

ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ
БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ
им. проф. М. А. БОНЧ-БРУЕВИЧА»
(спбгут)

В. Ю. Волков

**МАТЕМАТИЧЕСКИЕ МЕТОДЫ
В ТЕОРИИ РАДИОТЕХНИЧЕСКИХ СИСТЕМ**

ПРОГРАММЫ МОДЕЛИРОВАНИЯ В МАТЛАВ

УЧЕБНОЕ ПОСОБИЕ

СПб ГУТ)))

**САНКТ-ПЕТЕРБУРГ
2018**

УДК 621.369(075.8)
ББК 32.84я73
В 67

Рецензенты:
кандидат технических наук,
доцент кафедры радиосистем Санкт-Петербургского
государственного электротехнического университета «ЛЭТИ»
А. С. Маругин,
доктор технических наук, профессор кафедры радиосвязи и вещания СПбГУТ
Ю. А. Ковалгин

*Утверждено редакционно-издательским советом СПбГУТ
в качестве учебного пособия*

Волков, В. Ю.

В 67 Математические методы в теории радиотехнических систем.
Программы моделирования в Matlab : учебное пособие / В. Ю. Вол-
ков ; СПбГУТ. – СПб., 2018. – 19 с.
ISBN 978-5-89160-166-6

Написано в соответствии с рабочими программами дисциплин:
«Математические методы в теории радиотехнических систем», «Со-
временные методы обработки сигналов и полей».

Приводятся тексты программ формирования в среде Matlab сиг-
налов, включающих прямоугольные видео- и радиоимпульсы, им-
пульс с ЛЧМ и их смеси с гауссовским шумом. Моделируются алго-
ритмы корреляционной и фильтровой обработки в целях
обнаружения этих сигналов на фоне шума.

Предназначено для студентов, обучающихся по направлениям
(специальностям) 11.03.01 и 11.04.01 «Радиотехника», аспирантов и
специалистов в данной области.

**УДК 621.369(075.8)
ББК 32.84я73**

ISBN 978-5-89160-166-6 © Волков В. Ю., 2018

© Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Санкт-Петербургский государственный университет
телекоммуникаций им. проф. М. А. Бонч-Бруевича», 2018

СОДЕРЖАНИЕ

1. Моделирование прямоугольного видеоимпульса и прямоугольного радиоимпульса на фоне шума.....	4
2. Корреляционная обработка и согласованная фильтрация прямоугольного видеоимпульса.....	12
3. Корреляционная обработка и согласованная фильтрация прямоугольного радиоимпульса.....	21
4. Формирование и обработка одиночного ЛЧМ прямоугольного радиоимпульса на фоне шума.....	33
Задания.....	48

Программы моделирования в Matlab иллюстрируют теоретические положения, изложенные в учебном пособии В. Ю. Волкова «Математические методы в теории радиотехнических систем. Обнаружение и различение сигналов» (СПбГУТ. – СПб., 2018). Они могут служить исходным материалом для новых программных продуктов. В целях удобства просмотра результатов расчета и формируемых изображений программы включают функциональные остановки в конце каждого блока, для продолжения вычислений следует нажимать клавишу «Enter».

1. МОДЕЛИРОВАНИЕ ПРЯМОУГОЛЬНОГО ВИДЕОИМПУЛЬСА И ПРЯМОУГОЛЬНОГО РАДИОИМПУЛЬСА НА ФОНЕ ШУМА

Прямоугольный видеоимпульс

Программа формирует одиночный прямоугольный видеоимпульс на фоне аддитивного гауссовского шума с вычислением амплитудного спектра и спектрограммы.

Она позволяет исследовать влияние отношения сигнал/шум на характеристики процесса. Программа использует функцию `func_rect_impulse`, которая формирует на L точках прямоугольный видеоимпульс с амплитудой A , содержащий LS выборочных значений, задержанный на LD точек. Эта функция также выдает временной вектор $t = (0:L-1)Ts$ по заданной частоте выборки $Fs = 1/Ts$ в герцах и формирует полный временной интервал $T = L \cdot Ts$ в секундах.

Функцию `func_rect_impulse` следует поместить в ту же папку, где находится основная программа `main_rect_impulse`.

Сформированный прямоугольный видеоимпульс суммируется с дискретным гауссовским шумом, который имеет единичную дисперсию. Для обеспечения повторяемости реализации шума при повторных запусках программы производится инициализация датчика случайных чисел командой `rand('state', 0);`

В программе вычисляется односторонний спектр полученного наблюдения, а также спектрограмма (кратковременный текущий спектр).

```
% main_rect_impulse
% rectangular impulse plus Gaussian noise
% uses function = func_rect_impulse
clc
clear all
close all
START=0
```

```

L = 1000      % Total length [points]
LS = 200     % Length of signal [points]
LD = 300     % Signal appearance [points]
%%%%%%%%%%
Fs=1000      % Sampling frequency [Hz]
Ts = 1/Fs    % Sample time [s]
TS=LS*Ts    % Signal duration [s]
TD=LD*Ts    % Signal delay [s]
ts=input('STOP_0','s'); % Press ENTER to continue
%%%%%%%%%%

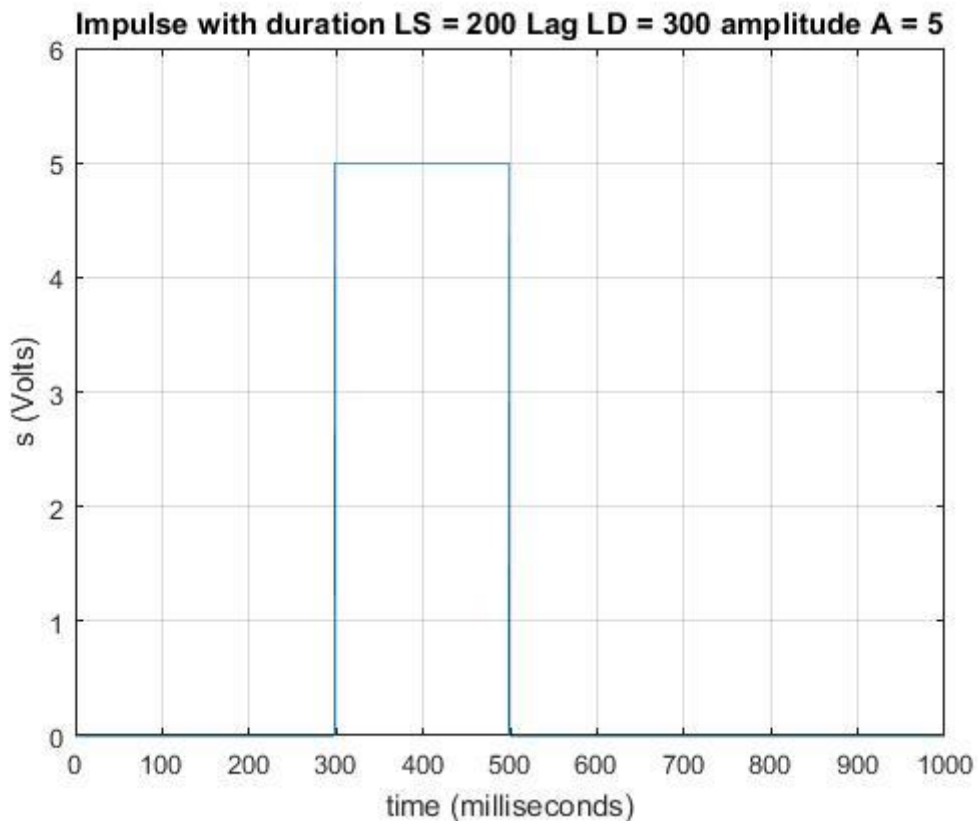
```

На данном шаге программы задается частота дискретизации F_s и определяется временной интервал дискретизации T_s . При фиксированном числе выборок изменения T_s приводят к изменениям длительности сигнала в секундах и ширины спектра в герцах.

```

PROCESS=1 % rect impulse
A=5 % Amplitude [V]
[s,T,t]=func_rect_impulse(Fs,LS,LD,L,A);
figure(100), plot(Fs*t(1:L),s(1:L),t,A+1),
title(['Impulse with duration LS = ',num2str(LS),...
' Lag LD = ',num2str(LD),' amplitude A = ',num2str(A)]),
xlabel('time (milliseconds)')
ylabel('s (Volts)'), grid
ts=input('STOP_1','s'); % Press ENTER to continue

```

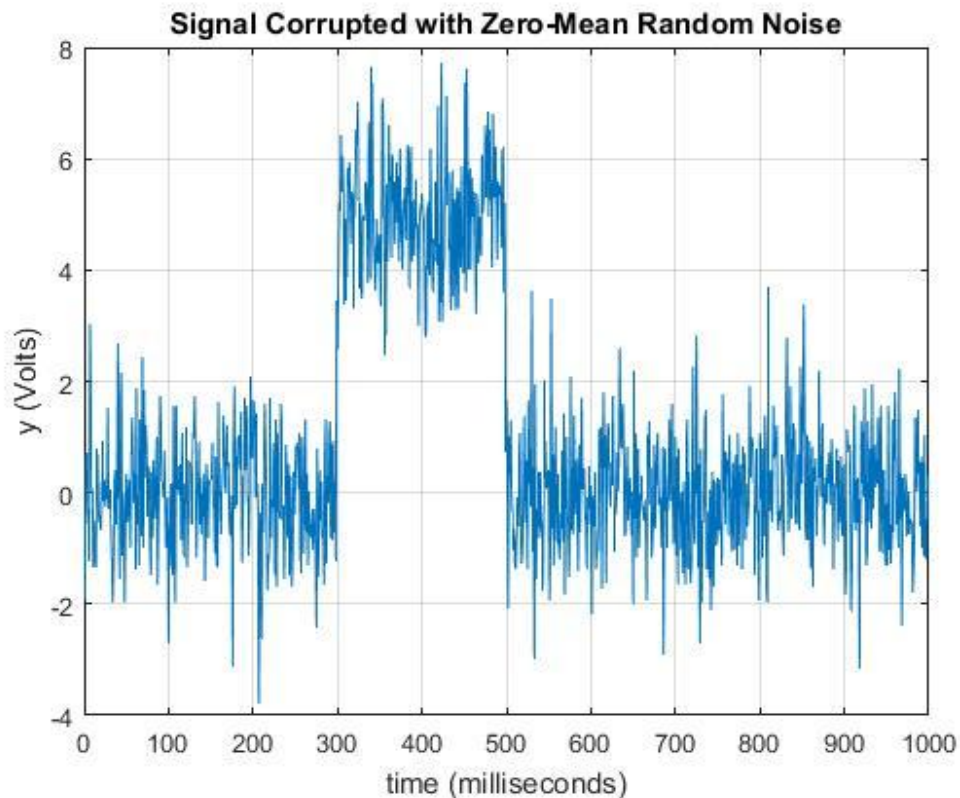


Входной прямоугольный видеоимпульс

```

%%%%%%%%%%
PROCESS=2 %Signal plus noise
rand('state',0);
y = s + randn(size(t));      % Observation
SNR=A^2% Power SNR
figure(200), plot(Fs*t(1:L),y(1:L))
title('Signal Corrupted with Zero-Mean Random Noise')
xlabel('time (milliseconds)')
ylabel('y (Volts)')
grid on
ts=input('STOP_2','s'); % Press ENTER to continue

```



Наблюдение, содержащее прямоугольный видеоимпульс

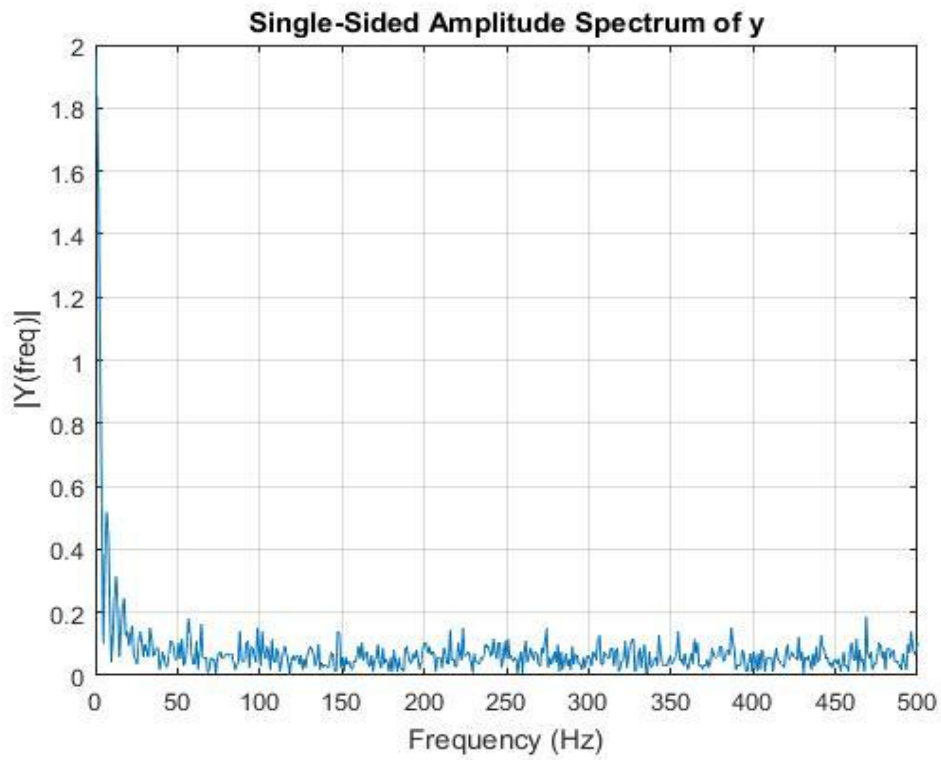
На этом шаге к сигналу добавляется дискретный гауссовский шум с единичной дисперсией $\sigma^2 = 1$. Отношение сигнал/шум по напряжению в каждом отсчете равно A/σ , а по мощности – $SNR = A^2/\sigma^2$.

```

%%%%%%%%%%
PROCESS=3 % Fast Fourier Transform and Spectrogram
nextpow2(L)
NFFT = 2^nextpow2(L) % Next power of 2 from length of y
Y = fft(y,NFFT)/L;
freq = Fs/2*linspace(0,1,NFFT/2+1);
% Plot single-sided amplitude spectrum.
figure(300), plot(freq,2*abs(Y(1:NFFT/2+1)))
title('Single-Sided Amplitude Spectrum of y')
xlabel('Frequency (Hz)')
ylabel('|Y(freq)|')

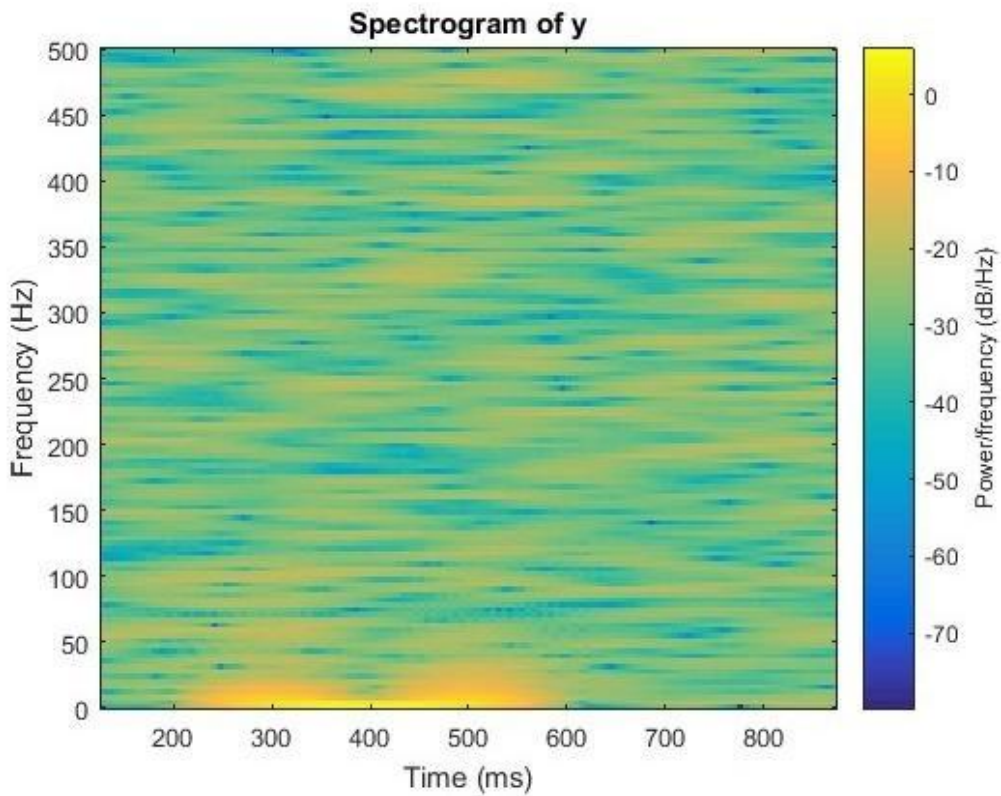
```

```
grid on
ts=input('STOP', 's');
```



Односторонний амплитудный спектр видеоимпульса в смеси с шумом

```
figure(301), spectrogram(y,256,250,256,1e3, 'yaxis'),
title('Spectrogram of y')
```



Спектрограмма

Спектрограмма показывает распределение энергии за кратковременный интервал по времени и по частоте.

END=0

%%%

```
function [s,T,t]=func_rect_impulse(Fs,LS,LD,L,A)
% Input
% Fs Sampling frequency [Hz]
% LS Length of signal [points]
% LD time of appearance [points]
% L Length of sample [points]
% A signal amplitude [V]
%Output
% T Time of realization [s]
% t time vector (through Ts)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Ts = 1/Fs; % Sample time [s]
T=L*Ts; % Time of realization [s]
t = (0:L-1)*Ts; % Time vector [s]
s=zeros(size(t));
for i=LD:LD+LS-1
    s(i)=s(i)+A;
end;
end
```

Прямоугольный радиоимпульс

Программа формирует одиночный прямоугольный радиоимпульс на фоне аддитивного гауссовского шума с вычислением амплитудного спектра и спектрограммы. Она использует функцию `func_sin_impulse`, которая формирует на L точках прямоугольный радиоимпульс с амплитудой A , частотой f , начальной фазой в градусах $teta$, длительностью LS , задержанный на LD . Эта функция также выдает временной вектор $t=(0:L-1)Ts$ по заданной частоте выборки $Fs=1/Ts$ в герцах, и формирует полный временной интервал $T=L \cdot Ts$ в секундах.

В программе вычисляется односторонний спектр суммы полезного сигнала и дискретного гауссовского шума с единичной дисперсией, а также спектрограмма (кратковременный спектр).

