RESEARCH **Open Access**

CrossMark

Generalized Hampel Filters

Ronald K. Pearson[1*] , Yrjö Neuvo[2], Jaakko Astola[3] and Moncef Gabbouj[3]

Abstract

The standard median filter based on a symmetric moving window has only one tuning parameter: the window width. Despite this limitation, this filter has proven extremely useful and has motivated a number of extensions: weighted median filters, recursive median filters, and various cascade structures. The *Hampel filter* is a member of the class of decsion filters that replaces the central value in the data window with the median if it lies far enough from the median to be deemed an outlier. This filter depends on both the window width and an additional tuning parameter t, reducing to the median filter when $t = 0$, so it may be regarded as another median filter extension. This paper adopts this view, defining and exploring the class of *generalized Hampel filters* obtained by applying the median filter extensions listed above: weighted Hampel filters, recursive Hampel filters, and their cascades. An important concept introduced here is that of an *implosion sequence*, a signal for which generalized Hampel filter performance is independent of the threshold parameter t. These sequences are important because the added flexibility of the generalized Hampel filters offers no practical advantage for implosion sequences. Partial characterization results are presented for these sequences, as are useful relationships between root sequences for generalized Hampel filters and their median-based counterparts. To illustrate the performance of this filter class, two examples are considered: one is simulation-based, providing a basis for quantitative evaluation of signal recovery performance as a function of t, while the other is a sequence of monthly Italian industrial production index values that exhibits glaring outliers.

1 Introduction

In their paper, "On a class of nonlinear filters," Sicuranza and Carini begin by noting [1]:

> "The set of nonlinear filters is extremely large since their definition simply excludes the applicability of the linear superposition property on which the theory of linear filters is based. However, from the very beginning, attempts have been done to suitably classify nonlinear filters on the basis of some peculiar properties, leading to the identification of certain classes of nonlinear filters."

This paper adopts a similar philosophy, restricting consideration to a class of nonlinear filters obtained by combining two previously studied filter classes: the *Hampel filter* described in Section 2, and the median filter extensions described in Sections 4 and 7. The result is a class of nonlinear filters we believe to be new, that includes all of these previously studied filters as special cases, but which exhibits a greater degree of design flexibility.

*Correspondence: ronald.k.pearson@gmail.com
[1]DataRobot, Inc., Boston MA, USA
Full list of author information is available at the end of the article

2 Standard median and Hampel filters

All of the filters discussed in this paper are based on the following moving data window, or some simple extension of it:

$$\mathbf{W}_k^K = \{x_{k-K}, \ldots, x_k, \ldots, x_{k+K}\}, \tag{1}$$

where K is a positive integer called the *window half-width*. The *standard median filter* $\mathcal{M}_K$ was introduced by J.W. Tukey in 1974 [2] and is obtained by computing the median of the moving data window $\mathbf{W}_k^K$:

$$m_k = \text{median}\{x_{k-K}, \ldots, x_k, \ldots, x_{k+K}\}. \tag{2}$$

The only tuning parameter for this filter is the window half-width parameter K, which limits its flexibility, but the real strength of the median filter lies in its extreme resistance to *local outliers* or *impulsive noise* in the input data squence $\{x_k\}$. Unfortunately, the median filter can also introduce significant distortion in the portion of the signal we wish to retain, making its utility strongly application-dependent. These filter characteristics have led to the development of a number of median filter extensions, including the recursive median filter discussed in Section 4 and others described in Section 7.

Pearson *et al. EURASIP Journal on Advances in Signal Processing* (2016) 2016:87

Page 2 of 18

A closely related filter is the *Hampel filter* $\mathcal{H}_K$, which belongs to the class of *decision-based filters* discussed in the book by Astola and Kuosmanen ([3] p. 194), who note that the basic concept has been reinvented again and again. The version considered here represents a moving-window implementation of the *Hampel identifier* described by Davies and Gather [4], an outlier detection procedure based on the median and the MAD scale estimator. Specifically, this filter's response is given by:

$$y_k = \begin{cases} x_k & |x_k - m_k| \le tS_k, \\ m_k & |x_k - m_k| > tS_k. \end{cases} \quad (3)$$

where m_k is the median value from the moving data window and S_k is the MAD scale estimate, defined as:

$$S_k = 1.4826 \times \text{median}_{j \in [-K,K]}\{|x_{k-j} - m_k|\}. \quad (4)$$

The factor 1.4826 makes the MAD scale estimate an unbiased estimate of the standard deviation for Gaussian data.

The key observation on which this paper is based is that, when the threshold parameter t is set to zero, we recover the standard median filter:

$$y_k|_{t=0} = m_k. \quad (5)$$

It follows from this observation that we may regard the Hampel filter as a generalization of the median filter, with t as an additional tuning parameter. The central question explored in this paper is what the consequences of this generalization are when we combine it with other generalizations of the median filter that are well-known in the literature, as described in Sections 4 and 7.

A filter's *root sequences* are those sequences $\{x_k\}$ that are invariant under the action of the filter, and the root sequences for the standard median filter have been well-characterized (see, for example [5] or [6]). Thus, it is worth noting that the set $\mathcal{R}_t$ of root sequences for the Hampel filter with threshold t contains the median filter root sequence $\mathcal{R}_0$ for all $t \ge 0$. Specifically, if $s \le t$, it follows that:

$$|x_k - m_k| \le sS_k \le tS_k \Rightarrow \mathcal{R}_s \subset \mathcal{R}_t. \quad (6)$$

The practical implication of this result is that the Hampel filter may be viewed as a "less aggressive extension" of the median filter, generally becoming less aggressive with increasing threshold value t. In particular, for "most" sequences $\{x_k\}$, the Hampel filter varies from the median filter at is most aggressive (i.e., for $t = 0$) to an identity filter as $t \to \infty$. The important exception to this behavior is the class of *implosion sequences* described next.

3 Implosion sequences
The MAD scale estimator has the extremely desirable characteristic of exhibiting the maximum possible outlier resistance [4], but it does suffer from an unfortunate sensitivity to *implosion:* if more than 50 % of the data values are the same, the MAD scale estimate is zero, independent of the other values in the data sequence. The practical consequences for the Hampel filter are that if $K + 1$ or more of the values in the data window $\mathbf{W}_k^K$ have the same value, then $S_k = 0$, implying that $y_k = m_k$, independent of the threshold parameter t. Thus, we make the following definitions:

1. Define the window $\mathbf{W}_k^K$ to be an *implosion window* if $S_k = 0$;
2. Define the sequence $\{x_k\}$ to be an *implosion sequence* if all windows are implosion windows (i.e., if $S_k = 0$ for all k);
3. Define the sequence $\{x_k\}$ to be *implosion-free* if it contains no implosion windows (i.e., if $S_k > 0$ for all k).

The practical consequence of these definitions is that if $\{x_k\}$ is an implosion sequence, the output of the Hampel filter reduces to that of the median filter for all t, so the added flexibility of the Hampel filter offers no practical advantage for these sequences. Similarly, since the Hampel filter root set contains the median filter root set for all threshold values t, the added flexibility of the Hampel filter offers no practical advantage for these sequences, either. *Thus, the signals of greatest interest in characterizing Hampel filter performance are implosion-free sequences that are not median filter roots.*

As noted in Section 2, the Hampel filter reduces to the standard median filter when the threshold parameter has the value $t = 0$, and it becomes generally less aggressive with increasing t. It follows directly from the defining equations that the Hampel filter has no effect on the input signal if the following condition is satisfied:

$$\max_k |x_k - m_k| \le t \min_k S_k, \quad (7)$$

where the maximum on the left-hand side and the minimum on the right-hand side of this condition are taken over all moving data windows. If $\{x_k\}$ is an implosion-free sequence, it follows that $\min_k S_k > 0$, so Eq. (7) can be inverted to yield the following condition for signal preservation:

$$t \ge \frac{\max_k |x_k - m_k|}{\min_k S_k}. \quad (8)$$

That is, if $\{x_k\}$ is an implosion-free sequence, the Hampel filter reduces to an identity filter for some sufficiently large but finite value of t. This result means that the practical characterization of Hampel filter performance can be restricted to the range $0 \le t \le t^*$, where t^* is this identity filter threshold value.

Theorem *The sequence $\{x_k\}$ is an implosion sequence for $\mathcal{H}_K$ if and only if, for all k, more than K elements of the window $\mathbf{W}_k^K$ have the same value.*

Proof 1. Assume $\{x_k\}$ is an implosion sequence for $\mathcal{H}_K$. This means:

$$\mathrm{median}\{|x_k - m_k|\} = 0,$$

implying $|x_k - m_k| = 0$ for at least $K + 1$ values, implying $x_k = m_k$ for at least $K + 1$ values in $\mathbf{W}_k^K$.

2. Conversely, suppose that at least $K + 1$ values in $\mathbf{W}_k^K$ are equal to some constant c. It follows immediately that the median value in this window is $m_k = c$, implying $|x_k - m_k| = 0$ for at least $K + 1$ values, implying $S_k = 0$ so that $\{x_k\}$ is an implosion sequence for $\mathcal{H}_K$.

□

This result allows us to construct some specific examples of Hampel filter implosion sequences from the signal components used by Gallagher and Wise to characterize median filter root sequences [6]. Specifically, given K, define the following four components:

1. A *constant neighborhood* is a sequence of at least $K + 1$ consecutive identical values;
2. An *edge* is a monotonically increasing or decreasing sequence, preceded and followed by constant neighborhoods of different values;
3. An *impulse* is a sequence of at most K values, preceded and followed by constant neighborhoods having the same value, with the values of the intermediate points distinct from those of the surrounding constant neighborhoods;
4. An *oscillation* is any sequence of values not contained in a constant neighborhood, an edge, or an impulse.

Based on these definitions, it can be shown that $\{x_k\}$ is a root sequence for the median filter $\mathcal{M}_K$ if and only if it consists entirely of constant neighborhoods and edges [6].

Note that by the above theorem, a sequence $\{x_k\}$ that consists entirely of constant neighborhoods will be an implosion sequence for $\mathcal{H}_K$. In this case, it follows by the above result that $\{x_k\}$ is also a root sequence for the median filter $\mathcal{M}_K$, so we expect no difference in behavior between the median and Hampel filters for this case by the root sequence nesting condition (6). A more interesting example is the case of a sequence $\{x_k\}$ composed of constant neighborhoods and impulses. Here again, it is easy to see that this sequence is an implosion sequence for $\mathcal{H}_K$, but it is not a median filter root sequence. In this case, the Hampel filter will reduce to the median filter for all threshold parameters t and map $\{x_k\}$ to a sequence of constant neighborhoods with the impulses removed. Note

that this sequence is a median filter root sequence. Finally, a third class of implosion sequences is the class of *binary oscillations*:

$$x_k = \begin{cases} a & k \text{ even,} \\ b & k \text{ odd,} \end{cases} \tag{9}$$

for any $a \neq b$. Since at any k, the moving window $\mathbf{W}_k^K$ will have K of one of these values and $K + 1$ of the other value, it follows immediately from the above theorem that $\{x_k\}$ is an implosion sequence for $\mathcal{H}_K$.

