Учебный практикум по LabVIEW

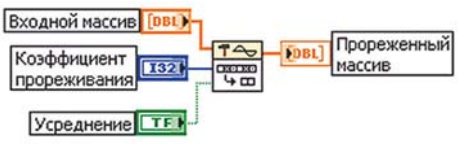
Кто-то ввиду производственной необходимости, а кто-то чисто из любопытства, уже немножко познакомился с возможностями обработки сигналов в LabVIEW. Перечень предлагаемых функций впечатляет. Чего здесь только нет! Вот эта функция знакома, об этой – что-то, когда-то слышал, и эта вроде как небесполезна...

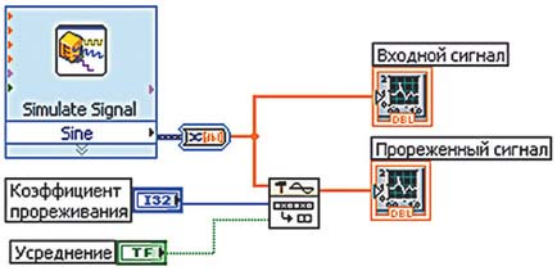
Но зачем их так много? А все для того, чтобы работа в LabVIEW была эффективна во всех отношениях для широкого круга пользователей и доставляла при этом еще и максимум удовольствия.

Первый шаг – цифровая фильтрация – уже позади (см. ПиКАД №4-2006), на очереди – функции обработки сигналов во временной области.

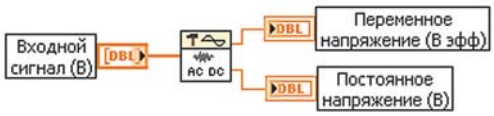
Принято считать, что основными функциями обработки сигналов во временной области являются операции свертки и вычисление корреляционных функций. Можно добавить еще интегрирование и дифференцирование, а также определение параметров импульсов. Это действительно так и есть, но давайте посмотрим, что содержится в соответствующей палитре LabVIEW ***(Functions»Signal Processing»Signal Operation)***. А там более двух десятков непонятных иконок, правда с подписями AC&DC Est..., Zero Padder.vi, Y[i]=Clip{X[i]..., из которых две похожи на Express VI: Conv&Corr и Scale&Map. Наверное, что-то стоящее? Не наверное, а точно. Внесем ясность, who is who.

Функция ***Decimate***, по-нашему "Прореживание", очень полезная функция. Этот VI формирует на своем выходе массив с уменьшенным числом элементов в соответствии с заданным коэффициентом прореживания. Т.е. в выходном массиве будут присутствовать только те элементы входного массива, номера которых кратны коэффициенту прореживания. А еще эта процедура возможна с проведением усреднения по каждому участку функции. Для этого следует установить в состояние ***TRUE*** управляющий вход "Усреднение". Кому все-таки непонятно, что есть децимация, иногда и такое определение можно встретить, рекомендуем написать программку и поэкспериментировать:

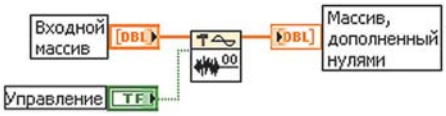




Название функции ***AC&DC Estimator*** простому переводу не подлежит. А выполняет этот VI оценку переменной и постоянной составляющей входного сигнала. Причем делается это с помощью окна Хана, что оказывается более эффективным, чем применение традиционных методов. Вот не было бы этой функции, пришлось бы применять спектральный анализ, выделять составляющую на нулевой частоте и суммировать значения всех остальных компонент спектра. Отличная функция!



С помощью функции ***Zero Padder*** входной массив дополняется нулями с тем, чтобы длина выходного массива стала равна ближайшей степени числа 2.

