Hindawi
International Journal of Antennas and Propagation
Volume 2018, Article ID 3574929, 7 pages
https://doi.org/10.1155/2018/3574929

Research Article

Broadband Frost Adaptive Array Antenna with a Farrow Delay Filter

Tao Dong,[1,2] Qiushi Wang,[3] Yunxiao Zhao,[3] Lixia Ji,[3] and Hao Zeng[3]

[1]*State Key Laboratory of Space-Ground Integrated Information Technology, Beijing 100095, China*
[2]*Beijing Institute of Satellite Information Engineering, Beijing 100095, China*
[3]*College of Communication Engineering, University of Chongqing, Shazheng Street 174, Chongqing 400044, China*

Correspondence should be addressed to Hao Zeng; haoz@cqu.edu.cn

Received 3 March 2018; Revised 4 July 2018; Accepted 30 July 2018; Published 3 September 2018

Academic Editor: Rodolfo Araneo

In a broadband adaptive array antenna with a space-time filter, a delay filter is required before digital beamforming when the Frost algorithm is used to obtain the weight vector. In this paper, we propose a Farrow structure instead of a direct form FIR structure to implement the time delay filter since it can satisfy the demand for real-time update and is very suited for the FPGA platform. Furthermore, a new off-line algorithm to calculate the Farrow filter coefficient is presented if the filter coefficient is symmetric. Finally, simulations are presented to illustrate that the design methodology for the Farrow filter is correct. The simulations also prove that the Frost broadband adaptive antenna effectively mitigates the interferences with a Farrow filter.

1. Introduction

In a broadband adaptive array antenna with space-time beamforming, each beamformer channel is a transversal filter. The Frost algorithm is a classic method to calculate the weight vector in space-time beamforming. In contrast to a narrowband system [1], Frost's broadband adaptive array demands that the desired signal must impinge normally on the array face [2]. A delay filter is needed before beamforming if the desired signal does not satisfy the normal incident requirement. There is a large body of work focusing on how to obtain the weight vector with the Frost algorithm ([3–5] and the references therein). However, very limited research has been done on the design of the delay filter in a broadband Frost array.

The traditional delay filter takes a direct form structure. With this structure, the methodology for delay filter design can be divided into two categories: (a) time-domain methodology [6] and (b) frequency-domain methodology [7]. Both methodologies are online methodologies. Typically, the Frost broadband array delay filter [8] is designed using the time-domain methodology [6]. One downside of using the time-domain methodology is that different delay times result in different filter coefficients, which implies that we have to redesign the filter coefficients if the direction of arrival (DOA) of the desired signal changes because DOA decides the delay time. Often, the DOA of the desired signal changes very fast, especially for high-speed vehicles such as missiles. So, updating the filter in real time is almost impossible when the delay filter is implemented on the FPGA [9]. To overcome this disadvantage, we propose the Farrow structure to realize the delay filter for a Frost broadband adaptive antenna. The Farrow filter structure for a delay filter was first proposed by Farrow [10]. The design for a Farrow filter is done in an off-line manner; that is, the coefficients of the Farrow filter need to be designed only once and kept constant even if the delay requirement is changed. So, the Farrow filter is often used in multirate signal processing [11, 12], but a broadband adaptive array employing a Farrow filter has not been researched before.

The Farrow filter can be designed in an off-line manner. Convex optimization [13] and neural networks [14] can be used to compute the coefficients of each subfilter in the Farrow filter, but these methods have large implementation complexity. The weighted least squares (WLS) algorithm presented in [15] is another approach for Farrow filter design, but [15] did not show how to choose the weight value. An improved WLS algorithm is proposed in [16] that improves the performance under the assumption that the weight function is separable and piecewise constant. In [17], it is pointed out that the coefficients of the Farrow filter are symmetric, which allows the modification of the WLS algorithm to reduce the computational cost further.

This paper is organized as follows. Background information is given in this section. The Frost broadband adaptive array with space-time beamforming and the Farrow delay filter are discussed in Section 2. Farrow filter design and performance analysis are detailed in Section 3. The simulation work is presented in Section 4. Finally, the paper is concluded in Section 5.

2. Broadband Space-Time Adaptive Filtering

2.1. Space-Time Digital Beamforming. Different from the narrow band adaptive array, the space-time beamformer in a broadband adaptive array uses a transversal filter while the narrow band array uses only a complex multiplier at each channel. The space-time beamformer is shown in Figure 1 [2], where the array antenna has M elements and the order of each transversal filter is $J-1$.

