ECE 4870 - Combining FIR Filters with Adaptive Antenna Arrays for High SIR Receivers

Jeffrey Wilcox[1]
Department of Electrical and Computer Engineering, Cornell University.

(Dated: 5 May 2024)

Adaptive antenna arrays are used to in a variety of applications, with the goal of focusing a beam to a specific angle and sometimes blocking an interferer at another angle. Some systems use this technology to callibrate to a desired source, others use continuous algorithms to track a target. This paper aims to develop an understanding of Adaptive Antenna Arrays from Uniform Linear Arrays (ULAs) and understand how the signal processing of an antenna system can change the effective array pattern.

I. INTRODUCTION & MOTIVATIONS

Radio frequency tracking systems have been continuously developed over the years for military applications, network improvements, and radio astronomy. General antenna arrays are good for single main lobes in a fixed position, but with time-evolving systems, methods are needed to calibrate antenna array systems, either at the start of each use or continuously throughout operation.

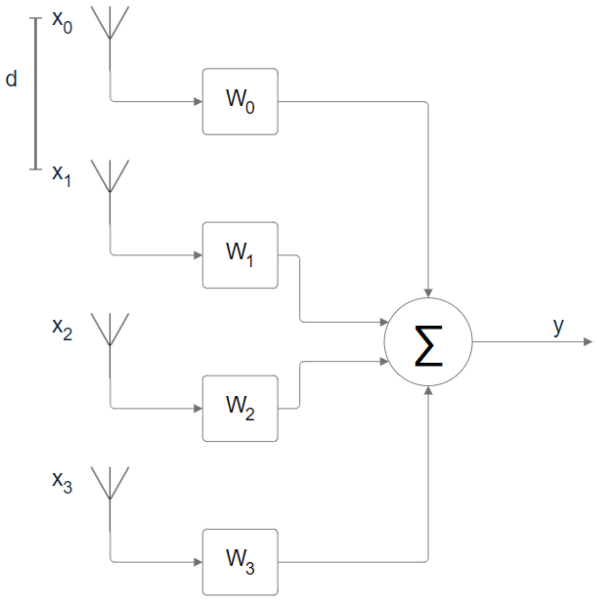

FIG. 1. Uniform Linear Array

II. UNDERSTANDING HOW A RECEIVER EFFECTS THE BEAM SHAPE OF A ULA

To begin, a uniform linear array is examined with more attention to the receiver. Figure 1 shows a ULA connected to its receiver processing network. Each antenna element is assumed to be the same and a half wavelength element separation distance is used. In a ULA, the array pattern is dictated by two things, the element separation (which has already been taken as $\lambda/2$) and the element phases. Although it may be feasible for some systems with large machinery or well controlled microelectromechanical devices, once the elements are placed it is typically difficult to change the separation distance. Due to this, the separation distance will be fixed at half wavelength as this allows narrow beam width with a little under 180 degree range for beam configuration.

In a general ULA, the phase between each consecutive element is the same. Because only the phases are being modified, we can take the magnitudes as unity and assume any scaling of the beam is done before the weighting to reduce the effect of noise, after the weighting for level control to some output, or both. With this assumption, each weight can be expressed as $w_n = e^{j\alpha_n}$ where n is the number of separation distances between the given antenna element and the reference antenna element which is denoted with $n = 0$. For most cases, $\alpha_n = n\alpha_0$ where α_0 is the separation between two consecutive elements.

To determine the array pattern, the weighted sum of the element signals is first found. For the four element example in Figure 1 and the uniform phase difference between elements, this becomes:

$$y = \sum_{n=0}^{3} e^{-j\omega_0 t} e^{jnkd\cos\theta} e^{jn\alpha_0} \tag{1}$$

Assuming the frequency is at a known, central frequency, we can remove the frequency dependence and achieve the array pattern:

$$F(\theta) = \sum_{n=0}^{3} e^{jn(kd\cos\theta+\alpha_0)} \tag{2}$$

To achieve a maximum at some angle, θ_0, array factor should be zero at that angle such that all elements add constructively. To achieve this, the phasing should be $\alpha_0 = -kd\cos\theta$. But what if the array application requires a null in a specific area? Or the target moves and a new angle is required? To meet these requirements, weighting can be modified either as calibration or continuously depending on the application. To meet these requirements, weighting optimization is required.

