Efficient Implementation of One-Dimensional Recursive Median Filters

SUNG-JEA KO, YONG HOON LEE, AND ADLY T. FAM

Abstract —**It is shown that in one-dimensional (1-D) recursive median (RM) filtering, the present output is fully determined by the input data in the window and by the most recent output. All other past outputs are shown to be redundant. Based on this result, efficient algorithms and VLSI implementation for 1-D RM filters are presented, and shown to compete favorably with those for standard median (SM) filtering.**

I. Introduction

The standard median (SM) filter is a simple nonlinear smoother that can suppress noise while retaining sharp sustained changes (edges) in signal values. It is particularly effective in reducing impulsive-type noise [1]. The output of the SM filter at a point is the median value of the input data inside the window centered at the point. If we let $\{X(k)|1 \leqslant k \leqslant L\}$ and $\{Y_s(k)|1 \leqslant k \leqslant L\}$, respectively, be the input and the output of the 1-D SM filter of window size $2N+1$, then

$$Y_s(k) = \text{med}\{X(k-N),\cdots,X(k),\cdots,X(k+N)\}. \quad (1)$$

Here to account for startup and end effects, $X(1)$ and $X(L)$, respectively, are repeated N times at the beginning and at the end of the input. The RM filter is a modification of the SM filter defined in (1). Specifically, the output $Y_r(k)$ of the RM filter of size $2N+1$ is given by

$$Y_r(k) = \text{med}\{Y_r(k-N),\cdots,Y_r(k-1),\ X(k),\cdots,X(k+N)\}. \quad (2)$$

At the beginning of the filtering, it is assumed that $Y_r(1-N) = \cdots = Y_r(0) = X(1)$; the end effect is considered as in SM filtering. RM filtering can extract signal roots better than SM filtering, and is a useful alternative to SM filtering in some applications [2]–[4]. While many efficient algorithms for SM filters have been proposed (e.g., [5]–[10]), no fast algorithm has been introduced specifically for RM filters. In general, RM filters are implemented by modifying an SM filtering algorithm [17] and, as a consequence, the implementation of RM filters is computationally and structurally more complex than that of SM filters.

In this paper, we first point out that in RM filtering the previous outputs $\{Y_r(k-N),\cdots,Y_r(k-2)\}$ are not necessary to determine the present output $Y_r(k)$, and are therefore redundant. Based on this observation, efficient algorithms and VLSI implementation for 1-D RM filters are proposed, and shown to compete favorably with those for SM filters.

II. Some Properties of 1-D RM Filters and their Application to Implementation

The property stated below shows that the previous outputs $\{Y_r(k-N),\cdots,Y_r(k-2)\}$ in (2) are redundant.

Manuscript received January 24, 1990. This work was supported by the National Science Foundation under Grant DCI-8611859, and by the United States Air Force under Contract F30602-88-D-0025, Task C-8-2400. This paper was recommended by Associate Editor P. H. Bauer.

S.-J. Ko is with the Department of Electrical and Computer Engineering, University of Michigan, Dearborn, MI 48128.
Y. H. Lee is with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, Cheongryang, Seoul, Korea.
A. T. Fam is with the Department of Electrical and Computer Engineering, State University of New York, Buffalo, NY 14260.

IEEE Log Number 9038555.

TABLE I
The Maximum Number of Operations which may be Required for Evaluating an Output of SM and RM Filters of Size $2N+1$ when Fast Algorithms A in [8] and B in [10] are Used. Here $\lfloor x \rfloor$ is the Integer Part of x.

Maximum Number of Operations		
Filter Type	Type of Fast Algorithm	
	A	B
SM	4N+2	$2\lfloor \log(2N+1) \rfloor + 2$
RM (without Property 2)	8N+2	$4\lfloor \log(2N+1) \rfloor + 4$
RM (with Property 2)	2N+4	$2\lfloor \log(N+1) \rfloor + 4$

Property 1: In RM filtering, the output $Y_r(k)$ is represented by

$$Y_r(k) = \text{med}\Big\{\overbrace{Y_r(k-1),\cdots,Y_r(k-1)}^{N \text{ times}},\ X(k),\cdots,X(k+N)\Big\}. \quad (3)$$