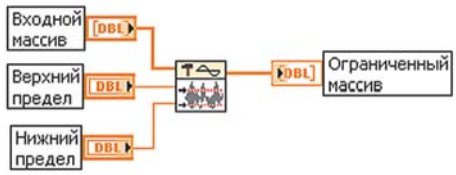


Эта функция имеет управляющий вход. Если его состояние ***TRUE***, то при входном массиве, состоящем, например, из 501 элемента, выходной массив будет содержать 512 элементов. Если же состояние управляющего входа ***FALSE***, то даже при размере входного массива равным числу степени 2, размер выходного массива все равно увеличится вдвое. Пример: 1024 отсчета превратятся в 2048, из которых половина будут нулями. Следует заметить, что ***Zero Padder*** работает как с вещественными ***DBL***, так и с комплексными переменными ***CDB***. Об этой функции следует знать прежде всего тем, кто в своих программах собирается использовать обработку на основе быстрых преобразований Фурье и Хартли. А иначе придется попотеть, создавая непростую подпрограммку из десяти иконок. Вот так.

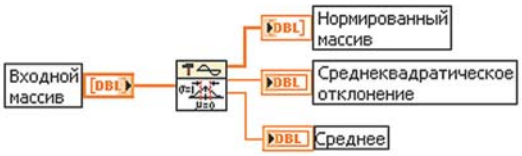
Для построения оригинальных алгоритмов обработки представляет интерес функция ***Y[i]=X[i-n]***. С ее помощью, задав значение параметра "сдвиг", указав при этом знак, можно в выходном массиве получить сдвинутые вправо или влево элементы входного массива. Размер массива не изменяется, потерянные при этом элементы будут заполнены нулями.

Если Вы хотите ограничить элементы массива по амплитуде сверху и снизу, используйте функцию ***Y[i]=Clip{X[i]}***. Задав этому VI верхний ***a*** и нижний ***b*** пре-

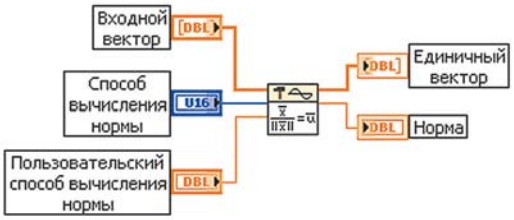
делы, элементы выходного массива будут удовлетворять условию: $y_i = a, x_i > a; y_i = b, x_i < b$ и $y_i = x_i$, при $b \le x_i \le a$.



Для нормализации массива, будь то вектор или матрица, можно использовать готовый VI ***Normalize***, который выполняет преобразование $Y=(X-\mu)/\sigma$, где σ - среднеквадратическое отклонение, а μ - среднее значение:



Получение единичного вектора обеспечивает процедура ***Unit Vector***. Этот VI реализует деление входного массива, представленного в формате DBL или CDB, на норму вектора: $Y=X/\|X\|$.

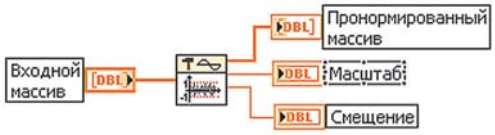


Причем способ вычисление нормы вектора $\|X\|$ может быть выбран из числа предлагаемых (1..4) или задан пользователем (5):

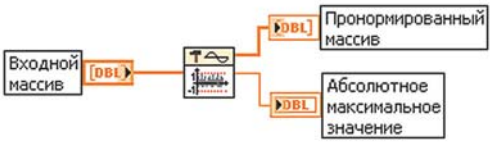
1. **1-norm** $\|X\|=|X_0|+|X_1|+\dots|X_{n-1}|$
2. **2-norm** $\|X\|=\sqrt{|X_0|^2+|X_1|^2+\dots|X_{n-1}|^2}$
3. **Inf-norm** $\|X\|=\max_i(|X_i|)$
4. **-Inf-norm** $\|X\|=\min_i(|X_i|)$
5. **user defined (y)** $\|X\|=\|X_0|^y+|X_1|^y+\dots|X_{n-1}|^y|^{1/y}$

В рассматриваемой палитре LabVIEW есть две функции масштабирования одномерных и двумерных массивов данных ***Scale*** и ***Quick Scale***, которые могут быть использованы для приведения входных массивов к диапазону [-1:1].