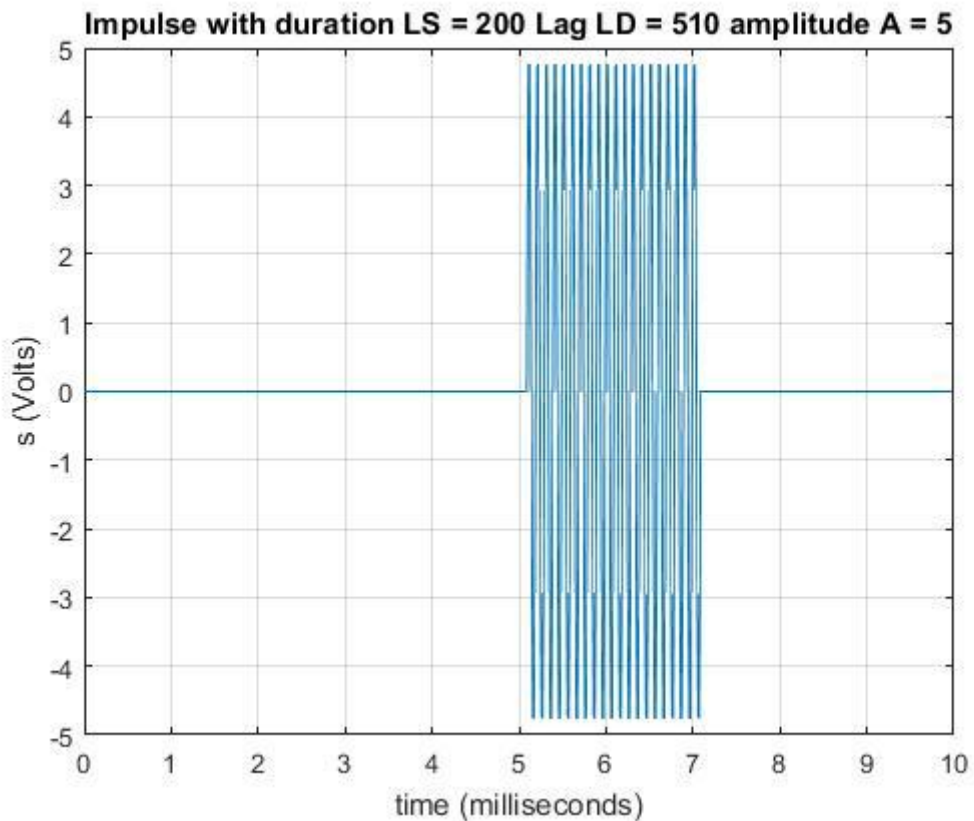
```
% main_sin_impulse
% sin impulse plus Gaussian noise
% uses function = func_sin_impulse
clc
clear all
close all
START=0
```



```

L=1000 % Total length (points)
LS = 200 % Length of signal (points)
LD = 510 % Signal appearance [points]
%%%%%%%%%%
Ts = 0.01 % Sample time [s]
Fs=1/Ts % Sample frequency [Hz]
%%%%%%%%%%
T=0.1 % Period of signal [s]
f=1/T % Frequency of signal [Hz]
%%%%%
teta=0 % teta in degrees
%%%%%%%%%%
PROCESS=1
A=5
[T,t,s]=func_sin_impulse(Fs,LS,LD,L,f,A,teta);
% s signal with lag LD
figure(100), plot(t,s),
title(['Impulse with duration LS = ',num2str(LS),...
' Lag LD = ',num2str(LD),' amplitude A = ',num2str(A)]),
xlabel('time (milliseconds)')
ylabel('s (Volts)'), grid
ts=input('STOP_1','s');

```

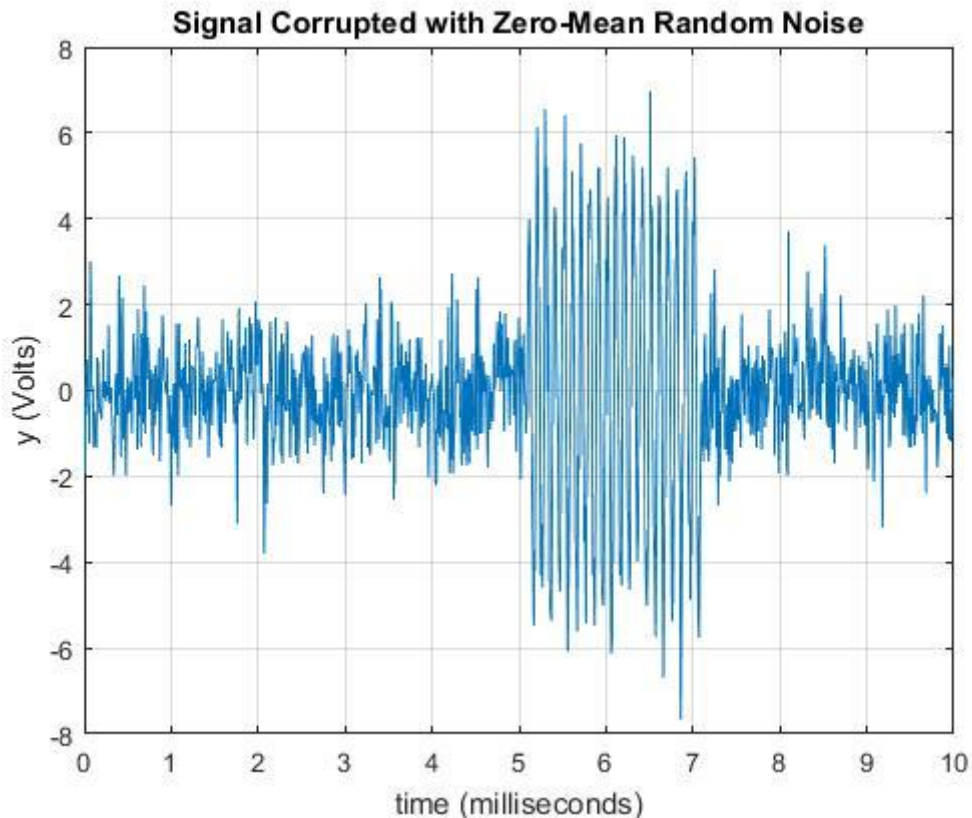


Входной радиоимпульс

```

%%%%%%%%%%
PROCESS=2
rand('state',0);
y = s + randn(size(t));      % Sin impulse plus noise
SNR=A^2 % Power SNR
figure(200), plot(t,y), grid,
title('Signal Corrupted with Zero-Mean Random Noise')
xlabel('time (milliseconds)')
ylabel('y (Volts)')
ts=input('STOP_2','s');

```

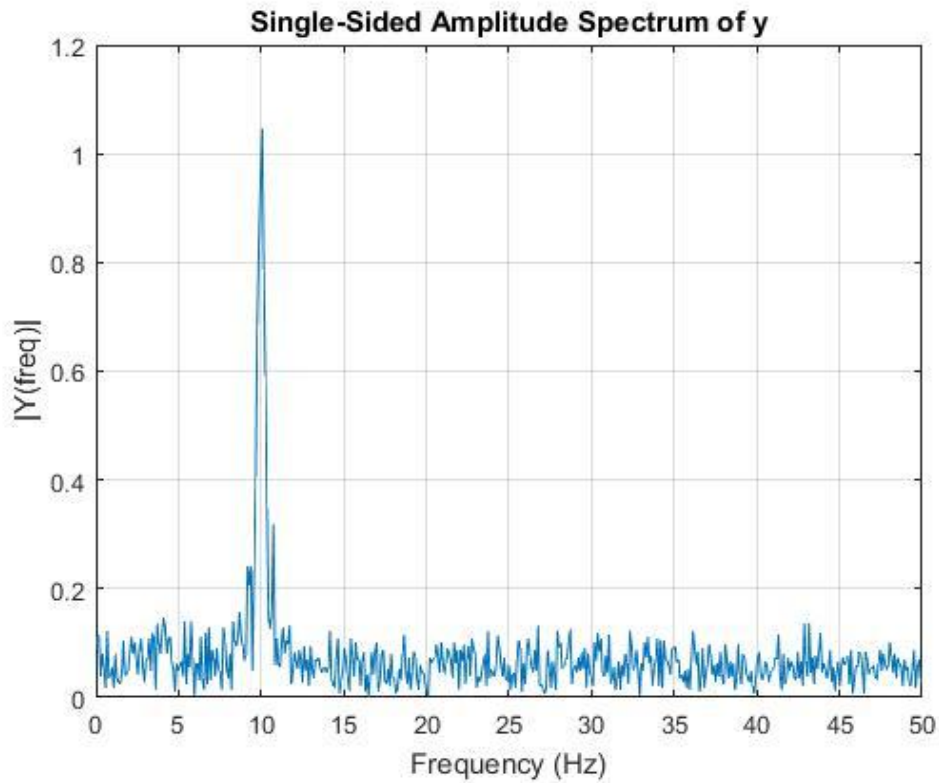


Наблюдение, содержащее входной радиоимпульс

```

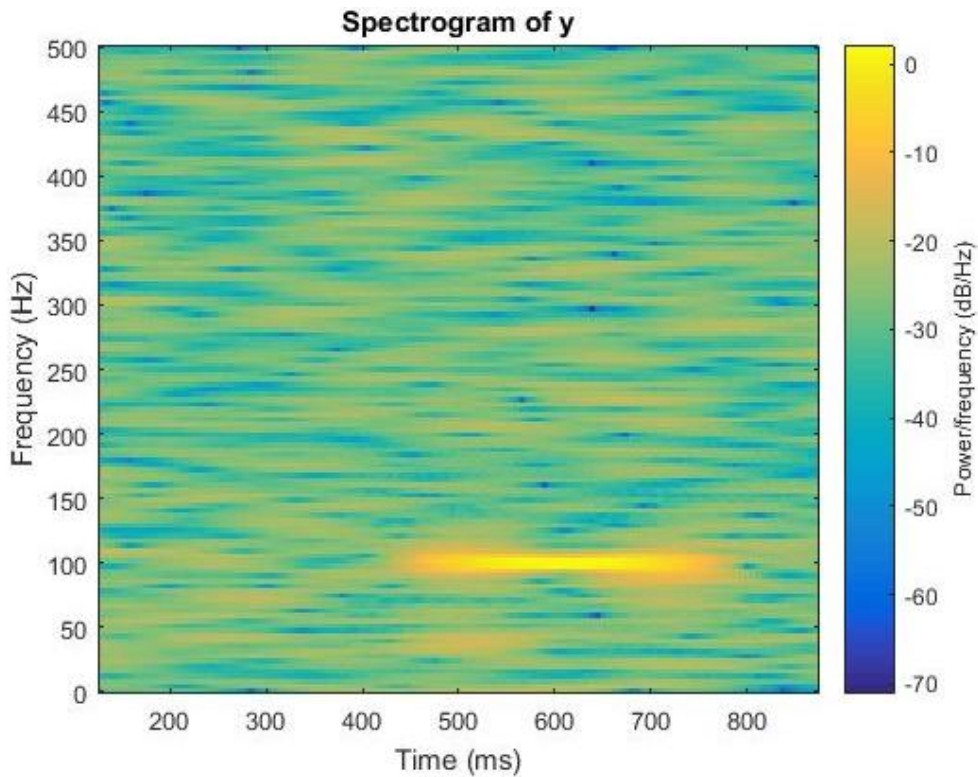
%%%%%%%%%%
PROCESS=3 % Fast Fourier Transform and Spectrogram
nextpow2(L)
NFFT = 2^nextpow2(L) % Next power of 2 from length of y
Y = fft(y,NFFT)/L;
freq = Fs/2*linspace(0,1,NFFT/2+1);
% Plot single-sided amplitude spectrum.
figure(300), plot(freq,2*abs(Y(1:NFFT/2+1)))
title('Single-Sided Amplitude Spectrum of y')
xlabel('Frequency (Hz)')
ylabel('|Y(freq)|')
grid on
ts=input('STOP','s');

```



Односторонний амплитудный спектр радиоимпульса в смеси с шумом

`figure(301), spectrogram(y,256,250,256,1e3,'yaxis'),`



Спектрограмма радиоимпульса в смеси с шумом

Спектрограмма показывает распределение энергии за кратковременный интервал по времени и по частоте.

```
END=0
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function [T,t,s]=func_sin_impulse(Fs,LS,LD,L,f,A,teta)
% Fs Sampling frequency [Hz]
% LS Length of signal [points]
% LD time of appearance (lag)[points]
% L Length of sample [points]
% f signal frequency [Hz]
% A signal amplitude [V]
% teta signal phase in grad
Ts = 1/Fs; % Sample time [s]
T=L*Ts; % Signal duration [s]
teta_rd=teta*pi/180; % teta in radians
t1 = (LD:LD+LS-1)*Ts; % Time signal vector [s]
S = A*sin(2*pi*f*(t1-LD*Ts)+teta_rd);
t = (0:L-1)*Ts; % Time vector
s=zeros(size(t));
for i=LD:LD+LS-1
    s(i)=s(i)+S(i-LD+1);
end;
end
```

Программы позволяют исследовать влияние отношения сигнал/шум на спектральные характеристики процессов. Для этого следует изменять амплитуду сигнала.

2. КОРРЕЛЯЦИОННАЯ ОБРАБОТКА И СОГЛАСОВАННАЯ ФИЛЬТРАЦИЯ ПРЯМОУГОЛЬНОГО ВИДЕОИМПУЛЬСА

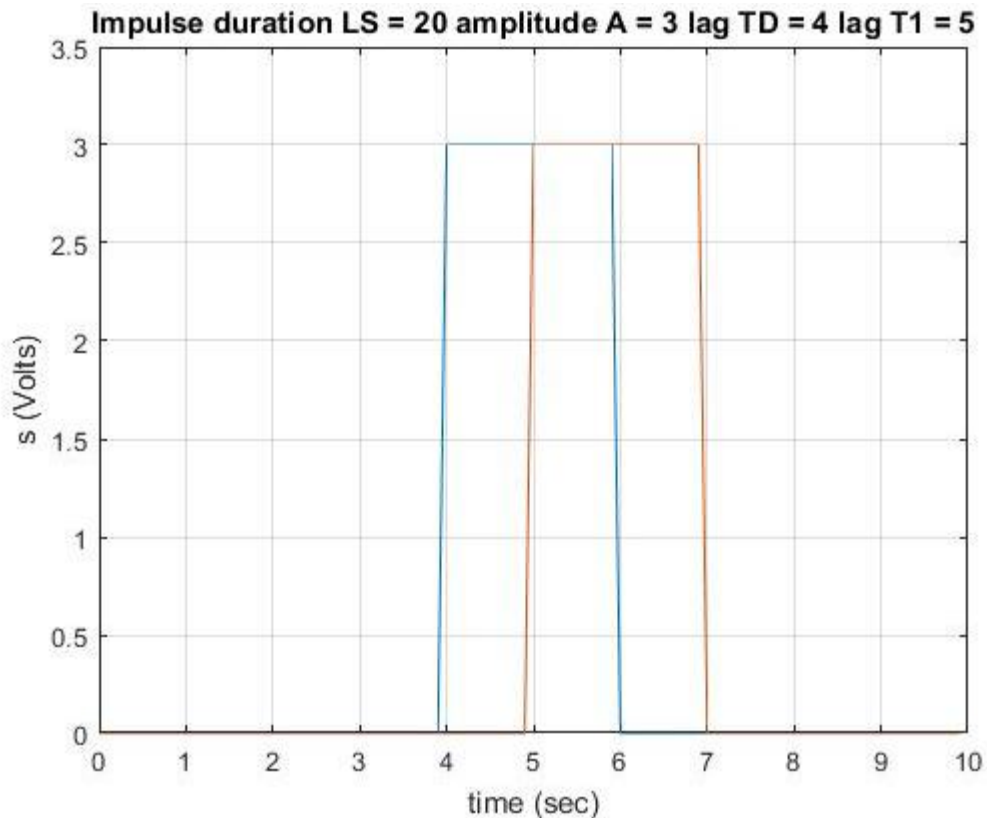
В программе производится корреляционная обработка и согласованная фильтрация наблюдений, полученных в первой программе, для задачи обнаружения прямоугольного видеоимпульса в шуме. Предусмотрена возможность установки ожидаемой задержки сигнала $L1$, отличной от истинной LD . Программа использует функцию `func_rect_impulse`, введенную ранее, которая помещается в ту же папку. При одинаковых значениях $L1$ и LD достигается максимальный эффект корреляционной обработки, который совпадает с максимальным выходным сигналом согласованного фильтра. Программа позволяет исследовать влияние различия значений истинной и ожидаемой задержки сигнала на выходной эффект при различных отношениях сигнал/шум.

```
% main_det_rect_impulse
% processing rectangular impulse plus noise
% uses function = func_rect_imp
clc
```

```

clear all
close all
START=0
L=100 % Total length (points)
LS = 20 % Length (duration) of signal (points)
LD = 41 % Signal appearance [points]
L1 = 51 % Expected signal appearance [points]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Ts = 0.1 % Sample time [s]
Fs=1/Ts % Sample frequency [Hz]
TS = LS*Ts % Signal Duration [s]
TD = (LD-1)*Ts % Signal Delay [s]
T1 = (L1-1)*Ts % Signal Expected Appearance [s]
ts=input('STOP_0','s');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
PROCESS=1 % Signal
A=3
s=func_rect_impulse(Fs,LS,LD,L,A);
s1=func_rect_impulse(Fs,LS,L1,L,A);
t=(0:L-1)*Ts;
figure(100), plot(t,s,t,s1,t,A*1.1), grid
title(['Impulse duration LS = ',num2str(LS),...
' amplitude A = ',num2str(A),' lag TD = ',num2str(TD),...
' lag T1 = ',num2str(T1)]),
xlabel('time (sec)'), ylabel('s (Volts)')
ts=input('STOP','s');

```

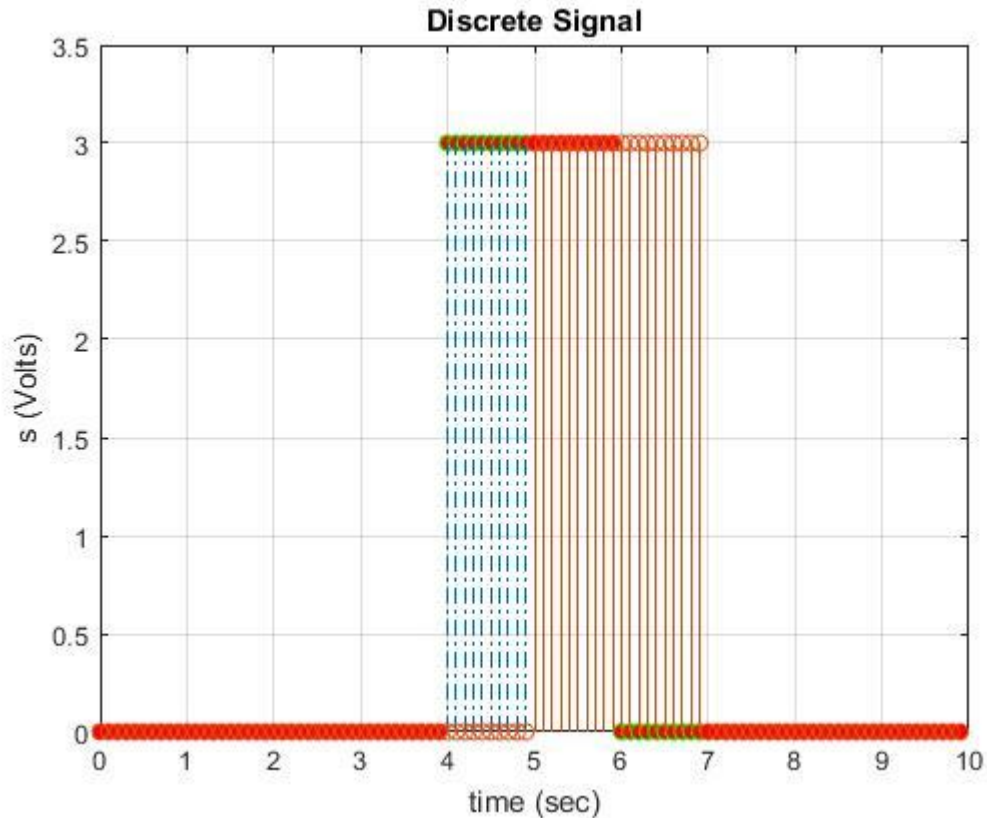


Входной и опорный видеоимпульсы

```

figure(101), stem(t,s,'LineStyle','-
.', 'MarkerFaceColor','red',...
'MarkerEdgeColor','green'), title('Discrete Signal');
hold on; stem(t,s1); plot(t,A*1.1);
xlabel('time (sec)'); ylabel('s (Volts)'), grid on
ts=input('STOP_1','s');

```

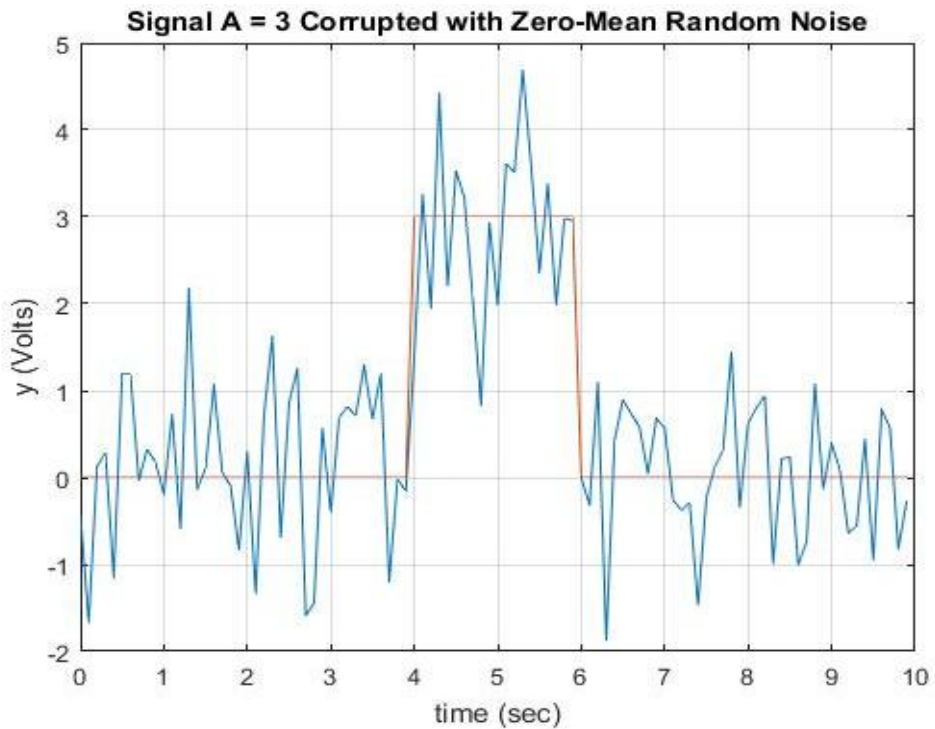


Входной и опорный видеоимпульсы в виде дискретных отсчетов

```

%%%%%%%%%%
PROCESS=2 % Observations
rand('state',0);
%
y = s + randn(size(t)); % Signal plus noise
% y = s;
figure(200), plot(t,y,t,s(1:L))
title(['Signal A = ',num2str(A),...
' Corrupted with Zero-Mean Random Noise'])
xlabel('time (sec)'), ylabel('y (Volts)'), grid
hold on; plot(t,A*1.1);
ts=input('STOP','s');

