Algebraic Characterizations of Nonlinear Digital Filters

Ronald K. Pearson
The Travelers Companies, Inc.
Hartford, CT USA

Moncef Gabbouj
Tampere University of Technology
Tampere, Finland

Abstract

The book, *Fundamentals of Nonlinear Digital Filtering*, by Jaako Astola and Pauli Kuosmanen presents both a useful selection guide for practitioners seeking a signal processing solution and a valuable panoramic perspective for theoreticians interested in the underlying principles on which nonlinear digital filters are based. For example, the results presented in their book make it clear that most popular nonlinear filters exhibit homogeneous scaling behavior, and are mostly based on non-smooth (i.e., "median-like") functions, rendering the Taylor series expansions popular in many engineering disciplines largely useless. What is less obvious is that these two characteristics are closely related, a result that comes from the theory of functional equations. This paper examines a range of algebraic ideas (e.g., categories, groupoids, clones, etc.) that can be useful in characterizing and designing nonlinear filters, building in part on our past joint work with Jaakko Astola.

1 Introduction

The mathematical notion of linearity has had a profound impact on engineering and the sciences, greatly facilitating both our ability to understand a range of important phenomena and our ability to design a variety of useful systems. This is particularly true in electrical engineering, where linearity has laid the foundation for both linear circuit analysis and linear systems theory. A case in point is the classical theory of digital signal processing, much of which depends essentially on linearity assumptions (e.g., the whole notion of frequency-domain characterization of linear digital filters) [32]. Conversely, there are important cases where linear characterizations are inadequate, such as the problem of designing impulsive noise removal filters discussed in Section 2. This has led to the development of nonlinear digital filters like those described in the book by Astola and Kuosmanen [7].

Mathemtaically, the notion of linearity is an extremely convenient one, forming the basis for the theories of linear ordinary and partial differential equations, linear vector spaces, and linear algebra. In the case of digital filters, linearity can be defined either structurally (giving an explicit prescription for their implementation) or behaviorally (giving an extremely useful basis for their characterization), and these two approaches are essentially equivalent, a point discussed further in Section 2. In contrast, the term "nonlinearity" strictly defines a *lack of structure* or a *lack of behavior* without providing any alternative structural or behavioral characteristics with which to work. Hence, the analysis of "nonlinear systems" is extremely challenging: there is, for example, no mathematical theory of "nonlinear algebra" comparable to that of linear algebra. One specific consequence of this state of affairs has been extensive reliance on simulation studies to understand the performance of specific filters in specific settings. For example, Velleman [45] described the median filter and a number of other simple nonlinear smoothing filters and summarized the state of affairs in 1977 as follows:

> "We do not yet have a deep mathematical understanding of the smoothers presented in this paper, but we have empirical evidence of their good performance from both computer simulation studies and practical data analysis experience."

Conversely, it is possible to obtain practically and theoretically useful results if we restrict consideration to nonlinear filter classes based on specific structures like median filters [21] or weighted median filters [49]. Similarly, it is also possible to exploit specific nonlinear *behaviors* like the notion of homogeneity discussed in Section 3: imposing this behavior as a constraint, we can identify compatible nonlinear structures and use them as the basis for designing useful digital filters. The basic objective of this paper is to identify some useful, *specific* forms of "nonlinear algebra"—like homogeneity—that can be used to design and characterize nonlinear digital filters. The results presented here build in a broad way on Jaakko Astola's work in the area of nonlinear digital signal processing as described in his book with Pauli Kuosmanen [7] and a number of other references (e.g., [5, 6, 50]), on the interest he has expressed in algebraic characterizations of nonlinear digital filters in private conversations, and on our prior collaborations with him on closely related topics [20, 36, 39].

2 The class of nonlinear digital filters

To keep the scope manageable, this paper restricts consideration primarily to the problems of designing and characterizing nonlinear digital filters that process a univariate, real-valued data sequence $\{x_k\}$ of finite length, defined for $k = 1, 2, \ldots, N$. Extensions to two-dimensional sequences that arise in applications like image processing are discussed briefly in Section 10, along with extensions to other data types (e.g., characterization of DNA or protein sequences described by finite, discrete alphabets). As is customary in the digital signal processing literature, the sequences $\{x_k\}$ considered here are treated essentially as uniformly spaced samples of a continuous signal $x(t)$, taken at times $t_k = t_0 + k\Delta$ for some constant intersample spacing $\Delta > 0$. In this context, the objective of a digital filter $\mathcal{F}$ (linear or nonlinear) is to map the finite sequence $\{x_k\}$ into a second, related sequence $\{y_k\}$ that is "better behaved" in some useful, application-specific sense than the original.

In the case of linear digital filters, attention is usually restricted to the class of *time-invariant* or *shift-invariant* filters that satisfy the following condition:

$$\mathcal{F}\{x_k\} = y_k \;\Rightarrow\; \mathcal{F}\{x_{k-j}\} = y_{k-j}, \tag{1}$$

for all integer j, positive or negative. We impose the same restriction here, allowing us to characterize the filters discussed in this article with the following explicit representation:

$$y_k = \mathcal{F}\{x_k\} = \Phi(y_{k-1}, y_{k-2}, \ldots, y_{k-p}, x_{k+n}, x_{k+n+1}, \ldots, x_{k+m-1}, x_{k+m}), \tag{2}$$

where p is an arbitrary non-negative integer, n and m are integers that may be positive or negative but must satisfy $m \geq n$, and $\Phi : R^{p+m-n+1} \rightarrow R$ is an arbitrary mapping.

In words, Eq. (2) means that the filter's response at time k, y_k, can depend on previous responses, y_{k-j}, and on a fixed but otherwise arbitrary finite portion of the input sequence $\{x_k\}$. In this paper, we will impose two restrictions on the integers m, n, and p, both of which have important theoretical and practical consequences but both of which are met by the majority of nonlinear digital filters now in common use, including almost all of those discussed in the books by Astola and Kuosmanen [7], Mathews and Sicuranza [29], and Pitas and Venetsanopoulos [41]. The first of these restrictions is to require $p = 0$, eliminating the explicit dependence in Eq. (2) of the current filter output y_k on prior outputs, y_{k-j}. This requirement restricts our attention to the subclass of *nonrecursive* digital filters, greatly simplifying stability considerations which are much more complicated in the case of nonlinear filters than they are in the linear case [33, 34]. Perhaps the nonlinear filter of greatest practical importance that is excluded by this restriction is the recursive median filter [31], which is useful in part because of its very special stability properties [34].

The second restriction is to the class of *symmetric moving window filters*, for which $m > 0$ and $n = -m$. Combining this requirement with the nonrecursive structure restriction $p = 0$, Eq. (2) simplifies to:

$$y_k = \Phi(x_{k-m}, \ldots, x_k, \ldots, x_{k+m}) = \Phi(\mathbf{w}_k), \tag{3}$$

where $\mathbf{w}_k$ denotes the moving window defined by samples x_{k-m} through x_{k+m}. Note that the resulting class of filters is *non-causal* since the current output y_k depends on both past inputs x_{k-j} and future inputs x_{k+j}. In typical off-line signal processing applications (e.g., time-series data pre-processing for spectral estimation or dynamic modeling), this lack of causality poses no difficulties, but real-time applications (e.g., industrial process control) may require filters that are causal (implying $n \geq 0$ in Eq. (2)) or even strictly causal (implying $n > 0$). In some cases, causality or strict causality may be achieved by simply introducing a fixed delay in the filter's response, although this modification can have nontrivial consequences. As a specific example, the root sequences for the causal median filter are *much* different from those for the symmetric median filter, a result that has important practical consequences for real-time data cleaning applications [30].

2.1 Linear filters

Since they serve as the essential counterweight around which nonlinear filters are defined, it is worth briefly discussing the linear reference case. As noted in Section 1, linear filters may be defined either behaviorally or structurally; while these definitions are essentially equivalent in the linear case, they form the basis for very different nonlinear generalizations. Behaviorally, the filter $\mathcal{L}$ is linear if it satisfies the *principle of superposition*:

$$\mathcal{L}\{\alpha x_k + \beta y_k\} = \alpha \mathcal{L}\{x_k\} + \beta \mathcal{L}\{y_k\}, \tag{4}$$

for all real numbers α and β and all real-valued sequences $\{x_k\}$ and $\{y_k\}$. Note that this description does not tell us directly how to construct the linear filter $\mathcal{L}$, but it does tell us how it behaves. It is possible, however, to obtain a constructive (i.e., structural) representation from this behavioral description. To do this, define the discrete impulse as:

$$\delta_k = \begin{cases} 1 & k = 0, \\ 0 & k \neq 0, \end{cases} \tag{5}$$

and note that any data sequence $\{x_k\}$ may be rewritten as the following weighted sum of impulses:

$$x_k = \sum_{j=-\infty}^{\infty} x_j \delta_{k-j}. \tag{6}$$

Substituting this result into Eq. (4), it follows that

$$\mathcal{L}\{x_k\} = \mathcal{L}\left(\sum_{j=-\infty}^{\infty} x_j \delta_{k-j} \right) = \sum_{j=-\infty}^{\infty} x_j h_{k-j}, \tag{7}$$

where the sequence $\{h_k\}$ is the *impulse response* of the filter $\mathcal{L}$, defined as its response to the unit impulse sequence $\{\delta_k\}$. The primary practical consequence of this result is the well-known one that any linear, time-invariant filter is completely characterized by its impulse response sequence $\{h_k\}$. Thus, at least at a high level, linear filter design may be viewed as the specification of useful impulse response sequences; this view overlooks important implementation details like coeficient sensitivity, but a key point of this article is that the corresponding high level specification problem for nonlinear filters is much more complicated.

2.2 Impulsive noise filters

As an example of a simple but practically important application where linear filters are inadequate but many nonlinear solutions exist, consider the problem of impulsive noise removal from a sequence $\{x_k\}$. Impulsive noise consists of "spikes" or *outliers* in the data sequence, characterized as isolated points x_j whose value lies far from those of neighboring values. Note that such a sequence may be decomposed into the sum $x_k = x_k^0 + o_k$, where $\{x_k^0\}$ is the nominal part of the sequence (i.e., the "true signal" we wish to observe) and $\{o_k\}$ is the impulsive noise part. In the case of a single outlier, this is given by:

$$o_k = \begin{cases} A & k = j, \\ 0 & k \neq j, \end{cases} \tag{8}$$

where A is the magnitude of the impulsive contamination, assumed here to be large relative to the range of variation seen in the nominal data sequence $\{x_k^0\}$. Suppose we apply a linear filter $\mathcal{L}$ with impulse response sequence $\{h_k\}$ to the contaminated sequence $\{x_k\}$. The filter response is given by:

$$\mathcal{L}\{x_k\} = \sum_{j=-\infty}^{\infty} x_j^0 h_{j-k} + Ah_k. \tag{9}$$

For an ideal data cleaning filter, we would like the first term to be equal to x_k^0 for all k and the second term to be zero for all k, but these desires are incompatible: the best linear filter for preserving the first term is the identity filter ($h_k = \delta_k$), which exactly preserves all sequences, and the best linear filter for eliminating the second term is the zero filter ($h_k = 0$ for all k), which eliminates all sequences.

