

ГУАП

КАФЕДРА 41

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

доцент, канд. техн. наук,

доцент

должность, уч. степень, звание

подпись, дата

О. О. Жаринов

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №4

ПРИМЕНЕНИЕ ФОРМИРУЮЩИХ ФИЛЬТРОВ ДЛЯ СОЗДАНИЯ ШУМОПОДОБНЫХ
АУДИОСИГНАЛОВ

по курсу: Мультимедиа технологии

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. №

4016

подпись, дата

А. А. Абрамочкин

инициалы, фамилия

Санкт-Петербург 2024

Цель работы

Изучить методологию создания шумоподобных сигналов на основе формирующих фильтров.

Теоретические сведения

Один из способов создания аудиосигналов с эмулированием шума и желаемыми акустическими свойствами — это использование метода формирования через фильтр. Суть этого метода заключается в том, чтобы пропустить сигнал от искусственного источника белого шума через определенный тип фильтра. В зависимости от параметров фильтра выходной сигнал может иметь определенный "окрашенный" спектр, что вызывает определенные звуковые восприятия. Например, при использовании узкополосного фильтра, шум может восприниматься как "звнящий", а при использовании нижнечастотного фильтра результатом может быть глухой шум, напоминающий звуки водопада, фонтана, шелест листвы и т. д.

Смешивание различных шумов в разных пропорциях, а также изменение интенсивности одного или нескольких процессов, позволяет получать разнообразные звуковые эффекты.

Важно отметить, что метод формирования через фильтр - это лишь один из подходов к синтезу звуковых сигналов, и его возможности ограничены. Например, низкочастотные фильтры хорошо подходят для имитации звуков природы (шум листвы, дождя, моря и т. д.), в то время как узкополосные фильтры могут использоваться для создания разнообразных искусственных звуков и необычных эффектов. Модуляция интенсивности по случайному или гармоническому закону также способствует разнообразию звуковых эффектов.

Ход работы

Данная лабораторная работа не предполагает индивидуального варианта, однако требует применения полученных ранее знаний в области обработки аудиосигналов средствами фильтрации.

В ходе работы требуется сформировать аудиосигнал, используя шумы и примененные к ним фильтры.

Перед тем, как приступить к созданию собственной аудиодорожки, необходимо запустить программу, приведенную в методических указаниях и поэкспериментировать с границами фильтрации.

Ниже на листинге 1 можно рассмотреть код для формирования шумоподобного сигнала с динамическим изменением интенсивности звука. В ходе экспериментов будут изменяться значения переменных **f1_low** и **f1_up**.

Листинг 1. Программный код, формирующий шумоподобный сигнал с динамическим изменением интенсивности звука

```
clc, clear, close all
Fd = 44100; % задаем стандартную частоту дискретизации в Гц
Td = 1./Fd; % период дискретизации – обратный частоте
Ts = 10; % задаем длительность формируемого сигнала в секундах
N = Fd*Ts; % количество дискретных отсчетов
x1 = randn(N,2); % создаем первый источник белого шума
% параметры основного шума - граничные частоты в Гц:
f1_low = 700; f1_up = 1300;
% от этих параметров очень сильно зависит характер звука!!
% применяем полосовой фильтр для создания первого аудиосигнала:
y1 = bandpass(x1, [f1_low, f1_up], Fd);
x2 = randn(N,2); % создаем второй источник белого шума
% параметр низкочастотного процесса, для последующего
% использования при изменении интенсивности звука, в Гц
f2_low = 0.05; % в среднем период колебаний составит 20 сек
% в качестве второго формирующего фильтра задаем ФНЧ:
[b a] = butter(2, 2*f2_low./Fd, 'low');
% создаем второй процесс:
y2 = filter(b, a, x2);
% *****
% изменяем интенсивность первого процесса, используя второй:
y12 = y1 .* y2;
% *****
% нормировка по амплитуде:
Am = 8192; % константа, задающая максимум громкости
% (Am не должна превышать 32767)
maxy12 = max(y12(:,:)); miny12 = min(y12(:,:));
norm1 = abs(miny12(1)); norm2 = abs(miny12(2));
if (maxy12(1)>abs(miny12(1))) norm1 = maxy12(1); end
if (maxy12(2)>abs(miny12(2))) norm2 = maxy12(2); end
y(:,1) = Am.*(y12(:,1) ./ norm1);
y(:,2) = Am.*(y12(:,2) ./ norm2);
output_signal = int16(y);
% записываем новый аудиофайл:
audiowrite('output_audio.wav', output_signal, Fd);
% построение графиков сигналов
start=1; stop=N;
figure(1)
subplot(2,1,1); plot(y(start:stop,1));
subplot(2,1,2); plot(y(start:stop,2));
```