III. ANALOGY OF FIR FILTER TO BEAMFORMING

In practicality, beamforming is a spatial FIR filter[1]. To understand this, first let us examine what an FIR filter is. As shown in Figure 2, the FIR filter has four main components, an input source (the antenna in this case), delay elements (represented with z^{-1}), complex weights, and a summing junction. The delay blocks sample and hold the given input for one clock period, so the value at the input moves down the chain of delay blocks at a rate of one block per clock cycle. For those familiar with the digital domain, this is the equivalent of an analog flip flop. Although this domain description helps understand how these blocks function, they are actually frequency domain blocks so the delay is represented by multiplying by $\frac{1}{z}$ instead of time shifting the value (this is as deep as we will go into z-transforms, so have no fear if you are unfamiliar with them). Additionally, the weighting is carried out in the frequency domain, as is the summing.

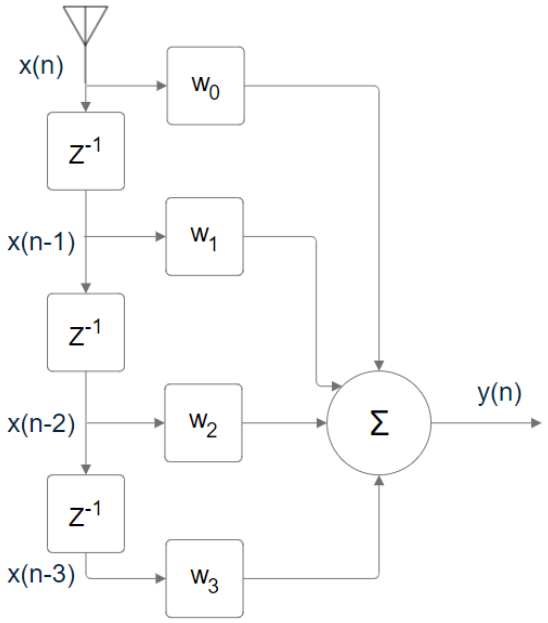

FIG. 2. FIR Filter

The reason this is named an FIR filter is because an FIR system has a finite duration impulse response and it is a frequency filter. If a delta signal was input to this system, it would have a non-zero response for as many clock cycles as there are in the FIR, which would be 4 for the given system in Figure 2. If the impulse response in Equation 3 is convolved with the input signal in the time domain, the output can be interpreted, however it is easier to Fourier transform the impulse response and simply multiply it by the input to understand the similarities between the FIR filter and the adaptive array. It should be noted that t denotes a discrete time input (i.e. $t \in \mathbb{Z}$).

$$h(t) = \sum_{n=0}^{3} w_n \delta(t-n) \tag{3}$$

$$H(\omega) = \sum_{n=0}^{3} w_n e^{j\omega n} \tag{4}$$

If we take w_k to have magnitude of 1 and equal phasing separation between delay blocks, we find that the FIR filter is analagous to an antenna array with ω in the FIR filter corresponding to $kd\cos\theta$ in the antenna array. Modifying the weightings in either, both in magnitude and non-uniform phases separation, gives rise to filters with specific response shapes. For these reasons, it can be helpful to think of the antenna array or beamformer as an angular filter which can be tuned by adjusting the weighting.

