A Neural Network Approach to MVDR Beamforming Problem

Po-Rong Chang, *Member, IEEE,* Wen-Hao Yang, and Kuan-Kin Chan, *Member, IEEE*

Abstract—**A Hopfield-type neural network approach which leads to an analog circuit for implementing the real-time adaptive antenna array is presented. An optimal pattern of the array can be steered by updating the weights across the array in order to maximize the output signal-to-noise ratio (SNR). Furthermore, it is shown that the problem of adjusting the array weights can be characterized as a constrained quadratic nonlinear programming. Practically, the adjustment of settings is required to respond to a rapid time-varying environment. Many numerical algorithms have been developed for solving such problems using digital computers. The main disadvantage of these algorithms is that they generally converge slowly. To tackle this difficulty, a neural analog circuit solution is particularly attractive in real-time applications with minimization of a cost function subject to constraints. A novel Hopfield-type neural net with a number of graded-response neurons designed to perform the constrained quadratic nonlinear programming would lead to such a solution in a time determined by RC time constants, not by algorithmic time complexity. The constrained quadratic programming neural net has associated it with an energy function which the net always seeks to minimize. A fourth-order Runge–Kutta simulation is conducted to verify the performance of the proposed analog circuit. It shows that the circuit operates at a much higher speed than conventional techniques and the computation time of solving a linear array of 10 elements is about 0.1 ns for RC = 5×10^{-9}.**

I. Introduction

BEAMFORMING is one of the main functions of a passive phased-array processing system. It involves forming multiple beams through applying appropriate delay and weighting elements to the signals received by the sensors. The purpose is to suppress unwanted jamming interferences and to produce the optimal beamformer response which contains minimal contributions due to noise. The most commonly employed technique for deriving the adaptive weights uses a closed loop gradient descent algorithm where the weight updates are derived from estimates of the correlation between the signal in each channel and the summed output of the array. This process can be implemented in an analog fashion using correlation loops [1] or digitally in the form of the Widrow least mean square (LMS) algorithm [2]. The fundamental limitation for this technique is one of poor convergence for a broad dynamic range signal environment. Several different approaches for choosing optimum beamformer weights are summarized in [3]. In many applications none of those approaches is satisfactory. The desired signal may be of unknown strength and may always be present, resulting in signal cancellation with the multiple sidelobe canceller and preventing estimation of signal and noise covariance matrices in the maximum signal-to-noise (SNR) processor. These limitations can be overcome through the application of linear constraints to the weights. The basic concept of linearly constrained minimum variance (LCMV) beamforming is to constrain the response of the beamformer such that the desired signals are passed with specified gain and phase. The weights are chosen to minimize output power subject to the response constraint. When the beamformer has unity response in the look direction, the LCMV problem would become the minimum variance distortionless response (MVDR) beamformer problem, which is a very general approach employed to control beamformer response.

The weights of an MVDR-based beamformer should be updated in real time in order to respond to the rapid time-varying environment. Meanwhile, the evaluation of weights is computationally intensive and can hardly meet the real-time requirement. Systolic implementations of optimum beamformers have been studied to improve the computational speed by a number of investigators. McWhirter and Shepherd [5] showed how a triangular systolic array of the type proposed by Gentleman and Kung [6] can be applied to the problem of linearly constrained minimum variance problem, subject to one or more simultaneous linear equality constraints. Their fully pipelined implementation requires $O(p^2 + kp)$ arithmetic operations per cycle time where p is the number of antenna elements and k is the number of look direction constraints. As an alternative to the digital approach, an analog approach based on Hopfield-type neural networks could operate at much higher speed and requires less hardware than digital implementation.

Tank and Hopfield [7] have shown how a class of neural networks with symmetric connections between neurons presents a dynamics that leads to the optimization of a quadratic functional. Recently Chua and Lin [8] and Kennedy and Chua [9], [10] extended the design of Hopfield network and introduced a canonical nonlinear programming circuit which is able to handle more general optimization problems. They showed that a canonical neural network assigned to solve the optimization problem would reach a solution in a time determined by RC time constant, not by algorithmic time complexity. Therefore, the converge speed of reaching the optimal solution is dramatically improved. Experiments show that a MVDR-based neural analog circuit is quite robust and independent of signal power level and the computation time

Manuscript received June 10, 1991; revised November 7, 1991. This work was supported in part by the National Science Council, Republic of China, under Contract NSC81-0404-E009-580.

The authors are with the Department of Communication Engineering, National Chiao Tung University, Hsinchu, Taiwan, Republic of China.

IEEE Log Number 9106609.

0018-926X/92$03.00

of solving a linear array of 10 elements is about 0.1 ns for $RC = 5 \times 10^{-9}$.

II. Problem Formulation

Considering a linear array composed of L isotropic antenna elements which receive signals from sources of variation frequency f_0 located far from the array, $x_l(t)$ is defined as a complex output of the lth element at the sampling time t, and can be expressed as [4]

$$x_l(t) = m(t)e^{j2\pi f_0(t+\tau_l(\theta,\phi))} + n_l(t) + x_{Il}(t) \quad (1)$$

where

$$\tau_l(\theta,\phi) = \frac{\mathbf{r}_l \cdot \hat{\mathbf{s}}(\theta,\phi)}{c} \quad (2)$$

is the time delay of the lth element relative to a reference point chosen at origin. $\mathbf{r}_l$ is the position vector of the lth element. $\hat{\mathbf{s}}(\theta, \phi)$ is an unit vector in the direction (θ, ϕ) of the source, and c is the propagation speed of the plane wave in free space.

The source amplitude $m(t)$ is characterized statistically by

$$E[m(t)] = 0 \quad (3)$$

$$E[m(t)m^*(t)] = p_s \quad (4)$$

where $E[\cdot]$ is the expectation operator, p_s is the power of the source, and the asterisk denotes the complex conjugate. $x_{Il}(t)$ is the component of the directional interferences received by the lth element and possess the same statistics as the source. In addition, $n_l(t)$ is a white random noise with properties

$$E[n_l(t)] = 0, \qquad l = 1, 2, \cdots, L \quad (5)$$

$$E[n_l(t)n_k^*(t)] = \sigma_n^2\delta_{lk}, \qquad l, k = 1, 2, \cdots, L. \quad (6)$$

Let the signal waveforms derived from the L elements of a beamformer be represented by an L-dimensional complex vector

$$\mathbf{X} \overset{\text{def}}{=} [x_1, x_2, \cdots, x_L]^T \quad (7)$$

and the weights of element outputs be represented by L-dimensional complex vector $\mathbf{W}$,

$$\mathbf{W} \overset{\text{def}}{=} [w_1, w_2, \cdots, w_L]^T \quad (8)$$

where T denotes the transpose of the vector. Then the output of the beamformer can be written as

$$y(t) = \sum_{l=1}^{L} w_l^* x_l(t) = \mathbf{W}^H\mathbf{X}(t) \quad (9)$$

where H denotes the complex conjugate transpose of a vector.

Since each component of $\mathbf{X}(t)$ is modeled as a zero mean stationary process, the mean output power of the beamformer is given by

$$P(\mathbf{W}) = E[y(t)y^*(t)] = \mathbf{W}^H\mathbf{R}\mathbf{W} \quad (10)$$

where $\mathbf{R}$ is the array correlation matrix.