Supposing that the received signal by the mth element is $x_m(t)$, the signal at each node could be described as

$$x_{m,i}(t) = x_m(t - iT_s), \quad m = 0, \dots, M-1, i = 0, \dots, J-1. \tag{1}$$

Clearly, T_s is the sampling period. Then, the whole signal in the space-time filter is an MJ dimension column vector; that is,

$$\mathbf{x}(t) = \left[\mathbf{x}_0(t) \cdots \mathbf{x}_{J-1}(t)\right]^T, \tag{2}$$

where

$$\mathbf{x}_i(t) = \left[x_{0,i}(t) \quad \cdots \quad x_{M-1,i}(t)\right]^T. \tag{3}$$

Obviously, the corresponding weight for the whole beamformer is

$$\mathbf{w} = \left[\mathbf{w}_0 \quad \cdots \quad \mathbf{w}_{J-1}\right]^T. \tag{4}$$

And we see

$$\mathbf{w}_i = \left[w_{0,i} \quad \cdots \quad w_{M-1,i}\right]^T. \tag{5}$$

So the output of the array is

$$y(t) = \mathbf{w}^H \mathbf{x}(t) = \sum_{m=0}^{M-1} \sum_{i=0}^{J-1} w^*_{m,\mathrm{i}} x_{m,i}(t). \tag{6}$$

The minimum variance distortionless response (MVDR) beamformer assumes the desired signal induced into the array from the normal direction, so that all the elements

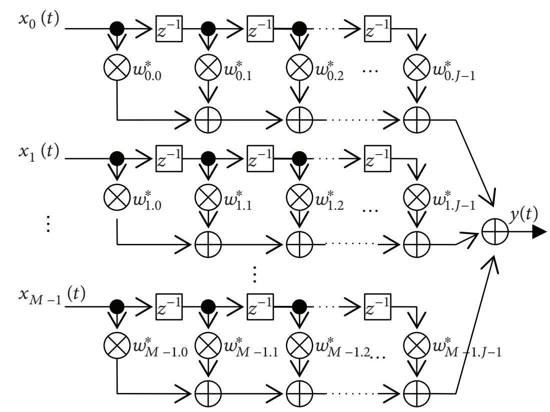

Figure 1: The structure of the space-time beamformer.

receive the same desired signal without any phase difference [2, 8]. Therefore, the optimal problem for space-time beamforming is as follows:

$$\mathbf{w}_{\text{opt}} = \arg\min_{w} \quad \mathbf{w}^H \mathbf{R}_x \mathbf{w} \quad \text{subject to } \mathbf{C}^H \mathbf{w} = \mathbf{f}, \tag{7}$$

where the covariance matrix $\mathbf{R}_x$ is

$$\mathbf{R}_x = E\left[\mathbf{x}(t)\mathbf{x}^H(t)\right]. \tag{8}$$

The symbol $E[\cdot]$ indicates the mean value.

$$\mathbf{f} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T, \qquad \mathbf{C} = \begin{bmatrix} \mathbf{e} & & 0 \\ & \cdots & \\ 0 & & \mathbf{e} \end{bmatrix}, \tag{9}$$

and

$$\mathbf{e} = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T. \tag{10}$$

The constraint requires the signal to impinge normally on the array, but this is not true generally. Then, a delay filter is required in serial with beamforming to adjust the input signal, so that the signal phase for each element is equal. The scheme of the broadband adaptive array is illustrated in Figure 2.

2.2. Frost Algorithm for Beamforming. To solve the optimum problem of (7), the Frost algorithm is employed which is a kind of LMS adaptive filter. The weight update equation of the Frost algorithm is

$$\mathbf{w}(n+1) = \mathbf{C}\left(\mathbf{C}^H\mathbf{C}\right)^{-1}\mathbf{f} + \mathbf{P}[\mathbf{w}(n) - \mu y^*(n)\boldsymbol{x}(n)], \tag{11}$$

where μ is the step size to assure convergence of the iteration only if

$$\mu < \frac{2}{\lambda_{\max}}. \tag{12}$$

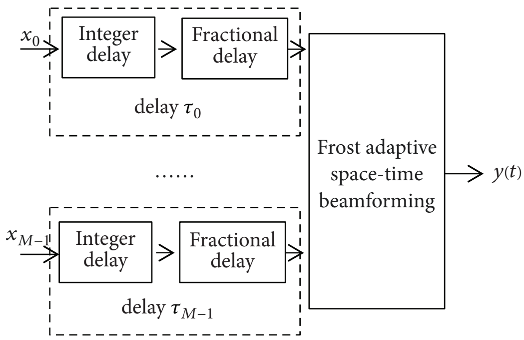

FIGURE 2: The scheme of the beamformer with a delay module.

