ПРИЛОЖЕНИЕ А
Программа модели

```
tic;


% Экспериментальная величина (для построения нескольких кривых на графике)
exp = [0];
expTitle = 'Надпись на легенде...';
expNumber = length(exp);

figure;

% Информационая последовательность
informationLength = 100;        % Длина информационного сигнала
lowLevel = -1;                  % Низкий уровень двоичного сигнала
highLevel = 1;                  % Высокий уровень двоичного сигнала
informationSequence = getRandomBinarySequence(lowLevel, highLevel, informationLength);
packetNumber = 1000;            % Количество замеров

for m = 1:expNumber
    %M-последовательность
    bpow = 4;       % Степень двойки, определяющая длину последовательности (6 -> 2^6 =
64)
    %modulatingSequence = mseq(2, bpow, 0);

    modulatingSequencePositives = [1,2,3,4,7,11,13];
    modulatingSequenceNegatives = [5,6,8,9,10,12];
    modulatingSequence = getBinarySequence(modulatingSequencePositives,
modulatingSequenceNegatives, highLevel, lowLevel);

    SNR = 0.01:0.02:0.3;                % Сигнал/Шум
    %SIR = 0.1:0.01:1;                  % Сигнал/Помеха
    %Frel = 0:0.02:0.5;                 % Частота относительно Fpm

    % Значения оси x
    xval = SNR;

    base = length(modulatingSequence);  % База сигнала
    Fs = 150;                           % Частота дискретизации
    T = 1/Fs;                           % Период дискретизации
    L = informationLength*base*Fs;      % Длина сигнала
    time = (0:L-1)*T;                   % Вектор времени

    % ФМ-сигнал
    PMAmpl = 1;     % Амплитуда ФМ
    fpm = 10;       % Частота ФМ

    % Шум
    mu = 0;

    % Помеха
    Ui = 1;         % Амплитуда гармонической помехи
    fi = fpm + 0.1; % Частота гармонической помехи
    phi = 0;        % Фаза гармонической помехи

    % Детектор
    Uref = 1;       % Амплитуда опорного сигнала
    fref = fpm;     % Частота опорного сигнала
    phref = 0;      % Фаза опорного сигнала

    % Ограничитель
    Amax = 1;           % Максимальная амплитуда ограничителя
    Amin = -1;          % Минимальная амплитуда ограничителя
    Lmax = 0.3;         % Верхний порог
```

```
    Lmin = -0.3;        % Нижний порог

    % ФНЧ
    fpass = 5;
    fstop = fpass + 5;
    Apass = 1;
    Astop = 60;

    % Интегратор
    threshold = 0;

    modulatingSignal = getPeriodicalSignal(modulatingSequence, time);
    informationSignal = getSignal(informationSequence, time, base);
    modulatedSignal = informationSignal.*modulatingSignal;
    PMSignal = getPMSignal(modulatedSignal, PMAmpl, fpm, time);
    lpf = getLPF(fpass, fstop, Apass, Astop, Fs);

    detectorHarmonic = getHarmonic(Uref, fref, phref, time);

    results = zeros(1, length(xval));
    for i = 1:length(xval)
        % Для Сигнал/Шум
        sigma = PMAmpl/SNR(i);

        % Для частоты относительно Fpm (С/Ш = 0.15)
        %sigma = PMAmpl/0.15;
        %fi = fpm + Frel(i);

        % Для Сигнал/Помеха
        %Ui = PMAmpl/SIR(i);

        interferenceHarmonic = getHarmonic(Ui, fi, phi, time);
        packetsResults = zeros(length(informationSequence), packetNumber);
        parfor j = 1:packetNumber
            noise = getNoise(mu, sigma, time);

            % Сигнал + шум + помеха
            interferenceSignal = PMSignal + noise + interferenceHarmonic;

            % Сигнал + помеха
            %interferenceSignal = PMSignal + interferenceHarmonic;

            detectorSignal = interferenceSignal.*detectorHarmonic;
            LPFSignal = filter(lpf, detectorSignal);

            % Ограничитель включен
            limiterSignal = getLimiterSignal(LPFSignal, Amax, Amin, Lmax, Lmin);

            % Ограничитель выключен
            %limiterSignal = LPFSignal;

            multiplierSignal = limiterSignal.*modulatingSignal;
            integratorMatrix = reshape(multiplierSignal, [Fs*base informationLength]);
