РАБОТА №4. АВТОКОРРЕЛЯЦИОННАЯ И ВЗАИМНОКОРРЕЛЯЦИОННАЯ ФУНКЦИЯ ДИСКРЕТНЫХ И НЕПРЕРЫВНЫХ СИГНАЛОВ

Цель работы: изучение АКФ и ВКФ дискретных и непрерывных сигналов.

Планируемая продолжительность: 2 академических часа.

Тип работы: с использованием компьютерных средств

В данной работе в качестве основного инструмента используется язык программирования Python. Общие принципы решения задач описаны только для Python вне зависимости от какого-либо окружения, поэтому вы можете использовать любой редактор кода, например JupyterLab, Jupyter Notebook, VSCode и т.д.

**Теоретические основы**

В общем случае взаимная корреляция сигналов описывается выражением:

а в некоторых источниках встречается другое обозначение, отличное знаком перед t:

где обозначает комплексное сопряжение и в случае используемы нами действительны сигналов может быть опущен.

Эта функция характеризует степень «похожести» на при текущем .

То есть, по взаимно корреляционной функции (ВКФ) двух сигналов можно построить график со значениями сдвига по оси абсцисс. На данном графике видно, где сигналы более похожи – при смещении второго на положительный сдвиг (часть графика при ) или же при его смещении на отрицательный сдвиг (часть графика при ).

Пример графика ВКФ приведен на рисунке ниже. Здесь мы видим, что сигналы максимально похожи при сдвиге второго из них на .

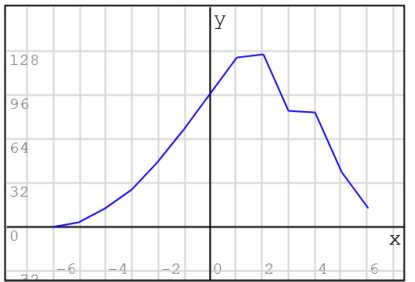


Рисунок 4.1 – Пример ВКФ

Автокорреляционная функция (АКФ) — это частный случай ВКФ, в котором в качестве второго сигнала используется первый (то есть ВКФ сигнала с самим собой):

Функция АКФ симметрична относительно и обязательно имеет максимум в  , поскольку сигнал не может быть более похож сам на себя, чем при нулевом сдвиге. В случае с АКФ можно строить только половину графика (обычно ту, что соответствует ) из-за его симметричности. Пример АКФ представлен на рисунке ниже.

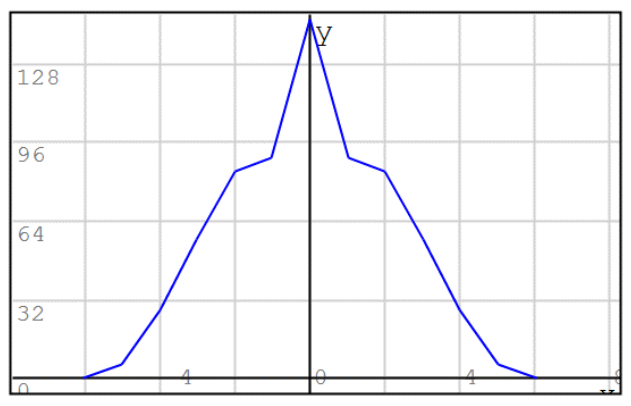


Рисунок 4.2 – Пример АКФ

АКФ используют для выявления зависимостей в сигналах, не поддающихся визуальному анализу. К примеру, АКФ случайного сигнала имеет максимум при и не имеет никаких выраженных зависимостей на графике при . Если же, как на рисунке ниже, визуально случайный сигнал является результатом сложения периодического сигнала (в данном случае синусоиды) и шума, периодическая составляющая будет четко видна на графике АКФ.

