Учебное пособие

Учебно-методический комплект

по дисциплине «Цифровая обработка сигналов»

Конспект лекции

«Корреляционный анализ»

МГТУ имени Н.Э. Баумана

1 Корреляция

Главными задачами корреляционного анализа являются выявление зависимости или сходства между какими-либо сигналами. Корреляционный анализ имеет широкое применение в различных областях: например, поиск изображения по образцу, поиск полезных сигналов «под шумами», обработка сигналов радаров и т.п.

Корреляция в простом виде – это поэлементное произведение двух сигналов. Если получим большое число – сигналы похожи, если маленькое – нет:

$$r_{12} = \sum_{n=0}^{N-1} x_1[n] \cdot x_2[n] \tag{1}$$

Однако, результат выражения (1) будет зависеть от количества отсчётов сигналов. Чтобы это учесть, результат нормируют на количество отсчётов N:

$$r_{12} = \frac{1}{N} \sum_{n=0}^{N-1} x_1[n] \cdot x_2[n] \tag{2}$$

Рассмотрим пример. Возьмём два меандра, сдвинутых по фазе на 180° и посчитаем их корреляцию по формуле (2):

Листинг 1 – Корреляция двух меандров, часть 1

```
clear;

fs = 100;
ts = 0:1/fs:5-1/fs;

x1 = (square(2*pi*ts)+1)/2;
x2 = (square(2*pi*ts+pi)+1)/2;

subplot(2,1,1);
plot(x1), grid on;
xlabel('Временные отсчёты, n'), ylabel('x1[n]');

subplot(2,1,2);
plot(x2), grid on;
xlabel('Временные отсчёты, n'), ylabel('x2[n]');

r12 = sum(x1.*x2)
```

В результате выполнения скрипта получим график, показанный на рисунке 1 и значение корреляции `r12 = 0`.

Получается, что сигналы похожи, а корреляция равна нулю. Значит, выражение (2) необходимо модифицировать и добавить временной сдвиг j одного сигнала относительно другого:

$$r_{12}[j] = \frac{1}{N} \sum_{n=0}^{N-1} x_1[n] \cdot x_2[n+j] \tag{3}$$

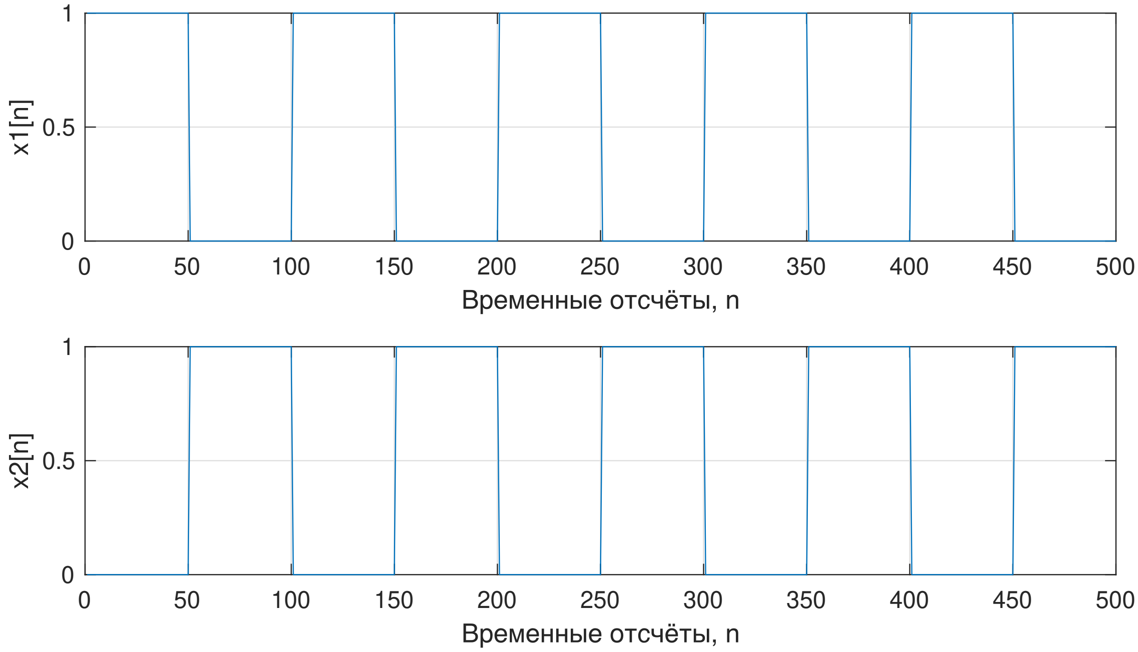

Рис. 1 – Два меандра, сдвинутых по фазе на 180°

Дополним листинг 1 и построим график корреляционной функции r_{12}. Для этого воспользуемся функцией Matlab взаимной корреляции двух сигналов `xcorr`:

