Comparison of different outlier filtering methods with applications to ultrasonic flow measurements

Peter Gruber
pgconsult, Switzerland

Abstract

Several strategies for digital filtering of strong variations including elimination of outliers in measured data are presented and compared. The filters are applied to data from acoustic transit time measurements. The methods are however not restricted to these measurements. The data to be filtered exhibits a number of specific characteristics: first strong outliers are present due to measurement errors or high frequency random physical phenomena and second the data might contain a trend. The different filter strategies are split into separate groups: linear/nonlinear, of FIR or IIR type and applicable for on-line or off-line implementation. Among the methods considered are well-known strategies like the Grubb's filter (see appendix of IEC41 norm), MAD and clipping filters. Extensions of some of these filters are proposed. Special care has been given to the parametrization of the filters. Good results were obtained with a modification of the MAD filter: the original threshold values are made dependent on the trend in the data. This method requires the following operations:

- Trend evaluation by the method of least squares
- Sorting
- Threshold determination
- Rejection of outliers if needed
- Replacement strategy
- Low pass filtering of processed data

1. Introduction

Hydraulic efficiency measurements rely heavily on properly recorded pressure, flow, temperature and power signals. The time series of these quantities are usually corrupted by noise and outliers. For evaluating the efficiency for example, the corresponding formula is valid for quantities which are 1) low pass filtered and free of outliers and 2) correspond to a steady state or quasi-steady state condition of the plant. In some cases a trend can also be accepted for further processing of the signals. In all these cases two filtering mechanisms are crucial: the detection and removal of outliers and the detection of steady states or quasi-steady states. The following questions have to be answered: When is a recorded signal an outlier or when is it still a noisy measured value of the underlying physical process? How long can a measurement be delayed by filtering? Are the measurements used in a closed loop or are they only used for monitoring? What is known about the process and the measurement noise? The following paper deals specifically with discrete time filtering of outliers in case of constant or time varying operating conditions of the plant under inspection. The filters are split into the following classes :

- Linear non recursive (FIR) filters
- Linear recursive (IIR) filters
- Nonlinear recursive clipping filter
- Nonlinear non recursive (FIR) filters
- A novel nonlinear FIR filter for outliers in trendy data

The implementation issue of these filters will be addressed separately. Finally some of the filters will be applied to recorded flow measurement data and compared to each another. The presentation follows the report from etaeval GmbH for the hydropower plant in Wettingen, Switzerland [1].

2. Linear non recursive (FIR) low pass filtering

The simplest low pass filter of this kind is the moving average of length n: $y_k = \frac{1}{n}\sum_{i=0}^{n-1} x_{k-i}$ (1)

with the corresponding z-transform of its transfer function

$$G(z) = \frac{1}{n}(1 + z^{-1} + .. + z^{-(n-1)}) = \frac{1}{n}\frac{1 - z^{-n}}{1 - z^{-1}}$$

and frequency response: $G(e^{j\omega T_s}) = \frac{1}{n}\frac{\sin(\frac{n\omega T_s}{2})}{\sin(\frac{\omega T_s}{2})}$ T_s : sampling time

The zeros of this transfer function are at the following positions: $f_i = \frac{i \cdot f_s}{n}$ $i = 1,..,n-1$

If a periodic signal of period T has to be suppressed by such a filter, n has to be chosen in the following way: $n = i \cdot T / T_s = i \cdot 20$ $i = 1,...$, which gives a minimum length filter for i=1 as: $n = T / T_s$.

It is possible to improve the filter characteristic by cascading two such filters in series. If they have the same length n, the cascaded filter length will be doubled (2n-1). The frequency response has still the same zeros as the individual filters, but the frequencies close to the zeroes are much heavier damped.

Two such filters can also be put in parallel. The filter length of each filter can be chosen such that the individual outputs can follow different signal variations. The combination of the outputs can then be done in a linear or nonlinear way in a subsequent block.

A moving average filter is a mediocre low pass filter. To design a better filter, different design methods (see S.K. Mitra & J.F. Kaiser [2]) are available e.g. in the Matlab software environment. Here a design method for FIR filter is shown and compared to the moving average: polynomial filters R.W.Hamming [3]. As an example the filter coefficients are given for a quadratic filter of order 5,7,9 and 11:

$$\begin{aligned}
n = 5:\ & g = \frac{1}{35}[-3, 12, 17, 12, -3] \\
n = 7:\ & g = \frac{1}{21}[-2, 3, 6, 7, 6, 3, -2] \\
n = 9:\ & g = \frac{1}{231}[-21, 14, 39, 54, 59, 54, 39, 14, -21] \\
n = 11:\ & g = \frac{1}{429}[-36, 9, 44, 69, 84, 89, 84, 69, 44, 9, -36]
\end{aligned} \qquad (2)$$

The zeros of these FIR filters are not as easily determined as in the case of the moving average. These filters can then again be cascaded if a better attenuation is required next to the zeros.