In marked contrast, there exists a wide variety of nonlinear filters that are capable of completely eliminating impulsive noise from certain classes of sequences. The simplest example is the *median filter*, defined by:

$$\Phi(x_{k-m}, \ldots, x_k, \ldots, x_{k+m}) = \text{median}\{x_{k-m}, \ldots, x_k, \ldots, x_{k+m}\}, \tag{10}$$

where the median is defined as the central order statistic $x_{(0)}$ in the rank-ordered sequence:

$$\{x_{k-m}, \ldots, x_k, \ldots, x_{k+m}\} \;\rightarrow\; \{x_{(-m)} \leq x_{(-m+1)} \leq \cdots \leq x_{(0)} \leq \cdots \leq x_{(m-1)} \leq x_{(m)}\}. \tag{11}$$

In particular, it is a standard result that the median filter completely rejects isolated impulses while preserving all sequences in a known class of *root sequences* [6, 7, 14, 21, 31]. Indeed, the median filter is one of a group of nonlinear digital filters whose impulse responses are identically zero, emphasizing that while the impulse response completely characterizes linear filters, it gives essentially no information about many nonlinear filters.

This last observation provides a specific illustration of both the advantages and disadvantages of nonlinear digital filters: there are certain applications (like impulsive noise rejection) that demand nonlinear solutions and for which effective nonlinear solutions exist, but reasonably complete characterizations of these nonlinear filters are largely lacking, greatly complicating their design and use.

2.3 The issue of finite sequence length

An important practical question that arises in applying moving window filters—whether linear or nonlinear—is how to handle "end effects." That is, given a sequence $\{x_k\}$ of finite length N, the output of any filter specified by Eq. (3) is not well-defined for $k < m + 1$ or for $k > N - m$. This problem can be addressed in various ways, none of them entirely satisfactory, but the approach taken here is probably the most popular. Specifically, we artificially append m additional copies of the initial value x_1 to the beginning of the sequence as elements x_{-m+1} through x_0, and we append m additional copies of the final value x_N to the end of the sequence, as elements x_{N+1} through x_{N+m}. Like the restrictions to nonrecursive filters and symmetric moving windows imposed above, this strategy for dealing with finite sequence effects can have unexpected consequences, such as the existence of oscillatory binary root sequences for the median filter [6, 14].

3 Homogeneous nonlinear filters

The behavioral characterization of linearity given in Section 2.1 for linear filters may be expressed in terms of two component parts: the filter $\mathcal{F}$ is *additive* if

$$\mathcal{F}\{x_k + y_k\} = \mathcal{F}\{x_k\} + \mathcal{F}\{y_k\}, \tag{12}$$

for all real-valued sequences $\{x_k\}$ and $\{y_k\}$, and $\mathcal{F}$ is said to be *homogeneous* if

$$\mathcal{F}\{\lambda x_k\} = \lambda \mathcal{F}\{x_k\}, \tag{13}$$

for all real λ. It follows from the theory of Cauchy's functional equation [2] that the only practical filters satisfying the additivity condition (12) are linear [33, Section 2.6]. In marked contrast, the class of nonlinear filters $\mathcal{F}$ satisfying the homogeneity condition (13) is large, including many of those discussed by Astola and Kuosmanen [7]. An even larger class results—one that encompasses almost all of the nonlinear filters described by Astola and Kuosmanen—if we require Eq. (13) to only hold for all $\lambda > 0$, corresponding to the requirement of *positive homogeneity*, also known as *scale-invariance* in the statistics literature [42, p. 159].

The class of L-filters provides a simple illustration of the distinction between these different notions. While the class can be defined more generally [7, 13, 27], it is convenient for the discussions presented here to define the class rather strictly as the following normalized linear combination of the moving-window order statistics defined in Eq. (11):

$$y_k = \mathcal{F}\{x_k\} = \sum_{i=-K}^{K} \alpha_i x_{(i)}, \quad 0 \leq \alpha_i \leq 1, \quad \sum_{i=-K}^{K} \alpha_i = 1. \tag{14}$$

Since positive scaling does not alter the ranks r_k of the data sequence $\{x_k\}$ (that is, if x_k has rank r_k in the sequence $\{x_k\}$, then λx_k has the same rank r_k in the sequence $\{\lambda x_k\}$), it follows that all filters in the L-filter class defined by Eq. (14) are positive-homogeneous. In contrast, since multiplication by negative numbers reverses the ordering inequalities in Eq. (11), it follows that scaling the original data sequence by a negative number reverses the ranks: $x_{(j)} \rightarrow \lambda x_{(-j)}$ for $\lambda < 0$. Hence, the only fully homogeneous members of the class of L-filters are the symmetric ones, with $\alpha_{-i} = \alpha_i$. Since the median filter defined in Eq. (10) is an L-filter with $\alpha_0 = 1$ and $\alpha_i = 0$ for all other i, it follows that the median filter is symmetric and thus fully homogeneous. Finally, note that since rank-ordering is a nonlinear operation, it follows that the only member of the class of L-filters that is linear is the one for which all coefficients are constant, $\alpha_i = 1/(2K+1)$ for all i, corresponding to the symmetric $2K+1$-point unweighted linear moving average smoothing filter.

Before leaving this example, it is worth noting that the class of L-filters does satisfy an extremely limited but important form of additivity, known as *location-invariance* in the statistics literature [42, p. 159]. Specifically, note that adding the same constant value c to all elements of the data sequence $\{x_k\}$ does not change the ranks (that is, changing x_k to $x_k + c$ for all k changes the ordered samples $x_{(i)}$ to $x_{(i)} + c$). Since the coefficients of the L-filter sum to 1, it follows that:

$$\mathcal{F}\{x_k + c\} = \mathcal{F}\{x_k\} + c, \tag{15}$$

for all real c. In fact, it follows from this result that the entire class of L-filters exhibits linear behavior when restricted to the set of constant input sequences. While this result is too restrictive to be practically useful, it does point the way to the more useful notion of restricted linearity introduced in Section 9.

Positive-homogeneity is perhaps the most useful generalization of the behavioral definition of linearity in current practice, but several extensions of this notion are possible and it is useful to introduce a few of them here. First, it is worth noting the following general result presented by Aczel and Dhombres [2]. The function $f : R^n \to R$ is said to exhibit *generalized homogeneity* if it satisfies the following functional equation for some function $g : R \to R$:

$$f(\lambda x_1, \lambda x_2, \ldots, \lambda x_n) = g(\lambda) f(x_1, x_2, \ldots, x_n), \tag{16}$$

for all real x_i and all $\lambda \neq 0$. Letting $\lambda = \mu\nu$ for $\mu, \nu \neq 0$ and substituting into Eq. (16), it follows that

$$g(\mu\nu) f(x_1, x_2, \ldots, x_n) = g(\mu) g(\nu) f(x_1, x_2, \ldots, x_n), \tag{17}$$

for all $\mu, \nu \neq 0$ and all real x_i. If we exclude the trivial solution $f(x_1, x_2, \ldots, x_n) = 0$ for all real x_i, it follows that $g(\cdot)$ satisfies *Cauchy's power equation*

$$g(\mu\nu) = g(\mu) g(\nu), \tag{18}$$

for all $\mu, \nu \neq 0$. Again excluding the trivial solution $g(\lambda) = 0$ for all $\lambda \neq 0$, the only possible solutions of Cauchy's power equation that are continuous at any point or bounded on any set of positive measure are [2, p. 31]:

$$g(\lambda) = |\lambda|^c \quad \text{or} \quad g(\lambda) = |\lambda|^c \text{sign}\lambda. \tag{19}$$

In other words, the only possible forms of generalized homogeneity involve power law scaling for some power c. Also, note that both of the solutions in Eq. (19) reduce to $g(\lambda) = \lambda^c$ if attention is restricted to $\lambda > 0$ (i.e., generalizations of positive homogeneity).

The definition of homogeneity given in Eq. (13) corresponds to the special case $c = 1$, the most common and perhaps the most useful form of generalized homogeneity, although higher integer powers sometimes arise (e.g., $c = 2$ or $c = 3$) [35]. On the other hand, the case $c = 0$ is particularly noteworthy as it forms the basis for a large class of nonlinear filter structures. For example, consider the class of filters defined by

$$\mathcal{F}\{x_k\} = \sum_{i=-K}^{K} \rho_j(\mathbf{w}_k) x_{k-j}, \tag{20}$$

where $\{\rho_j(\cdot)\}$ is a family of $2K+1$ positive homogeneous functions of order zero and $\mathbf{w}_k$ is the moving data window defined in Eq. (3). Thus, $\rho_j(\lambda \mathbf{w}_k) = \rho_j(\mathbf{w}_k)$ for all $\lambda > 0$ and it follows that the filter defined in Eq. (20) is positive homogeneous. An important special case that may be taken as a prototype for this construction is the class of *combination filters* discussed by Gandhi and Kassam [22]: there, $\rho_j(\mathbf{w}_k) = c(r_j, j)$ where r_j is the rank of the j^{th} sample x_j in the moving data window and $c(\cdot, \cdot)$ is an arbitrary real-valued function. The positive homogeneity of combination filters follows from the fact noted earlier that the ranks r_j are invariant under positive rescaling of the data observations x_j, implying that they are positive homogeneous functions of order zero. This construction also makes use of the fact that if $f : R^n \to R$ is a positive homogeneous function of order zero and $g : R \to R$ is any arbitrary function, the composition $g \circ f(\mathbf{x}) = g(f(\mathbf{x}))$ is also positive homogeneous of order zero. Indeed, the class of positive homogeneous functions of order zero is closed under a wide range of algebraic operations: sums, differences, products, and quotients (provided they are well-defined) of these functions retain this property. As a consequence, the class of nonlinear filters defined by Eq. (20)

is enormous and appears to be largely unexplored. Exceptions include the class of combination filters just described and the class of affine filters proposed by Arce and Hasan for suppressing unwanted cross-terms in time-frequency distributions [4].

Finally, since many practically important nonlinear filters exhibit positive homogeneity, the following result is especially noteworthy. Aczel, Gronau and Schwaiger have shown that any positive homogeneous function is differentiable at zero if and only if it is linear [3, Prop. 9]. If the function is also location-invariant, it follows immediately that differentiability *anywhere* is equivalent to linearity. Since most nonlinear filters commonly encountered in practice are both positive homogeneous and location-invariant, it follows that the functions $\Phi(\cdot)$ on which they are based are nowhere differentiable. A closely related result is that any positive homogeneous functions of order zero that is continuous at the origin is constant [35]. An important consequence of these observations is the fact that, in contrast to the situation commonly encountered in engineering, Taylor series linearization does not provide a useful basis for analyzing most nonlinear digital filters. The one noteworthy exception is the class of Volterra filters, which are based on polynomial nonlinearities that may be viewed as truncated Taylor series approximations of more general smooth nonlinear functions [7, 16, 18, 19, 29, 33, 41].

4 Cascade interconnections and categories

Cascade interconnections of simple linear filters are widely used in the modular implementation of more complex linear filters with better signal separation characteristics. An important characteristic of linear filter cascades is that the order of interconnection is immaterial: the cascade interconnection of filter $\mathcal{L}_1$ followed by filter $\mathcal{L}_2$ exhibits the same overall behavior as the cascade interconnection formed in the opposite order. This characteristic no longer holds in the nonlinear case and this observation means that the cascade interconnection of nonlinear filters is an even more powerful design tool in the nonlinear case than it is in the linear case.