In order to achieve the optimal utilization of the mean output power of the beamformer, the weights are chosen based on the statistics of the data received at the array such that the output contains minimal influence due to noise as well as interference signals arriving from other directions. Different criteria exist for choosing optimum beamformer weights, which are summarized in [4]. A general approach is to constrain the response of the beamformer so that the desired signals are passed with specified gain and phase. The weights are chosen to minimize the output power subject to the required constraints. This has the effect of preserving the desired signal while minimizing contributions due to noise and interfering signals arriving from directions other than the direction of interest. Based on the above concept, determination of weights with linear constraints to the weight vector is called the linearly constrained minimum variance beamforming problem, which is usually formulated as

$$\min_{\mathbf{W}} \quad \phi(\mathbf{W}) = \mathbf{W}^H\mathbf{R}\mathbf{W} \quad (11)$$

$$\text{subject to} \quad \mathbf{W}^H\mathbf{S}_0 = r \quad (12)$$

where r is a complex constant, $\mathbf{S}_0$ is the steering vector associated with the look direction and is given by

$$\mathbf{S}_0 = \left[1, \exp\left(j\frac{2\pi d}{\lambda_0}\cos\theta_0\right), \cdots, \exp\left(j\frac{2\pi d}{\lambda_0}(L-1)\cos\theta_0\right)\right]^T \quad (13)$$

where d is the element spacing, λ_0 is the wavelength of the plane wave in free space, and θ_0 is the look direction angle (the angle between the axis of the linear array and the direction of the desired signal source).

The method of Lagrange multipliers can be used to solve (11) resulting in

$$\hat{\mathbf{W}} = r\frac{\mathbf{R}^{-1}\mathbf{S}_0}{\mathbf{S}_0^H\mathbf{R}^{-1}\mathbf{S}_0}. \quad (14)$$

Note that in practice the presence of uncorrelated noise ensures that $\mathbf{R}$ is invertible. If $r = 1$, then (11) is often termed the minimum variance distortionless response beamformer.

The MVDR beamforming problem defined in both (11) and (12) is indeed a complex-value constrained quadratic programming problem, which cannot be solved by neural network directly. In order to meet the requirement of neural-based optimizer, one should convert it into a real-value constrained quadratic programming formulation. To achieve this goal, the complex vectors $\mathbf{W}$, $\mathbf{W}_0$, and matrix $\mathbf{R}$ should first be decomposed into their real and imaginary constituents, or

$$\mathbf{W} = \mathbf{W}_r + j\mathbf{W}_i$$

$$\mathbf{R} = \mathbf{R}_r + j\mathbf{R}_i \quad (15)$$

and

$$\mathbf{S}_0 = \mathbf{S}_{0_r} + j\mathbf{S}_{0i} \quad (16)$$

where $\mathbf{W}_r$, $\mathbf{R}_r$, $\mathbf{S}_{0r}$ and $\mathbf{W}_i$, $\mathbf{R}_i$, $\mathbf{S}_{0i}$ are the real parts and imaginary parts of $\mathbf{W}$, $\mathbf{R}$, and $\mathbf{S}_0$, respectively.

Next, substituting (15) into (10), the mean output power becomes

$$\begin{aligned}\mathbf{W}^H\mathbf{R}\mathbf{W} &= (\mathbf{W}_r + j\mathbf{W}_i)^H(\mathbf{R}_r + j\mathbf{R}_i)(\mathbf{W}_r + j\mathbf{W}_i)\\ &= \mathbf{W}_r^T\mathbf{R}_r\mathbf{W}_r - \mathbf{W}_r^T\mathbf{R}_i\mathbf{W}_i + \mathbf{W}_i^T\mathbf{R}_i\mathbf{W}_r\\ &\quad + \mathbf{W}_i^T\mathbf{R}_r\mathbf{W}_i + j\left[\mathbf{W}_r^T\mathbf{R}_r\mathbf{W}_i + \mathbf{W}_r^T\mathbf{R}_i\mathbf{W}_r\right.\\ &\quad \left. - \mathbf{W}_i^T\mathbf{R}_r\mathbf{W}_r + \mathbf{W}_i^T\mathbf{R}_i\mathbf{W}_i\right]. \end{aligned} \tag{17}$$

Since $\mathbf{R}$ is a positive-definite Hermitian matrix, both $\mathbf{R}_r$ and $\mathbf{R}_i$ are identified as the symmetric and skew-symmetric matrices, respectively. Employing the above fact, it can be shown that the imaginary part of $\mathbf{W}^H\mathbf{R}\mathbf{W}$ vanishes and consequently

$$\begin{aligned}\mathbf{W}^H\mathbf{R}\mathbf{W} &= \mathbf{W}_r\mathbf{R}_r\mathbf{W}_r - \mathbf{W}_r^T\mathbf{R}_i\mathbf{W}_i + \mathbf{W}_i^T\mathbf{R}_i\mathbf{W}_r + \mathbf{W}_i^T\mathbf{R}_r\mathbf{W}_i\\ &= \mathbf{v}^T\begin{bmatrix}\mathbf{R}_r & -\mathbf{R}_i\\ \mathbf{R}_i & \mathbf{R}_r\end{bmatrix}\mathbf{v}\end{aligned} \tag{18}$$

where $\mathbf{v}$ is a $2L$-dimensional real weight vector defined by

$$\mathbf{v} = \begin{bmatrix}\mathbf{W}_r\\ \mathbf{W}_i\end{bmatrix}. \tag{19}$$

Similarly, the linear constraint can be written as

$$\begin{bmatrix}\mathbf{S}_{0_r}^T & \mathbf{S}_{0_i}^T\\ -\mathbf{S}_{0_i}^T & \mathbf{S}_{0_r}^T\end{bmatrix}\mathbf{v} = \begin{bmatrix}1\\ 0\end{bmatrix}. \tag{20}$$

For clarity, we let

$$\mathbf{G} = \begin{bmatrix}2\mathbf{R}_r & -2\mathbf{R}_i\\ 2\mathbf{R}_i & 2\mathbf{R}_r\end{bmatrix} \tag{21}$$

$$\mathbf{B} = \begin{bmatrix}\mathbf{S}_{0_r}^T & \mathbf{S}_{0_i}^T\\ -\mathbf{S}_{0_i}^T & \mathbf{S}_{o_r}^T\end{bmatrix} \tag{22}$$

and

$$\mathbf{e} = \begin{bmatrix}1\\ 0\end{bmatrix} \tag{23}$$

where $\mathbf{G}$, $\mathbf{B}$, and $\mathbf{e}$ are $(2L) \times (2L)$ symmetric, positive-definite matrix, $2 \times (2L)$ matrix, and (2×1) column vector, respectively. Therefore, the complex-value MVDR problem becomes the following equivalent real canonical quadratic programming problem with linear equality constraints

$$\min_{\mathbf{v}} \quad \phi(\mathbf{v}) = \tfrac{1}{2}\mathbf{v}^T\mathbf{G}\mathbf{v} \tag{24}$$

$$\text{subject to} \quad \mathbf{f}(\mathbf{v}) = \mathbf{B}\mathbf{v} - \mathbf{e} = 0 \tag{25}$$

where $\mathbf{f}(\mathbf{v})$ is a 2×1 column vector.