Example n=11, filter length 11 and 21, moving average and quadratic polynomial filter of 11th order

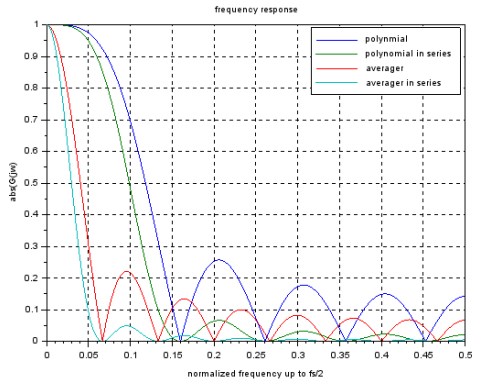

Figure 1: amplitude of the frequency response of simple FIR filters (length=11 resp. 21)

These filters are not well suited for eliminating outliers. They can only reduce the effect of the outlier.

Example: the input u is always 1 except for an outlier at k=10 of height 10

u_k=[1 1 1 1 1 1 1 1 1 1 **10** 1 1 1 1 1 1 1…] k=0,1,2,….. (3)

For a simple moving average of length n=11, the maximal output is $u_{max} = (10 \cdot 1 + 1 \cdot 10)/11 = 1.82$ for k between 10 and 20 (see Fig.2), whereas the simple quadratic polynomial filter of order n=11 (last line of equation 2) reaches a maximal output $u_{max} = (340 \cdot 1 + 10 \cdot 89)/429 = 2.86$ at k=11.
The polynomial filter has a better passband characteristic, is however worse in suppressing the outlier (28.6% instead of 18.2%).

3. Linear recursive (IIR) low pass filtering

A linear recursive (IIR) low pass filter requires for the same filter effect a much lower order than a FIR filter at the expense of a nonlinear phase and an infinite memory. The simplest first order discrete low pass filter with input u and output y with amplification factor 1 is given by the following relations:

$$y_k = (1-\alpha) y_{k-1} + \alpha u_{k-1}$$
$$G(z) = \frac{\alpha z^{-1}}{1-(1-\alpha) z^{-1}} \qquad (4)$$

$(1-\alpha)$ is the forgetting factor, α is the weighting factor for the incoming data. Typical values of α lie between 0.05 und 0.4. For $\alpha = 0.2$, the step response of the recursive filter reaches 90% of its final value at position k+1=11 (corresponding to a moving average length of n=10), see Fig. 2. The capability of suppressing an outlier depends on α: the lower α is, the higher the suppression is and the slower the filter reacts.

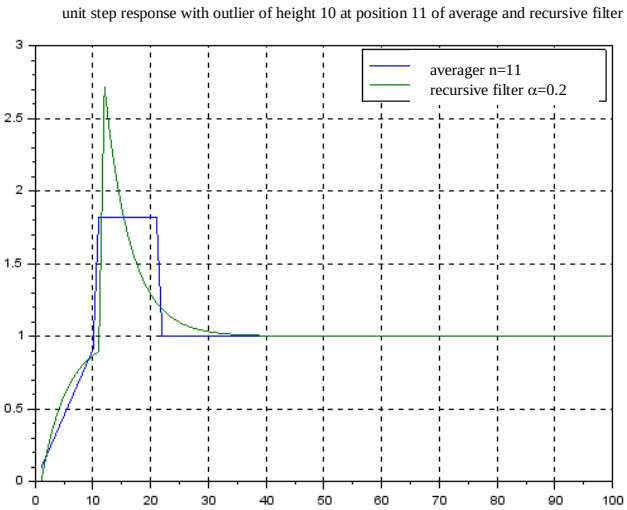

Figure 2: outlier behaviour of recursive filter of first order (α =0.2) and moving average of length 11, input as equation (3)

As can be seen in Fig. 2, the recursive filter weights the input too strong: in order to have the same effect as the moving average α must be halved. Linear recursive filters of higher order may improve the situation but not substantially. What is therefore needed are nonlinear filters.

4. Recursive nonlinear (Clipping) low pass filtering

If we look at the response of the recursive filter to an outlier (Fig. 2), it is clear that some kind of limiting operation helps. The clipping filter of first order can be realized by an integrator in a feedback loop (Fig. 3). If the limiter is not active, the loop acts like a linear low pass filter of first order. If however the limiter is active, the limit $u_{\max}$ is kept and the integration speed is limited. $u_{\max}$ and K are the two tuning parameters of the filter.

.