Листинг 2 – Корреляция двух меандров, часть 2

```
figure;
[r12,lags] = xcorr(x1,x2);

plot(lags,r12), grid on;
xlabel('Временной сдвиг, j'), ylabel('r_{12}[j]');
```

Результат выполнения кода из листинга 2 показан на рисунке 2. Из графика видно, что при нулевом сдвиге j корреляция двух сигналов равна нулю $r_{12}[0] = 0$, при дальнейшем сдвиге влево или вправо значение r_{12} увеличивается, затем снова уменьшается. Это происходит периодически, однако амплитуда пиков r_{12} постепенно уменьшается к краям графика. Наверняка многие ожидали увидеть в качестве $r_{12}(j)$ периодический треугольный сигнал. Однако, из-за того, что сигналы `x1[n]` и `x2[n]` имеют конечную длину, получается, что при сдвиге одного сигнала относительно другого, наступает момент, когда они не перекрываются и вместо парных произведений отсчётов получаем произведения на «пустые отсчёты», или на ноль. Чем дальше два сигнала сдвигаются друг относительно друга, тем больше получается произведений с нулевым результатом, тем меньше получается результирующая r_{12}. Эта неприятность называется **краевой эффект**.

Один из способов убрать влияние краевого эффекта – увеличить длину одного из массивов (например, `x2[n]`) в два раза:

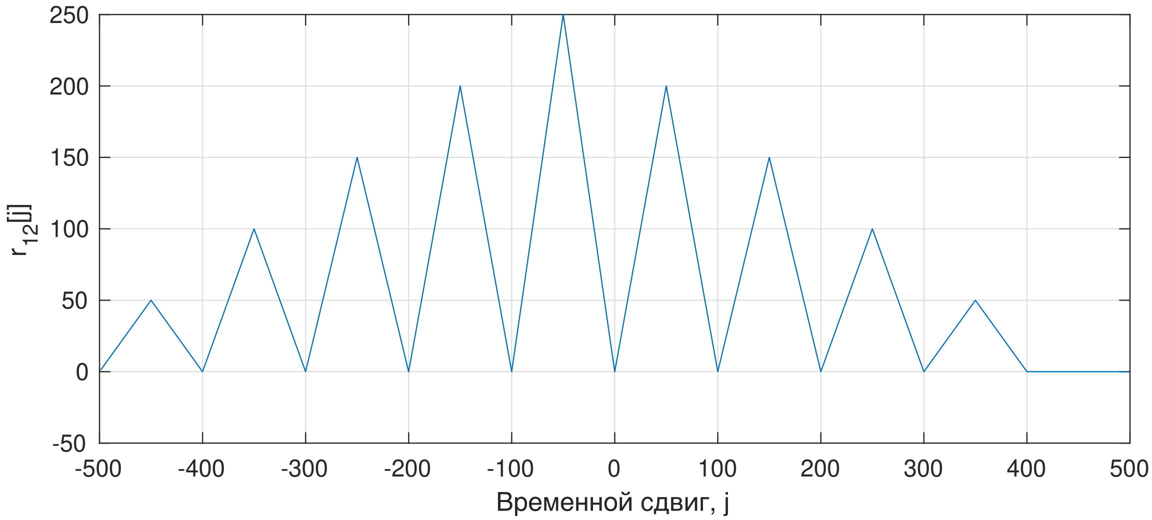

Рис. 2 – Корреляционная функция двух меандров

Листинг 3 – Корреляция двух меандров, часть 3

```
figure;
N = length(x1);
x2 = [x2 x2];        % добавляем в конец x2 его копию

r12=zeros(1,N);

% расчёт корреляционной функции:
for j=1:N-1
    r12(j) = sum(x1.*x2(j:j+N-1));
end

plot(r12), grid on;
xlabel('Временной сдвиг, j'), ylabel('r_{12}[j]');
```

Результат выполнения листинга 3 представлен на рисунке 3. Так гораздо лучше. А что, если амплитуду сигнала `x2[n]` увеличить в два раза? Отредактируем строку 7 листинга 1:

Листинг 4 – Корреляция двух меандров, изменение в строке 7

```
7 x2 = (2*square(2*pi*ts+pi)+2)/2;
```

Запустим заново скрипт и увидим, что амплитуда треугольного сигнала возросла вдвое (см. рисунок 4).