Эксперимент №1

Значения границ фильтрации:

$f1_low = 700$

$f1_up = 1300$

В результате работы программы был сохранен wav-файл, по звучанию напоминающий утрированный звук приближающейся полиции, доносящийся из окна. Подобное аудио хорошо бы вписалось в кинофильм.

На рисунке 1 можно рассмотреть график сигнала.

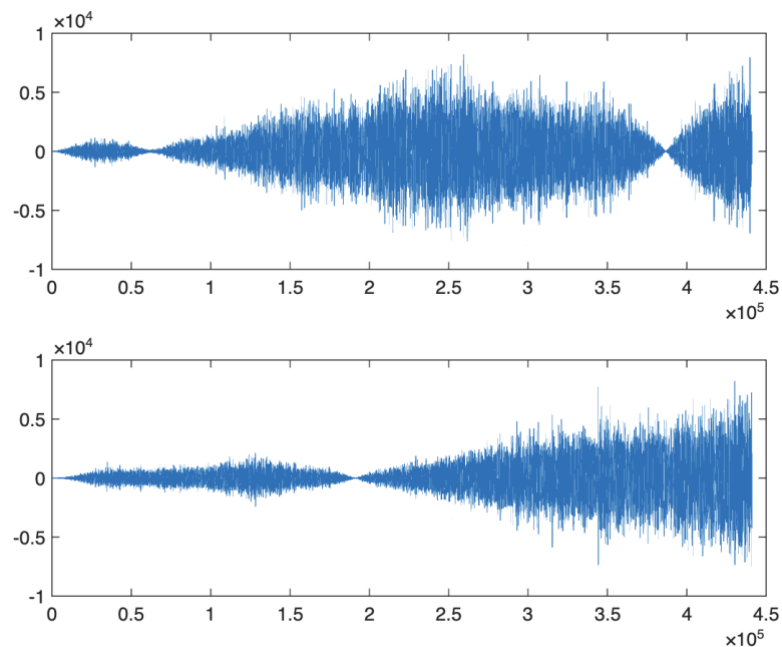


Рисунок 1 - График сигнала

Эксперимент №2

Значения границ фильтрации:

$f1_low = 2200$

$f1_up = 3000$

Фильтрация в этих частотах напоминает звук природы ночью, в особенности звук кузнечиков. Звук носит утрированный характер и на низкой громкости действительно может создаться впечатление ночи вокруг.

На рисунке 2 можно рассмотреть график сигнала.