III. A Neural-Based Canonical Nonlinear Programming Circuit

To allow a beamformer to respond to a rapid time-varying environment, the weights should be adaptively controlled to satisfy (14) in real time. However, it is computationally intensive and is very costly to implement using discrete components. Digital systolic implementations of optimal beamformers have been studied by a number of investigators [5]. They are usually designed to both compute and implement the adaptive weights. As an alternative to the digital approach, an analog approach based on Hopfield-type neural networks could operate at a much higher speed and requires less hardware than digital implementation. It is shown that Hopfield-type neural networks can solve a number of difficult optimization problems [7]–[10] in a time determined by the system RC time constants, not by algorithmic time complexity. Based on this fact, the neural-based analog circuits are suggested to be one of the favorable choices for the real-time implementation for solving the MVDR problem.

Artificial neural networks contain a large number of identical computing elements or neurons with specific interconnection strengths between neuron pairs. The massively parallel processing power of neural network in solving difficult problems lies in the cooperation of highly interconnected computing elements. Tank and Hopfield [7] demostrated that their networks have the real-time capability in solving various optimization problems, especially, the linear programming and signal decomposition/decision problems, by the programming of synaptic weights stored as a conductance matrix. Recently, Chua and Lin [8] and Kennedy and Chua [10] have extended the method to deal with more general nonlinear programming problems. They proposed a canonical nonlinear programming circuit, which includes dc voltage and current sources, multiport transformers and a network of conductances. Chua and Lin [8] also showed that the linear programming network of Tank and Hopfield is, in fact, a special case of the canonical nonlinear programming circuit. Since the risk of instability in the network is ever presented, Kennedy and Chua [10] introduced their modified canonical design which can guarantee the stability of the network solution. In order to obtain a robust and stable solution of MVDR problem in real time, the circuit proposed by Kennedy and Chua is particularly considered in the design of our implementation in this paper.

The general nonlinear programming problem can be stated as the attempt to minimize a scalar cost function

$$\phi(v_1, v_2, \cdots, v_n). \tag{26}$$

This minimization is to be accomplished subject to a set of m inequality (or equality) constraints

$$f_j(v_1, v_2, \cdots, v_n) \geq 0(\text{or} = 0), \qquad 1 \leq j \leq m \tag{27}$$

where m and n are two independent integers.

The canonical nonlinear programming circuit shown in Fig. 1 consists of controlled current and voltage sources, nonlinear resistors, and linear capacitors. The voltages v_i across the capacitors on the right-hand side of Fig. 1 represent the values of the variables involved in the nonlinear programming. The currents i_j through the voltage-controlled nonlinear resistors $g_j(f_j(\mathbf{v}))$ represent constraint satisfaction,

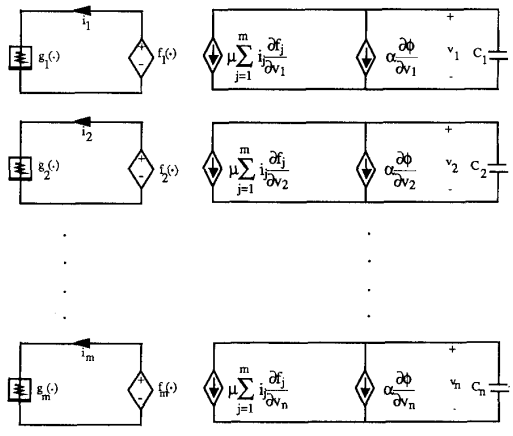


Fig. 1. Canonical nonlinear programming circuit.

where $\mathbf{v} = [v_1, v_2, \cdots, v_n]^T$. Note that the nonlinear functions $g_j(\cdot)$ are used to impose the constraints in the circuit realization. By reading directly from Fig. 1, the circuit equations for the network are given by

$$C\frac{dv_k}{dt} = -\alpha\frac{\partial \phi}{\partial v_k} - \mu\sum_{j=1}^{m} i_j\frac{\partial f_j}{\partial v_k}, \qquad k = 1, 2, \cdots, n \tag{28}$$

where $i_j = g_j(f_j(\mathbf{v}))$, and both α and μ are positive scaling factors.

Next, we would like to show that the circuit equation of (28) converges to a minimum of the cost function $\phi(\mathbf{v})$ subject to a set of constraints. Before discussing this critical issue, several considerations involved in the constrained problem should be identified. The constraints in (27) define a subspace of the multidimensional parameter space called the feasibility region. Solving a constrained problems is, hence, the process of finding that point inside the corresponding feasibility region (including the boundary) where the value of the cost function $\phi(\mathbf{v})$ is the minimum one. To solve a constrained problem defined in (26) and (27), we convert it in an equivalent unconstrained problem. The way to do this is to define a pseudo-cost function $E(\mathbf{v})$ as follows:

$$E(\mathbf{v}) = \alpha\phi(\mathbf{v}) + \mu P(\mathbf{v}) \tag{29}$$

where $\phi(\mathbf{v})$ is the original cost function, $P(\mathbf{v})$ is referred to as the penalty function, and α and μ are called the acceleration factor and the penalty multiplier, respectively.

Different penalty function alternatives can be used in practice. Owing to the considerations in [8], [9], it can be concluded that for a function to qualify as a valid penalty function it must monotonically increase as the $f_j(\mathbf{v})$ deviates from the satisfaction of those constraints. In particular, either the absolute value or the square operator fulfills this requisite, resulting in the following expressions for $P(\mathbf{v})$:

$$P(\mathbf{v}) = \begin{cases} \sum_{j=1}^{m} |\, g_j(f_j(\mathbf{v}))\,|, & \text{for absolute value} \\ \frac{1}{2}\sum_{j=1}^{m} g_j^2(f_j(\mathbf{v})), & \text{for square operator.} \end{cases} \tag{30}$$

Moreover, the output of the jth constraint amplifier $g_j(f_j(\mathbf{v}))$ can be defined as follows:

$$g_j(f_j(\mathbf{v})) = \begin{cases} f_j(\mathbf{v}), & \text{for equality constraint} \\ U(-f_j(\mathbf{v}))f_j(\mathbf{v}), & \text{for inequality constraint} \end{cases} \tag{31}$$

where $U(\cdot)$ is the unit step function.

