Online outlier detection and removal

P.H. Menold, R.K. Pearson, F. Allgöwer *
Institut für Automatik
ETH Zürich
CH-8092 Zürich

Abstract

Outliers occur regularly enough in real-world measurement data to constitute a significant practical problem that is not adequately addressed by traditional smoothing filters designed to reduce the effects of high-frequency noise. To address this problem, this paper describes a simple data cleaning filter for outlier detection and removal which is based on a causal moving data window that is appropriate to real–time applications like closed loop control. This filter is an extension of the well–known median filter: the observed data point y_k is compared to the median $y_k^\dagger$ of present and past data points. If the distance between these points is large relative to a specified threshold, y_k is declared an outlier and replaced with a more reasonable value $y_k^\star$. In the most favorable circumstances alters the above described data cleaning filter only outliers (e.g., shot noise) and does not modify nominal data points. Simple implementations of this filter require few tuning parameters and no explicit process model is required for filter tuning. This paper presents some useful tuning guidelines based on simple characterizations of the nominal variation seen in outlier-free portions of the data. To illustrate the utility of this filter, applications are presented for both real data examples and a simulation example where the exact results are known and performance can be assessed more precisely. It is also demonstrated that the data cleaning filter described here can be combined with traditional linear smoothing filters to achieve both protection against outliers and effective noise reduction, but the outlier filter should preceed the noise filter to achieve these results.

1 Introduction

Outliers or anomalous data points occur frequently in measurement data. We have seen this behavior in a variety of industrial process datasets, blood pressure measurements, helicopter pitch angle measurements, and speed measurements on flexible shafts. A specific example is the shaft speed measurement data shown in Fig. 1; the "spikes" evident in this dataset are clearly inconsistent with the majority of the data values (approximately 20000 samples). In some control applications (particularly those using derivative control), these outliers can have an extremely damaging effect on controller performance. This paper describes the real-time extension of a simple nonlinear digital filter (Ling *et al.* (1984); Astola and Kuosmanen (1997); Pearson (1999)) which has proven to be quite effective in removing outliers for *off–line* data analysis. The goal of our work is to extend these ideas to a filter that allows on–line real–time applications. We state the basic ideas behind the new on–line filter in Section 2 and call it throughout the paper the *data cleaning filter*. This data cleaning filter is a generalization of the median filter (Gallagher and Wise (1981); Astola *et al.* (1987); Astola and Kuosmanen

*correspondence author: Tel. +41-1-6323557, Fax. +41-1-6321211, email. allgower@aut.ee.ethz.ch

(1997)) and appears to be equally effective in removing outliers but introduces less distortion in the nominal part of the data (ideally, none). Thus, under normal operating conditions, the data cleaning filter has no effect on control system performance, but in the presence of outliers, control system performance can improve significantly. The results of applying the data cleaning filter to the flexible shaft data are shown in Fig. 2, which illustrates both its intent and its performance: the objective here is not smoothing (a task better suited to noise filters like those discussed in Hamming (1983)), but rather the automatic replacement of severe outliers with values that are more consistent with the rest of the dataset. We also want to remark at this point that whenever we refer to linear filters we mean linear noise filters.

The simplest version of the data cleaning filter involves only three tuning parameters, and this paper considers both the practical determination of these tuning parameters and an interesting extensions of the basic data cleaning filter, also suitable for real-time applications.

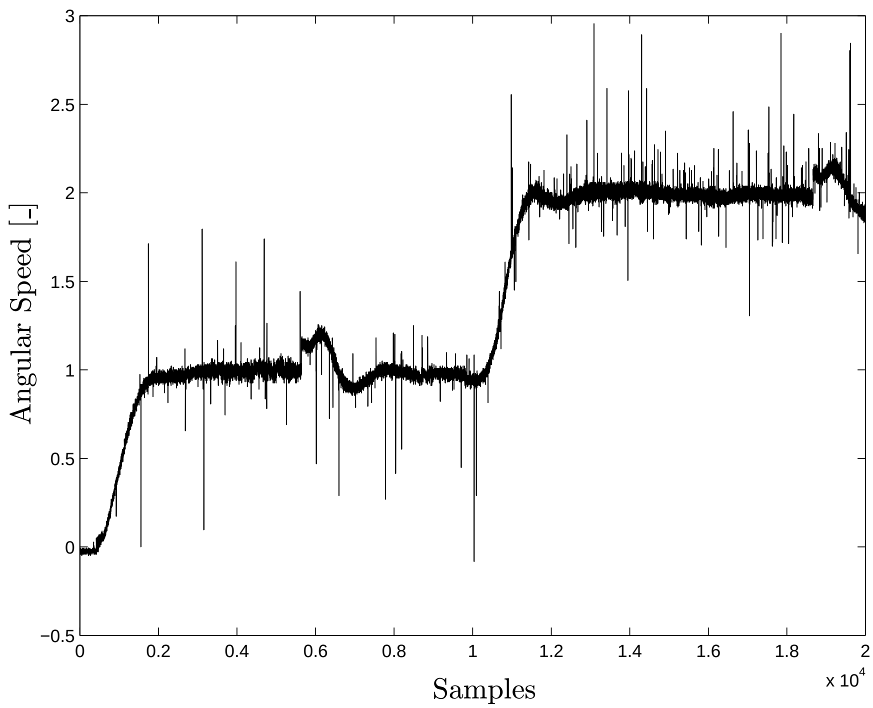

Figure 1: Original shaft speed dataset.

2 Description of the data cleaning filter

We assume that the generation of outliers can be described by the additive outlier model, popular in robust time-series analysis (Denby and Martin (1979); Martin and Yohai (1986)):

$$y_k = x_k + o_k. \tag{1}$$

Here $\{y_k\}$ is the measured data sequence, $\{x_k\}$ is the "nominal" data sequence we are interested in and $\{o_k\}$ represents a sequence of contaminating outliers. The values of the sequence $\{o_k\}$ are assumed to be zero except for a few time instances when the magnitude of $\{o_k\}$ is "large"

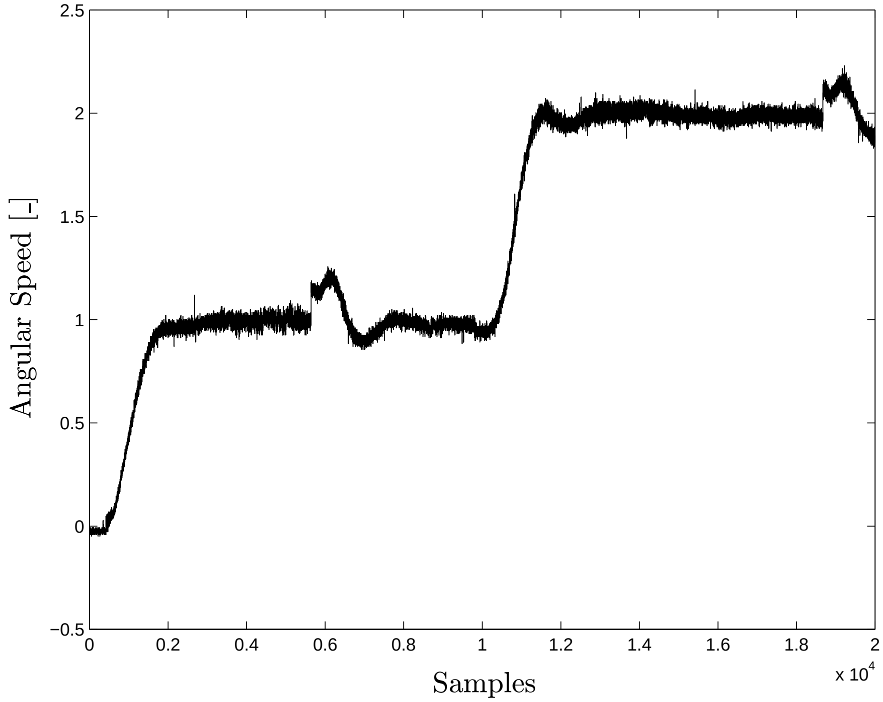

Figure 2: Filtered shaft speed dataset.

relative to the nominal variation seen in the data. We seek an approximation of x_k based only on the current and past data observations y_{k-j} for $k \geq j \geq 0$, in contrast to the non-causal (off–line) smoother described previously in Ling *et al.* (1984); Astola and Kuosmanen (1997) that also assume knowledge of future values, i.e. y_{k+j} for $j > 0$. Specifically, y_k and the past $N-1$ data values y_{k-j} are stored in a data window W_k of width N:

$$W_k = \{y_{k-N+1}, y_{k-N+2}, \ldots, y_k\}. \tag{2}$$

The data values in this window are rank–ordered to obtain

$$R_k = \{y^k_{(1)} \leq y^k_{(2)} \leq \cdots \leq y^k_{(N)}\}, \tag{3}$$

and the median $y_k^\dagger$ of the sequence R_k is computed as:

$$y_k^\dagger = \begin{cases} y^k_{((N+1)/2)} & \text{for } N \text{ odd} \\ (y^k_{(N/2)} + y^k_{(N/2+1)})/2 & \text{for } N \text{ even.} \end{cases} \tag{4}$$

The median value $y_k^\dagger$ provides a nominal reference against which the current data point y_k is evaluated; specifically, define the distance d_k between y_k and $y_k^\dagger$:

$$d_k = |y_k^\dagger - y_k|. \tag{5}$$

If this distance exceeds some specified threshold $T_k \geq 0$, we declare y_k to be an outlier and replace it with a prediction $y_k^\star$ to obtain the filtered data sequence $\{f_k\}$ with:

$$f_k = \begin{cases} y_k & \text{if } d_k \leq T_k \\ y_k^\star & \text{if } d_k > T_k. \end{cases} \tag{6}$$

To see the possible range of behavior of the data cleaning filter, it is useful to consider the two extreme limits, $T_k = 0$ and $T_k = +\infty$. In the first case, the outlier selection criterion is always satisfied, so the filter output is simply $f_k = y_k^\star$; this limit corresponds to the most aggressive data filtering since the sequence y_k is always modified, in general. Conversely, if $T_k = +\infty$, the outlier selection criterion can never be satisfied and $f_k = y_k$; in other words, no filtering is performed in this limit. If an intermediate threshold value T_k can be chosen appropriately and if y_k does not correspond to an anomalous data point, the data sequence is unmodified ($f_k = y_k = x_k$), but if an outlier occurs at time k, then y_k will lie far from the median value in the data window and will be replaced with the prediction $y_k^\star$. Thus, under normal operating conditions, the data cleaning filter does not alter the measured sequence and thus has no effect on control system performance. In the presence of outliers, these will be removed and therefore the control system performance can improve significantly.

