0,4	0,7	0,8	1	1	1	1

Обчислення СКВ білого шуму виконано у відповідності з формулою 3.1. Частота виділення повідомлення наведена за 10 запусків програм для кожного співвідношення с/п, що не забезпечує достатню точність, але цілком дозволяє зробити необхідні висновки. Ми можемо тепер розглянуті вище недоліки штатної програми оцінити в одиницях співвідношення с/п для заданого рівня частоти виділення повідомлення. Для частоти, близької до **0,9**, втрати становлять близько **10 дБ**.

На рис. 3.46 наведена реалізація амплітудної розгортки для співвідношення с/п 4 дБ, при якому частота виділення повідомлення після доопрацювань близько 0,5. Червона лінія відповідає порогу для уточнення зони присутності сигналу. Добре бачимо, що запропонована проста процедура уточнення зони присутності сигналу, що добре працює за потужними сигналами (рис. 3.40), призведе до сильного розширення зони за рахунок випадкових завадових викидів:

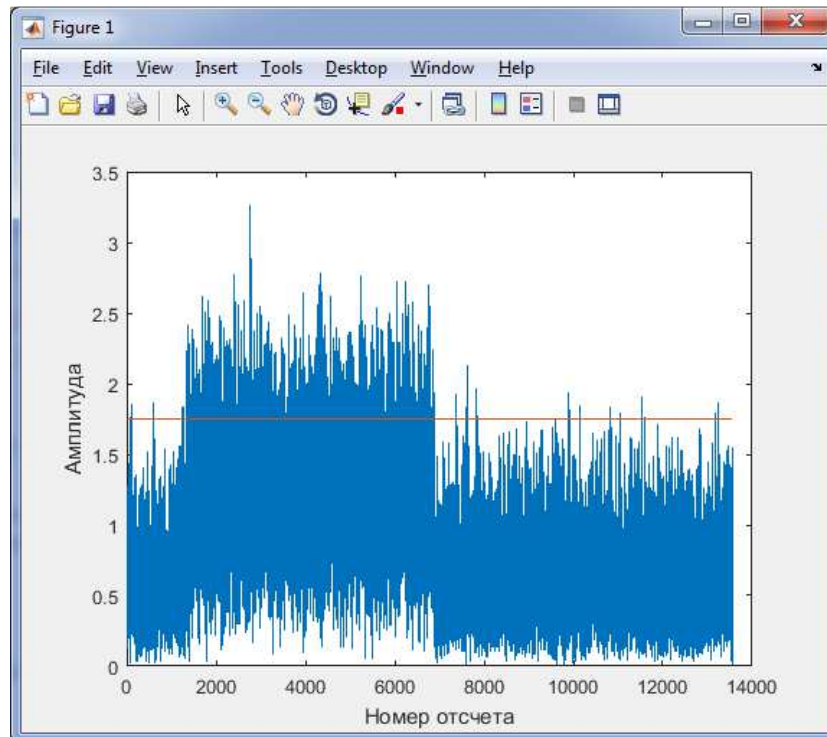


Рис. 3.46. Інтерпретація роботи блоку уточнення зони сигналу для малих значень співвідношення с/п

Таким чином, одне з основних напрямів подальшого вдосконалення алгоритму виділення повідомлень повинно полягати у вдосконаленні методу визначення зони присутності сигналу. Причому ми легко можемо оцінити граничне значення виграшу в цьому випадку. Для цього проведемо виділення сигналу, вказавши в програмі зону сигналу, обчислювану при дії на сигнал тільки реальних перешкод.

Для співвідношення с/п **-2 дБ** частота виділення сигналу дорівнює **0,7**, а для **-1 дБ** — **1.0**. Таким чином, граничне значення виграшу від вдосконалення блоку визначення зони присутності сигналу становить близько **8 дБ**. Зрозуміло, що візуальне спостереження сигналу при співвідношенні с/п **-1 дБ** на амплітудній розгортці не можливо, але фаза сигналу ще добре зберігається, що підтверджується коефіцієнтом кореляції **0,975** з синхроімпульсом і рис. 3.47 (порівняйте з рис. 3.34, на відміну від нього тут зсув частоти прибрано фазовим методом):

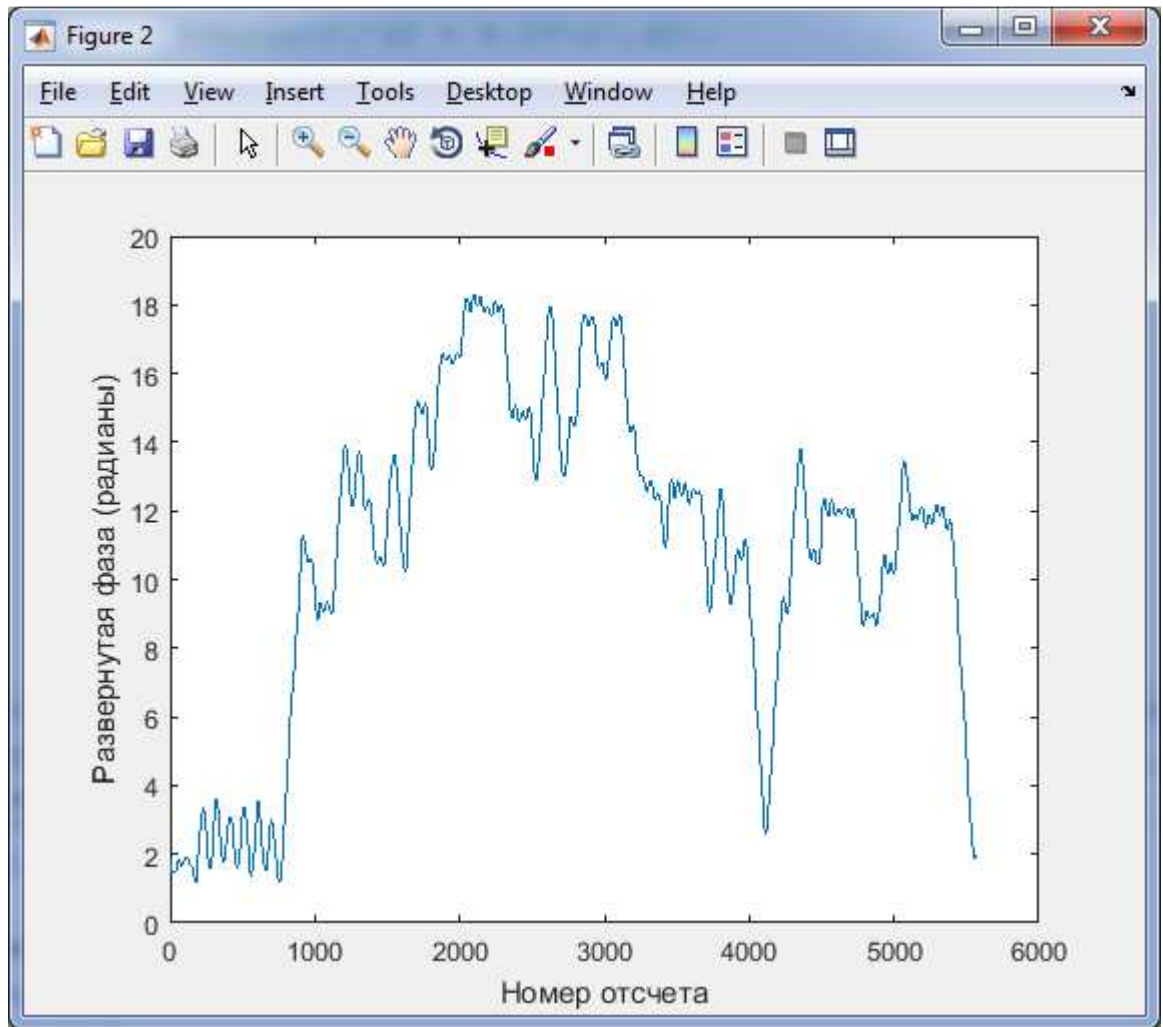


Рис. 3.47. Розгорнута фаза 1-го сигналу фрейму 1 (судно 367164250) при додатковому впливі білого шуму з підсумковим співвідношенням с/п -1 дБ

Результати коректного виділення першого сигналу з другого фрейму ($p_s = 1,269$, $q_0 = 19,5$ дБ) для обох програм наведені в таблиці 3.7:

Таблиця 3.7

Програма	Співвідношення сигнал/перешкода (дБ)									
	-1	0	1	2	3	4	5	6	7	8
	Частота виділення повідомлення									
Штатна	0,5	0,6	0,8	0,8	0,9	0,9	0,9	0,9	0,9	0,9
Доопрацьована	0,5	0,6	0,7	0,9	1	1	1	1	1	1

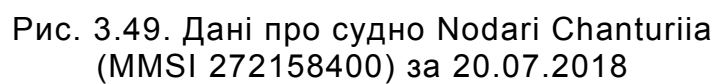
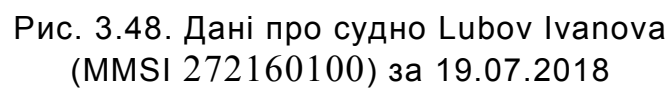
Як бачимо з таблиць 3.6 і 3.7, для впевненого виявлення першого сигналу з другого фрейму потрібно приблизно на **4 дБ** менше співвідношення с/п, ніж при виявленні першого сигналу з першого фрейму, причому результати обох програм майже зрівнялися. В першу чергу це пояснюється значно меншим зсувом по частоті сигналу з другого фрейму (**-6.5 кГц** замість **-57 кГц**, таблиця 3.4). Недоліки штатної програми проявилися у відсутності виходу на одиничний рівень частоти виділення сигналу принаймні до співвідношення с/п **8 дБ**.

3.4.3. Дослідження сигналів судів в акваторії Бузького лиману

Розглянемо тепер результати виявлення сигналів суден в акваторії Бузького лиману. На жаль, в аудиторіях Інституту автоматики і електротехніки, сигнали великих суден класу А, що постійно у великій кількості знаходяться в акваторії Бузького лиману від Чорноморського суднобудівного заводу до підприємства «Нібулон», сильно послабляються міськими забудовами, і виділити їх на фоні шумів не вдається. Ці кораблі дуже рідко проходять вище за течією в бік Миколаївського суднобудівного заводу ім. 61 комунара. Однак, особливо влітку, там ходять невеликі прогулянкові та інші судна класу В. Мінімальна відстань до таких судів від будівлі Інституту автоматики і електротехніки становить всього близько 1 км, і цей простір не дуже забудований. Виявилося, що сигнали таких судів цілком добре можуть бути декодовані розглянутими програмами. У таблиці 3.8 наведені отримані в кінці липня 2018 року зведені результати сеансів виявлення сигналів AIS (назви суден наведені відповідно до латинської транскрипції, що передаються самими суднами в типі повідомлення 5):

Таблиця 3.8.

Дата	Час	MMSI	Назва судна	Кількість отриманих повідомлень		Дальність до судна (км)
				У штатній програмі	У доопрацьованій програмі	
19.07.2018	22:17:22-22:19:52	272160100	Lubov Ivanova	1	3	1,11-1,29
20.07.2018	20:59:49-21:08:50	272158400	Nodari Chanturiia	10	29	1,21-1,45
24.07.2018	16:37:29-16:59:58	272157700	Yurii Makarov	21	75	1,24-1,93
24.07.2018	17:39:07-18:04:38	272157700	Yurii Makarov	17	58	1,18-1,58
24.07.2018	19:59:18-20:08:18	272160100	Lubov Ivanova	2	2	1,15-1,39
24-25.07.2018	22:39:29-00:58:28	272158400	Nodari Chanturiia	45	148	1,35-1,89
26.07.2018	14:48:58-15:15:07	272160300	Mykolaivets	6	31	1,24-1,96
Всього				101	343	



