[5] T. A. Brubaker and J. C. Becker, "Multiplication using logarithms implemented with read-only memory," *IEEE Trans. Comput.*, vol. C-24, pp. 761–764, Aug. 1975.
[6] E. E. Swartzlander, Jr. and A. G. Alexopoulos, "The sign/logarithm number system," *IEEE Trans. Comput.*, vol. C-24, pp. 1238–1242, Dec. 1975.
[7] S. C. Lee and A. D. Edgar, "The FOCUS number system," *IEEE Trans. Comput.*, vol. C-26, pp. 1167–1170, Nov. 1977.
[8] T. Kurokawa, J. A. Payne, and S. C. Lee, "Error analysis of recursive digital filters implemented with logarithmic number systems," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 706–715, Dec. 1980.
[9] C. F. Kurth, K. J. Bures, P. R. Gagnon, and M. H. Etzel, "A per-channel, memory-oriented transmultiplexer with logarithmic processing," *IEEE Trans. Commun.*, vol. COM-30, pp. 1520–1527, July 1982.
[10] F. J. Taylor, "An extended precision logarithmic number system," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 232–234, Feb. 1983.
[11] E. E. Swartzlander, Jr., D. V. S. Chandra, H. T. Nagle, Jr., and S. A. Starks, "Sign/logarithm arithmetic for FFT implementation," *IEEE Trans. Comput.*, vol. C-32, pp. 526–534, June 1983.
[12] G. L. Sicuranza, "On efficient implementations of 2-D digital filters using logarithmic number systems," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 877–885, Aug. 1983.
[13] M. L. Frey and F. J. Taylor, "A table reduction technique for logarithmically architected digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 719–719, June 1985.
[14] A. G. Stouraitis, "Logarithmic number system theory, analysis and design," Ph.D. dissertation, Univ. Florida, Gainesville, 1986.

Efficient Impulsive Noise Suppression Via Nonlinear Recursive Filtering

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

Abstract—**A nonlinear recursive filter for the suppression of impulsive noises is proposed. This filter selects from each window a sample closest in value to the most recent output, and is thus named the *last output reference* (LOR) filter. A relationship between the LOR and recursive median filters is derived, and some statistical properties are studied through computer simulations. The results indicate that this filter preserves edges while suppressing impulsive noise. It is shown that LOR filters are more effective in suppressing impulses, and are often simpler to implement than median filters.**

I. Introduction

In digital signal processing, signals are sometimes corrupted by impulsive noise that appears as very large spikes of short duration. For example, in digital image or speech communications, channel transmission errors usually result in noise impulses in the received signal [1]–[3]. Various restoration techniques have been proposed for the suppression of impulses. Early techniques apply a linear operator such as averaging prior to using a threshold algorithm [2]. Since linear operators are sensitive to impulses, the performance of such restoration methods deteriorates rapidly as the probability of impulse occurrence increases. In addition, their performance is not satisfactory in the neighborhood of abrupt sustained changes in the signal (e.g., edges of an image) because linear operators often smooth such step changes. Recent impulse suppression techniques apply nonlinear filters, like median-type filters or generalized (or nonlinear) mean filters [2]–[7], which are inherently resistive to impulses and can preserve edges of a signal. It is often suggested to combine a threshold algorithm with these filters in order to suppress impulses without affecting the undistorted signal [4], [6]. In this correspondence, we propose an alternative nonlinear filter for impulse suppression and compare its impulse suppression characteristics to those of standard median (SM) and recursive median (RM) filters.

The SM and RM filters are defined as follows. Let $\{X(\cdot)\}$ and $\{Y(\cdot)\}$ be the input and the output, respectively, of median-type filters with window size $2N + 1$. The output of the SM filter is given by $Y(k) = \text{median}\{X(k-N), \cdots, X(k), \cdots, X(k+N)\}$, and the output of the RM filter is given by $Y(k) = \text{median}\{Y(k-N), \cdots, Y(k-1), X(k), \cdots, X(k+N)\}$.

II. The Last Output Reference Filter

Before describing the proposed filter, we state an observation for the SM filter which indicates that the SM filter tends to select a sample from each window close in value to the last output.

