Цифровые фильтры в SCADA

Анисимов Сергей Юрьевич
anis@pmg.org.ru

Цифровые фильтры используются для уменьшения случайных ошибок в измерениях. Каждый тип фильтр имеет свои преимущества и недостатки. Некоторые значительно подавляют шум, но имеют значительную задержку отклика. Другие не имеют задержки, но плохо фильтруют те или иные шумы. Есть фильтры, которые хорошо фильтруют шум и имеют небольшую задержку, но плохо реагируют на шум произвольной частоты или шум, связанный с большой динамикой технологического процесса.

Выбор между фильтрацией шума и задержкой отклика фильтра является не тривиальной задачей. Кроме этого, надо учитывать, что помимо случайного шума в измерениях может присутствовать систематические ошибки, такие как сдвиг, дрейф, неисправность прибора измерения, потеря точности. Так же ошибки могут возникать в случае быстрых изменений в процессе, когда приборы не могут измерять при тех или иных условиях. Для того чтобы отличить истинное изменение в процессе или ошибку в измерениях иногда необходимо иметь довольно сложную диагностическую систему. Причем желательно, чтобы такая система была совмещена с датчиком или с АЦП. Но это идеальный случай, что делать, когда такого интеллектуального датчика или АЦП нет?

Для оценки качества фильтрации разных фильтров можно использовать интеграл абсолютных ошибок (integral of absolute errors - IAE). IAE вычисляется как сумма абсолютных разностей между истинными значениям в заданные моменты времени и фильтрованными значениями. Если истинных значений нет, то можно использовать сильно сглаженные значения, или восстановить значения вручную.

Экспоненциальные фильтры

Экспоненциальный фильтр или экспоненциальное сглаживание (exponential filter) - это самый простой и самый распространенный фильтр, который используется во многих DCS и SCADA системах. Он так же называется фильтром первого порядка. Аналитически он может, описан следующим уравнением:

$y_k = \theta * x_k + (1-\theta) * y_{k-1}$, или $y_k = y_{k-1} + \theta * (x_k - y_{k-1})$

где x_k – измеренное значение в момент времени k, y_k – фильтрованное значение в момент времени k, θ - коэффициент фильтрации (от 0 до 1, так же он может задаваться в процентах). Чем больше θ, тем меньше степень фильтрации, чем меньше θ, тем больше степень фильтрации. Если θ равно нулю, то фильтрации максимальна (все время на выходе фильтра значение y_{k-1}), если θ равно единице, то фильтрация минимальна (все время на выходе фильтра значение x_k). Обычно, θ равно 0.2 или не больше 0.5.

Как видно зависимость степени фильтрации от коэффициента θ обратная, что не удобно, чтобы избежать этого, можно переписать уравнение таким образом:

$y_k = (1-\theta) * x_k + \theta * y_{k-1}$, или $y_k = x_k + \theta * (y_{k-1} - x_k)$

Экспоненциальный фильтр имеет несколько преимуществ:

1) простота вычисления;
2) не надо хранить несколько предыдущих значений;
3) хорошо подавляет резкие скачки в измерениях;
4) довольно быстро приближается к значениям после скачка.

С другой стороны этот фильтр имеет довольно большую задержку.

Экспоненциальный фильтр – это фильтр с бесконечной импульсной характеристикой (БИХ-фильтр, Infinite impulse response - IIR), т.е. на отфильтрованное значение влияют все предыдущие значения, так как он имеет обратную связь.

Есть разные модификации этого типа фильтра.

Двойной экспоненциальный фильтр или фильтр второго порядка эквивалентен двум фильтрам первого порядка, когда значение y_k подставляется вместо x_k во второй фильтр, для которого задано свое θ_2.

Нелинейный экспоненциальный фильтр использует такой способ вычисления θ:

$$\theta = \min[1, \frac{|x_k - y_{k-1}|}{R}]$$

где R – коэффициент настройки фильтра. Если $|x_k - y_{k-1}|$ больше R, тогда θ равно 1 и фильтрации нет, тем самым данный фильтр будет пропускать резкие скачки значений больше R. Если $|x_k - y_{k-1}|$ меньше R, то фильтр будет больше сглаживать, чем больше разница между x_k и y_{k-1}. Тем самым R позволяет отделить шум от резких изменений значений, которые могут иметь место по технологии. Величина R может быть оценена исходя из статистических свойств шума или возможных изменений в самом процессе.