Так как же оценить степень схожести сигналов? Какое значение r_{12} есть хорошо, а какое – плохо? Чтобы вы не задавали таких вопросов, придумали нормированную корреляционную функцию, в которой результат может принимать значения только в диапазоне $[-1, 1]$. Значению 1 соответствует полная корреляция (сигналы

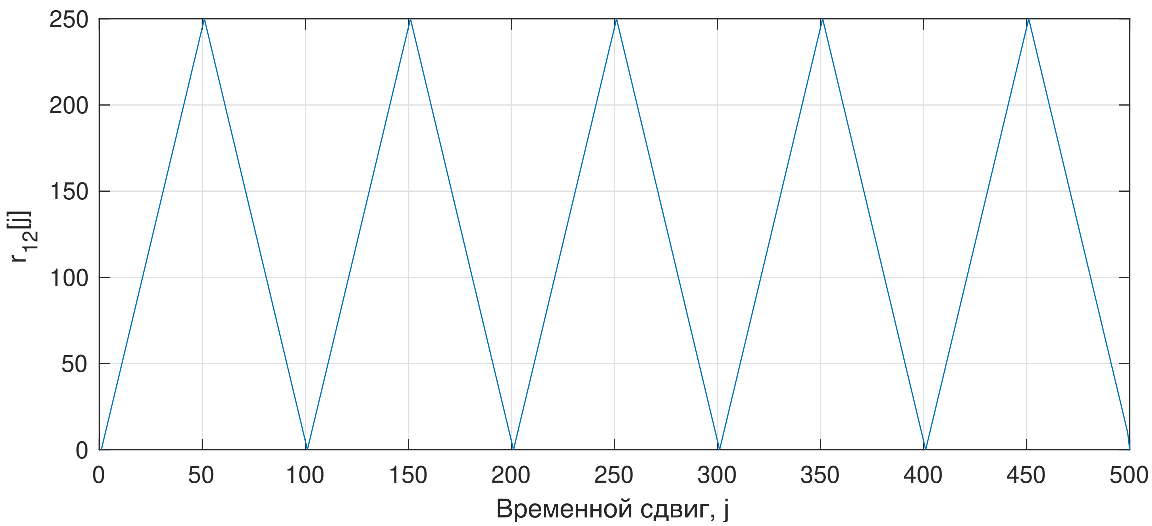

Рис. 3 – Корреляционная функция двух меандров

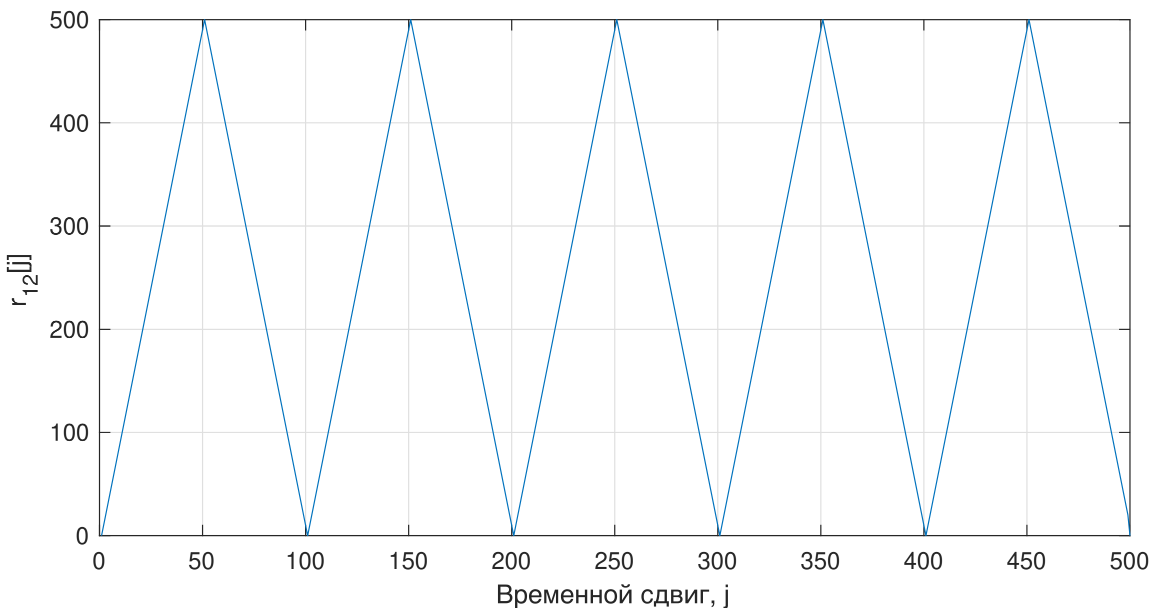

Рис. 4 – Корреляционная функция двух меандров (амплитуда x2[n] увеличена вдвое)

абсолютно похожи или зависимы), нулю соответствует полное отсутствие корреляции (сигналы абсолютно непохожи или независимы) и -1 соответствует полная противоположная корреляция (сигналы находятся в противофазе). Нормированная корреляционная функция имеет вид:

$$\rho_{12}[j] = \frac{r_{12}[j]}{\frac{1}{N}\sqrt{\sum\limits_{n=0}^{N-1} x_1^2[n] \cdot \sum\limits_{n=0}^{N-1} x_2^2[n]}} \tag{4}$$

По аналогичной формуле корреляцию будет считать Matlab, если к функции `xcorr` добавить ещё один параметр `'normalized'` или `'coeff'`.

2 Автокорреляция

А что, если в выражении (3) в качестве сигнала $x_2[n]$ также взять сигнал $x_1[n]$ и начать сравнивать его с самим собой? Получится **автокорреляционная функция**:

$$r_{11}[j] = \frac{1}{N}\sum_{n=0}^{N-1} x_1[n] \cdot x_1[n+j] \tag{5}$$