```

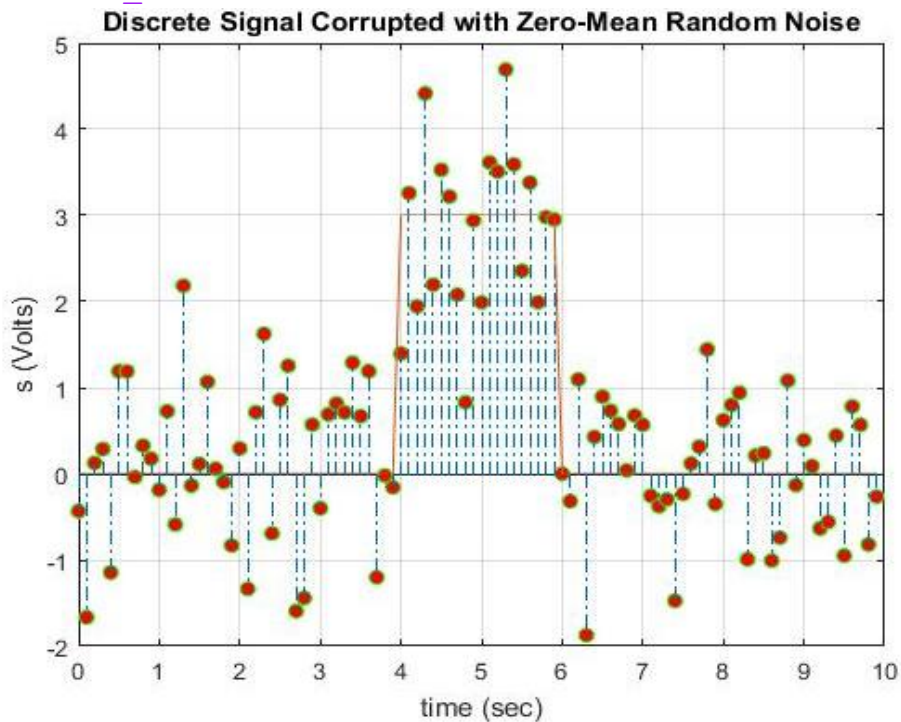



Наблюдение, включающее прямоугольный видеоимпульс в смеси с шумом

```

%%%%%%%%%%
figure(201), stem(t,y,'LineStyle','-.','...
    'MarkerFaceColor','red',...
    'MarkerEdgeColor','green'),
title('Discrete Signal Corrupted with Zero-Mean Random Noise');
hold on; plot(t,s,t,A*1.1),
xlabel('time (sec)')
ylabel('s (Volts)'), grid on
ts=input('STOP_2','s');

```

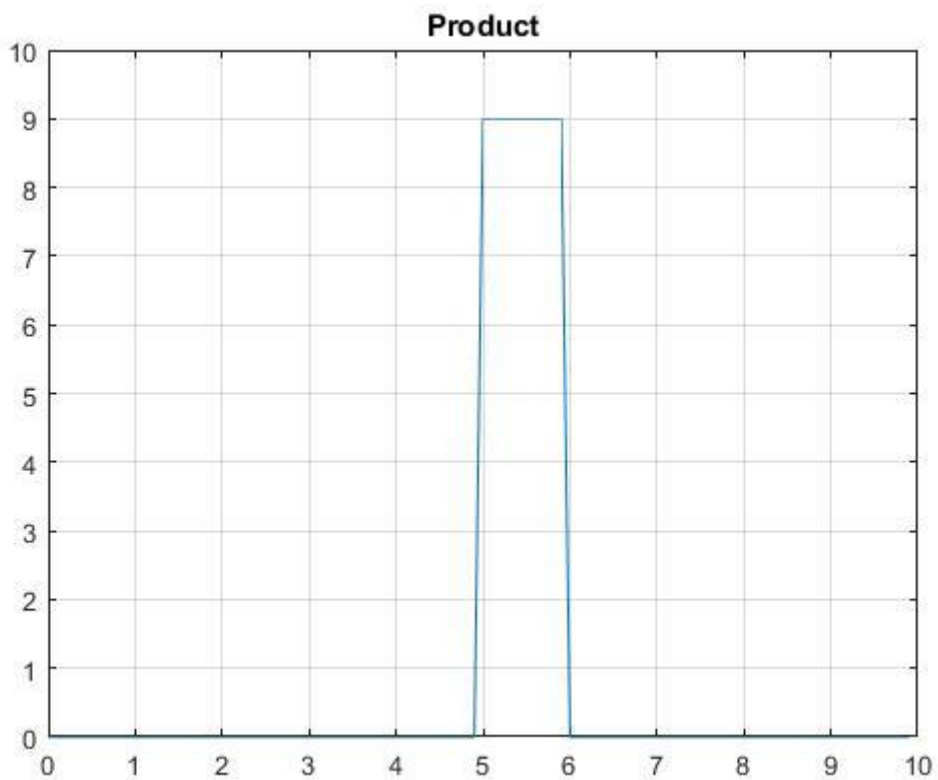


Наблюдение в виде дискретных отсчетов

```

%%%%%%%%%%
PROCESS=3 % CORRELATOR
LB=max(LD,L1)
LE=min(LD+LS,L1+LS)
TB=(LB-1)*Ts
TE=(LE-1)*Ts
for i=1:L
sm(i)=s(i).*s1(i);
end;
maxsm=max(sm)
figure(300), plot(t,sm,t,maxsm*1.1), title('Product'), grid
cly(1)=y(1)*s1(1);
for i=2:L
cly(i)=cly(i-1)+y(i)*s1(i);
end;

```



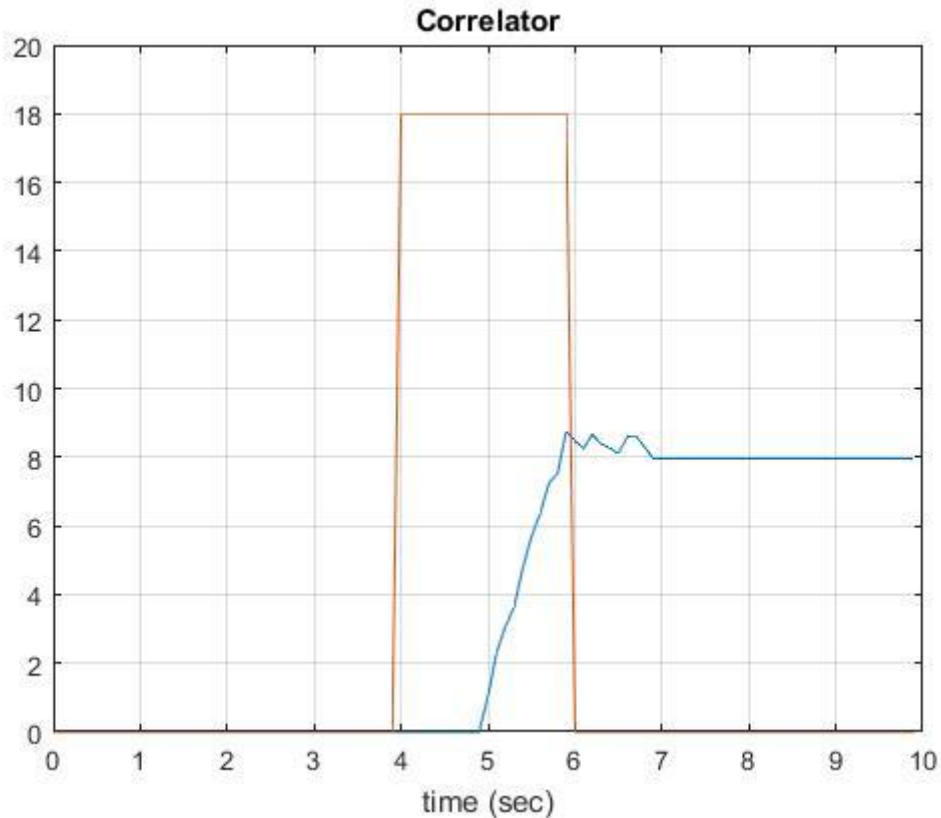
Произведение входного и опорного видеоимпульсов

Произведение входного и опорного видеоимпульсов определяет интервал накопления сигнальной составляющей на выходе коррелятора.

```

figure(301), plot(t,cly*Ts,t,s.^2*TS,t,max(s.^2*TS*1.1)),
title('Correlator'), grid;
xlabel('time (sec)')
ts=input('STOP_3','s');

```

Процесс на выходе коррелятора

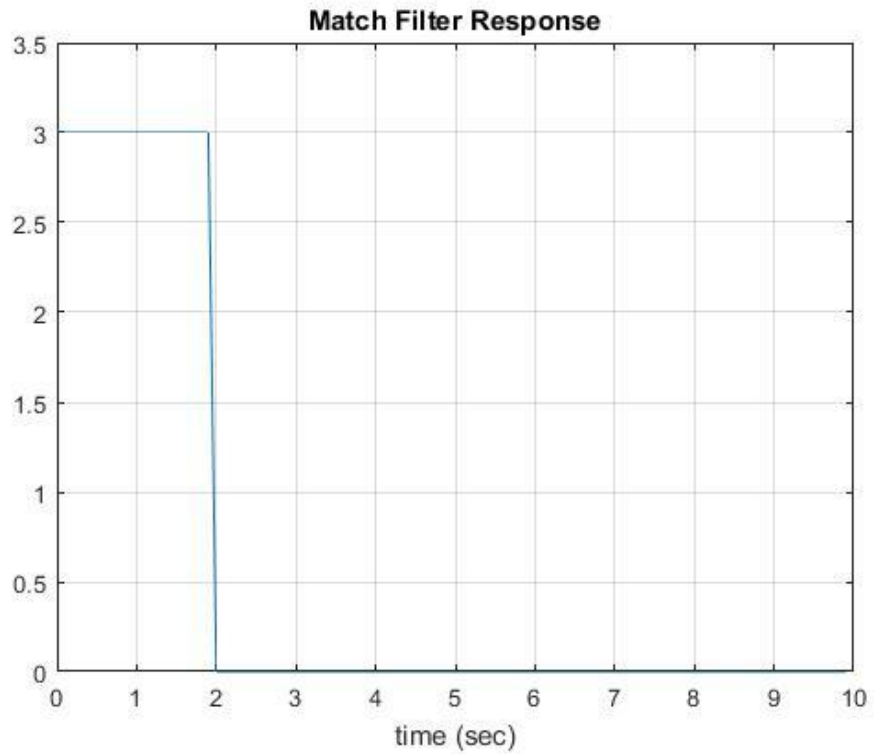
Шумовая составляющая на выходе коррелятора формируется в течение длительности опорного импульса и не зависит от момента прихода входного импульса. Сигнальную составляющую можно увидеть, если убрать шум на входе: $y = s$ на шаге PROCESS=2.

Если $L1=LS$, то сигнальная составляющая достигает максимального значения, равного энергии сигнала $E = 18 \text{ В}^2 \cdot \text{с}$. Поскольку в данном случае моменты прихода импульсов отличаются на половину длительности импульса, то сигнальная составляющая достигает только значения $9 \text{ В}^2 \cdot \text{с}$.

```

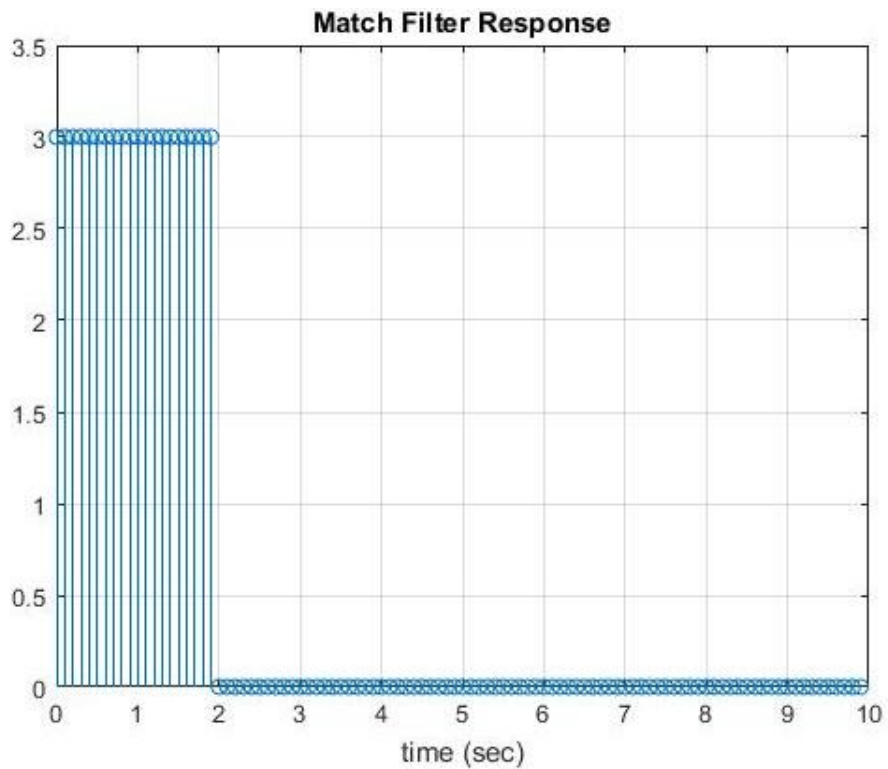
%%%%%%%%%%%% CONVOLUTION %%%%%%%%%%%%%
PROCESS=4 % Match Filter Response
s0=func_rect_impulse(Fs,LS,1,L,A);
% s0
h=zeros(1,L);
for i=1:LS
    h(i)=s0(LS-i+1);
end
maxh=max(h)
figure(400), plot(t,h,t,maxh*1.1), title('Match Filter Response'),
xlabel('time (sec)'), grid

```



Импульсная характеристика согласованного фильтра

```
figure(401), stem(t,h), title('Match Filter Response'),
xlabel('time (sec)'), grid
hold on
plot(t,maxh*1.1)
ts=input('STOP_4','s');
```

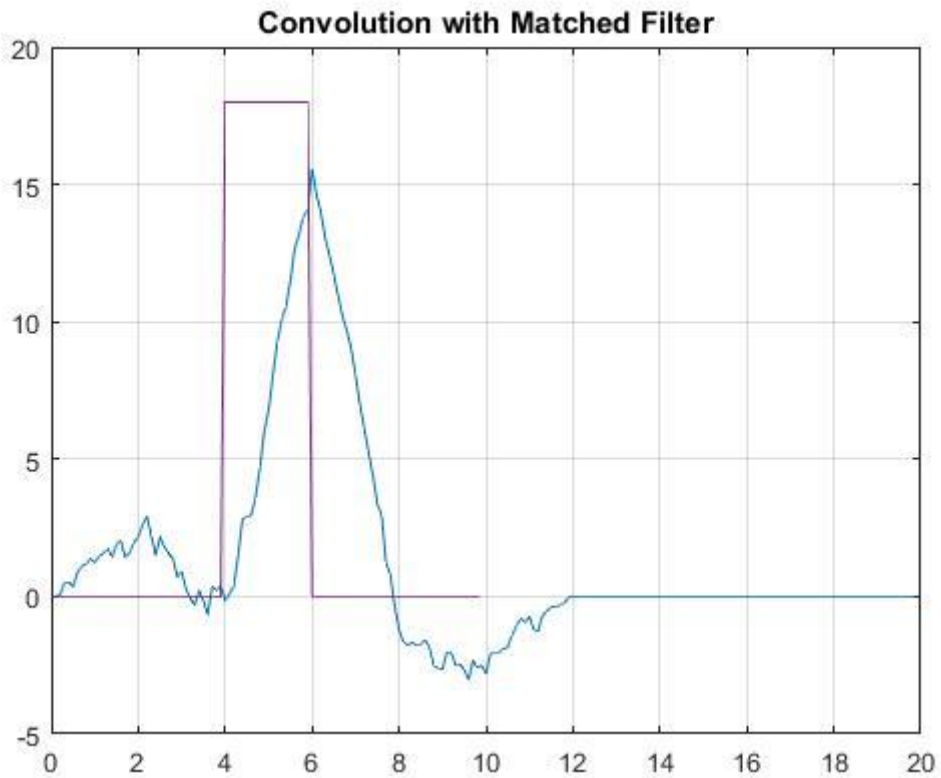


Импульсная характеристика согласованного фильтра в дискретном виде

```

%%%%%
PROCESS=5
cy=conv(y,h);
cy=cy*Ts;
mcy=max(cy); mcy=mcy*Ts
k1=-(L-1):(L-1);
k2=k1+LS+80;
figure(500), plot(k2*Ts,cy),
title('Convolution with Matched Filter'), grid
hold on
plot(t,mcy*1.1,t,s.^2*TS);
ts=input('STOP','s');

```



Процесс на выходе согласованного фильтра

```

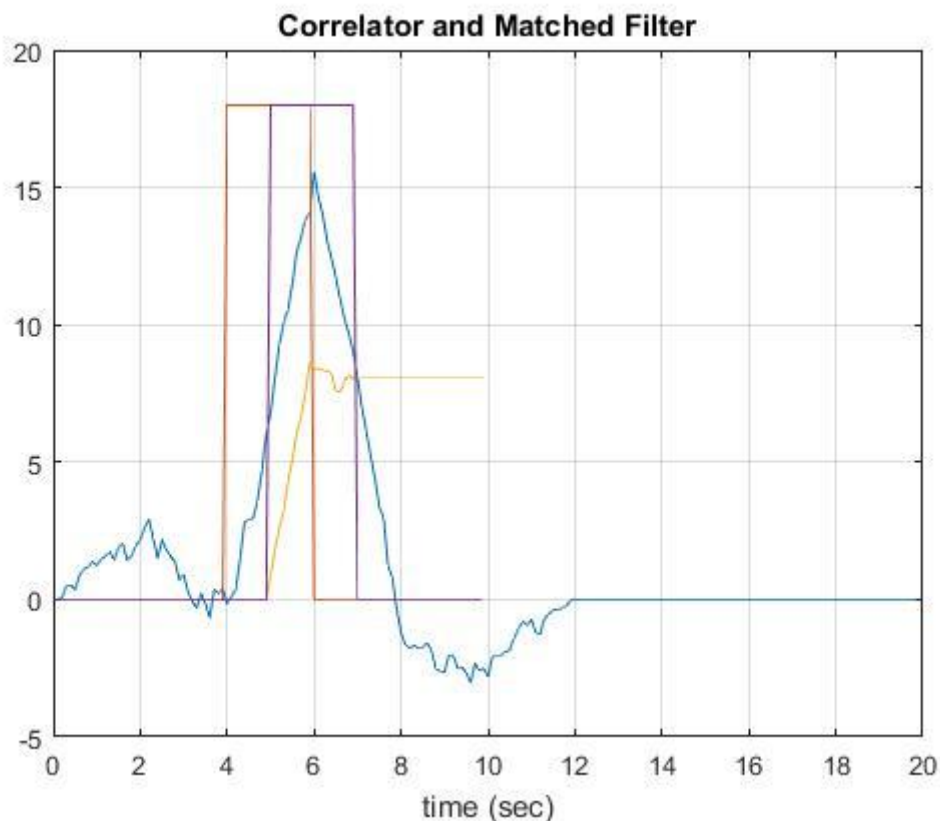
figure(501), stem(k2*Ts,cy),
title('Convolution with Matched Filter'), grid
xlabel('time (sec)')
hold on
plot(t,mcy*1.1,t,s.^2*TS);
ts=input('STOP_5','s');

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
PROCESS=6
figure(600), plot(k2*Ts,cy), grid
title('Correlator and Matched Filter')
xlabel('time (sec)')
hold on
plot(t,s.^2*TS,t,cly*Ts,t,s1.^2*TS,t,mcy*1.1)
ts=input('STOP_6','s');

```



Процессы на выходе коррелятора и согласованного фильтра

END=0

Согласованный фильтр формирует корреляционный интеграл в момент окончания импульса на входе. Если моменты прихода и окончания входного импульса известны, то оптимальный обнаружитель реализуется путем селекции корреляционного интеграла и сравнения его с порогом. При неизвестных моментах прихода и окончания импульса возникает проблема селекции корреляционного интеграла, часто берется максимальное значение выходного сигнала, но оно дает близкое к нему значение только при больших отношениях сигнал/шум.