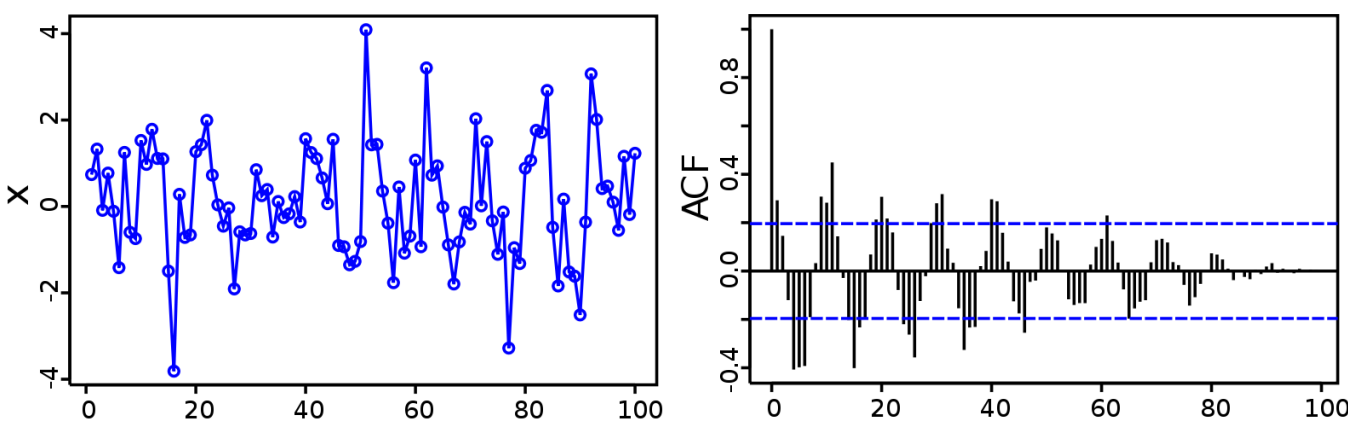


Рисунок 4.3 – Пример выявления полезной   
составляющей по АКФ

**ВКФ и АКФ дискретного сигнала**

Подготовим окружение. Через pip установим все необходимые пакеты.

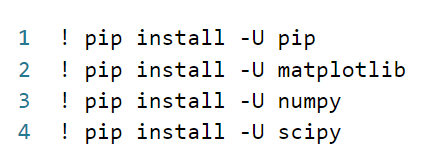


Рисунок 4.4 – Установка пакетов

И импортируем следующее:

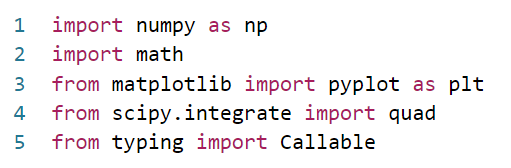


Рисунок 4.5 – Импортирование модулей

Напишем заранее метод для отображения графика для данных вида .

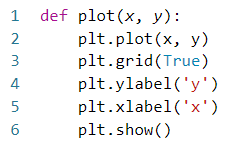


Рисунок 4.6 – Функция построения графика

Сперва зададим два дискретных сигнала, в виде массивов чисел с одинаковым размером:

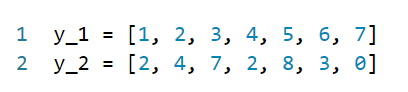


Рисунок 4.7 – Задание массивов

Для осуществления различных преобразований сигналов удобно задавать эти преобразования в общем виде функцией и вызывать для обработки конкретных массивов. Зададим функцию для вычисления ВКФ при текущем сдвиге, ориентируясь на следующее выражение:

где n – сдвиг между массивами (а также номер элемента в ВКФ),

k – номер элемента в первом массиве.

То есть, ВКФ двух массивов также будет массивом, причем каждый из его элементов будет рассчитываться как сумма произведений всех перекрывающихся элементов. Чтобы задать функцию вычисления текущего (k-го) элемента ВКФ, можно реализовать следующий метод.

