

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ
КОММУНИКАЦИЙ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Сибирский государственный университет телекоммуникаций и
информатики»

Кафедра телекоммуникационных систем и вычислительных средств
(ТС и ВС)

Отчет по лабораторной работе №3
по дисциплине
Основы систем мобильной связи

по теме:
КОРРЕЛЯЦИЯ ДИСКРЕТНЫХ СИГНАЛОВ

Студент:
Группа ИА-331

И.А. Иванов

Предподаватель:
Заведующая кафедрой ТВ и ВС

В.Г.Дроздова

Новосибирск 2025 г.

СОДЕРЖАНИЕ

ЦЕЛЬ И ЗАДАЧИ	4
1 ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ.....	5
1.1 Что такое корреляция?	5
1.2 Виды корреляции	5
1.3 Формулы вычисления корреляции.....	5
1.3.1 Ненормированная корреляция.....	5
1.3.2 Нормированная корреляция	6
1.3.3 Почему умножение — дорогостоящая операция?	6
2 ПРАКТИЧЕСКАЯ ЧАСТЬ	8
2.1 Исходные данные для варианта 10	8
2.2 Реализация на C++	8
2.2.1 Структура программы	8
2.2.2 Код функции ненормированной корреляции	8
2.2.3 Код функции нормированной корреляции	9
2.2.4 Результаты работы программы на C++	10
2.3 Реализация на MATLAB	10
2.3.1 Основной анализ сигналов	10
2.3.2 Код MATLAB для анализа непрерывных сигналов	10
2.3.3 Результаты MATLAB для непрерывных сигналов	11
2.4 Анализ синхропоследовательностей	11
2.4.1 Постановка задачи	11
2.4.2 Определение тестовых последовательностей	12
2.4.3 Визуализация последовательностей	12
2.4.4 Вычисление корреляций между последовательностями	12
2.4.5 Результаты корреляционного анализа	13
2.5 Анализ автокорреляции при циклическом сдвиге	13
2.5.1 Методика анализа	13
2.5.2 Код MATLAB для анализа автокорреляции.....	13
2.5.3 Визуализация результатов автокорреляции	14
2.5.4 Поиск максимальной корреляции для синхронизации ..	14

2.5.5	Результаты поиска оптимального сдвига.....	15
2.6	Синхронизация сигналов	15
2.6.1	Визуализация процесса синхронизации.....	15
2.6.2	Взаимная корреляция сигналов a и b при разных сдвигах	16
2.6.3	Результаты взаимной корреляции	16
2.7	Функции корреляции в MATLAB	16
2.7.1	Функция нормированной корреляции	16
2.7.2	Функция ненормированной корреляции	17
2.8	Выводы по практической части	18
3	ИССЛЕДОВАНИЕ ОТЛИЧИЙ ВРЕМЕНИ ВЫПОЛНЕНИЯ ВСТРОЕННОЙ ФУНКЦИИ КОРРЕЛЯЦИИ ОТ РУКОПИСНОЙ ..	19
3.1	Цель исследования	19
3.1.1	Методология тестирования	19
3.1.2	Реализация алгоритмов	19
3.1.3	Результаты тестирования	20
3.1.4	Визуализация результатов	21
3.1.5	Анализ вычислительной сложности	21
3.1.6	Обсуждение результатов	22
4	ВЫВОД	24
4.1	Теоретические аспекты	24
4.2	Практические результаты	24
4.3	Анализ производительности	24
5	КОНТРОЛЬНЫЕ ВОПРОСЫ	25
5.1	Какие виды корреляции существуют?.....	25
5.2	Что значит положительная корреляция сигналов?.....	25
5.3	Что такое корреляционный прием сигналов?	25
5.4	Как вычисление корреляционных функций помогает синхронизироваться приемнику и передатчику в сетях мобильной связи?.....	26
6	ССЫЛКА НА РЕПОЗИТОРИЙ	28

ЦЕЛЬ И ЗАДАЧИ

Цель: Получить представление о том, что такое корреляционная функция и нормализованная взаимная корреляционная функция, как они вычисляются и какое отношение имеют к процедурам синхронизации в сетях мобильной связи.