Observation 1: The output of an SM filter of span $2N + 1 > 3$ satisfies $Y(k) \in \{Y(k-1), X(N+2; k-1), X(N; k-1), X(k+N)\}$ where $X(i; k-1)$ is the ith smallest element spanned by the window centered at $k - 1$. The output $Y(k)$ is equal to $X(k+N)$ only if $X(N; k-1) \le X(k+N) \le X(N+2; k-1)$.

The proof of this observation is rather simple, and is omitted. This suggests the LOR filter defined as follows. Let $\{X(i) \mid 1 \le i \le L\}$ be the input sequence of length L. The output $Y(k)$ of the LOR filter with window size W is $X(k+i)$, $0 \le i \le W-1$ if $|X(k+i) - Y(k-1)| \le |X(k+j) - Y(k-1)|$ for all $j \ne i$, $0 \le j \le W-1$. Here we assume that $Y(1) = X(1)$. In addition, to account for the end effect, $W - 1$ samples with values equal to $X(L)$ are appended at the right end of the input sequence. Notice that $Y(k)$ is one of $\{X(k), X(k+1), \cdots, X(k+W-1)\}$, and that the window size W can be any positive integer. In LOR filtering, the output at each point is the sample value closest in value to the last output among the data inside the window.

The LOR filter is a nonlinear recursive filter like the RM filter. The following observation shows that these two filters become equivalent when the input is binary valued.

Observation 2: If the input $\{X(i) \mid 1 \le i \le L\}$ is a binary sequence, then the output of the RM filter of span $2N + 1$ is equivalent to that of the LOR filter of span $N + 1$. Here in RM filtering, to account for start up and end effects, $X(1)$ and $X(L)$, respectively, are repeated N times at the beginning and at the end of the input.

Proof: Let $Y_R(i)$ be the output of the RM filter of span $2N + 1$ and let $Y_L(i)$ be the output of the LOR filter of span $N + 1$. Note that $Y_L(i) \ne Y_L(i-1)$ if and only if $X(i) = X(i+1) = \cdots = X(i+N) \ne Y_L(i-1)$. Since $Y_L(1) = Y_R(1)$, it is sufficient to show that $Y_R(i) \ne Y_R(i-1)$ if and only if $X(i) = X(i+1) = \cdots = X(i+N) \ne Y_R(i-1)$. Suppose that $X(i) = X(i+1) = \cdots = X(i+N) \ne Y_R(i-1)$. Then $Y_R(i) = \cdots = Y_R(i+N) = X(i)$. Thus, $Y_R(i) \ne Y_R(i-1)$. Now let $Y_R(i) \ne Y_R(i-1)$. Then $Y_R(i-N) = \cdots = Y_R(i-1)$ since the output after one pass of the RM filter is a root [7]. Hence, $X(i) = \cdots = X(i+N) \ne Y_R(i-1)$. This completes the proof.

This equivalence does not hold unless the input is binary. Note that the impulse noise suppression characteristics of filters can be examined by inputting binary signals to filters. Observation 2 indicates that the LOR filter is inherently resistive to impulses like the RM filter. We can roughly say that in the suppression of impulses, the window size of the LOR filter can be smaller by a factor of two than that of a similarly effective RM (or SM) filter. This is shown statistically in the following section.

Manuscript received August 11, 1986; revised June 16, 1988. This work was supported by the National Science Foundation under Grant DCI-8611859.

The authors are with the Department of Electrical and Computer Engineering, State University of New York at Buffalo, Buffalo, NY 14260.

IEEE Log Number 8825142.