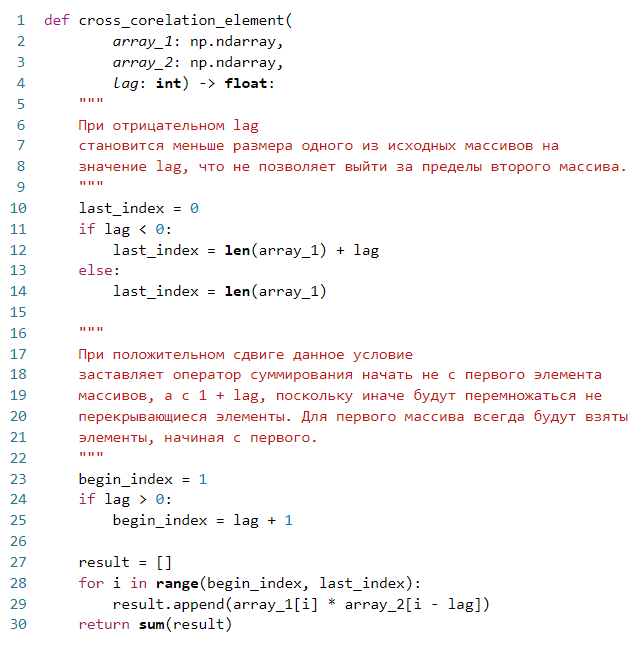


Рисунок 4.8 – Алгоритм вычисления   
текущего (k-го) элемента ВКФ

Пояснения в шести кавычках писать не нужно, это всего лишь комментарий к тому, как работает алгоритм. Также необязательно писать типизацию к входным и выходным параметрам, это, например np.ndarray или ­­ ­­. Такая типизация может помочь интегрированным средам разработки, понять, что за объект будет на входе и выходе, и вызвать подсказки к методам в соответствии с типом, если среда опять же эти подсказки поддерживает.

­­Теперь нужно задать сдвиг. Как было сказано в рамках прошлого занятия, для дискретных отсчетов функция корреляции может быть найдена в диапазоне сдвига от до , где N – число отсчетов в сигнале. Напишем для этого функцию, приведённую ниже.

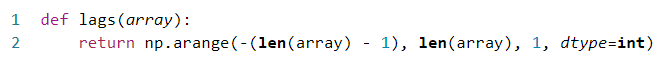


Рисунок 4.9 – Функция создания смещений   
на основе существующего массива

Чтобы вручную не задавать пределы массива смещений на вход будем передавать некоторый массив и на его основе получаем N встроенной функцией len. При помощи функции [np.arange](https://numpy.org/doc/stable/reference/generated/numpy.arange.html) создаём массив смещений от –(len(array) - 1) до –(len(array) - 1) (второй аргумент в np.arrange берёт предел не включительно).

Теперь осталось скомбинировать наши функции, при каждом сдвиге нужно записать результат вычисления   
текущего (k-го) элемента ВКФ в массив.

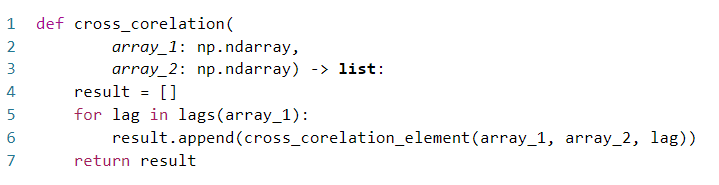


Рисунок 4.10 – Функция создания смещений   
на основе существующих массивов

Вызовем эти функции и нарисуем полученные графики при помощи реализованного в самом начале метода plot.

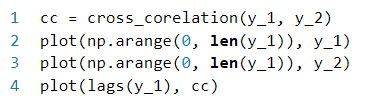


Рисунок 4.11 – Вызов функций

Таким образом получим следующие графики.