$$H(\omega) = \sum_{n=0}^{3} e^{jn(\omega+\alpha_0)} \tag{5}$$

$$F(\theta) = \sum_{n=0}^{3} e^{jn(kd\cos\theta+\alpha_0)} \tag{6}$$

IV. BEAMFORMING WITH OPTIMIZATION

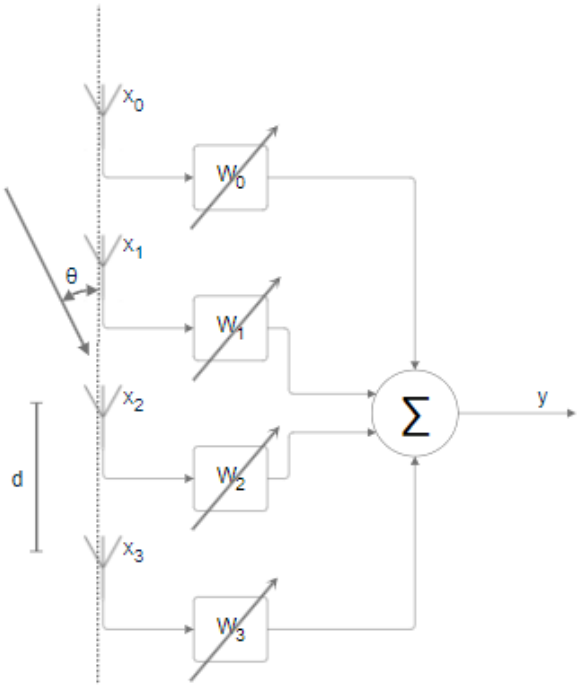

FIG. 3. Beamformer with Generalized, Tunable Weights

We now look at a new depiction of the system receiver which includes four elements as before but now with tunable weighting. By removing the restrictions of uniform phasing and amplitude, the weighting can be further utilized to implement more specific pattern details, such as multiple maximums or minimums. With these new freedoms on the weights, we can rewrite Equation 1 with more generalized weights.

$$y = \sum_{n=0}^{3} w_n x_n = \vec{w}^H \vec{x} \tag{7}$$

where

$$\vec{w} = \begin{bmatrix} w_0^* \\ w_1^* \\ w_2^* \\ w_3^* \end{bmatrix} \text{ and } \vec{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Each weight, w_n, is complex, w_n^* is used to denote the complex conjugate of w_n and $\vec{w}^H$ is defined as the Hermitian of $\vec{w}$ which is equivalent to the complex conjugate transpose of $\vec{w}$.

If it is known that a signal is being received from a certain angle θ_0, weight optimization can be used to block or enhance the signal. We will consider these separately and then combine them to develop a process for beamforming.

A. Signal of Interest (SOI) Case

In the signal of interest case, the signal is desired to pass through so the radiation pattern should have a main lobe in the direction of arrival (DOA) of the target. To make this happen, we assume that the incoming signal at each element has an amplitude of 1 and the desired signal output also has a magnitude of 1. We continue to assume that all incoming signals are at the receivers operating frequency ω_0, which could be achieved by bandpass filtering.

B. Signal Not of Interest (SNOI) Case

For interferers or signals not of interest, the signal is not desired and will likely overwhelm the receiver or make the desired signal harder to interpret. For these reasons, we desire a null at the DOA of the interferers to prevent them from passing through the receiver. We will start by analyzing these signals with an amplitude of 1, but in practicality interferer signals will have much larger amplitudes than the desired signal.

C. Combining to Make an Optimization Model

For any specific DOA θ_0, we can express the output of the beamformer as a function of that angle θ_0:

$$y(\theta_0) = \sum_{n=0}^{3} w_n x_n(\theta_0) = \vec{w}^H \vec{x}(\theta_0) \tag{8}$$

Notice that $\vec{w}$ is not a function of this angle, as the same weights will be applied for every signal that comes in regardless of input. The goal of the optimization problem is to develop a set of weights that optimizes the output of the beamformer to match the desired output. We call this desired output $d(\theta_0)$ and can derive an error function for the beam former as:

$$\varepsilon(\theta_0) = d(\theta_0) - y(\theta_0) \tag{9}$$

which can also be written as

$$\varepsilon(\theta_0) = d(\theta_0) - \vec{w}^H \vec{x}(\theta_0) \tag{10}$$

If there were a single element, there would be no effect on weighting its output, as there is no constructive or destructive interference to tune. However, if there are multiple elements, the antenna weighting can be tuned such that the outputs of each element do have constructive or destructive interference at specific angles. This is the general idea of using antenna arrays, but with optimization we are able to specify a desired output and receive a low error weighting configuration.