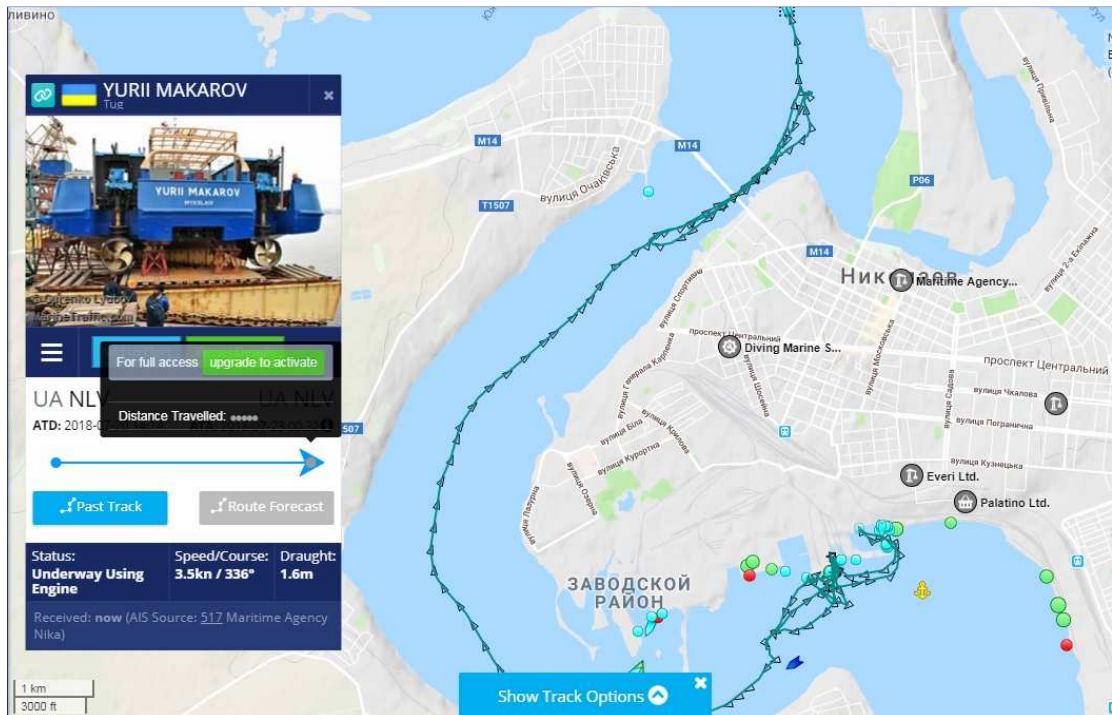


Рис. 3.50. Дані про судно Yurii Makarov (MMSI 272157700) за 24.07.2018

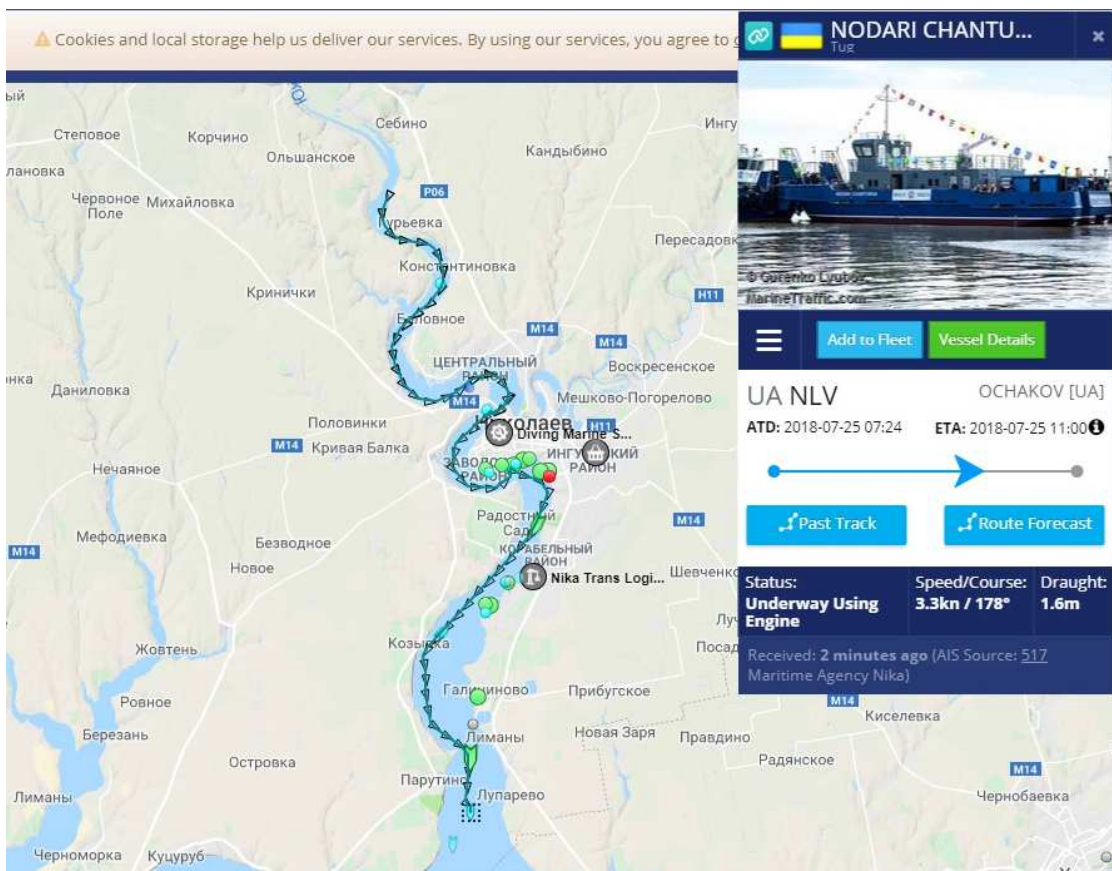


Рис. 3.51. Дані про судно Nodari Chanturiia (MMSI 272158400) за 24-25.07.2018

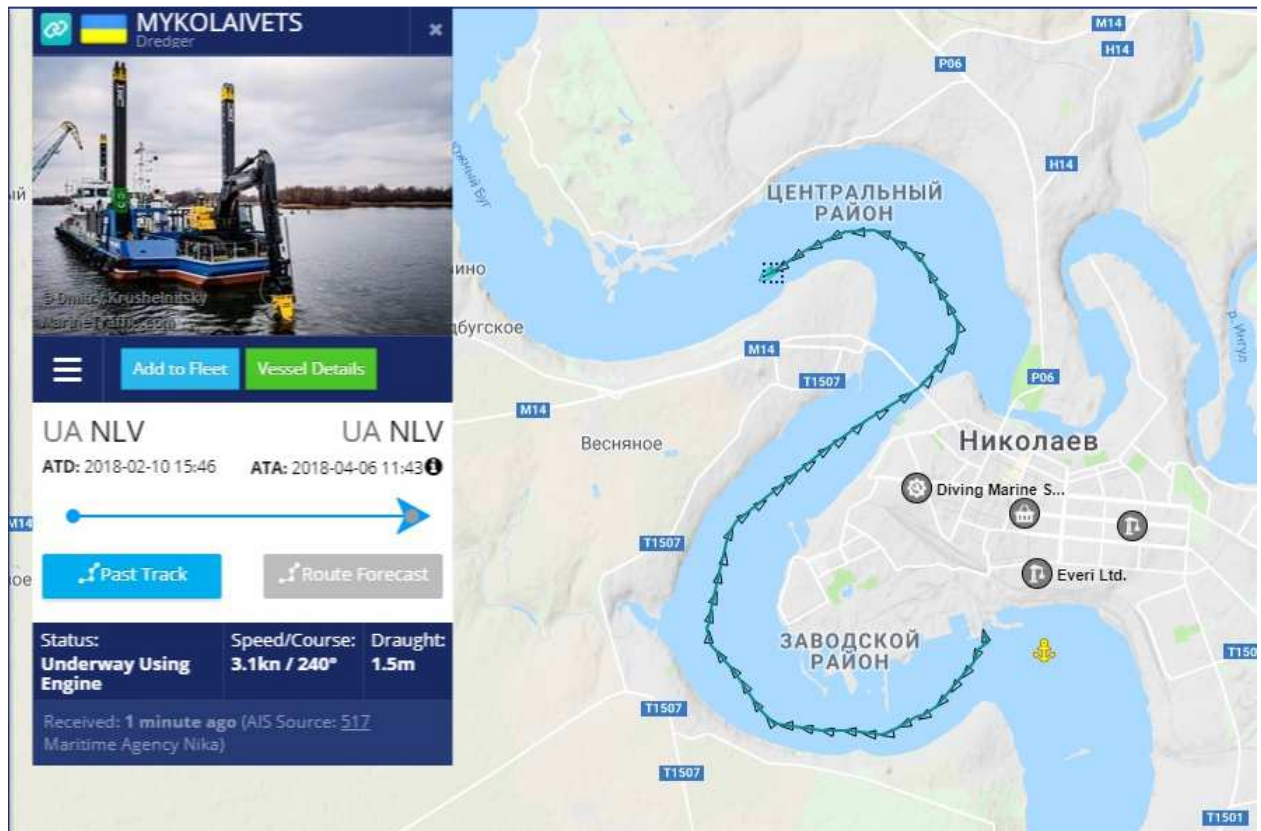


Рис. 3.52. Дані про судно Mykolaivets (MMSI 272160300) за 26.07.2018

Відзначимо, перш за все, що на рис. 3.49–3.51 ми дійсно бачимо значну кількість суден, що знаходяться в акваторії Бузького лиману від Чорноморського суднобудівного заводу до підприємства «Нібулон» (відображаються на карті кольоровими кружками).

Нанесення на карту географічних координат, переданих судном, ще раз підтверджують знаходження судна в момент отримання від нього сигналів AIS, в частині акваторії Бузького лиману найближчої до приймальної антени (будівля інституту автоматики та електротехніки). Наприклад, для судна Nodari Chanturiya в сеансі спостереження за 20.07.2018 його координати на карті розташовані наступним чином (рис. 3.53):

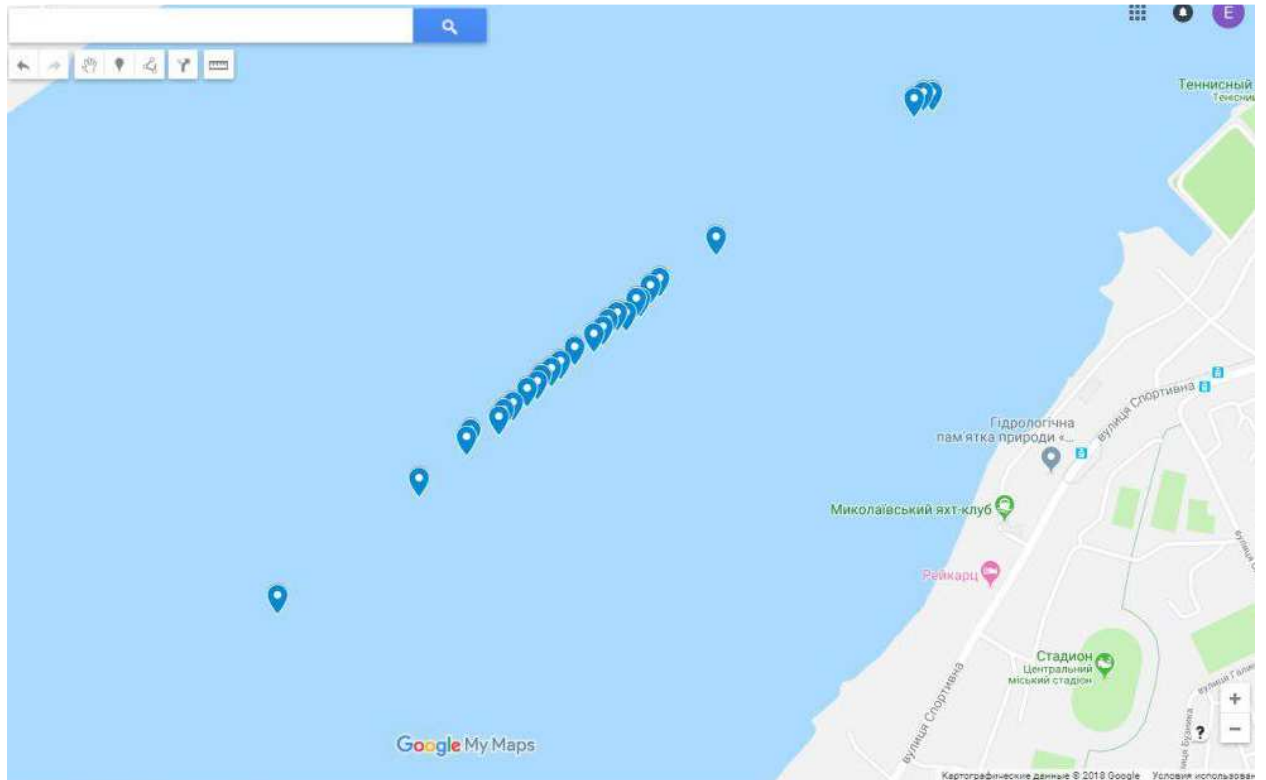


Рис. 3.53. Передані судном Nodari Chanturiya координати в сеансі спостереження 20.07.2018

Відстань від судна до приймальної антени при отриманні першого сигналу склала 1,38 км, при отриманні останнього — 1,45 км, мінімальна відстань — 1,21 км.

Від цього ж судна в ніч з 24 на 25 липня отримано 145 повідомлень протягом більше двох годин з 22:39:29 до 0:58:28 з перервою з 22:56:59 до 23:47:12, очевидно, це пов'язано з виходом і поверненням судна в зону прийому сигналів AIS. На карті отримані координати виглядають наступним чином (при відсутності змін переданих координат вони не наносилися):

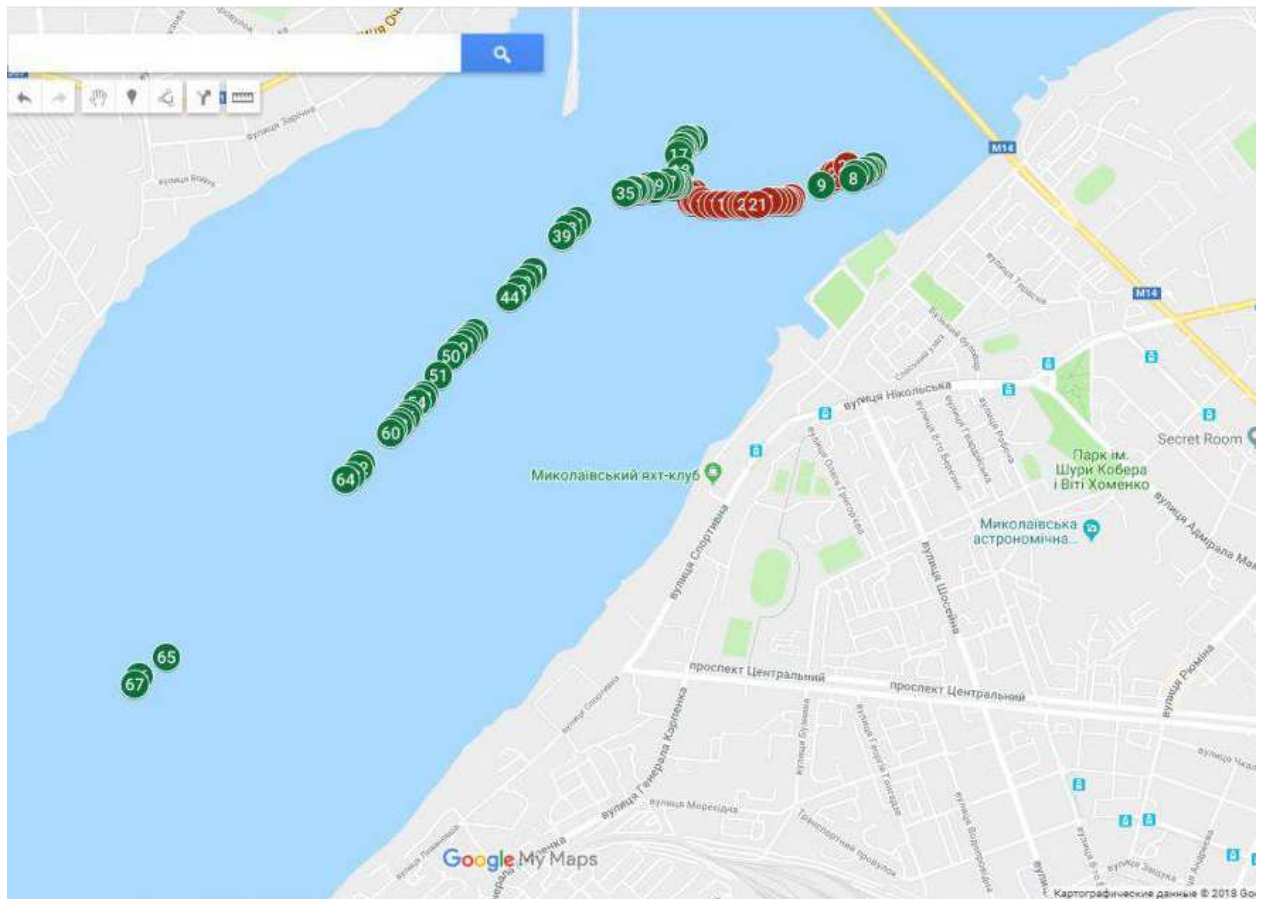


Рис. 3.53. Передані судном Nodari Chanturiya координати в сеансі спостереження 24-25.07.2018

Тут червоні мітки відповідають отриманим даним до виходу із зони контролю, зелені – після повторного повернення. Як бачимо, судно зупиняло рух і деякий час дрейфувало недалеко від берега. Відстань від судна до приймальної антени при отриманні першого сигналу склало 1,48 км, при отриманні останнього — 1,87 км, мінімальна відстань — 1,35 км. Таким чином, дальність прийому сигналів AIS від суден класу В складає близько 2 км при порівняно не щільній забудові капітальними будівлями шляху розповсюдження сигналів.