0096-3518/89/0200-0303$01.00 © 1989 IEEE

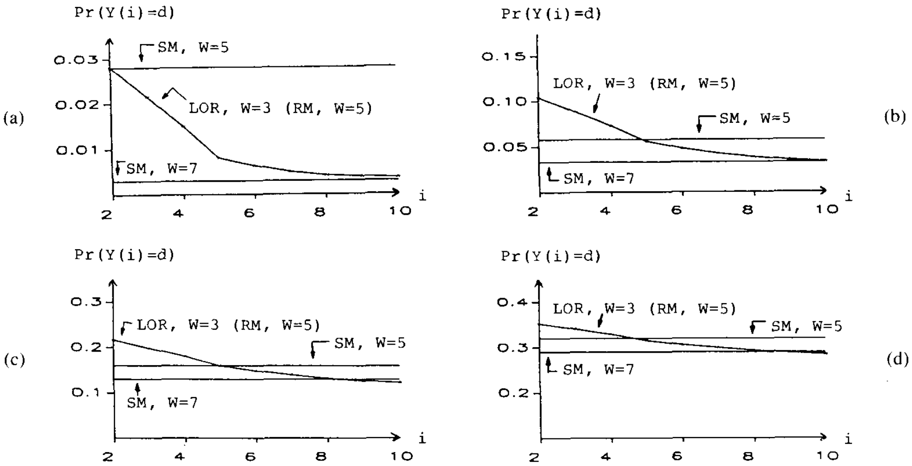


Fig. 1. Error probabilities associated with the LOR, SM, and RM filters, (a) $p = 0.1$, (b) $p = 0.2$, (c) $p = 0.3$, and (d) $p = 0.4$.

III. Statistical Properties

In this section, we gain some insight into the statistical properties of the LOR filter through computer simulation, and the results are compared to the properties of SM and RM filters. We shall show that LOR filters can preserve edges while suppressing impulses.

A. Impulse Noise Attenuation

The signal model in [8] is considered. The input $X(i)$ is a constant signal corrupted by impulses given by $X(i) = d$ with probability (w.p.) P and $X(i) = s$ w.p. $1 - P$ where s is the signal value and d is the value of impulses. We evaluated the error probability $\Pr\{Y(i) = d\}$ associated with the LOR filter of size 3 as follows. 1) All possible binary sequences of length 12 were generated. 2) Each sequence was LOR filtered. 3) The error probability associated with $Y(i)$ was obtained by $\Pr\{Y(i) = d\} = \Sigma_j P_j I(Y_j(i))$ where P_j is the probability of occurrence of the jth sequence, $1 \le j \le 2^{12}$, and $I(Y_j(i))$ is the indicator function which is equal to one if $Y_j(i)$ associated with the jth sequence is d, and zero otherwise. Note that the error probability of the LOR filter with size 3 is equal to that of the RM filter with size 5 because the input is binary. The results are shown in Fig. 1. The error probabilities associated with SM filters are also shown in Fig. 1, which are from [8]. Since we assume that $Y(1) = X(1)$ in LOR filtering, $\Pr\{Y(1) = d\} = P$, but $\Pr\{Y(i) = d\}$ decreases rapidly as i increases. When $i = 10$, the performance of the LOR filter with size 3 is comparable to that of the SM filter with size 7. Therefore, we can say that impulse suppression characteristics of the LOR filter with size 3 and the RM filter with size 5 are similar to that of the SM filter with size 7.

B. Response to Noisy Edges

Next the action of LOR filters on noisy step edges is studied. The input sequence representing a noisy step edge is as follows: $X(i) = N(i)$ if $1 \le i \le 10$, and $X(i) = h + N(i)$ if $10 < i \le 20$ where $N(i)$ is independently and identically distributed (i.i.d.) with normal density $N(0, 1)$. We are mainly interested in edges with large h, say $h > 4$, because for such edges, edge preserving properties of filters are more notable. Throughout this subsection, h is set to be 6. We generated 10 000 samples of the sequence $\{X(i) \mid 1 \le i \le 20\}$, and the empirical expected values and root mean-squared error (rmse) of the SM, RM, and LOR filtered signals have been calculated from these sample sequences. Here the rmse is defined as $\sqrt{E[Y(i) - s]^2}$ with $s = 0$ if $1 \le i \le 10$, and $s = 6$ if $10 < i \le 20$. The results are shown in Fig. 2. The LOR, SM, and RM filters tended to preserve the edge, while the SM and RM filters performed slightly better than the LOR filter.

IV. Computational Aspects

We first show that the output of the LOR fitler need *not* be evaluated at each point because the LOR filter tends to produce piecewise constant signal segments.

Observation 3: If the output of the LOR filter is $Y(k) = X(k + m)$ for some m, $1 \le m \le W - 1$, then $Y(k) = Y(k + 1) = \cdots = Y(k + m)$.