One popular model for many optimization algorithms within adaptive antennas is the MMSE or Minimized Mean Square Error criterion[2]. This takes the error given above and develops a cost function to optimize with respect to the weights. The following shows the derivation of weights using the MMSE criterion:

$$\begin{gathered} J(\varepsilon) = E[\varepsilon^2] \\ J(\varepsilon) = E[(d - \vec{w}^H \vec{x})^2] \\ J(\varepsilon) = E[|d|^2 - d\vec{x}^H \vec{w} - \vec{w}^H \vec{x} d^H + \vec{w}^H \vec{x}\vec{x}^H \vec{w}] \\ \nabla_{\vec{w}} J = 0 = E[-\vec{x}d^H - \vec{x}d^H + \vec{x}\vec{x}^H \vec{w}_{opt} + \vec{x}\vec{x}^H \vec{w}_{opt}] \\ 2E[\vec{x}\vec{x}^H]\vec{w}_{opt} = 2E[\vec{x}d^H] \\ \overline{\overline{R_{xx}}}\vec{w_{opt}} = \vec{r}_{xd} \\ \vec{w}_{opt} = \overline{\overline{R_{xx}}}^{-1} \vec{r}_{xd} \end{gathered}$$

One thing to note is how the expectation operator affects the analysis. In practice, this is important to keep through the computation as the input vector $\vec{x}$ is always changing and there is noise, but the expectation of the input is what we are tuning the system to. This can be found through processing in the receiver. For purposes of understanding optimization of the system, we assume $E[\vec{x}] = \vec{x}$ in later sections. Because the weights and desired signal are set by the user, the expectation will have no effect on them as they are equal to their expectation. It should be noted that $\overline{\overline{R_{xx}}}$ represents the covariance of the input signals and $\vec{r}_{xd}$ represents the cross correlation.

Expanding this to more dimensions introduces an optimization problem focused around satisfying the desired signal output over a number of angles. We now consider a vector of error functions to minimize, $\vec{\varepsilon}$, given by the following:

$$\vec{\varepsilon} = \vec{d} - \vec{w}^H \overline{\overline{X}} \tag{11}$$

where

$$\vec{\varepsilon} = \begin{bmatrix} \varepsilon(\theta_1) & \varepsilon(\theta_2) & \cdots & \varepsilon(\theta_k) \end{bmatrix} \tag{12}$$

$$\vec{d} = \begin{bmatrix} d(\theta_1) & d(\theta_2) & \cdots & d(\theta_k) \end{bmatrix} \tag{13}$$

$$\overline{\overline{X}} = \begin{bmatrix} x_0(\theta_1) & x_0(\theta_2) & \cdots & x_0(\theta_k) \\ x_1(\theta_1) & x_1(\theta_2) & \cdots & x_1(\theta_k) \\ \vdots & \vdots & \ddots & \vdots \\ x_n(\theta_1) & x_n(\theta_2) & \cdots & x_n(\theta_k) \end{bmatrix} \tag{14}$$

$$\vec{w} = \begin{bmatrix} w_0^* \\ w_1^* \\ \vdots \\ w_n^* \end{bmatrix} \tag{15}$$

Reconfiguring the result of the MMSE criterion with these higher dimensional vectors produces a method for determining a weighting configuration for a number of desired angle responses.

$$\vec{w}_{opt} = \left(\overline{\overline{X}}\,\overline{\overline{X}}^H\right)^{-1} \overline{\overline{X}}\vec{d}^H \tag{16}$$