An interesting open question is whether there are other classes of implosion sequences for $\mathcal{H}_K$ besides the three just described. Since any root sequence for the median filter $\mathcal{M}_K$ is also a root for all Hampel filters $\mathcal{H}_K$, regardless of threshold, the important implosion sequences are those that are not median filter roots: these sequences are modified by the median filter and also modified in exactly the same way by the Hampel filter, independent of the threshold parameter t.

4 Recursive median and Hampel filters

The recursive median filter is obtained by replacing the symmetric moving window $\mathbf{W}_k^K$ defined in Eq. (1) with the following recursive data window:

$$\mathbf{R}_k^K = \{m_{k-K}, \ldots, m_{k-1}, x_k, x_{k+1}, \ldots, x_{k+K}\}, \tag{10}$$

where m_{k-j} represents the output at prior time $k - j$ of the standard median filter applied to the input sequence $\{x_k\}$. This extension exhibits a number of interesting properties, including *idempotence* [7], i.e., a single application of the recursive median filter maps $\{x_k\}$ into the filter's root set. Further, it has also been shown that the root set for the recursive median filter is identical to that for the standard median filter.

The *recursive Hampel filter* is defined analogously, replacing the recursive window defined in Eq. (10) based on prior median filter outputs, with the alternative window:

$$\mathbf{R}_k^{t,K} = \{H_{k-K}^t, \ldots, H_{k-1}^t, x_k, x_{k+1}, \ldots, x_{k+K}\}, \tag{11}$$

where H_{k-j}^t represents the output at prior time $k - j$ of the Hampel filter with threshold parameter t applied to the input sequence $\{x_k\}$.

It follows by direct extension of the root set nesting result given in Eq. (6) for the nonrecursive case that the recursive Hampel filter root set contains the recursive median filter root set. Specifically, if $\{r_k\}$ is a root for the Hampel filter with threshold s for $0 \le s \le t$, then:

$$|r_{k-j} - m_{k-j}| \le sS_{k-j} \le tS_{k-j} \implies H_{k-j}^s = H_{k-j}^t = r_{k-j}. \tag{12}$$

Thus, if we let $\tilde{\mathcal{R}}_t$ denote the root set for the recursive Hampel filter with threshold parameter t, the following two conclusions are immediate:

Pearson *et al. EURASIP Journal on Advances in Signal Processing* (2016) 2016:87

Page 4 of 18

1. The recursive and non-recursive Hampel root sets are identical for every threshold parameter: $\tilde{\mathcal{R}}_t = \mathcal{R}_t$ for all t;
2. The recursive Hampel root sets nest: for all $0 \leq s \leq t$, it follows that $\tilde{\mathcal{R}}_s \subset \tilde{\mathcal{R}}_t$.

Beyond these results, the following interesting questions are open at present:

1. The recursive median filter is idempotent—does this behavior extend to recursive Hampel filters for arbitrary t? If not, is the recursive median filter the *only* idempotent member of this family? More generally, how does idempotence depend on t?
2. What is the relationship between implosion sequences for the recursive and non-recursive Hampel filters?

5 The influence of *t* on filter performance

To provide quantitative filter performance results, the following section presents a brief case study that examines the influence of the Hampel filter tuning parameter t on the performance of both the standard Hampel filter and the recursive Hampel filter. Since the primary question of interest is the influence of the tuning parameter t, this example considers a fixed window half-width parameter (specifically, $K = 5$, yielding an 11-point moving window filter) and examines filter performance over a range of t values. The basis for these performance comparisons is a simulated data example described in Section 5.1: the advantage of considering a simulation-based example is that we can be explicit about the signal components we wish to recover and can therefore quantify signal recovery performance. More specifically, this example considers two possible signal recovery problems described in detail in Section 5.1 and characterizes performance in terms of two metrics: the root mean square signal recovery error (RMSE) and the mean absolute signal recovery error (MAE).

5.1 A simulated data example

To provide a basis for comparing the different filters considered in this paper, we apply them to the 420-point simulated data sequence shown in Fig. 1, which contains four components:

1. Step-and-ramp sequence (median filter root) for $k = 1, 2, \ldots, 420$;
2. Low-level Gaussian noise (partial: nonzero only for $k = 1, 2, \ldots, 240$);
3. Sinusoid (partial: nonzero only for $k = 101, 102, \ldots, 420$);
4. Impulsive noise, randomly distributed throughout the sequence.

More specifically, the step-and-ramp sequence consists of eight segments:

1. $y_k = 0$ for $k = 1$ to $k = 40$;
2. a linear increase from $y_k = 0$ to $y_k = 1$ from $k = 41$ to $k = 100$;
3. $y_k = 1$ for $k = 101$ to $k = 140$;
4. $y_k = 2$ for $k = 141$ to $k = 220$;
5. a linear decrease from $y_k = 2$ to $y_k = 0$ from $k = 221$ to $k = 300$;
6. $y_k = 0$ for $k = 301$ to $k = 320$;
7. $y_k = -1$ for $k = 321$ to $k = 400$;
8. $y_k = 0$ for $k = 401$ to $k = 420$.

The Gaussian noise component has mean zero and standard deviation $\sigma = 0.1$, and the sinusoid has period 29 and amplitude 0.3. The impulsive noise component is an additive term that is zero everywhere except for the following eight values of k, where it takes the nonzero values indicated in parentheses: $k = 20$ (+1), $k = 35$ (−1), $k = 120$ (+1), $k = 190$ (−1.5), $k = 220$ (−2.5), $k = 300$ (+1), $k = 350$ (+2.5), and $k = 410$ (+1.5).

The primary question of interest here is how well the different filters considered eliminate the isolated spikes in this signal while preserving the low-level details, especially the sinusoidal component. The presence of the low-level noise in approximately the first half of the signal raises a subtle practical issue, however: is a "good" filter one that simply removes the impulsive spikes from the data sequence, or should it also address the low-level noise? Given that median filters and their extensions are much better suited to the removal of impulsive noise than the smoothing of low-level noise, the first formulation seems the more reasonable here, but the question is raised to emphasize that filter performance criteria are generally problem-specific.

Additional insights can be obained from this example by considering filter performance for the three qualitatively distinct signal subsequenes separated by dashed vertical lines in Fig. 1. Specifically, the first 100 points of the sequence—denoted "Noise Only" in Fig. 1—consists of a median filter root sequence, contaminated with both low-level Gaussian noise and impulsive noise spikes. The second subsequence, from $k = 100$ to $k = 240$ and labelled "Noise + Sine," contains all four of the signal components listed above, while the third subsequence, from $k = 240$ to $k = 420$ and labelled "Sine Only," consists of a median filter root sequence with a superimposed sinusoid and isolated spikes, but no low-level noise.

The two signal recovery problems considered here are the following:

P1   the impulsive noise removal problem, where the signal to be recovered consists of the sum of the first three components listed above;
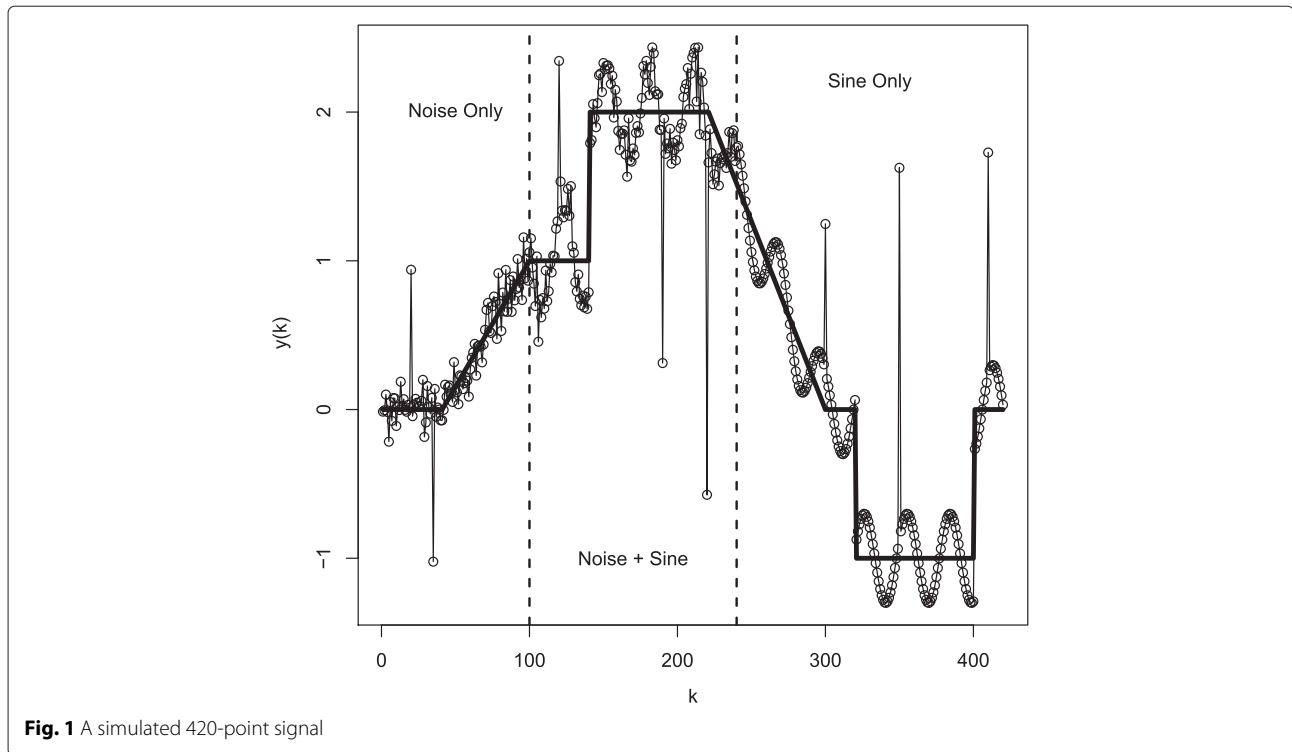
Pearson *et al. EURASIP Journal on Advances in Signal Processing* (2016) 2016:87

Page 5 of 18

Fig. 1 A simulated 420-point signal

P2   the complete noise removal problem, where the signal to be recovered consists of the sum of the two deterministic components (i.e., the median filter root plus the sinusoid), without either low-level or impulsive noise.