This data filtering approach corresponds to the class of *decision-based filters* discussed in Astola and Kuosmanen (1997) and apparently first proposed in the image processing literature (Ling *et al.* (1984)). Application of the data cleaning filter to off-line process control applications (in particular, empirical model identification) is described in Pearson (1999). For real-time applications, practical implementation of the data cleaning filter requires specification of the following four attributes:

1. the algorithm for determining the value of the filter output $y_k^\star$ with which outliers are replaced,
2. the threshold T_k (which can depend on the time index k),
3. the window width N, and
4. filter initialization (i.e., how f_k is defined for $k < N$).

These issues are discussed in detail in the following sections, after which filter performance is examined for three examples, one based on simulations for which exact results are known and two based on real datasets.

3 Outlier replacement strategies

Many different ways to determine the prediction $y_k^\star$ are possible. One of the simplest and analytically most interesting possibilities is *median replacement*, for which $y_k^\star = y_k^\dagger$. In this case, the maximally aggressive data cleaning filter obtained by taking $T_k = 0$ for all k corresponds to the causal median filter, discussed in some detail in the next section. Analogously, other possible replacement strategies would be to take $y_k^\star$ to be the output of any other filter — linear or nonlinear — based on the data in the window W_k; in particular, any of the nonlinear filters described in Astola and Kuosmanen (1997) could be considered here. In general, the disadvantage of linear filters is that if any of the other data points in the window W_k are outliers, the response of the linear filter is likely to be a poor prediction of the uncontaminated data value x_k at time k. In particular, it is important to note that successive outlier patches occur in some applications, and linear predictions $y_k^\star$ can be expected to perform poorly in such cases. The possibility of patchy outliers is also an important consideration with respect to window width selection and is discussed further in connection with that problem.

Another particularly interesting replacement strategy is to replace anomalous values with the last valid data point. Specifically, define $y_k^\star = y_{k-j}$ where j is the smallest integer such that $|y_{k-j} - y_k^\dagger| \leq T_k$. This replacement strategy sometimes yields much better results than the median replacement strategy, as the following simple example illustrates. Suppose the uncontaminated sequence $\{x_k\}$ is monotonically increasing, from which it follows that

$$x_k \geq x_{k-i} \geq x_k^\dagger, \tag{7}$$

for all $i < \frac{N+1}{2}$ (assume N odd for simplicity). In this case, if the last valid data point occurs for $j < \frac{N+1}{2}$, it follows that x_{k-j} is a better estimate of x_k than $y_k^\dagger = x_k^\dagger$. Conversely, there are sequences for which median replacement provides better results than the last valid point strategy described here. For example, if x_k is a periodic sequence with period 2 (i.e., a binary oscillation), median replacement is a better strategy. This replacement strategy is also the one used throughout the rest of the paper especially in the examples in Section 9 and 10.

4 The causal median filter

Because it is the most aggressive data cleaning filter with median replacement, consideration of this special case can provide some useful insights into the general behavior of this data cleaning filter. Further, many results are available for the *off-line* (non–causal) median filter, generally considered in the symmetric form (Gallagher and Wise (1981)). In particular, the output m_k of the symmetric median filter is the median value from the $2H+1$-point moving data window:

$$W_{M_k} = \{y_{k-H}, \ldots, y_{k-1}, y_k, y_{k+1}, \ldots, y_{k+H}\}. \tag{8}$$

Characterizations are available for the *root sequences* of this median filter, defined as those sequences $\{r_k\}$ that are invariant under the application of the filter. In particular, following Astola *et al.* (1987), define the following sequence features:

1. A *constant neighborhood* is a region of at least $H + 1$ consecutive data points with the same value.

2. An *edge* is a monotonically increasing or decreasing sequence of points surrounded on both sides by constant neighborhoods not destroying the monotonicity.

It can be shown (Gallagher and Wise (1981)) that a sequence of finite length can only be a root sequence for the non–causal median if it consists entirely of constant neighborhoods and edges. In the case of infinitely long sequences, periodic binary sequences can also be root sequences of a median filter; the period of these root sequences depends on the window width of the filter (see Astola *et al.* (1987)).

It is clear from the definitions that, if $N = 2H+1$, the output y_k of the real-time median filter considered here corresponds to the symmetric median filter delayed by H samples: $y_k^\dagger = m_{k-H}$. The following results follow immediately from this observation.

Proposition 4.1 *If $\{r_k\}$ is a root sequence for the symmetric median filter of width $N = 2H+1$, the response of the causal median filter is $y_k^\dagger = r_{k-H}$.*

Proposition 4.2 *If $\{r_k\}$ is a root of both the symmetric median filter of width $N = 2H+1$ and the causal median filter, it is periodic with period H.*

This result, together with the characterization of finite length root sequences for the symmetric median filter noted above lead immediately to the following result:

Proposition 4.3 *The only finite-length sequences $\{r_k\}$ that are roots of both the symmetric median filter of width $N = 2H + 1$ and the causal median filter are constant, i.e. $r_k = c$ for all k.*

The significance of this last result follows from the fact that the class of finite root sequences for the median filter is rather large and includes many "nice" sequences that we might wish to preserve unmodified in the process of outlier removal. In particular, the class of symmetric median filter root sequences includes all finite length monotone sequences. This observation immediately implies the following result:

Corollary 4.1 *The only monotonic root sequences for the causal median filter are constant.*

Conversely, non-constant infinite-length root sequences do exist for the causal median filter:

Theorem 4.1 *Any period 2 sequence $\{s_k\}$ is a root sequence for any causal median filter of odd width $N = 2H + 1$.*

Proof 4.1 *If $\{s_k\}$ is periodic with period 2, it assumes only two distinct values, one for k even and the other for k odd. Since the data window W_k contains an odd number of points, it will contain $H + 1$ points with value s_k and H points with value s_{k-1}. Hence, the median value in the data window is s_k and the output of the median filter is simply $y_k^\dagger = s_k$.*

The practical consequences of these results are that only very special sequences of rather limited interest are invariant under the application of the real-time median filter. Hence, the causal median filter does *not* appear to be a particularly good data cleaner for real-time control applications: at best, it introduces a delay of H samples with its attendant phase shift, and at worst, it distorts the nominal data sequence we wish to preserve. This point is seen clearly in Fig. 3, which shows a detailed view of a portion of one of the helicopter data sequences discussed later in this paper, overlaid with the response of the causal median filter; in particular, note both the successful outlier rejection at $k \simeq 120$ and the significant distortions of the nominal data variation indicated with arrows. Consequently, if we consider data cleaning filters based on median replacement (i.e., $y_k^\star = y_k^\dagger$), it is important not to make the threshold values T_k too small in order to avoid these difficulties. These results stand in marked contrast to those for the symmetric median filter, widely used in signal processing applications in part because its associated root sequence class includes so many useful sequences (e.g., edges in image processing applications). Conversely, it is also important not to make these threshold values too large since the filter then offers no outlier protection at all.

5 Threshold selection

Ideally, we would like all nominal data sequences $\{x_k\}$ to be a root sequence of the data cleaning filter, and it is clear from the results presented in the previous section that, for median replacement, this behavior is only possible if T_k is chosen large enough. The following discussion describes several different strategies for threshold specification and presents some detailed recommendations, based on different assumptions concerning the behavior of the nominal data sequence and the outliers. One basic approach is to use pre-defined thresholds T_k, specified on the basis of knowledge about the measurements and the system generating the data but independent of the data sequence $\{y_k\}$. Alternatively, it is also possible to use thresholds T_k that are computed from the data sequence. The following discussion considers both approaches and concludes with some ideas on how to combine them.

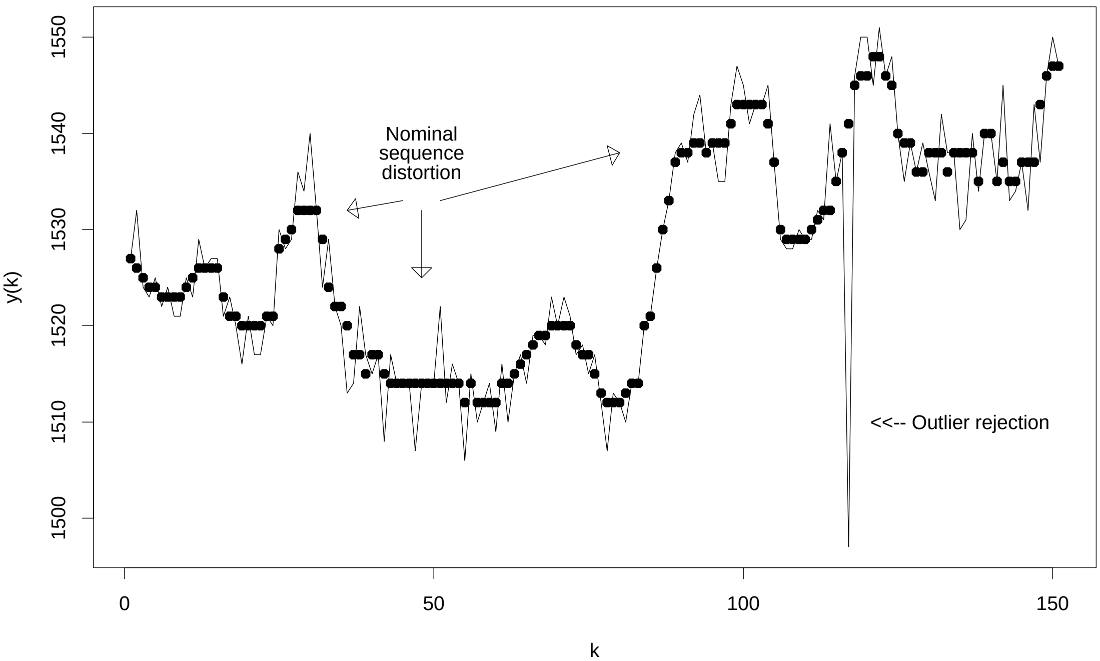

Figure 3: Helicopter data and causal median filter response (indicated by thick dots).