An extremely useful mathematical construct for discussing cascade interconnections of nonlinear filters is category theory. A *category* is a collection of mathematical *objects*, together with sets of *morphisms* relating pairs of objects [12]. One of the most familiar examples is the category of linear vector spaces, where the objects are the vector spaces R^n and the morphisms are the $n \times m$ matrices mapping one vector space into another. To qualify as a category, a collection of objects and morphisms must also exhibit a *composition law* $\circ$ under which the morphism sets are closed: if $\mathbf{M}_1$ is a morphism relating object A to object B (written as $\mathbf{M}_1 : A \to B$) and $\mathbf{M}_2$ is a morphism relating object B to object C (i.e., $\mathbf{M}_2 : B \to C$), the composition $\mathbf{M}_2 \circ \mathbf{M}_1 : A \to C$ must be a morphism in the category. In the category of linear vector spaces just described, composition of morphisms generally corresponds to ordinary matrix multiplication, although other definitions are possible (e.g., Hadamard products of matrices), leading to different categories based on the same objects and morphisms. Further, to qualify as a category, there must exist an *identity morphism* associated with each object that relates the object to itself (i.e., $\mathbf{I}_A : A \to A$) and whose compositions with all morphisms acting on that object yields the original morphism (i.e., $\mathbf{M} \circ \mathbf{I}_A = \mathbf{M}$ and $\mathbf{I}_B \circ \mathbf{M} = \mathbf{M}$ for all morphisms $\mathbf{M} : A \to B$). Again, in the usual category of linear vector spaces, the identity morphism associated with the linear vector space R^n is simply the $n \times n$ identity matrix $\mathbf{I}_n$. Conversely, it is important to note that the identity morphisms and the composition law are intimately related; for example, if we define composition to be the Hadamard (i.e., elementwise) product of two matrices, the identity morphisms in the resulting category are not the usual identity matrices, but rather the $n \times n$ matrices $\mathbf{E}_n$ whose elements are all 1. Finally, composition of morphisms must satisfy the following associativity condition:

$$\mathbf{X} \circ (\mathbf{Y} \circ \mathbf{Z}) = (\mathbf{X} \circ \mathbf{Y}) \circ \mathbf{Z}, \tag{21}$$

for all compatably defined morphisms **X**, **Y**, and **Z**. It is not difficult to show that both ordinary matrix multiplication and Hadamard products satisfy this associativity condition.

The objects of primary interest here are real-valued sequences $\{x_k\}$, and the morphisms are filters $\mathcal{F}$ mapping one such sequence into another. Composition of morphisms is defined as the cascade interconnection of two filters:

$$[\mathcal{G} \circ \mathcal{F}]\{x_k\} = \mathcal{G}[\mathcal{F}\{x_k\}]. \tag{22}$$

The identity morphism associated with each sequence is simply the identity mapping taking the sequence into itself, and it follows directly from the associativity of the composition of functions that the composition law considered here satisfies the associativity condition (21). It is a standard result that the cascade interconnection of two linear filters yields a third linear filter, from which it follows that linear filters form a category. More generally, a family **F** of filters defined over a class $\mathcal{O}$ of sequences defines a category if the following three conditions are met:

1. $\mathcal{F}\{x_k\} \in \mathcal{O}$ for all $\{x_k\} \in \mathcal{O}$,
2. **F** must include the identity filters for all $\{x_k\} \in \mathcal{O}$,
3. **F** must be closed under cascade interconnections.

It is not difficult to show that the class of symmetric moving window filters defined in Eq. (3) satisfies all three of these conditions: these filters map one real-valued data sequence into another, the identity filter for any input sequence $\{x_k\}$ is obtained by letting $\Phi(\cdot)$ return the central element in the data window (i.e., $\Phi(\mathbf{w}_k) = x_k$), and the cascade interconnection of the symmetric moving window filters defined by $\Phi_1(\cdot)$ of half-width m_1 and $\Phi_2(\cdot)$ of half-width m_2 is the filter defined by $\Psi(\cdot)$ of half-width $m_1 + m_2$ given by:

$$\begin{aligned} \Psi(x_{k-m_1-m_2}, \ldots, x_k, \ldots, x_{k+m_1+m_2}) &= \Phi_1(\Phi_2(x_{k-m_1-m_2}, \ldots, x_{k-m_1}, \ldots, x_{k-m_1+m_2}), \\ &\quad \ldots, \Phi_2(x_{k-m_2}, \ldots, x_k, \ldots, x_{k+m_2}), \ldots, \\ &\quad \Phi_2(x_{k+m_1-m_2}, \ldots, x_{k+m_1}, \ldots, x_{k+m_1+m_2})). \end{aligned} \tag{23}$$

In the following discussion, it will be convenient to denote the category of nonlinear symmetric moving window filters defined by Eq. (3) as **NSMWF**. The practical difficulty with the category **NSMWF** is that it is so large that nothing useful can be said about it in general. It is therefore desirable to consider smaller categories, either defined structurally like **NSMWF** (i.e., every filter is defined by specifying a window half-width m and a function $\Phi : R^{2m+1} \to R$), or defined behaviorally.

While most of the nonlinear filter families that have been defined and studied to date do not by themselves define categories, any filter or collection of filters *generates* a category. In particular, given any collection $\{\mathcal{F}_i\}$ of nonlinear filters, note that the set consisting of the identity mapping together with all possible cascade interconnections of filters from $\{\mathcal{F}_i\}$ represent the morphisms in a category with cascade interconnection as the composition rule for morphisms. As a specific example, while the family of median filters does not define a category in and of itself, the set of all possible median filter cascades does define a category **MED**. Note that this category includes all median filters as morphisms, along with cascade structures like the *data sieve* described by Bangham [10].

One of the advantages of describing cascade interconnections in terms of category theory is that nonlinear filter categories can also be defined behaviorally, as noted above and as the example discussed below illustrates. The practical advantage of this observation is that, if a particular filter behavior defines a category and a collection of filters is known to exhibit that behavior, it follows immediately that the behavior is inherited by all cascade interconnections of filters in the collection. More formally, a category **S** is a *subcategory* of the category **C** if the following four conditions are satisfied [12, p.7]:

1. Every object of **S** is also an object of **C**;

2. For all objects X, Y of **S**, the set of morphisms relating X and Y in **S** is a subset of the morphisms relating X and Y in **C**;

3. Composition of morphisms $\circ$ is the same in **S** and **C**;

4. For all objets X in **S**, the associated identity morphism in **S** is the same as it is in **C**.

It is not difficult to show that if all filters $\{\mathcal{F}_i\}$ exhibit a behavioral characteristic $\mathcal{B}$ that defines a category **B**, then the category **F** generated by $\{\mathcal{F}_i\}$ is a subcategory of **B**.

As a specific illustration of a behaviorally defined category, note that the class of positive homogeneous filters defined in Section 3 defines a category, which will be denoted **PH** [33, Section 7.4.4]. To prove this, it is only necessary to show that positive homogeneity is preserved under cascade interconnection, as follows. Suppose filters $\mathcal{F}$ and $\mathcal{G}$ are positive homogeneous; then, for every $\lambda > 0$, we have:

$$\mathcal{F} \circ \mathcal{G}\{\lambda x_k\} = \mathcal{F}[\lambda \mathcal{G}\{x_k\}] = \lambda \mathcal{F}[\mathcal{G}\{x_k\}] = \lambda \mathcal{F} \circ \mathcal{G}\{x_k\}. \tag{24}$$

Since, as noted in Section 3, the median filter is positive homogeneous, it follows that the category **MED** of median filter cascades is a subcategory of **PH**, establishing that all median filter cascades (e.g., Bangham's data sieves) also exhibit positive homogeneity.

Other forms of behavior that define categories include linearity, location invariance, (full) homogeneity, homogeneity of order zero and positive homogeneity of order zero. Conversely, other well-defined forms of nonlinear behavior are not invariant under cascade interconnection and therefore do not define categories. A specific example is homogeneity of order ν for $\nu \neq 0, 1$: it is easy to show that the cascade interconnection of two homogeneous filters of order ν is homogeneous of order ν^2, which is distinct from ν unless $\nu = 0$ or 1. Another form of nonlinear behavior that does not define a category is idempotence, discussed in Section 8.

Before concluding this discussion, it is useful to note the following two general approaches to defining behavioral categories of nonlinear filters. The first is based on the notion of invariant sets: suppose $\mathcal{S}$ is a set of real-valued sequences exhibiting some particular characteristic; useful examples include boundedness, nonnegativity, or monotonicity, among many others. Now, suppose $\mathcal{F}$ and $\mathcal{G}$ are two filters that preserve this sequence property, $\mathcal{F}\mathcal{S} \subset \mathcal{S}$ and $\mathcal{G}\mathcal{S} \subset \mathcal{S}$, meaning:

$$\{x_k\} \in \mathcal{S} \;\Rightarrow\; \mathcal{F}\{x_k\} \in \mathcal{S} \;\text{ and }\; \mathcal{G}\{x_k\} \in \mathcal{S}. \tag{25}$$

It follows immediately that this behavior is preserved by cascade interconnection:

$$\mathcal{F} \circ \mathcal{G}\mathcal{S} \subset \mathcal{S}. \tag{26}$$

Since identity mappings leave any set $\mathcal{S}$ invariant, it follows that the set of all filters $\mathcal{F}$ leaving a given set $\mathcal{S}$ invariant define a category. Taking $\mathcal{S}$ as the set of all bounded sequences leads to the category **BIBO** of filters that are *bounded-input, bounded-output (BIBO) stable*, a large category that includes all symmetric moving window filters based on continuous functions $\Phi(\cdot)$ [33, p. 149].

A possibly more interesting subset-based nonlinear filter category is that obtained by taking $\mathcal{S}$ as the set of all monotone sequences. It is well known that monotone sequences are invariant under median filtering. More generally, these sequences are invariant under rank ordering, from which it follows that the L-filter introduced in Section 3 reduces to a *linear filter* when the input sequence is monotone [36], a result considered further in Section 9. Also, it is known that linear filters preserve monotonicity if and only if their impulse responses are positive [33, p. 341], so the class of L-filters defined in Eq. (14) preserve monotonicity due to the nonnegativity constraint imposed on the coefficients there. Thus, although this class of L-filters does not define a category (i.e., it is not closed under cascade interconnection), the category it generates (i.e., positive L-filters and all possible cascade interconnections) is a subcategory of **MONO**, the category of monotonicity-preserving nonlinear filters.