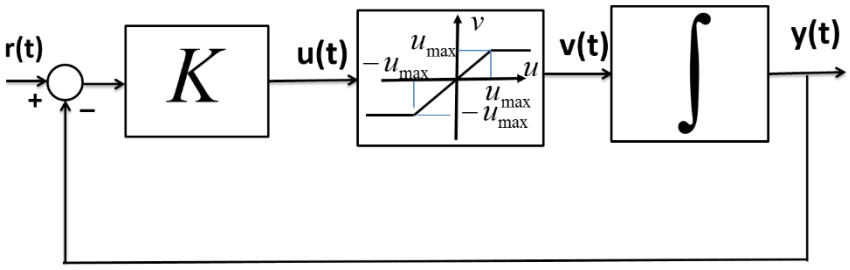

Figure 3: first order low pass filter with clipping

For the transfer function G(s) between r(t) and y(t) we can write in the linear case:

$G(s) = \frac{K}{s+K}$, with a gain 1 and a time constant of 1/K. If the transfer function is discretized with the Euler forward rule, one obtains the first order difference equation: $y_{k+1} = (1 - KT_s)y_k + KT_s u_k$, with $\alpha = KT_s$.

The limiter has the following static nonlinearity:

$$v_k = f(u_k) = f(K(r_k - y_k)) = \begin{cases} u_{\max} & u_k \geq u_{\max} \\ u_k & -u_{\max} < u_k < u_{\max} \\ -u_{\max} & u_k \leq -u_{\max} \end{cases} \quad (5)$$

With equation (5), the nonlinear difference equation can be written as: $y_{k+1} = y_k + T_s f(K(r_k - y_k))$.

For the same outlier case as before, the situation can be improved substantially (see Fig.4): though the filter reacts faster as in the former cases, the maximum output can be reduced to 1.2 with the tuning parameter K=0.7 and $u_{\max} = 0.2$ $(T_s = 1)$.

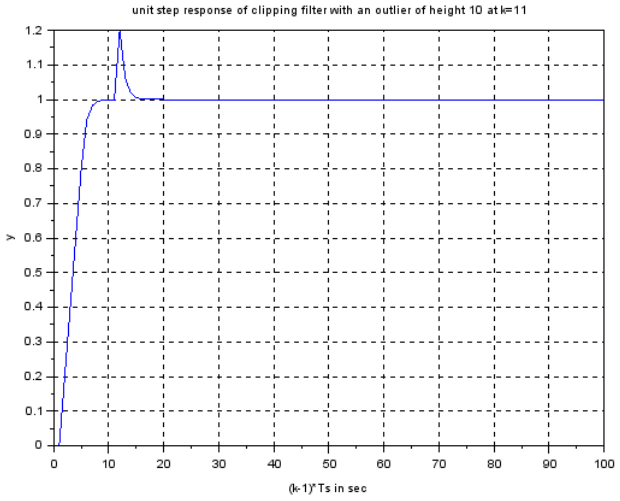

Figure 4: Clipping filter with K=0.7 and $u_{\max} = 0.2$ $(T_s = 1)$

5. Nonlinear nonrecursive outlier-filtering

In order to eliminate outliers completely, a criteria is needed that decides if a measured value is still a valid one or an outlier. The criteria is a check of the data samples against thresholds. The filters of this class considered here are:

1) Median filter (median filter with fixed, MAD-filter with variable) thresholds
2) Standard deviation filter (Grubbs or Nalimov) with variable thresholds

5.1 Median-filter

For the median filter no assumptions must be made about the probability distribution of the data (normal, symmetric, one peak only, …). The following example is an extended version of an example from H.P. Beck-Bornholdt/H.H. Dubben [4]:

Example 1: Given is a time serie of n=15 data points (samples) x:

x = [48 55 35 51 60 47 75 55 76 66 87 102 90 135 168]

This series is now sorted in increasing order in order to determine the median value, which is defined as the one value, for which 50% of the data are lying to the left and 50% to the right. This operation is a nonlinear one because the ordering of the data is interchanged. The newly sorted list looks like

x_{sort} = [35 47 48 51 55 55 60 **66** 75 76 87 90 102 135 168],

with $med_x = 66$.

Additionally we can also compute the mean and the standard deviation of the data series:

mean: $\mu_x = 76.66$ standard deviation: $\sigma_x = 35.96$

5.1.1 *Fixed number of deleted data points*

The simplest way of deleting minimal and maximal values of the n samples of the time serie, is to delete the uppermost and lowest values of the sorted list x_{sort}. For n=15, typically 2 or 3 smallest and largest values are deleted. This leads in the example to the following reduced list x_{sort_red} of length 11 (2 samples deleted on each side):

x_{sort_red} = [48 51 55 55 60 66 75 76 87 90 102]

This new data series has still the same median value of 66, but a new mean of 69.55 and a new standard deviation of 17.84. If an estimate of the mean of the new data series is required, it is possible to average over an arbitrary number of data points which are grouped symmetrically around the median value. If no averaging is applied the median value is chosen as output y, in example 1: $y = med_x = 66$. The other extreme case is the averaging over all values of the reduced list, in example 1: $y = \mu_x = 76.66$. All other symmetrical choices of number of data points (3,5,7,9) are also possible. The following questions arise by this way of deleting extreme values: How many values should be deleted on both sides of the original list? If a number of values are deleted, in what confidence interval (in %) lies the median value of the underlying population distribution inside the span of the not deleted values?