            integratorSequence = trapz(integratorMatrix);
            receivedSequence = applyThreshold(integratorSequence, highLevel, lowLevel,
threshold);

            diffVector = informationSequence - receivedSequence;

            packetsResults(:,j) = diffVector;
        end
```

```
        negative = length(find(packetsResults));
        Perr = negative/numel(packetsResults);
        results(i) = Perr;
    end

    semilogy(xval, results);
    hold on;
end

legendStrings = cell(1, expNumber);
for i = 1:expNumber
    legendStrings{i} = num2str(exp(i));
end

lgd = legend(legendStrings);
title(lgd, expTitle);
grid on;

toc

function result = applyThreshold(signal, highLevel, lowLevel, threshold)
    result = zeros(1, length(signal));
    for i = 1:length(result)
        value = signal(i);
        if value >= threshold
            result(i) = highLevel;
        else
            result(i) = lowLevel;
        end
    end
end

function result = getBinarySequence(highLevelNumbers, lowLevelNumbers, highLevel,
lowLevel)
    sequenceLength = length(highLevelNumbers) + length(lowLevelNumbers);
    result = zeros(1, sequenceLength);
    for i = 1:sequenceLength
        if ismember(i, highLevelNumbers)
            result(i) = highLevel;
        elseif ismember(i, lowLevelNumbers)
            result(i) = lowLevel;
        end
    end
end

function result = getHarmonic(amplitude, frequency, phase, range)
    result = zeros(1, length(range));
    for i = 1:length(result)
        result(i) = amplitude*cos(2*pi*frequency*range(i) + phase);
    end
end

function result = getLimiterSignal(signal, Amax, Amin, Lmax, Lmin)
    result = zeros(1, length(signal));
    for i = 1:length(result)
        n = signal(i);
        if n >= Lmax
            result(i) = Amax;
        elseif n <= Lmin
            result(i) = Amin;
        else
            result(i) = n;
```

```
        end
    end
end

function Hd = getLPF(Fpass, Fstop, Apass, Astop, Fs)
%FILTERDESIGN Returns a discrete-time filter object.

% MATLAB Code
% Generated by MATLAB(R) 9.12 and Signal Processing Toolbox 9.0.
% Generated on: 06-Oct-2022 12:43:36

% Butterworth Lowpass filter designed using FDESIGN.LOWPASS.

% All frequency values are in kHz.
%Fs = 300;  % Sampling Frequency

%Fpass = 1;           % Passband Frequency
%Fstop = 3;           % Stopband Frequency
%Apass = 1;           % Passband Ripple (dB)
%Astop = 60;          % Stopband Attenuation (dB)
match = 'passband';  % Band to match exactly

% Construct an FDESIGN object and call its BUTTER method.
h  = fdesign.lowpass(Fpass, Fstop, Apass, Astop, Fs);
Hd = design(h, 'butter', 'MatchExactly', match);

% [EOF]

function result = getNoise(mu, sigma, range)
    result = normrnd(mu, sigma, size(range));
end

function result = getPeriodicalSignal(sequence, range)
    result = zeros(1, length(range));
    sLength = length(sequence);
    for i = 1:length(range)
        tIndex = floor(range(i));
        sIndex = mod(tIndex, sLength) + 1;
        result(i) = sequence(sIndex);
    end
end

function result = getPMSignal(signal, ampl, frequency, range)
    result = zeros(1, length(signal));
    for i = 1:length(signal)
        result(i) = ampl*signal(i)*cos(2*pi*frequency*range(i));
    end
end

function result = getRandomBinarySequence(lowLevel, highLevel, slength)
    result = randi([0 1],1,slength);
    for i = 1:length(result)
        k = result(i);
        if (k == 1)
            result(i) = highLevel;
        else
            result(i) = lowLevel;
        end
    end
end

function result = getSignal(sequence, range, scale)
```

```
    result = zeros(1, length(range));
    for i = 1:length(range)
        index = floor(range(i)/scale) + 1;
        result(i) = sequence(index);
    end
end
```