У процесі спостереження найбільше отримано повідомлень 1-го типу, у кілька разів менше повідомлень 3-го типу; судно Lubov Ivanova передавало тільки повідомлення 18-го типу (довжина повідомлень 168 бітів без коду CRC), а також практично від усіх суден отримані розширені повідомлення 5-го типу (довжина повідомлення 424 біта), які після декодування виглядають так:

```
MMSI:272158400 Call Sign:UXEH@@@ Ship Name:NODARI_CHANTURIIA@@@
Ship Type:52 Destination:NIKOLAEV_@@@@@
```

```
MMSI:272157700 Call Sign:UXDZ@@@ Ship Name:YURII_MAKAROV@@@@@@@
Ship Type:52 Destination:NIKOLAEV_@@@@@@@@@@@
```

Розглянемо тепер самі отримані радіосигнали. Вони відчутно слабкіші сигналів, записаних в околицях Бостона. Максимальні значення відношення сигнал/перешкода досягають 14 дБ (при вимірюванні у всій смузі сигналу) і на амплітудній розгортці виглядають, наприклад, так (рис 3.54, червоним кольором показана зона виділення сигналу):

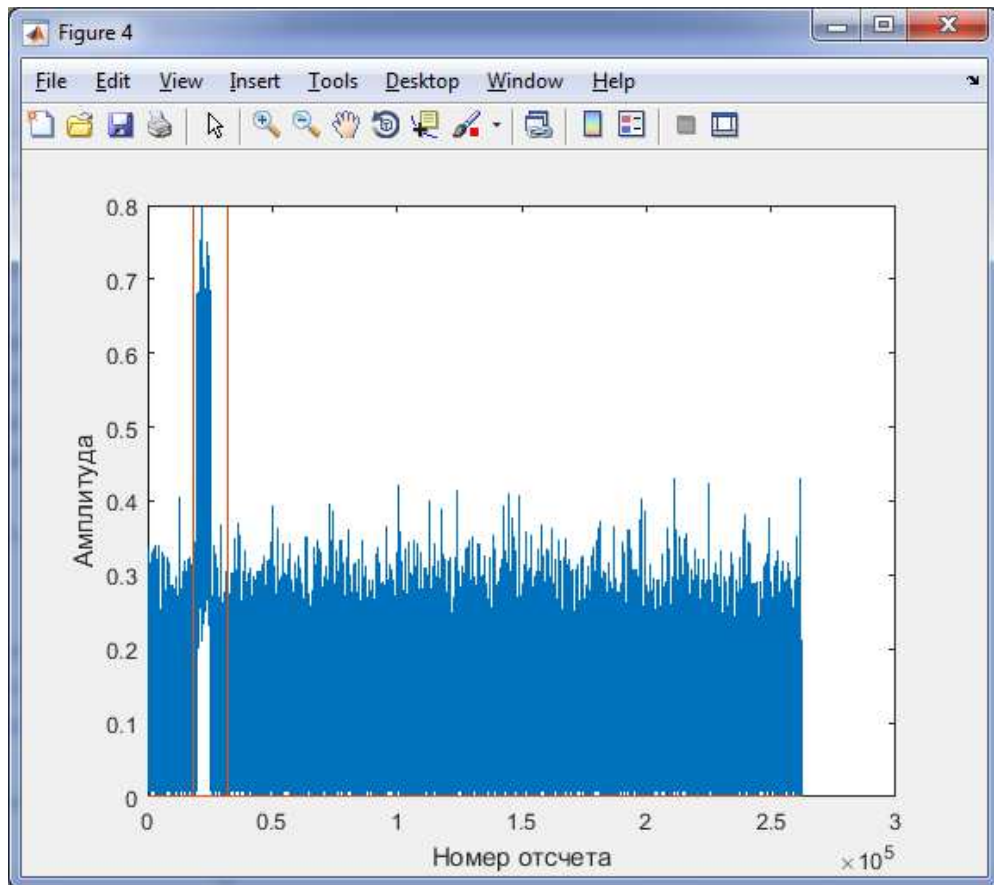


Рис. 3.54. Амплітудна розгортка сигналу у всьому фреймі від судна Nodari Chanturiya в сеансі спостереження 20.07.2018 (час 21:08:30)

Як бачимо, зона сигналу приблизно в 2 рази завищена, розглянута вище процедура уточнення зони присутності сигналу (див. рис. 3.40) і тут дозволяє точно визначити межі сигналу (рис. 3.55):

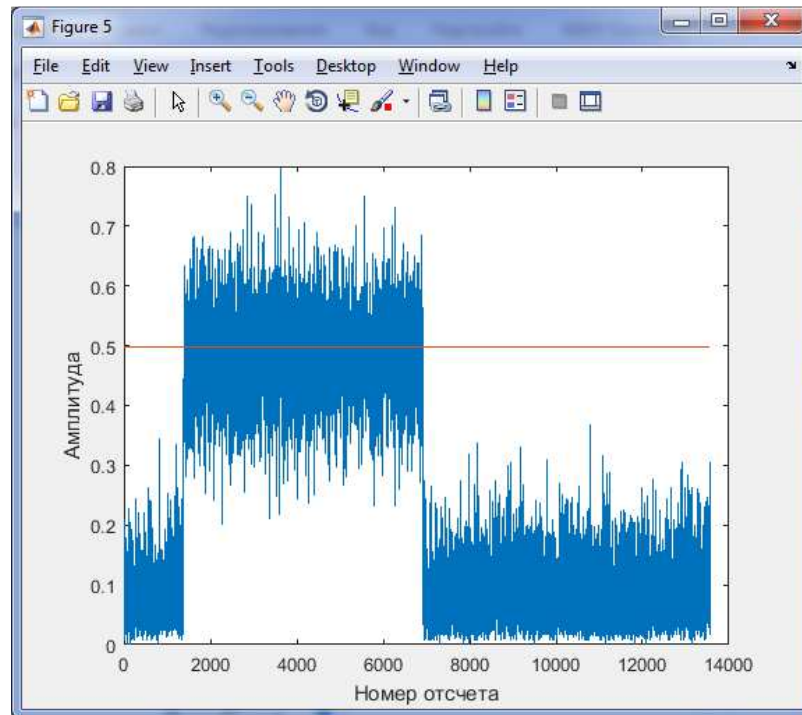


Рис. 3.55. Амплітудна розгортка сигналу в первісно визначеній зоні від судна Nodari Chanturiya в сеансі спостереження 20.07.2018 (час 21:08:30)

Спектр сигналу має відносно невелике негативне зміщення -9.3 кГц (рис 3.56):

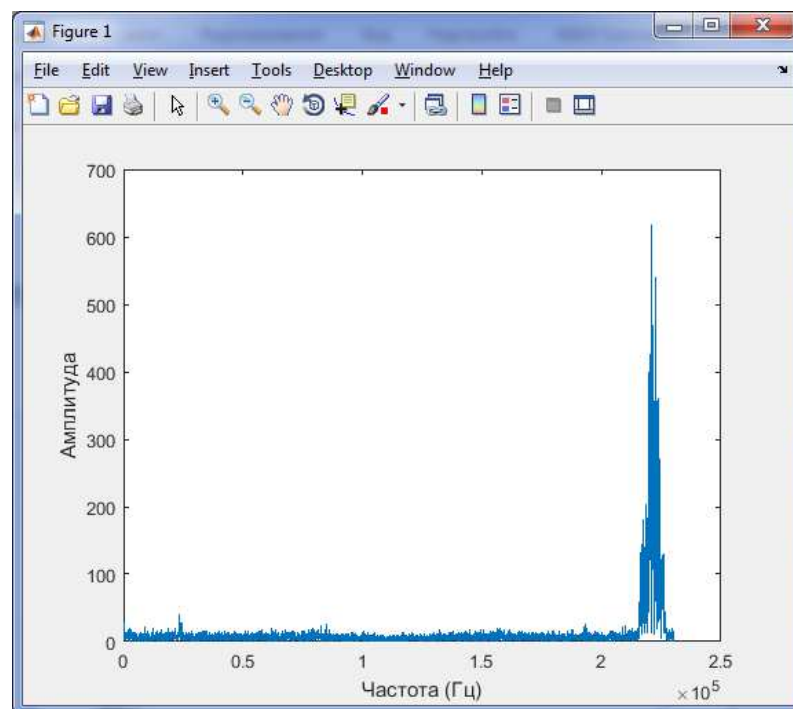


Рис. 3.56. Спектр сигналу в уточненій зоні від судна Nodari Chanturiya в сеансі спостереження 20.07.2018 (час 21:08:30)

Як бачимо, відношення сигнал/перешкода в максимумі спектральної потужності досягає 35 дБ. Природно, в цих умовах фаза сигналу повинна виглядати ідеально, що підтверджується рис. 3.57:

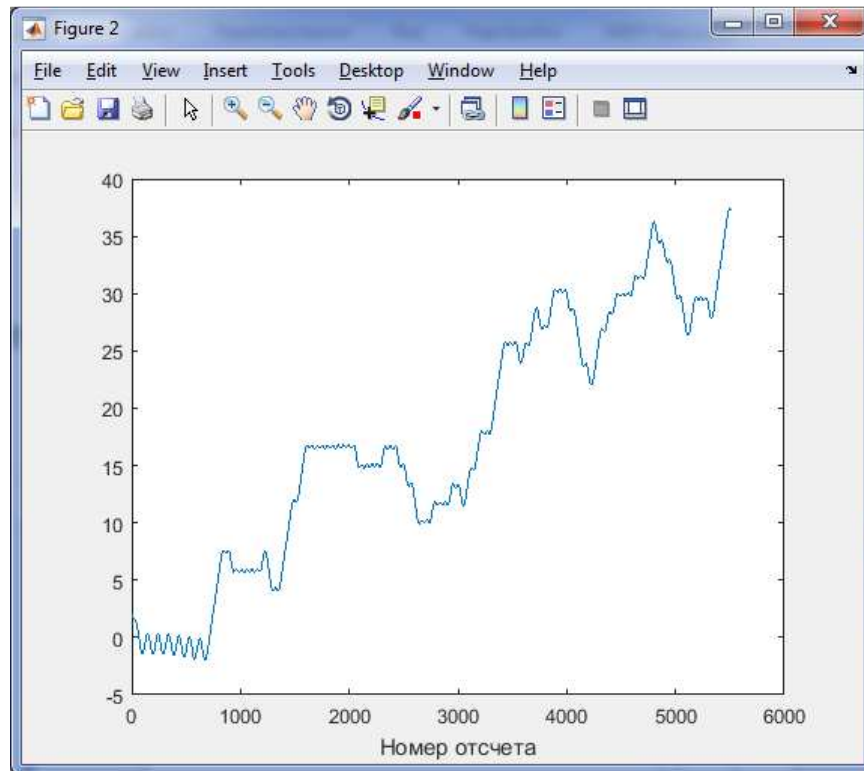


Рис. 3.57. Розгорнута фаза сигналу від судна Nodari Chanturiya в сеансі спостереження 20.07.2018 (час 21:08:30)

Зсув по частоті повністю прибрано, коефіцієнт кореляції з ідеальним синхроімпульсом 0,975.