The second general approach to defining behavioral filter categories is based on the following observation. Suppose $\mathcal{M}$ is a sequence mapping and consider the class of all filters $\mathcal{F}$ that commutes with $\mathcal{M}$:

$$\mathcal{F} \circ \mathcal{M} = \mathcal{M} \circ \mathcal{F}. \tag{27}$$

It is not difficult to show that this construction defines a category. Specifically, note that the identity map commutes with all mappings $\mathcal{M}$ and if $\mathcal{F}$ and $\mathcal{G}$ both commute with $\mathcal{M}$, then

$$(\mathcal{F} \circ \mathcal{G}) \circ \mathcal{M} = \mathcal{F} \circ (\mathcal{G} \circ \mathcal{M}) = \mathcal{F} \circ (\mathcal{M} \circ \mathcal{G}) = (\mathcal{F} \circ \mathcal{M}) \circ \mathcal{G} = (\mathcal{M} \circ \mathcal{F}) \circ \mathcal{G} = \mathcal{M} \circ (\mathcal{F} \circ \mathcal{G}). \tag{28}$$

As a specific application of this result, let $\mathcal{M}$ represent scaling by an arbitrary positive number, $\mathcal{M}\{x_k\} = \{\lambda x_k\}$, and note that the class of filters commuting with $\mathcal{M}$ corresponds to the class of positive homogeneous filters. Similarly, if $\mathcal{M}$ represents translation by an arbitrary real number, $\mathcal{M}\{x_k\} = \{x_k + c\}$, the class of filters commuting with $\mathcal{M}$ corresponds to the class of location-invariant filters introduced in Section 3. The fact that both of these classes—the positive homogeneous filters and the location-invariant filters—define filter categories follows from the above construction.

5 Parallel interconnections and groupoids

For two linear filters, $\mathcal{L}_1$ and $\mathcal{L}_2$, the parallel interconnection is obtained by driving both filters with the same input and summing the outputs:

$$y_k = \mathcal{L}_1\{x_k\} + \mathcal{L}_2\{x_k\}. \tag{29}$$

It follows that this interconnection defines another linear filter and this construction, like the cascade interconnection structures discussed in Section 4, represents a useful way of combining simple components into more complex structures while retaining their most useful characteristics (i.e., linearity). This approach can also lead to useful nonlinear filter structures, particularly when the filters involved exhibit either homogeneity or positive homogeneity—of any order—since it is easy to show that this behavior is preserved under the parallel sum interconnection implied by Eq. (29). A case in point is the mean-median (MEM) filter described by Aysal and Barner [8].

Alternatively, if we replace the addition appearing in Eq. (29) with some other binary operation $\oplus$, we obtain a nonlinear system. A specific example is the MMD Volterra filter structure described by Frank [19], where two linear systems are combined multiplicatively, replacing the sum in Eq. (29) with a multiplication, obtaining a second-order (i.e., quadratic) Volterra model with certain desirable characteristics [18, p.70]. In fact, the complete structure proposed by Frank follows this "multiplicatively parallel" interconnection with a third linear system connected in series, providing an example of the class of block-oriented filter structures discussed in Section 6.

The construction of Frank can be generalized further, replacing the addition appearing in Eq. (29) with other binary operations. The qualitative behavior of the resulting system will depend on both that of the two subsystems being interconnected and the binary operator selected. The range of possible binary operations is uncountably infinite since any function $f : R^2 \to R$ can be used to define a binary operation, i.e.

$$x \oplus y = f(x, y). \tag{30}$$

For the design and analysis of composite nonlinear filters, however, it is useful to restrict consideration to binary operations like addition that exhibit desirable algebraic characteristics. Perhaps the best-known generalization of addition is the notion of a group; while the defining axioms for a group can be restricted to the case of real variables of primary interest in this discussion, they are usually given for an arbitrary (frequently finite) set Σ and it is useful to do this here to lay the foundation for the extensions to other data types discussed briefly in Section 10.2. Specifically, the set Σ and the binary operator $\oplus$ define a group $\mathcal{G} = (\Sigma, \oplus)$ if the following four conditions are satisfied [28, p. 43]:

1. *closure:* $x \oplus y \in \Sigma$ for all $x, y \in \Sigma$;
2. *associativity:* $x \oplus (y \oplus z) = (x \oplus y) \oplus z$ for all $x, y, z \in \Sigma$;
3. *unit element:* there exists $e \in \Sigma$ such that $x \oplus e = e \oplus x = x$ for all $x \in \Sigma$; further, this element is necessarily unique;
4. *invertibility:* for every $x \in \Sigma$, there exists a unique element $y \in \Sigma$ such that $x \oplus y = y \oplus x = e$.

Since the set of real numbers R with addition form the group $(R, +)$, the notion of a binary operator $\oplus$ that defines a group $(R, \oplus)$ may be viewed as a generalization of addition. Other, more restrictive generalizations of addition are obtained by restricting consideration to operators $\oplus$ that only satisfy a subset of these four axioms. Thus, if $(\Sigma, \oplus)$ satisfies only axioms 1, 2, and 3, the result is a *monoid*, while axioms 1 and 2 together define a *semigroup*, and axiom 1 alone defines a *groupoid*. As the following discussions illustrate, we can base parallel interconnections of nonlinear component filters on any of these algebraic constructs, but the characteristics of the interconnected system will depend on which of these classes of binary operators we select. Also, while we do not exploit this observation here, note that every monoid defines a category.

Taking the most general case first, note that any mapping $f : R^2 \to R$ defines a groupoid operation on R via Eq. (30). Using this groupoid operation generalizes the standard parallel combination defined in Eq. (29) to:

$$y_k = (\mathcal{F}_1\{x_k\}) \oplus (\mathcal{F}_2\{x_k\}) = f(\mathcal{F}_1\{x_k\}, \mathcal{F}_2\{x_k\}). \tag{31}$$

While this groupoid construction is the most general possible extension of the standard sum-based parallel interconnection strategy for linear systems, it does introduce some important complications when considering extensions to more than two parallel components. Specifically, unless the binary operation $\oplus$ also satisfies Condition 2 above (i.e., $(R, \oplus)$ constitutes a semigroup), it is necessary to specify the order in which these interconnected components are combined. For example, the interconnection of three parallel components can be performed in any of six possible orders and the results are generally not equivalent:

$$x \oplus (y \oplus z) \neq x \oplus (z \oplus y) \neq y \oplus (x \oplus z) \neq y \oplus (z \oplus x) \neq z \oplus (x \oplus y) \neq z \oplus (y \oplus x). \tag{32}$$

In fact, even if $\oplus$ does define a semigroup, it is generally not true that the operation is commutative: that is, $x \oplus y \neq y \oplus x$, in general. One of the advantages of addition is that it is commutative, so that $(R, +)$ defines an *abelian group*. It is the fact that addition is both associative and commutative that allows us to re-order the terms of a finite sum at will, writing it as:

$$x_1 + x_2 + \cdots + x_n = \sum_{i=1}^{n} x_i. \tag{33}$$

We can recover much of the simplicity of addition by restricting consideration to the following class of binary operations, which have been used to define a nonlinear generalization of the class of linear ARMA (autoregressive moving average) models for modeling dynamical systems [40].

The basis for this construction is the solution of the *associativity equation* [1, Ch. 1], a functional equation satisfied by any associative binary operator:

$$x \oplus (y \oplus z) = f(x, f(y, z)) = f(f(x, y), z) = (x \oplus y) \oplus z. \tag{34}$$

The operator $\oplus$ is said to be continuous if the function $f(\cdot, \cdot)$ in Eq. (30) is continuous, and it is *cancellative* if either $x_1 \oplus z = x_2 \oplus z$ or $z \oplus x_1 = z \oplus x_2$ for any z implies $x_1 = x_2$. (Note that if $\oplus$ defines a group, then both of these conditions are satisfied since these equations may be combined on either the right or the left by the inverse of the element z.) It has been shown that the operator $\oplus$ is continuous, associative and cancellative if and only if [1, Ch. 7, Thm. 1] it can be represented as:

$$x \oplus y = \phi^{-1}[\phi(x) + \phi(y)], \tag{35}$$

for some strictly monotone, continuous function $\phi : R \to R$. Note also that $e = \phi^{-1}(0)$ defines an identity element for $(R, \oplus)$ and that every real number x has an inverse given by $y = \phi^{-1}(-\phi(x))$. Further, it follows from the commutativity of addition that this operation is also commutative: $x \oplus y = y \oplus x$. Thus, as in the case of addition, $(R, \oplus)$ defines an abelian group and combinations of arbitrary numbers of terms may be re-ordered at will, allowing us to define the generalized summation operation:

$$\bigoplus_{i=1}^{n} x_i = x_1 \oplus x_2 \oplus \cdots \oplus x_n = \phi^{-1}\left[\sum_{i=1}^{n} \phi(x_i)\right]. \tag{36}$$

It is important to emphasize that, despite the defining restrictions imposed, this class of binary operators is very large, including both addition (for $\phi(x) = x$), multiplication (for $\phi(x) = \ln x$), along with the *parallel combination:*

$$x \| y = \frac{xy}{x + y}, \tag{37}$$

obtained by taking $\phi(x) = 1/x$ and motivated by the parallel combination of resistors in electrical networks, and the *projective addition operator* considered by Verriest [46] and obtained by taking $\phi(x) = x/(1 - x)$:

$$x \oplus y = \frac{2xy - x - y}{xy - 1}. \tag{38}$$

6 Block-oriented filter structures

While parallel combinations of two or more individual filters can lead to useful new composite structures, a much more flexible design strategy incorporates both parallel and cascade interconnections. This double combination approach leads to a class of nonlinear filters analogous to the class of *block-oriented nonlinear models* popular in characterizing biological system dynamics, consisting of series and parallel interconnections of linear dynamic models with static (i.e., memoryless) nonlinear elements [16]. Although any collection of nonlinear filter modules may be combined in this fashion, this approach is most useful when the interconnections preserve some useful nonlinear behavior, like positive homogeneity. This leads to a behavior-preserving "bottom up" design strategy that allows complex filter structures to be built from simpler, well-characterized modules. One specific example is the MMD Volterra filter structure of Frank [19], discussed in Section 5, which leads to useful Volterra filters of much simpler structure (i.e., fewer filter parameters to be specified) than the general case.

The key to making the block-oriented filter design strategy useful lies in coordinating the cascade and parallel interconnection strategies with the overall filter behavior we wish to preserve. As discussed in Section 4, cascade interconnections are inherently behavior-preserving—if filters $\mathcal{F}_1$ and $\mathcal{F}_2$ each leave some behaviorally-defined set of sequences invariant, so do the cascade interconnections $\mathcal{F}_1 \circ \mathcal{F}_2$

and $\mathcal{F}_2 \circ \mathcal{F}_1$—so the essential constraint on this block-oriented interconnection strategy is to choose a parallel combination operator $\oplus$ that is behavior-preserving. It is also desirable, as noted in Section 5, for $\oplus$ to be associative and commutative to simplify the interpretation of the resulting block diagrams.