Fixed thresholds

In principle, the simplest approach to threshold selection is to choose a fixed value of T_k based on general knowledge of the "typical" variation of the data sequence. For example, given knowledge of the time between measurements and the dominant dynamics of the physical system, it may be possible to establish limits on the range of measurement variation we could reasonably expect to observe in N successive samples. Choosing T_k somewhat larger than this value could then provide a basis for rejecting outlying observations. The difficulty with this approach is that it rests on certain assumptions about the nominal process changes (e.g., variations in operating conditions, external disturbances, etc.) that are responsible for the observed data variations. Unexpectedly large changes in these operating conditions or disturbances can result in large changes in measured responses that are *not* measurement anomalies, but rather legitimate process responses to which the control system should respond. For this reason, it appears to be more desirable to allow T_k to depend on some reliable estimate of the nominal variation seen in the measurement data. One effective implementation of this idea is the MAD-based data cleaning filter described next.

MAD scale based threshold

The basic philosophy behind using data-dependent thresholds is to first estimate the range of variation of the nominal data sequence $\{x_k\}$ and then assess each observation y_k with respect to the median $y_k^\dagger$ (taken as representative of the center of the nominal data sequence) and this estimated range of variation. Almost certainly, the best known estimate of variability is the standard deviation, but this range estimate is unsuitable because it is highly susceptible to outliers itself (Huber (1981); Davies and Gather (1993); Astola and Kuosmanen (1997)). A better alternative is the *median absolute deviation* (MAD), defined as follows for the data window W_k. First, compute the median $y_k^\dagger$ and the distances d_k of each data point from this reference value as in Eqs. (4) and (5). Next, rank-order these distances to obtain the sequence

$$d_{(1)} \leq d_{(2)} \leq \cdots \leq d_{(N)}. \tag{9}$$

The (un-normalized) MAD scale estimate S_k is defined as the median absolute deviation from the median, which is simply the median of this list, determined analogously to Eq. (4). This scale estimate is often normalized to $\tilde{S}_k = S_k/0.6745 \simeq 1.4826 S_k$ to make it an unbiased estimate of the standard deviation for Gaussian data (Huber (1981)). The key point, however, is that the MAD scale estimate is *much* more robust with respect to outlier contamination than the usual standard deviation estimate, which is inflated badly enough by the presence of outliers that simple rejection rules like "declare y_k suspicious if it lies more than 3 standard deviations from the mean" frequently fail to detect *any* outliers if more than one are present in the sample.

One of the principal outlier detection strategies considered in this paper is the *Hampel identifier* (Davies and Gather (1993)), obtained by setting

$$T_k = cS_k \tag{10}$$

for some constant $c \in \mathcal{R}^+$, chosen independent of the data in the window W_k. To see how this constant should be chosen, it is useful to first consider some general properties of the MAD scale estimate S_k. For convenience, we will assume the window width is an odd integer and write it as $N = 2H + 1$; qualitatively analogous results would be obtained for even window widths, with some minor differences in detail and resulting in somewhat more complex expressions, as in Eq. (4). Under this assumption, it follows that $S_k = d_{(H+1)}$, implying that $H+1$ of $2H+1$ data points (i.e., approximately 50% of the data) lie in the interval $[y_k^\dagger - S_k, y_k^\dagger + S_k]$. Further, note that in the notation of Eq. (3), the median is $y_k^\dagger = y_{(H+1)}$; hence, $H+1$ data points (including the median) lie in each of the intervals

$$y_{(1)} \leq y_{(2)} \leq \cdots \leq y_{(H)} \leq y_{(H+1)}$$

and

$$y_{(H+1)} \leq y_{(H+2)} \leq \cdots \leq y_{(2H)} \leq y_{(2H+1)}.$$

Consequently, it follows that $d_{(H+1)}$ cannot exceed the smaller of the two subinterval widths $y_{(H+1)} - y_{(1)}$ and $y_{2H+1} - y_{(H+1)}$. Further, since the minimum of two numbers is a lower bound on their average, this observation implies

$$0 \leq S_k \leq \min\{y_{(H+1)} - y_{(1)}, y_{(2H+1)} - y_{(H+1)}\} \leq \frac{y_{(2H+1)} - y_{(1)}}{2}. \tag{11}$$

It is useful to note that both of these limits are achievable: if $H+1$ data points have exactly the same value y_k, it follows that this data value is equal to the median $y_k^\dagger$ and $S_k = 0$; conversely, if the smallest data value $y_{(1)}$ and the largest data value $y_{(2H+1)}$ are both repeated H times and $y_{(H+1)}$ is their arithmetic average, S_k achieves the upper bound.

With respect to outlier detection and replacement, both of these limits of the MAD scale estimator are undesirable, and both can occur in practice. In particular, quantization due to A/D conversion or limited precision data storage (e.g., temperatures recorded only to the nearest tenth of a degree) can result in different data points within W_k having exactly the same value. If this number exceeds H, the MAD scale estimate becomes zero and the thresholds defined in Eq. (10) also go to zero, implying $f_k = y_k^\star$ for *any* y_k that differs from the median. To deal with this difficulty, various alternatives (Astola and Kuosmanen (1997)) and corrections (Rousseeuw and Leroy (1987)) have been proposed. Here, we advocate the use of a fixed lower bound, discussed further at the end of this section. Conversely, note that if S_k achieves the upper bound in Eq. (11) and $c \geq 1$ in Eq. (10), no outlier rejection occurs regardless of the range of the data. In either case, note that these difficulties are more likely to occur the smaller the data window width is taken.

More generally, if we adopt the MAD-based threshold defined in Eq. (10), the following two theorems provide useful guidance in selecting the constant c. The fundamental assumption underlying both of these theorems is that the *nominal* data sequence $\{x_k\}$ satisfies the conditions stated in the theorem. The results of the theorems provide lower limits for the constant c for which these nominal data sequences are invariant under the data cleaning filter.

Theorem 5.1 *Any monotonic sequence $\{x_k\}$ satisfying the growth rate restriction*

$$|x_{i+2} - x_{i+1}| \leq m|x_{i+1} - x_i| \text{ for some } m \in [0,1] \text{ and } \forall i \in \mathcal{N} \tag{12}$$

is invariant under the data cleaning filter of width $N = 4H + 1$ provided $c \geq 1 + m^H$.

Proof 5.1 *We prove the theorem for monotonically increasing sequences; the extension to monotonically decreasing sequences is immediate. By monotonicity, it follows that the median is*

$$x_k^\dagger = x_{k-2H}$$

and the basis for the proof is to show that

$$\hat{S}_k \stackrel{def}{=} x_{k-H} - x_k^\dagger = x_{k-H} - x_{k-2H}$$

is a lower bound for the MAD scale estimate S_k for any sequence satisfying Eq. (12). It follows from condition (12) that for any $\ell > j$,

$$\begin{aligned} x_{k-j} - x_{k-\ell} &= (x_{k-j} - x_{k-j-1}) + \cdots + (x_{k-\ell+1} - x_{k-\ell}) \\ &\leq m(x_{k-j-1} - x_{k-j-2}) + \cdots + m(x_{k-\ell} - x_{k-\ell-1}) = m(x_{k-j-1} - x_{k-\ell-1}). \end{aligned}$$

From this inequality, it follows that

$$\begin{aligned} x_{k-j} - x_{k-\ell} &\leq m^2[x_{k-j-2} - x_{k-\ell-2}] \\ &\leq \cdots \\ &\leq m^r[x_{k-j-r} - x_{k-\ell-r}] \end{aligned}$$

for any $r > 0$. In particular, we have the result

$$x_{k-H} - \underbrace{x_{k-2H}}_{=x_k^\dagger} \leq m^H(\underbrace{x_{k-2H}}_{=x_k^\dagger} - x_{k-3H}).$$

Since $0 \leq m \leq 1$ and thus $m^H \leq 1$, it follows that

$$\hat{S}_k = x_{k-H} - x_k^\dagger \leq m^H(x_k^\dagger - x_{k-3H}),$$

implying that no more than $2H+1$ *of the* $4H+1$ *points in the data window lie in the interval* $[x_k^\dagger - \hat{S}_k, x_k^\dagger + \hat{S}_k]$*, thus establishing the claim that* $\hat{S}_k \leq S_k$*.*

In the context of the data cleaning filter, the sequence $\{x_k\}$ *will be invariant provided*

$$x_k - x_k^\dagger \leq cS_k. \tag{13}$$

To see that this condition is satisfied, note that

$$\begin{aligned} x_k - x_k^\dagger &= x_k - x_{k-H} + x_{k-H} - x_k^\dagger \\ &\leq (1+m^H)(\underbrace{x_{k-H} - x_k^\dagger}_{=\hat{S}_k}). \end{aligned}$$

Since $\hat{S}_k \leq S_k$*, it follows that Eq. (13) is satisfied if* $c \geq 1 + m^H$ *and the proof is complete.*

This theorem establishes for example a lower bound for the threshold parameter to avoid introducing any distortion into nominal sequences like monotonic step responses of first order linear systems that settle out to constant limits. The following theorem establishes similar threshold limits for *sector bounded* sequences like the one shown in Fig. 4. Specifically, we define the following two types of sequences:

Definition 5.1 *A sequence of type I satisfies the following four conditions*

$$\begin{aligned} x_{k-2H} &= x_k^\dagger && \qquad (14)\\ 0 &< c_1 \leq c_2 < \infty && \qquad (15)\\ c_1(2H-i) &\leq x_{k-i} - x_k^\dagger \leq c_2(2H-i) \quad \forall \quad i \leq 2H && \qquad (16)\\ c_1(2H-i) &\geq x_{k-i} - x_k^\dagger \geq c_2(2H-i) \quad \forall \quad i > 2H && \qquad (17) \end{aligned}$$

for all k*.*

Definition 5.2 *A sequence of type II satisfies the following four conditions*

$$\begin{aligned} x_{k-2H} &= x_k^\dagger && \qquad (18)\\ 0 &> c_1 \geq c_2 > -\infty && \qquad (19)\\ c_1(i-2H) &\geq x_{k-i} - x_k^\dagger \geq c_2(i-2H) \quad \forall \quad i \leq 2H && \qquad (20)\\ c_1(i-2H) &\leq x_{k-i} - x_k^\dagger \leq c_2(i-2H) \quad \forall \quad i > 2H && \qquad (21) \end{aligned}$$

for all k*.*

Fig. 4 shows a (part of a) sequence of type I.Eq. (16) and (17) simply define the sector bounds for sequences of type I. The same holds for Eq. (20) and (21) for sequences of type II.