VI ***Scale*** определяет масштаб $scale=0.5\ (max(X_i)-min(X_i))$ и смещение $offset=min(X_i)-scale$, а далее выполняется следующее нормирование $Y_i=(X_i-offset)/scale$:

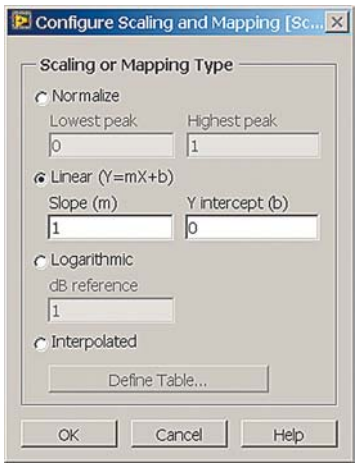


VI ***Quick Scale*** потому и назван быстрым, так как масштабирование выполняется по упрощенному алгоритму $Y_i=X_i/max(|X_i|)$, т.е. только с использованием максимального абсолютного значения входного массива:



Ну а теперь самое время рассмотреть Express VI ***Scale and Mapping***, который позволяет достаточно просто сконфигурировать модуль преобразования сигнала для реализации одного из четырех режимов: ***Normalize***, ***Linear***, ***Logarithmic*** и ***Interpolated***.

Выбираем ***Normalize***, задаем два параметра ***Highest peak*** и ***Lowest Peak***, и получаем на выходе Express VI нормализованный сигнал в заданных пределах. По умолчанию верхний и нижний уровни равны 1 и 0 соответственно.

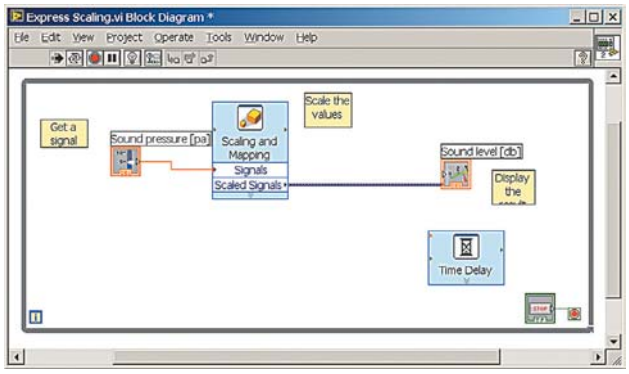


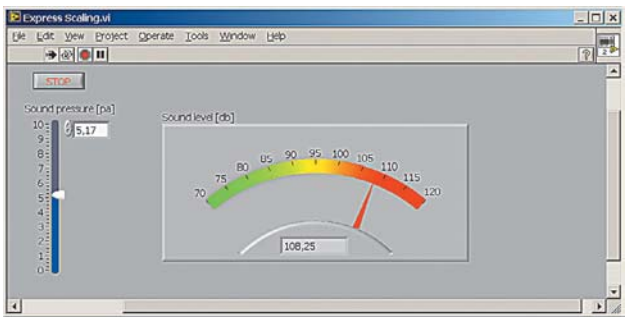
Режим ***Linear*** обеспечивает линейное преобразование сигнала по формуле $Y_i=mX_i+b$. Смещение следует задать в окне Intercept, а коэффициент масштабирования в окне Slope.

Если Вам нужно логарифмическое сжатие сигнала, используйте режим ***Logarithmic*** $Y_i=20\log_{10}(X_i/db\ reference)$.

Четвертый режим рассматриваемого Express VI ***Interpolated*** преобразует входной сигнал в соответствии с таблицей значений ***Define Table***, реализуя линейную интерполяцию.

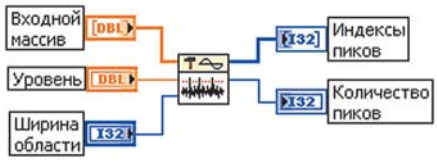
В примере использования Express VI ***Scale and Mapping*** звуковое давление акустического сигнала в диапазоне 0..10 Па преобразуется в уровни шкалы 70..120 дБ:



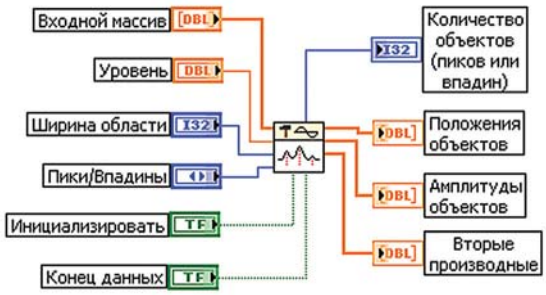


Нельзя не упомянуть о двух интересных функциях для анализа пиков в сигнале: ***Threshold Peak Detector*** и просто ***Peak Detector***.