Example 2: n=5. A single measurement value is per definition with a probability of 50% larger or smaller than the median value of the underlying population distribution of the measurement values. If we pick randomly 5 independent samples of the underlying population, the chance that all 5 samples are smaller (resp. greater) than the median value is:

$$p(\forall x_i < med_x) = \left(\frac{1}{2}\right)^5 = \frac{1}{32} \qquad \text{resp. } p(\forall x_i > med_x) = \left(\frac{1}{2}\right)^5 = \frac{1}{32} \tag{6}$$

For the probability that the median value lies inside the range of the 5 samples one gets:

$$p(med_x \text{ inside the range}) = 1 - \frac{1}{32} - \frac{1}{32} = 0.9375$$

Or in other words: the confidence interval of the chance that the median value of the underlying population is inside the range of 5 measurement values is 93.75%. In statistics often the 95% confidence interval is a measure for the quality of an estimate (here the test is a two sided one). Obviously this confidence interval cannot be reached with 5 samples. No sample can be deleted for the 95% interval. If one takes a larger number, then deleting of samples is possible to get a 95% confidence interval. The determination of the confidence interval if deleting of samples is allowed is more complex.

Example: n=8. In analogy to (6) one obtains:

$$p(\forall x_i < med_x) = \left(\frac{1}{2}\right)^8 = \frac{1}{256} \qquad \text{resp. } p(\forall x_i > med_x) = \left(\frac{1}{2}\right)^8 = \frac{1}{256} \tag{7}$$

This gives a confidence interval of 99.2%. If one extreme value is deleted, then first the probability has to be determined that one and only one out of 8 samples is larger or smaller than the median value. This probability is 8 times the probability that all 8 samples are larger or smaller than the median value

$$p(\exists x_i < med_x \; oder > med_x) = 8 \cdot \left(\frac{1}{2}\right)^8 = \frac{8}{256}$$

The probability that the median value is inside the range of the remaining 7 values is:

$$p(med_x \text{ inside the range of 7 remainig values}) = 1 - \frac{1}{256} - \frac{1}{256} - \frac{8}{256} = 0.9609$$

This means that one extreme sample can be deleted and a confidence interval of 95% is still guaranteed. It does not matter which of the extreme value is deleted. For the confidence probability P that the median value is inside the value range of the remaining values (n=total number of samples, x=number of samples deleted at the low end, y=number of samples deleted at the high end), the following formula holds:

$$P = \frac{\sum_{i=x+1}^{n-y-1} \binom{n}{i}}{2^n}$$

Table 1 is generated by this formula.

n	x	y	**n**	x	y	**n**	x	y	**n**	x	y
6	0	0	**18**	5	2	**31**	10	8	**85**	34	31
				4	4		9	9		33	32
7	0	0	**19**	5	4	**35**	12	9	**91**	37	32
							11	11		36	35
8	1	0	**20**	5	5	**41**	14	13	**95**	38	36
										37	37
9	1	1	**21**	6	4	**45**	16	14	**100**	41	36
				5	5		15	15		40	39
10	1	1	**22**	6	5	**51**	19	15	**105**	43	40
							18	18		42	41
11	2	1	**23**	7	4	**55**	20	19	**111**	46	41
				6	6					45	44
12	2	2	**24**	7	6	**61**	23	21	**115**	48	42
							22	22		47	45
										46	46

13	3 1 2 2	**25**	7 7	**64**	24 23	**120**	50 46 49 48
14	3 2	**26**	8 6 7 7	**65**	25 22 24 24	**121**	50 48 49 49
15	3 3	**27**	8 7	**71**	28 23 27 26	**123**	51 48 50 50
16	4 3	**28**	9 7 8 8	**75**	29 27 28 28	**124**	52 47 51 49 50 50
17	4 4	**29**	9 8	**81**	32 29 31 31		

Table 1: number x and y of deleted extreme values in function of sample size n for a 95% confidence interval for the median value of underlying population to be inside reduced sample range [4]

Example 3: n=11, x=3 and y=3. For a 95% confidence interval x=2 und y=1 is required according to Table 1. The confidence interval for x=3 and y=3 is actually much lower:

$$P = \frac{\sum_{i=4}^{7}\binom{11}{i}}{2^n} = \frac{\binom{11}{4}+\binom{11}{5}+\binom{11}{6}+\binom{11}{7}}{2^{11}} = \frac{\left(\frac{11!}{4!7!}\right)+\left(\frac{11!}{5!6!}\right)+\left(\frac{11!}{6!5!}\right)+\left(\frac{11!}{7!4!}\right)}{2^{11}} = \frac{2\cdot 330 + 2\cdot 462}{2^{11}} = 0.7734$$

That means a confidence interval of only 77%. In case of example 1, the confidence interval of 95% is met.