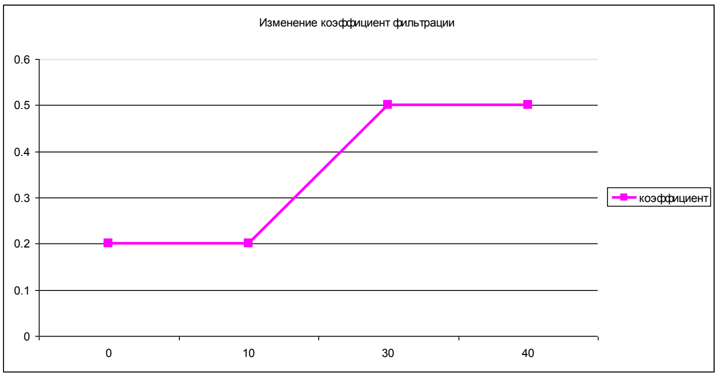
Опять же можно написать выражение для θ, которое поддерживает прямую зависимость степени фильтрации от θ:

$$\theta = \max[\,0{,}1 - \frac{|x_k - y_{k-1}|}{R}]$$

В другом типе нелинейного экспоненциального фильтра коэффициент θ вычисляется таким образом:

$$\theta_k = \begin{cases} \theta_0, если\ |x_k - y_{k-1}| \le \Delta_0 \\ \theta_1, если\ |x_k - y_{k-1}| \ge \Delta_1 \\ \left(\dfrac{\Delta_k - \Delta_0}{\Delta_1 - \Delta_0}\right) * (\theta_1 - \theta_0) + \theta_0, если\ \Delta_0 < |x_k - y_{k-1}| < \Delta_1 \end{cases},$$

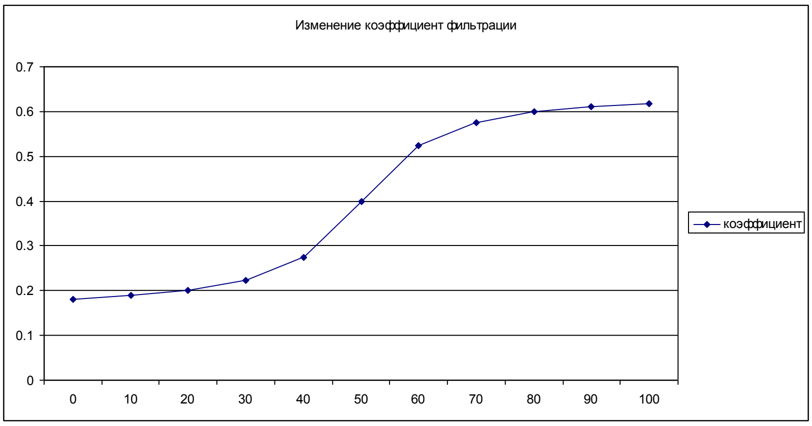
где $\Delta_k = y_{k-1} - x_k$, θ_k - значение коэффициента фильтрации на шаге k, Δ_0 и Δ_1 - минимальное и максимальное значения изменения параметра, соответственно, которые задают интервал линейного изменения θ от θ_0 до θ_1. Пример графика изменения коэффициента фильтрации при Δ_0 =10, Δ_1= 30, θ_0 = 0.2 до θ_1= 0.5.



Таким образом, можно построить множество разных нелинейных фильтров, используя разные функции для получения θ_k в зависимости от Δ_k. Пример такой функции:

$$\theta_k = \left(1 + {arctg\left(\left(\Delta_k * k_1\right) - \left(k_1/2\right)\right) * 2} \Big/ {\pi}\right)/2 + k_2$$

Где k_1 и k_2 подбираются в зависимости от входных данных. Например, если Δ_k меняется от 0 до 100, тогда $k_1 = 10$, $k_2=0.15$:



Если вычисление функции θ_k в зависимости Δ_k вызывает затруднение, то можно использовать табличное представление.