As a specific example, suppose we wished to consider block-oriented combinations of positive homogeneous components that preserved this behavioral characteristic. Adopting the class of parallel combination operators $\oplus$ defined by Eq. (35) to guarantee associativity and commutativity, we are led to seek solutions $\phi(\cdot)$ of the following functional equation:

$$(\lambda x) \oplus (\lambda y) = \lambda(x \oplus y) \;\Rightarrow\; \phi^{-1}(\phi(\lambda x) + \phi(\lambda y)) = \lambda \phi^{-1}(\phi(x) + \phi(y)), \tag{39}$$

for all $\lambda > 0$ and all real x and y. Note that if $\phi(x)$ exhibits positive homogeneity of any order $\nu \neq 0$, it follows that

$$\phi(\lambda x) = \lambda^{\nu} \phi(x) \;\Rightarrow\; \lambda x = \phi^{-1}(\lambda^{\nu} \phi(x)), \tag{40}$$

for all real x and all $\lambda > 0$. Substituting this result into the expression for $(\lambda x) \oplus (\lambda y)$ gives

$$\begin{aligned} \phi^{-1}(\phi(\lambda x) + \phi(\lambda y)) &= \phi^{-1}(\lambda^{\nu}[\phi(x) + \phi(y)]) \\ &= \phi^{-1}(\lambda^{\nu} \phi(x \oplus y)) \\ &= \lambda(x \oplus y), \end{aligned} \tag{41}$$

implying that Eq. (39) is satisfied. It is not difficult to show that the only positive homogeneous functions $\phi : R \to R$ of order ν are of the general form [33, p. 115]:

$$\phi(x) = \begin{cases} a|x|^{\nu} & x \geq 0, \\ b|x|^{\nu} & x < 0, \end{cases} \tag{42}$$

for arbitrary real constants a and b. Note that this function is continuous and monotonic (thus invertible [26, p. 181]) if and only if $\nu \neq 0$ and a and b are both nonzero and have the same algebraic sign. Taking $a = b = 1$ and $\nu = 1$ reduces $\oplus$ to ordinary addition, but taking $\nu \neq 1$ leads to more general parallel combinations.

7 Clones: generalized parallel interconnections

The block-oriented nonlinear filter class just described represents one generalization of the parallel interconnection strategy discussed in Section 5. An alternative generalization is based on the concept of a *clone* as defined in universal algebra [44, p. 11]. Take A as an arbitrary set, let $r \geq 1$ and $q \geq 1$ be integers, let $\phi_i : A^q \to A$ be arbitrary mappings for $i = 1, 2, \ldots, r$, and let $G : A^r \to A$ be another arbitrary mapping. Here, A^q denotes the q-fold cartesian product of A with itself and the dimension of the domain of the functions ϕ_i and G is defined as their *arity.* A *clone superposition* is a composite mapping of the form:

$$\Phi(\mathbf{x}) = G(\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \ldots, \phi_r(\mathbf{x})), \tag{43}$$

defined for all $\mathbf{x} \in A^q$. A *clone of A* is a set of mappings that are closed under clone superposition and that contains all *projections* of the form

$$P_i(\mathbf{x}) = x_i, \tag{44}$$

where x_i is the i^{th} component of $\mathbf{x}$. An important example of a nonlinear digital filter that may be recognized as a clone construction is the FIR-Median Hybrid (FMH) filter class introduced by Heinonen and Neuvo [24]. In the simplest case, $r = 3$, the outer function of arity 3 is $G(x, y, z) = \text{median}\{x, y, z\}$, and the inner functions $\phi_i(\cdot)$ of arity $q = 2m + 1$ are given by:

$$\begin{aligned} \phi_1(\mathbf{w}_k) &= \frac{1}{m}\sum_{j=1}^{m} x_{k-j}, \\ \phi_2(\mathbf{w}_k) &= x_k, \\ \phi_3(\mathbf{w}_k) &= \frac{1}{m}\sum_{j=1}^{m} x_{k+j}. \end{aligned} \tag{45}$$

More generally, the outer function $G(\cdots)$ in an FMH filter is a median of arity r and the inner functions ϕ_i are arbitrary linear FIR filters. Another closely related filter class that contains the FMH filters is the class of FIR-WOS filters [48], a clone construction based on linear FIR filters as the inner functions but replacing the outer median with one of the weighted order statistic (WOS) filters described below.

As in the case of categories, it is useful to consider filter classes that constitute clones, but as in that case, most structurally-defined filter classes do not by themselves constitute clones because they are not closed under clone superposition. Exceptions include both the class of linear FIR filters and the class of nonlinear symmetric moving window filters [39]. Still, as the examples of the FMH and FIR-WOS filters just described illustrate—and the other constructions described below further illustrate—it is extremely useful to consider the clones *generated by* a collection of filters. That is, given a finite collection $\{\Phi_i(\cdots)\}$ of functions defining a set $\{\mathcal{F}_i\}$ of symmetric moving window nonlinear filters together with all projections, the set of all composite filter structures defined by clone superpositions of these components define the clone generated by the filters $\{\mathcal{F}_i\}$. Also as in the case of categories, it is possible to define clones based on behavioral characterizations. Again, the advantage of this observation is that if the filters $\{\mathcal{F}_i\}$ all exhibit some desirable behavior $\mathcal{B}$ and this behavior defines a clone, it follows that all filters in the clone generated by the set $\{\mathcal{F}_i\}$ exhibit behavior $\mathcal{B}$.

As a specific example of a behaviorally defined clone, note that if the functions $G : R^r \to R$ and $\phi_i : R^q \to R$ for $i = 1, 2, \ldots, r$ are all positive homogeneous, it follows that

$$\Phi(\lambda\mathbf{x}) = G(\phi_1(\lambda\mathbf{x}), \ldots, \phi_r(\lambda\mathbf{x})) = G(\lambda\phi_1(\mathbf{x}), \ldots, \lambda\phi_r(\mathbf{x})) = \lambda G(\phi_1(\mathbf{x}), \ldots, \phi_r(\mathbf{x})) = \lambda\Phi(\mathbf{x}). \tag{46}$$

Since projection operators are positive homogeneous, it follows that the family of positive homogeneous filters defines a clone. Analogous reasoning shows that the family of location-invariant filters also defines a clone, as does the family of *range-preserving filters* (i.e., filters $\mathcal{F}$ such that $a \leq x_k \leq b$ for all k and $\{y_k\} = \mathcal{F}\{x_k\}$ implies $a \leq y_k \leq b$ for all k) [39].

An important aspect of the clone construction is that it is very general, encompassing *all* of the interconnection strategies described above—i.e., cascade interconnection, parallel interconnection, and block-oriented structures—together with a number of others. The easiest of these special cases to demonstrate is parallel interconnection: letting

$$G(x_1, x_2, \ldots, x_r) = x_1 \oplus (x_2 \oplus (\ldots \oplus x_r) \ldots)), \tag{47}$$

it follows that the parallel combination of r filters $\{\mathcal{F}_i\}$ corresponds to their clone superposition. Thus, including functions of arbitrary arity based on the binary operation $\oplus$ with the finite filter set $\{\mathcal{F}_i\}$ provides a clone-based representation of arbitrary parallel combinations.

Similarly, suppose $\mathcal{F}$ and $\mathcal{G}$ are two symmetric moving window filters and consider their cascade interconnection, given by:

$$\begin{aligned} z_k = \mathcal{F}\{y_k\} &= \Psi(y_{k-m}, \ldots, y_k, \ldots, y_{k+m}), \\ y_k = \mathcal{G}\{x_k\} &= \Phi(x_{k-n}, \ldots, x_k, \ldots, x_{k+m}). \end{aligned} \tag{48}$$

Combining these equations gives the following equivalent expression for the cascade interconnection:

$$\begin{aligned} z_k = \mathcal{F} \circ \mathcal{G}\{x_k\} &= \Psi(\Phi(x_{k-m-n}, \ldots, x_{k-m}, \ldots, x_{k-m+n}), \ldots, \Phi(x_{k-n}, \ldots, x_k, \ldots, x_{k+n}), \\ & \quad \ldots, \Phi(x_{k+m-n}, \ldots, x_{k+m}, \ldots, x_{k+m+n})) \\ &= \Psi(\phi_{-m}(x_{k-n}, \ldots, x_k, \ldots, x_{k+n}), \ldots, \phi_0(x_{k-n}, \ldots, x_k, \ldots, x_{k+n}), \\ & \quad \ldots, \phi_m(x_{k-n}, \ldots, x_k, \ldots, x_{k+n})), \end{aligned} \tag{49}$$

where the functions $\phi_i : R^{2n+1} \rightarrow R$ represent clone superpositions of the function $\Phi : R^{2n+1} \rightarrow R$ with the projections P_i:

$$\phi_i(x_{k-n}, \ldots, x_k, \ldots, x_{k+n}) = \Phi(x_{k-n+i}, \ldots, x_{k+i}, \ldots, x_{k+n+i}), \tag{50}$$

for $i = -m, \ldots, 0, \ldots, m$. It follows from this observation that the clone generated by a family of filters $\{\mathcal{F}_i\}$ contains the category generated by this family.

Combining these last two results, it follows that the complete family of block-oriented structures defined by a family $\{\mathcal{F}_i\}$ of filters and a binary operator $\oplus$ is contained in the clone of filters generated by the underlying functions that define these components. In addition, every clone containing functions of arbitrary arity (e.g., functions like the median that are well-defined for arbitrary numbers of arguments) also contains the corresponding *weighted filters*, defined by:

$$\mathcal{F}_{\mathbf{w}}\{x_k\} = \Phi(w_{-m} \diamond x_{k-m}, \ldots, w_0 \diamond x_k, \ldots, w_m \diamond x_{k+m}), \tag{51}$$

where $w_j \diamond x_{k+j}$ denotes w_j-fold replication of the observation x_{k+j}. This result follows from the fact that $w_j \diamond x_{k+j}$ may be represented as the clone superposition of x_k with w_j copies of the projection operator P_j. Thus, if a clone contains the unweighted filter $\mathcal{F}$, it also contains all possible weighted versions of the filter, $\mathcal{F}_{\mathbf{w}}$ for all integer weight vectors $\mathbf{w}$. Probably the best known example of a weighted filter is the weighted median filter [15, 25]:

$$\mathcal{F}\{x_k\} = \text{median}\{w_{-m} \diamond x_{k-m}, \ldots, w_0 \diamond x_k, \ldots, w_m \diamond x_{k+m}\}, \tag{52}$$

which has been studied extensively: the survey by Yin *et al.* [49] lists 126 references describing its properties, its relationship with other filters, and a range of applications in both one and two dimensions. More generally, the family of *weighted order statistic (WOS) filters* is defined by Eq. (51) where the function $\Phi(\cdot)$ is taken as the j^{th}-largest element in the augmented data window defined there [47].

As an illustration of the power of the clone construction, consider the clone generated by the family of L-filters defined in Eq. (14), which will be termed the "L-filter clone" in the following discussion. Since this family contains the standard unweighted median filter, it follows that the L-filter clone contains the category **MED**, and thus all median filter cascades. Similarly, it follows from the results presented above that the L-filter clone contains all weighted median filters with integer weights. Further, since the only linear L-filter is the unweighted average appearing in the simple FMH filter defined in Eq. (45), it follows that this filter—constructed from the clone superposition of the three-place median with suitable projections and two copies of the unweighted linear average filter—also belongs to the L-filter clone. More generally, note that the L-filter clone contains all *weighted* linear average filters of the form:

$$\mathcal{L}_{\mathbf{w}}\{x_k\} = \sum_{j=-m}^{m} \alpha_j x_{k-j}, \tag{53}$$

where the filter weights $\{\alpha_j\}$ are given by:

$$\alpha_j = \frac{w_j}{\sum_{\ell=-m}^{m} w_\ell}. \tag{54}$$

Note that this means all filters with rational, nonnegative weights summing to 1 can be implemented using this construction. Thus, it follows that all FMH filters based on such linear FIR filters are included in the L-filter clone. Similarly, it follows that all FIR-WOS hybrid filters based on these linear components are included in the L-filter clone. In fact, it is not difficult to show that most of the nonlinear digital filters discussed by Astola and Kuosmanen [7] are members of the L-filter clone [39]. Further, it is easily shown that the L-filters defined in Eq. (14) are positive homogeneous, location-invariant, and range-preserving; since all three of these behaviors are preserved under clone superposition, it follows that all of the filters contained in the L-filter clone also exhibit these characteristics.