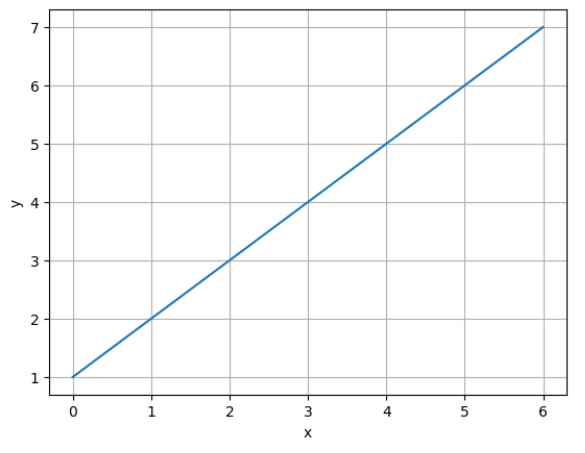


Рисунок 4.12 – График сигнала y\_1

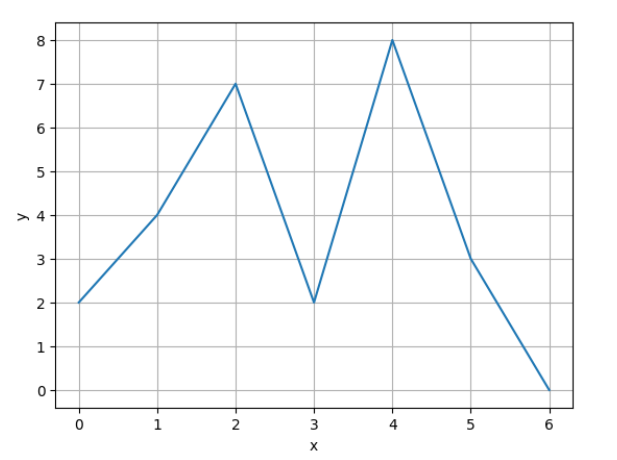


Рисунок 4.13 – График сигнала y\_2

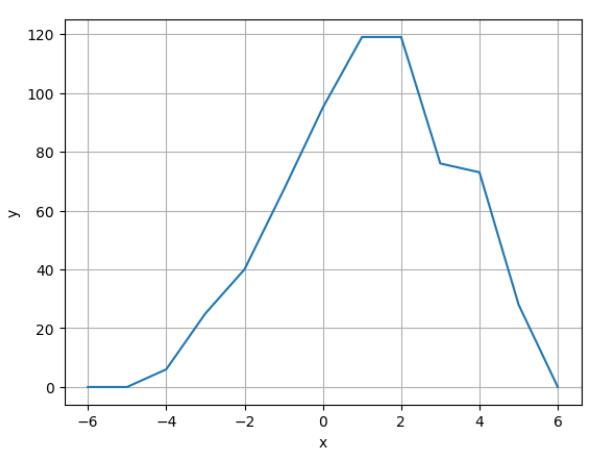


Рисунок 4.14 – График ВКФ y\_1 и y\_2

Как видно из последнего графика наиболее похожей функция y\_1 похожа на y\_2 при смещении равном 2.

Теперь найдем АКФ для сигнала, заданного некоторой функцией. Для этого сначала реализуем функцию по которой будем создавать массив данных.

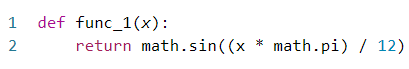


Рисунок 4.15 – Функция для генерации данных

Также зная то, что автокорреляционная функция (АКФ) – это частный случай ВКФ, в котором в качестве второго сигнала используется первый (то есть ВКФ сигнала с самим собой) напишем функцию АКФ.

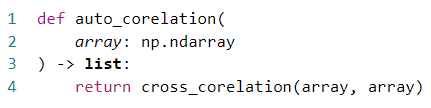


Рисунок 4.16 – Функция АКФ

Теперь создадим массив данных по func\_1 при помощи функции [np.linespace](https://numpy.org/doc/stable/reference/generated/numpy.linspace.html), которая создаст массив от 1 до 100 на 100 элементов с равными промежутками между элементами и выведем её.