It is interesting to note that the pseudo-cost function $E(\mathbf{v})$ can be identified as energy function for the system of circuit equations (28) and the system is ensured to be completely stable. By complete stability, the system will not oscillate, but will converge to a stable equilibrium state. For simplicitly of analysis, it is assumed the first-order time derivative of $E(\mathbf{v})$ exists and is continuous. In order to make the pseudo-cost function $E(\mathbf{v})$ be differentiable, the square operator would be particularly considered in the penalty function. By using the fact that $i_j = g_j(f_j(\mathbf{v}))$, the time derivative of E becomes

$$\begin{aligned} \frac{dE}{dt} &= \alpha\sum_{k=1}^{n}\frac{\partial \phi}{\partial v_k}\frac{dv_k}{dt} + \mu\sum_{j=1}^{m}\sum_{k=1}^{n} g_j(f_j)\frac{df_j}{\partial v_k}\frac{dv_k}{dt} \\ &= \sum_{k=1}^{n}\left[\alpha\frac{\partial \phi}{\partial v_k} + \mu\sum_{j=1}^{m} g_j(f_j)\frac{\partial f_j}{\partial v_k}\right]\frac{dv_k}{dt} \\ &= -\sum_{k=1}^{n} C_k\left(\frac{dv_k}{dt}\right)^2 \\ &\le 0. \end{aligned} \tag{32}$$

Since each C_k is strictly positive and $(dv_k/dt)^2 \ge 0$ for all k, the time derivative of the pseudo-cost function $E(\mathbf{v})$ is always less than zero. This implies that the circuit will force $E(\mathbf{v})$ to be monotonically decreased except at the equilibrium points where the time derivative vanishes. Nevertheless, the equilibrium points may be either local minimum or inflection points of $E(\mathbf{v})$. The second-order conditions, which are defined in terms of the Hessian matrix $\nabla^2 E(\mathbf{v})$ of second partial derivative of $E(\mathbf{v})$, must be derived in order to determine the status of those equilibrium points. It is shown in [12] that the equilibrium point is a local minimum if the Hessian matrix of $E(\mathbf{v})$ is positive-definite. Usually, the Hessian matrix $\nabla^2 E(\mathbf{v})$ is an $n \times n$ symmetric matrix defined by

$$\nabla^2 E(\mathbf{v}) = \left[\frac{\partial^2 E(\mathbf{v})}{\partial v_i v_j}\right] \tag{33}$$

where v_i is the ith component of $\mathbf{v}$.

By substituting (24) and (25) into (29), we have

$$E(\mathbf{v}) = \frac{\alpha}{2}\mathbf{v}^T\mathbf{G}\mathbf{v} + \mu(\mathbf{B}\mathbf{v} - \mathbf{e})^T(\mathbf{B}\mathbf{v} - \mathbf{e}). \quad (34)$$

As a consequence, its Hassian matrix becomes

$$\nabla^2 E(\mathbf{v}) = \alpha\mathbf{G} + 2\mu\mathbf{B}^T\mathbf{B}. \quad (35)$$

Since G and $\mathbf{B}^T\mathbf{B}$ are positive-definite and positive semidefinite, respectively, the Hassian matrix $\nabla^2 E(\mathbf{v})$ which is linear sum of $\mathbf{G}$ and $\mathbf{B}^T\mathbf{B}$ is positive definite. Therefore the equilibrium point is also the local minimum point. Furthermore, the feasible region over which the constraints of (25) are satisfied can be shown to be a convex set. Since the Hassian matrix of (35) is positive definite throughout the feasible region, it is shown in [12] that any local minimum of $E(\mathbf{v})$ is a global minimum over this feasible region. As a result, the circuit solution of the Hopfield-type network tends to a global minimum of the original cost function $\phi(\mathbf{v})$ within the region over which the constraints are satisfied, when $dE/dt = 0$.

IV. A Neural-Based Circuit Implementation for the MVDR Beamforming Problem

Basically, the circuit shown in Fig. 1 is used to solve a general nonlinear programming. In the case of MVDR-based constrained quadratic problem, a more compact neural-based circuit realization using the existing solid-state devices is possible. Usually, the circuit would include two particular modules. The first module is called the variable amplifier, which can perform the integral of a sum of $(m + 1)$ input currents $(-\alpha\partial\phi/\partial v_k)$ and $(-\mu i_j \partial f_j/\partial v_k)$ and then produces the desired output variable v_k. Fig. 2(a) shows the circuit implementation of the variable amplifier consisting of an integrator and a unity gain inverting amplifier. Op amp 1 produces a voltage v which is in proportion to the integral of the total input current I. The inverting amplifier including op amp 2 and resistor R reproduces this voltage, but with opposite sign. The second module is called the constraint amplifier which is used to perform the constraint satisfaction function $g_j(\cdot)$. Since the MVDR problem has two equality constraints, the output of each $g_j(\cdot)$ would be identical to its input. Therefore the circuit realization of $g_j(\cdot)$ is particularly simple and shown in Fig. 2(b). Without loss of generality, the penalty multiplier μ may be included in $g_j(\cdot)$. Thus, the circuit yields the input–output relation: $O = -\mu I$, where μ represents the magnitude of the resistance, and O and I are an output voltage and an input current, respectively. If the input current I is equal to $-f_j(\mathbf{v})$, then $O = -\mu(-f_j(\mathbf{v})) = \mu f_j(\mathbf{v}) = \mu g_j(f_j(\mathbf{v})) = \mu i_j$.

It should be noted that the canonical nonlinear programming circuit model uses both current i_j and voltage v_k as variables. However, both i_j and v_k are represented as "voltages" in the neural network implementation [10]. By employing the virtual short circuit property of the op amp [10], these voltages would be converted to currents which are suitable for performing the weighted sum operation. Looking upon the circuit system dynamics of (28), the controlled current sources $\partial\phi/\partial v_k$ and $\partial f_j/\partial v_k$ should be determined prior to the implementation. From (24) and (25), it follows that

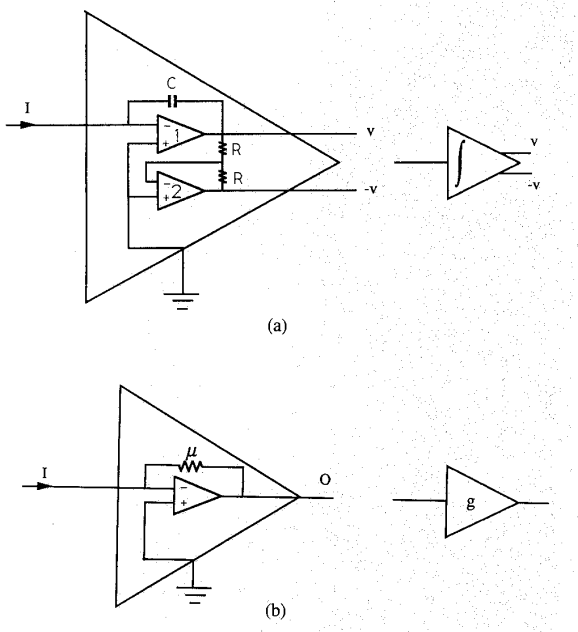


Fig. 2. Basic blocks of neural circuit. (a) Variable amplifier. (b) Constraint amplifier.

$$\frac{\partial\phi(\mathbf{v})}{\partial v_k} = \sum_{i=1}^{2L} g_{ki} v_i \quad (36)$$

and

$$\frac{\partial f_j(\mathbf{v})}{\partial v_k} = b_{jk} \quad (37)$$

where g_{ki} and b_{jk} are the (k, i) and (j, k) entries of $\mathbf{G}$ and $\mathbf{B}$, respectively.

Equation (36) shows that the input current $\partial\phi/\partial v_k$ is a linear sum if the v_k weighted by conductances g_{ki}. In the case of linear constraints, the weights $\partial f_j(\mathbf{v})/\partial v_k$ are constants and so may be implemented directly as conductances. Combining (28), (36), and (37), one would obtain the state equations to the circuit implementation as

$$i_j = \mu g_j(f_j(\mathbf{v})) = \mu\left(\sum_{i=1}^{2L} b_{ji} v_i - e_j\right) \quad (38)$$

and

$$C\frac{dv_k}{dt} = -\sum_{i=1}^{2L}(\alpha g_{ki})v_i - \sum_{j=1}^{2} i_j b_{jk} = -\sum_{i=1}^{2L} g'_{ki} v_i - \sum_{j=1}^{2} i_j b_{jk}. \quad (39)$$

Note that the acceleration factor α is included in g_{ki} and then g'_{ki} is defined as (αg_{ki}).