Мінімальне відношення сигнал/перешкода для коректно виділених повідомлень наближається до 0 дБ. Такі сигнали на амплітудній розгортці візуально практично не детектуються. Наприклад, один з сигналів, отриманий від судна Lubov Ivanova, виглядає так (рис. 3.58):

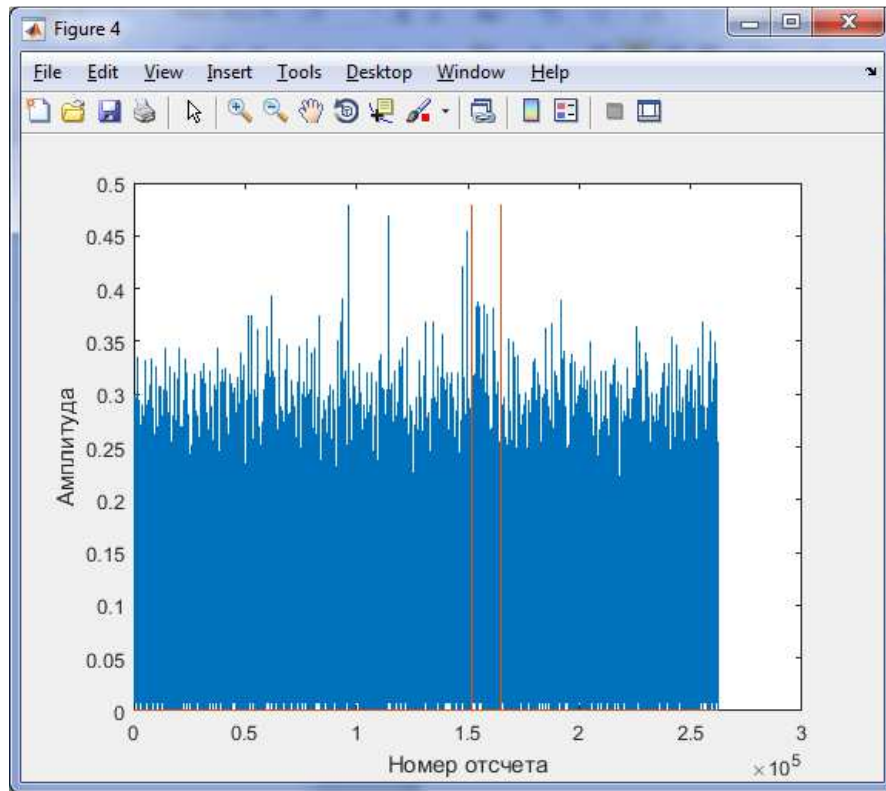


Рис. 3.58. Амплітудна розгортка сигналу у всьому фреймі від судна *Libov Ivanova* в сеансі спостереження 19.07.2018 (час 22: 17: 22)

Виміряне відношення сигнал/перешкода у всій смузі сигналу склало 0,4 дБ. Трохи більш помітний сигнал в обчисленій зоні (рис. 3.59):

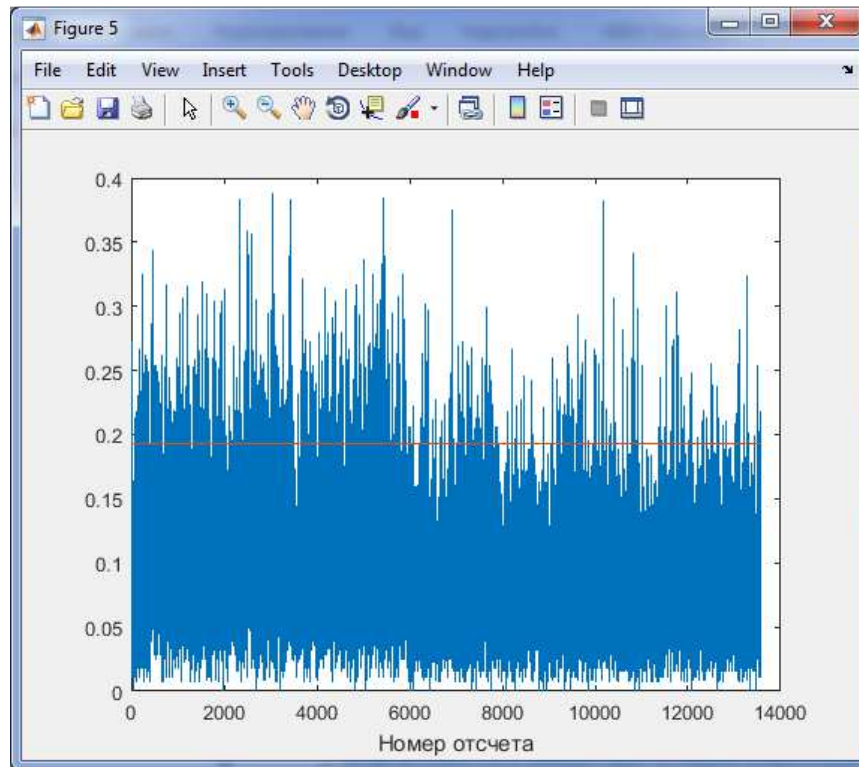


Рис. 3.59. Амплітудна розгортка сигналу в первісно визначеній зоні від судна Lubov Ivanova в сеансі спостереження 19.07.2018 (час 22:17:22)

Як бачимо, для таких слабких сигналів уточнити зону присутності сигналу не вдається. Проте, в спектральній області сигнал добре видно (рис 3.60):

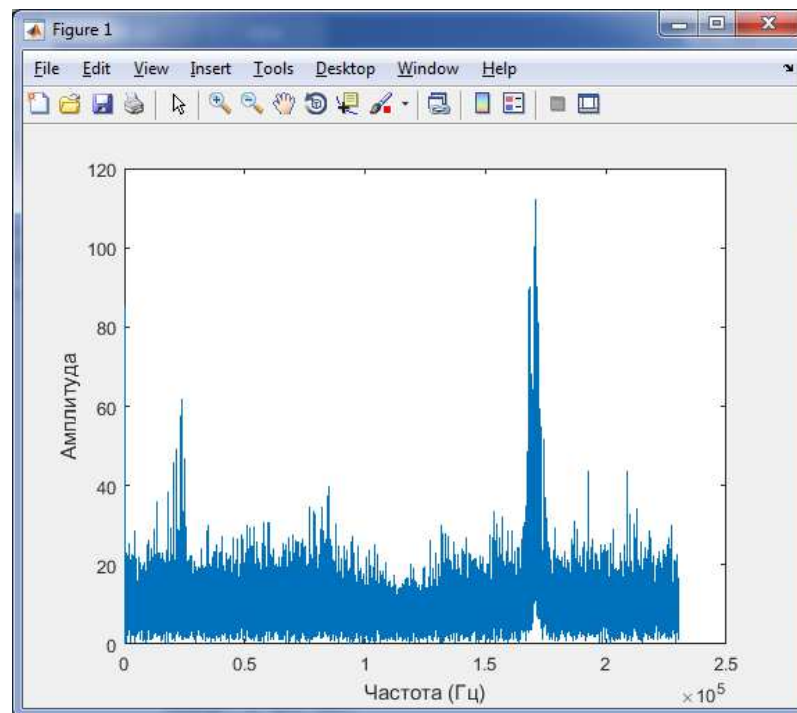


Рис. 3.60. Спектр сигналу від судна Lubov Ivanova у сеансі спостереження 19.07.2018 (час 22: 17: 22)

Зсув сигналу по частоті значно вищий, ніж у попереднього розглянутого сигналу, і склав -59 кГц. Розгорнута фаза сигналу в цьому випадку виглядає так (рис. 3.61):

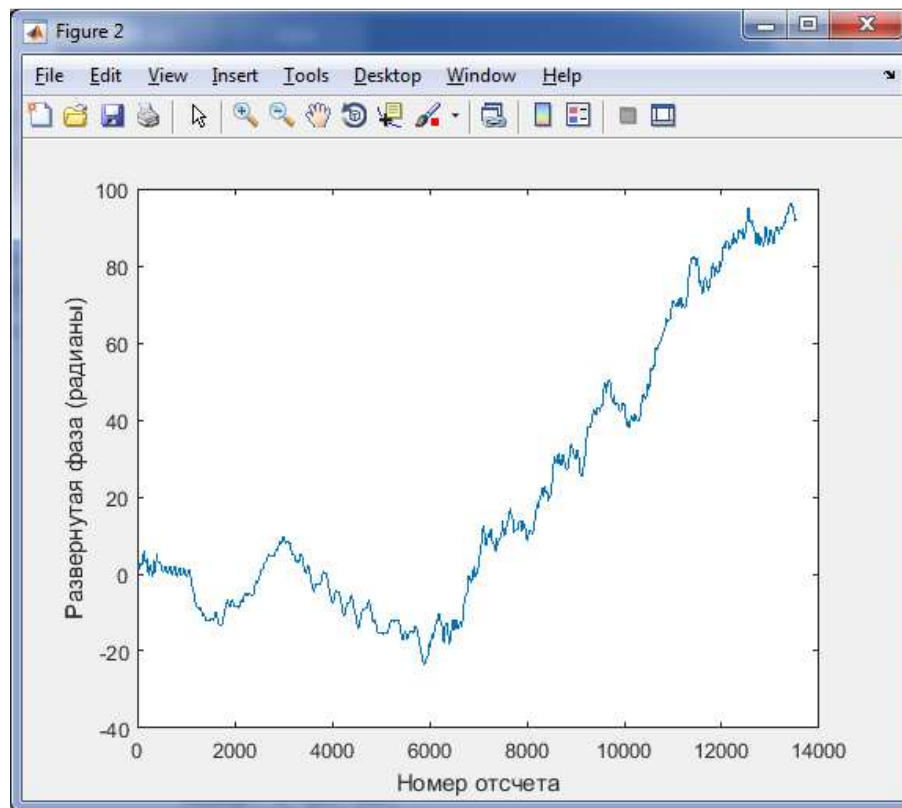


Рис. 3.57. Розгорнута фаза сигналу від судна *Lubov Ivanova* у сеансі спостереження 19.07.2018 (час 22: 17: 22)

Незважаючи на дворазове перевищення зони присутності сигналу, зсув по частоті вдалося виміряти практично точно і безпомилково визначити значення всіх бітів у повідомленні.

На закінчення можна порівняти різні дані, отримані в даному розділі. Перш за все, повернемося до таблиці 3.8, де наведено, що при виявленні сигналів AIS в акваторії Бузького лиману доопрацьованою програмою всього виділено 343 повідомлення з коректним кодом CRC, тоді як по записам квадратурних каналів штатна програма корпорації MathWorks виділяє всього 101 повідомлення. Таким чином, частота виявлення після запропонованих доробок збільшилася більш ніж в **3 рази**.

У підрозділі 3.2 ми отримали криву залежності BER від відношення с/п для GMSK-сигналу з параметрами, які використовуються в технології AIS (рис. 3.21). З рисунка зокрема випливає, що для досягнення $BER\ 10^{-4}$ досить співвідношення сигнал/перешкода близько -5 дБ, виміряне у всій смузі сигналу. Це забезпечить виявлення коротких повідомлень AIS довжиною 168 бітів з частотою 0,98. Для виявлення половини сигналів досить мати $BER\ 10^{-2.5}$, відповідно, співвідношення с/п має бути близько -7 дБ. Однак це

отримано для ідеалізованої моделі. Вище зазначалося (с. 110), що реалізована в MATLAB модель каналу передачі інформації з GMSK-модуляцією сильно спрощена, що не дозволить підтвердити зазначені характеристики в реальних каналах зв'язку. Спрощення, перш за все, відноситься до припущення ідеальної тимчасової синхронізації і відсутності зсуву по частоті між передавачем і приймачем. Після знайомства з реальними сигналами AIS та алгоритмами їх виявлення, ми добре знаємо, що вирішення цих двох проблем є ключовим для успішного функціонування всієї технології.

Тимчасова синхронізація має бути присутня в усіх без винятку системах зв'язку, а необхідна точність зведення частот залежить від виду використовуваної модуляції. Так для технології ADS-B, яка використовується для супроводу повітряних суден, нам не знадобилося цифрове зведення частот через використання амплітудної модуляції, яка набагато менш чутлива до неузгодженості частот, ніж сигнал GMSK, що переносить інформацію у фазі.

Дійсно, вимірювання можливостей виявлення реальних сигналів AIS на фоні білого шуму в п. 3.4.2. показало, що при виявленні сигналу зі значним зсувом частоти для виявлення приблизно половини сигналів AIS необхідно співвідношення с/п близько 4 дБ (таблиця 3.6), тоді як при малих зсувах для цього достатньо -1 дБ (таблиця 3.7).

Одиничне виявлення сигналу від судна *Lubov Ivanova* зі співвідношенням с/п 0.4 дБ і значним зсувом по частоті не суперечить отриманим даним. Точне ж вимірювання частоти виявлення від співвідношення с/п для сигналів AIS на фоні реальних перешкод досить складно, оскільки необхідно буде якось фіксувати наявність сигналу судна в умовах не збігу коду CRC.

3.5. Завдання для лабораторних, курсових робіт і дипломних проектів

Варіант 1.

Дослідити в MATLAB ефективність демодулятора `comm.GMSKDemodulator()`, що використовує алгоритм динамічного програмування Вітербі, для демодуляції сигналів AIS.

Варіант 2.

Дослідити можливість використання записів сигналів суден, викладених в Інтернет, для відпрацювання алгоритмів виявлення сигналів AIS (наприклад, за адресою: https://github.com/freerange/ais-on-sdr/wiki/Capturing-raw-AIS-data-using-rtl_fm-and-decoding-using-aisdecode)

Варіант 3.

На мові C++ або в MATLAB написати програму, що виділяє з 16-кового повідомлення AIS наступні параметри:

- індикатор повторення (Repeatindicator);
- навігаційний статус (Navigational status);
- швидкість розвороту (Rate of turn – ROT_{AIS}).

Варіант 4.

На мові C++ або в Matlab написати програму, що виділяє з 16-кового повідомлення AIS наступні параметри:

- швидкість руху (Speed over ground – SOG);
- точність вимірювання позиції (Position accuracy);
- курс по Землі (Course over ground – COG).

Варіант 5.

На мові C++ або в MATLAB написати програму, що виділяє з 16-кового повідомлення AIS наступні параметри:

- дійсний напрямок руху (True heading);
- мітка часу (Time stamp);
- індикатор спеціального маневру (Special maneuver indicator).

Варіант 6.

На мові C++ або в Matlab написати програму, що виділяє з 16-кового повідомлення AIS наступні параметри:

- RAIM-flag;
- стан зв'язку (Communication state).

Варіант 7.

Дослідження ефективності методів оцінки частотного зсуву на моделі сигналу GMSK у MATLAB.

Варіант 8.

Дослідження впливу гаусового фільтра на сигнали GMSK з великим зміщенням по частоті.

Варіант 9.

Реалізація та оцінка ефективності послідовного в часі алгоритму виявлення сигналів AIS.

Варіант 10.

Порівняльна оцінка ефективності різних варіантів відбору одного відліку на тривалості символу для прийняття рішення (відлік з максимальною амплітудою сигналу, середнє значення кількох відліків, один середній відлік).

Варіант 11.

Дослідження на моделях ефективності використання неперіодичних синхроімпульсів в технології AIS.

Варіант 12.

Дослідження стійкості розглянутих алгоритмів виявлення сигналів AIS до накладення на реальні сигнали морських суден білого гаусового шуму різної потужності.

Варіант 13.

Проведення робіт з приймання сигналів AIS від морських суден в акваторії Бузького лиману.

4. ДОСЛІДЖЕННЯ РЕАЛЬНИХ СУПУТНИКОВИХ СИГНАЛІВ

На жаль, частоти, на яких ретранслюють цифрову інформацію більшість супутників, знаходяться за межами робочого діапазону дешевих приймачів RTL-SDR (25 МГц – 1,75 ГГц). Супутникове мовлення ведеться у двох основних діапазонах — С-Band і Ku-Band. У діапазоні С (4 ГГц) віщають в основному американські та російські супутники, в діапазоні Ku (10,700-12,750 ГГц) — європейські.

В даний час у діапазоні С супутникові сигнали можна приймати за допомогою програмно-визначуваних радіосистем USRP (Universal Software Radio Peripheral). Такі радіопристрої забезпечують діапазон частоти до 6 ГГц з миттєвою шириною смуги пропускання до 56 МГц, однак, їх вартість починається з \$1000. У цьому посібнику ми обмежимося аналізом запису супутникового сигналу, наявного у розпорядженні авторів після проведення однієї з НДР у 2006–2008 роках.