As noted above, these signal recovery problems have different characters, with the first being more suitable for the filter class considered here, but the second problem is of considerable practical significance. Two performance measures are considered for both problems: the root-mean-square recovery error (RMSE) is more widely used, but may be less appropriate than the mean absolute recovery error (MAE) in the presence of impulsive noise.
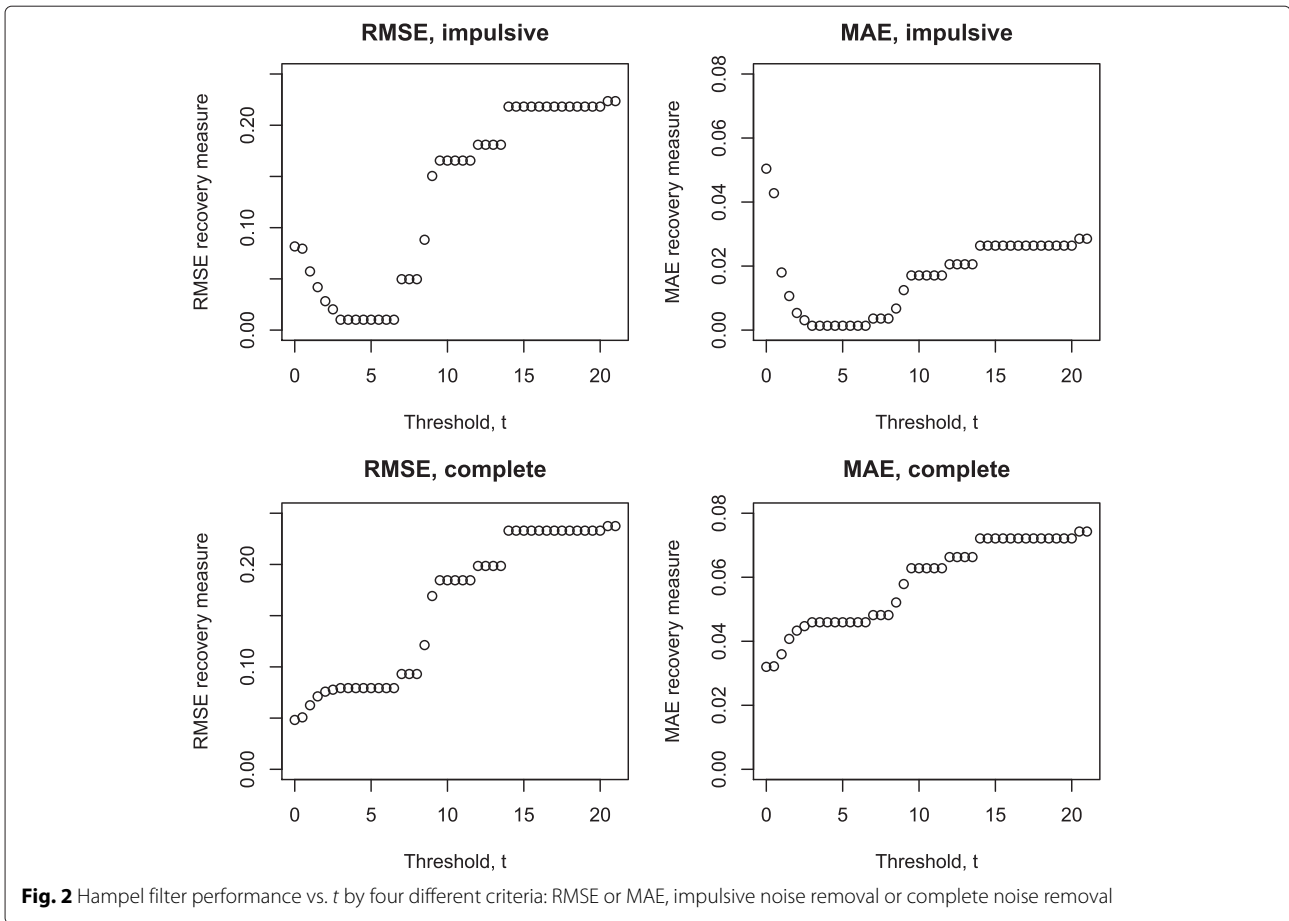
Finally, it is important to note that, for the filter window width considered here ($K = 5$), the signal sequence shown in Fig. 1 is implosion-free and is not a median filter root sequence. Thus, it follows that filter performance should depend on the threshold parameter t, and the objective of the following discussions is to illuminate the nature of this dependence.

5.2   Results for the Hampel filter

For the signal defined in Section 5.1, the identity filter threshold described in Section 3 is approximately $t = 21$, so the results presented here consider the performance of the Hampel filter over the range from $t = 0$ to $t = 21$, in increments of 0.5. Four views of the signal recovery performance of the Hampel filter over this range of t values

are presented in Fig. 2. The upper left plot shows the RMSE signal recovery measure for the impulsive noise removal problem, the upper right shows the corresponding MAE signal recovery measure, the lower left plot shows the RMSE measure for the complete noise removal problem, and the lower right shows the MAE measure for this problem. Note that the two RMSE plots are shown on the same scale to facilitate comparison, as are the two MAE plots, but the RMSE and MAE scales are different. For the impulsive noise removal problem (the upper two plots), both measures exhibit a broad but well-defined minimum for threshold parameters t between 3.0 and 6.5. For values much smaller than 3, performance degrades sharply as t decreases to the $t = 0$ median filter limit; similarly, performance again degrades as the t value increases from 6.5, particularly for the RMSE measure, as the Hampel filter approaches the identity limit. The MAE view in the upper right is particularly interesting here: this performance measure is poorest for the median filter, becoming consistently better than the median filter for all $t \geq 1.0$. This result reflects the significant distortion introduced into the signal sequence by the median filter, offsetting its ability to remove the noise spikes.
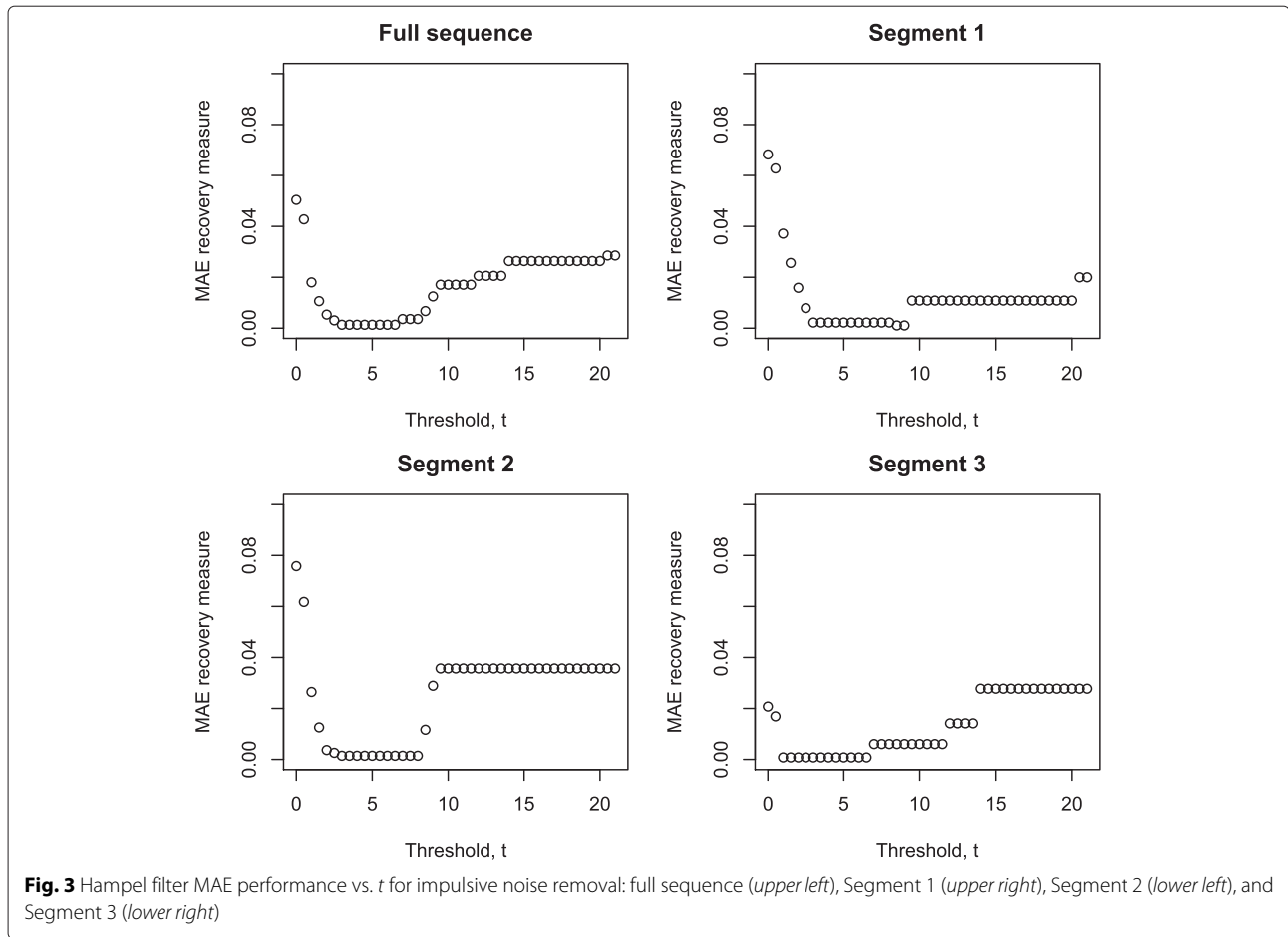
For the complete noise removal problem (the lower two plots), the dependence of filter performance on the threshold parameter is very different. In particular, performance degrades uniformly with increasing t for both the RMSE and MAE measures. Since the complete noise removal objective requires removal of both the impulsive

Pearson *et al. EURASIP Journal on Advances in Signal Processing* (2016) 2016:87

Page 6 of 18

Fig. 2 Hampel filter performance vs. *t* by four different criteria: RMSE or MAE, impulsive noise removal or complete noise removal

noise and the low-level noise, these results suggest that as *t* increases, the Hampel filter allows more of the low-level noise to pass through the filter unmodified, offsetting the performance advantage of lower distortion of the sinusoidal signal components. In particular, since the filter removes all of the impulsive noise spikes for *t* between 0 and 6.5, it follows that the poorer performance seen for the complete noise removal problem over the impulsive noise removal optimal performance range (*t* = 3.5 to *t* = 6.0) relative to the median filter limit *t* = 0 is caused by the filter's allowing more low-level noise into the output signal. These results emphasize the point made earlier that these filters are not well-suited to low-level noise removal problems.

Figure 3 shows the MAE performance for the impulsive noise removal problem as a function of *t*, broken down by signal segment: the upper left plot corresponds to the upper right plot in Fig. 2, characterizing the complete signal sequence, while the other three plots show the corresponding results for the three segments indicated in Fig. 1. The upper right plot presents the results for Segment 1 ("Noise only"), consisting of the median filter root sequence, low-level Gaussian noise, and impulsive noise

spikes. This plot clearly shows the low-level noise distortion effects for small *t* values, which is worst for the median filter (*t* = 0), decreasing monotonically until *t* = 3.0, where the filter is sufficiently non-aggressive to allow most of the low-level noise through unmodified. In fact, the optimal filter performance for this signal sequence occurs at *t* = 8.5 where the MAE is near zero. For *t* ≥ 9.5, the filter begins allowing impulsive noise spikes into the output, causing a dramatic increase in MAE. The lower left plot in Fig. 3 shows the results for Segment 2 ("Noise + Sine"). As in Segment 1, the performance is worst for the median filter, improving uniformly with increasing *t* until the optimal plateau between *t* = 3.0 and *t* = 6.5, where the filter is aggressive enough to remove all of the impulsive noise spikes but forgiving enough to pass the low-level noise and sinusoidal components without distortion. As *t* increases beyond this range, the Hampel filter quickly becomes an identity filter, passing all of the impulsive noise spikes for *t* ≥ 9.5. Finally, for Segment 3 ("Sine only," lower right plot), the distortion introduced in the sinusoidal component by the median filter reduces essentially to zero for *t* ≥ 1.0 and the Hampel filter exhibits optimal performance for 1.0 ≤ *t* ≤ 6.5. As *t* increases

Pearson *et al. EURASIP Journal on Advances in Signal Processing* (2016) 2016:87

Page 7 of 18

Fig. 3 Hampel filter MAE performance vs. *t* for impulsive noise removal: full sequence (*upper left*), Segment 1 (*upper right*), Segment 2 (*lower left*), and Segment 3 (*lower right*)

beyond this limit, the filter begins to pass impulsive noise spikes, becoming an identity filter for $t \geq 14.0$.