Theorem 5.2 *Any sequence* $\{x_k\}$ *of type I or II is invariant under the MAD-based data cleaning filter of width* $N = 4H+1$ *if* $c > 2c_2/c_1$*.*

Proof 5.2 *We prove the theorem for type I sequences, noting that the extension to type II signals is immediate. First, note that* $\{x_k\}$ *is invariant under the MAD-based data cleaning filter if and only if*

$$x_k - x_k^\dagger \leq cS_k, \tag{22}$$

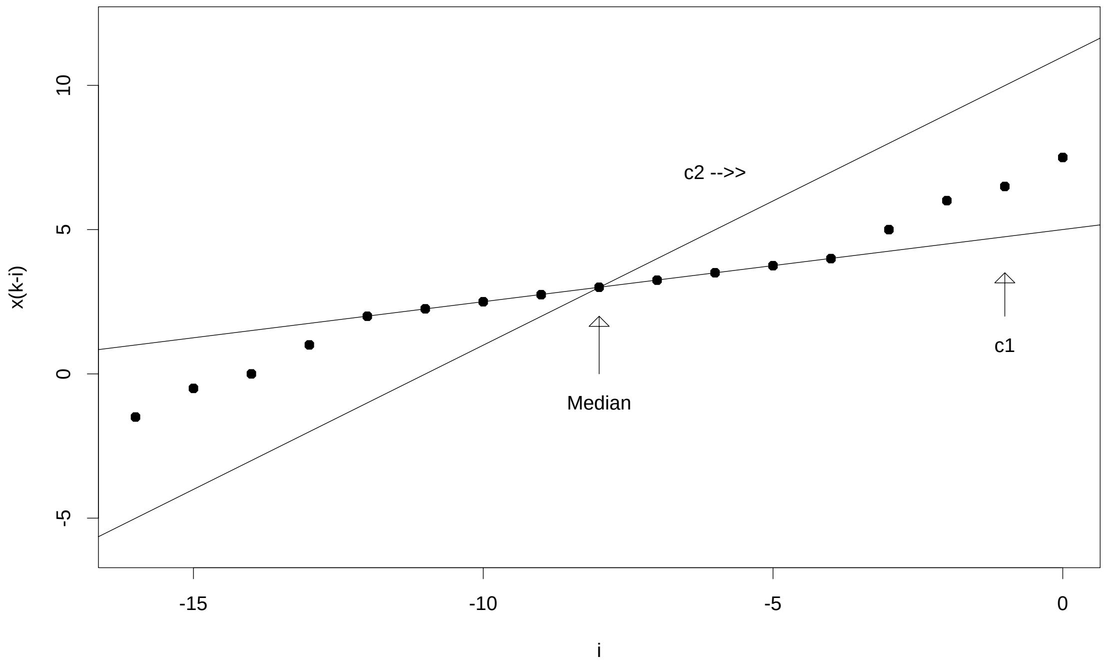

Figure 4: A sector-bounded, type I sequence $\{x_k\}$. The dots represent data points.

implying the theorem holds if c is chosen so that this condition is satisfied for the largest possible deviation $x_k - x_k^\dagger$ *and the smallest possible value of* S_k. *For a type I sequence, the largest possible deviation is* $x_k - x_k^\dagger = 2Hc_2$, *and the smallest possible value for* S_k *occurs when* $2H+1$ *points are as close to* $x_k^\dagger$ *as possible. This closest possible configuration occurs when the points* x_{k-3H} *through* x_{k-H} *lie on the line with the smallest possible slope (i.e.,* c_1*) passing through the median* $x_k^\dagger = x_{k-2H}$. *(This configuration is shown in Fig. 4 for* $H = 4$.*) The median distance* d_{k-i} *determined by this point configuration is* $S_k = d_{k-H} = d_{k-3H} = Hc_1$. *Thus, condition (22) will be satisfied if* $c > (2Hc_2)/(Hc_1) = 2c_2/c_1$.

Corollary 5.1 *If a subsequence of width* $4H+1$ *of a sequence* $\{x_k\}$ *satisfies the four conditions of a type I or II sequence then the point* x_k *(the last point of the subsequence) is unchanged by the MAD based data cleaning filter of width* $N = 4H+1$ *if* $c > 2c_2/c_1$.

Note that condition (14) in the definition of a type I sequence implies that $\{x_k\}$ is a root sequence for the symmetric (non–causal) median filter of width $4H+1$. Analogously, condition (18) implies that any type II sequence is also a root sequence for this symmetric (non–causal) median filter. In particular, note that any strictly increasing sequence $\{x_k\}$ satisfying

$$c_1 \leq |x_k - x_{k-1}| \leq c_2$$

is a type I sequence for arbitrary H and any strictly decreasing sequence satisfying this condition is a type II sequence. In practical terms, Theorem 5.2 establishes restrictions on the large class of well-behaved root sequences for the off-line median filter that are sufficient to permit these sequences to also be invariant under the causal MAD-based data cleaning filter, provided the threshold parameter c is taken large enough. Note that this result highlights the fundamental trade-off between the requirements of off-line data cleaning with the median filter where arbitrary monotonic sequences are invariant and arbitrary amplitude impulses are eliminated, and the requirements of on-line data cleaning where only certain monotonic sequences are invariant and only impulses of sufficiently large amplitude are eliminated.

As another interesting corollary to Theorem 5.2, note that any line with positive slope c_1 satisfies the type I conditions for $c_2 = c_1$ and any line with negative slope satisfies the corresponding type II conditions. Consequently, the MAD-based data cleaning filter will leave straight lines invariant if and only if $c \geq 2$. Also, note that if the normalized MAD scale estimate $1.4826S_k$ is used as a robust replacement for the standard deviation estimate in the "3σ edit rule," this choice would correspond to taking $c \simeq 4.45$. Again, it is important to emphasize the inherent trade-off here: the larger we take c, the less likely the MAD-based data cleaning filter will distort the nominal data sequence $\{x_k\}$, but if c is taken too large the filter will cease to be effective in detecting and rejecting outliers. Probably the simplest way to choose a reasonable value for c is to examine both nominal and contaminated data subsequences; in particular, if nominal data subsequences of type I or II can be identified that play an important role in the whole data sequence, the results of Theorem 5.2 can be used to establish practical lower bounds for c.

Combining both approaches

Probably the greatest practical difficulty with the MAD-based data cleaning filter is that the MAD scale estimate can assume the value 0 for quantized data sequences containing $H + 1$ identical values in a data window of width $2H + 1$. For example, if the nominal data sequence $\{x_k\}$ is a step of arbitrary amplitude, it cannot be invariant under the MAD-based causal data cleaning filter for any choice of c since $S_k = 0$ for a step sequence. As noted earlier, other robust scale estimators are available that do not suffer this limitation and one possible alternative is to use with one of these other estimators instead of the MAD scale estimator. Conversely, another possibility is to combine the MAD scale estimator with a fixed *lower bound* for T_k, to obtain the threshold:

$$T_k = \max\{cS_k, T_{kmin}\}. \tag{23}$$

Here S_k is the MAD scale estimate, c is the constant discussed at length in the previous section, and $T_{kmin} > 0$ is a fixed minimum value for T_k. In practice, this minimum value could be chosen from knowledge of the measurement noise level or analog-to-digital converter resolution (e.g., choose T_{kmin} such that changes less than m least significant bits cannot be rejected as outliers). Note that under this criterion, steps up to amplitude T_{kmin} are invariant under the data cleaning filter, but impulses of this amplitude are also invariant, so this minimum value should not be chosen too large. This strategy was used in the examples discussed in Section 9 and 10 of this paper.