Пороговый детектор пиков ***Threshold Peak Detector***, анализируя входную последовательность отсчетов, определяет количество пиков ***count*** и индексы первых элементов пиков ***indices***. Под пиком следует понимать участок последовательности, ширина которой определена числом элементов не меньше заданного значения ***width***, а уровень - не меньше значения порога ***threshold***.

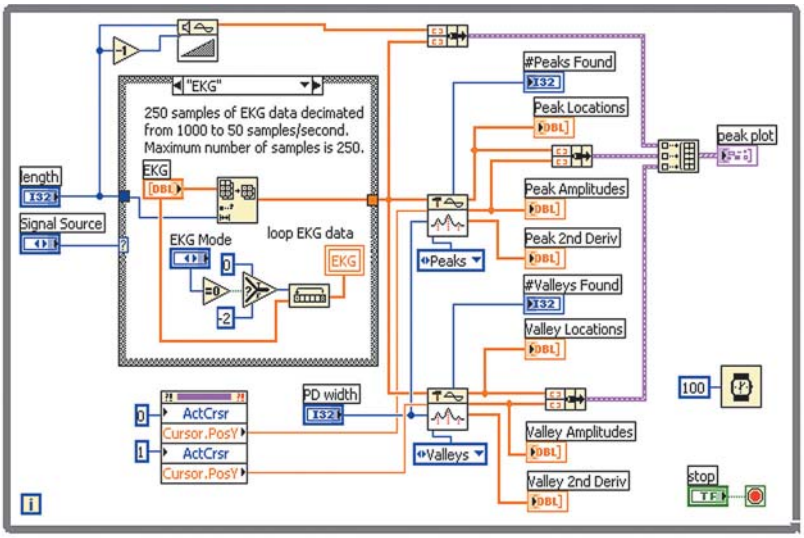


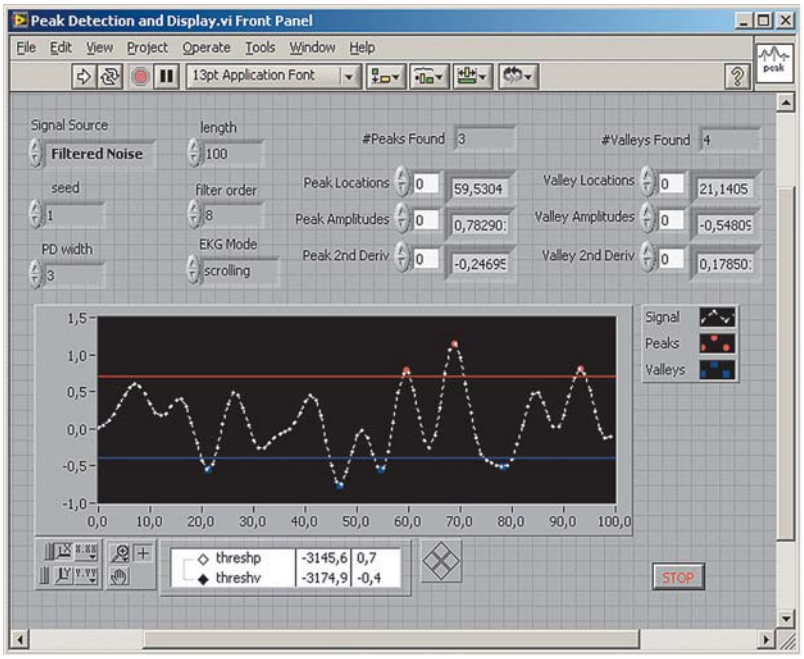
VI ***Peak Detector*** определяет во входном сигнале ***#found*** количество пиков ***peaks*** или впадин ***valleys***, их положение ***locations***, амплитуды ***amplitudes*** и значения вторых производных ***2nd Derivatives***. Для определения этих параметров используется алгоритм параболической аппроксимации. Основными исходными данными для анализа являются уровень сравнения ***threshold*** и размер области ***width***, заданной числом отсчетов, в которой выполняется аппроксимация. Значение ***width*** должно удовлетворять условию ***≥3***. Причем для зашумленных сигналов рекомендуется увеличивать это значение с целью уменьшения вероятности ложных регистраций, а для "чистых" сигналов - наоборот. Выбор режима анализа пики/впадины определяется состоянием входа ***peaks/valleys(0/1)***.