According to (38) and (39), a circuit realization is shown in Fig. 3. It should be noted that the elements of the $\mathbf{e}$, $\mathbf{G}'$,

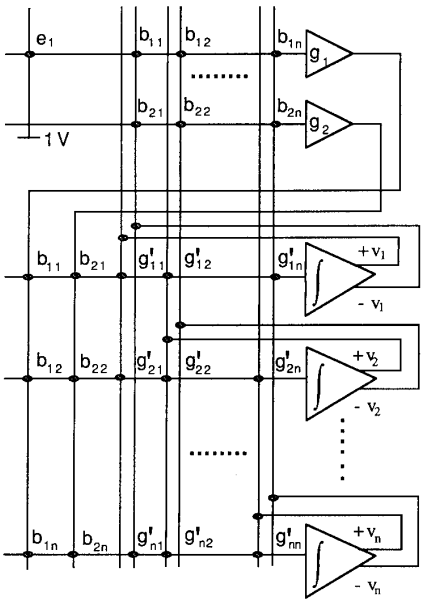


Fig. 3. Schematic diagram of neural circuit for MVDR-based beam forming problem.

and **B** matrices are realized directly as resistive connections and that their associated matrix entries correspond to conductance values. Considering the jth row of the upper constraint block, the input current I_j to the jth constraint amplifier is given by

$$\begin{aligned} I_j &= e_j + \sum_{i=1}^{2L} b_{ji}(-v_i) \\ &= e_j - \sum_{i=1}^{2L} b_{ji} v_i \\ &= -f_j(\mathbf{v}) \end{aligned} \tag{40}$$

where $e_1 = 1$ and $e_2 = 0$. Note that the resistor b_{ji} is connected to the negative output terminal of the ith variable amplifier.

Thus, we would obtain the output voltage of the jth constraint amplifier as

$$i_j = O = -\mu I_j = \mu f_j(\mathbf{v}). \tag{41}$$

Since g'_{ki} and b_{ji} are connected to the output terminal of the ith variable amplifier and the output of the jth constraint amplifier respectively, the current I_k flowing into the kth variable amplifier is obtained by

$$I_k = \sum_{i=1}^{2L} g'_{ki} v_i + \sum_{j=1}^{2} i_j b_{jk}. \tag{42}$$

By using the fact that $C dv_k/dt = -I_k$ for the kth variable amplifier, it yields

$$C\frac{dv_k}{dt} = -\sum_{i=1}^{2L} g'_{ki} v_i - \sum_{j=1}^{2} i_j b_{jk}. \tag{43}$$

As discussed in (40)–(43), it has been proven that the circuit schematic shown in Fig. 3 precisely implements the dynamic equation of (38) and (39) and then is applied to carrying out the optimal solution of the MVDR problem.

V. Illustrated Examples

To verify the effectiveness of MVDR-based neural analog circuit implementation, a linear array of ten elements with half-wavelength spacing is considered in the following examples. The variance of white noise present on each element, i.e, σ_n^2 is assumed to be equal to 0.1. In addition, there are two interference sources which fall in the main lobe and the first sidelobe of the conventional array pattern, respectively. The first interference makes an angle 72° ($\theta_1 = 72°$) with the line of the array and has the power which is taken to be 30 dB more than the white noise power. And, the second interference makes an angle of 98°($\theta_2 = 98°$) and has the power which is 10 dB more than the white noise power. Based on the above assumptions, it is concluded that the power levels for these two interferences would be identified as 100 and 1, that is, $p_1 = 100$ and $p_2 = 1$, respectively. The look direction of signal is assumed to be orthogonal to the array. Three signal powers varied from 0 to 20 dB above the white noise power are employed in our example.

The parameters g'_{ki} and b_{jk} involved in the proposed circuit for this particular example could be obtained according to the value of each component in $\mathbf{G}'$ and **B**. The $2 \times 20(L = 10)$ matrix **B** includes two 10×1 vectors associated with the look direction that are $\mathbf{S}_{0r}$ and $\mathbf{S}_{0i}$ shown in Table I. Another 20×20 symmetric positive-definite matrix $\mathbf{G}'$ consists of both $\mathbf{R}_r$ and $\mathbf{R}_i$ which are 10×10 matrices in terms of the signal power parameter p_s and the acceleration factor α. More details about the expression of $\mathbf{G}'$ are shown in Table I. In addition, the penalty multiplier μ in each constraint amplifier and the acceleration factor α are taken to be 0.4 and 10^{-3}, respectively.

We have simulated the MVDR-based neural analog circuit described by (38) and (39) with the initial guess $\mathbf{v}(0) = \mathbf{0}$, using the simultaneous differential equation solver (DVERK in the IMSL). This routine solves a set of nonlinear differential equations based on the fourth-order Runge–Kutta method. The capacitance C involved in each variable amplifier of the circuit is assumed to be 1 pF. Figs. 4 and 5 show the time evolutions of three output noise powers and the resulting output SNR's corresponding to their associated signal power levels 0.1, 1, and 10, respectively. It is worth observing that the converge time of each curve in Figs. 4 and 5 is almost independent of p_s and equal to 0.1 ns. Since the converge time is characterized by the time constant of system, one may use the dominant equivalent time constant τ, which is defined in Appendix I to verify this particular time behavior. By using the $\mathbf{G}'$ and **B** given in Table I, these dominant time constants are found to be also independent of p_s and equal to 5×10^{-9} s. It has been shown that the converge time of each curve is bounded by the dominant time constant. In addition, while reaching the equilibrium state, those resulting solution weights do not satisfy both equality constraints exactly. Appendix II shows that these two steady-state constraint viola-

TABLE I

$$\mathbf{R}_{rn} = \begin{bmatrix} 101.10 & 57.37 & -35.60 & -97.13 & -73.91 & 13.54 & 88.81 & 86.16 & 7.81 & -77.98 \\ 57.37 & 101.10 & 57.37 & -35.60 & -97.13 & -73.91 & 13.54 & 88.81 & 86.16 & 7.81 \\ -35.60 & 57.37 & 101.10 & 57.37 & -35.60 & -97.13 & -73.91 & 13.54 & 88.81 & 86.16 \\ -97.13 & -35.60 & 57.37 & 101.10 & 57.37 & -35.60 & -97.13 & -73.91 & 13.54 & 88.81 \\ -73.91 & -97.13 & -35.60 & 57.37 & 101.10 & 57.37 & -35.60 & -97.13 & -73.91 & 13.54 \\ 13.54 & -73.91 & -97.13 & -35.60 & 57.37 & 101.10 & 57.37 & -35.60 & -97.13 & -73.91 \\ 88.81 & 13.54 & -73.91 & -97.13 & -35.60 & 57.37 & 101.10 & 57.37 & -35.60 & -97.13 \\ 86.16 & 88.81 & 13.54 & -73.91 & -97.13 & -35.60 & 57.37 & 101.10 & 57.37 & -35.60 \\ 7.81 & 86.16 & 88.81 & 13.54 & -73.91 & -97.13 & -35.60 & 57.37 & 101.10 & 57.37 \\ -77.98 & 7.81 & 86.16 & 88.81 & 13.54 & -73.91 & -97.13 & -35.60 & 57.37 & 101.10 \end{bmatrix}$$