In the above equation, $\lambda_{\max}$ is the maximum eigenvalue of the covariance matrix $\mathbf{R}_x$. The output term is

$$y(n) = \mathbf{w}^H(n)\mathbf{x}(n). \tag{13}$$

The constant matrix is

$$\mathbf{P} = \mathbf{I} - \mathbf{C}\left(\mathbf{C}^H\mathbf{C}\right)^{-1}\mathbf{C}^H. \tag{14}$$

3. Farrow Filter Design

3.1. Farrow Filter Model. The digital delay filter in Figure 2 is an FIR filter. If the order of the filter is N, the coefficient vector can be expressed as

$$\mathbf{h} = [h_0 \quad \cdots \quad h_N]^T. \tag{15}$$

The delay includes n-time sampling periods and a term T_l which is shorter than one sampling period T_s; namely,

$$\tau = nT_s + T_l. \tag{16}$$

In the FPGA hardware platform, realizing delay nT_s is easy through clock control. Hence, the role of the delay filter is to deal with the delay T_l. This kind of delay filter is also called variable fractional delay filters [11–17]. We define the delay parameter as

$$D = \frac{T_l}{T_s} \in [-0.5, 0.5]. \tag{17}$$

Both time-domain and frequency-domain methods can design the coefficient vector $\mathbf{h}$ of the delay filter. Since the delay parameter D is decided by the DOA of the desired signal, it is necessary to redesign the coefficient vector $\mathbf{h}$ at each time when DOA changes. If FPGA is used to implement the delay filter, the real-time requirements cannot be guaranteed. Hence, a Farrow filter is proposed to solve this problem. Although the Farrow filter is used in software radio widely [11–14], it has not been considered for a broadband array before.

As for the Farrow digital delay filter, its each coefficient is regarded as M-order polynomial of the delay parameter D; namely,

$$h_n = \sum_{m=0}^{M} h_{nm}D^m, \quad n = 0, 1, \ldots, N. \tag{18}$$

So the Farrow filter can be derived into $N+1$ subfilters as shown in Figure 3. So long as we adjust the delay parameter D, the delay T_l will be manipulated without change h_{nm}.

The Farrow filter has symmetric coefficients to assure the linear phase [17]. For the sake of convenience, we redefine the coefficient vector of the Farrow filter as

$$\mathbf{c} = [c_{-N} \quad \cdots \quad c_N]^T. \tag{19}$$

Obviously, according to the definition of the Farrow structure, we have

$$c_n = \sum_{m=0}^{M} c(n,m)D^m, \quad n = -N, \ldots, 0, \ldots, N. \tag{20}$$

Meanwhile, assuming that the received signal of the array is a low pass signal and the delay filter is a low pass filter, so the pass band is $[0, \alpha\pi]$, in which

$$0 < \alpha < 1. \tag{21}$$

3.2. Coefficient-Symmetric Farrow Filter Design with WLS Methodology. The system function of the ideal delay filter is expressed as

$$H(\omega, D) = e^{-j\omega D}. \tag{22}$$

On the other hand, the frequency characteristic of the Farrow filter is

$$C(\omega, D) = \sum_{n=-N}^{N}\sum_{m=0}^{M} c(n,m)D^m e^{-jn\omega}. \tag{23}$$

Now, we define a weighted error function as

$$J(c) = \int_0^{\alpha\pi}\int_{-0.5}^{0.5} W(\omega, D)|E(\omega, D)|^2 d\omega dD. \tag{24}$$

The error function in the equation above is

$$E(\omega, D) = C(\omega, D) - H(\omega, D). \tag{25}$$

At the same time, $W(\omega, D)$ is a nonnegative function, which has these two characteristics:

$$\begin{aligned} W(\omega, D) &= W_1(\omega)W_2(D), \\ W(\omega, -D) &= W(\omega, D). \end{aligned} \tag{26}$$

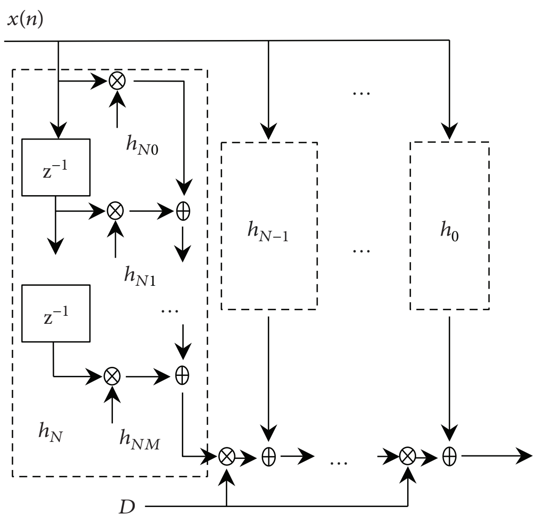

Figure 3: The structure of the Farrow filter.