8 Idempotent nonlinear filters and root sequences

As noted in Section 2.2, a *root sequence* $\{r_k\}$ for a filter $\mathcal{F}$ is one that is invariant under the action of the filter:

$$\mathcal{F}\{r_k\} = \{r_k\}. \tag{55}$$

Closely related to the notion of a root sequence is that of *idempotence:* the filter $\mathcal{F}$ is idempotent if it satisfies the condition:

$$\mathcal{F}^2\{x_k\} = \mathcal{F}[\mathcal{F}\{x_k\}] = \mathcal{F}\{x_k\}, \tag{56}$$

for all input sequences $\{x_k\}$. Letting $\{y_k\}$ denote the filter's response to the arbitrary input sequence $\{x_k\}$, it follows from this result that $\{y_k\}$ is necessarily a root sequence: $\mathcal{F}\{y_k\} = \{y_k\}$. In words, this observation means that an idempotent filter converts any input sequence into a root sequence in one pass.

While the concept does not seem to be nearly as useful in the theory of linear filters, it is worth noting that there is nothing inherently nonlinear about the notion of root sequences. That is, invariant sequences exist for linear filters as well: the zero sequence is a trivial root sequence for any linear filter, but other root sequences generally also exist. In contrast, the class of idempotent filters is an essentially nonlinear one. In particular, suppose $\mathcal{L}$ is a linear idempotent filter and recall that, like the impulse response characterization noted earlier, linear filters are also fully characterized by their frequency responses $H(\omega)$. Further, one advantage of this characterization is that the frequency response of the cascade interconnection of two filters is simply the product of the individual frequency responses. Since repeated application of the filter $\mathcal{L}$ is equivalent to the cascade interconnection of $\mathcal{L}$ with itself, it follows that an idempotent linear filter must satisfy the condition [37]:

$$H^2(\omega) = H(\omega), \tag{57}$$

for all frequencies ω, implying $H(\omega) = 0$ or 1 for all ω. These conditions are satisfied by unrealizable ideal filters that fully pass all signals within one frequency band while fully rejecting those in all other frequency bands. In fact, the only realizable solutions of this equation are the trivial cases $H(\omega) = 0$ for all ω—the zero filter, which rejects all signals—and $H(\omega) = 1$ for all ω—the identity filter, which passes all signals unmodified. In contrast, nontrivial idempotent nonlinear digital do exist, including the recursive median filter mentioned earlier [31], morphological opening and closing filters [43], and the recursive Class 1 and nonrecursive Class 2 filters described by Haavisto *et al.* [23, 49].

It is interesting to note that the class of idempotent filters does not define a category since the composition of two idempotent filters need not be idempotent (see Pearson and Gabbouj for a counterexample [37]). Since every clone contains the corresponding category, it also follows that the class of idempotent filters does not constitute a clone. Conversely, it is not difficult to show that if $\mathcal{F}$ and $\mathcal{G}$ are idempotent filters that *commute*, then their composition is idempotent [37]:

$$(\mathcal{F} \circ \mathcal{G})^2 = \mathcal{F} \circ \mathcal{G} \circ \mathcal{F} \circ \mathcal{G} = \mathcal{F} \circ \mathcal{G} \circ \mathcal{G} \circ \mathcal{F} = \mathcal{F} \circ \mathcal{G} \circ \mathcal{F} = \mathcal{F} \circ \mathcal{F} \circ \mathcal{G} = \mathcal{F} \circ \mathcal{G}. \tag{58}$$

In fact, this result is closely related to root the sequence characterizations for the filters $\mathcal{F}$ and $\mathcal{G}$. Let $\mathcal{R}_{\mathcal{F}}$ and $\mathcal{R}_{\mathcal{G}}$ denote the root sequence sets for $\mathcal{F}$ and $\mathcal{G}$ and note that since idempotent filters map arbitrary sequences into root sequences in one pass, it follows that:

$$\mathcal{F} \circ \mathcal{G}\{x_k\} \in \mathcal{R}_{\mathcal{F}} \text{ and } \mathcal{G} \circ \mathcal{F}\{x_k\} \in \mathcal{R}_{\mathcal{G}}, \tag{59}$$

for all sequences $\{x_k\}$. Thus, if the idempotent filters $\mathcal{F}$ and $\mathcal{G}$ commute, it follows that the action of either of these compositions must map all sequences into the intersection $\mathcal{R}_{\mathcal{F}} \cap \mathcal{R}_{\mathcal{G}}$ of these root sets. Hence, a necessary condition for idempotent filters to commute is that their root sets intersect. Further, suppose $\mathcal{R}_{\mathcal{G}} \subset \mathcal{R}_{\mathcal{F}}$ and note that this implies that, for all $\{x_k\}$,

$$\mathcal{F} \circ \mathcal{G}\{x_k\} = \mathcal{G}\{x_k\} \;\Rightarrow\; \mathcal{F} \circ \mathcal{G} = \mathcal{G}, \tag{60}$$

so the composition is trivially idempotent. Conversely, let $\{z_k\} = \mathcal{G} \circ \mathcal{F}\{x_k\}$ and note that since $\mathcal{G}$ is idempotent, $\{z_k\} \in \mathcal{R}_{\mathcal{G}} \subset \mathcal{R}_{\mathcal{F}}$. Hence,

$$(\mathcal{G} \circ \mathcal{F})^2\{x_k\} = \mathcal{G} \circ \mathcal{F}\{z_k\} = \mathcal{G}\{z_k\} = \{z_k\} = \mathcal{G} \circ \mathcal{F}\{x_k\}, \tag{61}$$

for all real sequences $\{x_k\}$, implying that the cascade interconnection $\mathcal{G} \circ \mathcal{F}$ is idempotent. Overall, these results suggest that the cases where idempotence is not preserved under cascade interconnection are those where the root sets are “too dissimilar” [37].

Although the class of idempotent filters does not define a category, the notions of category theory are extremely useful in characterizing these filters. In particular, Pearson and Gabbouj [37] develop a number of category-theoretic based constructions for idempotent filters that are based on factorization results for idempotent matrices and other results from linear algebra.

9 Restricted forms of linearity

As a final illustration of the applicability of algebraic ideas to the characterization of nonlinear filters, the following discussion briefly considers the class of *restricted linear filters*, a behaviorally-defined class of nonlinear filters defined in joint work with Jaakko Astola [36]. The basic idea of this filter class is to require linearity to hold, not for all sequences, but only for some subset $\mathcal{S}$. First, note that a *cone* of sequences is any set $\mathcal{K}$ satisfying the following condition [51, p. 28]:

$$\{x_k\}, \{y_k\} \in \mathcal{K} \;\Rightarrow\; \{\alpha x_k + \beta y_k\} \in \mathcal{K}, \tag{62}$$

for all $\alpha, \beta \geq 0$. Given a cone $\mathcal{K}$, define the filter $\mathcal{F}$ to be $\mathcal{K}$*-linear* if the following two conditions hold:

a. the cone $\mathcal{K}$ is an invariant set for $\mathcal{F}$, i.e., $\{x_k\} \in \mathcal{K} \Rightarrow \mathcal{F}\{x_k\} \in \mathcal{K}$;

b. the filter $\mathcal{F}$ is *positive-linear on* $\mathcal{K}$, meaning that:

$$\mathcal{F}\{\alpha x_k + \beta y_k\} = \alpha \mathcal{F}\{x_k\} + \beta \mathcal{F}\{y_k\},$$

for all $\{x_k\}, \{y_k\} \in \mathcal{K}$ and all $\alpha, \beta > 0$.

It is important to note that while any linear filter $\mathcal{L}$ satisfies Condition (b) on any cone $\mathcal{K}$, Condition (a) is generally not satisfied without imposing some restrictions on $\mathcal{L}$.

The specific class of $\mathcal{K}$-linear filters considered here is based on the cone $\mathcal{I}$ of *increasing pseudocausal sequences*, defined as those sequences $\{x_k\}$ such that $x_k = 0$ for all $k < k^*$ for some finite $k^* \leq 0$ and $x_{k+1} \geq x_k$ for all $k \geq k^*$. The resulting class of *monotone-linear filters* consists of all shift-invariant filters $\mathcal{F}$ that are $\mathcal{K}$-linear with respect to this cone. An interesting behavioral characterization of this class is that, like fully linear shift-invariant filters, they are completely characterized on the cone $\mathcal{I}$ by their unit step response [36]. To emphasize the inherent nonlinearity of this filter class, note that it contains the entire class of L-filters defined in Eq. (14). This result follows from the fact that all sequences $\{x_k\}$ in the cone $\mathcal{I}$ are invariant under rank-ordering (i.e., $x_{(j)} = x_{k+j}$ for all $j = -m, \ldots, 0, \ldots, m$), as noted earlier. Thus, the L-filter acts linearly on $\mathcal{I}$, establishing defining Condition (b); Condition (a) follows from the fact noted earlier that linear filters preserve monotonicity if and only if the filter coefficients are nonnegative.

It is also worth noting that the class of $\mathcal{K}$-linear filters for any cone $\mathcal{K}$ defines a category. Specifically, the objects in this category are the sequences $\{x_k\} \in \mathcal{K}$, the identity morphism for all objects is the identity filter which satisfies the $\mathcal{K}$-linearity conditions trivially, and composition of morphisms corresponds to cascade interconnection, as in Section 4. To see that $\mathcal{K}$-linearity is preserved under cascade interconnection, note that for all $\{x_k\}, \{y_k\} \in \mathcal{K}$ and all $\alpha, \beta > 0$, we have:

$$\begin{aligned} \mathcal{F} \circ \mathcal{G}\{\alpha x_k + \beta y_k\} &= \mathcal{F}[\alpha \mathcal{G}\{x_k\} + \beta \mathcal{G}\{y_k\}] \\ &= \alpha \mathcal{F} \circ \mathcal{G}\{x_k\} + \beta \mathcal{F} \circ \mathcal{G}\{y_k\}, \end{aligned} \tag{63}$$

for any pair of $\mathcal{K}$-linear filters $\mathcal{F}$ and $\mathcal{G}$, establishing Condition (b) for $\mathcal{K}$-linearity. Note that Condition (a) follows from the preservation of set-invariance for cascade interconnection noted in Section 4.

This last observation leads to the following general construction procedure for $\mathcal{K}$-linear filters: form cascade interconnections of filters of the following types:

a. linear filters $\mathcal{L}$ that leave the cone $\mathcal{K}$ invariant;

b. nonlinear filters $\mathcal{F}$ whose root set $\mathcal{R}_{\mathcal{F}}$ contains $\mathcal{K}$.