- 4.1. Стандарт передачі Інтернет-даних через супутникові канали зв'язку **Ошибка! Закладка не определена.**
- 4.2. Короткий опис запису супутникового сигналу **Ошибка! Закладка не определена.**
- 4.3. Математична модель сигналів з фазовою модуляцією **Ошибка! Закладка не определена.**
- 4.4. Результати порівняння реального супутникового сигналу з моделлю QPSK сигналів **Ошибка! Закладка не определена.**
- 4.5. Завдання для лабораторних, курсових робіт і дипломних проектів... **Ошибка! Закладка не определена.**

4.1. Стандарт передачі Інтернет-даних через супутникові канали зв'язку

Насамперед, відзначимо, що стандарт передачі цифрової інформації по супутниковим каналам в США і Європейських країнах дуже докладно описаний і доступний. Таким стандартом у США займається Асоціація телекомунікаційної промисловості – Telecommunications Industry Association (TIA).

На рис. 4.1 наведено титульний лист цього стандарту (версія 2006 року) [11]:

TIA STANDARD

IP over Satellite (IPOS)

TIA-1008-A

(Revision of TIA-1008-A)
May 2006

TELECOMMUNICATIONS INDUSTRY ASSOCIATION



Рис. 4.1. Стандарт передачі даних
через супутникові канали зв'язку

У документі на 247 сторінках дуже докладно і детально описані способи формування цифрової інформації для передачі по супутниковим каналам, починаючи з фізичного рівня.

Насамперед, подивимося, які види маніпуляції закладені в супутникові канали зв'язку. Повний набір видів маніпуляції і коректувальних кодів верхнього рівня для режиму DVB-S2 (digital video broadcasting via satellite, second generation — друге покоління цифрового супутникового телебачення) наведено в таблиці 4.13.12.1-2 стандарту (рис. 4.2):

Table 4.13.12.1-2. DVB-S2 Code for Modulation and Code Rate

ModCod	Modulation	FEC Code	Inner Code Rate
1	QPSK	LDPC+BCH	1/4
2	QPSK	LDPC+BCH	1/3
3	QPSK	LDPC+BCH	5/12
...	...	...	...
10	QPSK	LDPC+BCH	8/9
11	QPSK	LDPC+BCH	9/10
12	8-PSK	LDPC+BCH	3/5
13	8-PSK	LDPC+BCH	2/3
14	8-PSK	LDPC+BCH	3/4
...	...	...	...
17	8-PSK	LDPC+BCH	9/10
18	16APSK	LDPC+BCH	2/3
19	16APSK	LDPC+BCH	3/4
...	...	...	...
23	16APSK	LDPC+BCH	9/10
24	32APSK	LDPC+BCH	3/4
25	32APSK	LDPC+BCH	4/5
...	...	...	...
28	32APSK	LDPC+BCH	9/10

Рис. 2.2. Варіанти маніпуляції і коди верхнього рівня в режимі DVB-S2

Як бачимо, в цифровому телебаченні другого покоління використовуються знайомі нам види маніпуляції QPSK і 8-PSK [5], а також маніпуляція APSK (amplitude and phase-shift keying – амплітудно-фазова модуляція), яка дуже близька до знову ж знайомої нам QAM-маніпуляції. На відміну від QAM-маніпуляції, при APSK-маніпуляції комплексну площину ділять не паралельними осям лініями, а концентричними колами, як показано на рисунку 4.3 для 16-APSK:

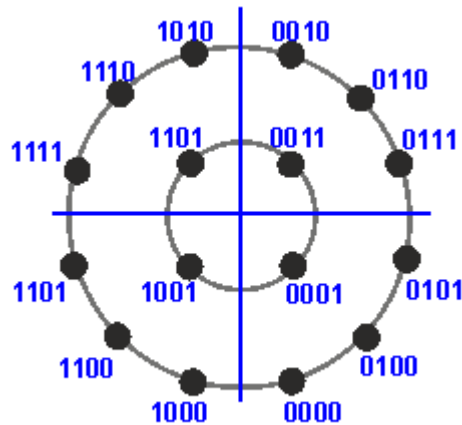


Рис. 4.3. Фазовий портрет (сузір'я) сигналу 16-APSK

До переваг APSK-маніпуляції в порівнянні з QAM відносять меншу кількість рівнів зміни амплітуди, меншу потужність сигналу і більше зменшення BER при використанні Грей-кодування точок сузір'я. В той же час відзначають перевагу QAM при великих кількостях точок сузір'я. На жаль, у MATLAB модулятор і демодулятор сигналів APSK не реалізований в тому вигляді, який ми використовували для розглянутих нами видів маніпуляції. Правда, на сайті компанії MathWorks можна знайти програми, що так чи інакше моделюють APSK-сигнали.

З цієї ж таблиці ми бачимо, що на верхньому рівні протоколу використовуються знайомі нам коди БЧХ і коди LDPC (Low-Density Parity-Check) – код з малою щільністю перевірок на парність. Це блоковий лінійний код з перевіркою парності. Особливістю є мала щільність значущих елементів перевіркової матриці, за рахунок чого досягається відносна простота реалізації засобів кодування. Його також називають кодом Галлагера, за іменем автора першої роботи на тему LDPC-кодів.

Інформаційний бітовий масив перед передачею тим чи іншим видом модуляції, піддається великій кількості перетворень, які забезпечують при належній обробці безпомилкове відновлення вихідного інформаційного потоку. Відразу відзначимо, що в стандарті відсутні будь-які рекомендації з обробки сигналів, що приймаються.

Для прикладу розглянемо схему верхнього рівня підготовки потоку цифрової інформації для передачі від віддалених терміналів до основного концентратора системи (inroute transmissions), наведеної на рисунку 3.5-1 стандарту (рис. 4.4):

Figure 3.5-1. High-Level IPoS Inroute Diagram

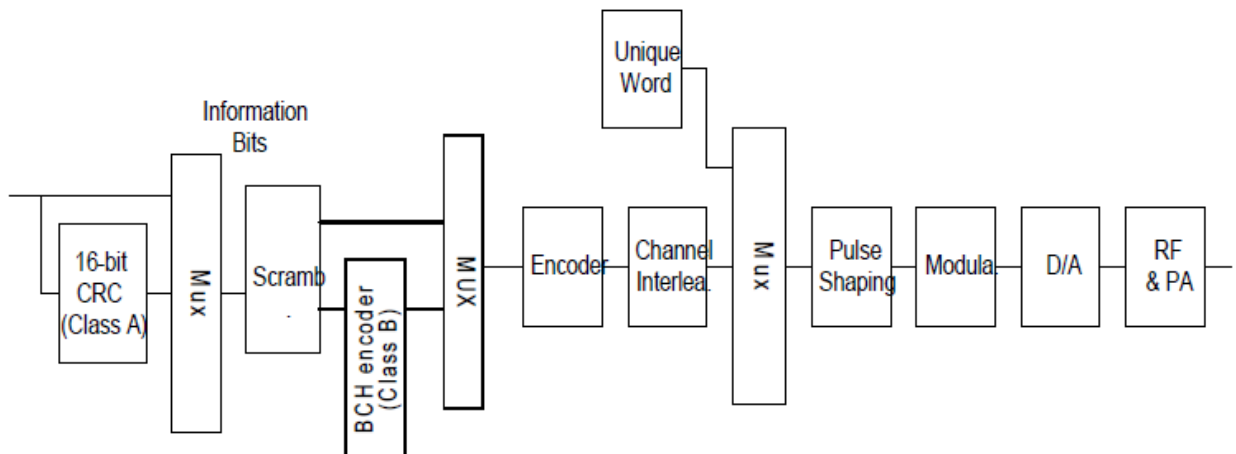


Рис. 4.4. Схема верхнього рівня підготовки потоку інформації в режимі inroute

Обробка трохи відрізняється для терміналів класів А і В. Для терміналів класу А збереження пакетів забезпечується 16-бітним CRC, для терміналів класу В — кодами БЧХ. В обох випадках використовується **скремблювання**, причому коди CRC додаються в пакет до скремблювання, коди БЧХ — після скремблювання. Далі застосовується турбо-код, та інформація піддається табличному **перемежуванню**. Після перемежування на початку блоку пакетів вставляються контрольні слова, і бітовий потік накладається на несучу одним з методів маніпуляції.

Основою турбо-коду є згортковий код (2, 3), схема якого наведена на рисунку 7.5:

На базі розглянутого коду реалізується завадостійкий паралельно-каскадний згортковий **турбо-код** Р2СЗ, широко використовуваний у супутникових системах зв'язку (рисунк 4.6):

Figure 3.5.3.2.3-1. Encoder Block Diagram for Parsed Parallel Concatenated Convolutional Codes

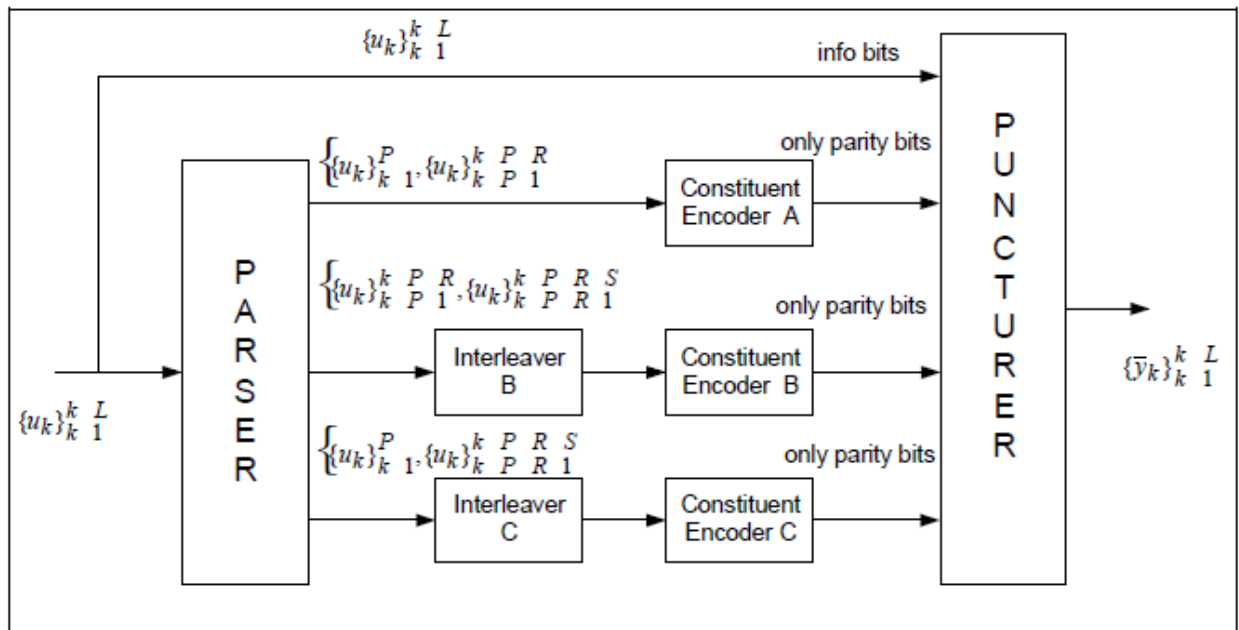


Рис. 4.6. Діаграма паралельно-каскадного загорткового турбо-коду P2C3

Як бачимо, код є систематичним. Особливістю коду є поділ на вході потоку бітів на три перекривні частини (*parser*) і роздільне формування перевірочних бітів для кожної частини. Перша частина не піддається перемежуванню, для другої і третьої частини використовуються різні схеми перемежування. Далі використовується згортковий код (2, 3) для кожної частини зі своєю процедурою виколування. Зверніть увагу, що перемежування всередині турбо-коду P2C3 і каналне перемежування (*channel interleaver*), наведене на рисунку 4.4, – дві окремі процедури, тобто перемежування використовується двічі (в межах коду використовується табличне перемежування, потім матричне каналне перемежування).

Повна модель генерування цифрової обробки для передачі інформації від віддалених терміналів до основного концентратора системи (*inroute transmissions*) була реалізована одним з авторів цього посібника для потреб обробки реальних супутникових сигналів [12].

Для передачі інформації від основного концентратора користувальницьким терміналам (*outroute transmissions*) використовуються стандарти DVB-S або DVB-S2 широкосмугових супутникових каналів, прийняті Європейським союзом мовлення (European Broadcasting Union — EBU). Спочатку вони призначалися для трансляції цифрового телебачення через супутникові канали зв'язку. Структура обробки інформації значно відрізняється від режиму *outroute transmissions*. Не вдаючись у подробиці, відзначимо дві найбільш суттєвих відмінності. При каналному кодуванні замість кодів CRC і BCH викори-

стовується код Ріда-Соломона (204, 188, 8), який є усіканням стандартного коду РС (255, 239, 8) на базі байтового алфавіту (рисунок 4.7):

Figure 3.4.6-1. Reed-Solomon (204,188, T=8) Error Protected Packet

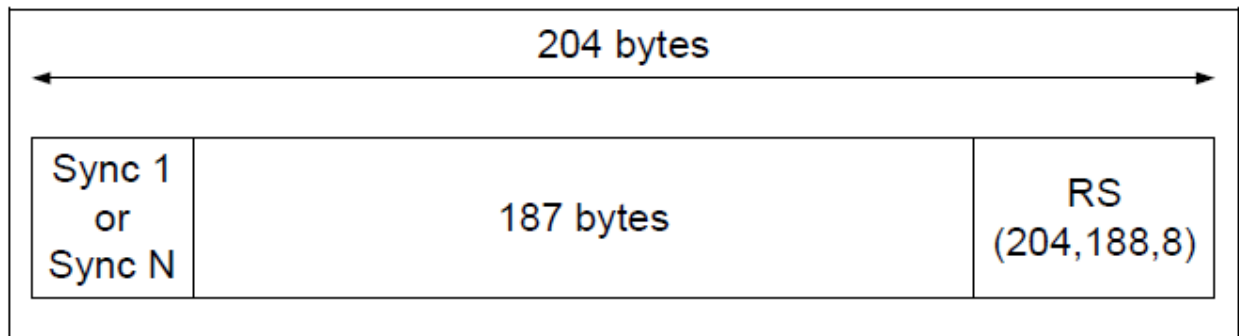


Рис. 4.7. Схема закриття пакету кодом Ріда-Соломона в режимі outroute transmissions

В якості внутрішнього коду використовується згортковий код швидкості $1/2$ з $k = 7$ (length constraint) і поліномами (0x171, 0x133). Додатково виконанням реалізуються швидкості $2/3$, $3/4$, $5/6$ і $7/8$.