6 The window–width

At least three factors influence the choice of the width N of the data window W_k on which the data cleaning filter is based. First is the fact that both the median $y_k^\dagger$ and the MAD scale estimate S_k become less variable as N increases. Although this observation might appear to argue in favor of large data windows, it is important to remember that outliers in the dynamic applications considered here correspond to violations of *local* variation patterns seen in the data. For example, the change from approximately 1.0 to approximately 2.0 over the $2,000$ point

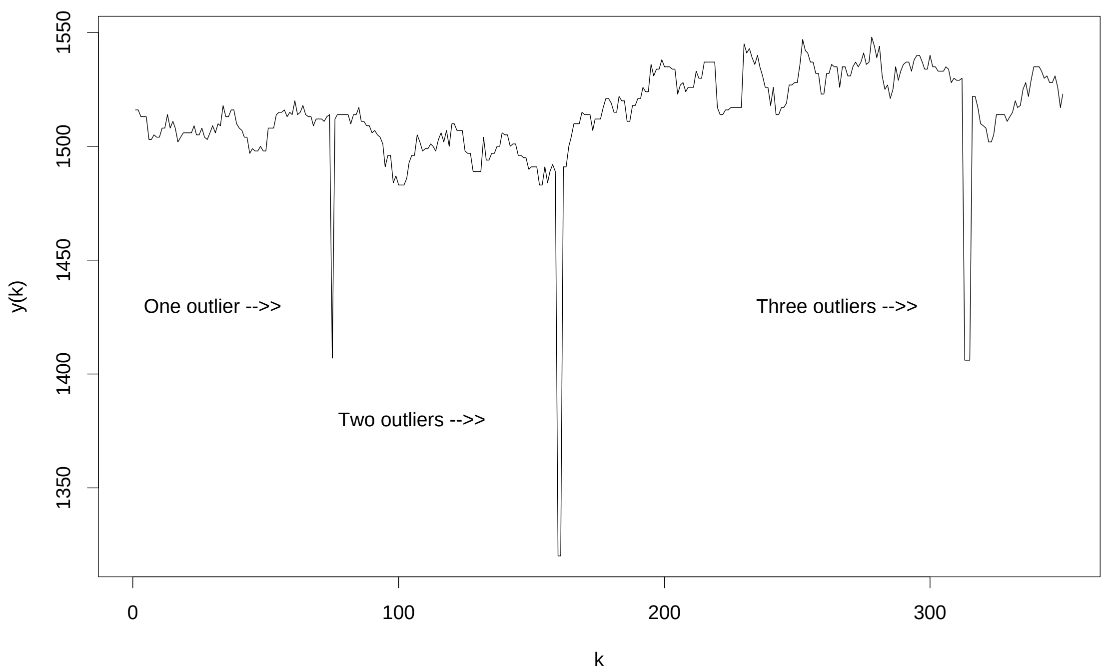

Figure 5: Isolated and patchy outliers.

interval from $k \simeq 1.0 \times 10^4$ to $k \simeq 1.2 \times 10^4$ in Fig. 1 is not anomalous, but the change just before $k \simeq 1.0 \times 10^4$ from approximately 1.0 to approximately 0 that occurs over just a few samples clearly is anomalous. Consequently, it is important to choose N small enough that the data window W_k gives a measure of the local dynamics on a time scale that is appropriate to the problem under consideration.

Conversely, it is important not to take N too small, either. In particular, note that successive patches of anomalous values can and do arise, as illustrated in Fig. 5 which shows another segment of one of the helicopter datasets discussed later in this paper. In this example, an isolated outlier occurs at $k \simeq 75$, a patch of two successive outliers occurs at $k \simeq 160$, and a patch of three successive outliers occurs at $k \simeq 315$. Note that, despite their resistance to the effects of outliers, both the median and the MAD scale estimate will be completely determined by a patch of $H + 1$ successive outliers of identical or similar value in a data cleaning filter of width $N = 2H + 1$. Consequently, if patchy outliers are possible in a particular application, it is important to choose the window width N larger than twice the width of the widest anticipated outlier patch.

7 Filter initialization

Here, we consider sequences $\{y_k\}$ defined for $k \geq 1$; therefore, because it contains $N - 1$ past data values, the window W_k is not well defined for $k < N$. However, practical implementation of a real-time data cleaning filter requires the filter output f_k to be defined for all k. One possible approach is to simply take $f_k = y_k$ for $k < N$; although this approach is easily implemented, it has the disadvantage that it offers no protection against outliers in the first $N - 1$ points of the data sequence y_k.

A second possible filter initialization is to define $y_{k-N} = y_1$ for $k < N$, making W_k well-defined for all k. This approach is frequently adopted in median filter applications (Gallagher and Wise (1981)) and is the approach used in the examples in Section 9 and 10. The disadvantage of this method is that if the first data point happens to be an outlier, it is likely to be replicated under this initialization scheme. Specifically, note that for $k \leq N$, the N point data window W_k contains $N + 1 - k$ copies of y_1 and the $k - 1$ additional data values y_2 through y_k. If $k < (N+2)/2$, it follows that the majority of the values in W_k are y_1, implying that the median value is $y_k^\dagger = y_1$. If $|y_k - y_k^\dagger| > T_k$, it follows that the good data value y_k will be replaced with $y_k^\star$; in the median replacement strategy considered here, the anomalous data value y_1 will be replicated approximately $N/2$ times. A possibly useful variation of this filter initialization is to make the threshold value T_k large for $k < N$ to decrease the likelihood of replicating an initial outlier y_1. The threshold T_k could then be decreased monotonically with increasing k until $k = N$.

Finally, a third possible filter initialization would be to replace the N point data window W_k with a window of length k for $k < N$. For $k = 1$, $W_k = \{y_1\}$ so $f_1 = y_1$ under the median replacement scheme considered here. Similarly, for $k = 2$, $W_k = \{y_1, y_2\}$, the median $y_2^\dagger$ is the average of these values and the MAD scale estimate is $|y_2 - y_1|/2$; if $c \geq 1$, it then follows that $f_2 = y_2$. For $3 \leq k < N$, the filter output f_k is simply the response of the k-point data cleaning filter with the threshold parameter T_k determined the same way as for $k \geq N$.

8 Recursive filters

An interesting extension of the symmetric median filter is the *recursive median filter*, obtained by replacing the data window W_{M_k} defined in Eq. (8) with the recursive window:

$$W_{R_k} = \{f_{k-H}, \ldots, f_{k-1}, y_k, y_{k+1}, \ldots, y_{k+H}\}. \tag{24}$$

Here, f_{k-j} represents the previous filter output obtained at time $k - j$. This non–causal off–line filter is described in the paper by Nodes and Gallagher (1982) where a number of interesting characteristics are described. First, it is shown that a sequence $\{r_k\}$ is a root sequence for the recursive median filter if and only if it is also a root sequence for the standard median filter. Secondly, a characteristic feature of the standard median filter is that any finite-length sequence will be reduced to a root sequence of the median filter after a finite number of standard median filter iterations (Gallagher and Wise (1981)); for the recursive median filter defined by Eq. (24), it is shown that any finite length sequence is reduced to a root sequence in one iteration. Finally, it is also noted that for any non-root sequence $\{y_k\}$, the root sequence $\{r_k\}$ obtained by recursive applications of the standard median filter and the root sequence $\{r'_k\}$ obtained by the recursive median filter are generally different.

This idea may also be applied to the causal data cleaning filters considered in this paper. Specifically, define the *recursive causal data cleaning filter* by replacing the data window W_k defined in Eq. (2) with the recursive window:

$$V_k = \{f_{k-2H}, \ldots, f_{k-H-1}, y_{k-H}, y_{k-H+1}, \ldots, y_k\}.$$

Note that this expression defines a data window of width $N = 2H + 1$ that contains H previous filtered data values f_{k-j}, H unfiltered past data values y_{k-j}, and the current data value y_k; for convenience, these N values will be denoted v_{k-j} for $j = 0, 1, \ldots, N$. Subsequent processing is the same as in the nonrecursive data cleaning filter described previously: the median value $v_k^\dagger$ is computed from this data window, distances $\delta_{k-j} = |v_{k-j} - v_k^\dagger|$ from the median are computed, and y_k is declared an outlier and replaced if $\delta_k > T_k$. In principle, any subset of past data values y_{k-j} from the window W_k could be replaced with the filtered data values f_{k-j}, but it is important that no more than H filtered data values be used since if the filter ever generates a sequence of $H+1$ successive outputs with the same value, all subsequent outputs would also have this same value *independent of the unfiltered data in the window* for that case.

9 A simulation example

The following simulation example is presented to provide an assessment of the data cleaning filter in a setting where precise performance evaluations are possible. In contrast to the real data applications considered in the next section, it is possible in this example to distinguish clearly between the fundamental system response of interest, observation noise, and outliers; consequently, it is possible to say precisely how many outliers are detected, how many are missed, and how many non-outlying valid data points are distorted by the different data cleaning strategies considered here. Specifically, this example considers the response of the linear discrete-time system:

$$G(z) = \frac{z - 0.3}{(z-0.4)(z-0.5)(z-0.6)},$$

to the following random step input sequence $\{u(k)\}$. At each time instant k:

1. The input lies in the range $-5 \leq u(k) \leq 5$
2. There is a 10% probability that the input changes (i.e., that $u(k) \neq u(k-1)$)
3. If the input does change, the new value $u(k)$ is drawn from a uniform distribution on the interval $[u(k-1) - 0.25, u(k-1) + 0.25]$, corresponding to $\pm 5\%$ of the total input range.

A typical response of the system to a 10000 point input sequence is shown in Fig. 6. The observed data sequence $\{y(k)\}$ is of the form

$$y(k) = v(k) + e(k) + o(k),$$

where $\{v(k)\}$ is the linear system response shown in Fig. 6, $\{e(k)\}$ represents observation noise, and $\{o(k)\}$ is the contaminating outlier sequence. In this example, the observation noise is an independent, identically distributed sequence of uniform random variables on the interval $[-0.5, 0.5]$ and the outliers are an independent sequence of discrete random variables, assuming the values 0 with 95% probability, +10 with 2.5% probability, and -10 with 2.5% probability. A typical observed sequence is shown in Fig. 7. The results of applying the causal median filter and the outlier filter described in this paper to this observed sequence are shown in Figs. 8 and 9, respectively. Both figures show error plots, i.e. the difference between the filtered and the unfiltered sequence. In both cases, a window width of $N = 7$ was used, and the data cleaning filter used the threshold T_k defined in Eq. (23) with the MAD scale factor $c = 5$ and the lower limit $T_{kmin} = 0.75$.