Note that any filter $\mathcal{F}$ satisfying this second condition acts as an identity filter on the cone $\mathcal{K}$ and is thus trivially $\mathcal{K}$-linear. The cascade interconnections $\mathcal{L} \circ \mathcal{F}$ and $\mathcal{F} \circ \mathcal{L}$ both simply reduce to $\mathcal{L}$ on the cone $\mathcal{K}$, but these filters are, in general, neither equivalent nor linear for sequences $\{x_k\} \notin \mathcal{K}$. As a specific example, note that $\mathcal{I} \subset \mathcal{R}_{\mathcal{M}}$ for the standard median filter $\mathcal{M}$ and, as noted above, linear filters with nonnegative coefficients leave the cone $\mathcal{I}$ invariant. Thus, cascades formed from these two filters are monotone-linear but exhibit very nonlinear behavior for nonmonotonic sequences, and the two cascades $\mathcal{L} \circ \mathcal{M}$ and $\mathcal{M} \circ \mathcal{L}$ formed by these two filters manifest this nonlinearity differently [36].

Another "bottom-up" procedure for construct $\mathcal{K}$-linear filters on an arbitrary cone $\mathcal{K}$ is to form the following weighted parallel combination:

$$\mathcal{G}\{x_k\} = \sum_{i=1}^{n} \alpha_i \mathcal{F}_i\{x_k\}, \tag{64}$$

where $\alpha_i \geq 0$. Combining this construction with the observation that $\mathcal{K}$-linear filters are closed under cascade interconnection, it follows that all block-oriented models built from $\mathcal{K}$-linear components using Eq. (64) as the parallel combination rule are again $\mathcal{K}$-linear. Again considering the specific example of monotone-linearity, note that this construction leads to the following monotone-linear filter which does not belong to the category generated by the class of L-filters:

$$\mathcal{G}\{x_k\} = \frac{\mathcal{L} \circ \mathcal{M}\{x_k\} + \mathcal{M} \circ \mathcal{L}\{x_k\}}{2}, \tag{65}$$

where $\mathcal{L}$ is an arbitrary linear FIR filter with nonnegative coefficients and $\mathcal{M}$ is the standard median filter. This example demonstrates that the category of monotone-linear filters is larger than the category generated by the class of L-filters; other monotone-linear filters not derived from L-filters are described in the paper by Pearson, Astola, and Gabbouj [36].

10 Some important extensions

To keep the scope of this article manageable, the focus has been restricted to filtering real-valued sequences, as noted in the introduction. The ideas presented here can be usefully applied in a number of other settings, however, and the following subsections briefly address two particularly important ones: two-dimensional filters, required for applications like image processing, and non-real data types, which arise naturally in a wide range of applications (e.g., analysis of DNA sequence or protein sequence data, processing of streaming data records of mixed type, etc.).

10.1 Two-dimensional filters

While detailed descriptions of the resulting filters are generally more complex in the two-dimensional case, all of the key ideas presented here—homogeneity, category-theoretic descriptions of cascade interconnections, parallel interconnections based on groupoids, block-oriented constructions, and clones—extend directly to signal sequences of two (or more) dimensions. Indeed, note that all of the one-dimensional nonlinear digital filters described by Astola and Kuosmanen—many of which are used to illustrate specific constructions or concepts here—are presented along with their two-dimensional analogs [7]. Further, the greater complexity of filters in the two-dimensional case argues for simple "bottom-up" construction procedures like those described here even more strongly than in the one-dimensional case.

The two-dimensional analog of the symmetric moving window filter defined in Eq. (3) maps the two-dimenaional input sequence $\{(x_k, y_k)\}$ into the output sequence $\{(v_k, w_k)\}$ defined by:

$$\begin{aligned} u_k &= \Phi(x_{k-m}, \ldots, x_k, \ldots, x_{k+m}, y_{k-m}, \ldots, y_k, \ldots, y_{k+m}) = \Phi(\mathbf{r}_k), \\ v_k &= \Psi(x_{k-m}, \ldots, x_k, \ldots, x_{k+m}, y_{k-m}, \ldots, y_k, \ldots, y_{k+m}) = \Psi(\mathbf{r}_k), \end{aligned} \tag{66}$$

where $\mathbf{r}_k$ is the rectangular moving window defined by the vertical components x_{k-m} through x_{k+m} and the horizontal components y_{k-m} through y_{k+m}. Note that this filter exhibits homogeneity of order ν if the functions $\Phi : R^{4m+2} \to R$ and $\Psi : R^{4m+2} \to R$ are both homogeneous of order ν. It would be possible to consider mixed homogeneity conditions with, for example, the two functions exhibiting homogeneity of different orders, but it is not at all clear when this might be practically advantageous.

As in the univariate case, cascade interconnection of two-dimensional filters corresponds to functional composition, which is amenable to the same category-theoretic description presented in Section 4. The same may be said of parallel combinations via some groupoid, semigroup, or abelian group operator as discussed in Section 5, again leading to the prospect of useful block-oriented structures like those described in Section 6. An important point is that both cascade and parallel interconnection strategies must consistently process both components of the two-dimensional signal. The simplest way of doing this is componentwise, leading to cascade interconnections described by:

$$\begin{aligned} u_k &= \Phi_1(\Phi_2(\mathbf{r}_{k-n}), \ldots, \Phi_2(\mathbf{r}_k), \ldots, \Phi_2(\mathbf{r}_{k+n})), \\ v_k &= \Psi_1(\Psi_2(\mathbf{r}_{k-n}), \ldots, \Psi_2(\mathbf{r}_k), \ldots, \Psi_2(\mathbf{r}_{k+n})), \end{aligned} \tag{67}$$

and parallel interconnections described by:

$$u_k = \Phi_1(\mathbf{r}_k) \oplus \Phi_2(\mathbf{r}_k), \text{ and } v_k = \Psi_1(\mathbf{r}_k) \oplus \Psi_2(\mathbf{r}_k). \tag{68}$$

Of course, more complex alternatives are certainly possible in both cases, and these may be advantageous in certain circumstances (e.g., correcting for angular distortions in images). For example, we could consider "inherently two-dimensional cascades" of the form:

$$\begin{aligned} u_k &= \Gamma_1(\Phi_2(\mathbf{r}_{k-n}), \ldots, \Phi_2(\mathbf{r}_k), \ldots, \Phi_2(\mathbf{r}_{k+n}), \Psi_2(\mathbf{r}_{k-n}), \ldots, \Psi_2(\mathbf{r}_k), \ldots, \Psi_2(\mathbf{r}_{k+n})), \\ v_k &= \Gamma_2(\Phi_2(\mathbf{r}_{k-n}), \ldots, \Phi_2(\mathbf{r}_k), \ldots, \Phi_2(\mathbf{r}_{k+n}), \Psi_2(\mathbf{r}_{k-n}), \ldots, \Psi_2(\mathbf{r}_k), \ldots, \Psi_2(\mathbf{r}_{k+n})), \end{aligned} \tag{69}$$

where $\Gamma_1 : R^{4n+2} \to R$ and $\Gamma_2 : R^{4n+2} \to R$ are moving window filter functions. As in the case of more general notions of "mixed homogeneity" noted above, it is not clear when the more complex cascade interconnection strategy described in Eq. (69) would be advantageous. Similar comments apply to the clone interconnection strategy, generalizing the two-dimensional FIR-Median Hybrid filter. In particular, the key challenge in applying the algebraic ideas presented here to filtering problems in two or more dimensions lies in using them to help manage the inherently greater complexity of these filters than that seen in the one-dimensional case.

10.2 Other data types

Both the one-dimensional nonlinear filters discussed throughout most of this article and the two-dimensional filters discussed briefly in Section 10.1 are based on real variables. While most signal processing procedures to date have been concerned with real-valued sequences, there is growing interest in other data types, both in applications like DNA or protein sequence analysis [11] involving *nominal* (i.e., categorical) variables and in data mining applications based on the *streaming data model* [9], appropriate either when the dataset to be analyzed is too large to fit in main memory or in applications like telecommunications, financial services, e-comerce, or sensor network data analysis where the length of the data sequence is effectively infinite. An important aspect of these latter applications is that they frequently involve *mixed data types:* multivariable sequences with components of different types (e.g., binary, real-valued, integer-valued, categorical and ordinal). An important characteristic of applications involving variables that are not real-valued is that standard linear operations (i.e., addition and multiplication) are either not defined (as in the case of categorical variables) or give rise to meaningless results (e.g., non-integer averages of integer-valued data sequences). As a consequence, in cases where useful signal processing is feasible for these variables, it is necessarily nonlinear.

The most challenging cases are those involving categorical variables with no additional mathematical structure since there we can only count the occurrances of individual levels in the sequence or in a portion of the sequence, or make binary comparisons (i.e., observations x_j and x_k either assume the same value or assume different values). In favorable cases (e.g., spectrum estimation), it is possible to incorporate the results of these counting and comparison operations into an intermediate real variable that can be processed by known methods (e.g., correlation estimation in the case of Blackman-Tukey spectrum estimation) [38], but in unfavorable cases, our signal processing options may be extremely limited.

The situation is much better for *ordinal variables*, endowed with a total order [17] since there we can compute ranks so techniques like median filters, median filter cascades, or weighted median filters remain useful, even though notions like homogeneity or location-invariance are no longer meaningful. It is important to emphasize, however, that order can only be reasonably exploited if it occurs naturally in the original data and is not *induced* through some sort of coding artefact. It has been shown, for example, that spectral analysis of numerically coded DNA sequence data exhibits an undesirable dependence on the details of the coding scheme [11].

Intermediate between ordinal variables (corresponding to the order-theoretic notion of *chains* where any two values can be ordered) and nominal variables (corresponding to *antichaims* where no two values can be ordered) is the class of *partially ordered sets* in which some pairs of elements can be compared but not all pairs can be [17]. There, it is possible in some cases to exploit the order relation to define useful binary operations (e.g., the Dedekind groupoid discussed by Pearson and Gabbouj [38]), but the inability to unambiguously rank all data values significantly reduces our range of flexibility in developing signal processing algorithms (e.g., median filters are no longer useful).

Despite the complications inherent in dealing with non-real or mixed data types, the algebraic approach to these problems seems particularly promising since many of the basic ideas presented here can be applied to these data sequences. In particular, the notions of categories, groups, semigroups, groupoids and abelian groups can all be defined on arbitrary sets, as can clones: indeed, the field of universal algebra is fundamentally concerned with the properties of operations on sets [44].

11 Summary and conclusions

While there is no "nonlinear algebra" comparable in power and applicability to standard linear algebra, this article has attempted to demonstrate that algebraic ideas can be extremely useful in defining, characterizing, and understanding certain *specific* classes of nonlinear digital filters. As noted in Section 1, one of the great advantages of linearity is that it can be defined either structurally or behaviorally, and these two definitions are essentially equivalent. One of the primary difficulties in dealing with nonlinear filters is that "nonlinearity" is neither inherently structural nor inherently behavioral, but the examples presented here have demonstrated that both structural and behavioral classes of nonlinear filters can be defined and used to enhance our ability to understand and design useful filters.