ПРИЛОЖЕНИЕ Б
Графические материалы

Анализ влияния помех на прием шумоподобного сигнала путем математического моделирования

Магистерская квалификационная работа

Выполнил: ст. гр. РТм-21-1 Чередниченко Алексей Романович

Руководитель: проф. Антипов И. Е.

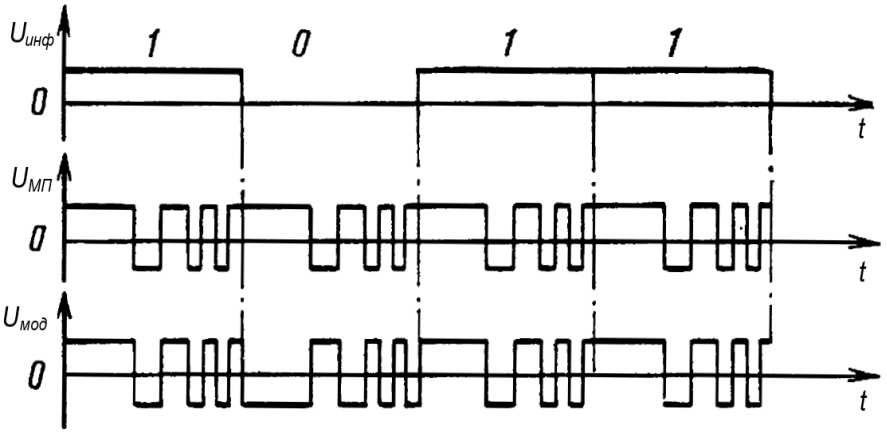
Харьков 2022

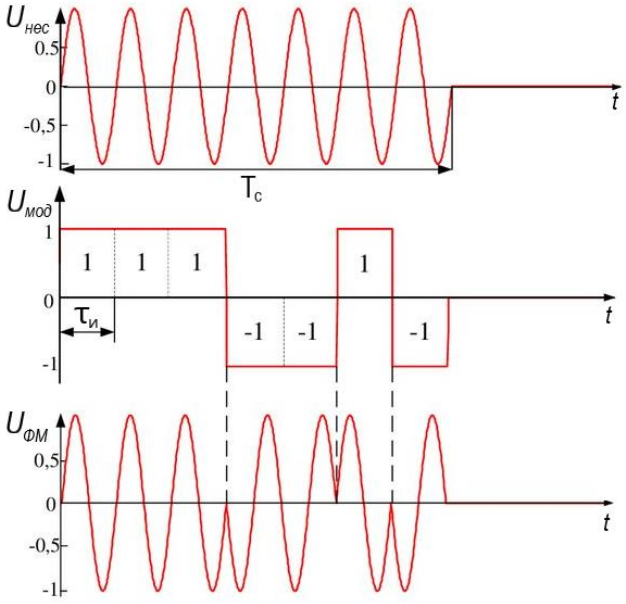
ФМ ШПС и его свойства

ШПС – сигнал, у которого база много больше единицы (*B = FT >> 1*)

ШПС обладают повышенной помехоустойчивостью и скрытностью относительно простого сигнала

Высокая помехоустойчивость систем, использующих ШПС, обусловлена так называемым коэффициентом усиления при обработке (*Кшпс = 2B*)



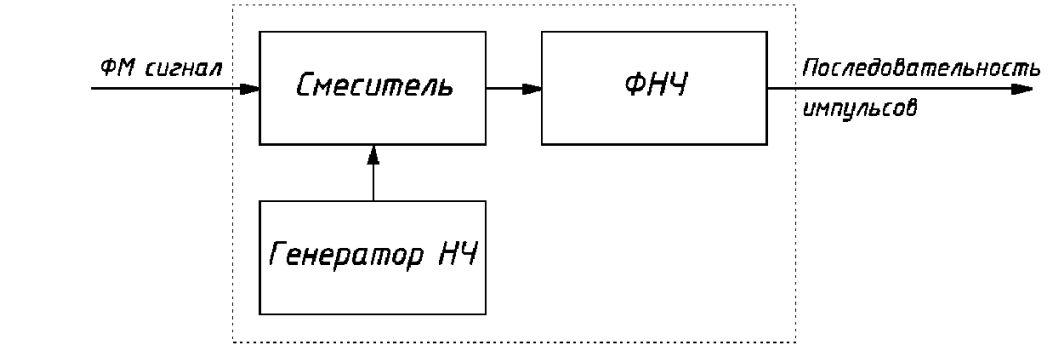


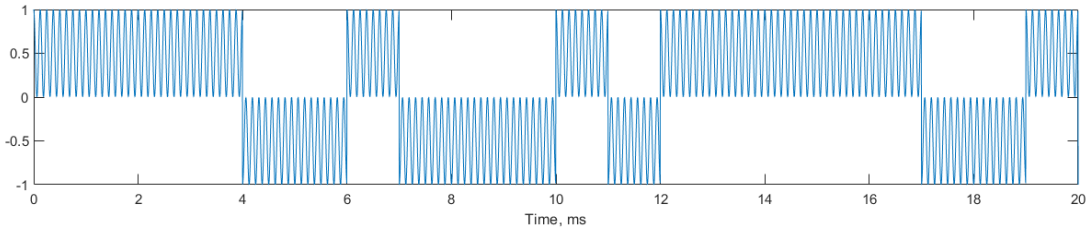
Воздействие узкополосной помехи на прием ШПС

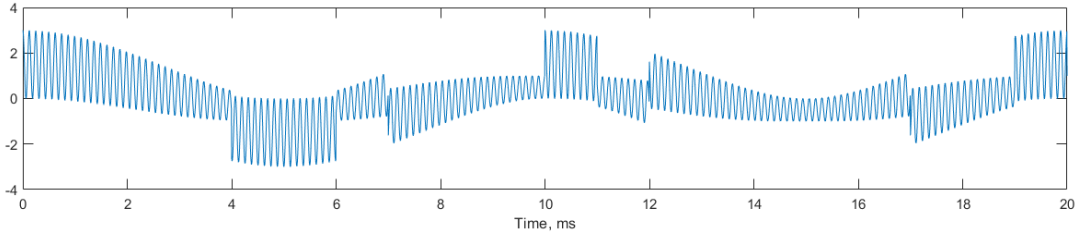
Узкополосная помеха – это гармонический сигнал, незначительно отличающийся по частоте от несущей ФМ ШПС.

Данный вид помехи может воздействовать только на ШПС, принимаемый схемой прямого преобразования, которая содержит фазовый детектор.

Узкополосная помеха вызывает эффект биений в фазовом детекторе, которые искажают принятые информационные символы.







Задание

Разработать модель формирования, приема и обработки ШПС с базой от 11 до 255 в условиях шумовой и/или узкополосной помехи.

В модели реализовать функции преобразования частоты, ФНЧ, амплитудного ограничения, корреляционной обработки принимаемого сигнала.