D. Implementation of MMSE in Python

The above method for optimizing the antenna array for a specified desired signal for different directions gives the mathematical footprint for an algorithmic implementation of beamformer development. I developed two algorithms for achieving optimal null and main lobe patterns, one focused on placing main lobes in specific places and minimizing the gain of the array pattern everywhere else, the other focused on placing nulls in specific areas and providing equal gain everywhere else. The code for the SOI focused algorithm is shown below:

```
import numpy as np
import scipy as sp
import matplotlib.pyplot as plt

elements = 90 # Number of elements in array
angles = 180 # number of angles being taken into account
angle = np.linspace(0,np.pi,angles)
d = np.zeros(angles, dtype='complex_')
SOI = np.array([np.pi/4, np.pi/2, np.pi*0.57, 3*np.pi/4])
for i in SOI:
    for n,m in enumerate(angle):
        if(np.abs(m-i) <= 0.6*np.pi/angles):
            d[n] = 1

X = np.empty([elements,angles], dtype='complex_')
j = (-1)**0.5
for n in range(elements):
    for k in range(angles):
        # print(k)
        X[n][k] = np.exp(j*n*np.pi*np.cos(angle[k]))

X = np.matrix(X)
D = np.matrix(d)
A = X*np.matrix.getH(X)
B = X*np.matrix.getH(D)
w = np.array(np.linalg.solve(A,B))
w = np.matrix(w)
calcD = np.array(np.matrix.getH(w)*X)

def array_pattern_spec(theta):
    X = np.empty([w.size], dtype='complex_')
    j = (-1)**0.5
    for n in range(elements):
        X[n] = np.exp(j*n*np.pi*np.cos(theta))
    X = np.matrix(X)
    X = np.transpose(X)
    return np.matrix.getH(w)*X

theta1 = np.linspace(0,np.pi,360)
array_pattern = np.vectorize(array_pattern_spec, otypes={'complex_'})
F = array_pattern(theta1)
plt.plot(theta1,F)

Flog = 10*np.log10(np.abs(F)**2)
plt.plot(theta1,Flog)
```

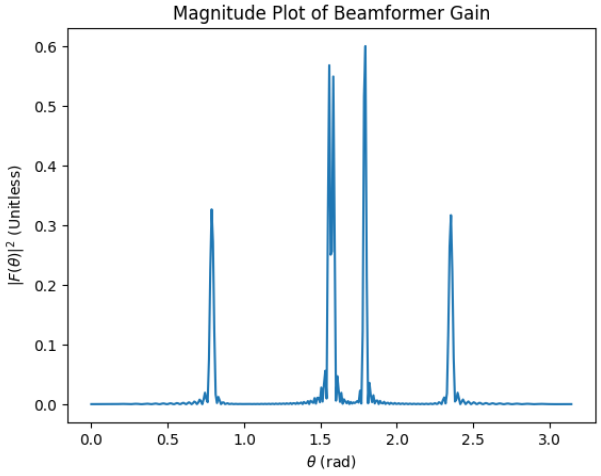

FIG. 4. Magnitude Response of the 90 Element Beamformer

This instantiation of the algorithm uses 180 degrees of input angles with the desired output set to one if it is in the list of SOI angles and zero otherwise. For all of the following cases, $\theta = [\pi/4, \pi/2, 0.57\pi, 3\pi/4]$ is used as the DOA angles for SOI or SNOI, depending on whether the algorithm is SOI or SNOI based. The computation of Equation 16 is then carried out and the array response for the achieved weighting is plotted. If more than 180 elements were used, the array pattern would be fully defined with no error so the least squares optimization would not be effective. The displayed code used 90 elements for good resolution without exceeding the number of angles considered as SOI or SNOI. Figure 4 and Figure

5 show the array response for the displayed code in magnitude and decibels, respectively.

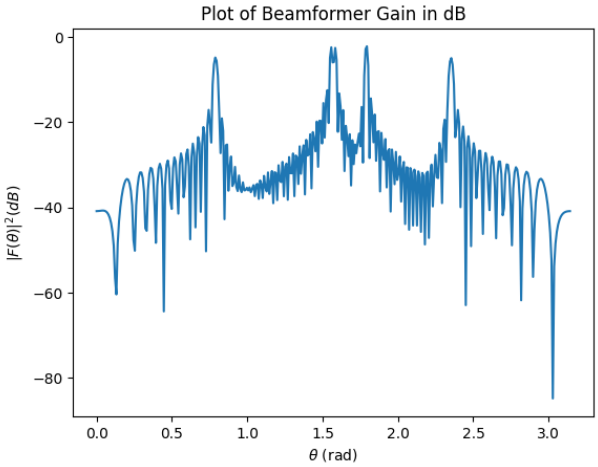

FIG. 5. Response of the 90 Element Beamformer in dB

Using a lower number of elements, the sidelobe levels and the half power bandwidth both increase, while gain decreases, as is the case in general antenna arrays. To demonstrate this, I ran the same algorithm using 10 elements instead of 90. The resulting array pattern can be seen in Figure 6 (Magnitude) and Figure 7 (dB).