It is clear from these figures that many of the outliers are found and removed in both cases. More specifically, the observed sequence contains 472 outliers, all of which are removed by the median filter and all but 2 of which are removed by the less aggressive data cleaning filter. Conversely, this data cleaning filter modifies only 2.2% of the valid data points while the median filter modifies 88.6% of the valid points. These results are consistent with the high levels of distortion noted for the causal median filter in the preceeding sections of this paper; clearly,

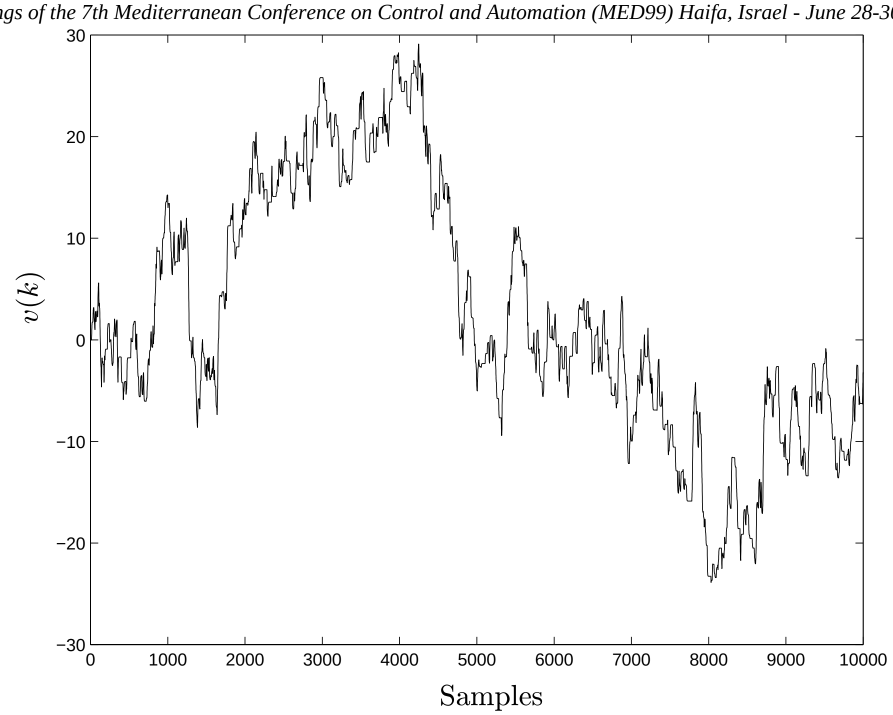

Figure 6: Noise-free simulation response.

the optimal trade-off between distortion of the uncontaminated part of the data sequence and outlier rejection will be application-dependent, but these results again suggest that the causal median filter is likely to be too aggressive in all but the most outlier-sensitive applications.

It was noted in the introduction that the data cleaning filters considered here are intended to reject outliers and are distinct from linear filters typically used for noise suppression (Hamming (1983)). This point is important enough to bear repeating and the following comparisons illustrate two important aspects of this difference. First, Fig. 10 shows the results of applying a linear smoothing filter to the observed data sequence $\{y(k)\}$ and then computing the difference to the nominal system output sequence $\{v(k)\}$. More specifically, the following first-order linear filter was used:

$$\ell(k) = 0.4\ell(k-1) + 0.6y(k). \tag{25}$$

Note that the steady-state gain of this linear filter is 1 and it is typical of linear smoothing filters used in process control applications. It is clear from Fig. 10 that, although the outliers are attenuated somewhat, they are certainly not eliminated; in particular, the performance of this filter with respect to outlier suppression is substantially poorer than that of either the median filter or the data cleaning filter shown in Figs. 8 and 9. What is less clear from Fig. 10 is that in addition to attenuating the outliers somewhat, the linear filter defined by Eq. (25) also broadens them, effectively converting isolated outliers into outlier patches. This observation follows from the fact that isolated outliers may be viewed as impulses, so the response of any linear filter to an outlier-contaminated data sequence will be the sum of the filtered nominal data sequence and a sequence of randomly spaced copies of the filter's impulse response. The requirements for smoothing (i.e., observation noise suppression) and outlier rejection are therefore strongly in conflict: the impulse response should decay "slowly enough" for smoothing but "rapidly enough" to avoid spreading isolated outliers into outlier patches.

The second important point here is that it is possible to gain the advantages of both linear smoothing filters and nonlinear outlier rejection filters by combining them. This point is illus-

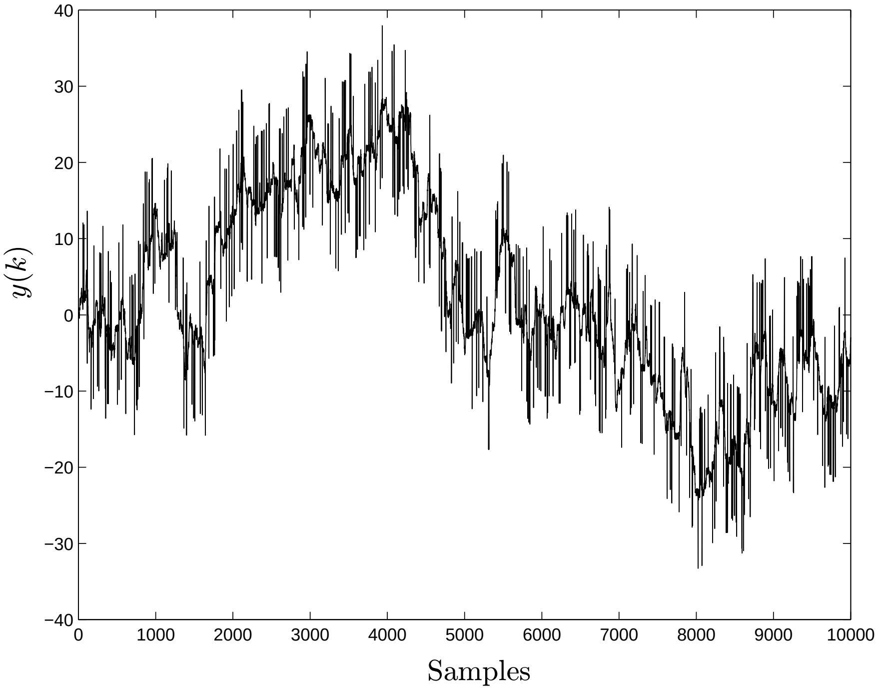

Figure 7: Observed output sequence with noise and outliers.

trated in Fig. 11, which shows the sequence obtained by first applying the data cleaning filter considered previously, then applying the linear filter defined in Eq. (25) and finally computing the difference to the nominal system output sequence $\{v(k)\}$. Because the outlier filter removes almost all of the contamination from the observed data sequence, the task of the linear filter is reduced to the smoothing for which it is well suited. Conversely, it is important to note the order here: nonlinear filtering for outlier removal should always be done *before* linear filtering for noise suppression. In particular, if the order of these operations is reversed, very little benefit from the data cleaning filter can be expected: outliers present in the data sequence will be partially attenuated and broadened by the linear filter, making them harder to detect and remove by the data cleaning filter.

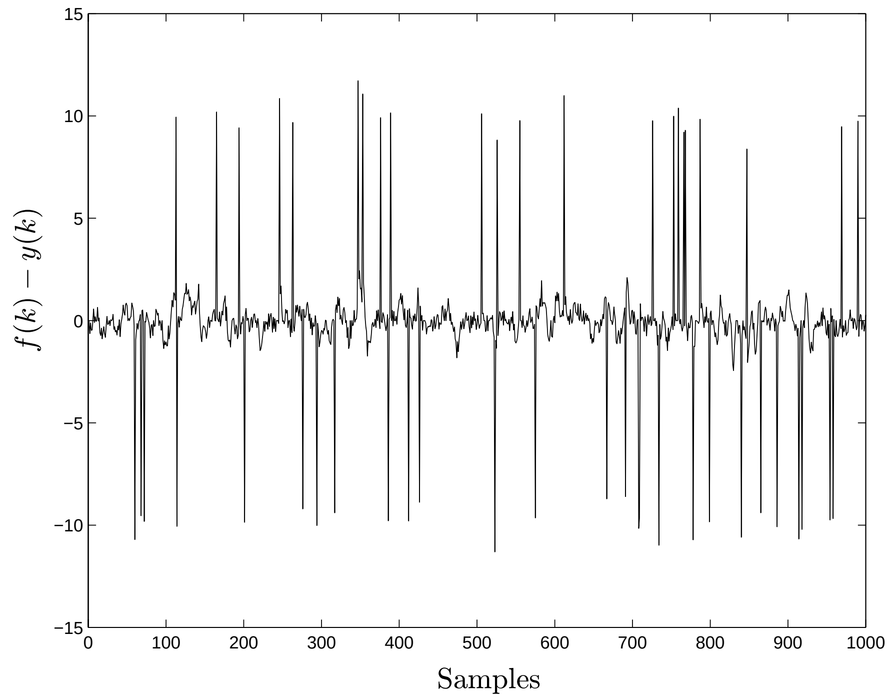

Figure 8: The first 1000 datapoints of the difference between the causal median filter output $f(k)$ and the noisy system output $y(k)$.

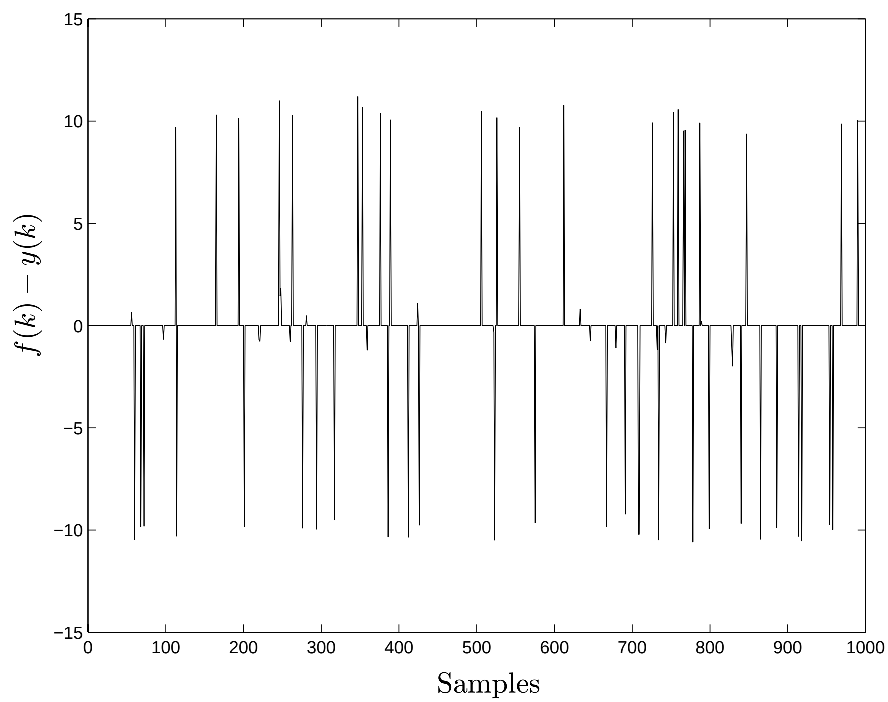

Figure 9: The first 1000 datapoints of the difference between the data cleaning filter output $f(k)$ and the noisy system output $y(k)$.