Figure 4 shows a plot of the median filter response ($t = 0$, represented by open circles), overlaid with a solid line representing the response of the Hampel filter with $t = 5$, falling in the optimal parameter range for the complete signal and all segments except Segment 1, where the performance is near-optimal. In addition, points where these two filter responses differ are indicated by solid rectangles. From these results, it is clear that the Hampel filter with $t = 5$ passes both the low-level noise components and the sinusolidal components essentially perfectly, while the median filter seriously distorts the portions of the signal contaminated with low-level noise, and it "clips" the tops and bottoms of the sinusoidal component. It is also clear that both filters remove all of the impulsive noise from the signal sequence.

The frequency of the sinusoidal component in this example is important. Specifically, the maximum possible frequency is that of the binary implosion sequence described in Section 3, implying that in this limit, the Hampel filter offers no advantage over the median filter. At the other extreme, if the sinusoidal frequency is low

enough, the N-point finite signal sequence will be monotonic, and thus a root sequence for the median filter and all Hampel filters. For intermediate frequencies; however, sinusoidal components are neither implosion sequences nor roots, and as this example illustrates, the response of the Hampel filter to these components generally varies strongly with t.

5.3 Results for the recursive Hampel filter
Figure 5 shows the complete sequence performance for the recursive Hampel filter. Specifically, the upper left plot shows the RMSE measure versus t for the impulsive noise removal problem, while the upper right plot shows the corresponding MAE results; the lower two plots present these same results for the complete noise removal problem. Comparing the upper two plots in Fig. 5 with those in Fig. 2, we see the same general behavior of the recursive Hampel filter as that for the standard Hampel filter, although the "optimal plateau" starts later and is slightly shorter. Indeed, Fig. 6 shows that the low-level distortion is worse for the recursive median filter than that for the standard median filter, although it declines rapidly with increasing t until the optimal plateau is
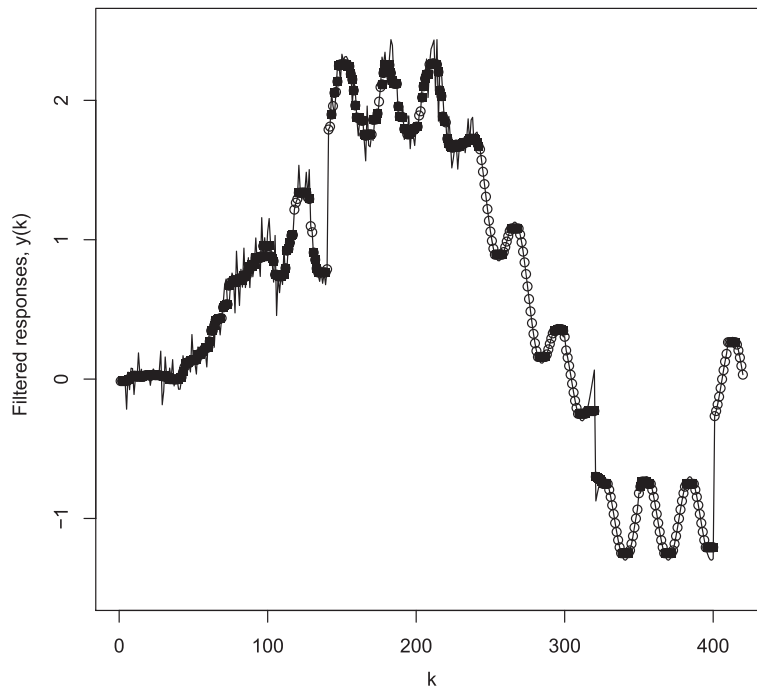
Pearson *et al. EURASIP Journal on Advances in Signal Processing* (2016) 2016:87

Page 8 of 18

Fig. 4 Median filter response (*open circles*) and Hampel filter response with $t = 5$ (*lines*), with points of disagreement marked as *solid squares*
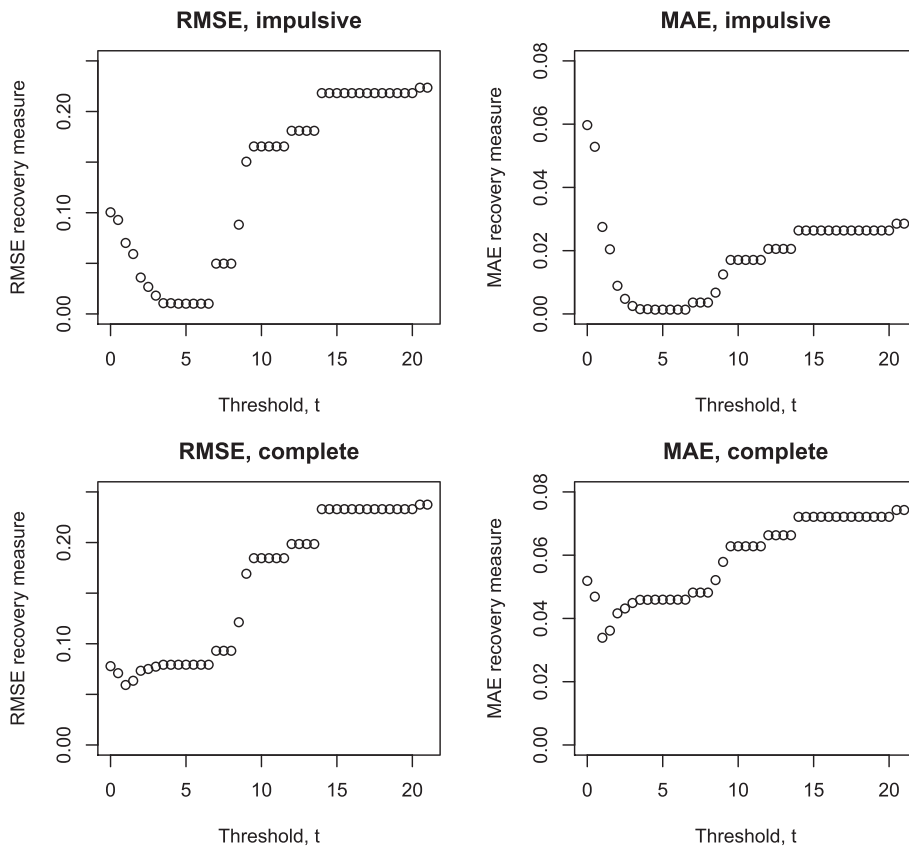

Fig. 5 Recursive Hampel filter performance vs. *t* by four different criteria: RMSE or MAE, impulsive noise removal or complete noise removal
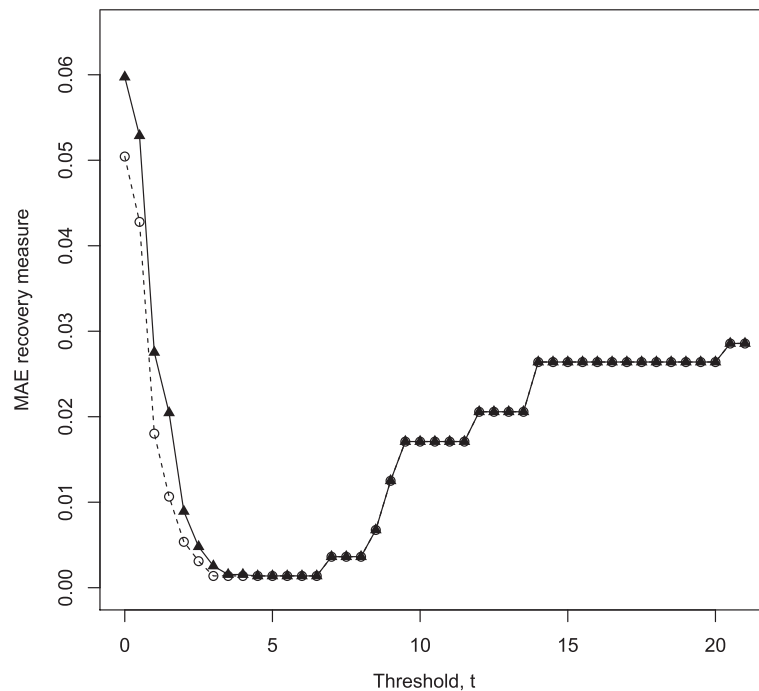
Pearson *et al. EURASIP Journal on Advances in Signal Processing*   (2016) 2016:87

Page 9 of 18

Fig. 6 Recursive (*solid triangles*) and non-recursive (*open circles*) Hampel filter MAE performance vs. *t* for impulsive noise removal

reached, after which the two filter responses appear to be identical.

The more interesting results in Fig. 5 are the two bottom plots for the complete noise removal problem: in contrast to the monotonic behavior seen in the corresponding plots in Fig. 2, the recursive filter exhibits a sharp optimum at $t = 1$. A more detailed comparison of the recursive and nonrecursive Hampel filter MAE performance is shown in Fig. 7: for small t, the recursive filter performance is much worse than the standard filter, although for t values between 1.0 and 3.0, the recursive filter actually performs slightly better; for larger t values, both filters exhibit essentially identical performance.

Figure 8 summarizes the recursive median filter's MAE performance for the complete noise removal problem as a function of t for the complete signal and the three segments marked in Fig. 1. The upper left plot shows the results for the complete signal and is the same as the lower right plot in Fig. 5, included here to facilitate visual comparisons. The upper right plot shows the results for Segment 1 ("Noise only") and here, the optimum at $t = 1$ is much sharper than that for the complete signal, with performance degrading much more rapidly as t increases beyond this value. The results for Segment 2 ("Noise + sine") shown in the lower left plot are a bit more complicated: optimal performance is again obtained for $t = 1$, but this optimum is shallower than that for Segment 1 and there is a second, small local minimum from $t = 2.5$ to $t = 3$, after which performance again degrades monotonically

with increasing t. Finally, the performance for Segment 3 ("Sine only", lower right plot) is virtually identical to that seen for the standard Hampel filter shown in the lower right plot in Fig. 3.