4.2. Короткий опис запису супутникового сигналу

Запис супутникового сигналу зроблено 8.09.2006 року (обсягом 39 Мб). Для передачі використовується модуляція QPSK. Умовний інтервал передачі одного символу 1 мс. На кожному символі з приймача квантувалося 20 комплексних амплітуд у форматі 2-байтових цілих чисел. Перед записом сигнал пропускався через фільтр Блекмана з порядком 43, рівнем бічних пелюсток – 51 дБ і нормованою смугою за рівнем пригнічення 0,2. На кожному символі з виходу фільтра залишалося 2 відліки (фільтрація здійснюється з зсувом на 10 відліків).

4.3. Математична модель сигналів з фазовою модуляцією

Перш ніж приступити до вивчення реального сигналу, розглянемо модель PSK-сигналів, дотримуючись [5]. Спрощено, модель фазової маніпуляції можна описати наступним чином:

$$u_s(t) = Ae^{j(\omega_0 + \phi + \psi_k)},$$

Де A – відома амплітуда сигналу;

ω_0 – кругова частота, $\omega_0 = 2\pi f_0$, f_0 – несуча частота (після зняття несучої — невідоме зміщення (помилка) по частоті);

ϕ – початкова фаза (набігає при поширенні сигналу);

ψ_k – модуляція фази згідно з переданими символом, змінюється через інтервал Δt , $k = 1 \dots k_0$, k_0 – кількість розбивань фазового простору, завжди дорівнює ступеню 2, тобто $k_0 = 2^{n_0} \cdot n_0$ відповідає кількості переданих бітів однією фазою. Якщо $F_\psi = \frac{1}{\Delta t}$, то швидкість передачі інформації $V = n_0 \cdot F_\psi$ (біт/с). У реальних системах зазвичай $f_0 \leq \frac{F_\psi}{3}$. Крок зміни фази $\frac{2\pi}{k_0}$.

Вимірювання проводяться через інтервал $\frac{\Delta t}{10} \dots \frac{\Delta t}{50}$, тобто на інтервалі одного символу проводиться від 10 до 50 вимірювань амплітуди квадратурних складових сигналу. При цьому, як правило, виникає тимчасова помилка за синхронізацією, тобто передбачуваний початок передачі нового символу на приймальній стороні зміщений на деяку величину θ , що не перевищує кілька інтервалів вимірювання. Початкова фаза передавальної сторони, як правило, невідома на приймальній стороні, і її також доводиться оцінювати.

Перерахуємо ще раз основні три величини, які змінюються у часі, отже, їх необхідно оцінювати у часі, тобто алгоритми демодуляції повинні адаптуватися до них:

- *Помилка по частоті.* Частота на передавальній і приймальній стороні не можуть бути зведені абсолютно. Тому після зняття високочастотної несучої на приймальній стороні ми маємо гармонійне коливання, а не константу.
- *Помилка по синхронізації.* Початок передачі нового символу на приймальній стороні може відрізнитися від реального початку.
- *Початкова фаза сигналу.* У процесі поширення сигнал набуває випадковий зсув по фазі.

Використовується декілька видів фазової маніпуляції. Найбільш часто використовується *диференціальна фазова маніпуляція*. У цьому варіанті фаза сигналу змінюється не на значення фази переданого символу, а на значення різниці фаз, переданого і попереднього символу. Наприклад, якщо двічі потрібно передати один і той же символ, то фаза взагалі не змінюється.

При $k_0 = 2$ говорять про двійкову фазову маніпуляцію, використовуються при цьому абревіатура **BPSK (binary phase shift keying)**. При $k_0 = 4$ говорять про квадратичну фазову маніпуляцію, використовується при цьому абревіатура **QPSK (quadrature phase shift keying)**. При подальшому збільшенні k_0 його значення приписують перед абревіатурою **PSK** (8-PSK, 16-PSK і т. ін.).

Будемо припускати, що при передачі сигналу на нього діє адитивна гаусова (нормальна) перешкода, тобто на приймальній стороні вимірюється сиг-

нал $y(t) = u_s(t) + u_p$, для простоти припустимо, що обидві квадратури перешкоди однаково розподілені і незалежні:

$$u_p = x_p + jy_p,$$

$$x_p, y_p \in N(0, \sigma_p^2).$$

Щодо тимчасової залежності відліків перешкоди можна розглядати два часто використовуваних варіанти. Якщо ми припустимо незалежність часових відліків перешкоди, то одержимо так званий *білий шум*.

Як правило, реальні сигнали мають тимчасову кореляцію, тобто кожне наступне значення сигналу залежить від його попередніх значень. Найбільш часто використовуваною моделлю таких сигналів є *марківські процеси* або, у дискретному випадку, *марківські послідовності*. Найбільш простими є марківські послідовності *першого порядку*, коли наступне значення залежить тільки від попереднього і не залежить від більш ранніх значень. Для нормального розподілу при цьому існують дуже прості формули для їх генерації:

$$x_{i+1} = \rho x_i + \sigma_p \sqrt{1 - \rho^2} \xi_i.$$

Тут ρ – заданий коефіцієнт кореляції між сусідніми відліками сигналу, тобто $\rho = \rho(x_i, x_{i+1})$, σ_p^2 – дисперсія (потужність перешкоди), $\xi_i \in N(0; 1)$ – нормально розподілена випадкова величина з нульовим середнім і одиничною дисперсією.

Марківські послідовності для різних коефіцієнтів кореляції реалізується наступним текстом:

```
n=2000;
kpr=3;
ro=[0,0.99,0.999];
sro=sqrt(1-ro.^2);
x=zeros(kpr,n);
x(:,1)=randn(kpr,1);
for i=1:kpr
    for j=2:n
        x(i,j)=ro(i)*x(i,j-1)+sro(i)*randn;
    end
end
figure
x(1,:)=x(1,:)+4;
plot(1:n,[x(1,:);x(2,:);x(3,:)]);
title('Марківські процеси');
legend('ro=0','ro=0.99','ro=0.999');
```

Одна з реалізацій виглядає так (рисунок 4.8):



Рис. 4.8. Реалізації білого шуму і марківського процесу для двох значень часової кореляції

При нульовому коефіцієнті кореляції сусідніх часових відліків ми отримуємо білий шум (на графіку реалізація білого шуму піднята для виключення накладення на два інших процеси). При збільшенні коефіцієнта кореляції дисперсія відліків щодо попередніх знижується, при цьому загальна дисперсія залишається без змін.

Відношення сигнал/перешкода в децибелах обчислюється за наступною формулою:

$$q = 20 \lg \frac{A}{\sigma_p}.$$

На приймальній стороні прийнятий сигнал завжди пропускається через формуючий фільтр.

4.4. Результати порівняння реального супутникового сигналу з моделлю QPSK-сигналів

Розглянута модель FSK-маніпулювання реалізована для QPSK-сигналів у програмі **QPSKdemod** одного з авторів цього посібника. Програма також дозволяє обробляти записи реальних сигналів. Розглянемо спочатку, як виглядає реальний супутниковий сигнал з QPSK-маніпулюванням у різних графічних поданнях. На рис. 4.9 по фазовому портрету ми бачимо, що дійсно записаний сигнал має QPSK-маніпуляцію.

Оцінено основні параметри сигналу, без адаптації за якими неможливе правильне декодування. Некомпенсований зсув по частоті досить великий і складає $\sim 18,5$ Гц. Наявність зміщення по частоті добре видно на графіку фази 4-го ступеня сигналу (рисунок 4.10). Оцінка швидкості зміни фази, тобто зсув по частоті, отримано лінійним методом найменших квадратів (МНК) після розгортання фази з інтервалу $[-180^\circ, 180^\circ]$ на всю пряму.

Зазначимо, що для оцінки швидкості зміни фази PSK-сигналів, як правило, використовують ступінь сигналу, що відповідає кількості розбивань фазового простору, оскільки це призводить до усунення маніпуляції. Дійсно, наприклад, для розглянутого QPSK-сигналу:

$$u_s^4(t) = e^{4j\left(2\pi f_0 t + \phi + k\frac{2\pi}{4}\right)} = e^{j(2\pi(4f_0)t + 4\phi + 2\pi k)} = e^{j(2\pi(4f_0)t + 4\phi)}.$$

Потрібно тільки не забути, виміряну таким чином швидкість зміни фази розділити на 4.

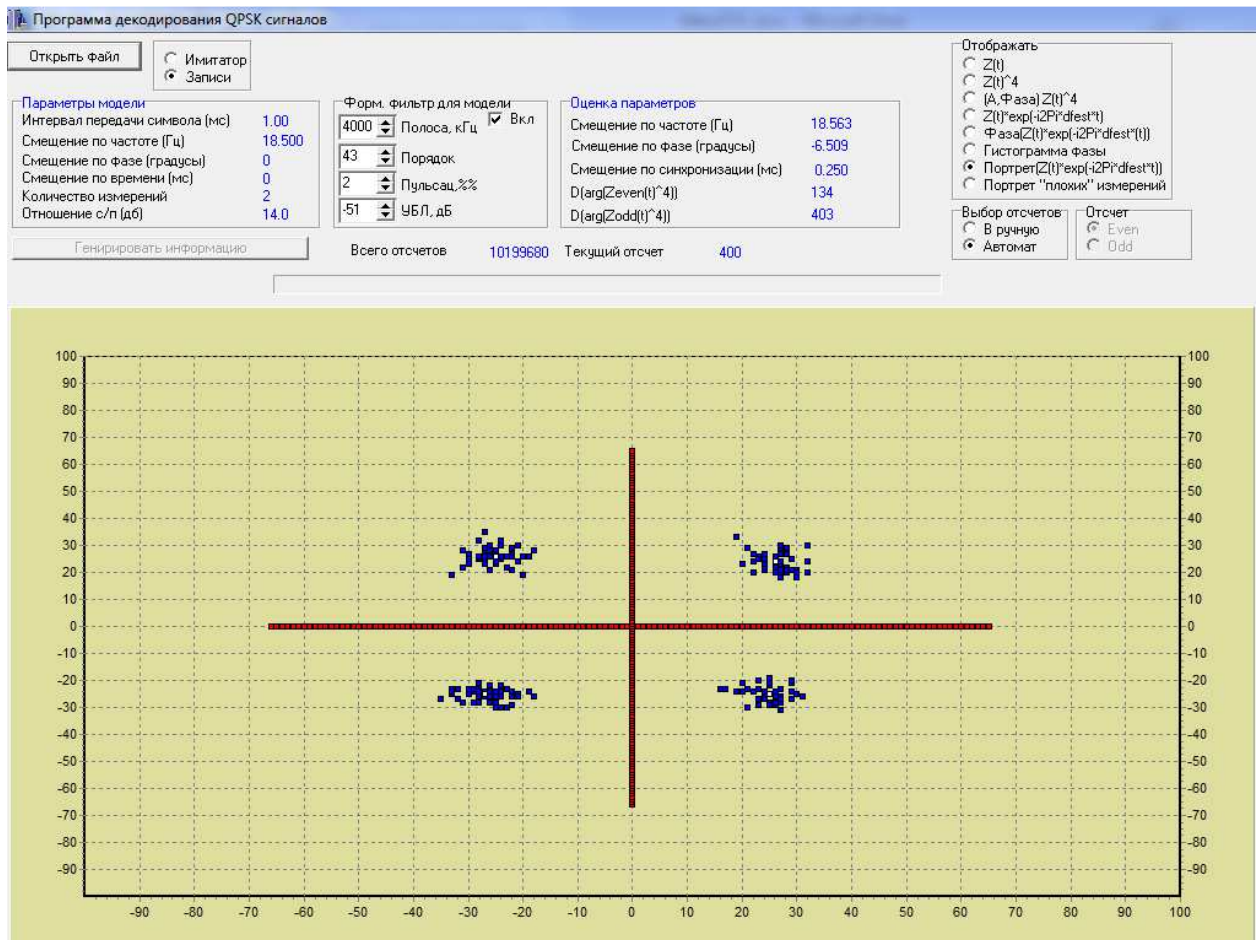


Рис. 4.9. Фазовый портрет реального спутникового сигнала з QPSK-маніпуляцією

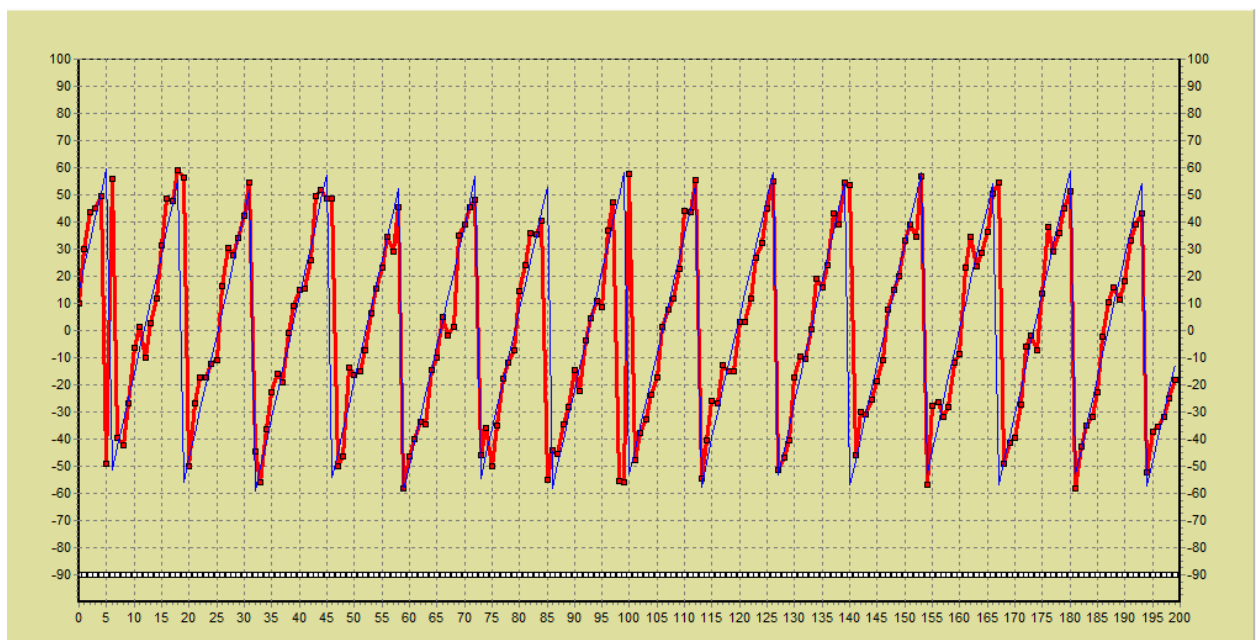


Рис. 4.10. Зміна фази в часі четвертого ступеня вихідного сигналу супутника та її апроксимація лінійним МНК