В данном случае имеется рассогласование между моментами прихода входного импульса и началом формирования опорного импульса коррелятора. Поэтому коррелятор не формирует корреляционный интеграл, и реализация оптимального обнаружителя невозможна. Для формирования корреляционного интеграла в корреляторе необходимо совпадение моментов прихода входного и опорного импульсов.

Как коррелятор, так и согласованный фильтр имеют одинаковый выходной сигнал в момент окончания опорного импульса, что видно на изображении.

3. КОРРЕЛЯЦИОННАЯ ОБРАБОТКА И СОГЛАСОВАННАЯ ФИЛЬТРАЦИЯ ПРЯМОУГОЛЬНОГО РАДИОИМПУЛЬСА

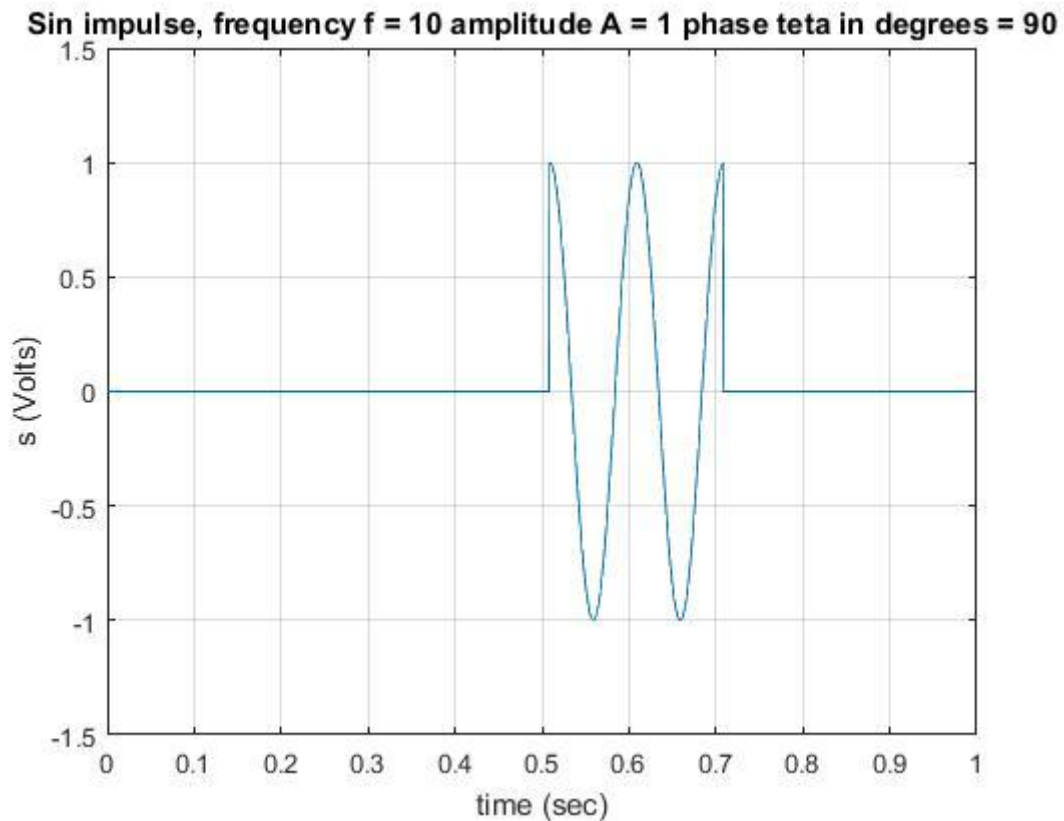
В программе производится корреляционная обработка и согласованная фильтрация наблюдений, полученных в первой программе, для задачи обнаружения прямоугольного радиоимпульса в шуме. Предусмотрена возможность установки ожидаемой дискретной задержки сигнала $L1$, отличной от истинной LD . Программа использует функцию `func_sin_impulse`, введенную ранее, которая помещается в ту же папку. При одинаковых значениях $L1$ и LD достигается максимальный эффект корреляционной обработки, который совпадает с максимальным выходным сигналом согласованного фильтра. Программа позволяет исследовать влияние различия значений истинной и ожидаемой задержек сигнала на выходной эффект при различных отношениях сигнал/шум.

```
%main_det_sin_impulse
%sin impulse corrupted by noise
%uses function = func_sin_impulse
clc
clear all
close all
%%%%%%%%%%
START=0
L = 1000 % Total length (points)
LS = 200 % Length of signal (points)
LD = 510 % Signal appearance [points]
L1 = 490 % Expected signal appearance [points]
% L1 = 510
%%%%%%%%%%
Ts = 0.001 % Sample time [s]
Fs=1/Ts % Sample frequency [Hz]
%%%%%%%%%
T0=0.1 % Period of signal [s]
f=1/T0 % Frequency of signal [Hz]
%%%
%teta=0 % teta in degrees
teta=90
ts=input('STOP_0','s');
%%%%%%%%%%
PROCESS=1
A=1
[s,T,t]=func_sin_impulse(Fs,LS,LD,L,f,A,teta);
% s signal with lag LD
[s1,T,t]=func_sin_impulse(Fs,LS,L1,L,f,A,teta);
% s1 signal with lag L1
T=T % Total length [s]
```

```

%%%%%%%%%%
maxs=max(s)
figure(100), plot(t,s,t,maxs*1.1,t,-maxs*1.1), grid
title(['Sin impulse, frequency f = ',num2str(f),...
' amplitude A = ',num2str(A), ' phase teta in degrees =
',num2str(teta)]),
xlabel('time (sec)'), ylabel('s (Volts)')
ts=input('STOP','s');

```



Входной радиоимпульс с начальной фазой 90°

В данном случае радиоимпульс имеет ненулевую начальную фазу. Дальнейшая обработка строится в предположении, что значение начальной фазы входного радиоимпульса точно известно. Это позволяет сформировать опорный радиоимпульс коррелятора как задержанную копию входного радиоимпульса. Однако задержка опорного радиоимпульса отличается от задержки входного.

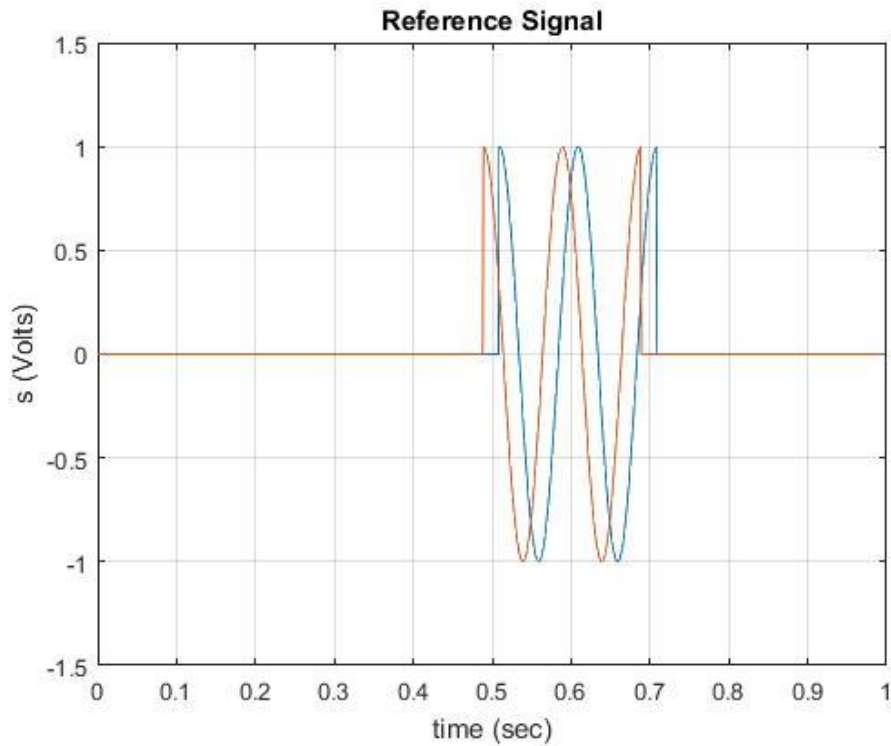
Импульсная характеристика согласованного фильтра также формируется с учетом известной начальной фазы радиоимпульса.

Тем не менее программа позволяет исследовать влияние неточного знания начальной фазы входного радиоимпульса на процессы обработки.

```

figure(101), plot(t,s,t,s1,t,maxs*1.1,t,-maxs*1.1), grid ,
title('Reference Signal')
xlabel('time (sec)'), ylabel('s (Volts)'),
ts=input('STOP','s');

```

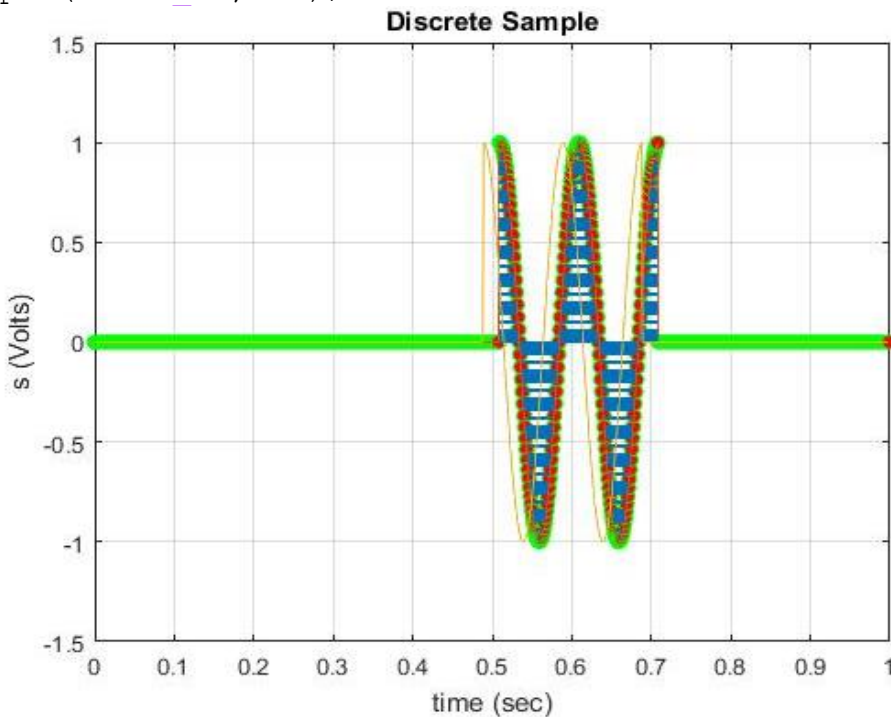


Входной и опорный радиоимпульсы (опорный импульс опережает входной)

```

%%%%
figure(102), stem(t,s,'LineStyle','-.',...
'MarkerFaceColor','red',...
'MarkerEdgeColor','green'), title('Discrete Sample'),
hold on
plot(t,s,t,s1,t,maxs*1.1,t,-maxs*1.1);
xlabel('time (sec)'), ylabel('s (Volts)'), grid
ts=input('STOP_1','s');

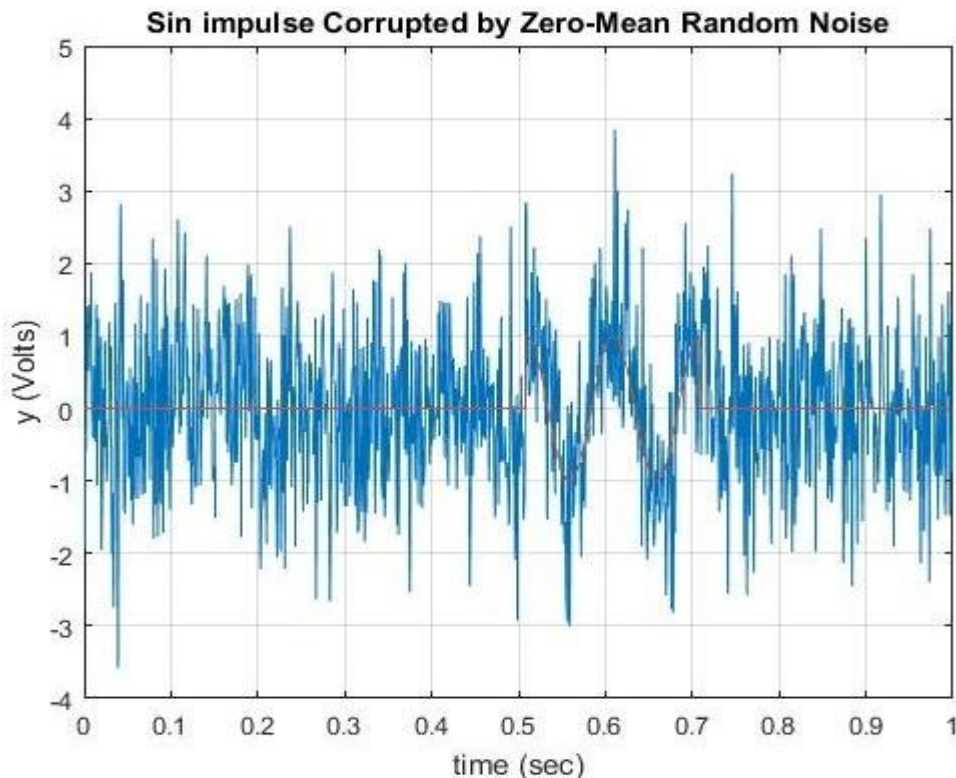
```



Входной радиоимпульс в дискретной форме

Энергия сигнала равна $E = A^2 \cdot LS \cdot Ts = 0,2 \text{ В}^2 \cdot \text{с}$.

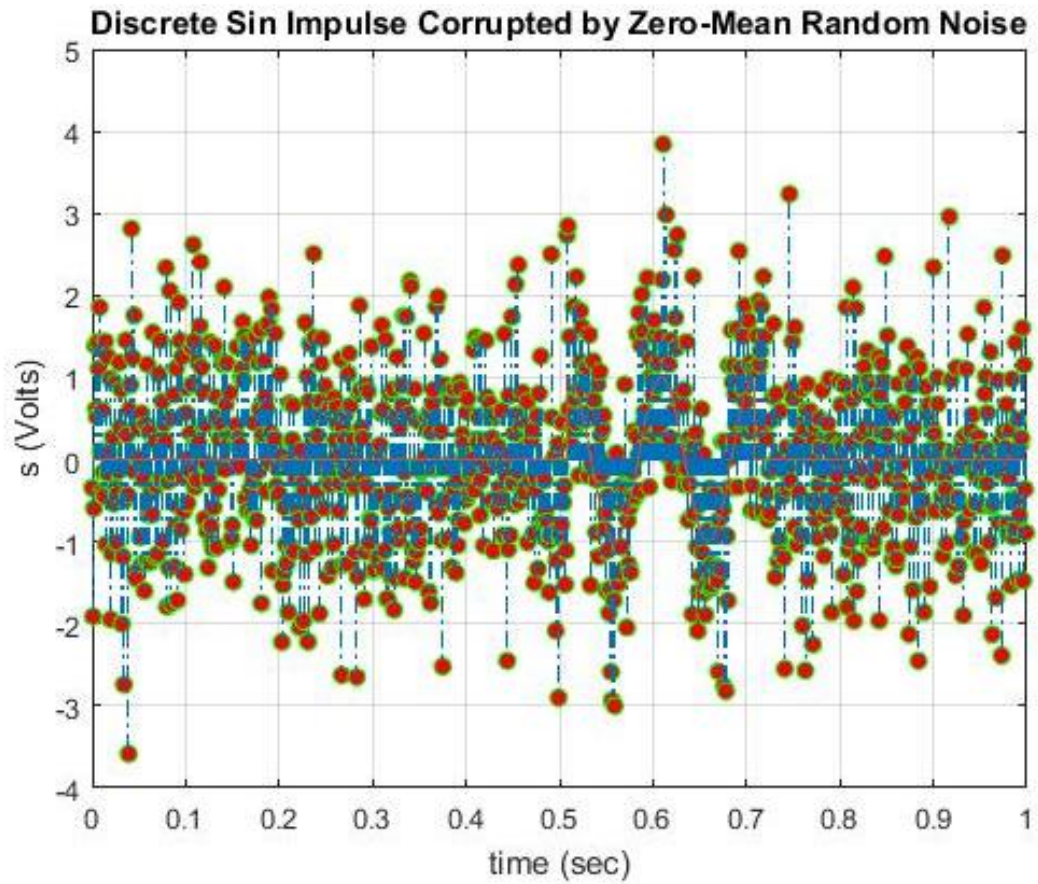
```
%%%%%%%%%%
PROCESS=2
rand('state',0);
%
y = s + randn(size(t));      % Sin impulse plus noise
% y = s;
maxy=max(y(:))
figure(200), plot(t,y,t,s,t,maxy*1.1)
title('Sin impulse Corrupted by Zero-Mean Random Noise')
xlabel('time (sec)'),ylabel('y (Volts)'), grid on
ts=input('STOP_200','s');
```



Наблюдение, содержащее входной радиоимпульс и аддитивный гауссовский шум

Отношение сигнал/шум $q^2 = A^2/2\sigma^2$ равно 1/2.

```
%%%
figure(201), stem(t,y,'LineStyle','-.',...
    'MarkerFaceColor','red',...
    'MarkerEdgeColor','green'),
title('Discrete Sin Impulse Corrupted by Zero-Mean Random Noise');
xlabel('time (sec)'), ylabel('s (Volts)'), grid on
hold on
plot(t,s,t,maxy*1.1);
ts=input('STOP_201','s');
```

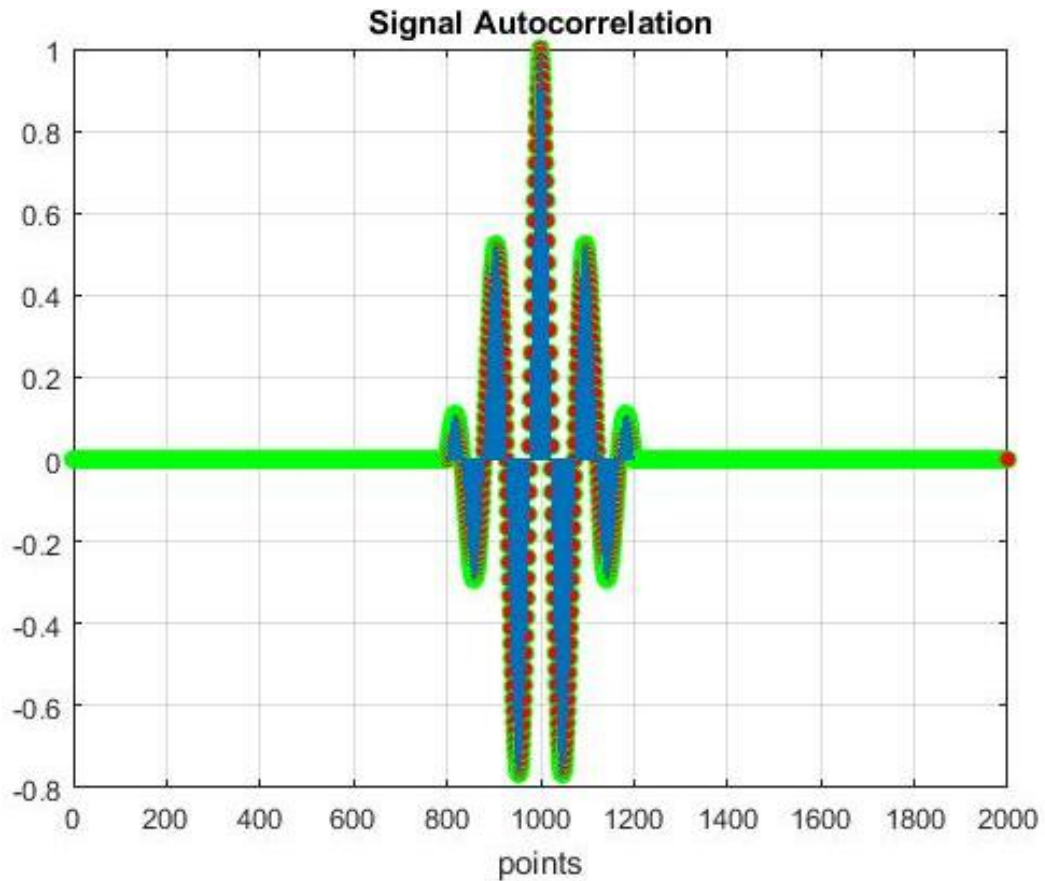



Наблюдение в дискретной форме

```

% save s.mat
% save y.mat
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
PROCESS=3
[rs,lagss] = xcorr(s,'coeff');
lrs=length(rs)
figure(300),% stem(rs), grid
% stem(rs,'LineStyle','-.',...
%       'MarkerFaceColor','red',...
%       'MarkerEdgeColor','green'),
stem(rs,'LineStyle','- ',...
      'MarkerFaceColor','red',...
      'MarkerEdgeColor','green'), title('Signal Autocorrelation'),
xlabel('points'), grid on
ts=input('STOP_3','s');