Proof: The output $\{Y_r(k)\}$ is a locally monotonic sequence of length $N+1$, that is, $\{Y_r(k-N),\cdots,Y_r(k)\}$ is either nondecreasing or nonincreasing for any k [2]. Consider the nondecreasing case, $Y_r(k-N) \leqslant \cdots \leqslant Y_r(k-1)$. Since the output is locally monotonic of length $N+1$, $Y_r(k-1) \leqslant Y_r(k)$. This implies that if the number of samples in $\{X(k),\cdots,X(k+N)\}$ that are greater than or equal to $Y_r(k-1)$ is n_k, then $n_k \geqslant 1$ and clearly $Y_r(k) = Y_r(k-1)$ if $1 \leqslant n_k \leqslant N$, and $Y_r(k) = \min\{X(k),\cdots,X(k+N)\}$ if $n_k = N+1$ because, otherwise, the local monotonicity is violated. Thus $Y_r(k)$ is represented as

$$Y_r(k) = \begin{cases} \min\{X(k),\cdots,X(k+N)\}, & \text{if } X(k+i) \geqslant Y_r(k-1) \text{ for all } i,\ 0 \leqslant i \leqslant N \\ Y_r(k-1), & \text{otherwise} \end{cases}$$
$$= \text{med}\Big\{\underbrace{Y_r(k-1),\cdots,Y_r(k-1)}_{N \text{ times}},\ X(k),\cdots,X(k+N)\Big\}.$$

Similarly, we can prove the nonincreasing case. □

It is interesting to note that the RM filter can be thought of as a recursive weighted median filter [11] that gives more weight only to the last output $Y_r(k-1)$. Obviously, the implementation of RM filtering will become easier by using this property. The following property is a direct sequel to Property 1 and is particularly useful in implementing 1-D RM filters on a general purpose computer.

Property 2: The output $Y_r(k)$ of the RM filter is given by

$$Y_r(k) = \text{med}\{X^k_{\min}, X^k_{\max}, Y_r(k-1)\} \quad (4)$$

where $X^k_{\min}$ and $X^k_{\max}$, respectively, are the minimum and the maximum of $\{X(k),\cdots,X(k+N)\}$.

By incorporating this property with fast 1-D SM filtering algorithms in [8] and [10], fast algorithms for 1-D RM filters may

0098-4094/90/1100-1447$01.00 ©1990 IEEE

be obtained. Table I compares the maximum number of operations that may be required for evaluating outputs of RM and SM filters of size $2N+1$ when the algorithms in [8] and [10] are used (for RM filters, filtering with and without using Property 2 is considered). As expected, in RM filtering, computational load is reduced significantly by incorporating Property 2 with the algorithms. It is seen that due to Property 2, the implementation of RM filtering is even simpler than (or at least comparable to) that of SM filtering.

The property stated below establishes an interesting relationship between RM and *last output reference* (LOR) filters [12], and leads to an efficient hardware structure of 1-D RM filters. Here the LOR filter of size $N+1$ selects a sample that is the closest in value to the most recent output $Y(k-1)$ from $\{X(k),\cdots,X(k+N)\}$ at each time k.

Property 3: If the input is a binary sequence, then

$$Y_r(k)=\begin{cases}X(k), & \text{if } X(k)=\cdots=X(k+N)\\ Y_r(k-1), & \text{otherwise.}\end{cases} \tag{5}$$

Again this is an obvious consequence of Property 1. For binary inputs, the output $Y_r(k)$ of the RM filter of size $2N+1$ can be thought of as one of $\{X(k),\cdots,X(k+N)\}$, which is the closest in value to the last output $Y_r(k-1)$. This observation indicates that if the input is binary, then the RM filter of size $2N+1$ is identical to the LOR filter of size $N+1$. The equivalence between the two filters is observed in [12], but is proved in a different manner. It should be pointed out that the equivalence does not hold unless the input is binary; for a multilevel input sequence, the RM filter having the threshold decomposition property [13] and the LOR filter, which does not have such a property, usually produce different results.