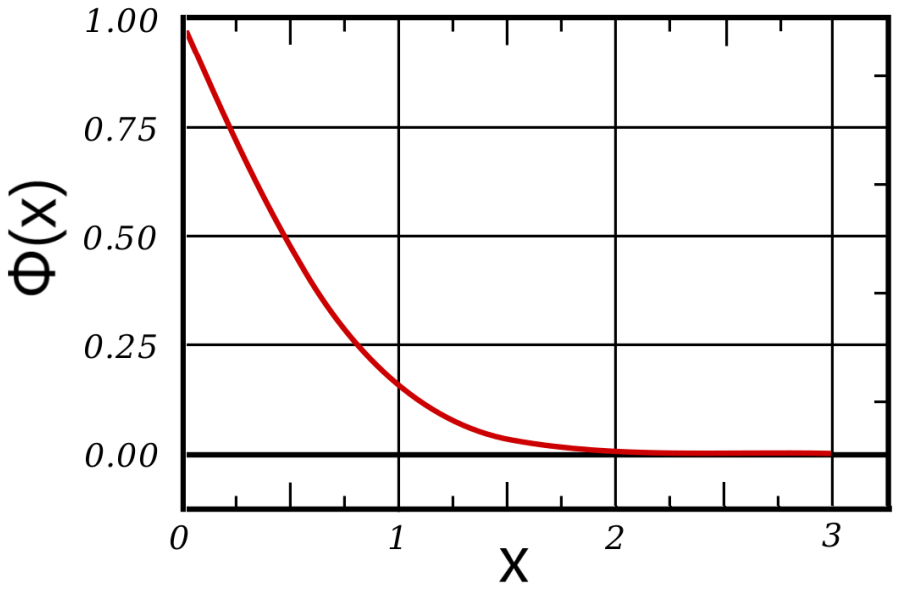
Провести моделирование приема и обработки ШПС для разных отношений сигнал/шум, сигнал/помеха.

Почему необходимо применять моделирование

Вероятность ошибки для ФМ сигнала с белым шумом:

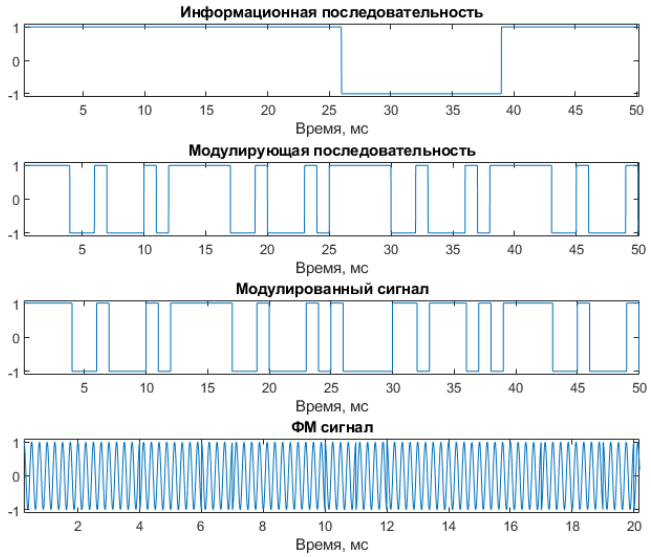
$$p_{\text{ФМ}} = 0.5\left[1 - \Phi\left(\sqrt{2E/N_0}\right)\right]$$

где E – энергия сигнала, N_0 – мощность шума на единицу полосы частот



Модель сигнала

Модель ФМ сигнала



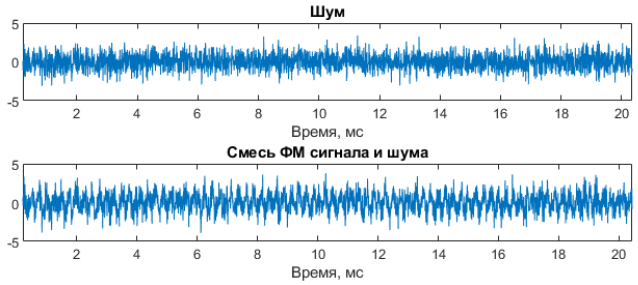
$U_{инф}(t)$ - набор псевдослучайных двоичных чисел длиной k

$U_{МП}(t)$ - задается вручную в виде последовательности, а затем периодически повторяется k раз