$$\mathbf{R}_{in} = \begin{bmatrix} 0.00 & -82.11 & -92.44 & -21.75 & 68.53 & 99.82 & 44.74 & -48.95 & -99.97 & -64.18 \\ 82.11 & 0.00 & -82.11 & -92.44 & -21.75 & 68.53 & 99.82 & 44.74 & -48.95 & -99.97 \\ 92.44 & 82.11 & 0.00 & -82.11 & -92.44 & -21.75 & 68.53 & 99.82 & 44.74 & -48.95 \\ 21.75 & 92.44 & 82.11 & 0.00 & -82.11 & -92.44 & -21.75 & 68.53 & 99.82 & 44.74 \\ -68.53 & 21.75 & 92.44 & 82.11 & 0.00 & -82.11 & -92.44 & -21.75 & 68.53 & 99.82 \\ -99.82 & -68.53 & 21.75 & 92.44 & 82.11 & 0.00 & -82.11 & -92.44 & -21.75 & 68.53 \\ -44.74 & -99.82 & -68.53 & 21.75 & 92.44 & 82.11 & 0.00 & -82.11 & -92.44 & -21.75 \\ 48.95 & -44.74 & -99.82 & -68.53 & 21.75 & 92.44 & 82.11 & 0.00 & -82.11 & -92.44 \\ 99.97 & 48.95 & -44.74 & -99.82 & -68.53 & 21.75 & 92.44 & 82.11 & 0.00 & -82.11 \\ 64.18 & 99.97 & 48.95 & -44.74 & -99.82 & -68.53 & 21.75 & 92.44 & 82.11 & 0.00 \end{bmatrix}$$

$$\mathbf{R}_{rs} = [1]_{10\times 10} \qquad \mathbf{R}_{is} = [0]_{10\times 10}$$

$$\mathbf{R}_r = \alpha(p_s\mathbf{R}_{rs} + \mathbf{R}_{rn}) \qquad \mathbf{R}_i = \alpha(p_s\mathbf{R}_{is} + \mathbf{R}_{in})$$

$$\mathbf{S}_{0r} = [1]^T_{1\times 10} \qquad \mathbf{S}_{0_i} = [0]^T_{1\times 10}$$

$$\mathbf{G}' = \begin{bmatrix} 2\mathbf{R}_r & -2\mathbf{R}_i \\ 2\mathbf{R}_i & 2\mathbf{R}_r \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} \mathbf{S}^T_{0_r} & \mathbf{S}^T_{0_i} \\ -\mathbf{S}^T_{0_i} & \mathbf{S}^T_{0_r} \end{bmatrix}$$

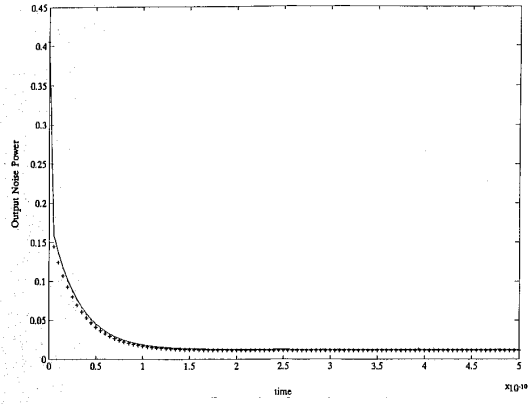


Fig. 4. Output noise power $\mathbf{W}^H\mathbf{R}_N\mathbf{W}$ versus the response time for a ten-element linear array with half-wavelength spacing. Two interferences are assumed: $\theta_1 = 72°$, $p_1 = 100$, $\theta_2 = 98°$, $p_2 = 1$. $\sigma_n^2 = 0.1$. The look direction angle is 90°. The initial values of the weights are zero. '-': $p_s = 0.1$, '$\cdots$': $p_s = 1$, '+': $p_s = 10$.

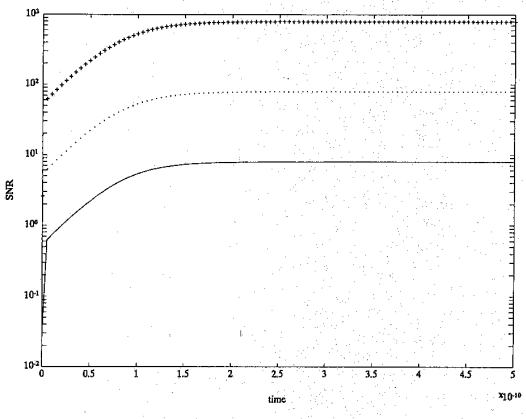


Fig. 5. Output signal-to-noise ratio versus the response time for a ten-element linear array with half-wavelength spacing. Two interferences are assumed: $\theta_1 = 72°$, $p_1 = 100$, $\theta_2 = 98°$, $p_2 = 1$. $\sigma_n^2 = 0.1$. The look direction angle is 90°. The initial values of weights are zero. '-': $p_s = 0.1$, '$\cdots$': $p_s = 1$, '+': $p_s = 10$.

tion errors could be estimated by $\hat{\text{err}}_1 = 2\alpha p_s/(\mu + \alpha p_s)$ and $\hat{\text{err}}_2 = 0$. As a result, the penalty multiplier should be chosen large enough to make $\hat{\text{err}}_1$ sufficiently small. The final results are illustrated in the following table.

Signal Power	Theoretical SNR Value	Simulation Results $\mathbf{v}(0) = \mathbf{0}$	Weighting α	Weighting μ	Constraint Error err_1	Constraint Error err_2	Error Estimates $\hat{\text{err}}_1$	Error Estimates $\hat{\text{err}}_2$
0.1	7.956	7.956	0.001	0.4	0.00056	0	0.0005	0
1	79.56	79.56	0.001	0.4	0.005	0	0.005	0
10	795.6	795.6	0.001	0.4	0.048	0	0.0488	0

It is worth simulating the case of large jammers by varying the INR's (interference-to-noise ratio) for both interfering signals from 30–70 dB simultaneously. Fig. 6 shows that the resulting output SNR of each curve is almost independent of INR and reaches the same steady state level. Since the feedback current into the variable amplifier is approximately proportional to the INR for the case of large jammers, it is expected that the converge time improves as INR increases. However, the acceleration factor is adjusted to smaller value in view of stability consideration and the dominant time constant in such case is about 10^{-5} s. Another situation of particular interest is the performance evaluation of the beamformer under the influence of broad-band jammers. This is given by way of examples which demonstrate how the resulting SNR varies as the element spacing changed from $0.35\ \lambda_0$

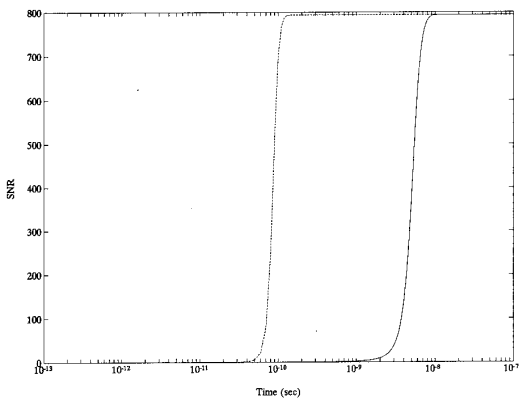


Fig. 6. Output signal-to-noise ratio versus the response time for 10 element linear array with half-wavelength spacing. Two interferences are assumed: $\theta_1 = 72°$, $\theta_2 = 98°$. $\sigma_n^2 = 0.1$. The look direction angle is 90°, $p_s = 10$. The initial values of the weights are zero. '-': INR = 30 dB, '--': INR = 50 dB, ' $\cdots$ ': INR = 70 dB.