The rule in Property 3, which determines the output of the RM filter, can be represented as a Boolean expression. For example, under the assumption that the input to this filter is restricted to binary values,

$$Y_r(k)=Y_r(k-1)[X(k)+X(k+1)+\cdots+X(k+N)] + [X(k)X(k+1)\cdots X(k+N)] \tag{6}$$

where we represent the OR operation by addition, and the AND operation by multiplication. Fig. 1 illustrates the logic network realizing the Boolean function in (6). Note that this Boolean function is realized by using two OR gates and two AND gates irrespective of the window size $2N+1$. In Fig. 2, we present the logic network that can produce the outputs of any binary RM filter whose window size is less than or equal to $2K+1$ where K is a positive integer.

The SM filtering operation can also be represented as a Boolean expression [14]. It is straightforward to see that the realization of a Boolean function of SM filtering with size $2N+1$ requires $(2N+1)!/[(N+1)!N!]$ AND/OR gates. Therefore, by making use of Property 3, the implementation of RM filtering for binary signals becomes remarkably more simple than that of SM filtering[1].

[1]The SM filter for binary signals may be implemented by using a counter followed by a comparator [14], which is also more complicated than the proposed RM filtering algorithm. In [15], in order to avoid the difficulty in implementing SM filters with the digital logic, an analog circuit structure was developed. The analog circuit for SM filtering, however, may require development of a customized chip, which is usually expensive, while the proposed digital structure admits inexpensive implementation using logic array chips.

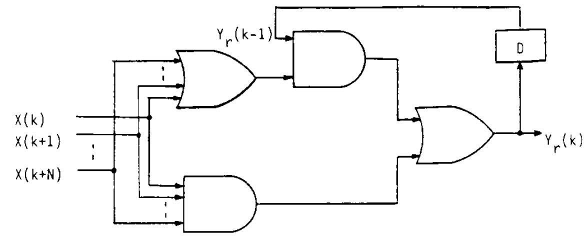

Fig. 1. The logic network for binary RM filtering (window size $2N+1$).

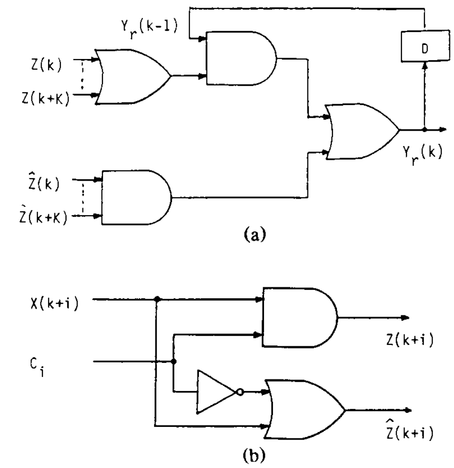

Fig. 2. (a) The logic network producing the outputs of any binary RM filter whose window size is less than or equal to $2K+1$ where $0\leqslant i\leqslant K$, $2N+1$ is the size of the RM filter, $K\geqslant N$, and $Z(k+i)$ and $\hat{Z}(k+i)$ are obtained from $X(k+i)$ through the network in (b). (b) The network producing $Z(k+i)$ and $\hat{Z}(k+i)$ from $X(k+i)$ where $C_i=1$ if $0\leqslant i\leqslant N$, and $C_i=0$ if $N+1\leqslant i\leqslant K$.

The RM filter for multilevel input signals can be implemented using the binary RM filters. In the following section, we present efficient VLSI structures for multilevel RM filters.

III. VLSI Realization

The output of an SM filter with a multilevel input sequence can be obtained from the outputs of binary SM filters. This has been done through either the threshold decomposition [14], [15] or the bit-serial approach [16]. RM filters have the threshold decomposition property [13], based on which they can be implemented, as shown in Fig. 3. Here the input is assumed to be an M-level signal. Obviously, the binary RM filter structure in Fig. 2 can be applied to this realization. Next we shall show that the binary RM filter in Fig. 2 can also be incorporated with the bit-serial approach.