$U_{мод}(t) = U_{МП}(t) \times U_{инф}(t)$

$U_{ФМ}(t) = U_{мод}(t) \times cos(\omega_0 t)$

Модель шума

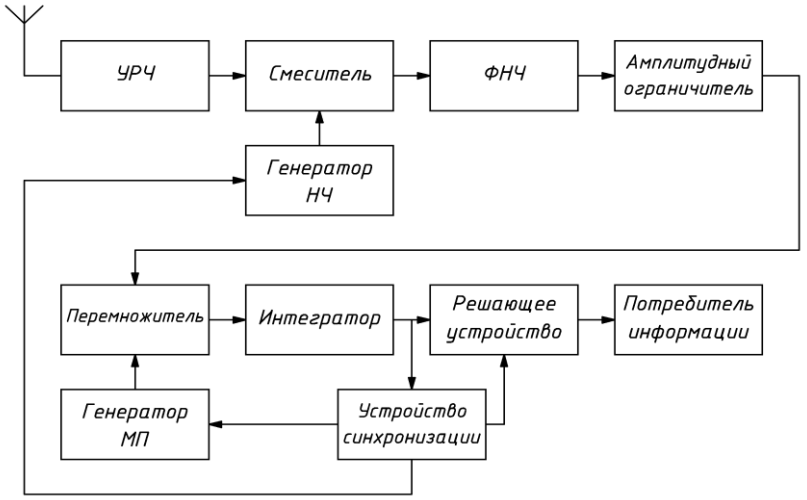


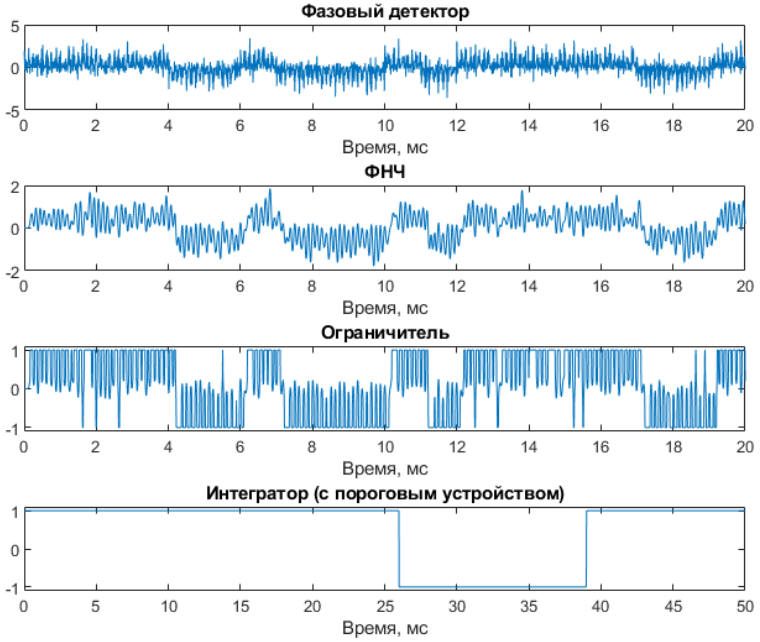
$n(t)$ - задается средствами MATLAB (функция *normrnd*); σ^2 определяется отношением сигнал/шум, $\mu_0 = 0$

$U(t) = n(t) + U_{ФМ}(t)$

Модель приемника

Модель приемника выполнена по схеме с прямым преобразованием частоты