Фильтры скользящего среднего

Фильтр скользящего среднего (moving average filter) другой классический фильтр. Обобщенное выражение можно записать в виде:

$$y_k = \sum_{i=k-N+1}^{N} w_i x_i \, ,$$

где N – число точек данных обрабатываемых фильтром, w_i - вес каждого измерения x_i, в начальный момент времени все значения x_i равняются x_0, затем добавляются все новые значения x_i, после того как значений x_i будет больше N – старые значения x_i отбрасываются. В простейшем случае можно присвоить всем w_i значения равные 1/N, тогда для работы фильтра нужно N измеренных (не отфильтрованных) значений x_i, которые суммируются и делятся на N, т.е. вычисляется скользящее среднее.

Фильтр скользящего среднего – это фильтр с конечной импульсной характеристикой (КИХ-фильтр, Finite impulse response –FIR), т.е. на отфильтрованное значение влияет только N последних измеренных значений.

Этот фильтр также как и экспоненциальный прост в вычислении, легко настраивается, но и так же при больших N имеет большую задержку и значение IAE. Кроме того, этот тип фильтра лучше подходит для вычисления среднего значения, а не текущего. Впрочем, можно установить весовые коэффициенты таким образом, чтобы последние значения имели наибольший вес. Например, линейно-сглаженное скользящее среднее:

$$y_k = \frac{x_k N + x_{k-1}(N-1) + \ldots + x_{k-N+1}}{N + (N-1) + \ldots + 1}$$

При этом задержка у фильтра снижается, но фильтрация сильного шума становиться хуже.

Для примера приведем подбор разных коэффициентов w для N=5.

$y_k = 1/16\ (x_{k-2} + x_{k+2}) + 1/4\ (x_{k-1} + x_{k+1}) + 3/8\ x_k$

Данный фильтр хорошо фильтрует высокочастотный шум, т.е. срезает резкие выбросы значений.

$yk = -\ 1/16\ (x_{k-2} + x_{k+2}) + 1/4\ (x_{k-1} + x_k+1) + 5/8\ x_k$

Хорошо фильтрует низкочастотный шум.

$y_k = 1/4\ (x_{k-1} + x_{k+1}) + 1/2\ x_k$

Некий компромисс между первым и вторым фильтром.

Можно объединить экспоненциальный фильтр и фильтр скользящего среднего в результате получиться фильтр экспоненциально взвешенного скользящего среднего (Exponentially Weighted Moving Average (EWMA):

$y_k = y_k + \theta*(\ y_{k-1-}\ y_k\ ),$

где y_k - скользящее среднее. Главное достоинство такого фильтра в том, что он включает в себя не только среднее за период, а все измеренные значения (за счет y_{k-1}). При этом последним значениям придается больший вес. Такой фильтр хорошо сглаживает шум, задержка снижается, но очень плохая реакция на резкие скачки значений. Впрочем, в этом случае можно использовать нелинейный экспоненциальный фильтр.

Полиномиальные фильтры

Как известно полиномами разной степени производят аппроксимацию экспериментальных данных, для того чтобы представить эти данные в виде функции. Этот же метод можно использовать и для фильтрации, аппроксимируя данные полиномами за какой-то период времени, и вычисляя значение полученного полинома в текущий момент времени.

Полиномиальный фильтра порядка m (polynomial filter) задается таким полиномом:

$$y_k = c_m t_k^m + c_{m-1} t_k^{m-1} + \ldots + c_2 t_k^2 + c_1 t_k^1 + c_0$$

где, t_k - текущее время, m = порядок полинома (больше 0), $c_0 \ldots c_m$ - коэффициента полинома, которые вычисляют с помощью метода наименьших квадратов, при этом надо найти минимум квадрата разницы между измеренными и рассчитанными по полиному значениями:

$$\underset{c_0 \ldots c_m}{Min} \sum_{i=1}^{N} (x_i - y_i)^2$$

где x_i – измеренное значение, y_i – отфильтрованное значение, N – число временных шагов (точек данных), которые включены в фильтрацию.

Полиномы очень хорошо сглаживают данные, соответственно, и данный способ фильтрации будет очень хорошо устранять шум. Чем больше степень полинома, тем лучше будет полином повторять исходные данные, соответственно, хуже будет фильтрация. Чем ниже степень полинома, соответственно, полином будет сильно сглаживать данные, и точность аппроксимации упадет. Кроме этого, данный фильтр – это фильтр с конечной импульсной характеристикой. Так же надо учесть, что при добавлении каждого нового измерения в фильтрацию, надо заново искать коэффициенты полинома, что довольно затруднительно при большой степени полинома. Поэтому, как правило, используется фильтр первого порядка.