Overall, these results—particularly those for the complete noise removal performance of the recursive Hampel filter—show that the performance of these filters depends strongly on the threshold value t, but very differently for different signal extraction problems and different signal characteristics. For example, for Segment 3 ("Sine only"), the performance of the recursive and standard Hampel filters are almost identical, both for the impulse noise removal problem and for the complete noise removal problem: distortion is observed for t less than 1.0, excellent performance is observed for t between 1.0 and 6.5, with consistent performance degradation as t is increased beyond this value. In contrast, for Segment 1 ("Noise only"), these performance curves are very different: for the impulsive noise removal problem with the standard Hampel filter, performance is worst in the median filter limit, improves uniformly as t increases to 3.5 where it remains near-optimal as t increases to 8.5; optimal performance—only slightly better—is achieved for t between 8.5 and 9.0, after which performance becomes discontinuously worse, but never approaches the level of poor performance seen for the median filter. In contrast, for the complete noise removal problem with the recursive Hampel filter for this data segment, a sharp optimum is observed at $t = 1.0$, with increasingly poorer performance
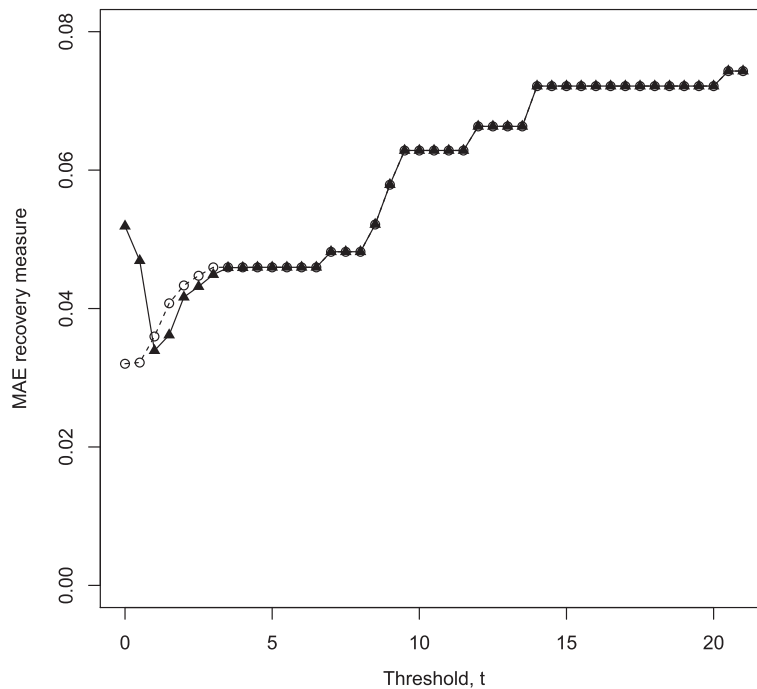
Pearson *et al. EURASIP Journal on Advances in Signal Processing*   (2016) 2016:87

Page 10 of 18

Fig. 7 Recursive (*solid triangles*) and non-recursive (*open circles*) Hampel filter MAE performance vs. *t* for complete noise removal
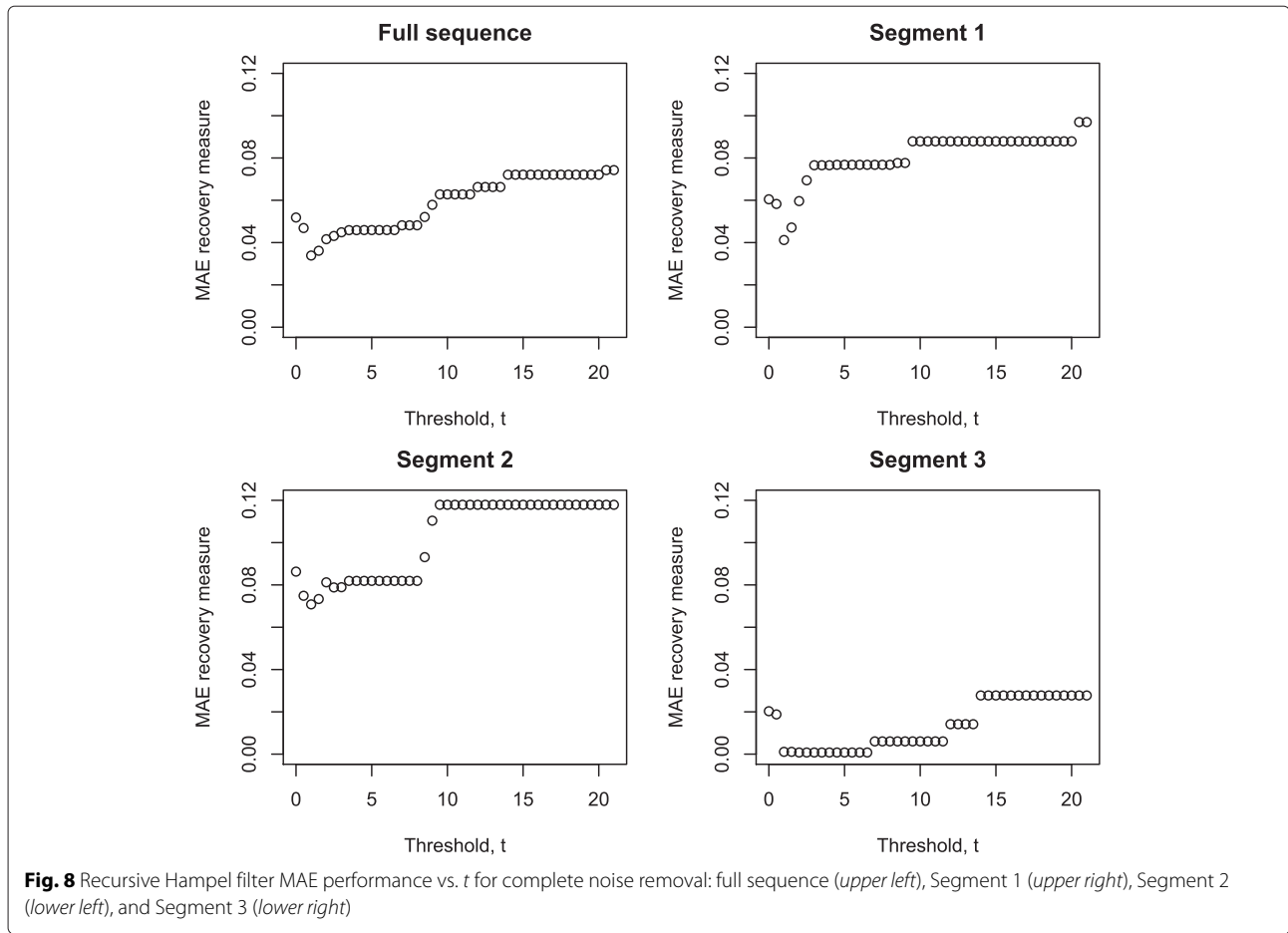
as t increases, exhibiting worse performance than the recursive median filter for all $t > 2$. Finally, as noted, the complete noise removal performance for the recursive Hampel filter for Segment 2 ("Noise + sine") is even more complicated, exhibiting local optima in its MAE vs. t performance curve.

6   A real data example

To provide an illustration of how the generalized Hampel filters described in this paper work with a real data example, the following section applies several of these filters to a publically-available time-series dataset. Specifically, this example is based on the *gipi* sequence included in the *tsoutliers R* package [8], available as one row of the *bde9915* data frame. This data sequence is a monthly time-series of Italian industrial production index from 1981 to 1996, consisting of 192 observations. A plot of this time-series is shown in Fig. 9, from which the presence of significant outliers in the data is clear. In fact, these anomalous data points occur at regular 12-month intervals and represent what Kaiser and Maravall call *seasonal outliers* [9]. If we apply standard time-series modeling procedures (e.g., fitting ARMA or ARIMA models to the data), the results will be profoundly influenced by the presence of these outliers, and at least two general strategies can be used to address these problems. The first is the development of specialized analysis procedures

that are resistant to the anomalies in the data, extending standard analysis methods using fundamental ideas from robust statistics, such as the robust time-series modeling approach described by Martin and Yohai [10] or the robust-resistant spectrum estimation approach described by Martin and Thomson [11]. The second approach is the use of simple data-cleaning filters like those described in this paper to remove the outliers from the data sequence, after which standard analysis procedures are applied. The primary objective of this example is to illustrate the range of results that may be obtained when different generalized Hampel filters are applied to the time-series shown in Fig. 9.

Figure 10 shows the results of two standard median filters (upper two plots) and two recursive median filters (lower two plots) applied to the Italian industrial production data shown in Fig. 9. The left-hand plots correspond to filters based on the window half-width parameter $K = 3$ (i.e., 7-point moving data windows), while the right-hand plots correspond to filters with $K = 5$ (i.e., 11-point moving data windows). In all cases, the vertical scale is the same to facilitate comparisons. All of these filters completely eliminate the seasonal outliers, but they also introduce significant distortion in the nominal part of the signal. This is less pronounced in the standard median filter results, where the original signal details are much better approximated than in the results obtained from the

Pearson *et al. EURASIP Journal on Advances in Signal Processing* (2016) 2016:87

Page 11 of 18

Fig. 8 Recursive Hampel filter MAE performance vs. *t* for complete noise removal: full sequence (*upper left*), Segment 1 (*upper right*), Segment 2 (*lower left*), and Segment 3 (*lower right*)

recursive median filters. It is also clear that the distortion introduced by these filters is worse for $K = 5$ than it is for $K = 3$.

Figure 11 shows comparative results for four different Hampel-based filters. As in Fig. 10, the top two plots are for nonrecursive Hampel filters, while the bottom two plots are for their recursive counterparts. Here, all of these filters are based on the half-width parameter $K = 5$, with the right- and left-hand plots differing in the Hampel threshold parameter t. Specifically, the left-hand plots correspond to the more aggressive threshold value $t = 1$, while the right-hand plots are based on $t = 2$. As before, all of these filters completely eliminate the seasonal outliers from the data, introducing much less distortion in the nominal part of the signal than the corresponding median filters do. Comparing the left-hand and right-hand plots, it is also clear that these filters introduce much less distortion with $t = 2$ than with $t = 1$. Comparing the upper and lower plots, it is also clear that while the recursive Hampel filter introduces more nominal signal distortion than the nonrecursive filter for this signal, this effect becomes much less pronounced with increasing t.