With regard to the definitions above, Farrow filter design is a process of looking for suitable coefficient set $\mathbf{c}$, namely, $\mathbf{c}(n, m)$, to minimize $J(\mathbf{c})$ according to the band width $\alpha\pi$ and delay parameter D.

Since the coefficients of the Farrow Filter are symmetric, we have the conclusion that

$$c(-n,m)=\begin{cases} c(n,m), & \text{even } m,\\ -c(n,m), & \text{odd } m,\end{cases}\qquad c(0,m)=0,\quad \text{odd } m. \tag{27}$$

By substituting the conclusion into (23), we get

$$C(\omega, D)=\mathbf{a}^T\mathbf{B}_e\mathbf{p}_e - j\mathbf{b}^T\mathbf{B}_o\mathbf{p}_o, \tag{28}$$

where

$$\mathbf{a}=\begin{bmatrix}1 & \cos(\omega) & \cdots & \cos(N\omega)\end{bmatrix}^T, \tag{29}$$

$$\mathbf{b}=\begin{bmatrix}1 & \sin(\omega) & \cdots & \sin(N\omega)\end{bmatrix}^T, \tag{30}$$

$$\mathbf{B}_e=\begin{bmatrix}\beta(0,0) & \beta(0,2) & \beta(0,4) & \cdots & \beta(0,M-1)\\ \beta(1,0) & \beta(1,2) & \beta(1,4) & \cdots & \beta(1,M-1)\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ \beta(N,0) & \beta(N,2) & \beta(N,4) & \cdots & \beta(N,M-1)\end{bmatrix}, \tag{31}$$

$$\mathbf{B}_o=\begin{bmatrix}\beta(1,1) & \beta(1,3) & \beta(1,5) & \ldots & \beta(1,M)\\ \beta(2,1) & \beta(2,3) & \beta(2,5) & \ldots & \beta(2,M)\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ \beta(N,1) & \beta(N,3) & \beta(N,5) & \ldots & \beta(N,M)\end{bmatrix}. \tag{32}$$

The element of the matrix is

$$\begin{gathered}\beta\left(0,2m'\right)=c\left(0,2m'\right),\quad n=0,\\ \beta\left(n,2m'\right)=2c\left(n,2m'\right),\quad n>0,\\ \beta\left(n,2m'+1\right)=2c\left(n,2m'+1\right),\quad n>0,\\ m'=0,\ldots,M'.\end{gathered} \tag{33}$$

Substituting the result above into error function and taking the conjugate property of the weighted function into consideration, we get the weighted error function.

$$J(\mathbf{c})=J(\mathbf{B}_e,\mathbf{B}_o)=\int_0^{\alpha\pi}\int_0^{0.5}W(\omega,D)\left|E(\omega,D)\right|^2 d\omega dD. \tag{34}$$

Compared to (24), the integration of the delay parameter D is defined in $[0, 0.5]$. Hence, the process to solve $\mathbf{c}$ (n, m) can be converted into the process to solve the matrix $(\mathbf{B}_e, \mathbf{B}_o)$ which minimizes $J(\mathbf{B}_e, \mathbf{B}_o)$.

The weighted error function can be expressed as

$$\begin{aligned}J(\mathbf{B}_e,\mathbf{B}_o)=&-2tr[\mathbf{B}_e,\mathbf{A}_1]+tr\left[\mathbf{B}_e\mathbf{A}_2\mathbf{B}_e^T\mathbf{A}_3\right]+tr\left[\mathbf{B}_o\mathbf{A}_4\mathbf{B}_o^T\mathbf{A}_5\right]\\ &-2tr[\boldsymbol{B}_o\boldsymbol{A}_6]+\text{constant},\end{aligned} \tag{35}$$