Suppose that the input level $M=2^P$ (P-bit signal). To simplify notation, we denote by $\{A_0,A_1,\cdots,A_{N+1}\}$ the set of data in the window $\{Y_r(k-1),X(k),\cdots,X(k+N)\}$ where $A_0=Y_r(k-1)$ and $A_i=X(k+i-1)$, $1\leqslant i\leqslant N+1$. Let the code words (radix-2 binary representation) of A_i and $Y_r(k)$, respectively, be $(a_i^1a_i^2\cdots a_i^P)$ and $(b^1b^2\cdots b^P)$ where a_i^1 and b^1 are the most significant bits. In the bit-serial realization, the output at each bit is obtained sequentially, starting with the most significant bit. At each bit-level, with the exception of the most significant bit, the binary input values at the level are modified before filtering depending on the outputs of more significant

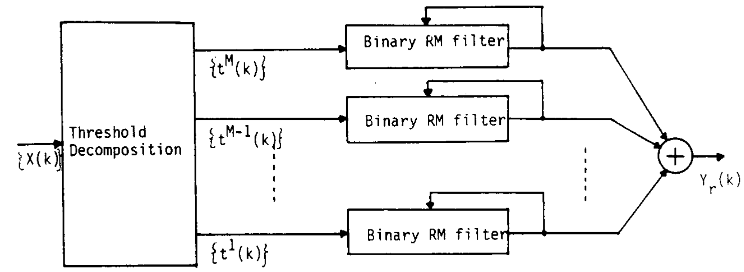

Fig. 3. The implementation of an RM filter based on the threshold decomposition where

$$t^j(k) = \begin{cases} 1, & \text{if } X(k) \geq j \\ 0, & \text{if } X(k) < j. \end{cases}$$

TABLE II
AN EXAMPLE ILLUSTRATING PROPERTY 4. HERE $N = 2$, $P = 4$, AND $\{A_0, A_1, A_2, A_3\} = \{11, 6, 5, 12\}$.

j	a_0^j	a_1^j	a_2^j	a_3^j	$\hat{a}_0^j$	$\hat{a}_1^j$	$\hat{a}_2^j$	$\hat{a}_3^j$	$b^j = \hat{a}_0^j[\hat{a}_1^j + \hat{a}_2^j + \hat{a}_3^j] + [\hat{a}_1^j \hat{a}_2^j \hat{a}_3^j]$
1	1	0	0	1	1	0	0	1	1
2	0	1	1	1	0	0	0	1	0
3	1	1	0	0	1	0	0	1	1
4	1	0	1	0	1	0	0	1	1

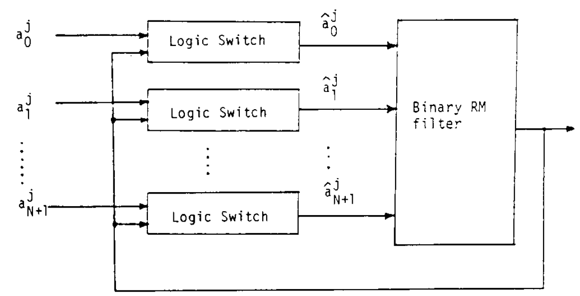

Fig. 4. Bit-serial realization of the RM filter of size $2N+1$.

bits. To implement the RM filter using the bit-serial approach and the binary RM filter in Fig. 2, the input data at each bit should be modified as in the following property.

Property 4: The output of the binary RM filter at the jth bit is given by

$$b^j = \hat{a}_0^j\left[\hat{a}_1^j + \hat{a}_2^j + \cdots + \hat{a}_{N+1}^j\right] + \left[\hat{a}_1^j \hat{a}_2^j \cdots \hat{a}_{N+1}^j\right] \tag{7}$$

where $\hat{a}_i^1 = a_i^1$, $0 \leq i \leq N+1$, and for each j, $2 \leq j \leq P$,

$$\hat{a}_i^j = \begin{cases} a_i^j, \text{ if } a_i^m = b^m, & \text{for all } m, 1 \leq m \leq j-1 \\ a_i^r, \text{ if } a_i^m = b^m, & \text{for all } m, 1 \leq m \leq r-1, \\ & \text{and for some } r, 1 \leq r \leq j-1, a_i^r \neq b^r. \end{cases} \tag{8}$$