One type of generalized Hampel filter that was not discussed in connection with the simulation example was the subclass of cascade interconnections of Hampel filters and/or recursive Hampel filters. Figure 12 shows the results obtained when four different filter cascades are applied to the Italian industrial production index data shown in Fig. 9. In the upper left plot, the results were obtained by first applying the standard median filter with $K = 3$ (i.e., the 7-point median filter) to the raw signal, and then applying the recursive median filter with $K = 5$ (i.e., the 11-point recursive median filter) to the output of this filter. Comparing this plot with those for either of the individual components of this cascade in Fig. 10 (i.e., the standard median filter with $K = 3$ shown in the upper left plot and the recursive median filter with $K = 5$ shown in the lower right plot), it is clear that the cascade results are intermediate in their tendency to emphasize the low-frequency trend in the data at the expense of key high-frequency details, while still removing the seasonal outliers. The upper right plot in Fig. 12 relaxes both components of this first cascade, increasing the threshold parameter from the median filter limit $t = 0$ to the
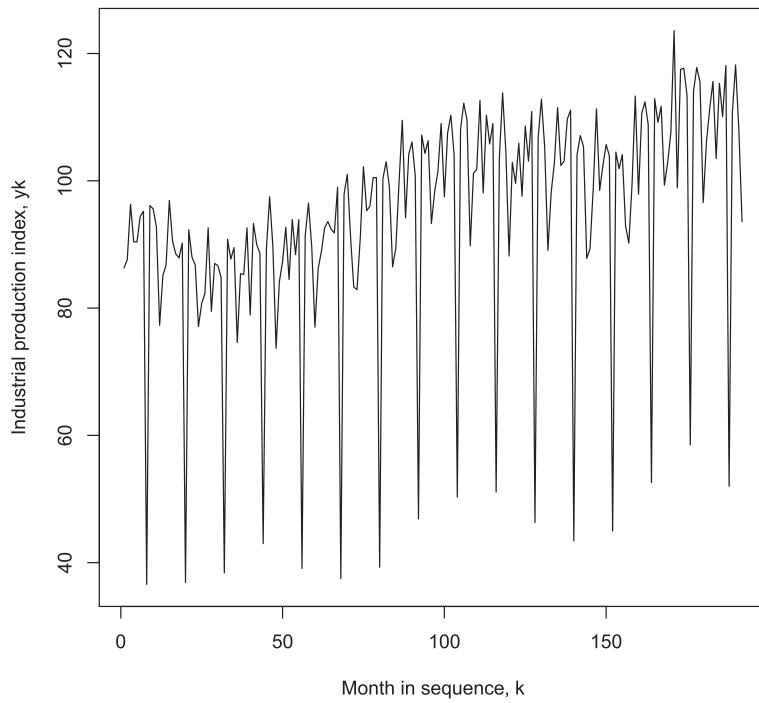
Pearson *et al. EURASIP Journal on Advances in Signal Processing* (2016) 2016:87

Page 12 of 18

Fig. 9 Monthly Italian industrial production index, 1981–1996

Fig. 10 Median filter and recursive median responses to the Italian industrial production index
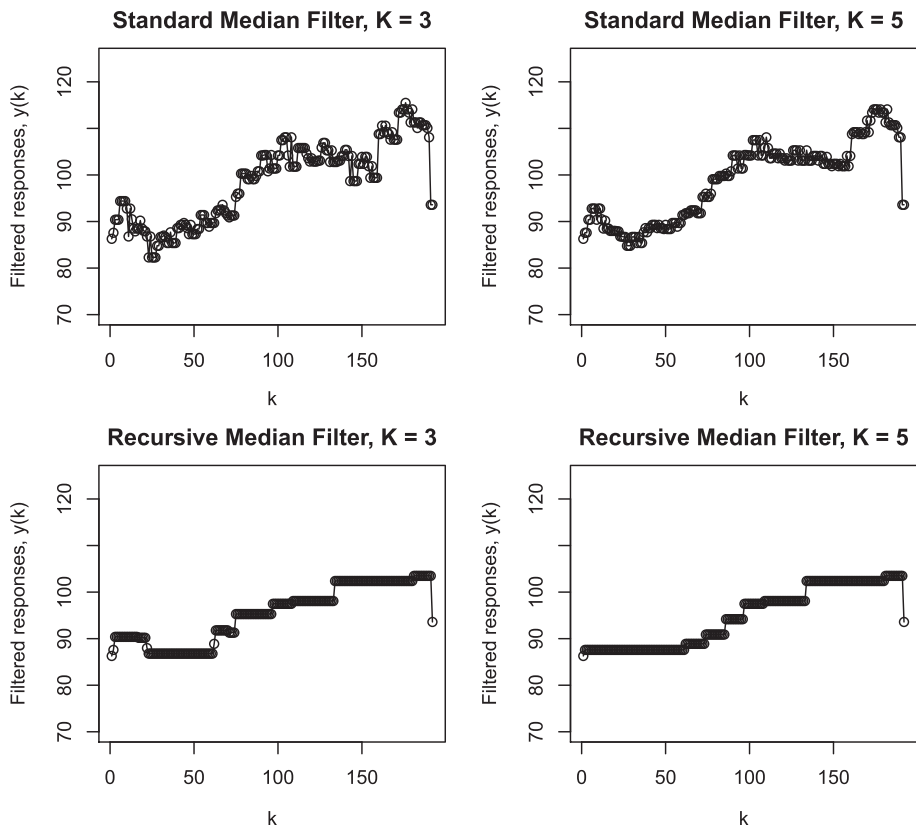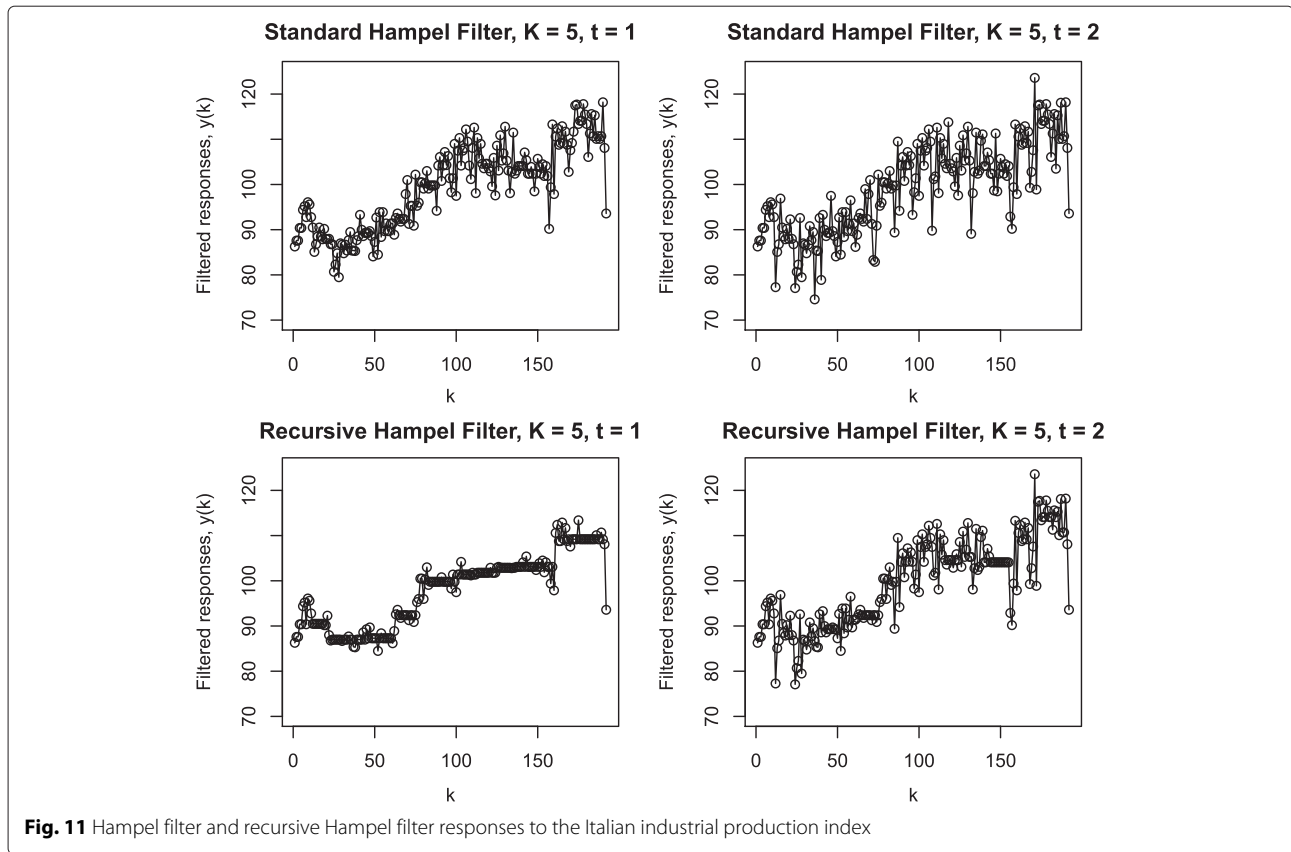
Fig. 11 Hampel filter and recursive Hampel filter responses to the Italian industrial production index

less aggressive value $t = 1$. Here, intermediate and high-frequency details are better preserved than in the median filter cascade results shown to the left, giving a result with fewer "flat streches" than seen for the recursive Hampel filter with $t = 1$ shown in the lower left plot in Fig. 11. The lower left plot in Fig. 12 represents the next step in this general trend, keeping the same basic cascade structure as in the previous two examples, but further increasing the threshold parameter for both filter components to $t = 2$. Interestingly, this cascade filter response preserves much less of the original nominal signal detail than the recursive Hampel filter with $K = 5$, as may be seen by comparing this result with the lower right plot in Fig. 11. Finally, the lower right plot in Fig. 12 shows the results of a similar cascade, but with the threshold of the recursive filter reduced from $t = 2$ as in the lower left plot to $t = 1$. As expected, by making this second cascade component more aggressive, we further attenuate many of the original signal details relative to the response shown in the lower left plot, but not nearly as much as in the still more aggressive cascade shown in the upper right plot directly above. The key point of this example is to demonstrate that cascade interconnection of simpler generalized Hampel filter components can significantly expand the range of possible filter behavior.

The final result presented here considers a filter that is not a member of the generalized Hampel family, but is conceptually similar in an important sense. Specificallly, recall that the basic idea behind the Hampel filter is to consider the central point in the moving data window and determine whether it is "anomalous:" if so, it is replaced with the "more reasonable" median value computed from the data window; otherwise, it is left unmodified. The A_n filter described by Rohwer ([12] p. 37) is based on a similar idea, but with a different definition of "anomalous" and a different replacement value for these points. This filter belongs to the LULU family, described briefly here; for a more detailed introduction, refer to Rohwer's book [12]. A less detailed introduction to these filters is also given in the book by Pearson and Gabbouj ([13] Section 6.2.3), which also provides *Python* implementations in the *NonlinearDigitalFilters* module.