```



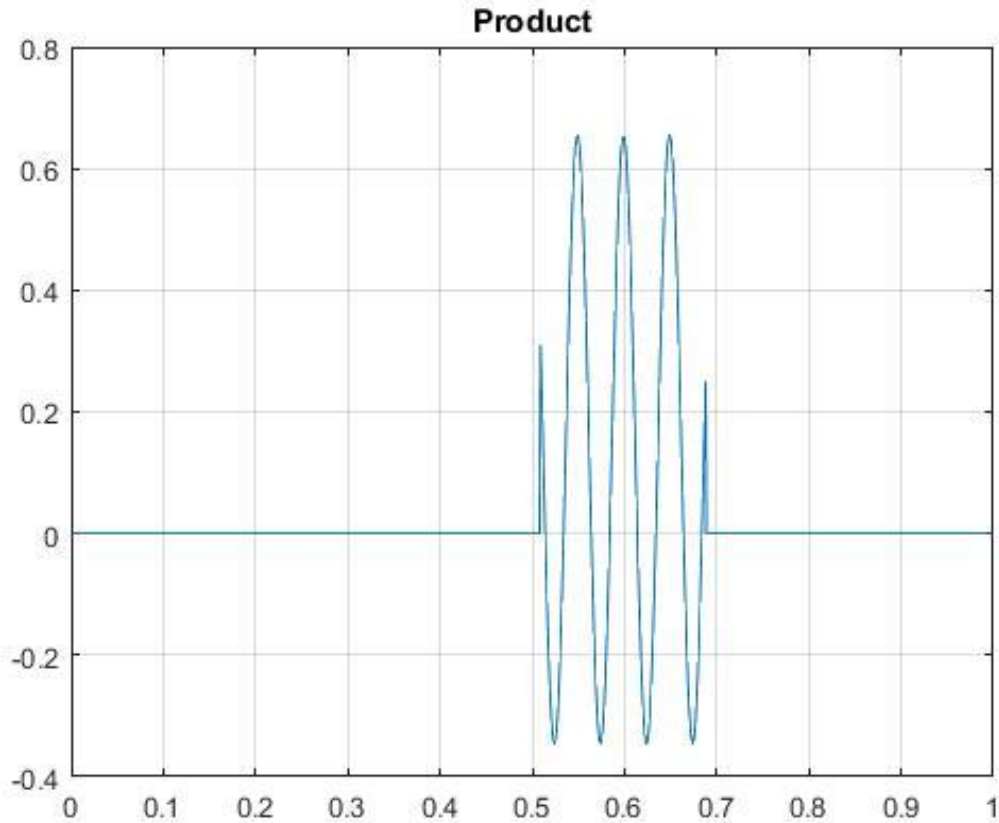
Автокорреляционная функция входного радиоимпульса,
нормированная к энергии

Отметим, что ненулевое значение начальной фазы практически не влияет на вид автокорреляционной функции, которая формируется как сигнальная составляющая на выходе согласованного фильтра.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
PROCESS=4 % CORRELATOR
LB=max(LD,L1)
LE=min(LD+LS,L1+LS)
TB=(LB-1)*Ts
TE=(LE-1)*Ts
for i=1:L
sm(i)=s(i)*s1(i);
end;
figure(400), plot(t,sm), title('Product'), grid

```



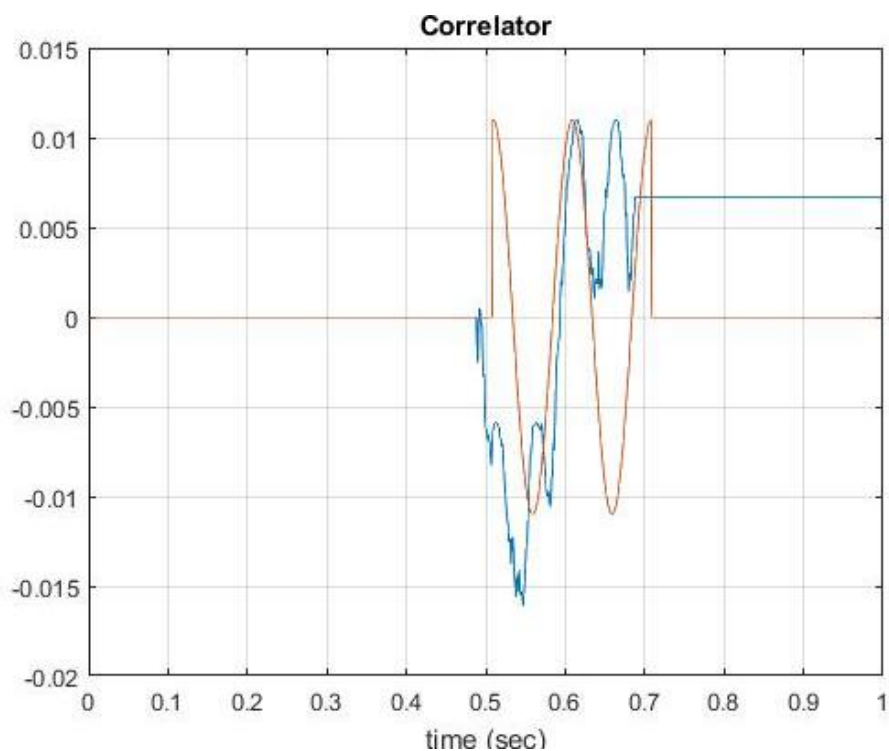
Произведение входного и опорного радиоимпульсов

Этот сигнал имеет удвоенную частоту по сравнению с исходным импульсом. При взаимных задержках, обеспечивающих полную когерентность (нулевая разность фаз), постоянная составляющая в этом произведении положительна, для противоположных сигналов (разность фаз кратна 180°) она отрицательна, а для ортогональных сигналов постоянная составляющая равна нулю.

```

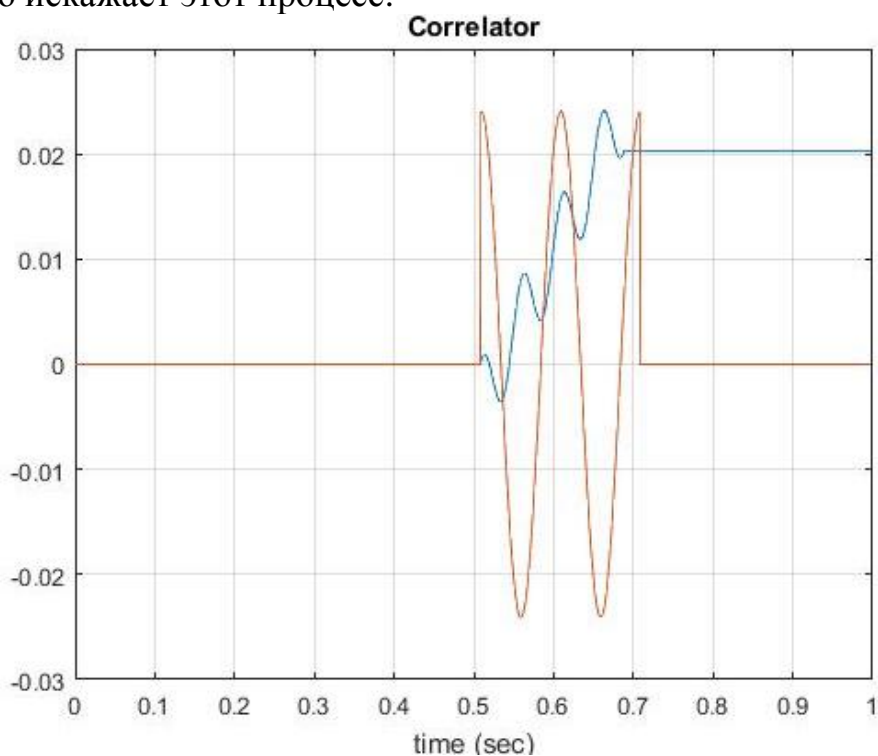
c1y(1)=s1(1)*y(1);
for i=2:L
c1y(i)=c1y(i-1)+y(i)*s1(i);
end;
c1y = c1y*Ts;
maxc1y=max(c1y(:))
figure(401), plot(t,c1y,t,maxc1y*s), title('Correlator'), grid;
xlabel('time (sec)')
hold on
plot(t,max(c1y)*1.1);
ts=input('STOP_4','s');

```



Выходной сигнал коррелятора, содержащий сигнальную и шумовую составляющие

Коррелятор формирует в качестве сигнальной составляющей значение автокорреляционной функции радиосигнала, соответствующее разности задержек входного и опорного импульсов. Это значение получается в момент окончания опорного импульса, как показано ниже. Действие шума существенно искажает этот процесс.

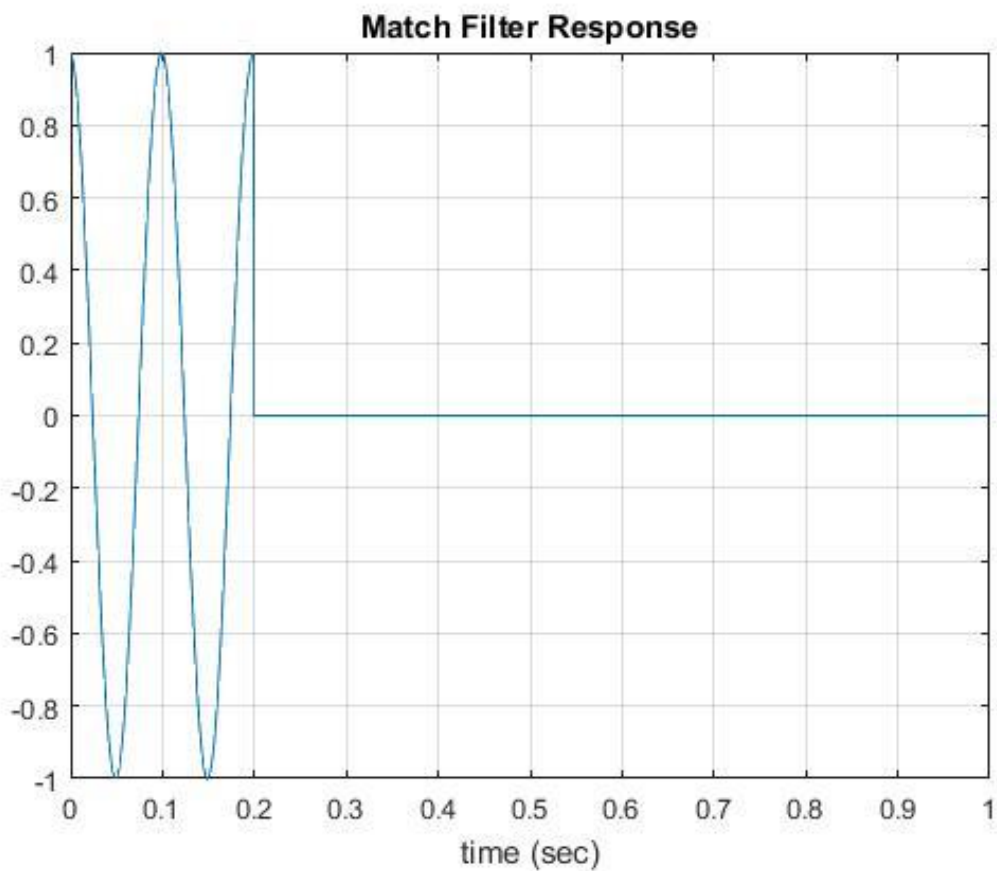


Формирование сигнальной составляющей на выходе коррелятора в отсутствие шума

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%% CONVOLUTION %%%%%%%%%%%
PROCESS=5
[s0,T1,t]=func_sin_impulse(Fs,LS,1,L,f,A,teta);
% s0 - undelayed signal
h=zeros(1,L);
for i=1:LS
    h(i)=s0(LS-i+1);
end
figure(500), plot(t,h), title('Match Filter Response'),
xlabel('time (sec)'), grid

```

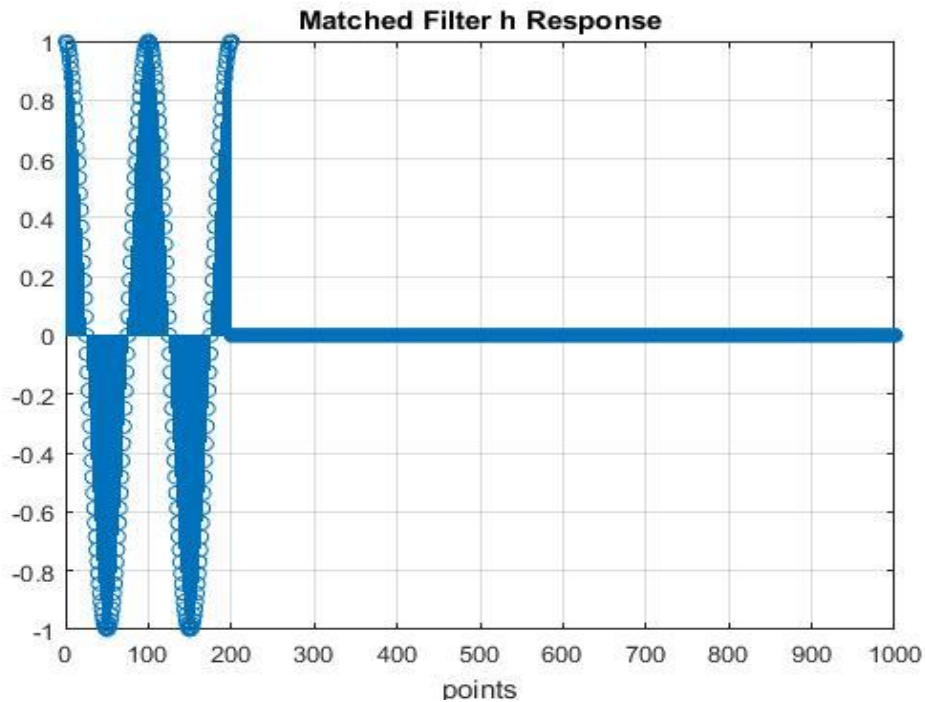


Импульсная характеристика согласованного фильтра

```

figure(501), stem(h),
title('Matched Filter h Response'), grid, xlabel('points')
ts=input('STOP_5','s');

```



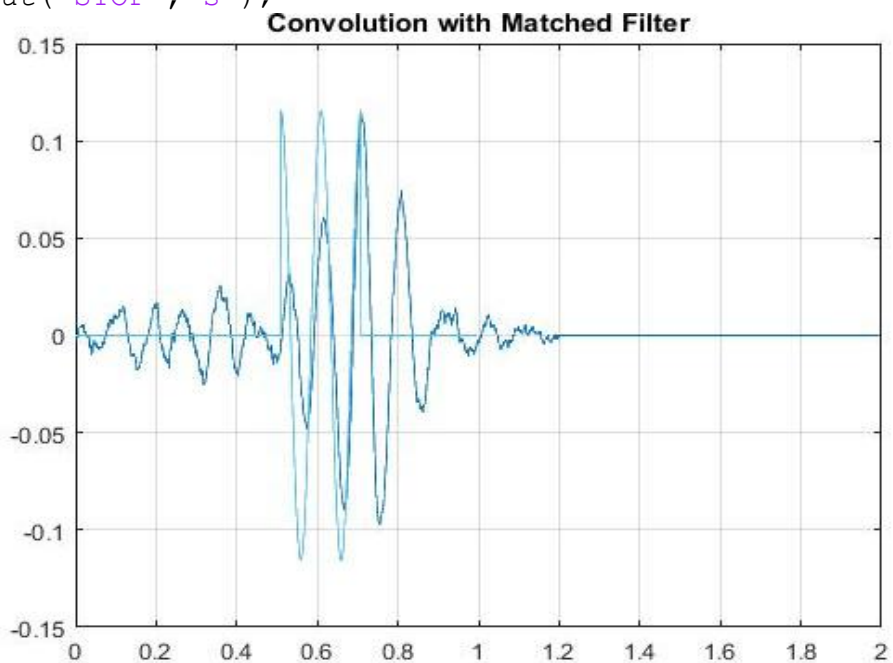
Импульсная характеристика согласованного фильтра в дискретной форме

%%%%%%%%%

```

PROCESS=6
cy=conv(y,h);
cy=cy*Ts;
maxcy=max(cy)
k1=-(L-1):(L-1);
k2=k1+LS+800;
figure(600), plot(k2*Ts,cy,k2*Ts,maxcy*1.1),
title('Convolution with Matched Filter'), grid
hold on
plot(t,s*maxcy)
ts=input('STOP','s');

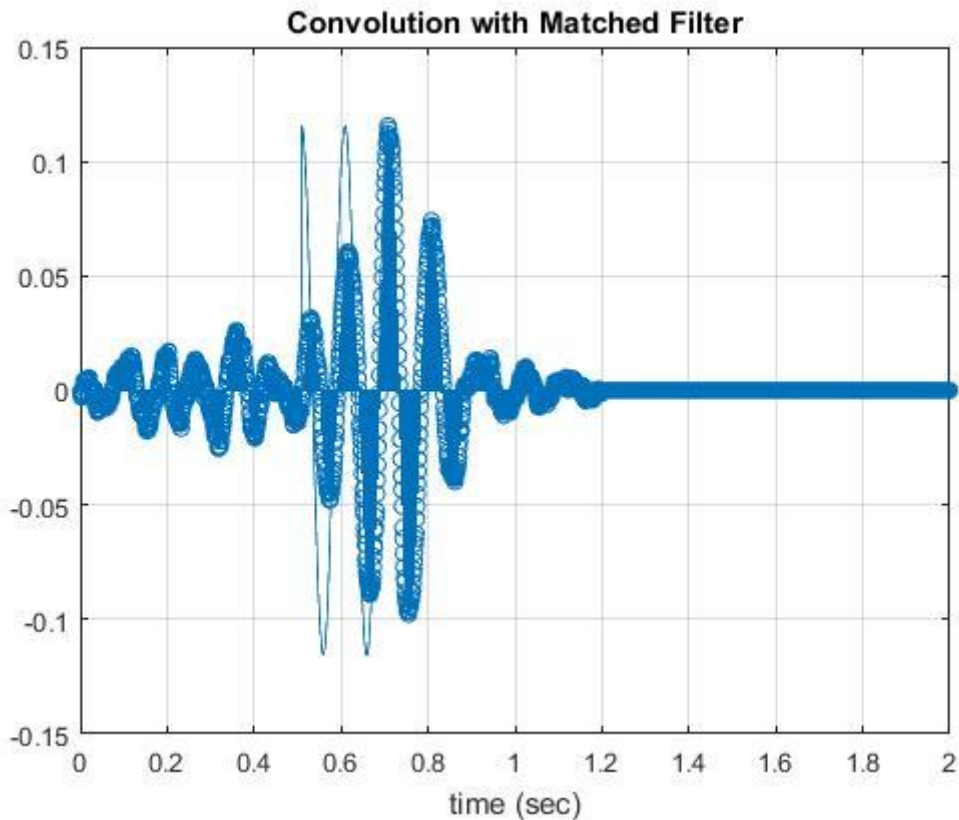
```