The proof is trivial. This observation indicates the interesting fact that in LOR filtering, the window can be advanced with jumps. For example, if $Y(k) = X(k + m)$ for some m, $1 \le m \le W - 1$, then $Y(k + 1) = \cdots = Y(k + m) = X(k + m)$ and the window can jump from k to $k + m + 1$ without evaluating the outputs between the two points. The interval m during which the outputs remain constant is called the *jump interval*. By advancing the window with jumps, the computational cost in LOR filtering is considerably reduced and LOR filters become computationally very efficient. In particular, for i.i.d. inputs, we can show that the average number of operations required to evaluate an output of the LOR filter is independent of the window size. This is shown below through the evaluation of the empirical expected value of the jump interval.

We generated 1000 sequences of length 256 which are i.i.d. with density $N(0, 1)$. The sequences were LOR filtered and the average number of points whose outputs should be calculated was obtained. The expected jump interval was estimated empirically by dividing 256 by the average. The results are shown in Table I. It is interesting to note that the resulting expected jump intervals are approximated by $W/2 + 0.5$ which is the center point of the window with size W. Thus, when the input is i.i.d., the expected jump interval is approximately half of the window size, and the expected number of output points that should be calculated is given by $L/[W/2 + 0.5]$ where L is the length of the input sequence to be LOR filtered. Suppose that the output $Y(k)$ of the LOR filter is obtained as follows. 1) Evaluate $|X(k + i) - Y(k - 1)|$ for each i, $0 \le i \le W - 1$. 2) $Y(k) = X(k + j)$ if $|X(k + j) - Y(k - 1)|$ is the minimum among the absolute differences. Step 1) requires W subtractions and step 2) requires $W - 1$ comparisons. When an i.i.d. sequence of length L is LOR filtered, the expected total number of substractions and comparisons are $LW/[W/2 + 0.5]$ and $L(W - 1)/[W/2 + 0.5]$, respectively, which are less than $2L$. Thus, the expected average number of operations in evaluating an output is at most two substractions and two comparisons irrespective of the window size.

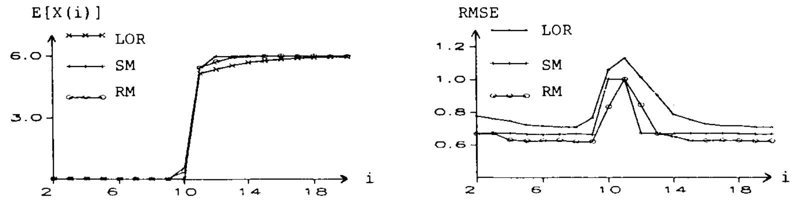
(a)  (b)

Fig. 2. (a) Empirical expected values and (b) rmse associated with LOR, SM, and RM filters with $W = 3$ for the input edge.

TABLE I
EMPIRICAL EXPECTED JUMP INTERVALS

	Window Size							
	2	3	4	5	6	7	8	9
Expected Jump Intervals	1.51	2.02	2.53	3.04	3.54	4.04	4.54	5.03