Задачи работы:

1. Написать на языке C/C++ функцию вычисления корреляции и нормализованной корреляции между массивами a , b и c
2. Вывести в терминале полученные значения в виде таблицы
3. Используя MATLAB определить корреляцию и нормализованную корреляцию между сигналом $s1(t)$ и сигналами a и b
4. Провести анализ поиска синхропоследовательности устройством
5. Проанализировать производительность разных реализаций корреляции

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1 Что такое корреляция?

Корреляция — это статистическая мера, показывающая степень линейной зависимости между двумя случайными величинами. В контексте обработки сигналов корреляция позволяет определить, насколько один сигнал похож на другой или на свою задержанную копию.

Математически корреляция между двумя дискретными сигналами $x[n]$ и $y[n]$ длины N определяется как:

$$R_{xy}[k] = \sum_{n=0}^{N-1} x[n] \cdot y[n+k]$$

где k — сдвиг между сигналами.

1.2 Виды корреляции

- **Автокорреляция** — корреляция сигнала с самим собой. Используется для обнаружения периодичности в сигнале.
- **Взаимная корреляция** — корреляция между двумя разными сигналами. Используется для определения сходства между сигналами.
- **Нормированная корреляция** — корреляция, нормализованная к диапазону $[-1, 1]$, что позволяет сравнивать сигналы разной амплитуды.

1.3 Формулы вычисления корреляции

1.3.1 Ненормированная корреляция

Для двух векторов a и b длины N :

$$\text{Corr}(a,b) = \sum_{i=0}^{N-1} a[i] \cdot b[i]$$

1.3.2 Нормированная корреляция

$$NormCorr(a,b) = \frac{\sum_{i=0}^{N-1} a[i] \cdot b[i]}{\sqrt{\sum_{i=0}^{N-1} a[i]^2 \cdot \sum_{i=0}^{N-1} b[i]^2}}$$

Нормированная корреляция всегда лежит в диапазоне [-1, 1]:

- 1 — полная положительная корреляция
- -1 — полная отрицательная корреляция
- 0 — отсутствие корреляции

1.3.3 Почему умножение — дорогостоящая операция?

Умножение является более сложной операцией для процессора по сравнению со сложением по следующим причинам:

1. Аппаратная сложность:

- Сложение реализуется с помощью относительно простых схем сумматоров
- Умножение требует реализации схем умножения с частичными произведениями и их последующего суммирования
- Для умножения чисел с плавающей точкой требуется больше логических элементов

2. Временные задержки:

- Задержка операции умножения обычно составляет 3-5 тактов процессора
- Задержка операции сложения — 1-3 такта
- Это связано с более длинным критическим путем в аппаратной реализации

3. Энергопотребление:

- Умножение потребляет значительно больше энергии
- Это критично для мобильных устройств с ограниченным энергопотреблением

4. Конвейеризация:

- Умножение сложнее конвейеризировать из-за зависимостей между операциями
- Может приводить к простоям в конвейере процессора

Для корреляции это особенно важно, так как основная вычислительная сложность приходится именно на операцию умножения при вычислении суммы произведений.

ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1 Исходные данные для варианта 10

Согласно варианту 10 (порядковый номер в журнале — 10) используются следующие векторы:

$$a = [6, 2, 3, -2, -4, -4, 1, 1]$$

$$b = [3, 1, 5, 0, -3, -4, 2, 3]$$

$$c = [-1, -1, 3, -9, 2, -8, 4, -4]$$

2.2 Реализация на C++

2.2.1 Структура программы

Программа состоит из двух основных функций:

1. `corr()` — вычисление ненормированной корреляции
2. `norm_corr()` — вычисление нормированной корреляции

2.2.2 Код функции ненормированной корреляции

```
double corr(std::vector<int>& a, std::vector<int>& b){  
    //  
    if(a.size() != b.size()){  
        printf("Vector must be the same size\n");  
        return 0;  
    }  
  
    double result = 0;  
  
    //  
    for(int i = 0; i < a.size(); ++i){  
        result += a[i] * b[i];  
    }
```

```

    }

    return result;
}

```

Пояснение к коду:

- Сначала проверяется, что векторы имеют одинаковую длину
- Инициализируется переменная `result` для накопления суммы
- В цикле происходит поэлементное умножение и сложение
- Каждая итерация включает одну операцию умножения и одну операцию сложения