Существующие методы регрессионного анализа (аппроксимации или приближения функций по методу наименьших квадратов) позволяют найти достаточно простые методы вычисления коэффициентов линейной регрессии:

$$c_1 = \frac{\sum_{i=1}^{N} x_i * \sum_{i=1}^{N} t_i - N * \sum_{i=1}^{N} x_i t_i}{\left(\sum_{i=1}^{N} x_i\right)^2 - N * \sum_{i=1}^{N} x_i^2}$$

$$c_0 = \frac{\sum_{i=1}^{N} t_i - c_1 * \sum_{i=1}^{N} x_i}{N}$$

Помимо, аппроксимации полиномами так же можно выполнять аппроксимацию сплайнами. Но в любом случае для данного типа фильтров характерно большой объем вычислений и трудность выбора степени точности аппроксимации (за счет увеличения или уменьшения степени полинома).

Фильтрация импульсных помех

В некоторых случаях, помимо случайного шума в измерениях содержатся некоторые выбросы, пики, помехи, которые носят импульсный, непредсказуемый характер. Для фильтрации такого рода помех применяются несколько разных типов фильтров.

Фильтр от импульсных помех (noise spike filter) использует предел изменения значений Δ, для фильтрации импульсов:

$$y_k = \begin{cases} x_k, \text{ если } |x_k - y_{k-1}| \le \Delta \\ y_{k-1} - \Delta, \text{ если } y_{k-1} - x_k > \Delta \\ y_{k-1} + \Delta, \text{ если } x_k - y_{k-1} > \Delta \end{cases}$$

Таким образом, фильтр действует, только на одно значение, если разница между текущим значением и предыдущим фильтрованным значением больше, чем предел, то измеренное значение заменяется на разницу или сумму предыдущего значения и предела, в зависимости от знака превышения.

Это достаточно простой фильтр, который будет «срезать» выбросы по пределу, при этом, если следующие измеренные значения x_k так же будут находиться далеко от y_{k-1}, то значения y_k будут постепенно приближаться к ним, каждый раз на предел измерения, тем самым, фильтр имеет некоторую задержку.

Медианный фильтр (median filter) использует массив из N последних значений, которые отсортированы по возрастанию. Значение, расположенное в середине отсортированного списка является фильтрованным. В начале массив имеет меньше, чем N значений, постепенно нарастая со временем. Если размер массива четный, то выбирается предыдущее перед серединой значение.

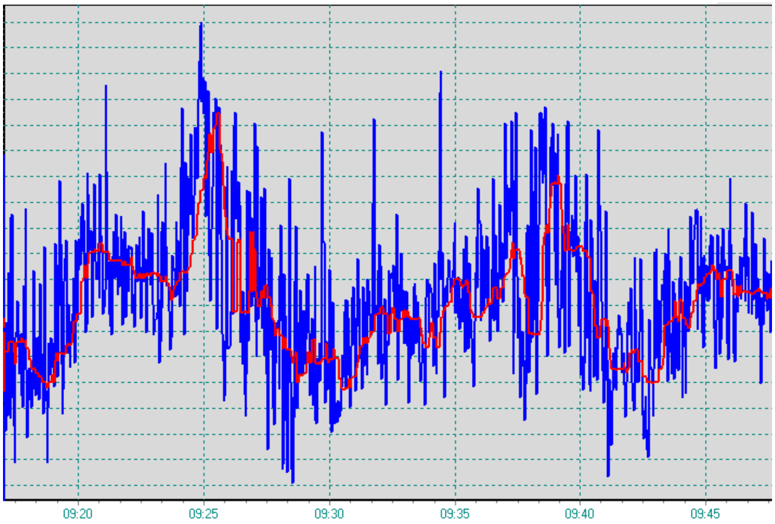
Рассмотрим, пример: пусть, x={2,3,5,80,4,1}, размер массива N=3, тогда по шагам:

1. y_1=2, отсортированный массив array_sort={**2**};
2. y_2=2, отсортированный массив array_sort={**2**,3};
3. y_3=3, отсортированный массив array_sort={2,**3**,5};
4. y_4=5, отсортированный массив array_sort={3,**5**,80};
5. y_5=5, отсортированный массив array_sort={4,**5**,80};
6. y_6=4, отсортированный массив array_sort={1,**4**,80};

Этот фильтр так же имеет некоторую задержку. Например, если значения параметра резко изменились и остались на некоторое продолжительное время, то этот фильтр не сразу будет давать на выходе значения после этого скачка, а с задержкой равной половины времени необходимой до заполнения массива новыми значениями после скачка. А это бывает не допустимо, так как оператор может поздно среагировать на изменения в процессе. В этом случае, можно ввести граничное значение, если не отфильтрованное значение больше, чем граничное, то не отфильтрованное значение не проходит фильтрацию, а сразу передается на выход.