Следует заметить, что входной сигнал для этого VI может быть представлен как в виде одного массива, так и в виде набора блоков данных. Синхронизация работы VI осуществляется по входам ***initialize(T)*** и ***end of data*** следующим образом. При обработке первого блока на входе ***initialize (T)*** необходимо поддерживать состояние ***True***, а последующих - ***False*** . Для входа ***end of data (T)*** должна быть реализована противоположная последовательность. Если обрабатывается один блок данных, названные входы можно не использовать.

И еще пару существенных замечаний к этому VI. Значения на выходе ***locations*** не являются целыми числами, а на выходе ***amplitudes*** отличаются от элементов входного массива, т.к. для их определения используется алгоритм параболической аппроксимации. Зачем нужен выход ***2nd Derivatives*** догадаться не трудно. Вторые производные характеризуют "остроту" пиков и впадин, а их знак идентифицирует тип экстремума. Возможности VI ***Peak Detector*** в полном объеме иллюстрирует приведенный пример.



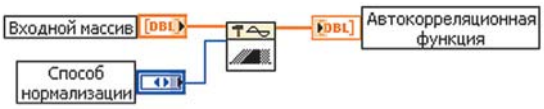


Тем кто знает, или не знает, что серьезный метрологический анализ невозможен без использования процедур свертки, а измерение скорости листа на прокатном стане или выделение гармонического сигнала при мониторинге ротационных машин немыслимо без корреляционного анализа, безусловно, будет интересно узнать, как эти функции реализованы в LabVIEW. Но прежде маленькое замечание: определение автокорреляционной, взаимокорреляционной функций и свертки в LabVIEW возможны для одно- и двумерных массивов, в вещественном или комплексном виде.

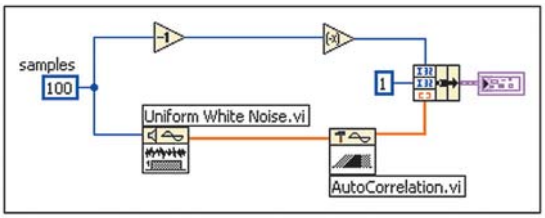
Автокорреляционная функция (АКФ), как количественная интегральная характеристика формы сигнала, определяется интегралом с бесконечными пределами от произведения двух копий сигнала, сдвинутых относительно друг друга. В случае конечной дискретной реализации АКФ вычисляется по формуле:

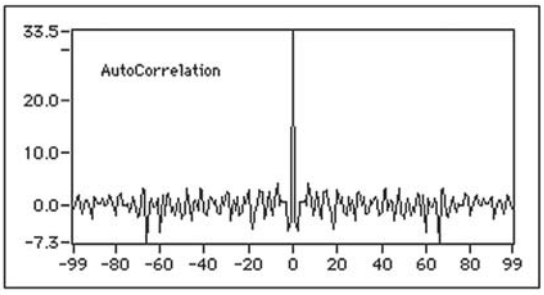
$$Y_i = \sum_{k=0}^{n-1} X_k X_{j+k},$$

где Y_i - отсчеты АКФ; X_i - отсчеты сигнала; n - число отсчетов сигнала; $j = -(n-1), -(n-2),..., -1, 0, 1,..., (n-2), (n-1)$; $X_k=0$ для $k<0$ и $k \geq n$. Выходная последовательность R_{XXi} в VI ***AutoCorrelation*** связана с Y_i соотношением $R_{XXi} = Y_{i-(n-1)}$ для $i = 0, 1, 2, ... , 2(n-1)$, т.к. индексы элементов массивов в LabVIEW не могут быть отрицательными. Т.е. элемент выходной последовательности, соответствующий нулевому сдвигу, имеет индекс n.