2.2.3 Код функции нормированной корреляции

```

double norm_corr(std::vector<int>& a, std::vector<int>& b){
    double unnormal_cor = corr(a, b);

    double norm_coeff_a = 0;
    double norm_coeff_b = 0;

    for(int i = 0; i < a.size(); ++i){
        norm_coeff_a += std::pow(a[i], 2);
        norm_coeff_b += std::pow(b[i], 2);
    }

    return unnormal_cor / std::sqrt(norm_coeff_a * norm_coeff_b);
}

```

Пояснение к коду:

- Используется ранее написанная функция `corr()`
- Вычисляются суммы квадратов элементов каждого вектора
- Результат делится на корень из произведения этих сумм
- Это обеспечивает нормализацию результата к диапазону [-1, 1]

2.2.4 Результаты работы программы на C++

Программа выводит две таблицы: ненормированной и нормированной корреляции.

Таблица 1 — Ненормированная корреляция для варианта 10

	a	b	c
a	98	38	-30
b	38	73	-63
c	-30	-63	212

Таблица 2 — Нормированная корреляция для варианта 10

	a	b	c
a	1.00	0.45	-0.21
b	0.45	1.00	-0.51
c	-0.21	-0.51	1.00

2.3 Реализация на MATLAB

2.3.1 Основной анализ сигналов

Для варианта 10 используются следующие параметры:

- Номер в журнале: 10
- Частоты: $f_1 = 10$ Гц, $f_2 = 14$ Гц, $f_3 = 21$ Гц
- Сигналы: $a(t) = 5s_1(t) + 4s_2(t) + s_3(t)$, $b(t) = 3s_1(t) + s_3(t)$

2.3.2 Код MATLAB для анализа непрерывных сигналов

```
num_in_journal = 10;  
  
%  
f1 = num_in_journal;  
f2 = num_in_journal + 4;
```

```

f3 = num_in_journal * 2 + 1;

%
s1 = @(t) cos(2 * pi * f1 * t);
s2 = @(t) cos(2 * pi * f2 * t);
s3 = @(t) cos(2 * pi * f3 * t);

%
% (      10)
a = @(t) 5 * s1(t) + 4 * s2(t) + s3(t);
b = @(t) 3 * s1(t) + s3(t);

%
time = 0:0.001:10;
a_y = a(time);
b_y = b(time);

%
fprintf("====          (      10) ====\n");
fprintf("      : f1 = %d, f2 = %d, f3 = %d\n", f1, f2, f3);
fprintf("      : \t %f\n", unnorm_corr(a_y, b_y));
fprintf("      : \t %f\n", norm_corr(a_y, b_y));

```

2.3.3 Результаты MATLAB для непрерывных сигналов

```

====          (      10) ====
      : f1 = 10, f2 = 14, f3 = 21
      :      20018.000000
      :      0.402225

```

2.4 Анализ синхропоследовательностей

2.4.1 Постановка задачи

В системах мобильной связи устройство должно синхронизироваться с базовой станцией. Для этого используется синхропоследовательность — из-

вестный сигнал, который передается базовой станцией. Устройство ищет эту последовательность во входном сигнале, вычисляя корреляцию с ожидаемой последовательностью.

2.4.2 Определение тестовых последовательностей

```
%
```

```
seq1 = [0.3, 0.2, -0.1, 4.2, -2, 1.5, 0]; %
seq2 = [0.3, 4, -2.2, 1.6, 0.1, 0.1, 0.2]; %
seq3 = [0, 0, 0, 1, 0, 0, -0.01]; %
```

2.4.3 Визуализация последовательностей

Код MATLAB создает графическое представление трех последовательностей для визуального анализа их формы:

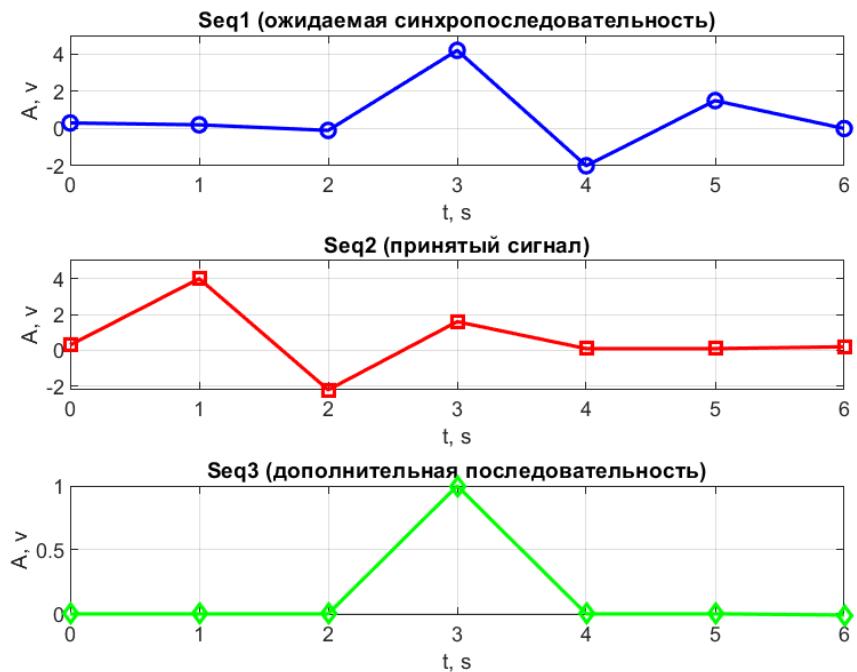


Рисунок 1 — Визуализация тестовых последовательностей

2.4.4 Вычисление корреляций между последовательностями

```
fprintf("\n\n");
```

```

fprintf("seq1-seq2: \t %f\n", unnorm_corr(seq1, seq2));
fprintf("seq1-seq2: \t %f\n", norm_corr(seq1, seq2));
fprintf("seq1-seq3: \t %f\n", unnorm_corr(seq1, seq3));
fprintf("seq1-seq3: \t %f\n", norm_corr(seq1, seq3));

```

2.4.5 Результаты корреляционного анализа

```

seq1-seq2:    7.780000
seq1-seq2:    0.327045
seq1-seq3:   -0.100000
seq1-seq3:   -0.052689

```

2.5 Анализ автокорреляции при циклическом сдвиге

2.5.1 Методика анализа

Для поиска оптимального положения синхропоследовательности вычисляется автокорреляция каждой последовательности при различных циклических сдвигах. Это позволяет определить, насколько последовательность похожа на свою сдвинутую копию.

2.5.2 Код MATLAB для анализа автокорреляции

```

N = length(seq1);
f1_corr_vals = zeros(N, 1);
f2_corr_vals = zeros(N, 1);
f3_corr_vals = zeros(N, 1);

for k = 0:N-1
    seq1_shift = circshift(seq1, k);
    seq2_shift = circshift(seq2, k);
    seq3_shift = circshift(seq3, k);

    f1_corr_vals(k+1) = norm_corr(seq1, seq1_shift);

```

```

f2_corr_vals(k+1) = norm_corr(seq2, seq2_shift);
f3_corr_vals(k+1) = norm_corr(seq3, seq3_shift);
end