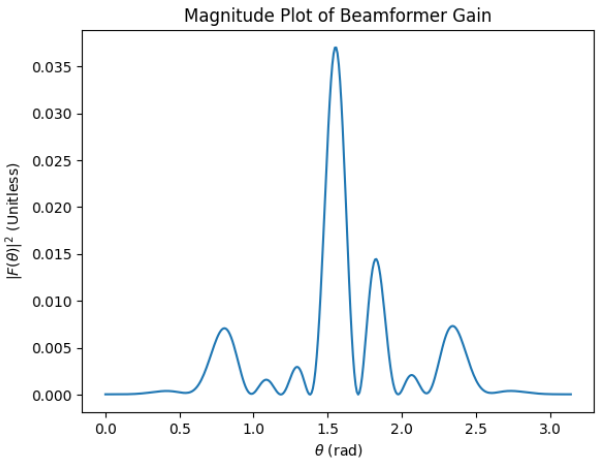

FIG. 6. Magnitude Response of the 10 Element Beamformer

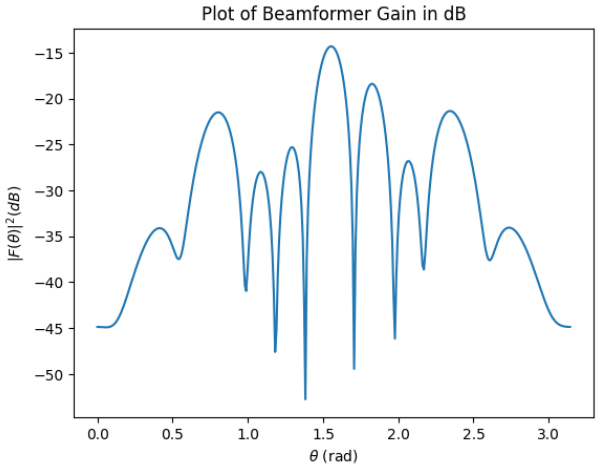

FIG. 7. Response of the 10 Element Beamformer in dB

The other algorithm considers every angle as containing an SOI unless otherwise specified by a list of SNOIs. The resulting array pattern with 90 elements in the SNOI algorithm can be seen in Figure 8 (Magnitude) and Figure 9 (dB). It should be noted that there is decent rejection ($\sim$ 10 dB) but variable gain across the pass angles (around $\pm$2 dB). For this reason, it is more favorable to use the SOI algorithm in most cases as a better SNOI focused algorithm can be developed from taking a unity gain input (as close to isotropic radiator as possible) and performing incoherent processing on both the unity gain and the beamformer gain. The output of the beamformer can be subtracted from the isotropic radiator output to get a relatively flat pass band with certain angle rejections across the angular domain. This can be seen in Figure 10. One visible downside is the higher gain values in the rejection range, with rejection not even breaking 4 dB, whereas the direct SNOI algorithm has attenuation of up to almost 10 dB.