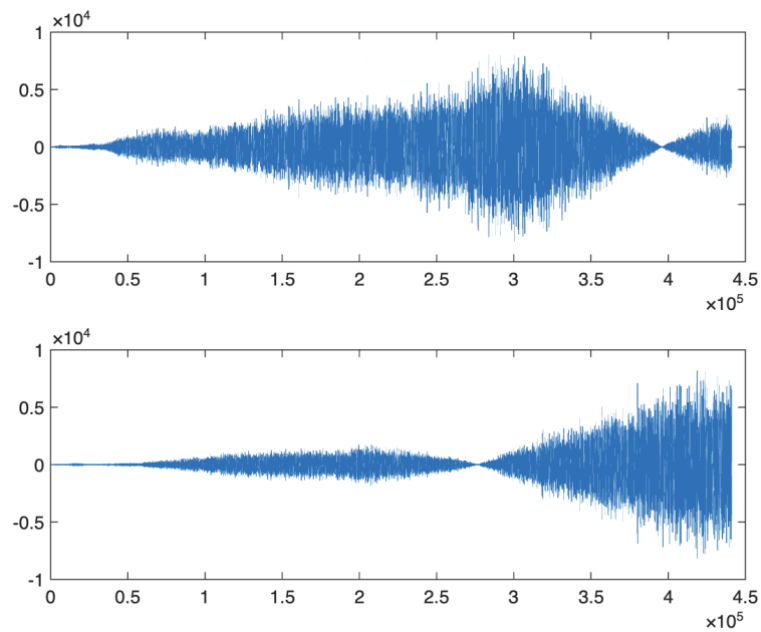


Рисунок 2 - График сигнала

Эксперимент №3

Значения границ фильтрации:

$f1_low = 100$

$f1_up = 345$

Сигнал, обработанный с данными параметрами, напоминает звук, записанный на небольшой глубине в воде.

На рисунке 3 можно рассмотреть график сигнала.

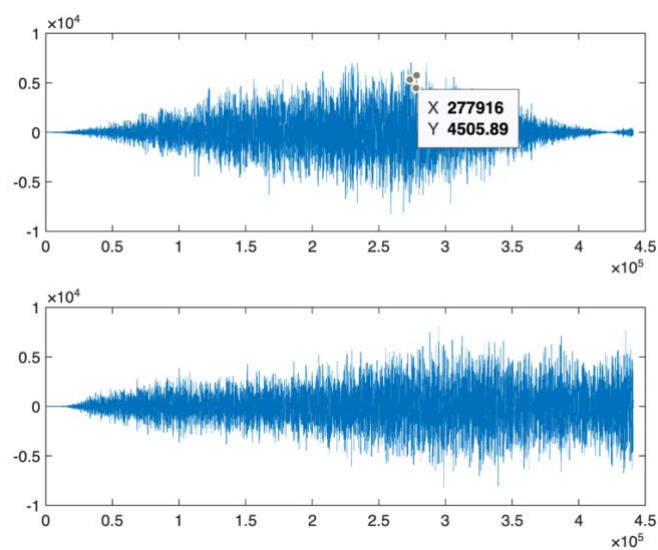


Рисунок 3 - График сигнала

Далее был исследован метод детерминированного закона. Код можно рассмотреть на листинге 2.

Листинг 2. Программный код, применяющий детерминированный закон изменения

```
clc, clear, close all
Fd = 44100; % задаем стандартную частоту дискретизации в Гц
Td = 1./Fd; % период дискретизации – обратный частоте
Ts = 10; % задаем длительность формируемого сигнала в секундах
N = Fd*Ts; % количество дискретных отсчетов
x = rand(N,2)*2 - 0.5; % создаем источник белого шума
eps1 = 0.01; % параметр, отвечающий за полосу каждого форм. фильтра
dn1 = 1-eps1; up1 = 1+eps1;
f1 = 880; % центральная частота первого ФФ
% применяем первый фф (в данном случае - полосовой):
y1 = bandpass(x, [dn1*f1, up1*f1], Fd);
% применяем второй ФФ:
eps2 = 0.1; % меняем полосу форм. фильтра
dn2 = 1-eps2; up2 = 1+eps2;
f2 = 7*f1; % меняем частоту настройки
% применяем три полосовых фильтра:
y2 = bandpass(x, [dn2*f2, up2*f2], Fd);
y12 = zeros(N,2);
% смешиваем процессы, меняя их интенсивности во времени:
% *****
for k=1:N
    % для смешивания применим сигмоидную функцию,
    % она меняется от почти 0 в начале до почти 1 в конце:
    alfa = 1/(1 + exp(-10*(2*(k-N/2)/N)));
    % собственно смешивание с переменными коэфф-ми:
    y12(k,:) = y1(k,:) * (1- alfa) + y2(k,:) * alfa;
    % и дополнительно периодическое изменение громкости:
    y12(k,:) = y12(k,:) * (0.5*(1-cos(2*pi*k*2/Fd)))^2;
end
% *****
% нормировка по амплитуде:
Am = 8192; % константа, задающая максимум громкости
% (Am не должна превышать 32767)
maxy12 = max(y12(:,:)); miny12 = min(y12(:,:));
norm1 = abs(miny12(1)); norm2 = abs(miny12(2));
if (maxy12(1)>abs(miny12(1))) norm1 = maxy12(1); end
if (maxy12(2)>abs(miny12(2))) norm2 = maxy12(2); end
y(:,1) = Am.*(y12(:,1) ./ norm1);
y(:,2) = Am.*(y12(:,2) ./ norm2);
output_signal = int16(y);
% записываем новый аудиофайл:
audiowrite('D:/output_audio.wav', output_signal, Fd);
```