$U_{фд}(t) = U_{оп}(t) \times U(t)$, $U_{оп}(t)$ – опорный сигнал

$U_{фнч}(t)$ – сигнал с ФНЧ (F_{pass} = 5 кГц, F_{stop} = 10 кГц)

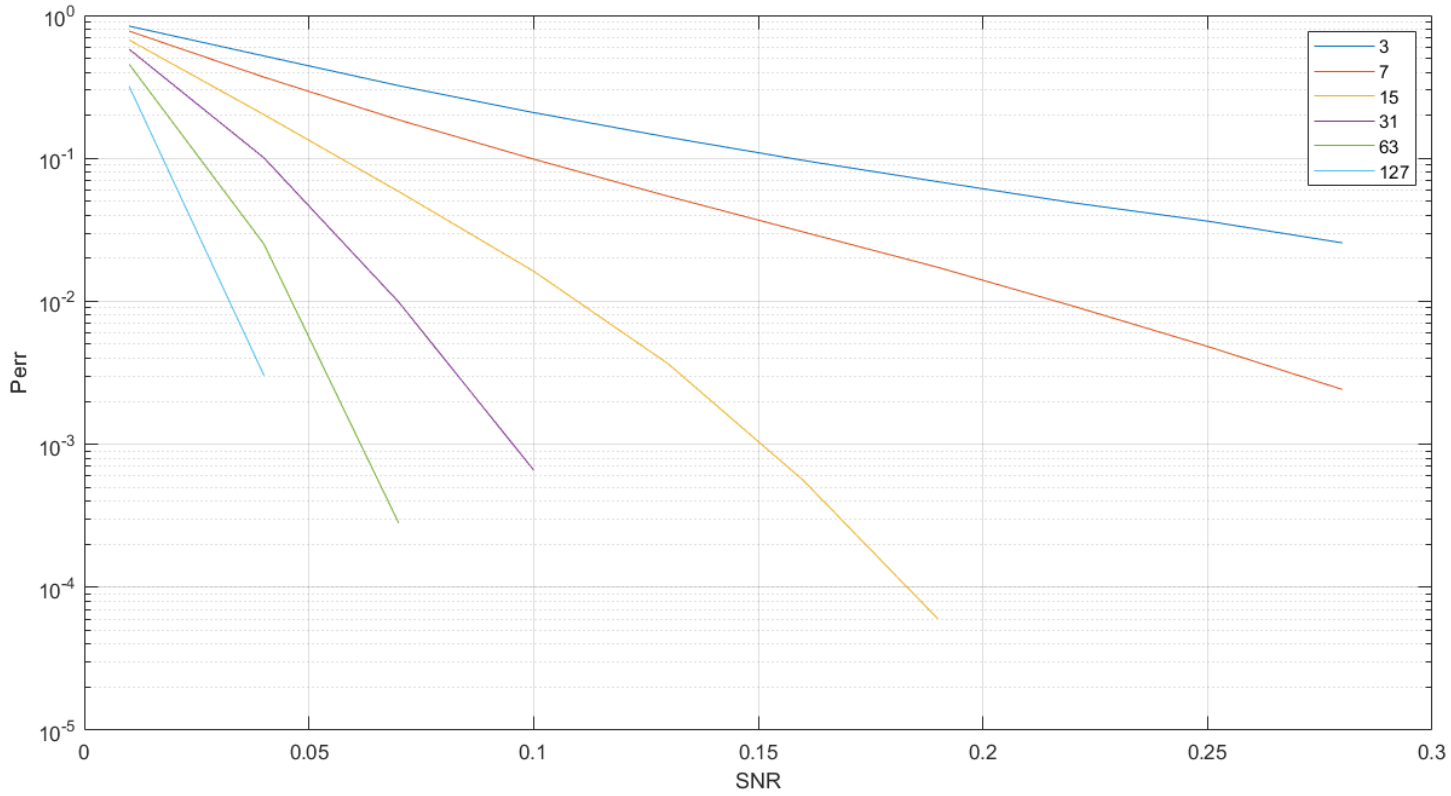
$$U_{вых}(t) = \begin{cases} 1 & \text{при} \quad U_{вх} \geq U_{max} \\ U_{вх} & \text{при} \quad U_{min} < U_{вх} < U_{max} \\ -1 & \text{при} \quad U_{вх} \leq U_{min} \end{cases}$$