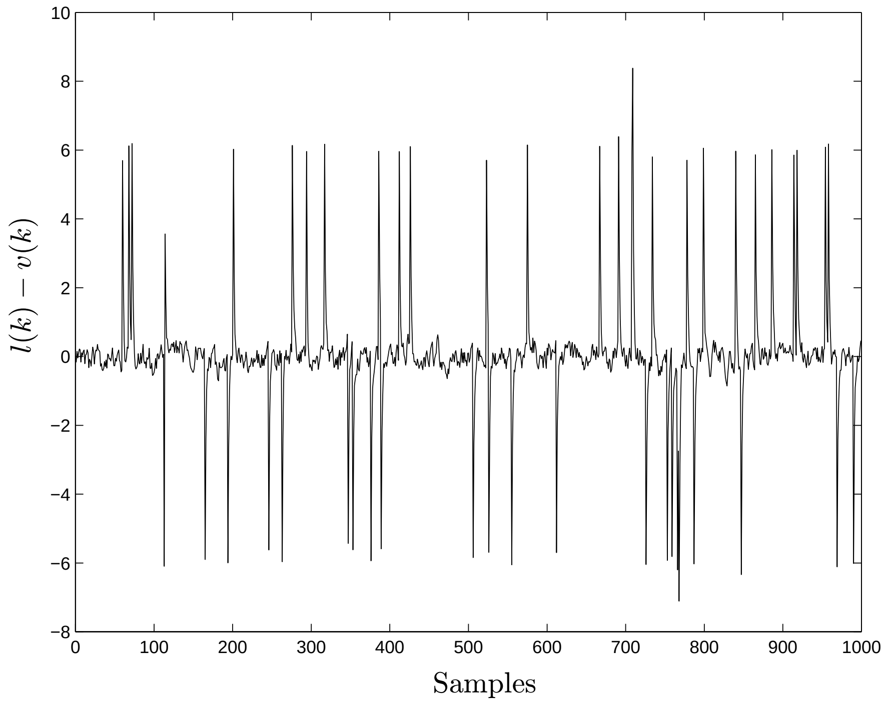

Figure 10: The first 1000 datapoints of the difference between noisy system output $y(k)$ after linear noise filtering with the filter defined by Eq. (25) and the noise free system output $v(k)$.

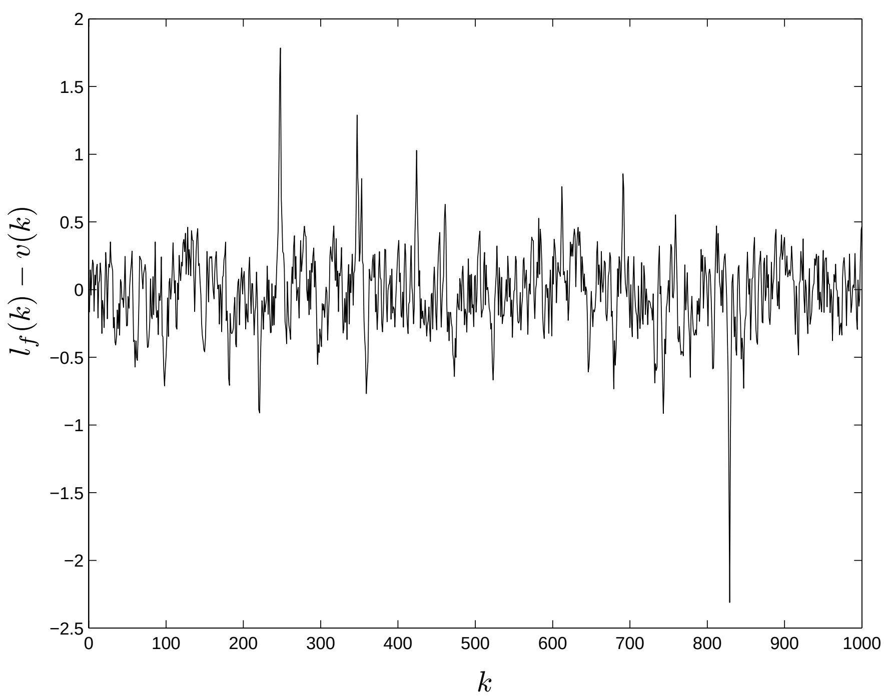

Figure 11: The first 1000 datapoints of the difference between noisy system output $y(k)$ after outlier filtering and afterwards noise filtering with filter (25) and the noise free system output $v(k)$.

10 Two real data examples

Finally, we consider two real data sequences. The first is the shaft speed measurement data sequence[1] shown in Fig. 1 and discussed previously in the introduction. As noted earlier, many outliers are evident in this aplication as can be seen in Fig. 1. This typical data sequence consists of 20000 points. The cleaned data sequence shown in Fig. 2 was obtained using the data cleaning filter described in this paper with window width $N = 9$ and the threshold T_k specified by Eq. (23) with $c = 3$ and $T_{kmin} = 0.08$. The scaling factor c was chosen with the help of Theorem 5.1 and the lower threshold bound T_{kmin} takes the noise level present in the data into considerations. This figure illustrates the effectiveness of the data cleaning filter since all of the large spikes evident in the original data sequence have been removed. For comparison, Fig. 12 shows the results of applying the causal median filter of the same width to this data sequence. Visually, the median filter appears to be more effective in outlier removal here, but as in the previous examples, it is important to consider the question of distortion of the nominal data. Although it is not possible to distinguish “outliers” from “good data points” with certainty in this example, it is useful to again compare the fraction of data points modified by each of these two data cleaning filters. In this example, the data cleaning filter modifies 1.2% of the points in the dataset, compared with 77.1% for the median filter. Although the small-amplitude spikes remaining in Fig. 2 suggest that T_k may be set too large in the data cleaning filter considered in this example, the large fraction of data values changed by the median filter again suggests that taking $T_k = 0$ is probably too aggressive.

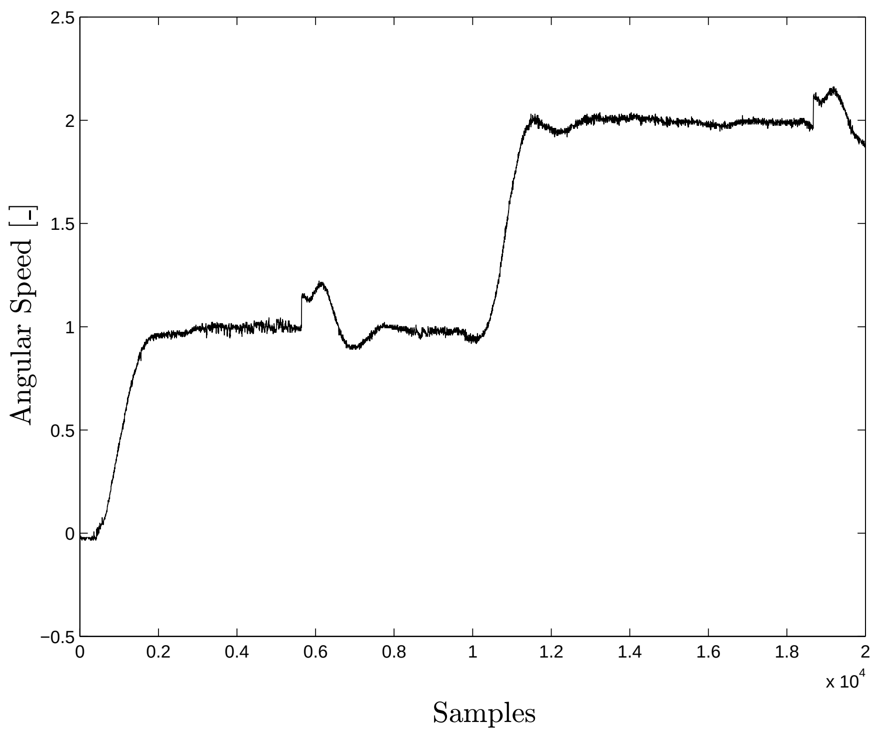

Figure 12: Median filtered shaft speed dataset.

The second real data example[2] considered here comes from a small remote–controlled heli-

[1] The dataset was provided by D. Farruggio from a flexible mechanical shaft used as a student laboratory experiment at the Automatic Control Laboratory at ETH, Zürich.

[2] The data was provided by J.Chapuis of the IMRT, ETH, Zürich.

copter, currently in development at the ETH, Zürich (Chapuis *et al.* (1997)). Two sequences from this dataset were shown in Figs. 3 and 5 and a third is shown in Fig. 13, which depicts a sequence of approximately 9000 values of a variable used to control the helicopter pitch angle. As in the previous example, outliers are clearly evident as spikes in this data sequence; here, these spikes result from noise in the triggering circuits of the digital counters from which these control signals are generated. Further, close examination of this dataset reveals that some of these outliers occur in small patches, as in the other data sequence shown in Fig. 5. Figs. 14 and 15 show the results obtained by applying the data cleaning filter and the median filter, respectively, to this contaminated data sequence. In both cases, the window width is $N = 13$. This choice is based on the fact that outlier patches up to width 6 occure in the data sequence. The outlier filter is based on Eq. (23) with a scaling factor $c = 3$ which is chosen according to Theorem 5.2. The lower threshold bound T_{kmin} is 0.25 taking the limited precision (through A/D conversion) of the helicopter data into account. Comparing the visual appearance of these plots, it appears that both filters are about equally effective in removing outliers from this dataset, but it is again worth examining the extent to which these filters change this dataset. Here, the data cleaning filter changes less than 2.5% of the data values, while the median filter changes approximately 65.8% of these values. Because the nominal variation evident in this data sequence is mostly monotonic, most of the distortion introduced by the causal median filter is the 6 sample delay of the causal median filter relative to the symmetric median filter. In closed loop control applications, this delay corresponds to a phase advance which generally reduces stability margins. Since the real-time outlier filter described here appears to offer nearly identical outlier rejection with no phase advance, it appears to be the better choice for this application.