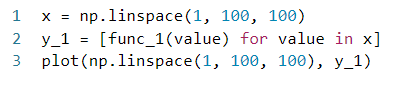


Рисунок 4.17 – Алгоритм создания данных

Получим график нашей функции на рисунке 4.17

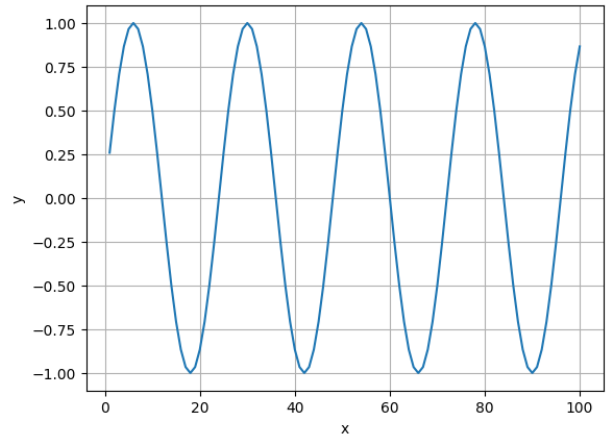


Рисунок 4.18 – График функции синуса

Осталось рассчитать АКФ y\_1 и вывести полученные данные в график.



Рисунок 4.19 – Вызов графика функции АКФ с расчётом

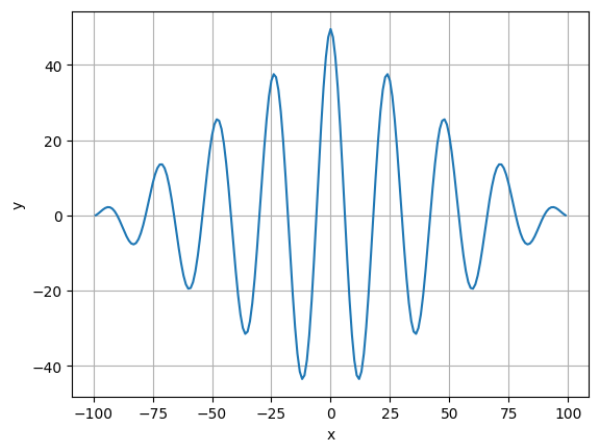


Рисунок 4.20 – График АКФ

**АКФ и ВКФ непрерывных сигналов**

Теперь рассмотрим возможность нахождения аналитического выражения АКФ для простых примеров. В нашем случае АКФ будет вычисляться согласно выражению:

где бесконечные пределы заменены на интервал [0;10] для простоты вычислений.

Обратите внимание, для самостоятельной работы при работе с АКФ сигналы в таблице с вариантами обозначаются как y(x) и y2(x).

Реализуем функцию contigious\_corelation. Обратите внимание, теперь функции не являются массивами данных, теперь это Callable объект, иными словами, сама функция без её вызова. Помимо этого, теперь в нашу функцию передаём переменные a и b – это пределы интегрирования. Также стоит упомянуть внутреннюю функцию mul\_func, для интегрирования нам не нужен результат работы функции, нам нужна сама функция, поэтому при помощи внутренней функции mul\_func, которая обладает областью видимости данных до входных переменных от contigious\_corelation указываем их перемножение при вызове mul\_func, указывая во входных параметр x и параметр смещения T. Для интегрирования используем функцию [quad](https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.quad.html), во входных параметрах передаём функцию mul\_func, пределы интегрирования и в дополнительный аргументах args выставляем смещение. Функция [quad](https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.quad.html) возвращает кортеж из двух элементов, а именно значение интеграла и абсолютную ошибку. Абсолютная ошибка нам не нужна, поэтому пропускаем её, выставляя прочерк вторым аргументом

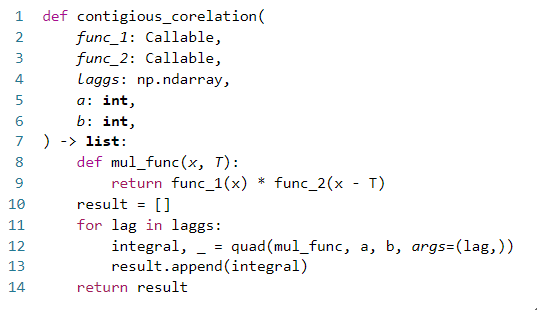


Рисунок 4.21 – Функция ВКФ для непрерывного сигнала

Создаём вторую функцию, от которой будем брать ВКФ.