$U_{max} = 0.3$
$U_{min} = - 0.3$

$U_{вых}(t)$ – полученный информационный сигнал

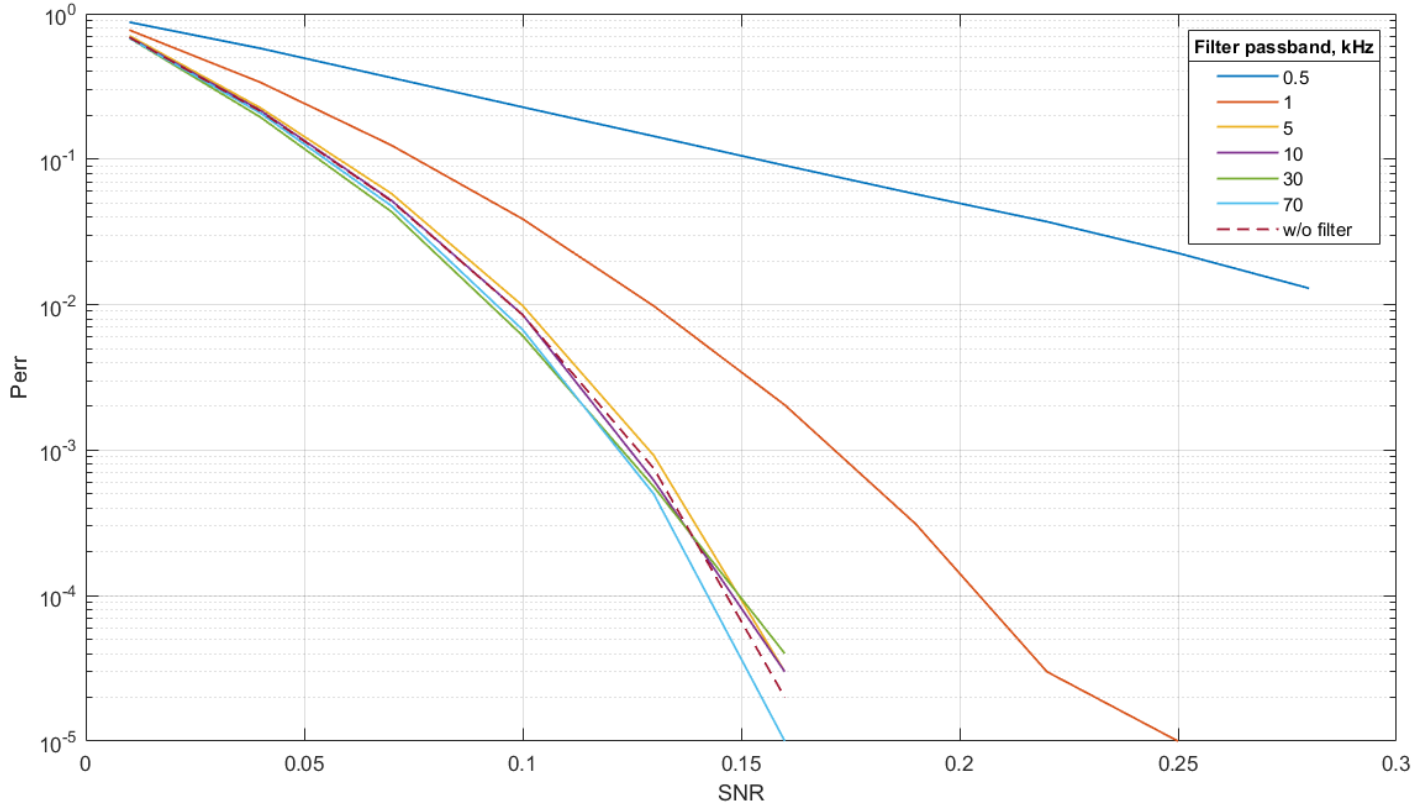
Результаты моделирования

Зависимость вероятности ошибки от базы ШПС



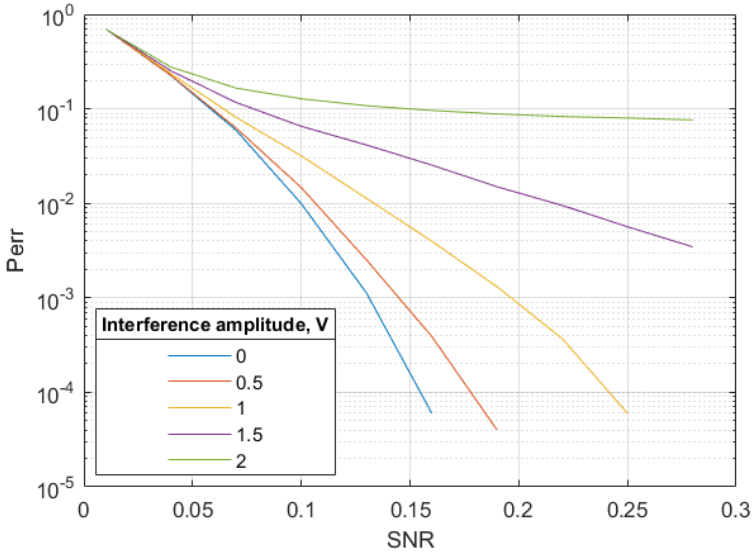
Результаты моделирования

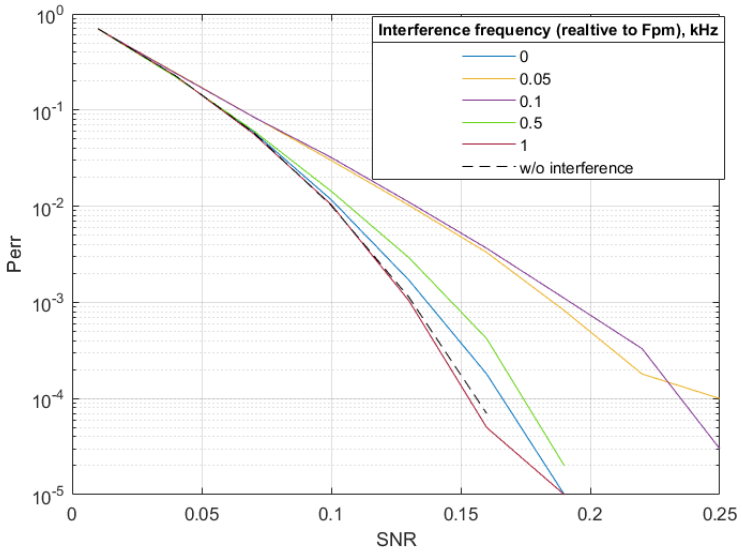
Зависимость вероятности ошибки от ширины полосы пропускания фильтра