The LULU filter class consists of filters constructed from cascade interconnections of the following two asymmetric moving window operators:

$$\bigvee_{K} \{x_k\} = \max\{x_k, x_{k+1}, \ldots, x_{k+K}\}$$

$$\bigwedge_{K} \{x_k\} = \min\{x_{k-K}, \ldots, x_{k-1}, x_k\}. \tag{13}$$

Pearson *et al. EURASIP Journal on Advances in Signal Processing* (2016) 2016:87
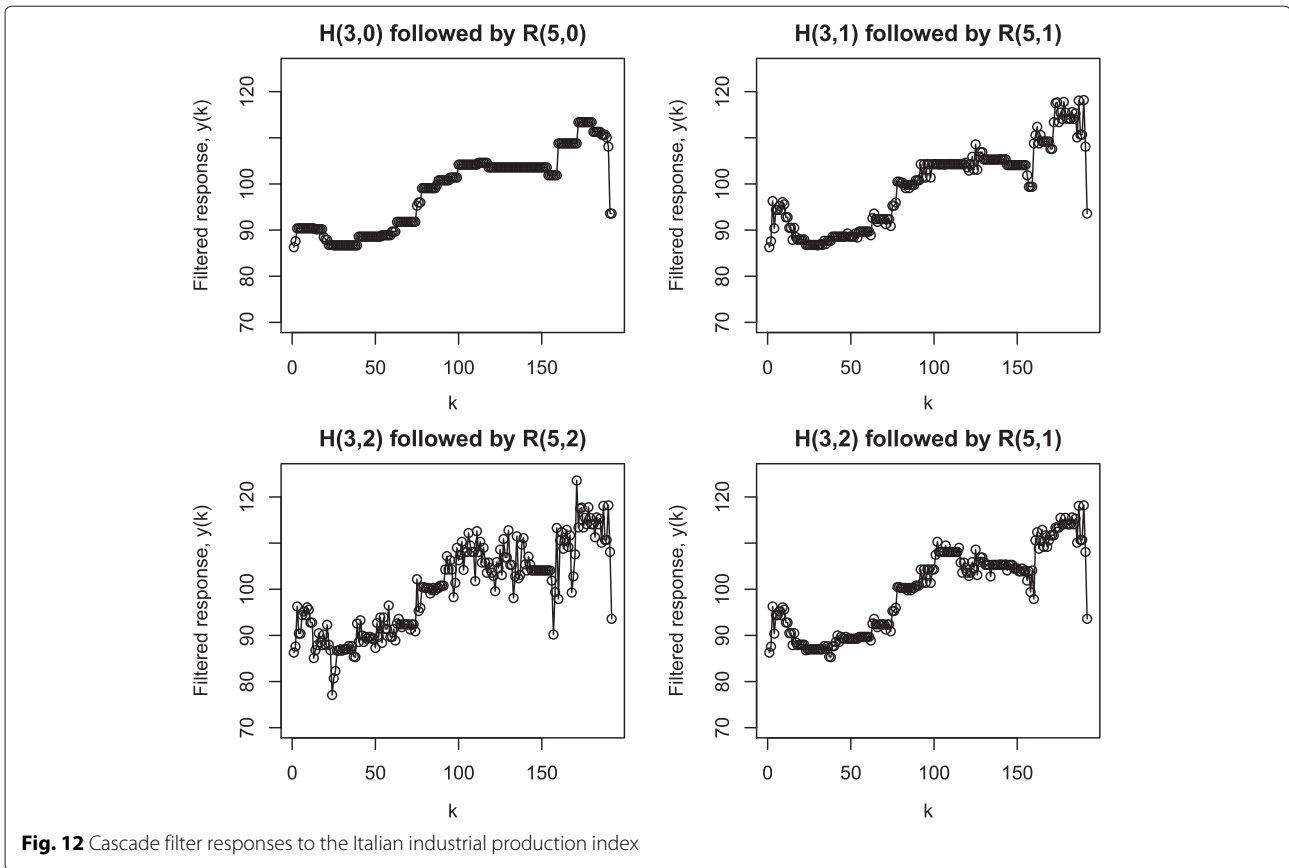
Page 14 of 18

Fig. 12 Cascade filter responses to the Italian industrial production index

Members of the LULU family consist of cascade interconnections of the component filters L_n and U_n built from these two operators:

$$L_K\{x_k\} = \bigvee_K \circ \bigwedge_K \{x_k\},$$

$$U_K\{x_k\} = \bigwedge_k \circ \bigvee_K \{x_k\}, \tag{14}$$

where the composition operator $\circ$ represents cascade interconnection, with the operator to the right of the $\circ$ symbol applied to the raw input signal, and the operator to the left of the $\circ$ symbol applied to the output of the first filter. It is not difficult to show that the filters L_K and U_K are symmetric moving window filters with window half-width parameter K, and it is traditional to drop the $\circ$ symbol when indicating cascades of these filter components. Rohwer shows that the response of the cascade filter $U_K L_K$ is a pointwise lower bound on the response of the standard median filter with half-width K, and that the response of the cascade filter $L_K U_K$ is a pointwise upper bound ([12] p. 23):

$$U_K L_K\{x_k\} \leq M_K\{x_k\} \leq L_K U_K\{x_k\}. \tag{15}$$

In fact, these filter responses are also lower and upper bounds on the response of the recursive median filter ([12]

p. 36). These observations motivate the definition of the A_n filter considered here, defined in a very similar spirit to the Hampel filter ([12] p. 37): if the central point x_k in the data window falls between the $U_K L_K$ and $L_K U_K$ bounds, the filter output is simply x_k, unmodified; otherwise, the filter output is the average of the upper and lower bounds.

Figure 13 shows the response of the A_5 filter applied to the Italian industrial production index data sequence, indicated as solid triangles. To see how this filter modifies the original signal, the original signal values are also shown on the plot as open circles, overlaid on a dotted line. Note that because the vertical axis limits cut off the lowest-valued seasonal outliers in the original data sequence, not all of these points are shown, but a few of the seasonal outliers are evident, including the one near the end of the sequence that the filter passes umodified. Also, note that most of most extreme non-outlying downward excursions in the original signal are modified by this filter, as are many of the largest upward excursions.

For comparison, Fig. 14 shows the corresponding results for the Hampel filter with $K = 5$ and threshold parameter $t = 2$, in the same format as Fig. 13. Note that here, none of the seasonal outliers are passed by the filter, most of the most extreme non-outlying lower excursions of the signal are left unmodified, as are all but one of the largest
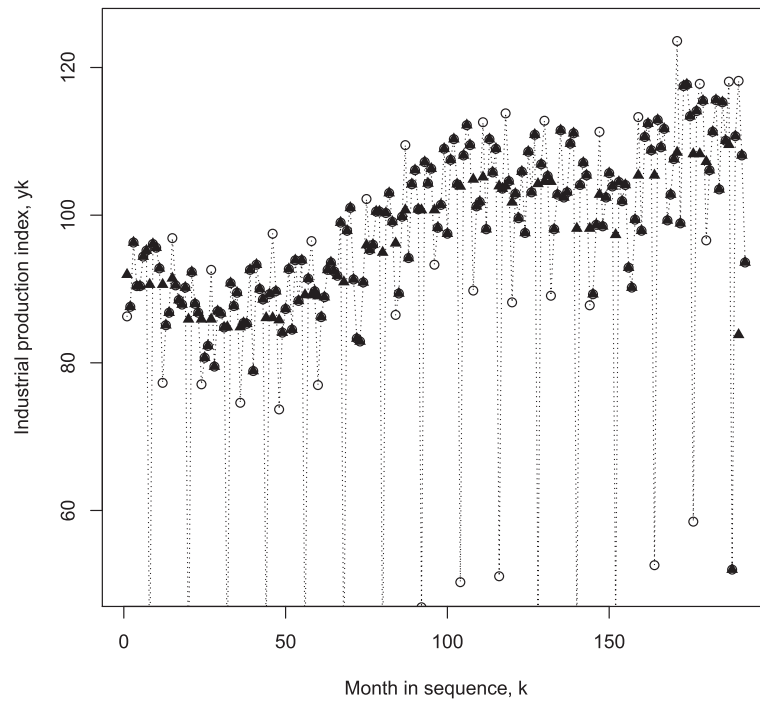
Pearson *et al. EURASIP Journal on Advances in Signal Processing* (2016) 2016:87

Page 15 of 18

Fig. 13 A_5 filter applied to the monthly Italian industrial production index (*solid triangles*), overlaid on the original signal (*dotted lines with open circles*)
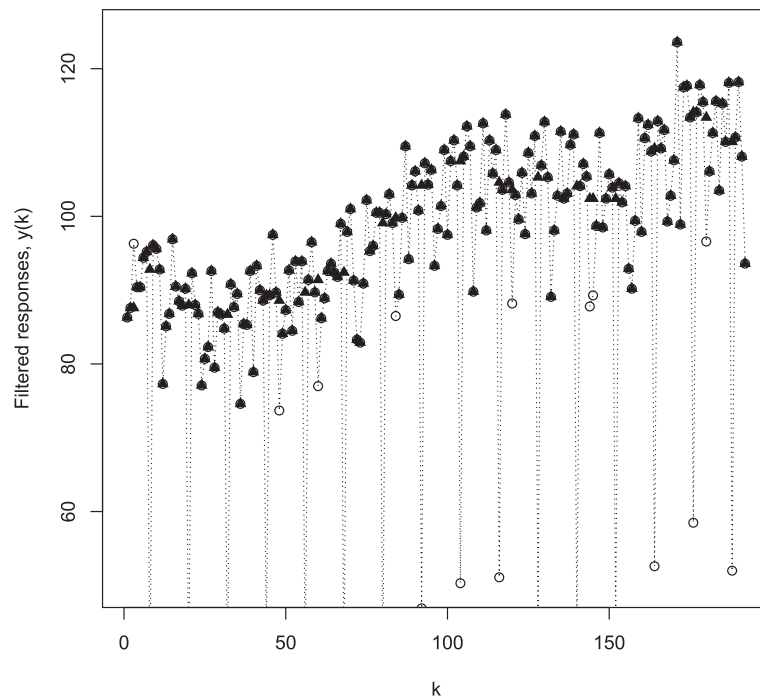

Fig. 14 Hampel filter ($K = 5$, $t = 2$) applied to the monthly Italian industrial production index (*solid triangles*), overlaid on the original signal (*dotted lines with open circles*)

upper excursions. In fact, careful comparisons of all of the filter results presented here show that, if our filtering objective is to remove *only* the seasonal outliers and leave the rest of the signal unmodified, the Hampel filter with $K = 5$ and $t = 2$ gives the best performance of any of the filters considered here. It is important to emphasize, however, that analogous results cannot be expected to hold in all situations. For example, in cases where some degree of smoothing of the low-level noise is also desirable, cascade filters like that shown in the lower right plot in Fig. 12 may be much better choices. The key point of this example has been to illustrate, first, the range of behavior possible when applying various members of the generalized Hampel filter class to a real data sequence, and second, to provide a comparison with a useful data cleaning filter that does not belong to this class.

7 Other generalizations of the Hampel filter

7.1 Weighted filters

Weighted median filters are obtained by replacing the moving window $\mathbf{W}_k^K$ defined in Eq. (1) with the following weighted data window:

$$\mathbf{Q}_k = \{w_{-K} \diamond x_{k-K}, \ldots, w_0 \diamond x_k, \ldots, w_K \diamond x_{k+K}\}, \quad (16)$$

where the operator $\diamond$ denotes replication ($m \diamond x_j$ creates a set with the data value x_j replicated m times), and $\{w_{-K}, \ldots, w_0, \ldots, w_K\}$ represents a sequence of positive integer weights. This extension greatly increases the median filter's flexibility, but it also greatly complicates the analysis of filter characteristics; for example, no complete characterization of the root sequences of arbitrarily weighted median filters is known. For a more detailed discussion of this filter class and what is known about it, refer to the survey paper by Yin et al. [14].

The *weighted Hampel filter* is defined by replacing the original data window $\mathbf{W}_k^K$ in the definition of the standard Hampel filter with the weighted window $\mathbf{Q}_k$ defined in Eq. (16). Specifically, the weighted Hampel filter is defined by:

$$y_k = \begin{cases} x_k & |x_k - m_k(Q)| \leq tS_k(Q), \\ m_k & |x_k - m_k(Q)| > tS_k(Q), \end{cases} \quad (17)$$

where $m_k(Q)$ is the median of the weighted window $\mathbf{Q}_k$ and $S_k(Q)$ is the corresponding MAD scale estimator. As with the standard Hampel filter, note that the weighted Hampel filter reduces to the weighted median filter for $t = 0$, and the root sequence nesting condition for these filters—for fixed weights—follows as before: $s \leq t$ implies $\mathcal{R}_s \subset \mathcal{R}_t$. Similarly, the concept of implosion sequences introduced in Section 3 also applies to the weighted Hampel filters, but the conditions for $\{x_k\}$ to be an implosion sequence now depend on the filter weights $\{w_k\}$. Given the lack of a general characterization for weighted median filter root sequences noted above and the strong

connection between standard Hampel filter implosion sequences and standard median filter roots shown in Section 3, it is likely that a complete characterization of weighted Hampel filter implosion sequences will be challenging.