in which

$$\begin{aligned}\mathbf{A}_1&=\int_0^{\alpha\pi}\int_0^{0.5}W_1(\omega)W_2(D)\cos(\omega D)\mathbf{p}_e\mathbf{a}^T d\omega dD,\\ \mathbf{A}_2&=\int_0^{0.5}W_2(D)\mathbf{p}_e\mathbf{p}_e^T dD,\\ \mathbf{A}_3&=\int_0^{\alpha\pi}W_1(\omega)\mathbf{a}\mathbf{a}^T d\omega,\\ \mathbf{A}_4&=\int_0^{0.5}W_2(D)\mathbf{p}_o\mathbf{p}_o^T dD,\\ \boldsymbol{A}_5&=\int_0^{\alpha\pi}W_1(\omega)\mathbf{b}\mathbf{b}^T d\omega,\\ \mathbf{A}_6&=\int_0^{\alpha\pi}\int_0^{0.5}W_1(\omega)W_2(D)\sin(\omega D)\mathbf{p}_o\mathbf{b}^T d\omega dD,\end{aligned} \tag{36}$$

while

$$\begin{gathered}\mathbf{p}_e=\begin{bmatrix}D^0 & D^2 & \cdots & D^{M-1}\end{bmatrix}^T,\\ \mathbf{p}_o=\begin{bmatrix}D^1 & D^3 & \cdots & D^M\end{bmatrix}^T.\end{gathered} \tag{37}$$

Assuming that the weight is known, all the equations above can be solved. Considering that A_2, A_3, A_4, and A_5 are symmetric and positively definite, they can be factorized into U_2, U_3, U_4, and U_5, respectively, via the Cholesky factorization.

Finally, according to the Lagrangian multiplier method, taking the derivative of $J(\mathbf{B}_e, \mathbf{B}_o)$ and letting it to be zero, we get

$$\begin{aligned}\mathbf{B}_e &= \mathbf{U}_3^{-1}\left(\mathbf{U}_3^T \mathbf{A}_1^T \mathbf{U}_2^{-1}\right) U_2^{-T},\\ \mathbf{B}_o &= \mathbf{U}_5^{-1}\left(\mathbf{U}_5^T \mathbf{A}_6^T \mathbf{U}_4^{-1}\right) U_4^{-T}.\end{aligned} \tag{38}$$

According to the definition (31), (32), and (33), $\mathbf{c}(n, m)$ can be determined.

3.3. Weight Determination. Generally, the weight $W(\omega, D)$ is unknown to us. In order to decide the correct $W(\omega, D)$, iterative design is proposed. When beginning to design, we often let

$$W_1(\omega) = W_2(D) = 1. \tag{39}$$

Following the methodology in the previous section, we can obtain the designed coefficient $\mathbf{c}(n, m)$. Then, we define the relative error as

$$\varepsilon_e = \left[\frac{\int_0^{\alpha\pi}\int_{-0.5}^{0.5} |E(\omega, D)|^2 d\omega dD}{\int_0^{\alpha\pi}\int_{-0.5}^{0.5} |H(\omega, D)|^2 d\omega dD}\right]^{1/2} \times 100\%. \tag{40}$$

If the required relative error cannot be satisfied, we define the weight function as segmentation:

$$\begin{aligned}W_1(\omega) &= \begin{cases} s_1, & \omega \in [0, \omega_0], \\ s_2, & \omega \in [\omega_0, \alpha\omega], \end{cases}\\ W_2(D) &= \begin{cases} s_3, & D \in [0, D_0], \\ s_4, & D \in [D_0, 0.5]. \end{cases}\end{aligned} \tag{41}$$

For some interval with greater error, we set a larger weight and redesign $C(\omega, D)$. Then, the error in this interval could be reduced.

4. Simulation Result

4.1. Proposed WLS Farrow Filter Design Methodology. In our simulations, we use a linear array with 5 elements. The radio frequency and intermediate frequency of the received signal are 2 GHz and 400 MHz, respectively. Assuming that the sampling frequency is 1.25 GHz and the bandwidth is 300 MHz, the signal frequency is located in the interval [0,0.9π] since the first Nyquist domain is 0~312 MHz. In addition, the delay interval is [0,0.5]. We set the order of the Farrow delay filter as $N = 34$ and $M = 7$.

We prove the validity of the proposed methodology for the Farrow filter design by evaluating the error $E(\omega, D)$, as defined in (25). We set the initial weight function as in (39) and compute the coefficient of the Farrow filter by the steps listed in Section 3.2. After obtaining the coefficient vector $\mathbf{c}$, it is easy to get the amplitude error $E(\omega, D)$ as shown in Figure 4.