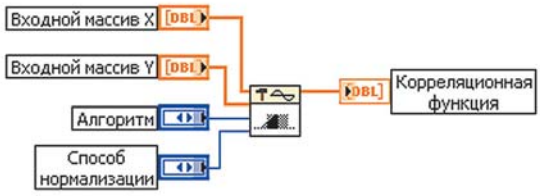
Если необходимо отцентрировать график АКФ, можно воспользоваться приведенной блок-диаграммой и получить АКФ с центром в нулевой точке.





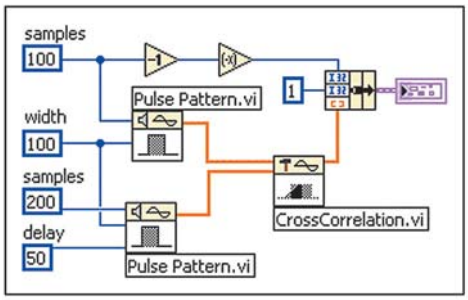
В VI ***1D Autocorrelation (DBL)*** и ***1D Autocorrelation (CDB)*** имеется вход управления ***normalization***, позволяющий выполнить нормирование АКФ со смещением и без него.

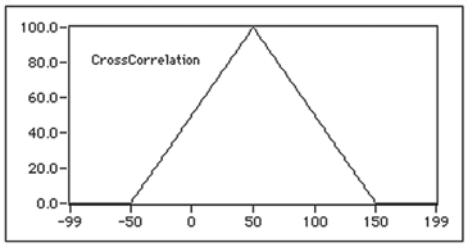
После приведенного выше описания VI ***AutoCorrelation*** в особых комментариях к VI ***CrossCorrelation*** почти нет необходимости:



Вход управления ***algorithm*** определяет метод вычисления взаимокорреляционной функции (ВКФ). Классический метод ***direct (0)*** лучше использовать при коротких выборках сигналов. Метод ***frequency domain (1)***, основанный на быстром преобразовании Фурье, более эффективен при больших реализациях. Однако, применив оба алгоритма к одним и тем же сигналам, можно получить разные результаты. Об этом надо помнить.

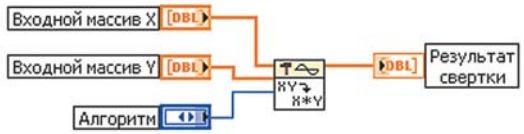
Соберите пример VI с двумя источниками прямоугольных импульсов, добавив индикатор, и убедитесь, что ВКФ двух одинаковых "прямоугольников" дает "треугольник". Это знает каждый, или должен знать. А если не вполне одинаковых? Что получится? У Вас есть возможность это увидеть... Но можно заглянуть и в учебник.



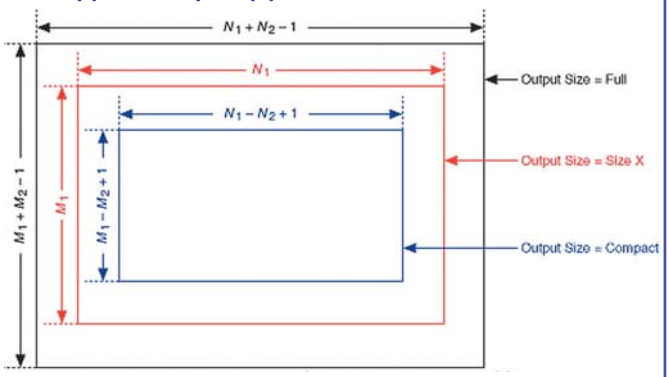


Там же, в учебнике по цифровой обработке сигналов, найдется и определение свертки. Если, мягко выражаясь, не совсем понятно, что такое последовательная линейная комбинация взвешенных единичных импульсов, принцип суперпозиции, интеграл Дюамеля и импульсная характеристика, то стоит компенсировать этот пробел. Ведь операция свертки широко используется, например, при обработке видеоинформации - коррекция эффектов размытия из-за плохого фокуса, смазывания при движении, выделение объектов и т.п.