Рисунок 4.22 – Вторая функция для ВКФ

Теперь осталось задать смещения вышеупомянутой функцией [np.linespace](https://numpy.org/doc/stable/reference/generated/numpy.linspace.html) вызвать реализованную функцию contigious\_corelation и отобразить график результата.

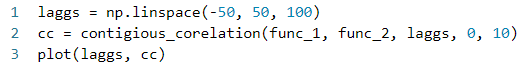


Рисунок 4.23 – Вызов ВКФ для непрерывного   
сигнала и графика для него

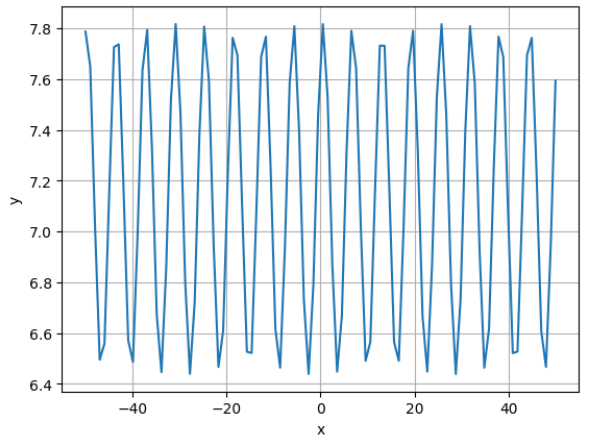


Рисунок 4.24 – График ВКФ

По аналогии с дискретным сигналом получить АКФ сигнала можно вызвав ВКФ с одинаковыми функциями. Реализуем данный подход и получим метод auto\_contigious\_corelation.

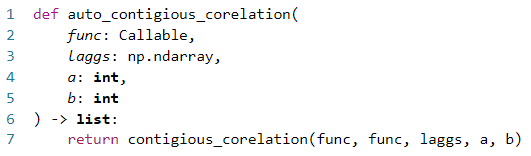


Рисунок 4.25 – АКФ   
для непрерывного сигнала

Как и до этого создаём смещения, вызываем функцию АКФ и строим график.

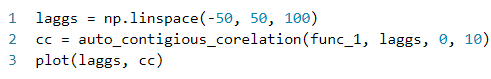


Рисунок 4.26 – Вызов АКФ

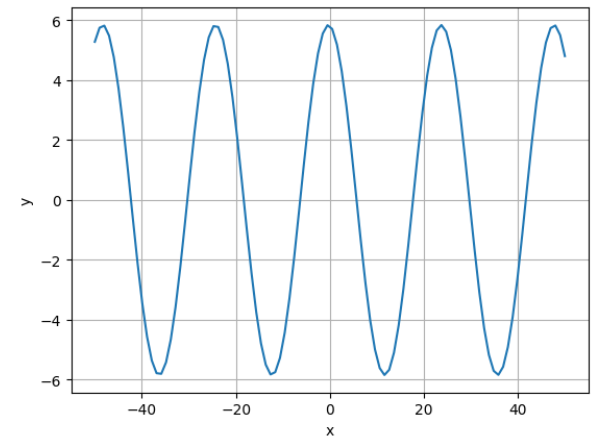


Рисунок 4.27 – График АКФ для непрерывной функции

**Задание**

Указанные значения заменяют собой соответствующие параметры для раздела, посвященного дискретным отсчетам и АКФ непрерывной функции.