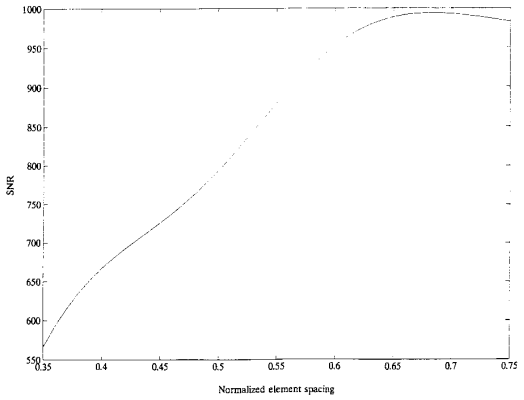


Fig. 7. Output signal-to-noise ratio versus the element spacing for a ten-element linear array. Two interferences are assumed: $\theta_1 = 72°$, $p_1 = 100$, $\theta_2 = 98°$, $p_2 = 100$. $\sigma_n^2 = 0.1$. The look direction angle is 90°, $p_s = 10$. The initial values of the weights are zero.

to 0.75 λ_0, where λ_0 is the wavelength and the signal source corresponds to the half-wavelength spacing. For comparison, all the parameter settings are taken to be the same as used in the previous case. One observes from Fig. 7 that the SNR starts to increase as the element spacing increases. It reaches its maximum level when the element spacing is equal to 0.68 λ_0. Beyond that the SNR level drops slightly. The reason is that the larger element spacing causes the grating lobes to appear in the array and thus degrades the overall output SNR. Finally, a comprehensive example is conducted to test the network performance for the case of closely spaced jammers. The phases of the interfering signals are assumed such that the angles of arrival for the two jammers are as close as $\theta_1 = 72°$ and $\theta_2 = 74°$, respectively. The powers of both interferences are taken to be of the same level and equal to 100. The signal power, look direction angle and element spacing are suggested to be 10, 90° and 0.5 λ_0 respectively. The resultant array output SNR is shown in Fig. 8 which reaches its maximum value of 950 rapidly. Fig. 9 compares the power patterns of the resultant adaptive array pattern with that of the conventional uniform array pattern. One observes clearly that two sharp nulls presented in the pattern correspond to the directions of arrival of the interferences. As a result, the interferences are suppressed and the output SNR of the adaptive array reaches the optimal value.

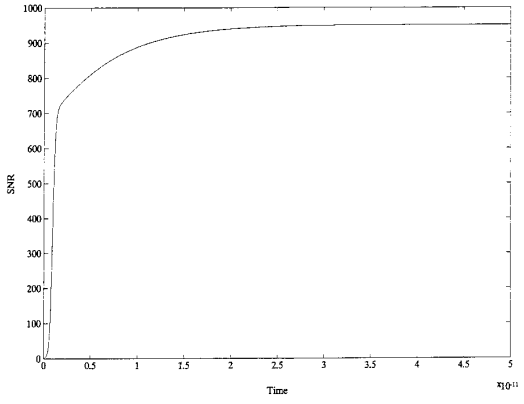


Fig. 8. Output signal-to-noise ratio versus the response time for a ten-element linear array with half-wavelength spacing. Two interferences are assumed: $\theta_1 = 72°$, $p_1 = 100$, $\theta_2 = 74°$, $p_2 = 100$. $\sigma_n^2 = 0.1$. The look direction angle is 90°, $p_s = 10$. The initial values of the weights are zero.

VI. Conclusion

In this paper, we have proposed a cost-effective analog circuit implementation for computing the MVDR beamforming problem based on Kennedy and Chua's cannonical neural network. Their novel neural-based optimizer is able to guarantee the stability and robustness of the solution, the converge time can be characterized by the dominant time constant of the network. It turns out that the speed of reaching a steady-state solution depends essentially on the time constant, not on the algorithmic time complexity. Finally, a linear array of 10 elements with three signal levels is constructed accordingly to verify the performance of the proposed cir-

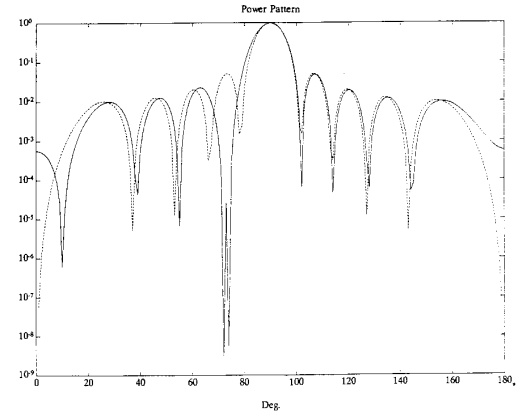


Fig. 9. The conventional uniform array pattern '--' and the resultant adaptive array pattern '-' for a 10-element linear array with half-wavelength spacing. Two interferences are assumed: $\theta_1 = 72°$, $p_1 = 100$, $\theta_2 = 74°$, $p_2 = 100$. $\sigma_n^2 = 0.1$. The look direction angle is 90°, $p_s = 10$. The initial values of the weights are zero.

cuit. It shows that the MVDR-based neural circuit is able to quickly attain its optimal performance in 0.1 ns when the dominant time constant is 5×10^{-9} s and work satisfactorily under the stringent environment of strong jammers as well as closely spaced jammers.

Appendix I
The Dominant Time Constant for an MVDR-Based Neural Circuit

Substituting (38) into (39), one may obtain the system of first-order differential equations in matrix form and given by

$$\frac{d\mathbf{v}}{dt} = -\frac{1}{C}\left[(\mathbf{G}' + \mu\mathbf{B}^T\mathbf{B})\mathbf{v} + \mu\mathbf{B}^T\mathbf{e}\right] = -\mathbf{M}\mathbf{v} + \mathbf{N}\mathbf{e} \tag{44}$$

where $\mathbf{M} = (\mathbf{G}' + \mu\mathbf{B}^T\mathbf{B})/C$ and $\mathbf{N} = -\mu\mathbf{B}^T/C$.