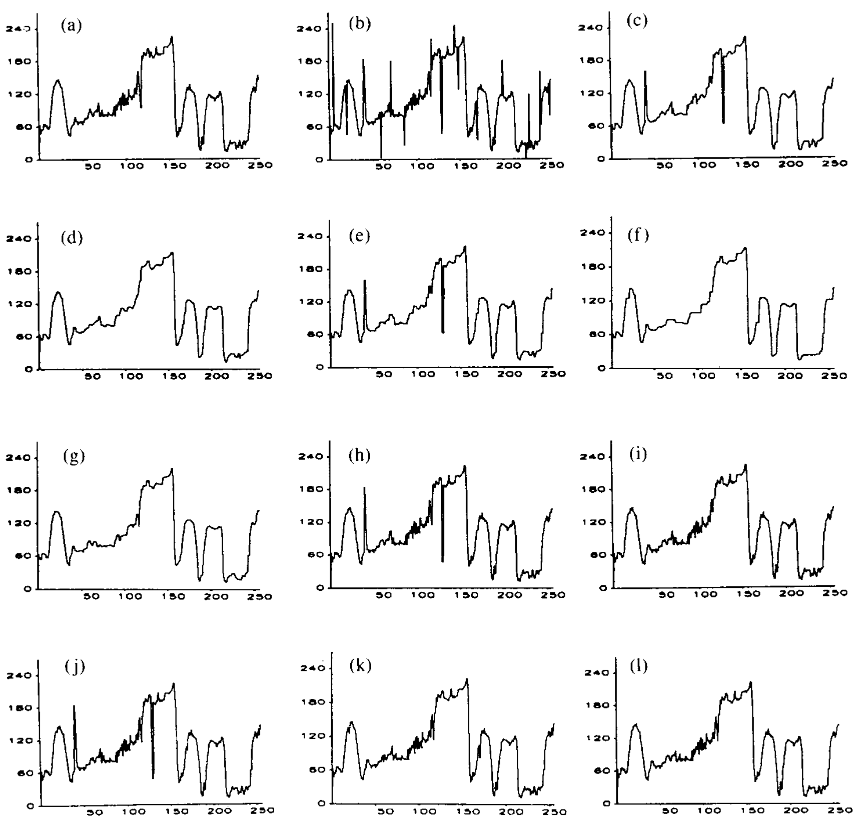


Fig. 3. (a) The original test signal, (b) the signal degraded by impulses, the results of (c) SM filtering, $W = 3$, (d) SM filtering, $W = 5$, (e) RM filtering, $W = 3$, (f) RM filtering, $W = 5$, (g) LOR filtering, $W = 3$. The enhanced signals (threshold $T = 25$); (h) via SM filtering, $W = 3$, (i) via SM filtering, $W = 5$, (j) via RM filtering, $W = 3$, (k) via RM filtering, $W = 5$, and (l) via LOR filtering, $W = 3$.

Although the property in Observation 3 is useful in implementation, this property implies that LOR filters tend to cause more signal distortion than SM and RM filters. A remedy for this is the use of a proper threshold algorithm after LOR filtering. The experimental results presented in the following section indicate that LOR filtering followed by a threshold algorithm can suppress impulse noises without disturbing noise-free signal points.

V. Experimental Results

An LOR filter is applied to enhance a signal degraded by impulsive noise. The following threshold algorithm is combined with filtering operations [6]: $Z(i) = Y(i)$ if $|X(i) - Y(i)| > T$ and $Z(i) = X(i)$ otherwise. Here $X(i)$ and $Y(i)$, respectively, are the input and output of a filter, T is a threshold, and $Z(i)$ is the enhanced signal.

The original signal which is a row of an image is shown in Fig. 3(a) ($L = 256$). Fig. 3(b) shows the signal degraded by impulses. This signal was filtered by SM and RM filters with sizes 3 and 5, and the LOR filter with size 3, and then the above threshold algorithm with $T = 25$ was applied. The results are shown in Fig. 3(c)–(l). The LOR filter with size 3 performed like the SM and RM filters with size 5.

VI. Conclusions

The LOR filter which is useful for impulsive noise suppression was introduced. Impulse suppression via LOR filtering seems more effective, and is often simpler to implement than that via SM or RM filtering. This is so because the LOR filter often requires less computation than the SM or RM filter with the same window, and moreover, a smaller window can be used in LOR filtering.

Since RM filters have the threshold decomposition property [9], from Observation 2, RM filters can be implemented by using the LOR filter structure as follows. The input is decomposed into a set of binary signals, and each binary signal is LOR filtered with window $N + 1$. Recombining the filtered binary data results in the output of the RM filter of span $2N + 1$. This seems to be another promising application of LOR filtering, and requires further research.

References

[1] N. S. Jayant, "Average and median-based smoothing techniques for improving speech quality in the presence of transmission errors," *IEEE Trans. Commun.*, vol. COM-24, pp. 1043–1045, Sept. 1976.
[2] W. K. Pratt, *Digital Image Processing*. New York: Wiley, 1978.
[3] T. S. Huang, Ed., *Two-Dimensional Digital Signal Processing II: Transforms and Median Filters*. New York: Springer-Verlag, 1981.
[4] A. Kundu, S. K. Mitra, and P. P. Vaidyanathan, "Application of two-dimensional generalized mean filtering for removal of impulse noises from images," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 600–609, June 1984.
[5] I. Pitas and A. N. Venetsanopoulos, "Nonlinear mean filters in image processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 573–584, June 1986.
[6] A. Nieminen, P. Heinonen, and Y. Neuvo, "Suppression and detection of impulsive type interference using adaptive median hybrid filters," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Dallas, TX, Apr. 1987, pp. 117–120.
[7] T. A. Nodes, and N. C. Gallagher, Jr., "Median filters: Some modifications and their properties," *IEEE Trans., Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 739–746, Oct. 1982.
[8] B. I. Justusson, "Median filtering: Statistical properties," in *Two-Dimensional Digital Signal Processing II*, T. S. Huang, Ed. New York: Springer-Verlag, 1981.
[9] 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.