This property can be proved through slight modification of the proof of theorem 3 in [16], and thus the proof is omitted. Equation (8) can be rewritten as follows:

$$\hat{a}_i^j = \hat{a}_i^{j-1} Q_i^j + a_i^j \overline{Q}_i^j \tag{9}$$

where $Q_i^j = [\hat{a}_i^{j-1} \oplus b^{j-1}] + Q_i^{j-1}$, and $Q_i^1 = 0$. Here we represent the exclusive-OR operation by $\oplus$, and the complement operation by $(\bar{\cdot})$. This Boolean expression would be useful for implementation. It is interesting to note that the algorithm in this observation replaces A_i with a certain value that is greater (smaller) than A_i, once it becomes evident that A_i is greater (smaller) than $Y_r(k)$. As an example, consider an RM filter of size 5 ($N = 2$). Suppose that the data inside the window are given by $\{A_0, A_1, A_2, A_3\} = \{11, 6, 5, 12\}$ ($P = 4$). The process of filtering using Property 4 is summarized in Table II. Note that the correct output value $11 = (1\ 1\ 0\ 1)$ is obtained, and $A_1 = 6$ and $A_2 = 5$, which are smaller than the output $= 11$, are replaced with zero while $A_3 = 12$, which is greater than the output, is replaced with $15 = (1\ 1\ 1\ 1)$.

Fig. 4 illustrates the bit-serial realization of an RM filter that uses the binary RM filter in Fig. 2. A realization of the logic switch producing the modified binary input data may be found in [16]. The logic switch may also be implemented using (9).

The implementation based on the threshold decomposition property is parallel and modular, but its hardware complexity grows exponentially with the number of bits in the inputs. On the other hand, the hardware complexity of the implementation based on the bit-serial approach grows linearly with the number of bits. However, its speed may be slower because the output at each bit is obtained sequentially. Parallel use of binary RM filters would speed up the computation of the output.

IV. CONCLUSION

All previous outputs except the most recent one are shown to be redundant in determining the current output. This and related properties result in efficient RM filtering algorithms.

The approach introduced in this work could lead to a simple framework in investigating statistical properties of RM filters [3], [4]. It is also of interest to develop similar properties for recursive weighted median filters.

REFERENCES

[1] T. S. Huang, Ed., *Two-dimensional Digital Signal Processing II: Transforms and Median Filters*. New York: Springer-Verlag, 1981.

[2] T. A. Nodes and N. C. Gallagher, "Median filters: Some modifications and their properties," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 739–746, Dec. 1982.

[3] M. P. McLoughlin and G. R. Arce, "Deterministic properties of the recursive separable median filter," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 98–106, Jan. 1987.

[4] G. R. Arce and N. C. Gallagher, "Stochastic analysis for the recursive median filter process," *IEEE Trans. Inform. Theory*, vol. IT-34, pp. 669–679, Jul. 1988.
[5] T. S. Huang, G. T. Yang, and G. Y. Tang, "A fast two-dimensional median filtering algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, pp. 13–18, Feb. 1979.
[6] E. Ataman, V. K. Aatre, and K. M. Wong, "A fast method for real-time median filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 415–420, Aug. 1980.
[7] K. Oflazer, "Design and implementation of a single chip 1-D median filter," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 1164–1168, Oct. 1983.
[8] J. B. Bednar and T. L. Watt, "Alpha-trimmed means and their relationship to median filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 145–153, Feb. 1984.
[9] V. V. B. Rao and K. S. Rao, "A new algorithm for real-time median filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 1674–1675, Dec. 1986.
[10] I. Pitas, "Fast algorithms for running ordering and max/min calculation," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 795–804, June 1989.
[11] O. Yli-Harja, J. Astola, and Y. Neuvo, "Analysis of the properties of weighted median filters using threshold logic and stack filter representation," Research Rep. 9/1988, Dep. of Information Technology, Lappeenranta University of Technology, Lappeenranta, Finland.
[12] Y. H. Lee, S.-J. Ko, and A. T. Fam, "Efficient impulsive noise suppression via nonlinear recursive filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 303–306, Feb. 1989.
[13] J. P. Fitch, E. J. Coyle, and N. C. Gallagher, Jr., "Median filtering by threshold decomposition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 1183–1188, Dec. 1984.
[14] P. D. Wendt, E. J. Coyle, and N. C. Gallagher, Jr., "Stack filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 898–911, Aug. 1986.
[15] R. G. Harber, S. C. Bass, and G. L. Neudeck, "VLSI implementation of a fast rank-order algorithm," in *Proc. ICASSP '85*, Tampa, FL, Mar. 1985.
[16] K. Chen, "Bit-serial realizations of a class of nonlinear filters based on positive Boolean functions," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 785–794, June 1989.
[17] D. S. Richards, "VLSI median filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 145–153, Jan. 1990.