В LabVIEW свертка двух сигналов ***X*** и ***Y***, т.е. $h_i=X_i*Y_i$, реализуется с помощью VI ***Convolution***:

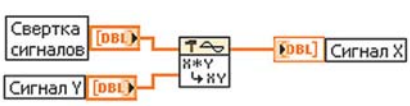


Назначение входов, в т.ч. и управляющего входа **algorithm** аналогично вышерассмотренным для ВКФ, а вот в VI ***2D Convolution (DBL)*** и VI ***2D Convolution (CDB)*** появляется еще один управляющий вход - ***output size***. Он определяет размерность выходного массива - ***Full (0)***, ***size X (1)*** или ***compact (2)***:



где ***N1*** и ***N2*** - число столбцов, а ***M1*** и ***M2*** - число строк в двумерных входных массивах ***X*** и ***Y***.

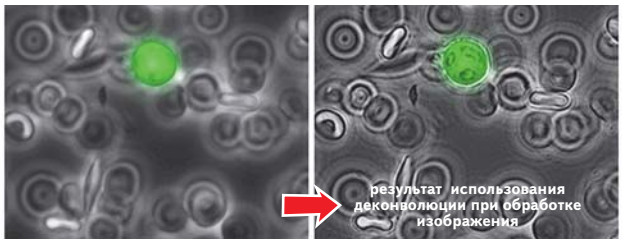
Обратная свертка, деконволюция или развертка - операция, обратная свертке сигналов, а проще - поиск решения уравнения свертки. VI ***Deconvolution*** выделяет информативный сигнал ***X*** из двух последовательностей ***X*Y*** и ***Y***, используя преобразование Фурье:



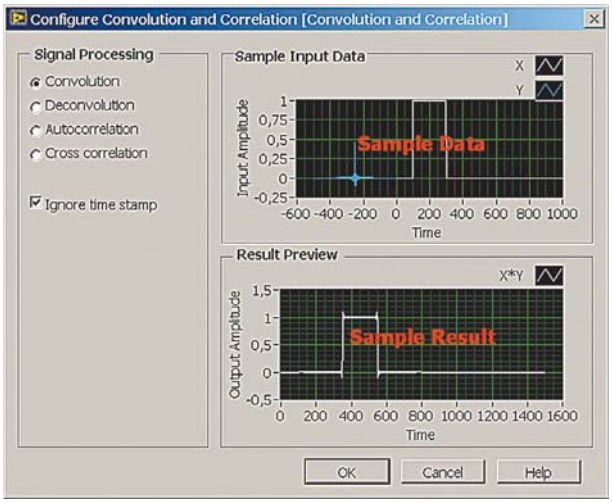
- выполняется преобразования Фурье входных последовательностей ***X*Y*** и ***Y***;
- Фурье преобразование ***X*Y*** делится на Фурье преобразование ***Y***;
- выполняется обратное преобразование Фурье результата деления.

Размер выходного массива ***X*** определяется как ***n–m+1***, где ***n*** - размер ***X*Y***, а ***m*** - размер ***Y***.

Следует отметить, что операция обратная свертке является нестабильной операцией что приводит к неоднозначным результатам. Объяснение достаточно простое - наличие нулей в Фурье преобразовании **Y**. И тем не менее, эта функция также широко используется при обработке графических данных:



Корреляционные функции, свертка и обратное ей преобразование в LabVIEW представлены также в виде единого Express Vi - ***Convolution and Correlation***. Используя диалог настройки, можно выбрать тип преобразований, который необходимо выполнить, в окне ***Sample Input Data*** - просмотреть образцы входных сигналов, а в окне ***ResultPreview*** - результат. Очень удобный инструментарий для начинающих:



Во многих из рассмотренных VI имеется выход **error**, который содержит код ошибки, возникающей при вычислениях. Само по себе числовое значение мало информативно, поэтому, для большей наглядности можно рекомендовать подключать к этому выходу специальный VI ***Error Cluster From Error Code***, который преобразует код ошибки в текстовое описание.

Что ждет Вас в следующем выпуске ПИКАДа? БПФ, весовая обработка или что-то полегче? Ведь ПИКАД №2, 2007 выйдет ближе к периоду отпусков, детских и студенческих каникул. Хотите сюрприз? Будет Вам сюрприз!

Материал практикума подготовлен сотрудниками фирмы "ХОЛИТ Дэйта системс", г.Киев