5.1.2 *MAD-filtering: number of deleted samples dependent on data*

The MAD (Median Absolute Deviation)-filter is a two stage sorting filter by which a threshold for the rejection of outliers can be found. The procedure is as following (see Menhold & al. [5]):

1) Sorting of samples and determination of the median value med_x as is done for the median filter.
2) Determination of the absolute deviations of the samples from the median value.
3) Sorting of the absolute deviation and determination of its median value $absmed_x$.
4) Scaling of $absmed_x$ by a factor 1.4826 such that $absmed_x$ is free of bias for a Gaussian statistics. The scaling can be done in other ways too, see Pearson & al. [6].
5) Determination of lower and upper thresholds.
6) Rejection of the samples if they are outside the thresholds.

Example 1 (section 5.1): x = [48 55 35 51 60 47 75 55 76 66 87 102 90 135 168]

x_{sort}= [35 47 48 51 55 55 60 **66** 75 76 87 90 102 135 168] $med_x = 66$

abs(x_i-med_x) = [18 11 21 15 6 19 9 11 10 0 21 36 24 69 102]

abs_{diff_sort} = [0 6 9 10 11 11 15 **18** 19 21 24 31 36 69 102]

$absmed_x = 18$

Scaled value: 26.68 lower threshold: 66-26.68=**39.3** upper threshold: 66+26.68=**92.68**

In the sorted list x_{sort} the lowest and the 3 highest values have to be eliminated (35 and 102 135 168) that means x=1 and y=3. With the resulting x and y the 95% confidence interval can be

met (P=98.19%) It is therefore a little larger than the one obtained with the median filter with x=y=3 (see Table 1). It is obvious that for an asymmetric elimination a new median value results.

5.2 Standard deviation filtering: number of deleted samples dependent on mean and standard deviation

These filters are implemented as follows:

1) Determination of mean μ and variance σ^2 resp. standard deviation σ of n samples.
2) Determination of g_{crit} (Grubbs Table 2) resp. q_{crit} (Nalimov Table 3) for a given n or f=n-2
3) Determination of the quantities (n>2) g_i and q_i

Grubbs: $g_i = \frac{|x_i - \mu|}{\sigma}$ for all samples x_i

Nalimov: $q_i = \frac{|x_i - \mu|}{\sigma}\sqrt{\frac{n}{n-1}}$ for all samples x_i

4) Check of all g_i and q_i values against the thresholds g_{crit} and q_{crit} for a given α listed in Tables 2 and 3 and removal if necessary. In the case of the Grubbs test [7] the maximal identified outlier has to be removed and a new test is carried out with n-1 samples. This is repeated until no outliers are detected anymore.

n	g_{crit} α=0.05	g_{crit} α=0.01	n	g_{crit} α=0.05	g_{crit} α=0.01	n	g_{crit} α=0.05	g_{crit} α=0.01
3	1.1543	1.1547	15	2.5483	2.8061	80	3.3061	3.6729
4	1.4812	1.4962	16	2.5857	2.8521	90	3.3477	3.7163
5	1.7150	1.7637	17	2.6200	2.8940	100	3.3841	3.7540
6	1.8871	1.9728	18	2.6516	2.9325	120	3.4451	3.8167
7	2.0200	2.1391	19	2.6809	2.9680	140	3.4951	3.8673
8	2.1266	2.2744	20	2.7082	3.0008	160	3.5373	3.9097
9	2.2150	2.3868	25	2.8217	3.1353	180	3.5736	3.9460
10	2.2900	2.4821	30	2.9085	3.2361	200	3.6055	3.9777
11	2.3547	2.5641	40	3.0361	3.3807	300	3.7236	4.0935
12	2.4116	2.6357	50	3.1282	3.4825	400	3.8032	4.1707
13	2.4620	2.6990	60	3.1997	3.5599	500	3.8631	4.2283
14	2.5073	2.7554	70	3.2576	3.6217	600	3.9109	4.2740

Table 2: thresholds according to Grubbs: α corresponds the 1-0.01*(confidence interval in %)

f	q_{crit} α=0.05	q_{crit} α=0.01	q_{crit} α=0.001	f	q_{crit} α=0.05	q_{crit} α=0.01	q_{crit} α=0.001
1	1.409	1.414	1.414	19	1.936	2.454	2.975
2	1.645	1.715	1.730	20	1.937	2.460	2.990
3	1.757	1.918	1.982	25	1.942	2.483	3.047
4	1.814	2.051	2.178	30	1.945	2.498	3.085
5	1.848	2.142	2.329	35	1.948	2.509	3.113
6	1.870	2.208	2.447	40	1.949	2.518	3.134
7	1.885	2.256	2.540	45	1.950	2.524	3.152

8	1.895	2.294	2.616	50	1.951	2.529	3.166
9	1.903	2.324	2.678	100	1.956	2.553	3.227
10	1.910	2.348	2.730	200	1.958	2.564	3.265
11	1.916	2.368	2.774	300	1.958	2.566	3.271
12	1.920	2.385	2.812	400	1.959	2.568	3.275
13	1.923	2.399	2.845	500	1.959	2.570	3.279
14	1.926	2.412	2.874	600	1.959	2.571	3.281
15	1.928	2.423	2.899	700	1.959	2.572	3.283
16	1.931	2.432	2.921	800	1.959	2.573	3.285
17	1.933	2.440	2.941	1000	1.960	2.576	3.291
18	1.935	2.447	2.959				

Table 3: thresholds according to Nalimov [8]: α corresponds the 1-0.01*(confidence interval in %)

Example 1(section 5.1): n=15, threshold Grubbs: 2.55 threshold Nalimov: 1.923

g= [1.15 0.82 0.79 0.71 0.60 0.60 0.46 0.29 0.04 0.01 0.28 0.37 0.70 1.62 2.54]

That means, Grubbs criteria identifies no outlier, Nalimov criteria identifies one.