Review and Discussion of Stability Criteria for Linear 2-Ports

RENE P. MEYS

Abstract —**Stability criteria for linear 2-ports have been discussed by many authors, leading to numerous formulations that are sometimes redundant or even not consistent. In this paper, a simple geometrical interpretation of the stability conditions is given from which it follows that absolute stability is insured by Rollett's fundamental condition $K > 1$ in conjunction with one well chosen complementary condition. Which complementary conditions are valid is examined and some generalization is introduced.**

I. Introduction

A linear 2-port is said to be absolutely stable if there is no passive source and load combination that causes the circuit to oscillate. A necessary condition for stability is that the real part of the immittance looking into any port remains positive with an arbitrary passive immittance connected to the other port. This condition may sometimes not be sufficient [1]. However, most authors concentrated on it, because it is more important with practical 2-ports. Bearing this remark in mind, let us briefly recall the most current stability criteria.

Manuscript received December 30, 1989. This paper was recommended by Associate Editor T. T. Vu.
The author is with the Service d'Electronique et Télécommunications, CP 165, Université Libre de Bruxelles, B-1050 Brussels, Belgium.
IEEE Log Number 9038556.

Describing the linear 2-port by its H, Y, or Z parameters, Rollett [2] proposes the following three conditions:

$$K = \frac{2G_{11}G_{22} - \mathrm{Re}(Y_{12}Y_{21})}{|Y_{12}Y_{21}|} > 1 \tag{1}$$

$$G_{11} = \mathrm{Re}(Y_{11}) > 0 \tag{2}$$

$$G_{22} = \mathrm{Re}(Y_{22}) > 0 \tag{3}$$

where Y_{ij} may be replaced by H_{ij} or Z_{ij}.

Using the transformation formulas for S-parameters under a reference impedance change, Bodway [3] states that necessary and sufficient conditions for absolute stability are

$$K = \frac{1 - |S_{11}|^2 - |S_{22}|^2 + |\Delta S|^2}{2|S_{12}S_{21}|} > 1 \tag{4}$$

$$B_1 = 1 + |S_{11}|^2 - |S_{22}|^2 - |\Delta S|^2 > 0 \tag{5}$$

with

$$\Delta S = S_{11}S_{22} - S_{12}S_{21}. \tag{6}$$

Although his analysis rests on the same principle, Kurokowa [4] gives three conditions:

$$K > 1$$

$$|S_{12}S_{21}| < 1 - |S_{11}|^2 \tag{7}$$

$$|S_{12}S_{21}| < 1 - |S_{22}|^2 \tag{8}$$

whereas Woods |1| uses

$$K > 1 \tag{9}$$

$$|\Delta S| < 1. \tag{10}$$

Finally, studying the input reflection factor for arbitrary passive loads, Carson [5] also proposes three conditions:

$$K > 1$$

$$|S_{11}| < 1 \tag{11}$$

$$|S_{22}| < 1. \tag{12}$$

II. Discussing Rollett's Criteria

Rollett's criteria consist of the main condition (1), which is common to all stability criteria and the two complementary conditions (2) and (3). It is shown below that

a) provided distinction is made between $K = +\infty$ and $K = -\infty$, the complementary conditions reduce to one;
b) they are in fact particular cases of a more general statement.

Let us first examine the case of nonunilateral devices ($Y_{21}Y_{12} \# 0$). Putting

$$Y_{12}Y_{21} = P + jQ$$

it follows from (3) that

$$2G_{11}G_{22} > \sqrt{P^2 + Q^2} + P$$

or

$$G_{11}G_{22} > 0.$$

In other words, if $K > 1$, then G_{11} and G_{22} can only be both positive or both negative.

In this derivation, it is assumed that $Y_{12}Y_{21} \# 0$. The property can however be easily extended to unilateral devices, if the

0098-4094/90/1100-1450$01.00 ©1990 IEEE