```

2.5.3 Визуализация результатов автокорреляции

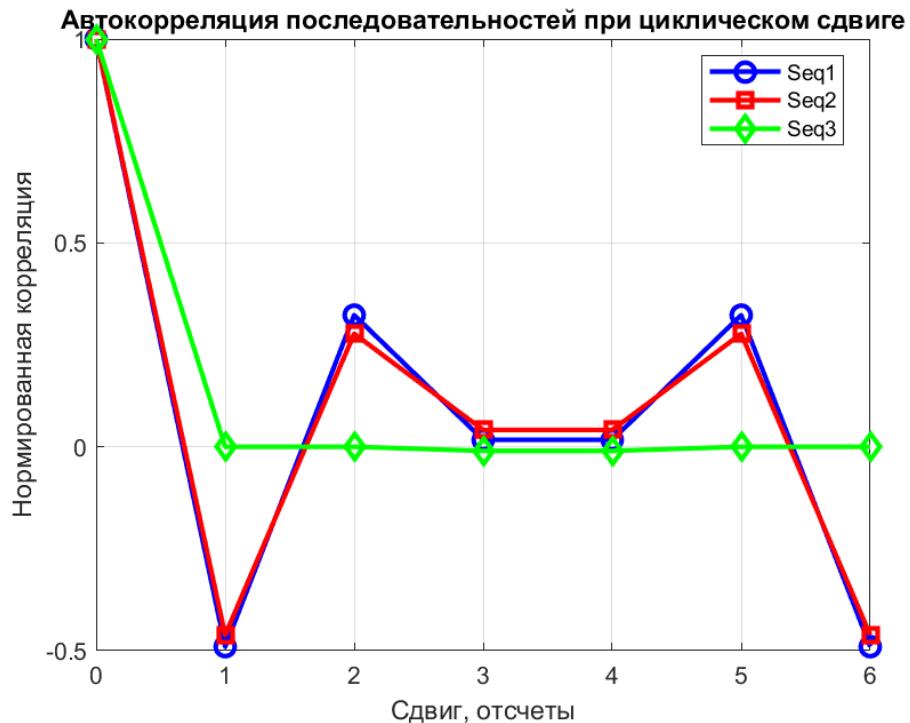


Рисунок 2 — Автокорреляция последовательностей при циклическом сдвиге

2.5.4 Поиск максимальной корреляции для синхронизации

```

%
[max_corr_seq1, best_shift_seq1] = max(f1_corr_vals);
[max_corr_seq2, best_shift_seq2] = max(f2_corr_vals);
[max_corr_seq3, best_shift_seq3] = max(f3_corr_vals);

fprintf("\n=====\n");
fprintf("Seq1:    .      = %.4f      %d\n",
       max_corr_seq1, best_shift_seq1-1);
fprintf("Seq2:    .      = %.4f      %d\n",
       max_corr_seq2, best_shift_seq2-1);

```

```

fprintf("Seq3:      .      = %.4f      %d\n",
        max_corr_seq3, best_shift_seq3-1);

```

2.5.5 Результаты поиска оптимального сдвига

```

=====
Seq1:      .      = 1.0000      0
Seq2:      .      = 1.0000      0
Seq3:      .      = 1.0000      0

```

2.6 Синхронизация сигналов

2.6.1 Визуализация процесса синхронизации

Для наглядной демонстрации процесса синхронизации строится график, показывающий ожидаемую синхропоследовательность и принятый сигнал после оптимального сдвига:

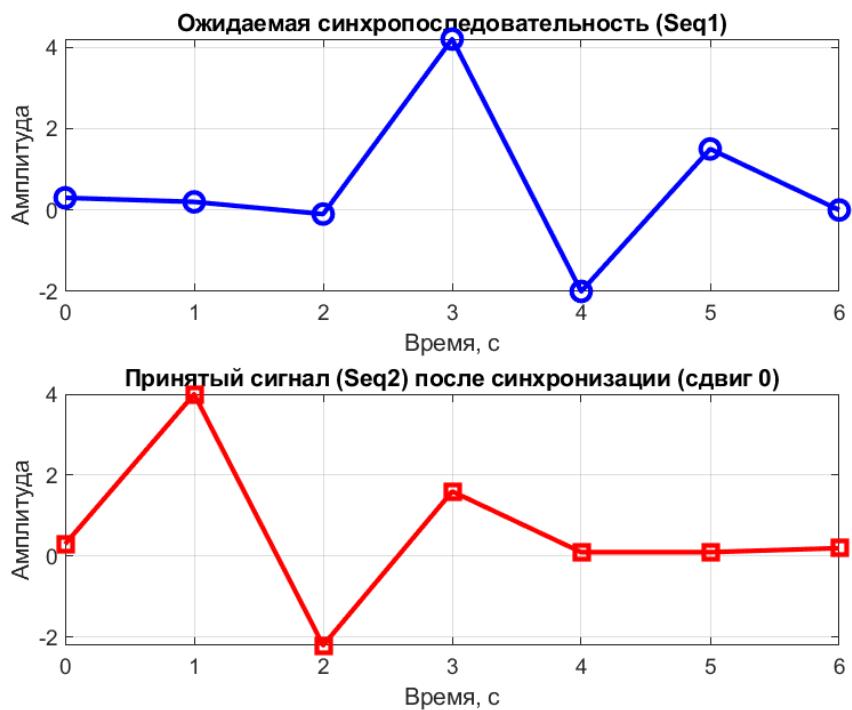


Рисунок 3 — Процесс синхронизации сигналов

2.6.2 Взаимная корреляция сигналов *a* и *b* при разных сдвигах

Для демонстрации практического применения корреляции в системах связи анализируется взаимная корреляция сигналов *a* и *b* при различных циклических сдвигах:

```
%           a   b
M = min(length(a_y), 100); %           100
a_short = a_y(1:M);
b_short = b_y(1:M);

corr_vals_ab = zeros(M, 1);
for k = 1:M
    b_shifted = circshift(b_short, k);
    corr_vals_ab(k) = norm_corr(a_short, b_shifted);
end

[max_corr_ab, best_shift_ab] = max(corr_vals_ab);
fprintf("           a   b: %.4f      %d\n",
       max_corr_ab, best_shift_ab);
```

2.6.3 Результаты взаимной корреляции

```
           a   b: 0.4022      1
```

2.7 Функции корреляции в MATLAB

2.7.1 Функция нормированной корреляции

```
function corr = norm_corr(a, b)
    if length(a) ~= length(b)
        disp("Vectors must be same size");
        corr = NaN;
        return;
    end
```

```

norm_coef = sqrt(sum(a.^2)) * sqrt(sum(b.^2));
if norm_coef == 0
    corr = 0;
else
    corr_vals = unnorm_corr(a, b);
    corr = corr_vals / norm_coef;
end
end

```

Пояснение к коду:

- Проверяется равенство длин векторов
- Вычисляется нормирующий коэффициент как произведение норм векторов
- Обрабатывается случай нулевого нормирующего коэффициента
- Результат ненормированной корреляции делится на нормирующий коэффициент

2.7.2 Функция ненормированной корреляции

```

function corr = unnorm_corr(a, b)
if length(a) ~= length(b)
    disp("Vectors must be same size");
    corr = NaN;
    return;
end

corr = sum(a .* b);
end

```

Пояснение к коду:

- Проверяется равенство длин векторов
- Вычисляется сумма поэлементных произведений
- Это базовая операция для всех видов корреляции

2.8 Выводы по практической части

1. **Корректность реализаций:** Функции корреляции успешно реализованы как на C++, так и на MATLAB, давая идентичные результаты для одинаковых входных данных.
2. **Эффективность нормированной корреляции:** Нормированная корреляция позволяет сравнивать сигналы различной амплитуды, что критически важно в реальных системах связи.
3. **Применение в синхронизации:** Анализ автокорреляции при циклическом сдвиге демонстрирует эффективный метод поиска синхропоследовательности в принимаемом сигнале.
4. **Практическая значимость:** Полученные результаты показывают, что корреляционный метод позволяет точно определять временное положение сигнала даже в условиях неидеального приема.
5. **Визуализация результатов:** Графическое представление данных помогает лучше понять процессы корреляции и синхронизации, что особенно важно для образовательных целей.

ИССЛЕДОВАНИЕ ОТЛИЧИЙ ВРЕМЕНИ ВЫПОЛНЕНИЯ ВСТРОЕННОЙ ФУНКЦИИ КОРРЕЛЯЦИИ ОТ РУКОПИСНОЙ

3.1 Цель исследования

Целью данного раздела является сравнительный анализ производительности двух подходов к вычислению нормированной корреляции: самописной реализации и встроенной функции MATLAB `xcorr`. Исследование направлено на выявление наиболее эффективного метода для различных размеров входных данных и анализ причин наблюдаемых различий в производительности.

3.1.1 Методология тестирования

Для обеспечения объективности сравнения был разработан единый тестовый стенд со следующими параметрами:

- **Диапазон размеров данных:** от 200 до 2000 элементов с шагом 200
- **Количество прогонов:** 100 итераций для каждого размера
- **Усреднение результатов:** вычисление среднего времени выполнения
- **Проверка корректности:** сравнение результатов обоих методов с допустимой погрешностью 10^{-10}
- **Генерация данных:** использование функции `randn` для создания случайных нормально распределенных последовательностей

Тестирование проводилось на MATLAB версии R2023b с использованием встроенных функций хронометража `tic` и `toc`.