Некоторые зададут вопрос: "Что за бред? Зачем это нужно?". Во-первых, для определения, является ли исследуемый сигнал случайным, или же в нём есть периодическая составляющая. Если есть периодическая составляющая, можно более явно увидеть, какая у неё частота.

Рассмотрим пример. Создадим зашумлённый синусоидальный сигнал частотой 0.5 Гц и построим его график. Затем построим график автокорреляционной функции этого сигнала и взглянем на него.

Листинг 5 – Определение периодичности зашумлённого сигнала

```
clear;

Fs = 10;
ts = 0: 1/Fs : 10-1/Fs;
N = length(ts);

x = 0.1*sin(2*pi*0.5*ts);
x = awgn(x,20);

subplot(2,1,1)
plot(x), grid on, title('Исходный сигнал')
xlabel('Время')

[xc, lags] = xcorr(x,'unbiased');
subplot(2,1,2)
plot(lags/Fs,xc), grid on, title('Автокорреляционная функция')
xlabel('Временной сдвиг')
```

Результаты выполнения скрипта показаны на рисунке 5.

Из верхнего графика рисунка 5 сложно сделать вывод о периодичности сигнала, однако, если взглянуть на график его автокорреляционной функции (снизу), то все сомнения отпадают.

А как же будет выглядеть автокорреляционная функция (АКФ) для случайного сигнала? Рассмотрим на примере. Разработаем скрипт, генерирующий случайный сигнал амплитудой от -1 до 1, состоящий из 256 отсчётов. Затем построим график самого сигнала и его АКФ:

Листинг 6 – АКФ для случайного сигнала

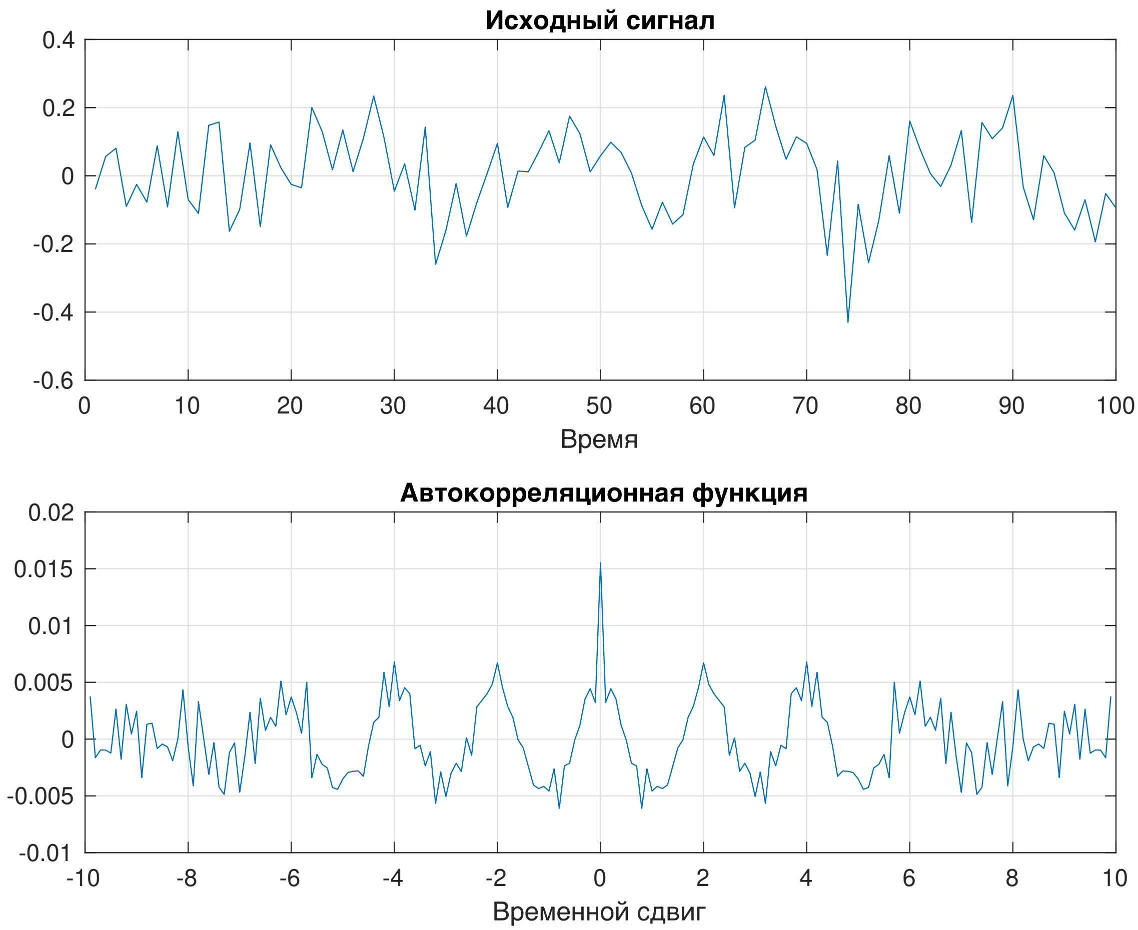

Рис. 5 – Результаты работы скрипта из листинга 5

```
clear;

% параметры случайного сигнала
N = 256;
a = -1;
b = 1;

% случайный сигнал
x = (a + (b - a) * rand(1, N));

% его АКФ
[c,lags] = xcorr(x,'coeff');

subplot(2,1,1);
```

```
plot(x), grid on, title('Случайный сигнал')
xlabel('Временные отсчёты, n'), ylabel('x[n]');

subplot(2,1,2);
plot(lags,c), grid on, title('АКФ случайного сигнала')
xlabel('Временной сдвиг, j'), ylabel('c[j]');
```

Результат выполнения скрипта из листинга 6 показан на рисунке 6. Из рисунка 6 видно, что для случайных сигналов график автокорреляционной функции имеет свой максимум при $j = 0$ и стремится к нулю с увеличением сдвига j.