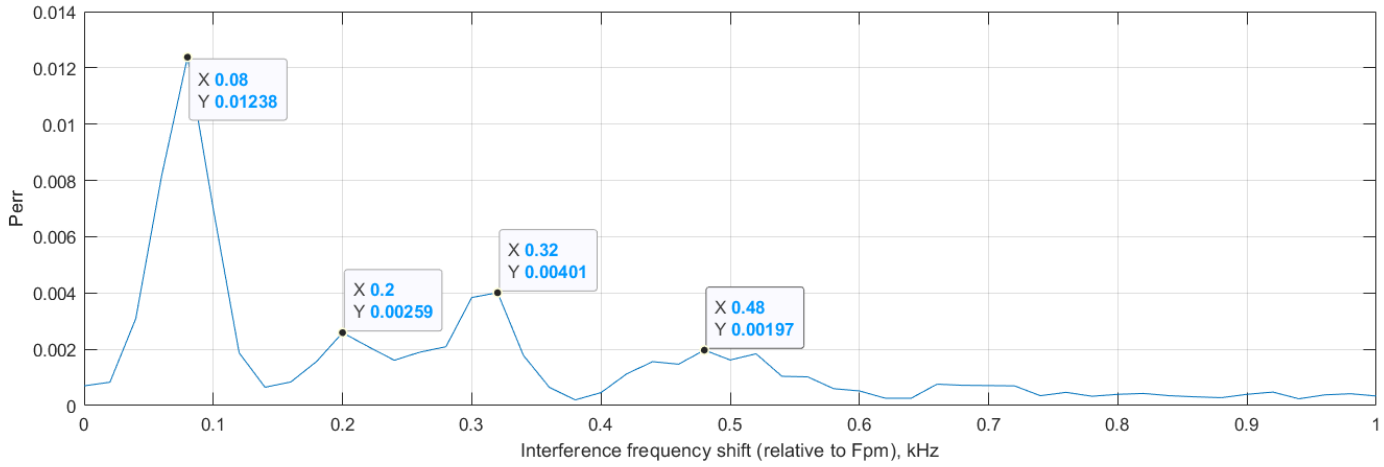


Результаты моделирования

Зависимость вероятности ошибки от параметров помехи

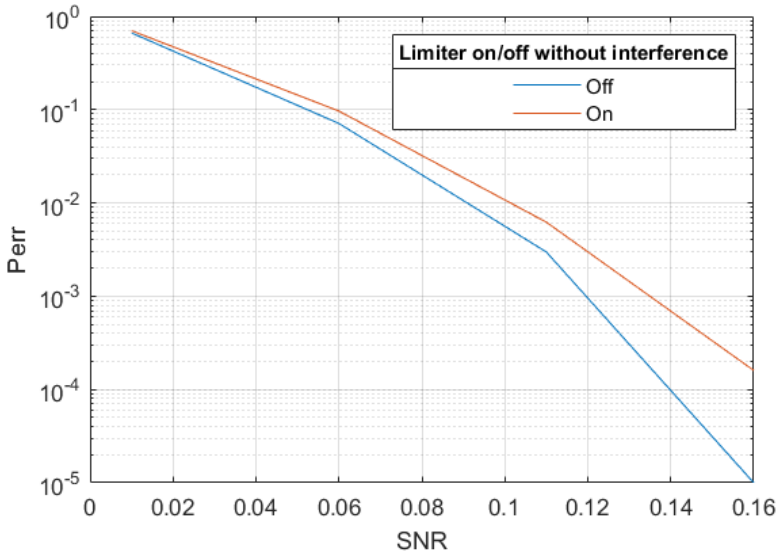


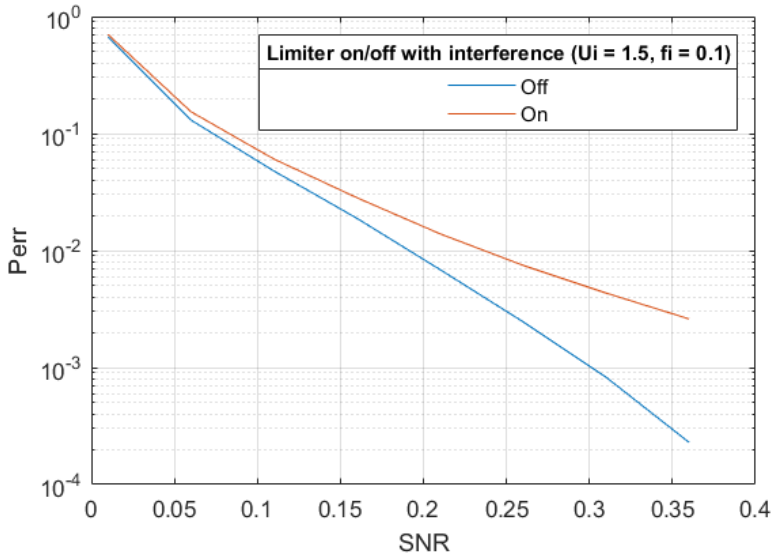




Результаты моделирования

Зависимость вероятности ошибки от амплитудного ограничителя





Выводы

1. Показано, что хотя ФМ ШПС и являются достаточно помехоустойчивыми, однако существует узкополосная помеха, которая может оказывать значительное влияние на прием ШПС

2. Разработана модель приема ШПС, которая позволяет построить зависимость вероятности ошибки от параметров сигнала и помехи.

3. Проведено моделирование приема и обработки ШПС в условиях шумовой и узкополосной помех.

4. В результате моделирования установлено, что узкополосная помеха может оказать значительное влияние на вероятность ошибки принятого сигнала.

5. Установлено, что воздействие помехи зависит от ее частоты и амплитуды. Влияние частоты зависит от вида модулирующей последовательности.

6. Из полученных результатов следует, что амплитудный ограничитель, в целом, негативно влияет на помехоустойчивость.

7. Также установлено, что расширение полосы пропускания ФНЧ выше некоторой не приводит к росту или уменьшению вероятности ошибки

Номер документа	*Позначення*	*Найменування*	*Прим.*
		Текстові документи	
1		*Пояснювальна записка*	*76 с.*
		Графічні документи	*Ел.ф*
1		*Титульний слайд*	*Ф.А4х1*
2		*ФМ ШПС та його властивості*	*Ф.А4х1*
3		*Вплив вузько спрямованої завади на пройм ШПС*	*Ф.А4х1*
4		*Завдання*	*Ф.А4х1*
5		*Чому необхідно застосувати моделювання*	*Ф.А4х1*
6		*Модель сигналу*	*Ф.А4х1*
7		*Модель приймача*	*Ф.А4х1*
8		*Результати моделювання*	*Ф.А4х4*
9		*Висновки*	*Ф.А4х1*

					ГЮІК.ХХХ.ХХХ.ХХХ Д1		
Зм	*Арк*	*№ докум.*	*Підп.*	*Дата*			
Розроб.		*Чередніченко*			Аналіз впливу завад на прийом шумоподібного сигналу шляхом математичного моделювання. *Відомість кваліфікаційної роботи магістра*	*Літ*	*Арк*	*Ар-*
Перевір.		*Антіпов*						1
Т.контр						*ХНУРЕ Кафедра КРіСТЗІ*	
Н.контр		*Милютченко*					
Затв.		*Антіпов*					