Результат свертки импульсной характеристики согласованного фильтра с наблюдением

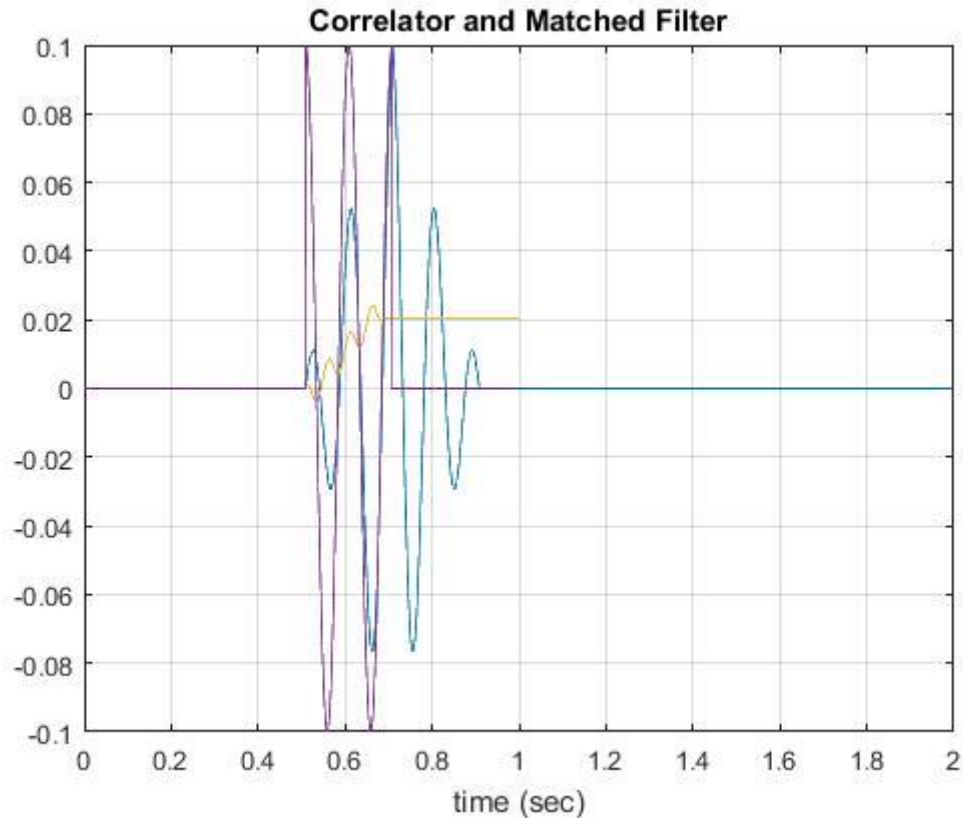
Сигнальная составляющая на выходе согласованного фильтра полностью повторяет временную автокорреляционную функцию радиоимпульса.

```
figure(601), stem(k2*Ts,cy),
title('Convolution with Matched Filter'), grid
xlabel('time (sec)')
hold on
plot(t,maxcy*1.1,t,s*maxcy);
ts=input('STOP_6','s');
```



Результат свертки в дискретной форме

```
%%%%%%%%%%%%%5
PROCESS=7
figure(700), plot(k2*Ts,cy), grid
title('Correlator and Matched Filter')
xlabel('time (sec)')
hold on
plot(t,s*maxcy,t,cly,t,s*maxcy)
ts=input('STOP_7','s');
```

Сравнение работы коррелятора и согласованного фильтра

%%%%%%%%%

END=0

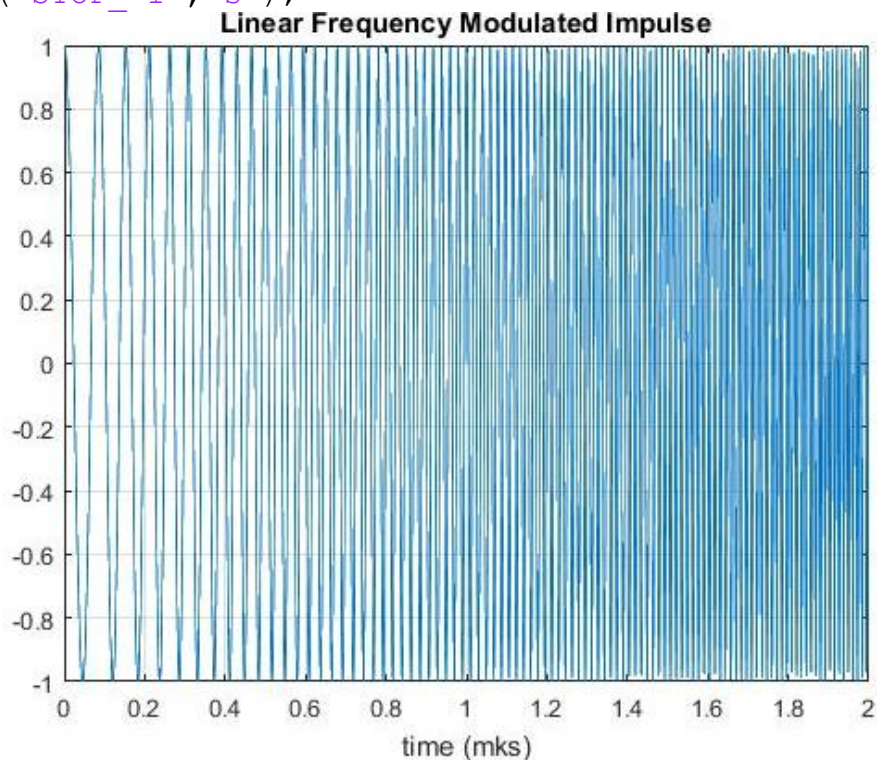
В момент окончания опорного импульса выходной сигнал коррелятора совпадает с выходным сигналом согласованного фильтра. При совпадении временных задержек входного и опорного сигналов в этот момент формируется корреляционный интеграл. При рассогласовании этих временных задержек коррелятор уже не формирует корреляционный интеграл.

Согласованный фильтр формирует корреляционный интеграл при любом запаздывании входного сигнала в момент его окончания. Говорят, что согласованный фильтр «инвариантен» к моменту прихода входного сигнала. Однако этот момент может оказаться неизвестным, что создает трудности временной селекции решающей статистики. Выбор максимального напряжения на выходе согласованного фильтра решает эту проблему только для достаточно больших отношений сигнал/шум.

4. ФОРМИРОВАНИЕ И ОБРАБОТКА ОДИНОЧНОГО ЛЧМ ПРЯМОУГОЛЬНОГО РАДИОИМПУЛЬСА НА ФОНЕ ШУМА

Программа формирует одиночный прямоугольный радиоимпульс с линейной частотной модуляцией (ЛЧМ) на фоне аддитивного гауссовского шума с вычислением амплитудного спектра и спектрограммы.

```
%main_chirp_impulse
clc
close all
clear all
START=0
% Generate a chirp with linear instantaneous frequency deviation.
%The chirp is sampled at 1 kHz for 2 seconds.
% The instantaneous frequency is f0 at t = 0
t = 0:1/1e3:2; % [ms] delta = 1/1000 ms
L=size(t,2) % 2001
f0=10 % [kHz]
f1=50 % [kHz]
T1=1
ts=input('STOP_0','s');
PROCESS=1
s = chirp(t,f0,T1,f1);
figure(100), plot(t,s); grid
title('Linear Frequency Modulated Impulse');
xlabel('time (mks)');
fa=(f1-f0)/2
ts=input('STOP_1','s');
```

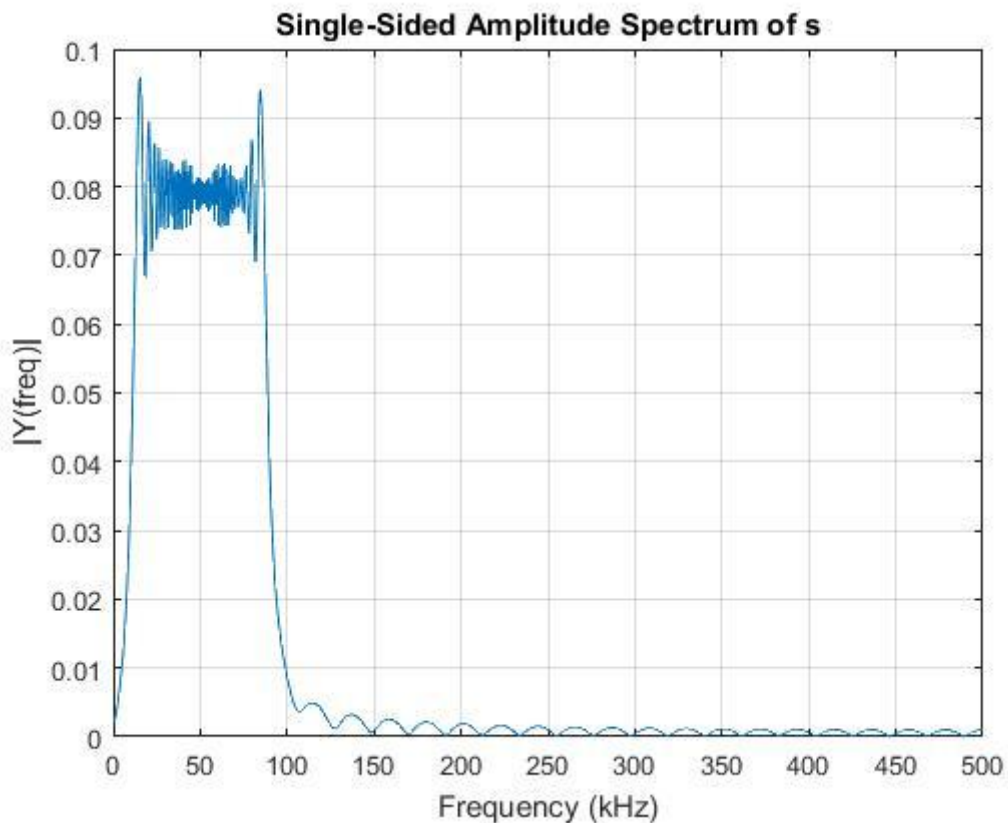


Сигнал с линейной частотной модуляцией

```

%%%%%%%%%
PROCESS=2
nextpow2(L)
NFFT = 2^nextpow2(L) % 4096 Next power of 2 from length of y
Y = fft(s,NFFT)/L;
%%%
Fs=1000% kHz
freq = Fs/2*linspace(0,1,NFFT/2+1);
% Plot single-sided amplitude spectrum.
figure(200), plot(freq,2*abs(Y(1:NFFT/2+1)))
title('Single-Sided Amplitude Spectrum of s')
xlabel('Frequency (kHz)'), ylabel('|Y(freq)|'), grid
ts=input('STOP', 's');

```

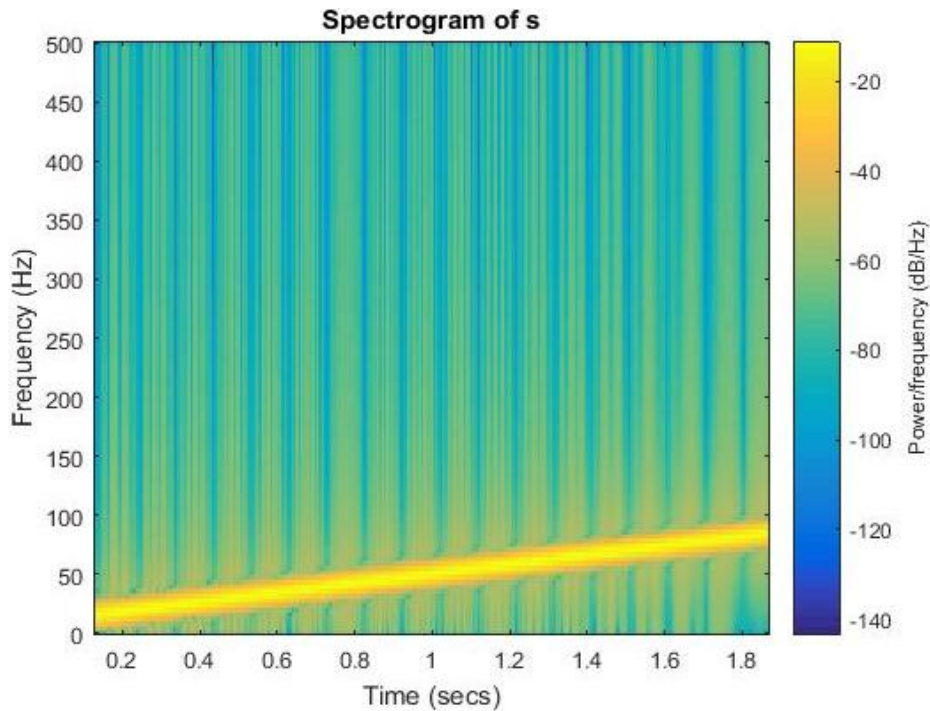


Односторонний амплитудный спектр ЛЧМ сигнала

```

%%%%%%%%%%
PROCESS=3
% Compute and plot the spectrogram of the chirp. Specify 256
DFT points,
% a Hamming window of the same length, and 250 samples of over-
lap.
figure(300),
spectrogram(s,256,250,256,1e3,'yaxis'),
title('Spectrogram of s');
ts=input('STOP_3', 's');

```

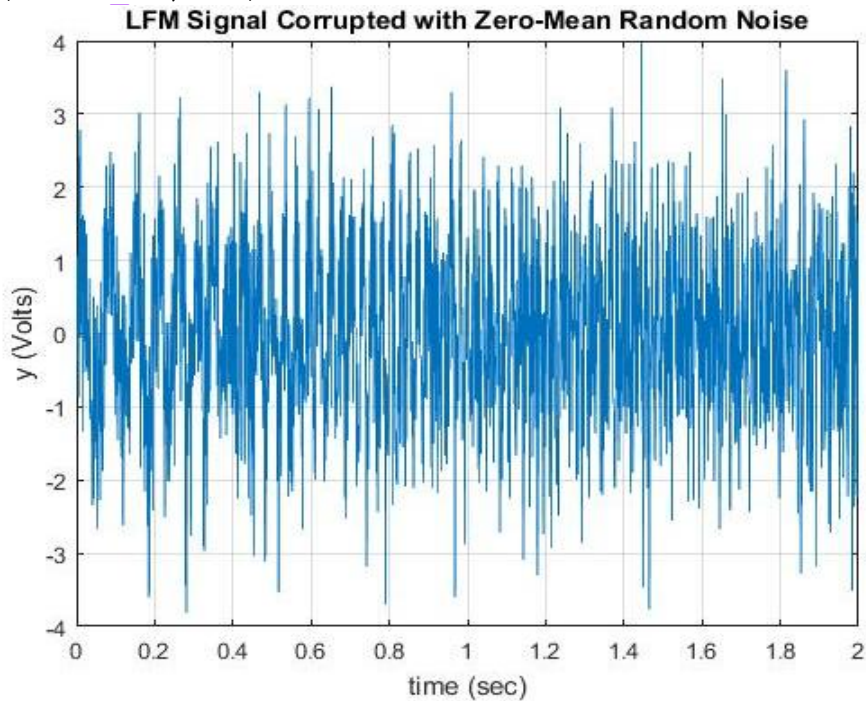


Спектрограмма (текущий спектр) ЛЧМ сигнала

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
PROCESS=4
rand('state',0);
U=1
y = U*s + randn(1,L);      % LFM impulse plus noise
figure(400), plot(t,y)
%plot(t(1:L),y(1:L))
title('LFM Signal Corrupted with Zero-Mean Random Noise')
xlabel('time (sec)'), ylabel('y (Volts)'), grid
ts=input('STOP_4','s');

```

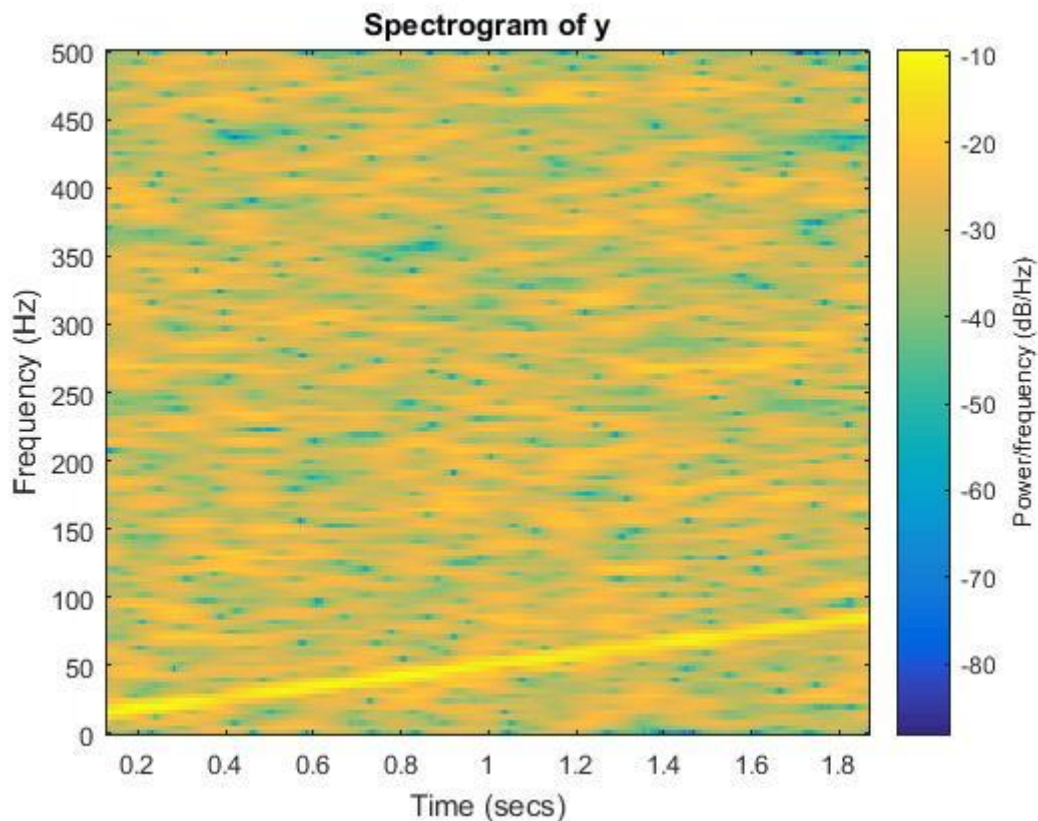


ЛЧМ сигнал в смеси с аддитивным белым шумом

```

PROCESS=5
figure(500),
spectrogram(y,256,250,256,1e3,'yaxis'),
title('Spectrogram of y')

```



Спектрограмма зашумленного ЛЧМ сигнала

```
END=0
```

Программа корреляционной обработки и согласованной фильтрации ЛЧМ импульса на фоне аддитивного гауссовского шума имеет вид:

```

%main_det_chirp_impulse
%chirp impulse corrupted by noise
%uses function = func_chirp_impulse
clc
clear all
close all
%%%%%%%%%%%%%%
START=0
Fs = 1000 % Sample Frequency [Hz]
Ts=1/Fs % Sample time [s]
L = 3000 % Total length (points)
LS = 1000 % Length of signal (points)
LD = 1000 % Signal appearance [points]
L1 = 1050 % Expected signal appearance [points]
f0 = 0.1 %[kHz]
f1=5