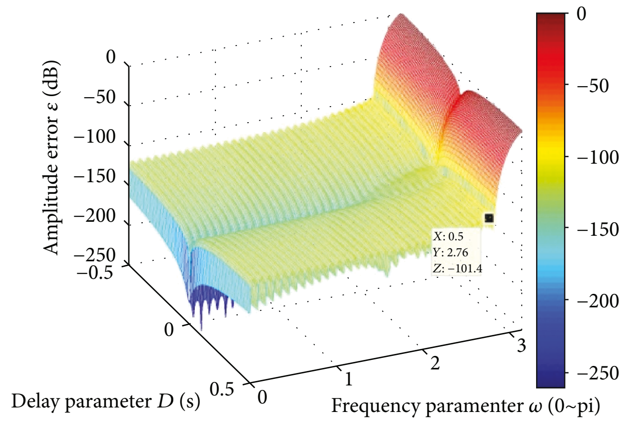

Figure 4: Error with weight function by (39).

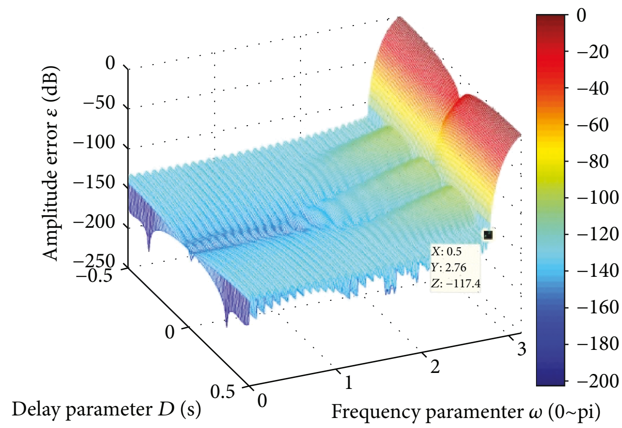

Figure 5: Error with weight function by (42).

We focus on the interval where the desired signal exits. We notice that the large error is located in the intervals [0,0.88π] and [0.4,0.5]. If this error meets the requirement, we finish the design and take $\mathbf{c}$ as the final result. And if not, we enlarge the weight function such as

$$\begin{aligned}W_1(\omega) &= \begin{cases} 9700, & \omega \in [0, 0.88\pi], \\ 1, & \omega \in [0.88\pi, 0.9\pi], \end{cases}\\ W_2(D) &= \begin{cases} 1, & D \in [0, 0.4], \\ 3700, & D \in [0.4, 0.5], \end{cases}\end{aligned} \tag{42}$$

and we redesign the Farrow filter again. The new error is shown in Figure 5, where the error is less than about 6 dB (Figure 4) due to the weight function. In fact, the error is also impacted by the order of the Farrow filter which will be discussed in the next section.

4.2. Influence of the Filter Order. At the same simulation condition as the above, we only change the order of the filter and compare the error between the proposed Farrow and direct form structure in [6–8]. A comparison of the relative error ε_e, defined in (40), for different filter structures is listed in Table 1. The result in Table 1 illustrates that the direct form

Table 1: Relative error with different filter orders.

Design method	Order 4 ε_e	6 ε_e	12 ε_e	30 ε_e	56 ε_e
Time domain [6, 8]	7.92%	7.01%	5.43%	3.70%	2.82%
Frequency domain [7]	6.80%	5.17%	2.82%	0.76%	0.17%
Farrow structure	15.23%	13.25%	3.55%	1.54%	0.74%

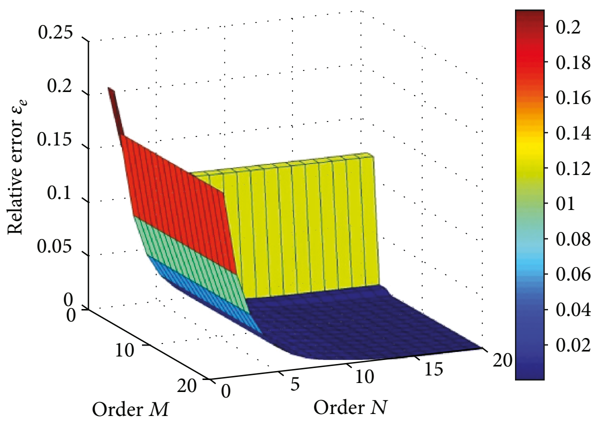

Figure 6: Error of the proposed Farrow filter with the order MN.

Table 2: Comparison of computational complexity.

	Clock (s)	ε_e (%)
Approach in [17]	4.605646	$1.132E-04$
New approach	2.331195	$1.128E-04$

structure filter outperforms the Farrow filter if the order is small. However, when the order of the filter is large, the Farrow structure outperforms the direct form filters. Note that the filter with the direct form structure has to be redesigned online when the time delay D changes. And hence it is not suitable for the condition where the DOA changes fast. On the other hand, the Farrow filter needs to be designed in an off-line manner only once and adjusts the input D to change the delay time as shown in Figure 3.