По аналогии с первым законом изменения интенсивности были проведены эксперименты.

Эксперимент №1

Изменяемые параметры:

$f1 = 880$

$\text{eps2} = 0.1$

Звук носит шипящий и звенящий характер.

Эксперимент №2

Изменяемые параметры:

$f1 = 320$

$\text{eps2} = 0.4$

Звук практически никак не изменился, однако стал более глухим.

Эксперимент №3

Изменяемые параметры:

$f1 = 10000$

$\text{eps2} = 0.4$

Звук так же остается шипящим и звенящим, но появился некий «песок» в сигнале.

Заключительным этапом лабораторной работы является написание собственной программы по формированию шумоподобного сигнала.

Листинг 3. Программа, разработанная для формирования шумоподобного сигнала

```
% Параметры
Fs = 44100; % Частота дискретизации
T = 10; % Длительность сигнала в секундах
t = 0:1/Fs:T-1/Fs; % Временной вектор

% Генерация гауссовского белого шума
noise = randn(size(t));

% Добавление синусоидальных осцилляций
f1 = 1000; % Частота первого осциллятора
f2 = 5000; % Частота второго осциллятора
```

```

oscillation1 = sin(2*pi*f1*t);
oscillation2 = sin(2*pi*f2*t);

% Добавление фильтров
% Фильтр нижних частот
lpf_cutoff = 1000;
lpf_order = 4; % Порядок фильтра верхних частот
lpf = designfilt('lowpassfir', 'FilterOrder', lpf_order, 'CutoffFrequency', lpf_cutoff, 'SampleRate',
Fs);

% Фильтр верхних частот
hpf_cutoff = 2000; % Частота среза фильтра верхних частот в Гц
hpf_order = 4; % Порядок фильтра верхних частот
hpf = designfilt('highpassfir', 'FilterOrder', hpf_order, 'CutoffFrequency', hpf_cutoff,
'SampleRate', Fs);

% применение фильтров
filtered_noise = filtfilt(lpf, noise);
filtered_noise = filtfilt(hpf, filtered_noise);

% Объединение звуков
signal = filtered_noise + 0.5*oscillation1 + 0.2*oscillation2;

% Нормализация сигнала
signal = signal / max(abs(signal));

% Сохранение
filename = 'сигнал.wav';
audiowrite(filename, signal, Fs);

```

Результатом данной программы стал сигнал, не имеющий никаких отличий от звука во время полета самолета в кабине.

Такой аудиосигнал получилось сформировать в результате «сложения» синусоид и белого шума.

Заключение

В ходе лабораторной работы были изучены и применены методы формирования шумоподобного сигнала. Проведены эксперименты с методом динамического изменения интенсивности звука. Таким же экспериментальным образом был применен метод, использующий детерминированный закон изменения интенсивности звука.

Завершающим этапом стала попытка формирования собственного шумоподобного сигнала, что потребовало не мало времени. В разработке данного аудиосигнала были изучены методы работы осцилляторов.

Ссылка на получившиеся аудиофайлы

<https://github.com/cestagape/signal-processing-labs/tree/main/4>

Список использованных источников

1. Жаринов О.О. Учебно-методические материалы к выполнению лабораторной работы №4 по дисциплине “Мультимедиа-технологии”, гр.4016,4017. ГУАП, 2024 (Дата обращения: 21.03).