The methods based on mean and standard deviation are less robust in removing outliers because the quadratic deviations make the thresholds larger. Only the worst outliers can be removed.

6. On-line implementation of the filters

If the filter is implemented in an on-line measurement system, the crucial question is how fast (with what time delay) a valid output y of the filtered data has to be generated. If the y is used purely for monitoring and off-line post processing, the delay can be larger as in the case of alarming or if the y is fed back in a closed loop control application.

6.1 Cleaning data first („off-line strategy“)

For purely measurement applications the following procedure for implementing an outlier filter can be used in the following way. At each time step the procedure is:

1) The oldest measurement value is discarded from the list and replaced with the newest sample.
2) Sorting algorithms (median values) are performed and/or statistical quantities of the new list are determined.
3) The thresholds are determined based on the quantities obtained from point 2).
4) The elimination of outliers which are beyond the thresholds is performed.
5) The remaining samples are filtered (average, low pass, median value) and the output y is generated.

The strategy of this implementation is that first the data of the old list are updated and cleaned resulting in a new list. Then the thresholds are computed. The newest sample is then checked against the new thresholds. This filter reacts slower compared to the implementation below, but has an inherent filtering mechanism.

6.2 Output generation first („on-line strategy”)

For control applications the output y of the measured value has to be generated fast. Therefore the following implementation strategy is applied at each time step:

1) The new incoming measurement value is compared to the thresholds obtained from the old n samples.
2) If the new measurement value is not identified as an outlier, it will be directed straight to the output y. The output can then be further low pass filtered if needed.
3) If the new measurement value is identified as an outlier, then it is discarded and the new output y is set as:
   - old median value
   - average of old samples
   - old output value.
4) In all cases the oldest sample of the old list is then eliminated and is replaced by the new sample.
5) Sorting algorithms (median values) and/or statistical quantities of the new list is performed.
6) The thresholds are determined based on the quantities obtained from point 5).

This strategy generates first an output and processes afterwards the necessary steps for adjusting the thresholds. Therefore the delay of the output y is small.

6.3 Initial conditions of filter

The filter memory is continuously filled up until the n delays are occupied. During this initialisation phase the output y is determined as the average of the already filled values. If the initialisation is complete the above procedures start. The length n of the filter is dependent on the application. If periodic components in the measurement signal should be filtered by the above filters, then n has to be adjusted accordingly.

7. Outlier filtering for trendy data

If time series show a trend, then it makes sense to adjust the threshold along the time window. With a least square estimate of offset and slope along the n samples of the time window, the trend can be found. For the samples x_i at times t_i, i=1,…, n, the slope is given by:

$$slope = \frac{n \cdot \sum_{i=0}^{n-1} t_i x_i - \sum_{i=0}^{n-1} t_i \sum_{i=0}^{n-1} x_i}{n \cdot \sum_{i=0}^{n-1} t_i^2 - \left(\sum_{i=0}^{n-1} x_i \right)^2} \qquad \Delta t = t_{i+1} - t_i$$

The newly proposed on-line filtering mechanism for trendy data is now for each time step as follows:

1) The new incoming measurement value is compared to the thresholds at the end of the window length n obtained from the old n samples.
2) If the new measurement value is not identified as an outlier, it will be directed straight to the output y. The output can then be further low pass filtered if needed.
3) If the new measurement value is identified as an outlier, then it is discarded and the new output y is set as old output value.
4) The oldest sample of the old list is then eliminated and is replaced by the newest sample.
5) The slope (trend) of the updated n samples of the new list is determined.
6) MAD procedure (see section 5.1.2) including $upper_{MAD}$ and $lower_{MAD}$ threshold determination is performed.
7) The upper and lower thresholds are adjusted along the window length n by the following adjustment :

$$upper(i) = upper_{MAD} + k \cdot slope \cdot (i-1)\Delta t$$
$$lower(i) = lower_{MAD} + k \cdot slope \cdot (i-1)\Delta t \qquad i = 1,...,n \qquad 0 < k < 1$$

Similar procedures could be derived for the thresholds of other outlier filters as the Grubbs filter.