[11] showed that the solution of (44) can be expressed as a linear combination of modes $t^k e^{-\lambda_i t}$ where λ_i is the ith eigenvalue of $\mathbf{M}$. It is known that the time behavior of system (44) would be dominated by the minimum eigenvalue $\lambda_{\min}(\mathbf{M}) = \min_i(\lambda_i)$. Consequently, a dominant time constant for the MVDR-based neural circuit is defined by

$$\tau = \frac{1}{\lambda_{\min}(\mathbf{M})}. \tag{45}$$

Appendix II
Estimation of Steady-State Constraint Violation Errors

Let the steady-state constraint violation errors for both equality constraints defined in (25) be represented by err_1 and err_2, respectively. While reaching the equilibrium state, these two constraints become

$$\sum_{k=1}^{2L} B_{1k}v_k = (1 - \mathrm{err}_1) \tag{46}$$

and

$$\sum_{k=1}^{2L} B_{2k}v_k = \mathrm{err}_2. \tag{47}$$

Since the noise power p_{noise} is assumed to be insignificant to the signal power, the objective function ϕ is given by

$$\phi = p_s\left[(1 - \mathrm{err}_1)^2 + \mathrm{err}_2^2\right] + p_{\text{noise}} \approx p_s\left[(1 - \mathrm{err}_1)^2 + \mathrm{err}_2^2\right] \tag{48}$$

and the penalty function is calculated as

$$P = \tfrac{1}{2}\left(\mathrm{err}_1^2 + \mathrm{err}_2^2\right). \tag{49}$$

Therefore, the energy function becomes

$$E\,|_{\text{steady state}} = \alpha\phi + \mu P = \tfrac{1}{2}\left\{2\alpha p_s(1 - \mathrm{err}_1)^2 + \mu\mathrm{err}_1^2 + (2\alpha p_s + \mu)\mathrm{err}_2^2\right\}. \tag{50}$$

By the chain rule and the fact that $dE/dt = 0$ at the equilibrium state, the time derivative of E becomes

$$\left.\frac{dE}{dt}\right|_{\text{steady state}} = \frac{\partial E}{\partial \mathrm{err}_1}\cdot\frac{d\mathrm{err}_1}{dt} + \frac{\partial E}{\partial \mathrm{err}_2}\cdot\frac{d\mathrm{err}_2}{dt} = 0. \tag{51}$$

Equation (51) implies that both partial derivatives of E with respect to err_1 and err_2 should vanish respectively, i.e.,

$$\left.\frac{\partial E}{\partial \mathrm{err}_1}\right|_{\mathrm{err}_1 = \hat{\mathrm{err}}_1} = -2\alpha p_s(1 - \hat{\mathrm{err}}_1) + \mu\hat{\mathrm{err}}_1 = 0 \tag{52}$$

and

$$\left.\frac{\partial E}{\partial \mathrm{err}_2}\right|_{\mathrm{err}_2 = \hat{\mathrm{err}}_2} = (2\alpha p_s + \mu)\hat{\mathrm{err}}_2 = 0 \tag{53}$$

where $\hat{\mathrm{err}}_1$ and $\hat{\mathrm{err}}_2$ are the estimates of err_1 and err_2, respectively. Then, we obtain

$$\hat{\mathrm{err}}_1 = \frac{2\alpha p_s}{\mu + \alpha p_s} \tag{54}$$

and

$$\hat{\mathrm{err}}_2 = 0. \tag{55}$$

References

[1] S. P. Applebaum, "Adaptive array," *IEEE Trans. Antennas Propagat.*, vol. AP-24, pp. 585–598, Sept. 1976.

[2] B. Widrow and S. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall; 1985.

[3] B. D. Van Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE ASSP Mag.*, pp. 4–24, Apr. 1988.

[4] L. C. Godara, "Error analysis of the optimal antenna array processors," *IEEE Trans. Aerospace Electron. Syst.*, vol. AES-22, pp. 395–409, July 1986.

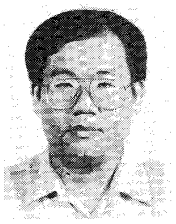
[5] J. G. McWhirter and T. J. Shepherd, "Systolic array processor for MVDR beamforming," *Proc. Inst. Elec. Eng.*, vol. 136, pt. F, no. 2, Apr. 1989.

[6] W. M. Gentleman and H. T. Kung, "Matrix triangularisation by systolic array," *Proc. SPIE*, Real time signal processing IV, 1981, p.298.

[7] D. W. Tank and J. J. Hopfield, "Simple 'neural' optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit," *IEEE Trans. Circuit Syst.*, vol. CAS-33, pp. 533–541, May 1986.

[8] L. O. Chua and G. N. Lin, "Nonlinear programming without computation," *IEEE Trans. Circuit Syst.*, vol. CAS-31, pp. 182–188, Feb. 1984.

[9] M.P. Kennedy and L. O. Chua, "Unifying the Tank and Hopfield linear programming network and the canonical nonlinear programming circuit of Chua and Lin," *IEEE Trans. Circuit Syst.*, vol. CAS-34, pp. 210–214, Feb. 1987.
[10] ——, "Neural networks for nonlinear programming," *IEEE Trans. Circuit Syst.*, vol. 35, pp. 554–562, May 1988.
[11] C. T. Chen, *Linear System Theory and Design*. New York: Holt, Rinehart and Winston, 1984.
[12] D. G. Luenberger, *Linear and Nonlinear Programming, 2nd ed.* Reading, MA: Addison-Wesley, 1984.

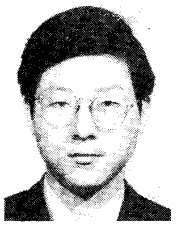


Po-Rong Chang (M'87) received the B.S. degree in electrical engineering from National Tsing-Hua University, Taiwan, Republic of China, in 1980, the M.S. degree in telecommunication engineering from National Chiao-Tung University, Hsinchu, Taiwan, in 1982, and the Ph.D. degree from Purdue University, West Lafayette, IN, in 1988.

From 1982 to 1984, he was a lecturer in the Chinese Air Force Telecommunication and Electronics School for his two-year military service. From 1984 to 1985, he was an instructor of electrical engineering at National Taiwan Institute of Technology, Taipei, Taiwan. From 1989 to 1990, he was a project leader of SPARC chip design team at ERSO of Industrial Technology and Research Institute, Chu-Tung, Taiwan, Republic of China. Currently he is an Associate Professor of Communication Engineering at National Chiao-Tung University. His current interests include neural network, HDTV signal processing, and human visual and audio system.

Wen-Hao Yang was born in Chyi, Taiwan, Republic of China, on June 12, 1964. He received the M.S. degree in communication engineering from the National Chiao-Tung University, Hsinchu, Taiwan in 1988, where he is currently pursuing the Ph.D. degree.

From 1984 to 1986 he fulfilled his military duty by serving in the Chinese Air Force. Since 1988, he has been an assistant research engineer at the Telecommunication Laboratories, Ministry of Communications, Republic of China, where he works on satellite communication ground station establishment and microwave filter design. His research interests are in adaptive array, pattern synthesis, and microwave circuits.



Kuan-Kin Chan (M'88) was born in China, in 1943. He received the B.S. degree in electrical engineering from Cheng Kung University in 1966, the M.S. degree in electronics from Chiao Tung University, Hsinchu, in 1969, and the Ph.D. degree in electrophysics from Polytechnic Institute of Brooklyn, Brooklyn, NY, in 1976.

He is now a Professor in the Department of Communication Engineering, National Chiao Tung University, Taiwan, Republic of China. His current research interests are in the area of electromagnetic scattering theory and applications.