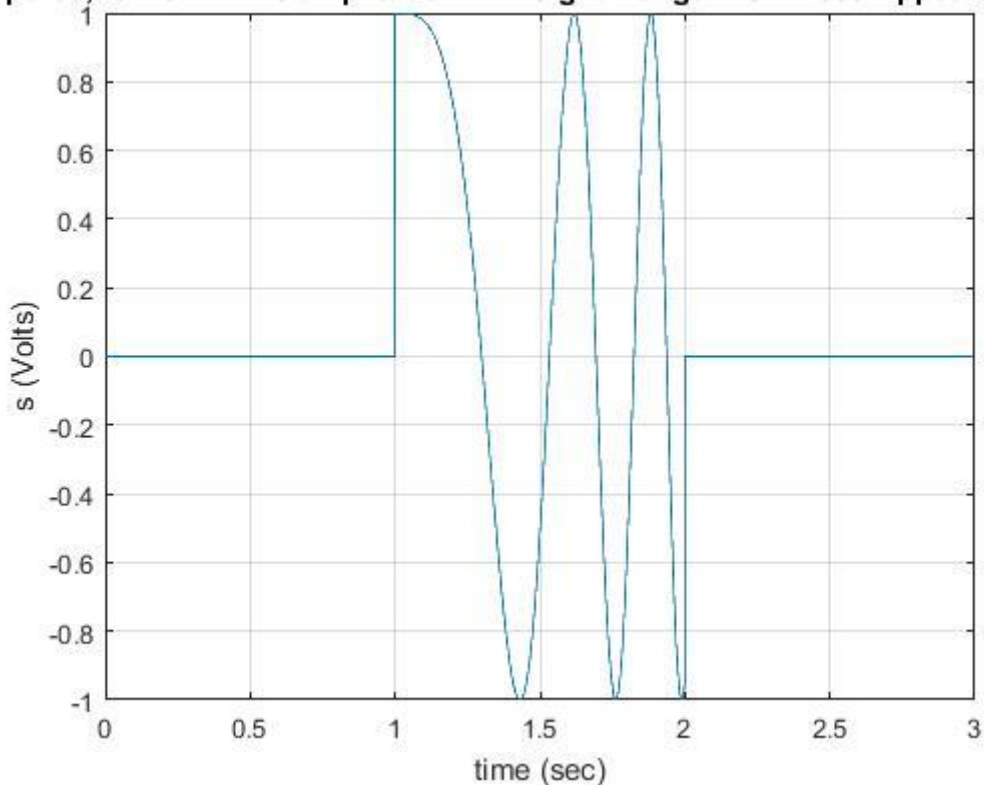
```

```

ts=input('STOP_0','s');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
PROCESS=1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[T,t,s]=func_chirp_impulse(Ts,LS,LD,L,f0,f1);
% s signal with lag LD
[T,t,s1]=func_chirp_impulse(Ts,LS,L1,L,f0,f1);
% s1 signal with lag L1
ls=length(s) % 3001
ls1=length(s1) % 3001
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(100), plot(t,s), grid
title(['Chirp impulse, f0 = ',num2str(f0),' f1 = ',num2str(f1)...
' amplitude A = ',num2str(max(s)),...
' Signal length LS = ',num2str(LS),...
' Appearance LD = ',num2str(LD)]),
xlabel('time (sec)'), ylabel('s (Volts)')
ts=input('STOP','s');

```

Chirp impulse, f0 = 0.1 f1 = 5 amplitude A = 1 Signal length LS = 1000 Appearance LD = 1

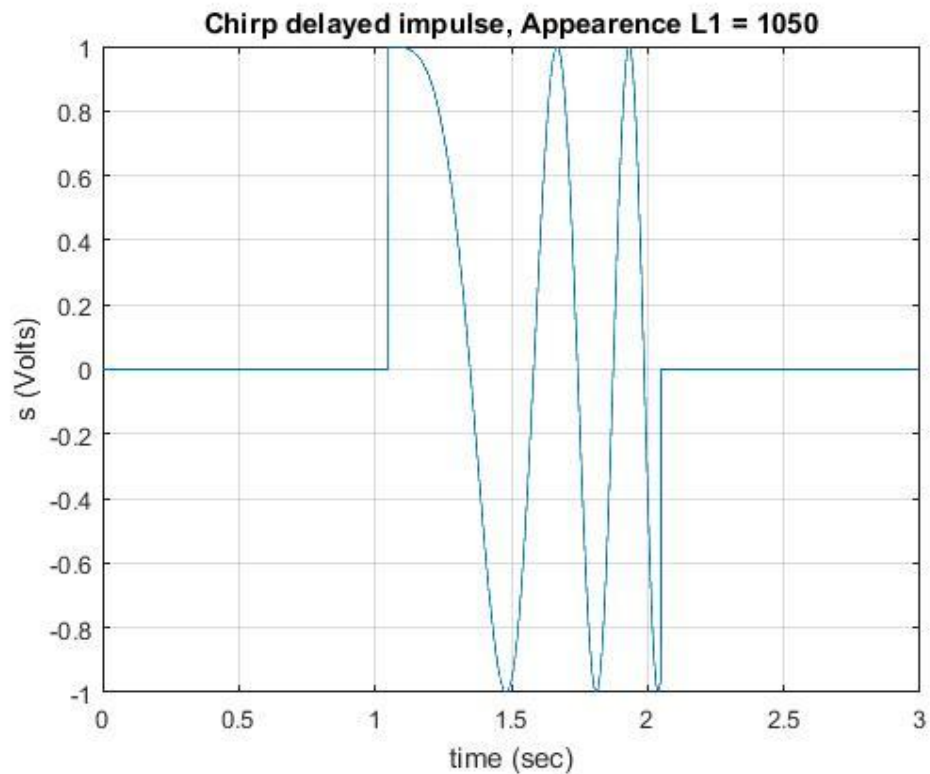


ЛЧМ импульс прямоугольной формы

```

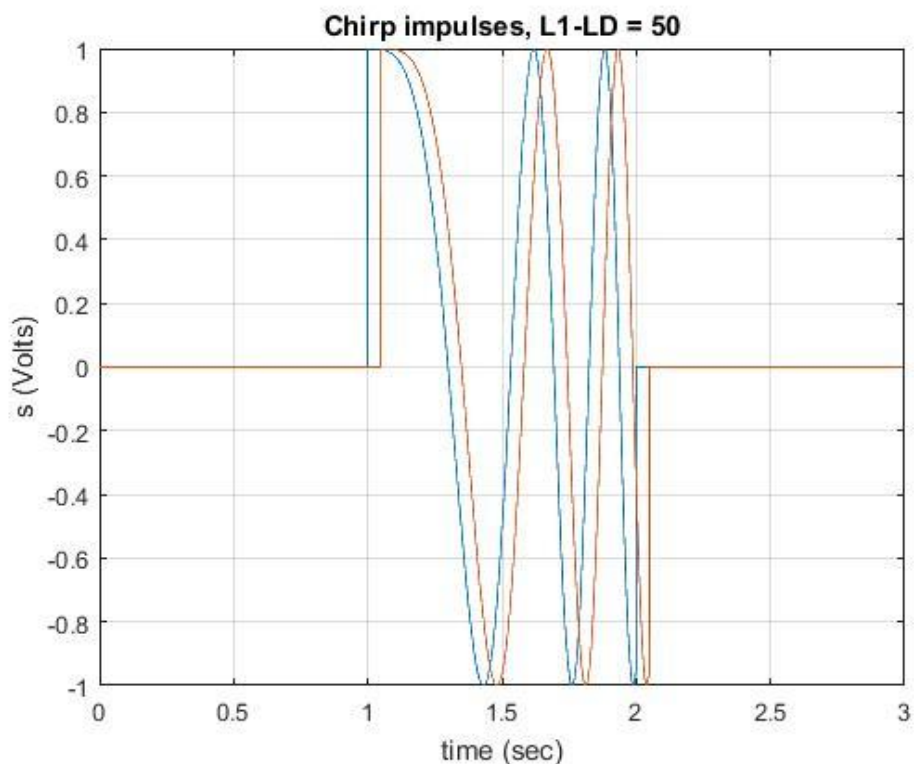
figure(101), plot(t,s1), grid
title(['Chirp delayed impulse, Appearance L1 = ',num2str(L1)]),
xlabel('time (sec)'), ylabel('s (Volts)')
ts=input('STOP','s');

```

Опорный сигнал (ожидаемый ЛЧМ импульс)

```
figure(102), plot(t,s,t,s1), grid
title(['Chirp impulses, L1-LD = ', num2str(L1-LD)]),
xlabel('time (sec)'), ylabel('s (Volts)')
ts=input('STOP_100', 's');
```

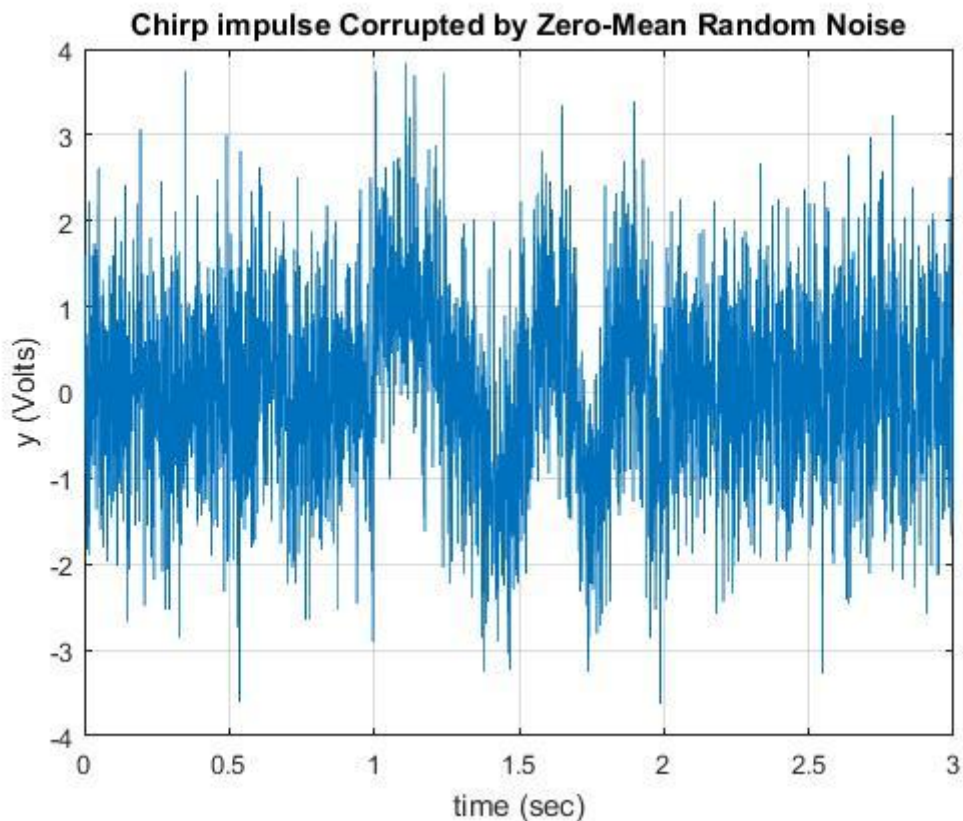


Входной и опорный ЛЧМ импульсы

```

%%%%%%%%%%
PROCESS=2
rand('state',0);
U=1
y = U*s + randn(size(t));      % Chirp impulse plus noise
figure(200), plot(t,y)
title('Chirp impulse Corrupted by Zero-Mean Random Noise')
xlabel('time (sec)'),ylabel('y (Volts)'), grid on
ts=input('STOP_200','s');

```

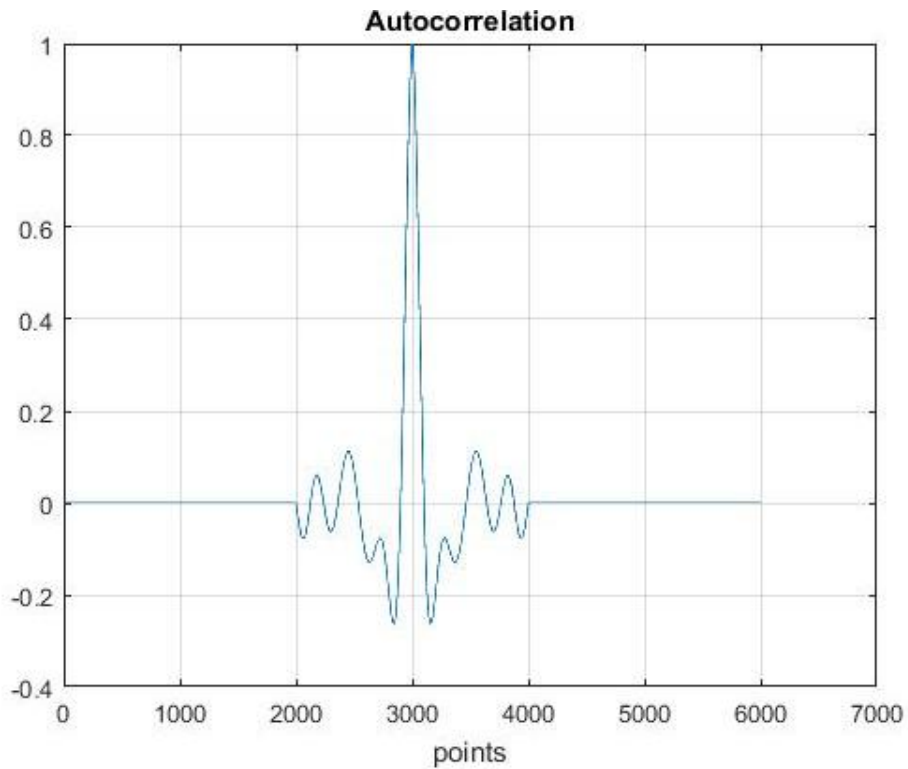


ЛЧМ импульс в смеси с белым гауссовским шумом

```

%%%%%%%%%%
PROCESS=3
[rs,lagss] = xcorr(s,'coeff');
lrs=length(rs) % 6001
%
figure(300), plot(rs),
title('Autocorrelation'),
xlabel('points'), grid
ts=input('STOP','s');

```

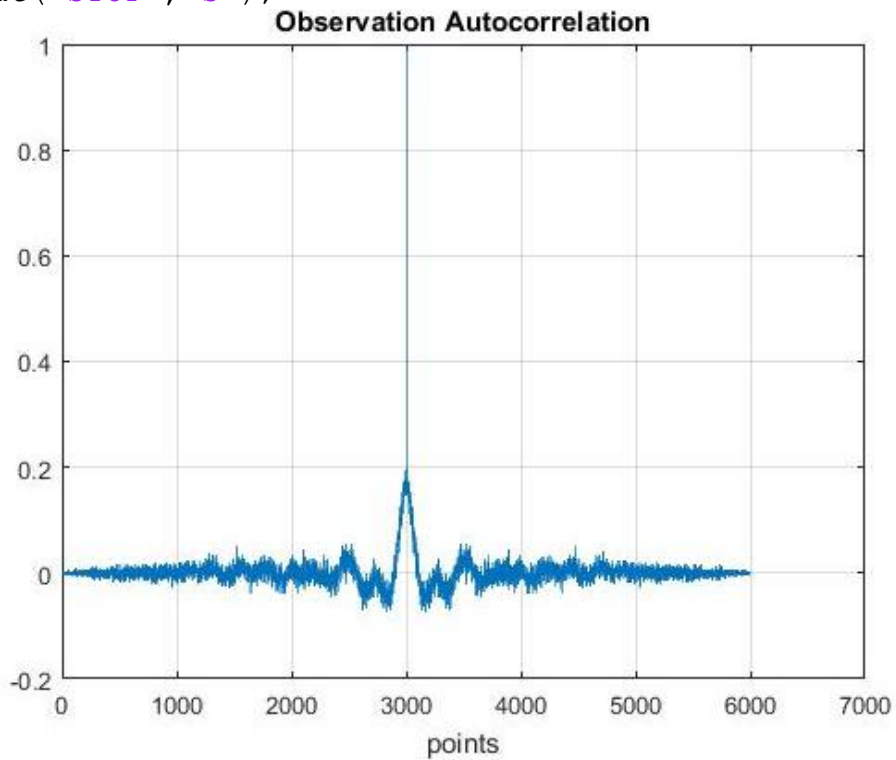


Автокорреляционная функция ЛЧМ импульса

```

%%%%%%%%%   XCORREL(y)   %%%%%%%%%%%%%%%
[ry,lagsy] = xcorr(y, 'coeff');
lry=length(ry) % 6001
figure(302), plot(ry), grid
title('Observation Autocorrelation'),
xlabel('points')
ts=input('STOP', 's');

```



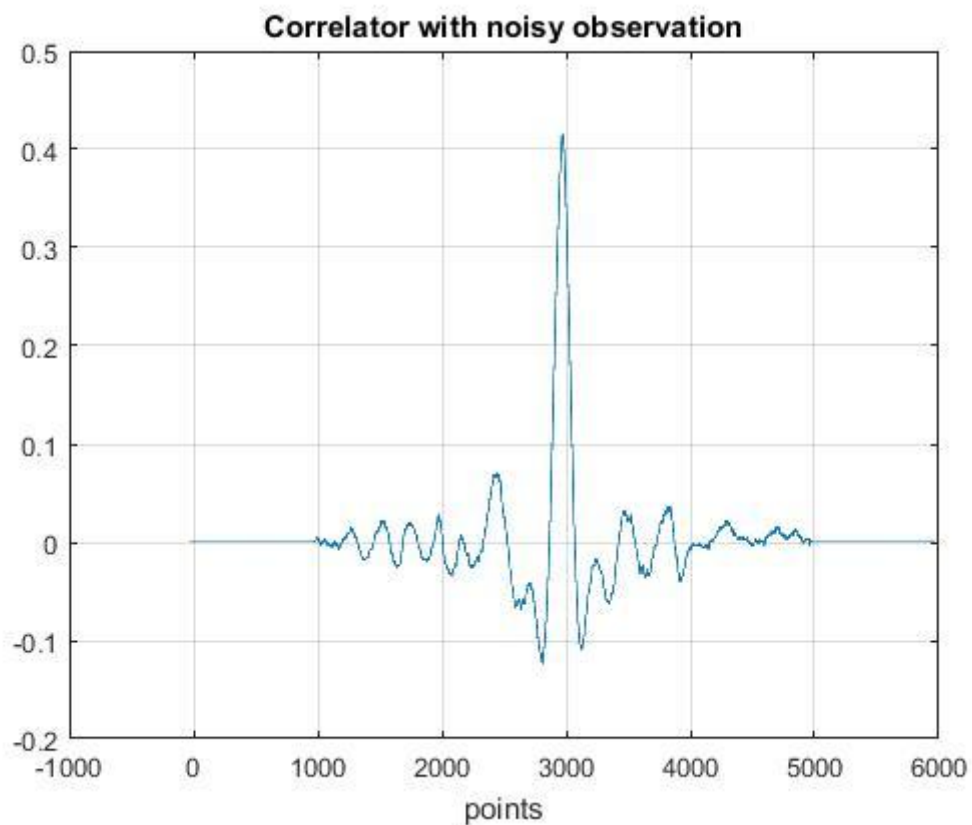
Автокорреляция наблюдения

Автокорреляция наблюдений при небольших рассогласованиях существенно ниже, чем временная автокорреляционная функция сигнала. Это определяет преимущества корреляционного приема (вычисление взаимной корреляции наблюдения с сигналом) перед автокорреляционным.

```

%%%%%%%%%%%%% XCORREL(x,y) %%%%%%%%%%%%%%
[rys,lagsys] = xcorr(y,s,'coeff');
lrys=length(rys)
figure(303), plot(1-31:lrys-31,rys),
title('Correlator with noisy observation'),
xlabel('points'), grid
ts=input('STOP_3','s');

```



Взаимная корреляция наблюдения и сигнала

```

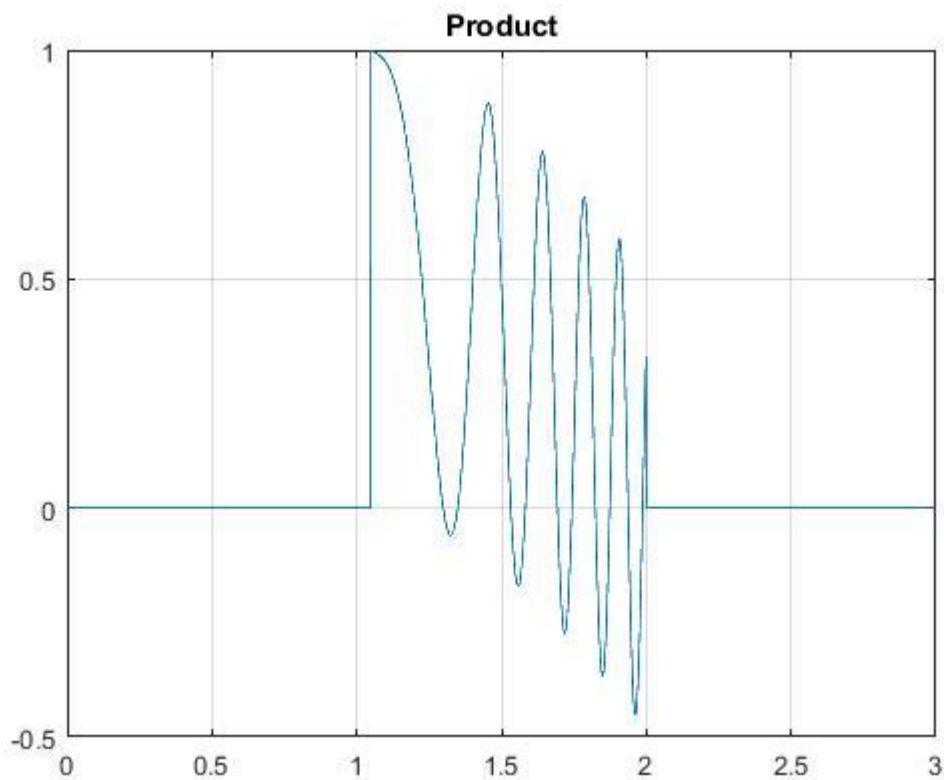
%%%%%%%%%%%%%
PROCESS=4 % CORRELATOR
LB=max(LD,L1) % 1200
LE=min(LD+LS,L1+LS) % 2000
TB=(LB-1)*Ts % 1.199
TE=(LE-1)*Ts % 1.999
sm=zeros(size(t));

```