8. ADM flow measurement example

The example is taken from [1]. The flow measurement signal in this application is supposed to be used in a feedback loop for regulating the flow through a Kaplan turbine. The measurement samples are updated with a sampling time of 0.2 sec and later decimated to 1 sec. The time delay of the signal should be kept due to the feedback a small as possible. Therefore only filters of the on-line type are applied. Two time series are examined:
1) Start up of the turbine which means with a trend in the data: the flow is increased from 0 to 27m^3/s. The outlier filter has to be tuned such, that the increase (trend) of the flow is not treated as an outlier. If the original data are analysed, some periodic components can be detected during the time intervals where no or only as small flow increase is recognizable. These periodic components have to be suppressed additionally by the filtering process. As the original data did not show any strong outliers, outliers were added artificially at t=220,350 and 535 (double) sec (see Fig. 5)

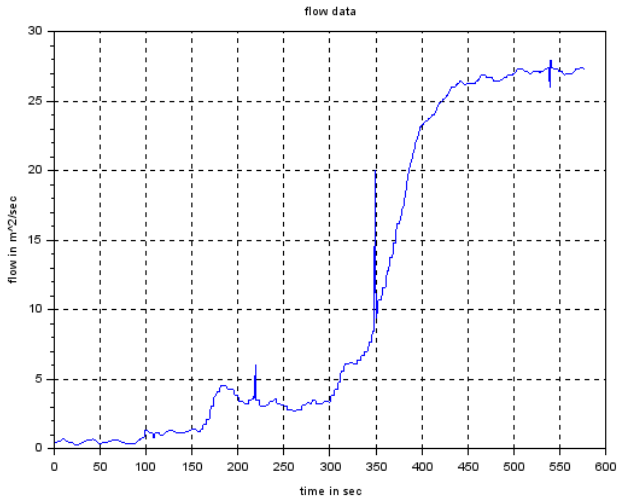

Figure 5: real start up flow data corrupted by three artificial outliers

2) Turbine in quasi steady state operation with real outliers at t=1830 until 1840 sec (Fig. 6). There is an unexpected drop of more than 15% which has to be eliminated as unwanted outliers.

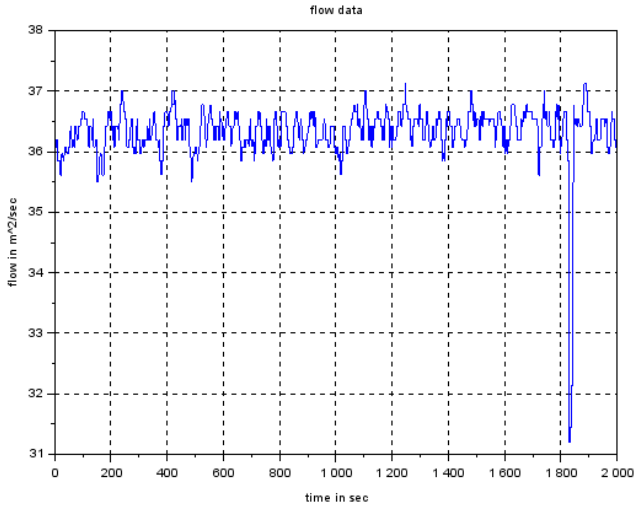

Figure 6: quasi-steady state flow data corrupted by real outliers

Different on-line outlier filters are applied to both situations:

- Median filter of filter length n=15, with fixed number of deleted samples left and right (3, 3) and averaging over remaining 9 samples

- Moving average of length n=15
- Clipping filter with K=0.2 and u_{max}=-u_{min}=0.5
- trend MAD on-line: Trend filter with MAD filtering (length n=15), k=0.2,

The filter parameters (same values for both cases) have been chosen such that the outliers are removed as good as possible, periodic components dampened, the delay in following the ramp never exceeds 20 seconds and the quasi steady state is filtered.

8.1 Turbine start up

The artificial outliers could all be suppressed by the four outlier filtering methods (see Fig. 7 and Fig. 8). The median fix on- line filter and the MAD on-line filter both have problems with following the steep changes. A tendency to keep the old value dominates the behaviour. This could be changed if the filters are tuned specifically for this situation. Clipping and the moving average have an inherent time delay in all situations.

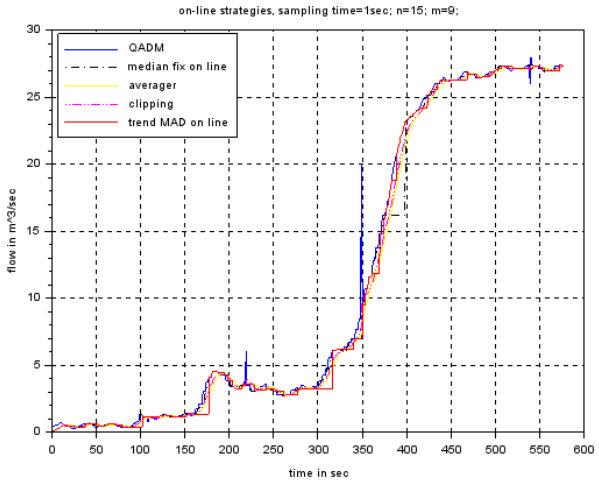

Figure 7: outlier filter behaviour for turbine start up.