Квадратури вхідного сигналу та їх 4-а ступінь зображені на рис. 4.11 і 4.12:

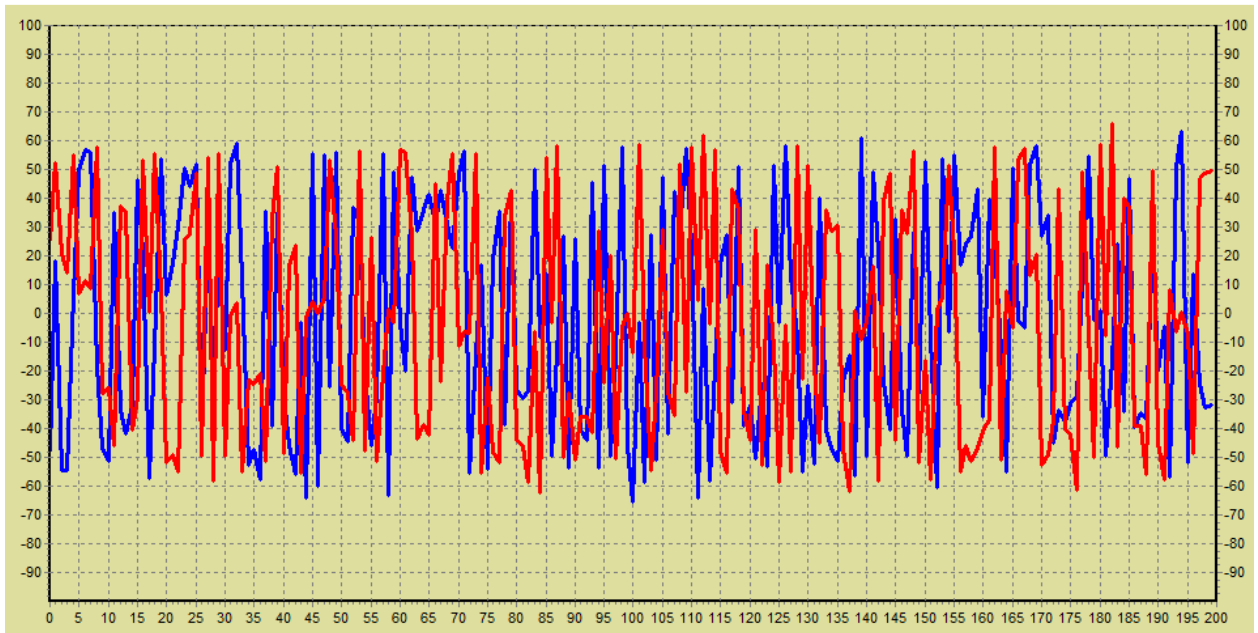


Рис. 4.11. Квадратури вихідного сигналу супутника

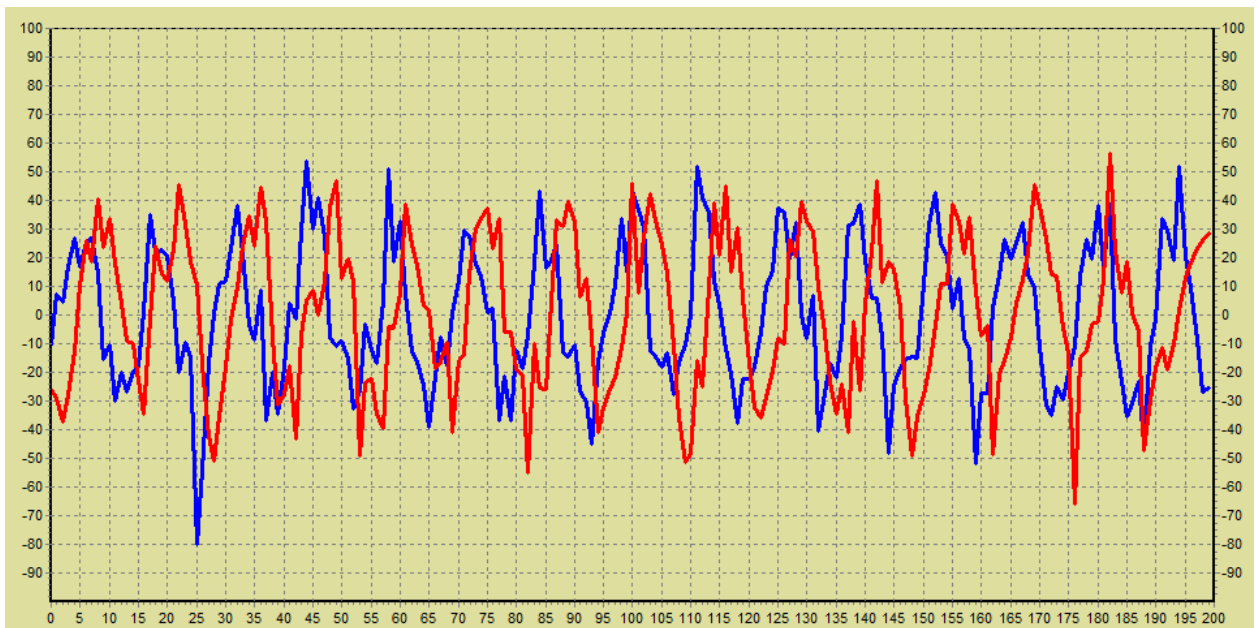


Рис. 4.12. 4-а ступінь квадратур вихідного сигналу супутника

Наведені вибірки з сигналу взяті на інтервалі хорошого співвідношення сигнал/перешкода, на цьому ж запису є інтервали часу з зменшенням рівня сигналу відносно перешкоди. Фазовий портрет при цьому виглядає так (рис. 4.13):

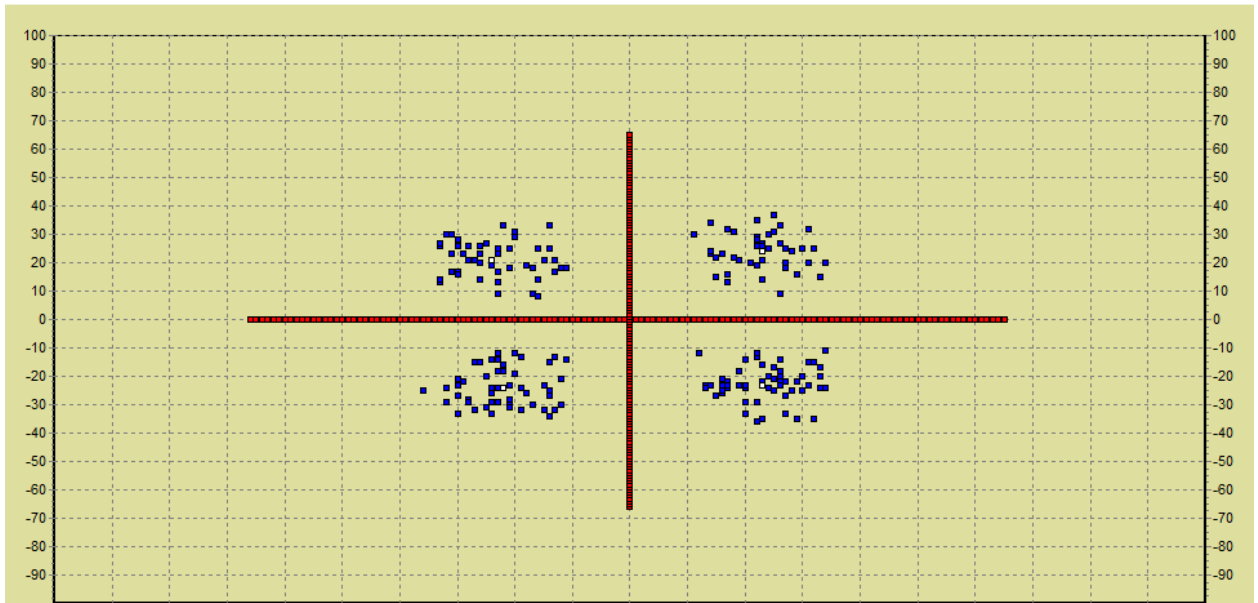


Рис. 4.13. Фазовий портрет супутникового сигналу на інтервалі зменшення співвідношення сигнал/перешкода

Тепер порівняємо реальну інформацію з тою, що генерується, за розглянутою вище простою математичною моделлю. Задамо в програму значення трьох основних параметрів сигналу (зсув по частоті, фазі і синхронізації), які були оцінені за реальним сигналом. На рисунку 4.14 наведено фазовий портрет модульованого сигналу і оцінки його параметрів при встановленому співвідношенні сигнал/перешкода 12 дБ.

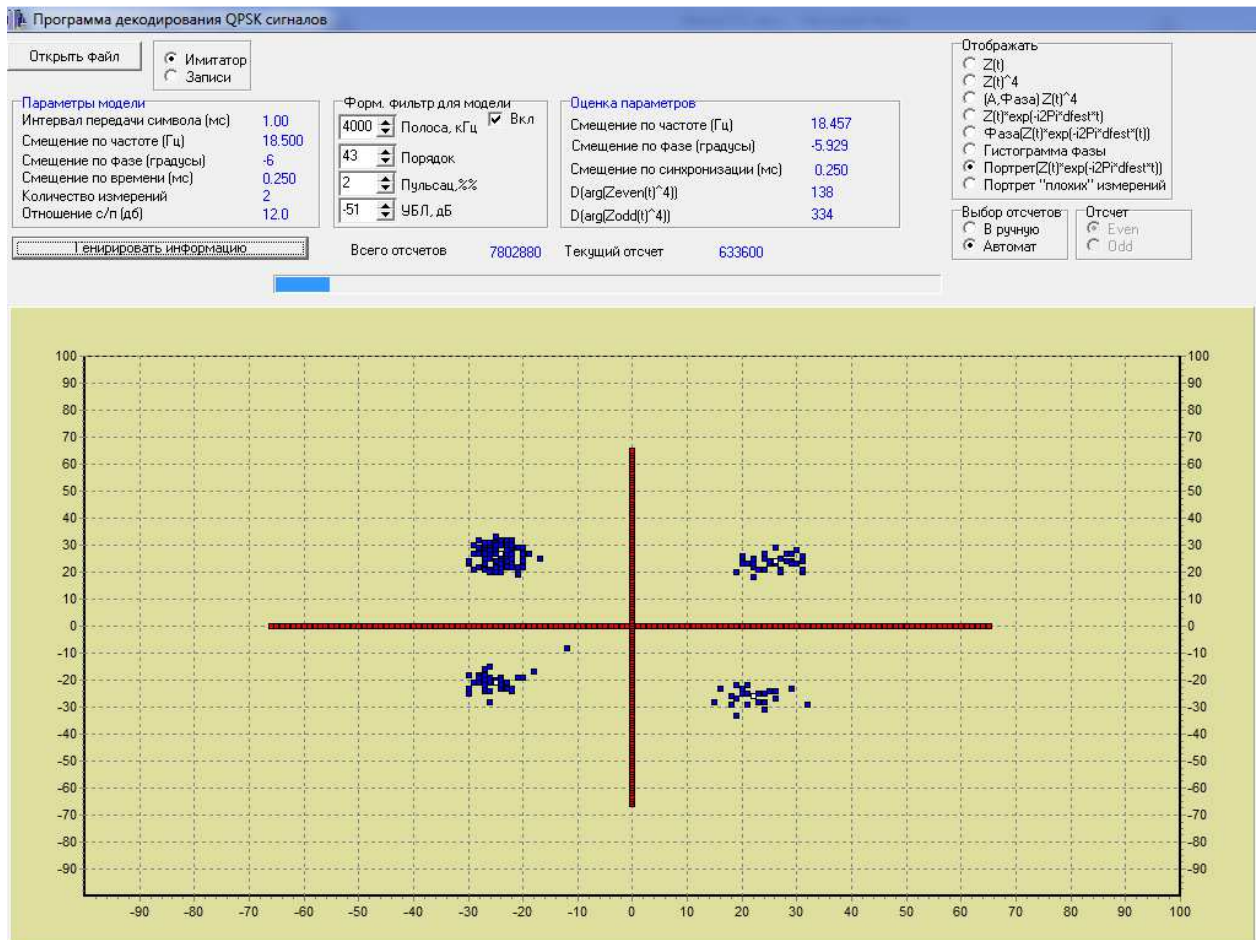


Рис. 4.14. Фазовый портрет модельованого QPSK сигналу з результатами оцінки параметрів

Як бачимо, всі параметри оцінені з хорошою точністю. Зсув по частоті оцінено з точністю до **0,05 Гц**, по фазі — **0,1°**, зсув по синхронізації — **точно**. Задане в моделі співвідношення сигнал/перешкода **12 дБ** за розмитості фазового портрета також відповідають наведеному інтервалу вимірювання реального сигналу. Крім візуального порівняння розмитості можна порівняти обчислювану у програмі дисперсію відліків, яка склала за реальним сигналом **134** умовні одиниці, по моделі — **138**. Наведемо тепер аналоги рисунків 4.10, 4.11, 4.12 для модельованого сигналу (рис. 4.15, 4.16, 4.17):