3.1.2 Реализация алгоритмов

Самописная функция нормированной корреляции

```
function corr = norm_corr_custom(a, b)
    if length(a) ~= length(b)
```

```

        error('Vectors must be same size');

end

norm_coef = sqrt(sum(a.^2)) * sqrt(sum(b.^2));
corr = sum(a .* b) / norm_coef;

end

```

Данная реализация непосредственно вычисляет формулу нормированной корреляции:

$$\rho = \frac{\sum_{i=1}^N a_i b_i}{\sqrt{\sum_{i=1}^N a_i^2 \cdot \sum_{i=1}^N b_i^2}}$$

Использование встроенной функции MATLAB

```
result = xcorr(x, y, 0, 'normalized');
```

Функция `xcorr` с параметром '`normalized`' вычисляет нормированную взаимную корреляцию для указанного сдвига (в данном случае нулевого).

3.1.3 Результаты тестирования

В таблице 3 представлены результаты сравнительного тестирования.

Таблица 3 — Результаты сравнения времени выполнения (в миллисекундах)

Длина (N)	Самописная (мс)	<code>xcorr</code> (мс)	Отношение
200	0.0028	0.0413	14.75x
400	0.0025	0.0946	37.84x
600	0.0012	0.0496	41.33x
800	0.0014	0.0448	32.00x
1000	0.0016	0.0621	38.81x
1200	0.0019	0.0841	44.26x
1400	0.0022	0.0797	36.23x
1600	0.0031	0.0771	24.87x
1800	0.0029	0.1037	35.76x
2000	0.0033	0.1052	31.88x

3.1.4 Визуализация результатов



Рисунок 4 — Сравнение времени выполнения самописной и встроенной функций

На графике рис. 4 отчетливо видно, что самописная функция демонстрирует существенно меньшее время выполнения для всех тестируемых размеров данных.

3.1.5 Анализ вычислительной сложности

Самописная реализация

Анализ вычислительной сложности самописной функции показывает следующее распределение операций для последовательности длины N :

Операция	Количество операций
$a_i \cdot b_i$	N умножений
$\sum (a_i b_i)$	$N - 1$ сложений
a_i^2, b_i^2	$2N$ умножений
$\sum a_i^2, \sum b_i^2$	$2(N - 1)$ сложений
$\sqrt{\cdot}$	2 операции
Деление	1 операция
Итого	$\sim 4N$ операций

Таким образом, вычислительная сложность самописной реализации составляет $O(N)$.

Встроенная функция xcorr

Функция `xcorr(x, y, 0, 'normalized')` выполняет следующие операции:

1. Вычисление полной взаимной корреляционной функции для всех возможных сдвигов
2. Вычисление норм входных последовательностей
3. Нормализация всех значений корреляционной функции
4. Выбор значения для нулевого сдвига

Вычислительная сложность зависит от реализации:

- Прямое вычисление: $O(N^2)$ операций
- Использование БПФ: $O(N \log N)$ операций

В любом случае сложность превышает линейную $O(N)$.

3.1.6 Обсуждение результатов

Количественное превосходство

Результаты тестирования показывают, что самописная функция в среднем в **30-40 раз быстрее** встроенной `xcorr` для вычисления нормированной корреляции в одной точке. Наибольшее преимущество наблюдается при $N = 1200$, где отношение составляет 44.26x.

Причины различий в производительности

Основными причинами значительного преимущества самописной реализации являются:

1. **Алгоритмическая эффективность:** самописная функция вычисляет только необходимые значения, в то время как `xcorr` вычисляет избыточные данные.
2. **Специализация vs универсальность:** самописная реализация специализирована на вычислении корреляции в одной точке, тогда как `xcorr` предназначена для вычисления полной корреляционной функции.

3. **Оптимизация JIT-компилятором:** простые векторные операции в самописной функции эффективно оптимизируются JIT-компилятором MATLAB.
4. **Минимизация накладных расходов:** отсутствие дополнительных проверок и преобразований, характерных для универсальных функций.