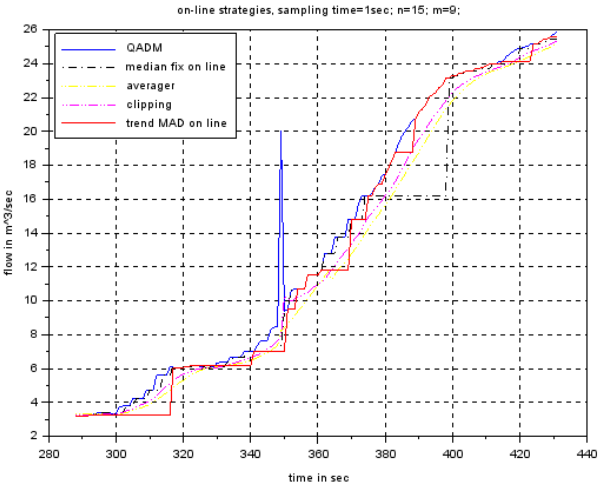

Figure 8: Zoom of Fig. 7 from 290 to 430 sec

8.2 Quasi steady state operation

For this situation all on-line filters behave well in following the signal in normal conditions. At the time of the outlier only the trend MAD on-line filter is capable to suppress the signal sufficiently. The cost of this positive benefit is the tendency of this filter to stay inert (long periods of constant output signal y).

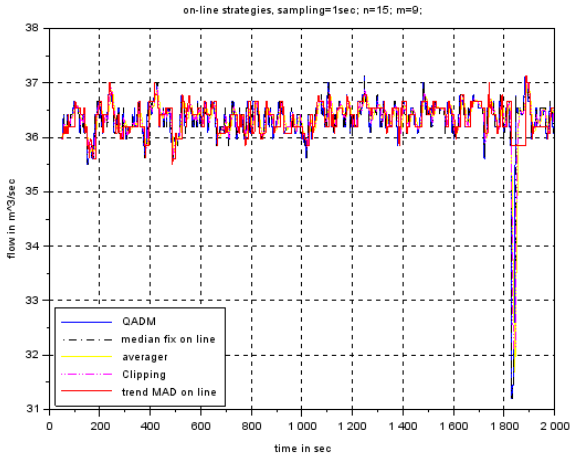

Figure 9: Outlier filter behaviour in quasi steady state

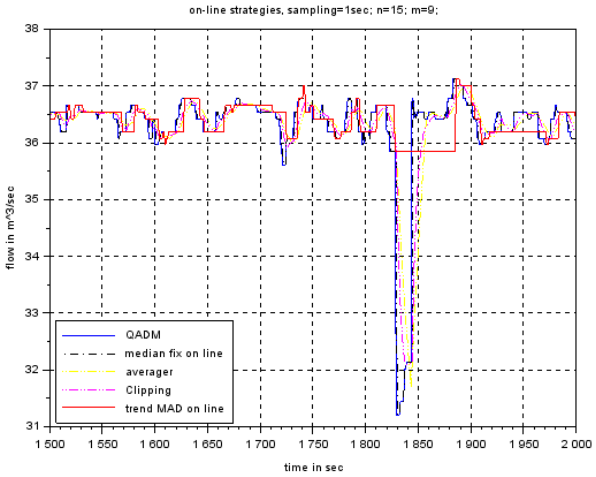

Figure 9: Zoom of Fig. 9 from 1500 to 2000sec

9. Conclusions

For on line applications it is recommended to use filters based on „on-line strategies“ only. If the filters should work in quasi steady state conditions and for trendy conditions then a compromise has to be chosen between outlier rejection ability and ability of following signal variations. The proposed trendy MAD on-line filter seems to be a good choice for these applications. This filter can also be tuned in for more efficient way if it is used in specific situations. This is only partially true for the other filters.

[1] P. Gruber: Outlier-Filterungsmethoden mit Anwendung auf Ultraschall-Durchflussmessdaten, erweiterte Version, etaeval Bericht 2015-10, 2015
[2] S.K. Mitra, J.F. Kaiser: Handbook for digital signal processing, John Wiley & sons, 1993
[3] R.W. Hamming: Digital Filters, Prentice Hall 1977
[4] H.P. Beck-Bornholdt/H.H. Dubben: Der Hund, der Eier legt, rororo Sachbuch 61154, 2006
[5] P.H. Menhold, R.K. Pearson, F. Allgöwer: On-line outlier detection and removal, MED99
[6] R.K. Pearson, Y. Neuvo, J. Astola, M. Gabbouj: The Class of Generalized Hampel filters, 23rd European Signal Conference (EUSIPCO), 2015, Nice
[7] IEC41: Field acceptance tests to determine the hydraulic performance of hydraulic turbines, storage pumps and pump-turbines, Appendix B, p. 353, 1991
[8] L. Sachs: Angewandte Statistik, 4. Auflage, Springer Verlag 1973