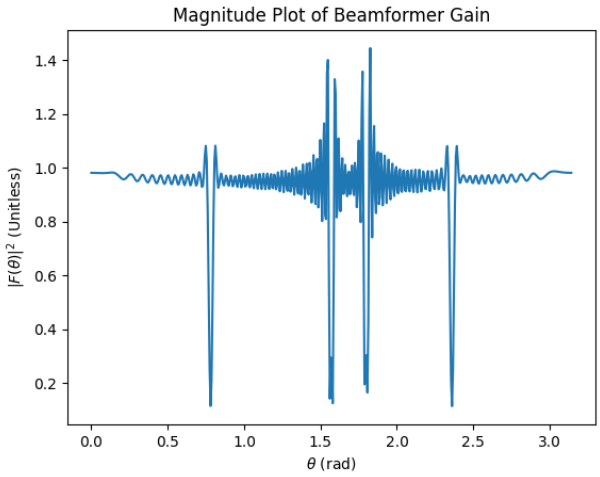

FIG. 8. Magnitude Response of the 90 Element SNOI Beamformer

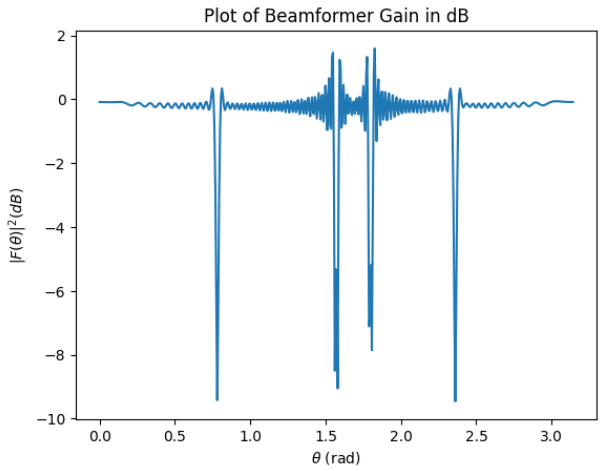

FIG. 9. Response of the 90 Element SNOI Beamformer in dB

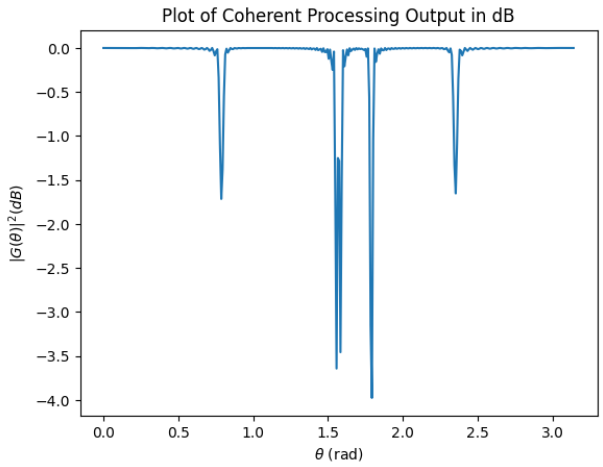

FIG. 10. Result of Coherent Processing on 90 Element Beamformer with an SOI configuration subtracted from Coherent Processing on Isotropic Receiver

V. ALGORITHMIC CONSIDERATIONS

There are many types of adaptive array algorithms that have been developed since the 1970s. The MMSE algorithm above works well for calibration, but due to the demanding matrix computations, this is less feasible in most applications, especially low-power systems. With the development of machine learning and artificial intelligence hardware, there has been significant investment into faster, more efficient matrix capable devices which could make the MMSE algorithm more enticing for some applications. Another algorithm for adaptive arrays is the LMS algorithm or Least Mean Squares, which aims to minimize the same cost function as the MMSE algorithm, however it uses a gradient descent algorithm to achieve the optimization without having to compute any inverse matrices. This is enticing for its ability to continuously recompute the optimal weight vector. The downside of this algorithm is cycle times, because it is iterative it can take some time to reach an optimal point depending how long it takes to recompute the weight vector and how commanding the cost function gradient is. To see this, the LMS algorithm can be examined. Starting with Equation 17, the weights are determined by taking the gradient of the cost function at one time value m and subtracting that from the weight vectors with a significance constant μ. The smaller μ is, the more accurate the algorithm is, but it will also increase the convergence time. There is also an upper bound on μ determined by the covariance matrix $\overline{\overline{X}}$ for convergence, given by $0 < \mu < (\text{Trace}[\overline{\overline{X}}])^{-1}$. The gradient of the cost function was derived above for the MMSE algorithm, but has been condensed into a clearer form in Equation 18.