Figure 6 illustrates the relative error with different M and N. It shows that as the order of the Farrow filter increases, the relative error ε_e decreases rapidly. But once the order is large enough, the performance keeps constant.

4.3. Computational Complexity. The WLS algorithm to design the Farrow filter in [17] does not consider symmetric coefficient property. Under the same conditions, we compare the computational complexity between the proposed methodology and the one in [17]. The order of the Farrow delay filter is set with $N = 34$ and $M = 7$. The running time of the MATLAB program and the error are listed in Table 2, where the CPU frequency is 2.4 GHz. We can see that the new approach could reduce the computational complexity about 50% with the similar relative error as compared to the reference methodology.

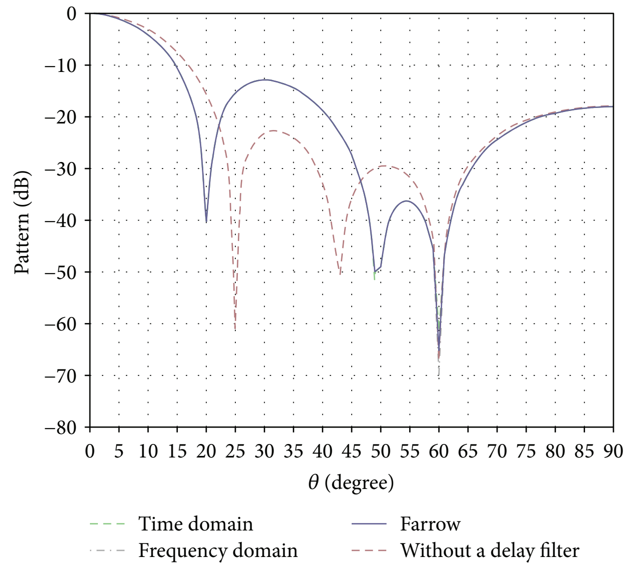

Figure 7: Patterns for broadband beamforming.

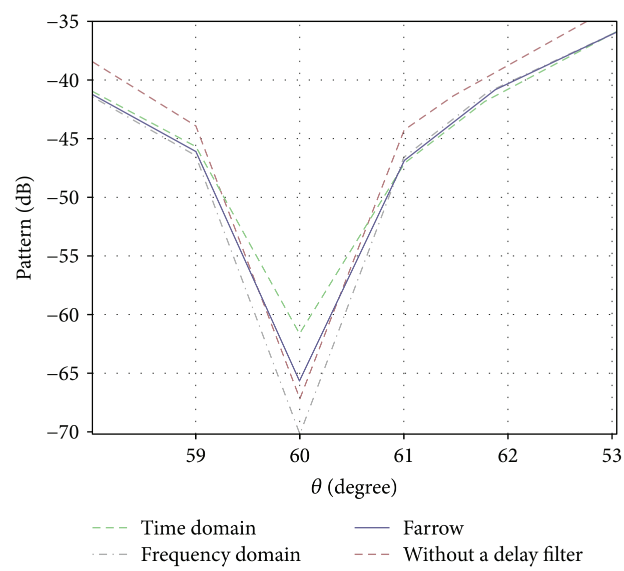

Figure 8: The nulls in the pattern for interference.

4.4. Beamforming Based on the Farrow Filter. To investigate the function of the Farrow filter in a space-time two-dimensional filter, we do the simulation about the array pattern. Suppose that the desired signal induces on the array with $\theta_0 = 25^\circ$ and the DOA of one strong interference is $\theta_1 = 60^\circ$. The radio frequencies of the two signals are both 2 GHz and the bandwidths are both 400 MHz.

Without a Farrow filter, the pattern is shown in Figure 7 by a dashed red line, where there are two nulls at $\theta_0 = 25^\circ$ and $\theta_1 = 60^\circ$, respectively. That means that the desired signal and interference are both suppressed. The other three lines in Figure 7 are the pattern when the delay filter is used through the three methodologies such as time domain [6, 8], frequency domain [7], and Farrow. They are very similar because the delay filters play the same functions. The interference is mitigated since the null still appears at $\theta = 60^{\circ}$. The main lobe points to $\theta = 0^{\circ}$ where the desired signal is impinging due to the role of calibration by the delay filter. Clearly, the result means that the delay filter could keep the desired signal to satisfy the requirement of the Frost algorithm. And it also proves that our proposed methodology for Farrow filter design is right.