```

for i=1:L+1
sm(i)=s(i)*s1(i);
end;
lsm=length(sm) % 3001
lt=length(t) % 3001
figure(400), plot(t,sm), title('Product'), grid
cly(1)=s1(1)*y(1);
for i=2:L+1
cly(i)=cly(i-1)+y(i)*s1(i);
end;
lcly=length(cly) % 3000
maxcly=max(cly(:)) %
ts=input('STOP', 's');

```

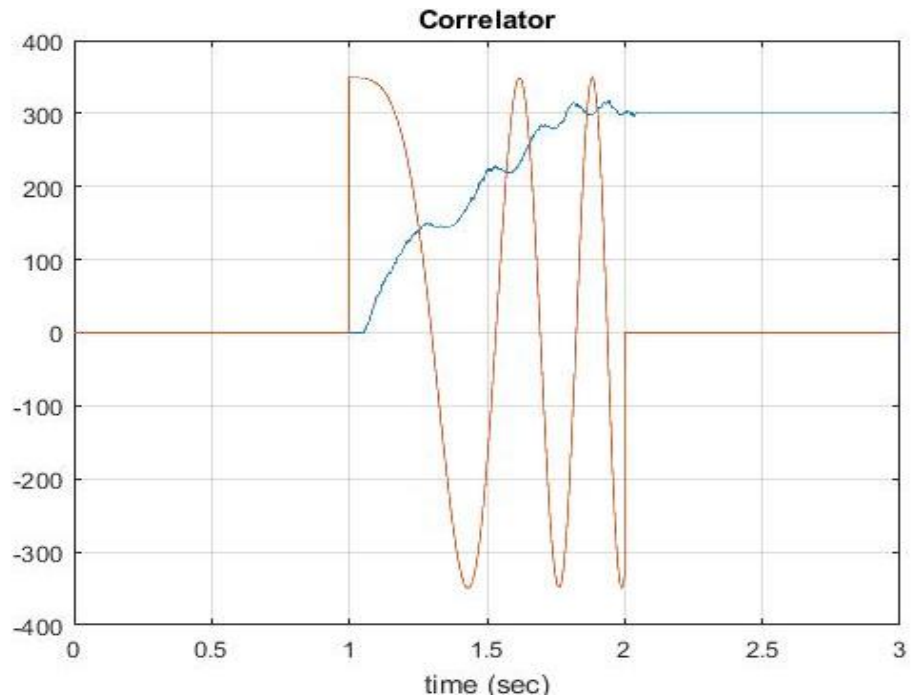


Результат перемножения входного и опорного ЛЧМ импульсов

```

figure(401),
plot(t,cly,t,maxcly*s*1.1),
title('Correlator'), grid;
xlabel('time (sec)')
hold on
plot(t,max(cly)*1.1);
ts=input('STOP_4', 's');

```

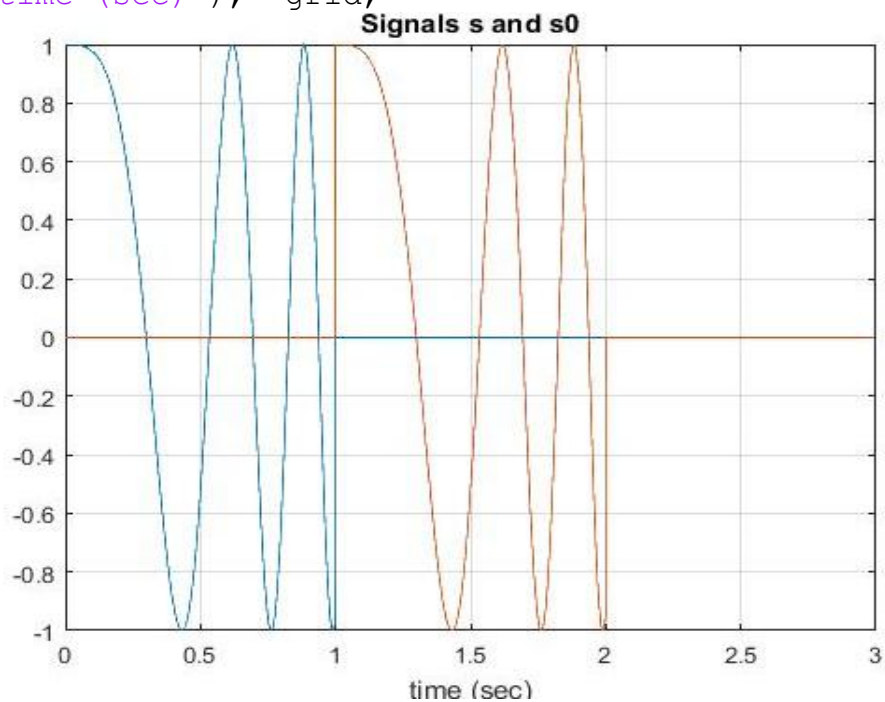


Результат корреляционной обработки

```

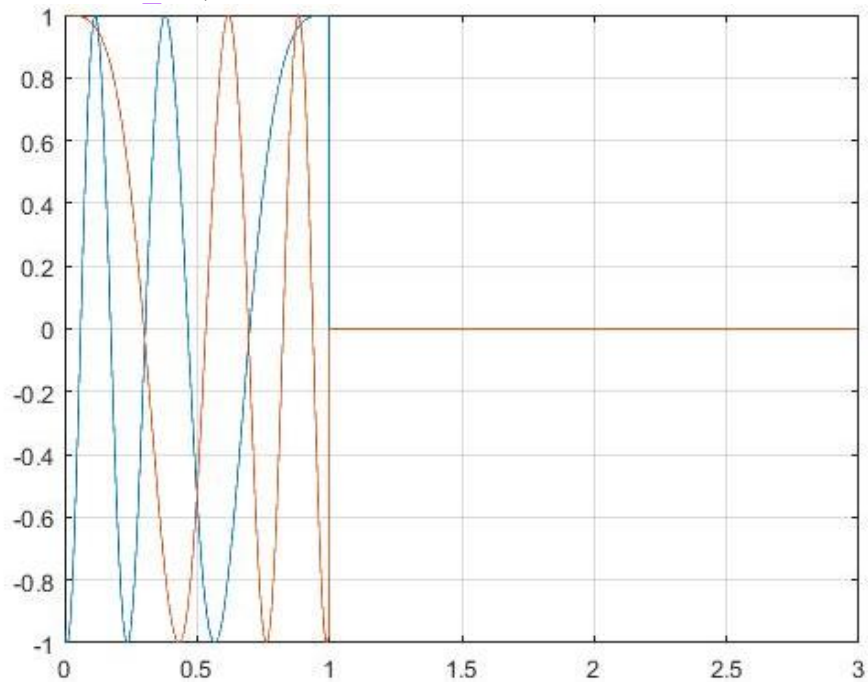
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
PROCESS=5
[T,t,s0]=func_chirp_impulse(Ts,LS,1,L,f0,f1);
% s0 - undelayed signal
ls0=length(s0) % 3001
h=zeros(1,L+1);
for i=1:LS
    h(i)=s0(LS-i+1);
end
figure(500), plot(t,s0,t,s),
title('Signals s and s0'),
xlabel('time (sec)'), grid,

```



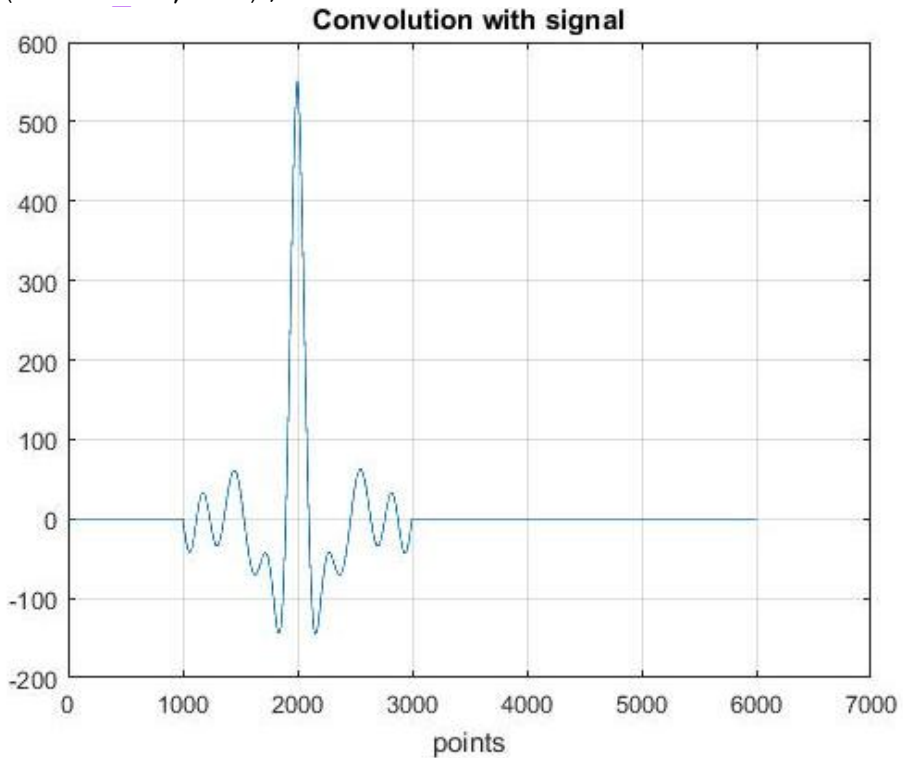
Задержанный и незадержанный ЛЧМ импульсы

```
figure(501), plot(t,h,t,s0), grid
ts=input('STOP_5','s');
```



Импульсная характеристика согласованного фильтра для ЛЧМ импульса

```
%%%%%%%%%%
PROCESS=6
cs=conv(s,h);
figure(600), plot(cs),
title('Convolution with signal'), grid
xlabel('points')
ts=input('STOP_6','s');
```



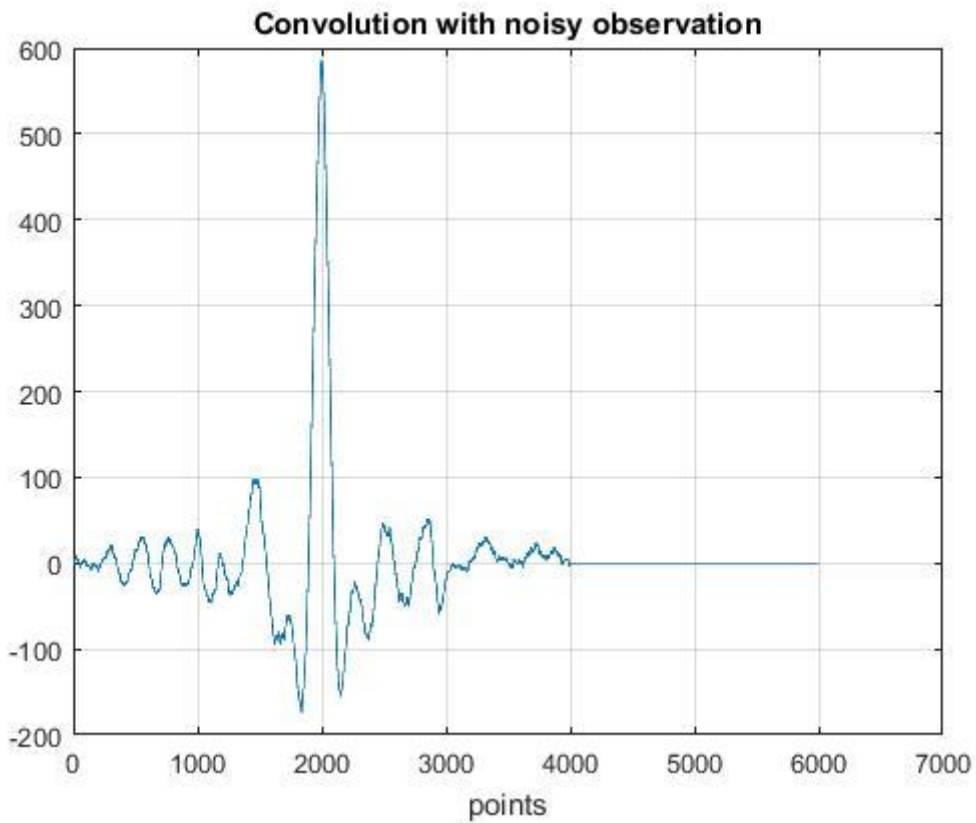
Сигнальная составляющая на выходе согласованного фильтра

Сигнальная составляющая иллюстрирует «сжатие» ЛЧМ импульса по времени.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
PROCESS=7
cy=conv(y,h);
lcy=length(cy)% 6001
maxcy=max(cy)
t1=1:6001;
figure(700), plot(t1,cy),
title('Convolution with noisy observation'), grid
xlabel('points')
ts=input('STOP_7','s');

```

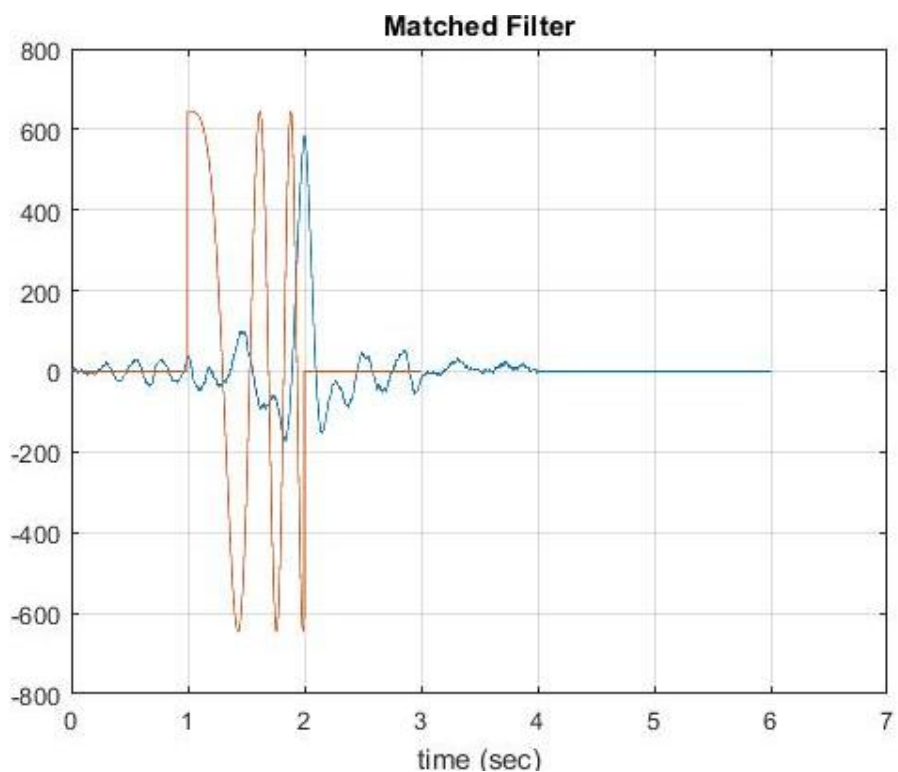


Результат согласованной фильтрации ЛЧМ импульса

```

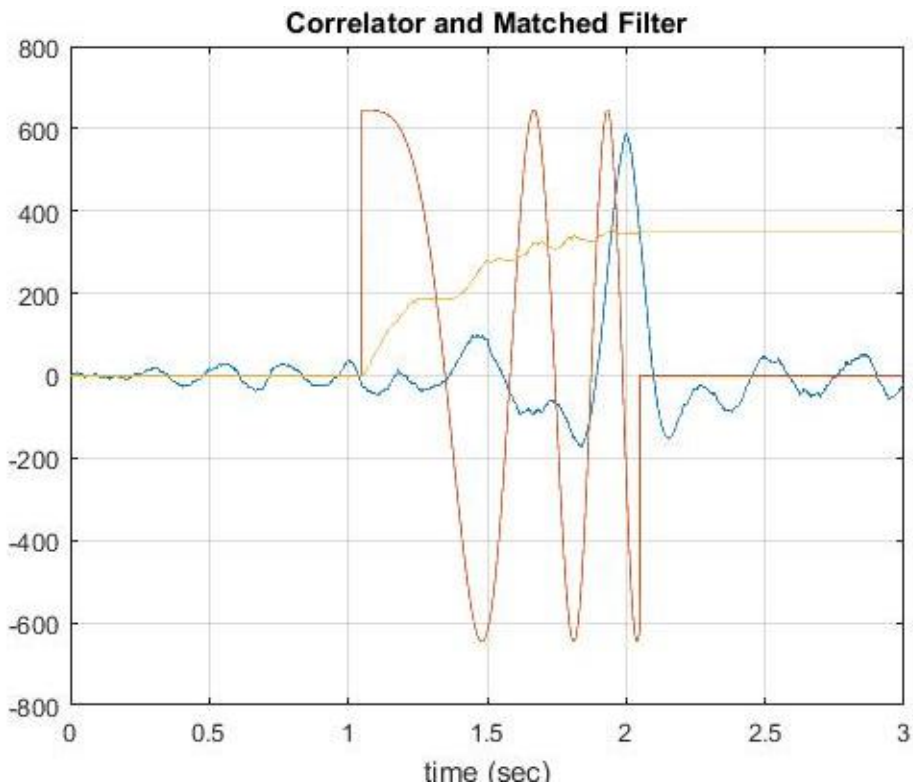
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
PROCESS=8
figure(800), plot(t1*Ts,cy), grid
title('Matched Filter')
xlabel('time (sec)')
hold on
plot(t,s*maxcy*1.1)

```



Результат согласованной фильтрации и входной ЛЧМ импульс

```
figure(801), plot(Ts*t1(1:L),cy(1:L)),
title('Correlator and Matched Filter')
xlabel('time (sec)')
hold on
plot(t,s1*maxcy*1.1,t,cly), grid
%%%%%%%%%%%%%%
END=0
```



Сравнение процессов на выходе коррелятора и согласованного фильтра

На иллюстрации показан опорный сигнал. В момент окончания входного ЛЧМ импульса согласованный фильтр формирует пик. Коррелятор формирует выходной сигнал в момент окончания опорного ЛЧМ импульса. В этот момент выходные сигналы коррелятора и фильтра одинаковы.

Программа использует функцию `func_chirp_impulse`, которую следует поместить в общую папку.

```
function [T,t,s_res]=func_chirp_impulse(Ts,LS,LD,L,f0,f1)
% LS Length of signal [points]
% LD Point of appearance (lag)[points]
% L Length of sample [points]
% f0 initial signal frequency [Hz]
% f1 final signal frequency [Hz]
% Ts Sample time [s]
t = 0:Ts:L*Ts; % Time vector
T=L*Ts; % Time of sample [s]
TS=LS*Ts; % Signal duration [s]
S = chirp(t,f0,TS,f1);
s_res=zeros(size(t));
for i=LD:LD+LS
    s_res(i)=S(i-LD+1);
end;
end
```

ЗАДАНИЯ

1. Рассчитать оптимальные пороги обнаружения и вероятности ошибок обнаружения прямоугольного импульса в шуме для двух критериев оптимальности (критерия Неймана-Пирсона, критерия максимального правдоподобия), смоделировать процесс принятия решения. Построить рабочие характеристики приемника и характеристики обнаружения, отметить на них рабочие точки. Сделать это для трех рассмотренных видов импульсов.

2. Моделировать пачку прямоугольных видеоимпульсов и реализовать корреляционную и фильтровую обработки сигналов, принимаемых в шуме. Рассчитать характеристики и определить выигрыши в отношении сигнал/шум.

3. Моделировать пачку прямоугольных радиоимпульсов и реализовать корреляционную и фильтровую обработки сигналов, принимаемых в шуме.

4. Рассмотреть обнаружение сигналов на фоне коррелированного гауссовского шума. Реализовать оптимальный алгоритм. Составить программу в Matlab. Исследовать влияние коррелированности шумовых отсчетов на характеристики обнаружения. Сравнить работу алгоритма и характеристики обнаружения для случаев коррелированного и некоррелированного шума.

5. Рассмотреть обнаружение сигналов на фоне некоррелированного негауссовского шума. Предложить алгоритм обнаружения для выбранной модели шума. Сравнить характеристики со случаем гауссовского шума.

6. Исследовать чувствительность алгоритмов обнаружения к изменению параметров распределений. Определить существенные относительные отклонения параметров. Предложить варианты обнаружения при априорной неопределенности.

7. Составить программу для коррелятора и согласованного фильтра в случае приема ЛЧМ импульса на фоне белого гауссовского шума.

8. Предложить алгоритм и реализовать программу в Matlab для обнаружения детерминированного сигнала на фоне шума в спектральной области.

9. Моделировать процессы автокорреляционного приема для рассмотренных моделей полезного сигнала. Сравнить с результатами оптимальной обработки.

10. Моделировать «энергетический» приемник для рассмотренных моделей полезного сигнала. Сравнить с результатами оптимальной обработки.

Волков Владимир Юрьевич

**МАТЕМАТИЧЕСКИЕ МЕТОДЫ
В ТЕОРИИ РАДИОТЕХНИЧЕСКИХ СИСТЕМ**

ПРОГРАММЫ МОДЕЛИРОВАНИЯ В MATLAB

Учебное пособие

План издания 2018 г., п. 19 б

Редактор *И. И. Щенсяк*

Подписано к печати 06.07.2018
Объем 3,0 усл.-печ. л. Тираж 26 экз. Заказ 867

Редакционно-издательский отдел СПбГУТ
193232 СПб., пр. Большевиков, 22

Отпечатано в СПбГУТ