$$\vec{w}(m+1) = \vec{w}(m) - \mu \nabla_{\vec{w}} J(m) \tag{17}$$

$$\nabla_{\vec{w}} J(m) = -2\vec{x}(m)\left(d^H(m) - \vec{x}^H(m)\vec{w}(m)\right) \tag{18}$$

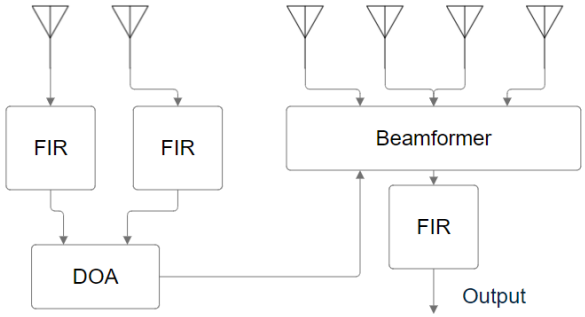

FIG. 11. Hybrid Array Using FIR Filtering and Beamforming

VI. COMBINING FIR FILTERS WITH BEAMFORMERS FOR TARGET TRACKING

Due to the ability of the beam pattern to change while operating, there are many applications of the adaptive array. However, the adaptive array requires a known direction of signal (or interferer) to determine optimal weights. Using an FIR filter is helpful for selecting frequencies, but does not tell the user much about the angular input as that is antenna dependent. But what if these could be combined to block signals at different frequencies and angles? First, an antenna array would have each element tied to an identical FIR filter, presenting the same output delay from the antenna and frequency filtering characteristics. This would be tuned based on a known desired frequency. Next, the output of the FIR filters would be sent through a DOA (Direction of Arrival) estimator to determine where the desired signal is located. This estimate is sent into the adaptive antenna array where it is used within an algorithm like the MMSE or LMS to target the desired signal. This would provide rejection to other angles and increase

the SIR (Signal to Interference Ratio). Lastly, the output of the beamformer would go through another identical FIR filter to reduce the power of any signals of different frequencies that came from the same direction as the desired signal. This dual filtering scheme allows the receiver to track the desired target from knowing only the operating frequency.

VII. DISCUSSION AND CONCLUSIONS

A. Discussion

Although there has been a focus on the theoretical implementations, it is important to consider what is practical with modern hardware. The FIR filter would likely be constructed using switched capacitor circuits in the analog domain[3,4]. The beamforming could be done in a few ways, as there is infrastructure for both analog and digital beamforming[5]. If the beamforming is done digitally, an analog to digital converter (ADC) will be needed for beamforming and a digital to analog converter (DAC) will be needed for input to the FIR filter.

The phase differences will be due to time delay in receiving between the different elements. This is also dependent on the carrier frequency, so this beamforming scheme will work best for amplitude modulated coding and less well for frequency modulation due to the phase dependence on frequency.

B. Conclusion

The increase in machine learning research and hardware has made matrix computation more efficient and feasible, creating a space for more matrix computations to be carried out with speed and efficiency. This could allow the MMSE algorithm to be used within systems such as the proposed FIR hybrid filter. The use of optimization theory in both the frequency (FIR) and angular (beamformer) domains is desirable for easily adaptable systems. There are many applications for adaptive antenna arrays, from networking communications for increased throughput to target tracking across a large range of angles.

[1] B. Van Veen and K. Buckley, "Beamforming: a versatile approach to spatial filtering," IEEE ASSP Magazine **5**, 4–24 (1988).
[2] C. A. Balanis, *Antenna Theory: Analysis and Design* (Wiley, 2016).
[3] B. Razavi, *Design of Analog CMOS Integrated Circuits* (McGraw-Hill Education, 2017).
[4] P. Pawłowski, R. Długosz, and A. Dąbrowski, "Switched-capacitor finite impulse response rotator filter," in *2020 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)* (2020) pp. 133–137.
[5] S. H. Talisa, K. W. O'Haver, T. M. Comberiate, M. D. Sharp, and O. F. Somerlock, "Benefits of digital phased array radars," Proceedings of the IEEE **104**, 530–543 (2016).