If we zoom in the null in Figure 8, it can be seen that the depth is different. This is because the error of the delay filter is different. With the same filter order 56, the result is consistent with the result shown in Table 1 and that less error means deeper null.

5. Conclusion

The Farrow delay filter is taken in the broadband adaptive array instead of the delay filter with a direct form structure. A new methodology is presented to design the Farrow filter according to the symmetric coefficient. The computational cost is lower than that of the classic WLS algorithm. Although the Farrow filter is suited to the FPGA platform, the coefficient computation is heavily complicated. How to choose the order of the Farrow filter and error requirement is an important work in the future.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no financial and personal relationships with other people or organizations that can inappropriately influence their work; there are no professional or other personal interests of any nature or kind in any product, service, and company that could be construed as influencing the position presented in the paper.

Acknowledgments

The authors would like to acknowledge the support of innovation funds from China Academy of Space Technology (Project no. CAST2016021) and funds from Chongqing University (Project no. 2018CDGFTX0014).

References

[1] W. F. Gabriel, "Adaptive processing array systems," *Proceedings of the IEEE*, vol. 80, no. 1, pp. 152–162, 1992.

[2] W. Liu and S. Weiss, *Wideband Beamforming - Concepts and Techniques*, John Wiley & Sons, Ltd, 2010.

[3] M. S. Hossain, G. N. Milford, M. C. Reed, and L. C. Godara, "Efficient robust broadband antenna array processor in the presence of look direction errors," *IEEE Transactions on Antennas and Propagation*, vol. 61, no. 2, pp. 718–727, 2013.

[4] W. Liu, R. Wu, and R. J. Langley, "Design and analysis of broadband beamspace adaptive arrays," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 12, pp. 3413–3420, 2007.

[5] T. K. Sarkar, H. Wang, S. Park et al., "A deterministic least-squares approach to space–time adaptive processing (STAP)," *IEEE Transactions on Antennas and Propagation*, vol. 49, no. 1, pp. 91–103, 2001.

[6] T. I. Laakso, V. Valimaki, M. Karjalainen, and U. K. Laine, "Splitting the unit delay [FIR/all pass filters design]," *IEEE Signal Processing Magazine*, vol. 13, no. 1, pp. 30–60, 1996.

[7] G. Oetken, T. Parks, and H. Schussler, "New results in the design of digital interpolators," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 23, no. 3, pp. 301–309, 1975.

[8] Z. Ahmad, S. Yaoliang, and Q. du, "Adaptive wideband beamforming based on digital delay filter," *Journal of Microwaves, Optoelectronics and Electromagnetic Applications*, vol. 15, no. 3, pp. 261–274, 2016.

[9] H. Li, G. Torfs, T. Kazaz, J. Bauwelinck, and P. Demeester, "Farrow structured variable fractional delay Lagrange filters with improved midpoint response," in *2017 40th International Conference on Telecommunications and Signal Processing (TSP)*, pp. 506–509, Barcelona, Spain, 2017.

[10] C. W. Farrow, "A continuously variable digital delay element," in *1988 IEEE International Symposium on Circuits and Systems*, vol. 3, pp. 2641–2645, Espoo, Finla, 1988.

[11] H. Johansson, "Farrow-structure-based reconfigurable bandpass linear-phase FIR filters for integer sampling rate conversion," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 58, no. 1, pp. 46–50, 2011.

[12] M. Abbas, O. Gustafsson, and H. Johansson, "On the fixed-point implementation of fractional-delay filters based on the farrow structure," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 4, pp. 926–937, 2013.

[13] J. Vesma and T. Saramaki, "Design and properties of polynomial-based fractional delay filters," in *2000 IEEE International Symposium on Circuits and Systems*, vol. 1, pp. 104–107, Geneva, Switzerland, 2000.

[14] T. Ma, Y. Wei, and X. J. Ma, "A new method for designing Farrow filters based on cosine basis neural network," in *2017 IEEE International Conference on Digital Signal Processing*, pp. 154–158, Beijing, China, 2017.

[15] A. Tarczynski, G. D. Cain, E. Hermanowicz, and M. Rojewski, "WLS design of variable frequency response FIR filters," in *1997 IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 2244–2247, Hong Kong, 1997.

[16] W.-S. Lu and T.-B. Deng, "An improved weighted least-squares design for variable fractional delay FIR filters," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46, no. 8, pp. 1035–1040, 1999.

[17] T. B. Deng, "Symmetry-based low-complexity variable fractional-delay FIR filters," in *2004 IEEE International Symposium on Communications and Information Technology*, vol. 1, pp. 194–199, Sapporo, Japan, 2004.