Frequency Difference of Arrival Accuracy

ROBERT K. OTNES

Abstract—**The problem considered occurs in the geolocation of a fixed emitter using observations from two moving collectors. Estimates for the time difference and frequency difference of signal arrivals are employed. Frequency difference of arrival (FDOA) is estimated through the use of a mixing product. Standard regression analysis procedures are then applied to estimate the slope of the unwrapped phase angle. The derivation of an expression for the standard error of FDOA is given, and this result is related to several other well-known results.**

Manuscript received October 13, 1987; revised June 16, 1988.
The author is with MAXIM Technologies, Inc., Santa Clara, CA 95054.
IEEE Log Number 8825143.

Introduction

The June 1981 issue of the IEEE Transactions on Acoustics, Speech, and Signal Processing was a special issue devoted largely to time delay estimation. The papers by Piersol [2] and Stein [3] were of particular interest to this author.

Piersol developed a method for computing the relative time delay of a signal as observed by two different collectors through the technique of fitting a straight line to the phase of the mixing product, and then finding the delay as the slope of that line. An advantage of this method is that well-known regression analysis techniques enable one to find the sample variability of the resulting delay. This sample variability is very useful when evaluating results.

This correspondence addresses the problem of computing relative Doppler, or *frequency difference of arrival* (FDOA), using the regression technique Piersol used for delay, and then gives a derivation of the standard error formula of Stein based on the expression for standard error found in regression analysis. This derivation is elementary and requires few assumptions as to the nature of the signal and the noise functions received. It is not general in that it assumes narrow-band filtering of the mixing product.

Definition of the Problem

Assume that a signal of interest is being collected by two spatially separated receivers, so that $r_1(t)$ is seen by receiver one and $r_2(t)$ is the signal observed by receiver two. Next suppose that the two collectors see slightly different versions of the signal of interest, with different noise being added into each signal. Denote the noise by $n_1(t)$ and $n_2(t)$. The noise is assumed to be stationary with means and variances given by

$$\mu = E[n_1(t)] = E[n_2(t)] = 0$$
$$\sigma_k^2 = E\left[|n_k(t)|^2\right], \; k = 1, 2. \tag{1}$$

Also, the assumption is made that the noise is uncorrelated:

$$E[n_1(t) n_2^*(t + \tau)] = 0, \quad \text{all } \tau. \tag{2}$$

A somewhat simplified model will be employed, one that ignores time delay. Effectively, the assumption is that all delay has been removed from the data. The received signals then are

$$r_k(t) = A_k m(t)\, e^{j(\omega_k t + \phi_k)} + n_k(t), \qquad k = 1, 2. \tag{3}$$

The definitions of these terms are as follows:

$$m(t) = \text{the modulation}$$
$$A_k = \text{the received amplitude of signal } k$$
$$\omega_k = \text{the received carrier frequency of signal } k$$
$$\phi_k = \text{the received random phase of signal } k. \tag{4}$$

Next, define the *mixing product* function $p(t)$ by

$$\begin{aligned} p(t) &= r_1(t)\, r_2^*(t) \\ &= A_1 A_2^* |m(t)|^2\, e^{j[(\omega_1 - \omega_2)t + (\phi_1 - \phi_2)]} \\ &\quad + A_1 m(t)\, e^{j(\omega_1 t + \phi_1)}\, n_2^*(t) + A_2^* m(t)\, e^{-j(\omega_2 t + \phi_2)} n_1(t) \\ &\quad + n_1(t)\, n_2^*(t). \end{aligned} \tag{5}$$

Taking the expectation of the mixing product, we have

$$\begin{aligned} E[p(t)] &= E[r_1(t) r_2^*(t)] = A_1 A_2^* |m(t)|^2 e^{j[(\omega_1 - \omega_2)t + (\phi_1 - \phi_2)]} \\ &= A_1 A_2^* |m(t)|^2\, e^{j(\omega_0 t + \phi_0)} \end{aligned} \tag{6}$$