Кроме этого, очень сложно подобрать размер массива, в некоторых случаях помех настолько много, что они «забивают» массив, и невозможно отличить истинные данные от импульсов.

Рассмотрим пример работы такого фильтра с N=30, период опроса 1 сек:



Синим цветом – значения, не фильтрованного параметра, красным – фильтрованного. Как видно из графика фильтр довольно хорошо отбрасывает помехи, если они длятся не более 30 секунд (высокочастотный шум), в тоже время, если помехи длятся более 30 сек – фильтрации не происходит (низкочастотный шум).

Отсечение значений по условиям

В некоторых случаях необходимо просто отсечь («обрезать») значение, при котором не выполняются одно или несколько условий. Такие алгоритмы нельзя отнести к фильтрам, тем не менее, они позволяют избавиться от помех, которые трудно фильтруются другим образом.

Простейшее отсечение производится по заданной нижней и верхней границе:

$$y_k = \begin{cases} x_k, \textit{если } x_k > x^{\min} \text{ и } x_k < x^{\max} \\ y_{k-1}, \textit{если } x_k \le x^{\min} \\ y_{k-1}, \textit{если } x_k \ge x^{\max} \end{cases},$$

где $x^{\min}$ и $x^{\max}$ – минимальная и максимальная границы значений. В некоторых случаях, вместо предыдущего значения y_{k-1} можно использовать минимальное или максимальное значение, которое может принимать параметр.

Данное ограничение можно применять в тех случаях, когда значение параметра не может быть больше или меньше какого-то заданного значения, например: если значение расхода меньше 5% от шкалы, то, скорее всего это не достоверное значение и можно присвоить значению расхода ноль; если уровень в емкости выше пороговой отметки, то это не достоверное значение и его можно игнорировать, сохраняя последнее истинное значение.

Тем не менее, в некоторых случаях данное отсечение не применимо исходя из возможного поведения параметра. Например, значение расхода резко меняется, иногда уходит в начало и/или в конец шкалы. В этом случае можно ввести отсечение по верхней и по нижней границе, сохраняя последнее значение до скачка. Но в случае, когда расход отсутствует, всегда будет выводить последнее значение до нижней границы, что не верно. В данном случае необходимо ввести максимальное время отсечения значений параметра, после истечения, которого условия отсечения не будут выполняться.

Если параметр не может менять быстрее, чем какое-то заранее известное значение (например, уровень в емкости не может убывать быстрее, чем максимальная сумма всех входящих или исходящих потоков в емкости), то можно применить следующие ограничение.

$$y_k = \begin{cases} x_k, \text{если } |x_k - x_{k-1}| \le \Delta \\ y_l, \text{если } |x_k - x_{k-1}| \ge \Delta \end{cases},$$

где Δ - максимальная разница между измеренным и предыдущим значением, y_l - последнее достоверное значение.

Данный фильтр будет хорошо работать, если значения резко меняются, а не «ступеньками», постепенно, за некоторое время. Для отсечения «ступенек» необходимо ввести максимальное время скачка t_{max}, т.е. отсечение будет работать в течение времени t_{max}, если разница между последним достоверным значением и текущим все еще больше максимально допустимой разницы за это время. В таком случае максимальную разницу лучше всего вводить как максимальную скорость изменения параметра в секунду.

Гибридные фильтры

Используя несколько фильтров подряд можно существенно повысить степень фильтрации от помех. Например, можно использовать первым медианный фильтр, чтобы отфильтровать импульсы, затем или экспоненциальный фильтр (сглаживание), или применить фильтр скользящего среднего (усреднение).

Работа любого фильтра базируется на некоторых предположениях. Обладая полной статистической информацией об объекте, а так же зная законы по которым работает объект можно подобрать подходящий фильтр для оптимальной фильтрации сигнала.