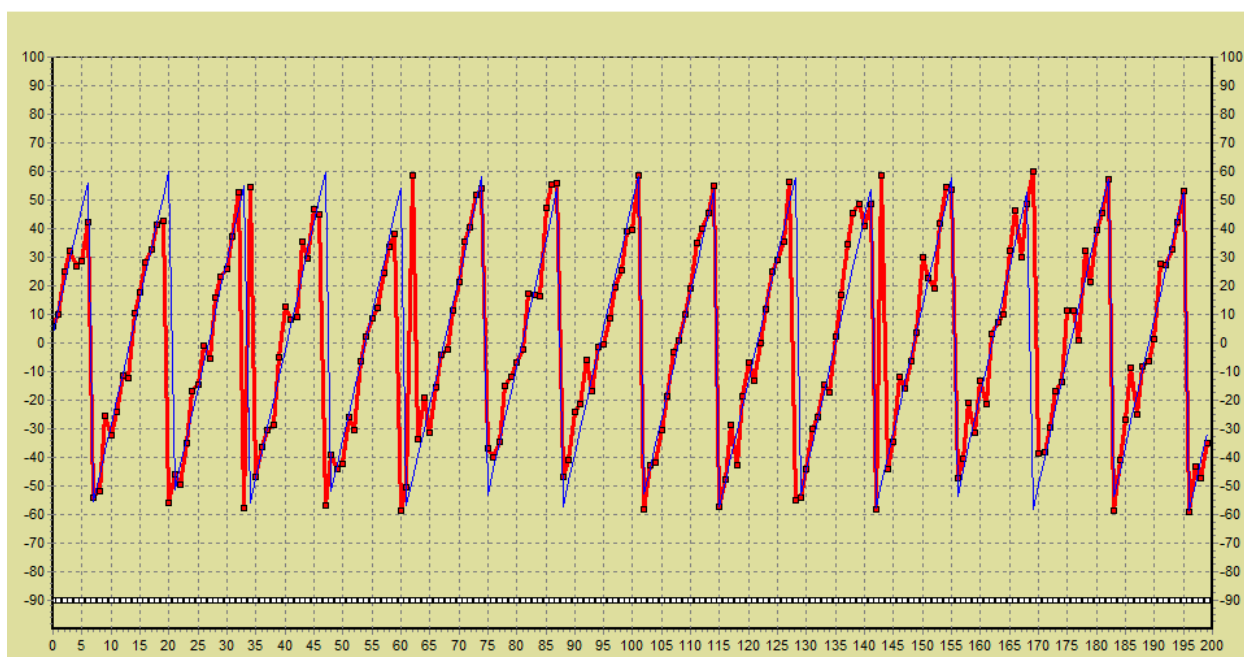


Рис. 4.15. Зміна фази під час четвертого ступеня вихідного сигналу моделі та її апроксимація МНК

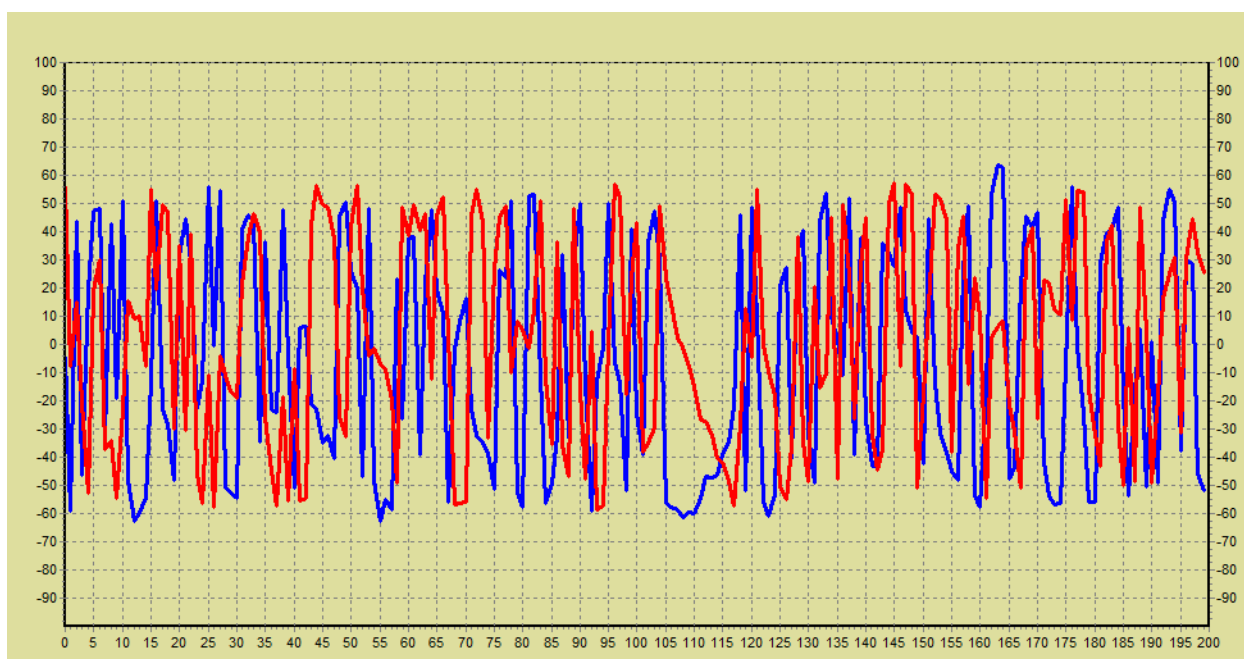


Рис. 4.16. Квадратури вихідного сигналу моделі

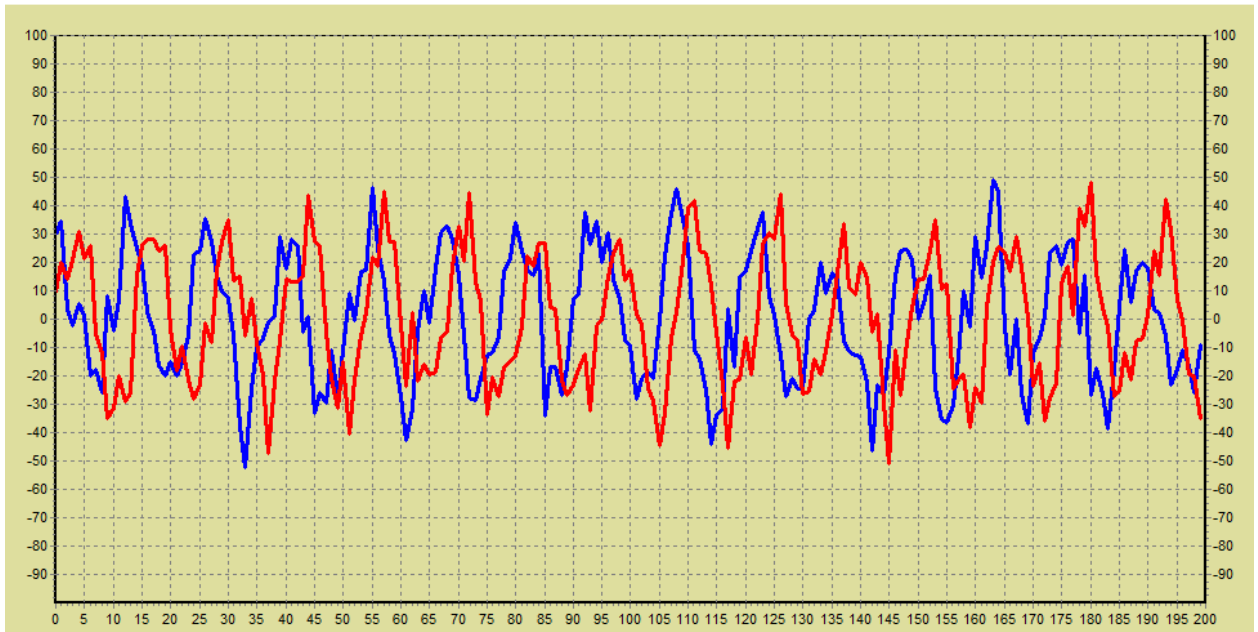


Рис. 4.17. 4-а ступінь квадратур вихідного сигналу моделі

Зміна фази в часі сигналу супутника і модельованого сигналу (рис. 4.10 і 4.15) практично не відрізняються. Квадратури сигналів відрізняються (рис. 4.11 і 4.16), однак треба врахувати, що маніпуляція цих квадратур проведена різними послідовностями символів. Після зняття QPSK-маніпуляції зведенням квадратур в четверту ступінь (рис. 4.12 та 4.17), схожість реально-го супутникового сигналу з модельованим сигналом досить висока.

Наведені дані дозволяють зробити висновок про хорошу відповідність розглянутої простої математичної моделі PSK-сигналів реальним сигналам. Вихідна інформація моделі може використовуватися для відпрацювання алгоритмів демодуляції. На закінчення відзначимо, що в програмі **QPSKdemod** реалізована можливість обробляти і моделювати Offset QPSK- і Differential QPSK-сигнали. В режимі Offset QPSK квадратури зміщують відносно один одного на половину інтервалу передачі символу, що виключає зміну фази на 180° , яка призводить до завмирання сигналу. В режимі Differential QPSK (відносна або диференціальна фазова маніпуляція) фаза сигналу змінюється на значення, що дорівнює різниці фаз між поточним символом і попереднім.

4.5. Завдання для лабораторних, курсових робіт і дипломних проектів

Пропоновані завдання повинні бути реалізовані на мові C++.

1. Розробка спектр-аналізатору комплексного сигналу.
2. Розробка функції обчислення розгорнутої фази комплексного сигналу.
3. Оцінка частотного зсуву комплексного сигналу за його спектром з урахуванням декількох відліків в околиці спектрального відліку з максимальною амплітудою.
4. Оцінка частотного зсуву комплексного сигналу по швидкості зміни розгорнутої фази сигналу.
5. Розробка функцій для відображення на екрані різних характеристик комплексного сигналу (сигнальне сузір'я, квадратури, амплітуда, фаза).
6. Розробка моделі PSK-сигналів з моделюванням зміщення по частоті, фазі і порушенням прив'язки до межі бітів.
7. Розробка демодулятора PSK-сигналів та його відпрацювання за моделюваною і реальною супутниковою інформацією.
8. Розробка моделі APSK-сигналів з моделюванням зміщення по частоті, фазі і порушенням прив'язки до межі бітів.
9. Розробка демодулятора APSK-сигналів та його відпрацювання за моделюємою інформацією.
10. Розробка функції для скремблювання та дескремблювання бітової інформації.
11. Реалізація табличного і матричного перемежування бітової інформації.
12. Реалізація турбо-коду P2C3.
13. Реалізація повної моделі перетворень бітового потоку при підготовці його до передачі в супутникових каналах зв'язку у відповідності зі стандартом IPOS.

ОСНОВНІ СКОРОЧЕННЯ

ADS-B	– Automatic Dependent Surveillance-Broadcast – радіомовне автоматичне залежне спостереження
AIS	– Automatic Identification System – автоматична ідентифікаційна система
APSK	– Amplitude and Phase Shift Keying – амплітудно-фазова маніпуляція
BER	– Bit Error Rate або Bit Error Ratio – коефіцієнт бітових помилок
BPSK	– Binary Phase Shift Keying – двійкова фазова маніпуляція
CPR	– Compact Position Reporting – компактний формат передачі розташування
CRC	– Cyclic Redundancy Check – циклічний надлишковий код
GMSK	– Gaussian Minimum Shift Keying – гаусівська двохпозиційна частотна маніпуляція з мінімальним зсувом
GNSS	– Global Navigation Satellite System – супутникова система навігації
GPS	– Global Positioning System – система глобального позиціонування
ICAO	– International Civil Aviation Organization – Міжнародна організація цивільної авіації
MMSI	– Maritime Mobile Service Identity – ідентифікатор морської рухомої служби
MSK	– Minimal Shift Keying – маніпуляція з мінімальним зсувом
PSK	– Phase Shift Keying – фазова маніпуляція
QPSK	– Quadrature Phase Shift Keying – квадратурна фазова маніпуляція
SDR	– Software Defined Radio – програмно-конфігурована радіосистема
АЦП	– аналого-цифровий перетворювач
АЧХ	– амплітудно-частотна характеристика
KX	– короткі хвилі
НІХ	– нескінченна імпульсна характеристика
с/п	– відношення сигнал/перешкода
СІХ	– скінченна імпульсна характеристика
СКВ	– середнє квадратичне відхилення
ФНЧ	– фільтр низьких частот
ШПФ	– швидке перетворення Фур'є

ЛІТЕРАТУРА

1. Описание элементов управления и настроек SDRSharp. [Интернет ресурс]: <http://rtl-sdr.ru/page/opisanie-elementov-upravlenija-i-nastroek-sdrsharp>
2. Robert W. Stewart, Kenneth W. Barlee, Dale S.W. Atkinson, Louise H. Crockett. Software Defined Radio using MATLAB & Simulink and the RTL-SDR. // University of Strathclyde, Glasgow, Scotland, UK, 2015, 674p.
3. Junzi Sun. ADS-B Decoding Guide. Release. 2017, 34 p.
4. The 1090MHz Riddle. The book about decoding Mode-S and ADS-B data. [Интернет ресурс]: <http://mode-s.org/decode/>
5. Ихсанов Ш.М. Методичні вказівки для вивчення теорії та виконання лабораторних робіт з дисципліни «Теорія електричного зв'язку» / Ш.М. Ихсанов // Миколаїв: НУК ім. адм. Макарова, 2016, 99 с.
6. Теория и практика цифровой обработки сигналов. Сигналы с минимальной частотной манипуляцией (Minimum Shift Key - MSK). [Интернет ресурс]: www.dsplib.ru/content/msk/msk.html.
7. Теория и практика цифровой обработки сигналов. MSK-сигналы с гауссовой огибающей (GMSK). [Интернет ресурс]: www.dsplib.ru/content/gmsk/gmsk.html.
8. Ихсанов Ш.М. Методичні вказівки для вивчення теорії та виконання лабораторних робіт з дисципліни «Адаптивна обробка сингалів». — Миколаїв : НУК ім. адм. Макарова, 2018. — 69 с.
9. U.S. Department of Homeland Security. Navigation center. Class A AIS position report (messages 1, 2, and 3). [Интернет ресурс]: www.navcen.uscg.gov/?pageName=AIMessagesA
10. U.S. Department of Homeland Security. Navigation center. AIS standard class B equipment position report (message 18). [Интернет ресурс]: www.navcen.uscg.gov/?pageName=AIMessagesB
11. Telecommunications Industry Association Standard. IP over Satellite (IPOS). Revision of TIA-1008-A. — USA, May 2006. — pages 247.
12. Ихсанов Ш.М. Генератор выхода информационного канала, использующего помехоустойчивый параллельно-каскадный сверточный код P2C3. Тези доповідей на міжнародну науково-практичну конференцію "Компьютерные науки: образование, наука, практика" — Миколаїв : НУК ім. адм. Макарова, 2012. — С. 79—81.