Useful structural characteristics include cascade and parallel interconnections, which may be at least partially characterized in terms of category theory and notions closely related to group theory (i.e., groups, semigroups, groupoids, and abelian groups). In practical terms, note that cascade interconnections form the basis for filters like the data sieves discussed in Section 4, while parallel interconnection strategies arise in filters like the mean-median (MEM) filter described in Section 5. Combining both of these ideas leads to the notion of block-oriented nonlinear filter classes, analogous to the block-oriented nonlinear dynamic model classes used in modeling biological system dynamics [16] and which has been exploited in the design of practical Volterra filters like the MMD structure described in Section 6. A still more general extension of cascade and parallel interconnections is that based on the notion of a clone from universal algebra, which generalizes the construction on which FIR-Median Hybrid (FMH) and FIR-WOS filters are based. Indeed, it was noted in Section 7 that most of the nonlinear filters described in the book by Astola and Kuosmanen [7] belong to the clone generated by the class of L-filters.

One of the reasons these algebraic notions are useful is that in some cases the interconnection strategies just described can be shown to preserve useful nonlinear behavior. Specific examples considered here include homogeneity, positive homogeneity, location-invariance, boundedness, and monotonicity. The practical advantage of this observation is that simple nonlinear filters exhibiting these characteristics are often known and these "bottom-up" construction procedures (e.g., cascade and/or parallel interconnections, clones, etc.) can be applied to generate more complex—and thus more flexible—nonlinear filters that preserve these desirable characteristics.

We can also attempt to design nonlinear filters in a "top-down" fashion, specifying a desirable behavior and seeking filter structures that exhibit it. Often, this approach leads to functional equations like the Cauchy equations discussed in Section 3 which can, in favorable cases, be solved to provide explicit characterizations. As a specific example, a complete characterization of positive homogeneous functions is known [1, p. 44] and this has been used as a basis for designing positive homogeneous nonrecursive filters [35]; the extension to the recursive case is much more difficult [34]. An intermediate case is that of idempotent filters discussed in Section 8: as the results presented there demonstrated, some useful insights into the nature of idempotent nonlinear filters can be obtained using the algebraic ideas presented here, but the results fall short of general prescriptions for constructing idempotent filters (for a further discussion along these lines, refer to the paper [37]).

Finally, note that while the discussion of two-dimensional filters and non-real or mixed data types given in Section 10 was quite brief, it appears that the algebraic ideas presented here can be extremely useful in these applications. This is particularly the case in dealing with non-real data types, where signal processing is inherently nonlinear due to the frequent lack of well-defined addition and multiplication operations. Since the basic algebraic ideas of categories, groups, groupoids, and clones can all be applied in very general settings, they represent some of the only analytical tools available for dealing with these problems.

References

[1] J. Aczel, *A Short Course on Functional Equations*, Reidel, Dordrecht, 1987.

[2] J. Aczel and J. Dhombres, *Functional Equations in Several Variables*, Cambridge University Press, 1989.

[3] J. Aczel, D. Gronau, and J. Schwaiger, "Increasing Solutions of the Homogeneity Equation and of Similar Equations," *J. Math. Anal. Appl.*, v. 182, 1994, pp. 436–464.

[4] G.R. Arce and S.R. Hasan, "Elimination of interference terms of the discrete Wigner distribution using nonlinear filtering," *IEEE Trans. Signal Proc.*, v. 48, 2000, pp. 2321–2331.

[5] J. Astola, P. Haavisto, and Y. Neuvo, "Vector median filters," *Proc. IEEE*, v. 78, 1990, pp. 690–710.

[6] J. Astola, P. Heinonen and Y. Neuvo, "On root structures of median and median-type filters," *IEEE Trans. Acoustics Speech Signal Proc.*, v. 35, 1987, pp. 1199–1201.

[7] J. Astola and P. Kuosmanen, *Fundamentals of Nonlinear Digital Filtering*, CRC Press, 1997.

[8] T.C. Aysal and K.E. Barner, "Generalized Mean-Median Filtering for Robust Frequency-Selective Applications," *IEEE Trans. Signal Proc.*, v. 55, 2007, pp. 937–948.

[9] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and issues in data stream systems," in *Proc. 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS'02)*, Madison, WI, USA, June, 2002, pp. 1–16.

[10] A. Bangham, "Properties of a series of nested median filters, namely the data sieve," *IEEE Trans. Signal Proc.*, v. 41, 1993, pp. 31–42.

[11] K.M. Bloch and G.R. Arce, "Analyzing protein sequences using signal analysis techniques," in *Computational and Statistical Approaches to Genomics*, W. Zhang and I. Schmulevich, eds., Kluwer, 2002, pp. 113–124.

[12] T. Blyth, *Categories*, Longman, 1986.

[13] A.C. Bovik, T.S. Huang, and D.C. Munson, Jr., "A generalization of median filtering using linear combinations of order statistics," *IEEE Trans. Acoustics Speech Signal Proc.*, v. 31, 1983, pp. 1342–1350.

[14] J. Brandt, "Invariant Signals for Median-Filters," *Utilitas Math.*, v. 31, 1987, pp. 93–105.

[15] D.R.K. Brownrigg, "The weighted median filter," *Communications ACM*, v. 27, 1984, pp. 807–818.

[16] H.W. Chen, "Modeling and identification of parallel nonlinear systems: Structural classification and parameter estimation methods," *Proc. IEEE*, v. 83, 1995, pp. 523–537.

[17] B.A. Davey and H.A. Priestley, *Introduction to Lattices and Order*, Cambridge University Press, 1990.

[18] F.J. Doyle III, R.K. Pearson, and B.A. Ogunnaike, *Identification and Control Using Volterra Models*, Springer, 2002.

[19] W.A. Frank, "An efficient approximation to the quadratic Volterra filter and its application in real-time loudspeaker linearization," *Signal Proc.*, v. 45, 1995, pp. 97–113.

[20] M. Gabbouj and J. Astola, "Nonlinear Order Statistic Filter Design: Methodologies and Challenges," *Proc. European Signal Processing Conf. (EUSIPCO 2000)*, Tampere, Finland, September, 2000, pp. 377–384.

[21] N.C. Gallagher and G.L. Wise, "A Theoretical Analysis of the Properties of Median Filters," *IEEE Trans. Acoustics Speech Signal Proc.*, v. 46, 1981, pp. 1136–1141.

[22] P.P. Gandhi and S.A. Kassam, "Design and performance of combination filters for signal restoration," *IEEE Trans. Signal Processing*, v. 39, 1991, pp. 1524–1540.

[23] P. Haavisto, M. Gabbouj, and Y. Neuvo, "Median based idempotent filters," *J. Circuits Systems Computers*, v. 1, 1991, pp. 125–148.

[24] P. Heinonen and Y. Neuvo, "FIR-Median Hybrid filters," *IEEE Trans. Acoustics Speech Signal Proc.*, v. 35, 1987, pp. 832–838.

[25] B.I. Justusson, "Median filtering: statistical properties," in *Two-Dimensional Digital Signal Processing II*, T.S. Huang, ed., Springer-Verlag, 1981.

[26] G. Klambauer, *Mathematical Analysis*, Marcel Dekker, 1975.

[27] Y.H. Lee and S.A. Kassam, "Generalized median filtering and related nonlinear filtering techniques," *IEEE Trans. Acoustics Speech Signal Proc.*, v. 33, 1985, pp. 672–683.

[28] S. MacLane and G. Birkhoff, *Algebra*, Macmillan, 2nd ed., 1979.

[29] V.J. Mathews and G.L. Sicuranza, *Polynomial Signal Processing*, Wiley, 2000.

[30] P.H. Menold, R.K. Pearson, F. Allgöwer, "Online outlier detection removal," *Proc. 7th Mediterranean Conf. Control Automation (MED99)*, Haifa, Israel, June, 1999, pp. 1110–1133.

[31] T.A. Nodes and N.C. Gallagher, "Median filters: Some modifications and their properties," *IEEE Trans. Acoustics Speech Signal Proc.*, v. 30, 1982, pp. 739–746.

[32] A.V. Oppenheim and R.W. Schafer, *Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1975.

[33] R.K. Pearson, *Discrete-Time Dynamic Models*, Oxford University Press, 1999.

[34] R.K. Pearson, "A recursive variant of Mallows' filter class," *IEEE Trans. Signal Proc.*, v. 49, 2001, pp. 2745–2752.

[35] R.K. Pearson, "Scale-invariant nonlinear digital filters," *IEEE Trans. Signal Proc.*, v. 50, 2002, pp. 1986–1993.

[36] R.K. Pearson, J. Astola, and M. Gabbouj, "Nonlinear Filters Satisfying Restricted Linearity Conditions," *Proc. Nonlinear Signal Image Processing Conf. (NSIP'07)*, 2007.

[37] R.K. Pearson and M. Gabbouj, "Idempotent nonlinear filters," *Proc. Nonlinear Signal Image Processing Conf. (NSIP'03)*, Grado-Trieste, Italy, June, 2003.

[38] R.K. Pearson and M. Gabbouj, "Finite-set signal processing," Ch. 3 in *Advances in Nonlinear Signal and Image Processing*, S. Marshall and G. Sicuranza, eds., Hindawi, 2006, pp. 49–78.

[39] R.K. Pearson, M. Gabbouj, and J. Astola, "Nonlinear FIR Filter Clones," *Proc. Nonlinear Signal Image Processing Conf. (NSIP'03)*, Grado-Trieste, Italy, June, 2003.

[40] R.K. Pearson, Ü. Kotta and S. Nõmm, "Systems with Associative Dynamics," *Kybernetika*, v. 38, 2002, pp. 585–600.

[41] I. Pitas and A.N. Venetsanopoulos, *Nonlinear Digital Filters*, Kluwer, 1990.

[42] P.J. Rousseeuw and A.M. Leroy, *Robust Regression and Outlier Detection*, Wiley, 1987.

[43] J. Serra, *Image Analysis and Mathematical Morphology*, v. 2, Academic Press, 1988.

[44] A. Szendrei, *Clones in Universal Algebra*, Les Presses de l'Universite de Montreal, 1986.

[45] P.F. Velleman, "Robust nonlinear data smoothers: Definitions and recommendations," *Proc. National Academy of Sciences*, v. 74, 1977, pp. 434–436.

[46] E.I. Verriest, "Linear systems over projective fields," *Proc. 5th IFAC Symposium on Nonlinear Control Systems (NOLCOS'01)*, St. Petersburg, Russia, 2001, pp. 207–212.

[47] P.D. Wendt, E.J. Coyle, and N.C. Gallagher, Jr., "Stack filter," *IEEE Trans. Acoustics, Speech, Signal Proc.*, v. 34, 1986, pp. 898–911.

[48] L. Yin and Y. Neuvo, "Fast adaptation and performance characteristics of FIR-WOS hybrid filters," *IEEE Trans. Signal Processing*, v. 42, 1994, pp. 1610–1628.

[49] L. Yin, R. Yang, M. Gabbouj, and Y. Neuvo, "Weighted Median Filters: A Tutorial," *IEEE Trans. Circuits and Systems–II: Analog and Digital Signal Processing,* v. 43, 1996, pp. 157–192.

[50] O. Yli-Harja, J. Astola, and Y. Neuvo, "Analysis of the properties of median and weighted median filters using threshold logic and stack filter representation," *IEEE Trans. Signal Processing*, v. 39, 1991, pp. 395–410.

[51] G.M. Ziegler, *Lectures on Polytopes*, Springer-Verlag, 1995.