7.2 Weighted recursive filters

The class of weighted recursive median filters is obtained by adopting both of the modifications just described: using the recursive moving window $\mathbf{R}_k^K$ defined in Section 4 with the weight-based replication scheme described in Section 7.1. Specifically, the resulting moving data window has the form:

$$\mathbf{Z}_k = \{w_{-K} \diamond y_{k-K}, \ldots, w_0 \diamond x_k, \ldots, w_K \diamond x_{k+K}\}, \quad (18)$$

where y_{k-j} is the output of the weighted median filter at prior sample $k - j$. Since this median filter generalization includes both of the previous ones as proper subsets, the flexibility of this class is even greater, as is the complexity of its analysis. The survey paper by Yin et al. also includes a discussion of these filters [14].

The *weighted recursive Hampel filter* is defined by replacing the original data window $\mathbf{R}_k^{t,K}$ in the definition of the recursive Hampel filter with the weighted window $\mathbf{Z}_k$ defined in Eq. (18) where y_{k-j} is the output of the weighted Hampel filter defined in Section 7.1 at time $k-j$. Specifically, the weighted Hampel filter is defined by:

$$y_k = \begin{cases} x_k & |x_k - m_k(Z)| \leq tS_k(Z), \\ m_k & |x_k - m_k(Z)| > tS_k(Z), \end{cases} \quad (19)$$

where $m_k(Z)$ is the median of the recursive weighted window $\mathbf{Z}_k$ and $S_k(Z)$ is the corresponding MAD scale estimator. It follows by the reasoning presented in Section 4 that the recursive weighted Hampel filter root sets are identical with the non-recursive weighted Hampel filter root sets, and that the recursive weighted Hampel filter root sets nest for increasing threshold parameters t. Again, it is likely that complete characterizations of the weighted recursive Hampel filter root sequences and implosion sequences will be challenging.

7.3 Extensions to image processing

A detailed discussion of the extension of the one-dimensional generalized Hampel filters discussed here to image processing applications is beyond the scope of this paper, but this extension is important enough to warrant a brief discussion. All of the filters defined in this paper can be extended to two-dimensional images in at least two different ways. The first and simpler is analogous to that described in Section 1.3.3 of the book by Astola and Kuosmanen [3]: the one-dimensional moving window

Pearson *et al. EURASIP Journal on Advances in Signal Processing* (2016) 2016:87

Page 17 of 18

considered here can be replaced by a square $(2K + 1) \times (2K + 1)$ two-dimensional window that is moved across the image. The median and MAD scale estimate can then be computed from these $(2K+1)^2$ pixel intensities exactly as in the one-dimensional case, and the same logic applied as before: if the central point in the data window lies more than t times the MAD scale estimate from the median value, the filter's output is the median value; otherwise, the filter's output is the unmodified central value. As in the one-dimensional case, setting $t = 0$ reduces this filter to the two-dimensional median filter, and increasing t makes the filter less aggressive.

Two-dimensional recursive filters are also possible, generalizing the two-dimensional recursive median filter, although as noted by Astola and Kuosmanen, the results obtained with this filter will depend on the order in which the pixels are processed ([3] p. 203). That is, since there is no unique total order on the points in an image, it is necessary to impose such an order for the "prior filter outputs" required in a recursive filter implementation to be well-defined. This can be done in different ways (e.g., left-to-right lexical order, top-to-bottom lexical order, etc.), generally yielding different results.

Finally, an alternative approach is to construct multi-stage Hampel image processing filters that combine the outputs of subfilters like those discussed by Nieminen and Neuvo [15], corresponding to vertical, horizontal, diagonal, cross- or x-shaped subwindows applied to the image. This general construction is described in Section 3.7 of the book by Astola and Kuosmanen [3], and it can also be readily extended to generalized Hampel filters by simply replacing the median filters defined on these subwindows with the corresponding Hampel filters.

8 Conclusions

The Hampel filter introduced in Section 2 is effectively a moving window outlier detector that replaces the original signal value with the median filter response if that value is deemed an outlier. This determination is based on a threshold parameter t chosen by the user and the MAD scale estimate for the moving window, and the filter reduces to the standard median filter if $t = 0$. The central idea of this paper was to view the Hampel filter as a generalization of the median filter and ask what the consequences of this generalization are, first for the standard Hampel filter and then for novel extensions like the recursive Hampel filter. One important aspect of this investigation was the partial characterization in Section 3 of *implosion sequences*, for which this generalization has no effect: these are sequences for which the response of the Hampel filter is independent of t. In addition, it was shown that Hampel filter root sequences nest, with the median filter root set included in all Hampel filter root sets. Thus, the input sequences of greatest interest

here are neither implosion sequences nor root sequences, where the Hampel filter may be tuned from its most aggressive limit ($t = 0$, corresponding to the median filter) to an identity filter for sufficiently large t.

A detailed description of the recursive Hampel filter was given in Section 4, where it was shown that this filter's root set for each t is the same as the standard Hampel filter root set for the same value of t, generalizing the well-known result for the recursive median filter [7]. One of the interesting characteristics of the recursive median filter is its idempotence—the fact that it reduces any input sequence to a root sequence in a single pass—and an intriguing question is whether this behavior extends to the recursive Hampel filter for any $t > 0$.

Section 5 presented a brief simulation-based case study exploring the performance of the standard and recursive Hampel filters as a function of t for a simulated signal sequence that was neither a median filter root sequence nor an implosion sequence. More specifically, this signal consisted of a median filter root sequence with three additional components superimposed on it: low-level Gaussian noise for one part of the signal, a sinusoid for another part of the signal, and impulsive noise spikes. Two performance measures were considered—RMSE and MAE—for two signal recovery problems: impulsive noise removal, and a complete noise removal problem that also attempted to remove low-level Gaussian noise from the signal. Not surprisingly, performance was much better for the impulsive noise removal problem, but the real point of this example was to provide specific illustrations of how much performance does depend on t, and how strongly this dependence varies between different problem formulations and signal characteristics (e.g., different signal subsequences exhibiting different combinations of the components listed above).

To provide a more representative illustration of the performance of generalized Hampel filters, Section 6 applied several members of this filter class to a monthly Italian industrial production index series that contains glaring outliers every 12 months (*seasonal outliers* [9]). The filters applied to this example included the standard and recursive median filters for two different window half-width parameters, both standard and recursive Hampel filters, and four cascade interconnections of filters from the generalized Hampel family. If our objective is simply the removal of the seasonal outliers, it appears that the standard Hampel filter with a sufficiently large threshold parameter t is the optimum choice here, but one of the points illustrated by these filtering results was that cascade interconnections of Hampel and recursive Hampel filters exhibit smoothing behavior that is much less extreme than that of the recursive median filter and which may be advantageous in some applications. For comparison, results were also presented for a promising data

Pearson *et al. EURASIP Journal on Advances in Signal Processing* (2016) 2016:87

Page 18 of 18

cleaning filter that is not a member of the generalized Hampel family: the A_n filter defined by Rohwer ([12] p. 37) from the LULU filter family. For this example, the A_n filter was not sufficiently aggressive, failing to eliminate the least extreme of the seasonal outliers in the data sequence, but again, it is important to emphasize that the "best" filter can be expected to depend strongly on the details of the application.

Finally, three other generalizations of the Hampel filter were described briefly in Section 7: the weighted Hampel filter, the recursive weighted Hampel filter, and extensions to two-dimensional image processing applications. The first two of these filters are generalizations of the weighted median filter and the recursive weighted median filter, respectively, which are more difficult to characterize than their non-weighted counterparts. For this reason, characterizations of roots, implosion sequences, and other performance characteristics of these generalized Hampel filters appears likely to be much more challenging than the corresponding characterizations of the standard and Hampel recursive filters. Finally, while a detailed treatment of image processing applications is beyond the scope of this paper, the one-dimensional filters described here can all be extended to these applications in much the same way as median filters have been.

References
1. GL Sicuranza, A Carini, in *Festschrift in Honor of Jaakko Astola on the Occasion of his 60th Birthday*, ed. by I Tabus, K Egiazarian, and M Gabbouj. On a class of nonlinear filters (Tampere International Center for Signal Processing, Tampere, Finland, 2009), pp. 115–143
2. JW Tukey, Nonlinear (nonsuperposable) methods for smoothing data. Abstracts, Cong. Rec. EASCON-74, 673 (1974)
3. J Astola, P Kuosmanen, *Fundamentals of nonlinear digital filtering*. (CRC Press, Boca Raton, FL, USA, 1997)
4. L Davies, U Gather, The identification of multiple outliers. J. Am. Stat. Assoc. **88**, 782–801 (1993)
5. J Astola, P Heinonen, Y Nuevo, On root structures of median and median-type filters. IEEE Trans. Acoustics Speech Signal Proc. **35**, 1199–1201 (1987)
6. NC Gallagher, GL Wise, A theoretical analysis of the properties of median filters. IEEE Trans. Acoustics, Speech, Signal Proc. **29**, 1136–1141 (1981)
7. TA Nodes, NC Gallagher, Median filters: some modifications and their properties. IEEE Trans. Acoust. Speech Signal Proc. **30**, 739–746 (1982)
8. López-de-Lacalle J, tsoutliers: Detection of Outliers in Time Series. *R* package version 0.6 (2015). https://CRAN.R-project.org/package=tsoutliers
9. R Kaiser, A Maravall, Seasonal outliers in time series, Banco de Espana, Servicio de Estudios. Working paper number, 9915 (1999). http://www.bde.es/f/webbde/SES/Secciones/Publicaciones/PublicacionesSeriadas/DocumentosTrabajo/99/Fic/dt9915e.pdf
10. RD Martin, VJ Yohai, Influence Functionals for time series. Ann. Stat. **14**, 781–818 (1986)
11. RD Martin, DJ Thomson, Robust-resistant spectrum estimation. Proc. IEEE. **70**, 1097–1114 (1982)
12. Rohwer C, *Nonlinear smoothing and multiresolution Analysis. (Birkhäuser)*, (Basel, CH, 2005)
13. RK Pearson, M Gabbouj, *Nonlinear digital filtering with Python*. (CRC Press, Boca Raton, FL, USA, 2016)
14. L Yin, R Yang, M Gabbouj, Y Neuvo, Weighted median filters: a tutorial. IEEE Trans. Circ. Syst. II: Analog Digital Signal Process. **43**, 157–192 (1996)
15. A Nieminen, Y Neuvo, Comments of theoretical anaysis of the max/median filter. IEEE Trans. Acoust. Speech Signal Proc. **36**, 826–827 (1988)