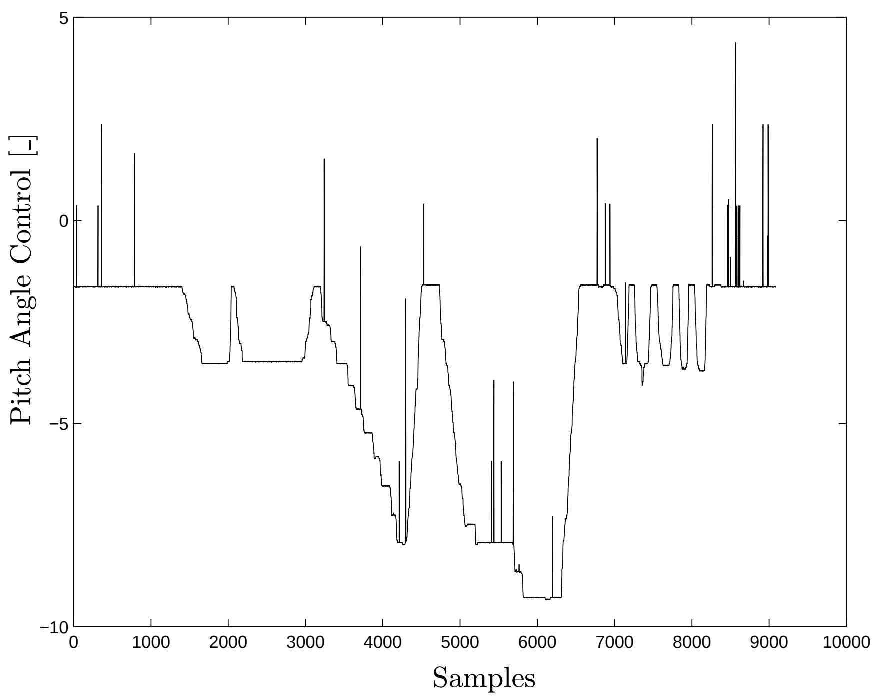

Figure 13: Helicopter pitch angle control variable.

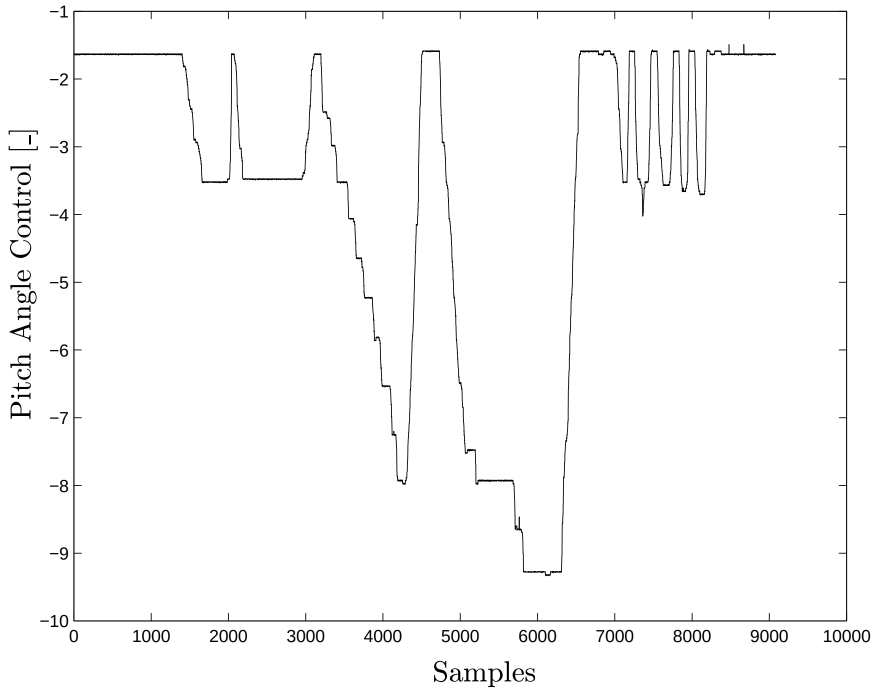

Figure 14: Pitch angle control variable after outlier filtering.

11 Conclusions

This paper describes a simple outlier filter based on a causal moving data window that is appropriate to real–time applications like closed loop control. If the current data value y_k lies too far from the median value in the data window, it is declared an outlier and replaced with a more realistic value $y_k^\star$. Otherwise, the filter leaves y_k unmodified. The main advantage of this filter is that it is simple and universal since no process model is needed for the filter tuning. The basic concept has been described here, and its application to simple real data examples has been presented to illustrate its effectiveness. An important special case of this data cleaning filter is the causal median filter, also described here and corresponding to a "maximally aggressive" data cleaning filter. In the examples considered here, the causal median filter generally appears to be too aggressive, introducing undesirable distortion into the nominal (i.e., uncontaminated) part of the observed data sequence. To aid in selecting an effective but less aggressive data cleaning filter, some useful tuning guidelines are given, based on simple characterizations of the nominal part of the data. In addition, it is useful to note that the nonlinear data cleaning filters described here may be used together with linear noise smoothing filters for reducing the effects of high-frequency measurement noise, but these linear filters should always *follow* the nonlinear data cleaning filters so that outlier replacement preceeds linear smoothing.

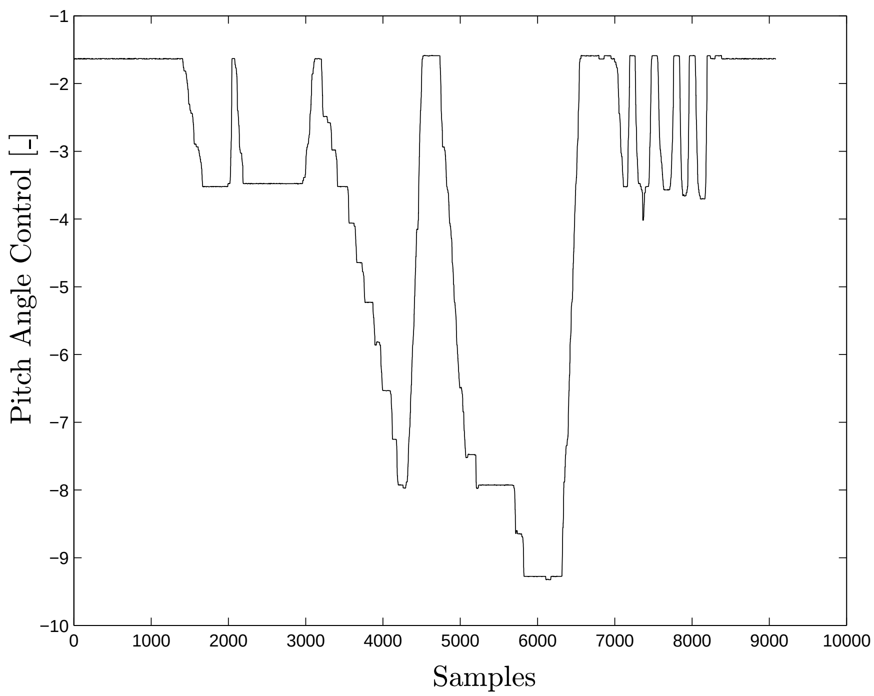

Figure 15: Pitch angle control variable after median filtering.

References

Astola, J., P. Heinonen, and Y. Neuvo (1987). "On root structures of median and median–type filters," *IEEE Trans. Acoust., Speech, Signal Processing*, **35**, no. 8, pp. 1199–1201.

Astola, J. and P. Kuosmanen (1997). *Fundamentals of Nonlinear Digital Filtering*, CRC Press, Boca Raton, New York.

Chapuis, J., C. Eck, and H. Geering (1997). "Autonomously flying helicopter," in *Proc. 4th IFAC Symposium Advances Control Education*, pp. 119–124.

Davies, L. and U. Gather (1993). "The identification of mutiple outliers," *J. Amer. Statist. Assoc.*, **88**, pp. 782–801.

Denby, L. and R. Martin (1979). "Robust Estimation of the First–Order Autoregressive Parameter," *J. Amer. Statist. Assoc.*, **74**, pp. 140–146.

Gallagher, N. J. and G. Wise (1981). "A theoretical analysis of the properties of median filters," *IEEE Trans. Acoust., Speech, Signal Processing*, **29**, no. 6, pp. 1136–1141.

Hamming, R. (1983). *Digital Filters, 2nd ed.*, Prentice–Hall.

Huber, P. (1981). *Robust Statistics*, Wiley, New York.

Ling, L., R. Yin, and X. Wang (1984). "Nonlinear filters for reducing spiky noise: 2–dimensions," in *Proc. 1984 IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pp. 31.8.1–31.8.4.

Martin, R. and V. Yohai (1986). "Influence Functionals for Time Series," *Ann. Statist.*, **14**, pp. 781–818.

Nodes, T. and N. J. Gallagher (1982). "Median filters: Some modifications and their properties," *IEEE Trans. Acoust., Speech, Signal Processing*, **30**, no. 5, pp. 739–746.

Pearson, R. (1999). "Data cleaning for dynamic modeling and control," *accepted for Europ. Contr. Conf. 1999 (ECC'99).*

Rousseeuw, P. and A. Leroy (1987). *Robust Regression and Outlier Detection*, Wiley, New York.