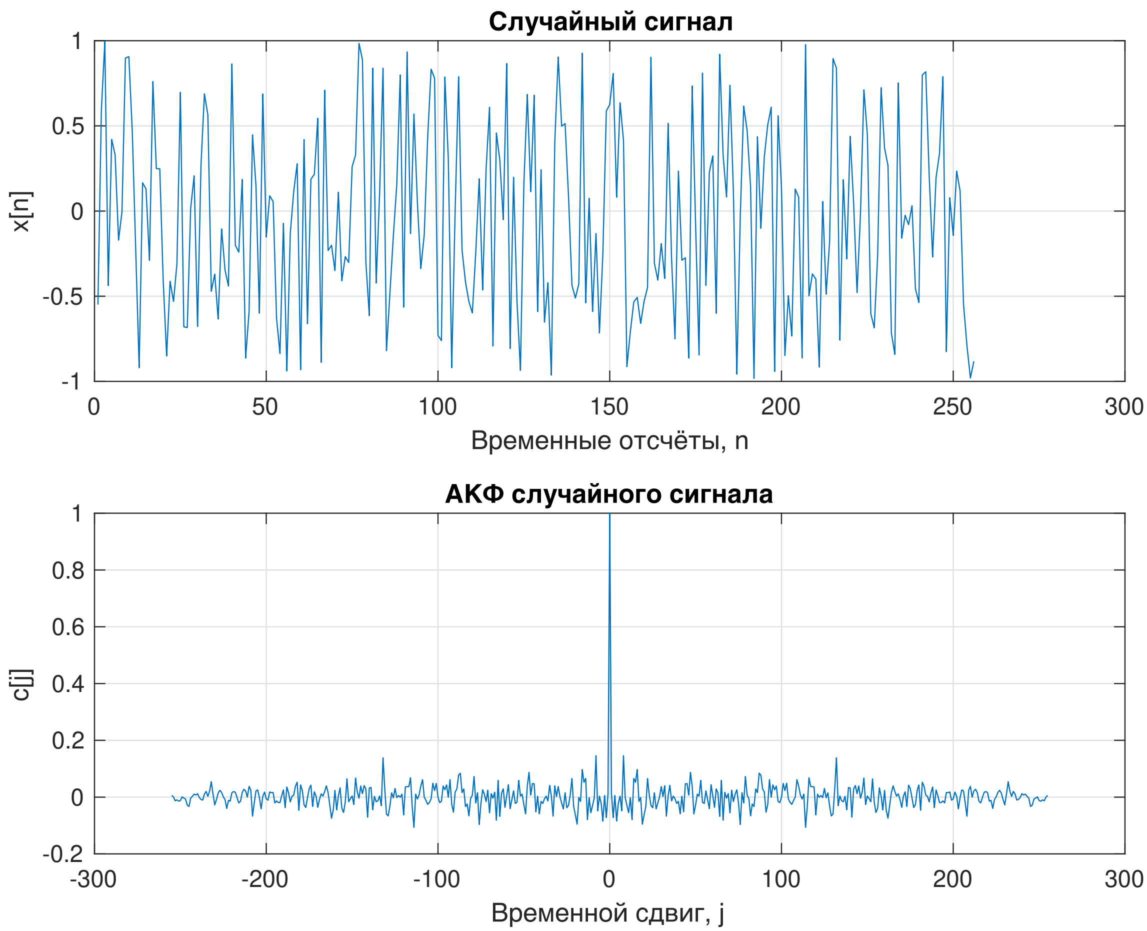

Рис. 6 – Автокорреляционная функция случайного сигнала

Также, стоит обратить внимание, что при $j = 0$ выражение (5) принимает вид:

$$r_{11}[0] = \frac{1}{N}\sum_{n=0}^{N-1} x_1^2[n] = S \qquad (6)$$

где S - **нормированная энергия сигнала**.

3 Корреляция изображений

Помимо одномерных сигналов, корреляционный анализ можно проводить и с двумерными сигналами. Например, с изображениями для поиска какого-либо шаблона в другом изображении. Корреляционная функция для изображений может иметь вид:

$$\gamma(u,v) = \frac{\sum\limits_x \sum\limits_y (f[x,y] - \overline{f}_{u,v})(t[x-u,y-v] - \bar{t})}{\sqrt{\sum\limits_x \sum\limits_y (f[x,y] - \overline{f}_{u,v})^2 \sum\limits_x \sum\limits_y (t[x-u,y-v] - \bar{t})^2}} \tag{7}$$

где:

- $f[x,y]$ - исходное изображение
- $t[u,v]$ - искомый шаблон
- $\overline{f}_{u,v}$ - среднее значение интенсивности изображения под шаблоном
- $\bar{t}$ - среднее значение интенсивности шаблона

Именно такой формулой пользуется Matlab при расчёте взаимной корреляции двумерных сигналов с помощью функции `normxcorr2`.

4 Задания к семинару:

1. Построить график корреляционной функции двух произвольных сигналов.
2. Разработать скрипт, позволяющий определить периодичность (или её отсутствие) произвольного сигнала.
3. Наложить на произвольный аудиофайл эффект «Эхо» аналогично заданию из первого семинара. Разработать скрипт, который с помощью автокорреляции убирает наложенное эхо.
4. Разработать скрипт, который с помощью двумерной корреляции позволяет найти шаблон изображения внутри другого изображения.