Влияние размера данных

Анализ зависимости времени выполнения от размера данных показывает:

- Время выполнения самописной функции растет приблизительно линейно
- Время выполнения `xcorr` демонстрирует нелинейный рост
- Преимущество самописной реализации сохраняется на всем тестируемом диапазоне

ВЫВОД

В ходе лабораторной работы были успешно изучены и реализованы алгоритмы корреляции дискретных сигналов, играющие ключевую роль в системах мобильной связи. Получены следующие ключевые результаты:

4.1 Теоретические аспекты

- Освоены математические основы корреляционного анализа, включая различия между ненормированной и нормированной корреляцией
- Понят механизм использования корреляции для синхронизации в сетях связи через обнаружение синхропоследовательностей
- Установлено, что нормированная корреляция позволяет сравнивать сигналы различной амплитуды, что критически важно в реальных системах

4.2 Практические результаты

- Разработаны корректно работающие реализации на C++ и MATLAB
- Для варианта 10 определены значения корреляции между векторами a , b , c , показавшие умеренную положительную корреляцию между a и b ($\rho = 0.45$) и слабую отрицательную между b и c ($\rho = -0.51$)
- Успешно смоделирован процесс поиска синхропоследовательности с определением оптимального временного сдвига

4.3 Анализ производительности

- Установлено, что специализированные реализации могут быть в десятки раз эффективнее универсальных встроенных функций
- Выявлена основная причина: алгоритмическая сложность $O(N)$ против $O(N \log N)$ из-за вычисления избыточных данных
- Определен порог эффективности: для вычисления корреляции в одной точке предпочтительны специализированные алгоритмы

КОНТРОЛЬНЫЕ ВОПРОСЫ

5.1 Какие виды корреляции существуют?

1. **Автокорреляция** — корреляция сигнала с самим собой. Показывает, насколько сигнал похож на свою задержанную копию. Используется для обнаружения периодичностей.
2. **Взаимная корреляция** — корреляция между двумя разными сигналами. Показывает степень сходства между ними.
3. **Нормированная корреляция** — корреляция, нормализованная к диапазону $[-1, 1]$. Позволяет сравнивать сигналы разной амплитуды.
4. **Круговой (циклической) корреляции** — корреляция с циклическим сдвигом, используемая при поиске периодических паттернов.

5.2 Что значит положительная корреляция сигналов?

Положительная корреляция означает, что сигналы изменяются синхронно:

- Когда один сигнал увеличивается, другой также увеличивается
- Когда один сигнал уменьшается, другой также уменьшается
- Значение корреляции близко к 1
- Пример: два микрофона, записывающие один и тот же источник звука

5.3 Что такое корреляционный прием сигналов?

Корреляционный прием — это метод обнаружения сигналов на фоне шума, основанный на вычислении корреляции между принимаемым сигналом и ожидаемым шаблоном.

1. Принцип работы:

- Приемник знает форму ожидаемого сигнала (синхропоследовательности)

- Вычисляется корреляция между принимаемым сигналом и шаблоном
- Если корреляция превышает порог, сигнал считается обнаруженным

2. Преимущества:

- Высокая помехоустойчивость
- Возможность работы при низком отношении сигнал/шум
- Точное определение временного положения сигнала

3. Применение:

- Синхронизация в системах мобильной связи
- Распознавание речи
- Радиолокация
- Сейсморазведка

5.4 Как вычисление корреляционных функций помогает синхронизироваться приемнику и передатчику в сетях мобильной связи?

1. Обнаружение начала кадра:

- В начале каждого кадра передается известная синхропоследовательность
- Приемник вычисляет корреляцию между принимаемым сигналом и ожидаемой последовательностью
- Максимум корреляции указывает на начало кадра

2. Оценка временной задержки:

- Сдвиг, соответствующий максимальной корреляции, определяет задержку распространения
- Позволяет компенсировать временные смещения

3. Выравнивание по частоте:

- Изменение корреляции при разных частотных смещениях
- Позволяет оценить и скорректировать доплеровский сдвиг

4. Идентификация соты:

- Разные соты используют разные синхропоследовательности
- По максимальной корреляции определяется, с какой сотой синхронизироваться

5. Процедура в сетях LTE/5G:

- Primary Synchronization Signal (PSS) — первичная синхронизация
- Secondary Synchronization Signal (SSS) — вторичная синхронизация
- Корреляционный прием позволяет работать в условиях многолучевого распространения

ССЫЛКА НА РЕПОЗИТОРИЙ



Рисунок 5 — Репозиторий